DATACOM PRODUCTS

DATABOOK

1st EDITION

SGS-THOMSON
MICROELECTRONICS

DATACOM PRODUCTS

DATABOOK

1st EDITION

DECEMBER 1989

USE IN LIFE SUPPORT DEVICES FOR SYSTEMS MUST BE EXPRESSLY AUTHORIZED

SGS-THOMSON PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF SGS-THOMSON Microelectronics. As used herein:

1. Life support devices or systems are those which (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided with the product, can be reasonably expected to result in significant injury to the user.

2. A critical component is any component of a life support device or system whose failure to perform can reasonably be expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

TABLE OF CONTENTS

AT SGS-THOMSON THE FUTURE OF DATA COMMUNICATION IS NOW.

Pin-for-pin compatibility between key devices in both LAN and WAN applications enables you to develop new products in considerably less time, at dramatically lower costs. Essentially, your investment is reduced to one hardware and software design effort. This same compatibility will be maintained whenever possible through the evolution of future SGS-THOMSON data communication products.

Our commitment to service is just as strong as our commitment to technology. Ontime delivery is one impressive example. We have the manufacturing capacity, quality and technical support needed tomorrow, today.

Four and six-inch wafer production capacity at our Dallas facility is just a part of the extensive billion-dollar SGS-THOMSON manufacturing machine that spans the globe. The Dallas plant utilizes 1.2 and 1.5 micron CMOS technology to produce data communication ICs in large quantities.

At SGS-THOMSON quality is a corporate-wide mission in which every employee shares responsibility. Demanding programs insure strict compliance with the quality control procedures essential to service you, our customer.

Technical support is also a key determinate of success. SGS-THOMSON application engineers around the world are ready to assist you with any problems that might arise during the design cycle.

Regardless of your needs, you can expect the same close cooperation, manufacturing quality and advanced technology that have made SGS-THOMSON Microelectronics a one billion dollar company and one of the world's leading semiconductor suppliers.

ALPHANUMERICAL INDEX

PRODUCT GUIDE

DEVICE	DESCRIPTION	WIDTH (MILS)	CERAMIC	PLASTIC	SURFACE MOUNT	Page
MK2180A	T1 TRANSCEIVER	600	_	40-PIN DIP	PLCC 44	277
MK5025	X.25 CONTROLLER	600	48-PIN DIP	_	PLCC 52	205
MK5027	CCS# 7 CONTROLLER	600	48-PIN DIP	_	PLCC 52	255
MK5030	STARLAN HUB	600	48-PIN DIP	48-PIN DIP	PLCC 52	9
MK5032	ETHERNET/ STARLAN V-LANCE CONTROLLER	600	48-PIN DIP	–	PLCC 52 PLCC 68	25
MK5033	MANCHESTER ENC./DEC.	300	28-PIN DIP	28-PIN DIP	_	85
MK5035	MANCHESTER ENC./DEC.	300	_	20-PIN DIP	_	99
MK50351	MANCHESTER ENC./DEC.	300	_	20-PIN DIP	_	111
MK68564	SERIAL COMMUNICATIONS CONTROLLER	600	48-PIN DIP	48-PIN DIP	PLCC 52	307
MK68592	ETHERNET SIA	300	CERDIP 24-PIN DIP	–	–	123
MK68901	MULTI FUNCTION PERIPHERAL	600	48-PIN DIP	48-PIN DIP	PLCC 52	353
MK85C30	SERIAL COMMUNICATION CONTROLLER	600	_	40-PIN DIP	PLCC 44	387
TS68HC901	HCMOS MULTI FUNCTION PERIPHERAL	600	_	48-PIN DIP	PLCC 52	481
Z8530	SERIAL COMMUNICATION CONTROLLER	600	40-PIN DIP	40-PIN DIP	PLCC 44	527

LOCAL AREA NETWORK DEVICES

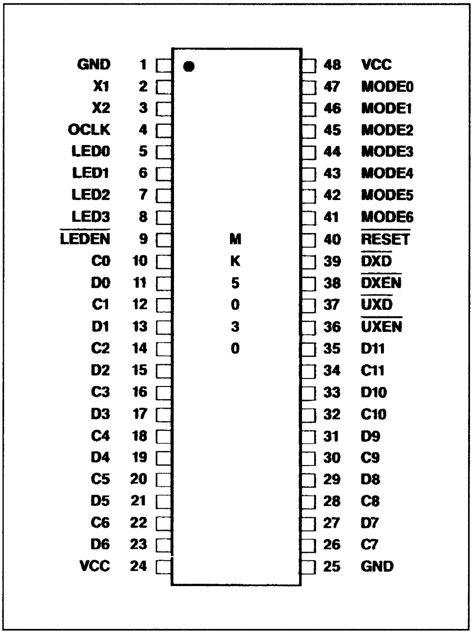
SGS-THOMSON
MICROELECTRONICS

MK5030P/N

StarLAN HUB CHIP
COMMUNICATIONS PRODUCTS

- COMPLETE HUB LOGIC DEVICE COMPATIBLE TO STARLAN SPECIFICATION
- SUPPORTS MULTI-POINT EXTENSION (MPE)
- AUTO COMPENSATION FOR WIRING REVERSAL
- 12 PORT HUB
- OPTIONAL RETIME CIRCUIT
- CASCADABLE. TWO LEVELS MAY BE CASCADED AND STILL APPEAR TO THE NETWORK AS ONE YIELDING UP TO A 121 PORT HUB
- AUTO PREAMBLE GENERATION TO MINIMIZE BIT LOSS
- SELECTABLE ACTIVE CARRIER POLARITY SENSE
- JABBER FUNCTION ISOLATES NETWORK FAILURES
- OPTIONAL MINIMUM FRAME LENGTH ENFORCEMENT
- COLLISION DETECTION
  - multiple inputs
  - missing mid-bit transition
  - transitions too close together
  - transitions too far apart
  - AT & T Release 1 Collision presence signal
- 6X CLOCK YIELDS 167NS JITTER TOLERANCE
- TRANSMIT DATA TRAILER ENFORCEMENT
- INPUT PROTECTION AT END OF FRAME (20µs)
- PIN SELECTABLE HIGH-END HUB VERSUS INTERMEDIATE HUB
- OPTIONAL INTERNAL PULSE STRETCHER FOR CARRIER SENSE SQUELCH
- UPLINK AND DOWNLINK ACTIVE STATUS OUTPUTS
- UPLINK AND DOWNLINK COLLISION STATUS OUTPUTS
- PER PORT JABBER STATUS OUTPUT
- ON CHIP CRYSTAL OSCILLATOR CIRCUITRY
- 35mW TYPICAL POWER DISSIPATION
- CMOS TECHNOLOGY
- 48 PIN DIP
- SINGLE 5-VOLT SUPPLY
- ALL INPUTS AND OUTPUTS TTL COMPATIBLE*
- INDUSTRIAL VERSION AVAILABLE

Figure 1 : MK5030 Pin Assignment.



DESCRIPTION

The MK5030-HUB is a 48 pin CMOS VLSI device that simplifies the design and implementation of a StarLAN compatible HUB. This chip provides all the digital logic necessary in a HUB.

PIN DESCRIPTION

X1, X2 Inputs. Either connect a 6.00 ± 0.05% MHz crystal between the pins, or leave X2 not connected and apply a 6.0MHz ± .01% square wave to X1. (Refer to figure 7).

OCLK Output. Provides a TTL CLK output from the above crystal oscillator. This is useful when cascading devices.

C0-C11 Schmitt inputs. Carrier sense inputs. Active state is selectable. Refer to MODE4 below.

D0-D11 Inputs. Received data streams.

$\overline{\text{UXD}}$ Output. UPLINK data stream to the next UPPER HUB, if any, in the network hierarchy.

$\overline{\text{UXEN}}$ Output. UPLINK transmit enable indicating that $\overline{\text{UXD}}$ contains valid data.

$\overline{\text{DXD}}$ Output. Downlink data stream to all ports. If MODE1 = 1, then this data stream is derived from Port 11. If MODE1 = 0 then this data stream is identical to $\overline{\text{UXD}}$.

$\overline{\text{DXEN}}$ Output. Downlink transmit enable to all ports. If MODE1 = 1, then this data stream is derived from Port 11. If MODE1 = 0, then this data stream is identical to $\overline{\text{UXEN}}$.

MODE0 Input. Testmode. Should be tied high for normal operation. Testmode is useful only for semiconductor device production test.

MODE1 Input. If MODE1 = 1, then the chip is an intermediate hub ; and C0-10/D0-10 inputs will be used as UPLINK inputs and C11/D11 as the DOWNLINK input. If MODE1 = 0, then the chip is a high-end hub ; C0-11/D0-11 inputs will be used as UPLINK inputs and DOWNLINK outputs ($\overline{\text{DXD}}$, $\overline{\text{DXEN}}$) will internally be connected to the UPLINK outputs ($\overline{\text{UXD}}$, $\overline{\text{UXEN}}$). See figure 5.

MODE2 Inputs. If MODE2 = 0, then the input carriers have been externally stretched and no internal stretcher is desired. This stretcher is required as part of the squelch function. Refer to figure 2. If MODE2 = 1, then the input carriers are threshold set off the incoming data stream and an internal pulse stretcher must be used. With MODE2 = 1, users do not need to use one-shots externally for each port.

MODE3 Input. If MODE3 = 1, then the retimer circuit is enabled as specified by IEEE 802.3. If MODE3 = 0, then the retimer circuit is disabled. This allows HUBs in close physical proximity to be cascaded together and appear to the network as one HUB. Refer to figure 3.

MODE4 Input. Selects carrier active state. If MODE4 = 0, then the carrier inputs (C0-C11) are active low. If MODE4 = 1 then the carrier inputs (C0-C11) are active high.

MODE5 Input. If MODE5 = 1, then auto preamble generation is enabled. The chip will initiate preamble transmission once phase lock is obtained. Once the transmit FIFO is sufficiently full, transmit data is obtained from the FIFO. This decreases the amount of bits a HUB implementation will lose. Depending on the implementation, bits may actually be gained. If MODE5 = 0, then automatic preamble generation is disabled.

MODE6 Input. If MODE6 = 1, then minimum frame length of 96 bits is enforced. If MODE6 = 0, then no minimum frame length is enforced.

$\overline{\text{RESET}}$ Schmitt input. Active low. A low causes the device to reset. Input must be high for normal operation.

$\overline{\text{LEDEN}}$ Input/output open drain. If $\overline{\text{LEDEN}}$ is externally connected to GND, then the LED (0-3) provide static status information. Otherwise, $\overline{\text{LEDEN}}$ should be pulled high through a 2Kohm pullup resistor and LED (0-3) will provide an ID of an internal status monitoring point and will pull $\overline{\text{LEDEN}}$ low if that monitoring point is active. This allows connection of a single external multiplexer to drive indicating devices. Refer to figure 4.

LED0-LED3 Output. If $\overline{\text{LEDEN}}$ is connected to GND :

LED0 UPLINK transmit enabled
LED1 UPLINK collision sense
LED2 DOWNLINK transmit enabled
LED3 DOWNLINK collision sense.

If $\overline{\text{LEDEN}}$ is not connected to GND, then $\overline{\text{LEDEN}}$ = 0 indicates that the LED (0-3) specified function is active :

LED3	LED2	LED1	LED0	Description	$\overline{\text{LEDEN}}$ = 0	$\overline{\text{LEDEN}}$ = 1
0	0	0	0	PORT 0 jabber	active	inactive
0	0	0	1	PORT 1 jabber	active	inactive
0	0	1	0	PORT 2 jabber	active	inactive
0	0	1	1	PORT 3 jabber	active	inactive
0	1	0	0	PORT 4 jabber	active	inactive
0	1	0	1	PORT 5 jabber	active	inactive
0	1	1	0	PORT 6 jabber	active	inactive
0	1	1	1	PORT 7 jabber	active	inactive
1	0	0	0	PORT 8 jabber	active	inactive
1	0	0	1	PORT 9 jabber	active	inactive
1	0	1	0	PORT 10 jabber	active	inactive
1	0	1	1	PORT 11 jabber	active	inactive
1	1	0	0	UPLINK	inactive	active
1	1	0	1	UPLINK Collision	yes	no
1	1	1	0	DOWNLINK	inactive	active
1	1	1	1	DOWNLINK Collision	yes	no

Note : 1. UPLINK and DOWNLINK status outputs are normally on and will blink off for 147mS when a frame is transmitted. LED on-time of 147mS is guaranteed between each off blink.
2. UPLINK and DOWNLINK collision outputs are normally off and will blink on for 147mS when a collision is detected. LED off time of 147ms is guaranteed between each on blink.

VCC Power supply pin. + 5 VDC ± 5%

GND Ground. 0 VDC.

Figure 2 : Internal Versus External Time Squelch.

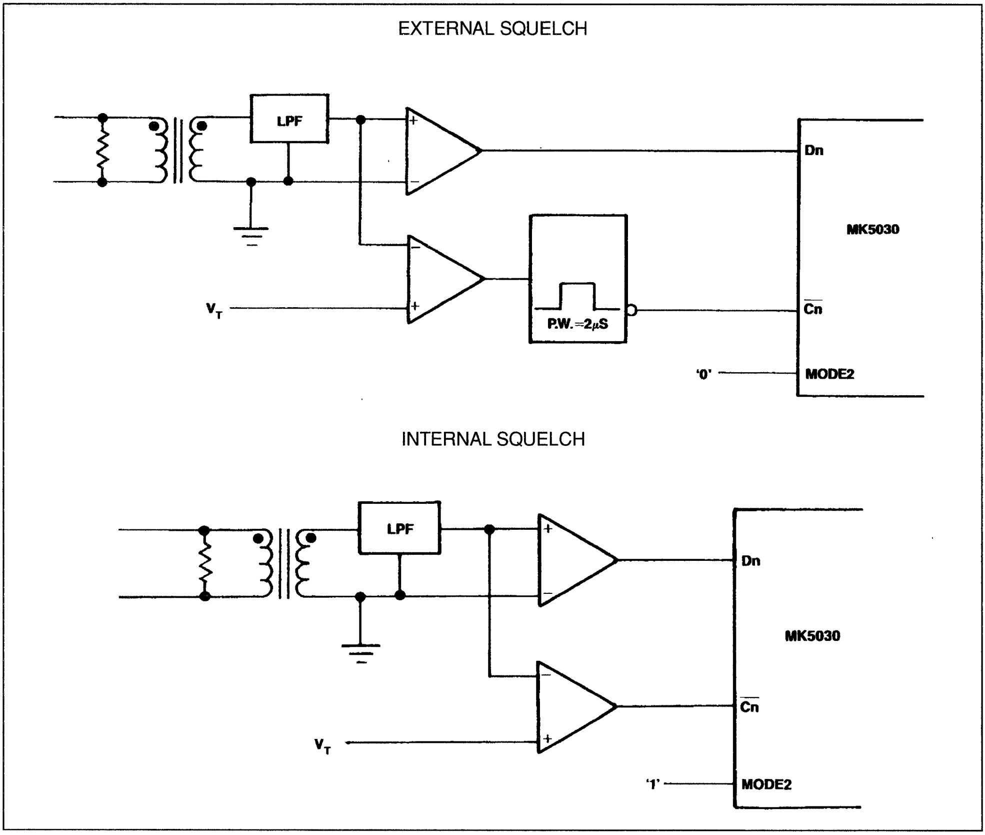
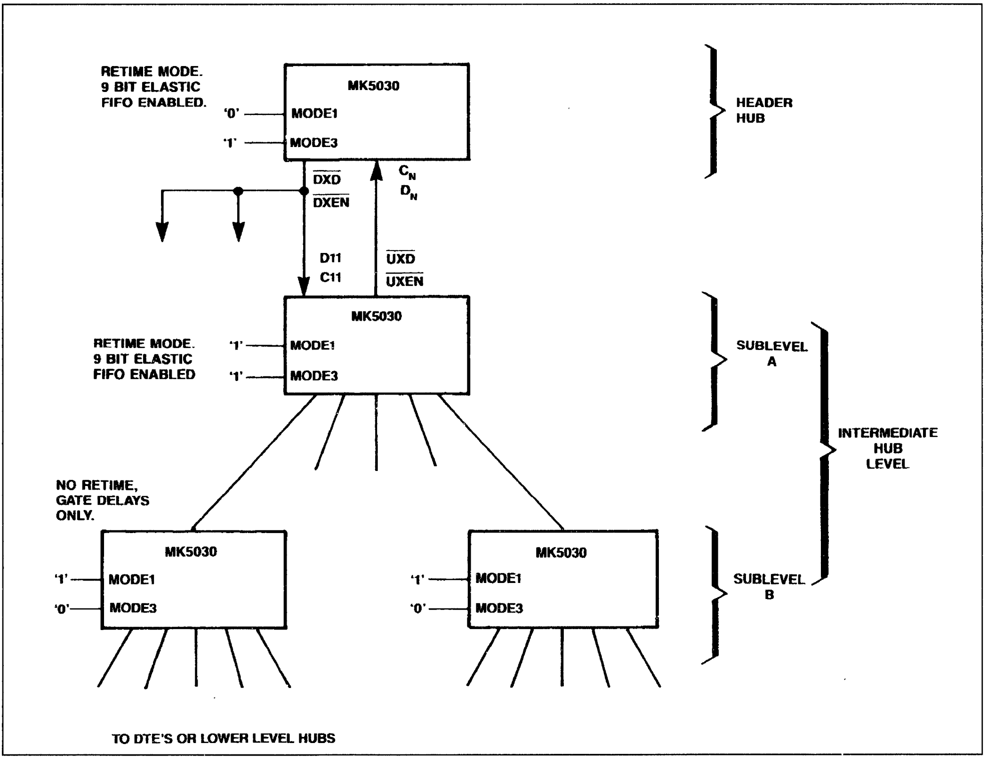
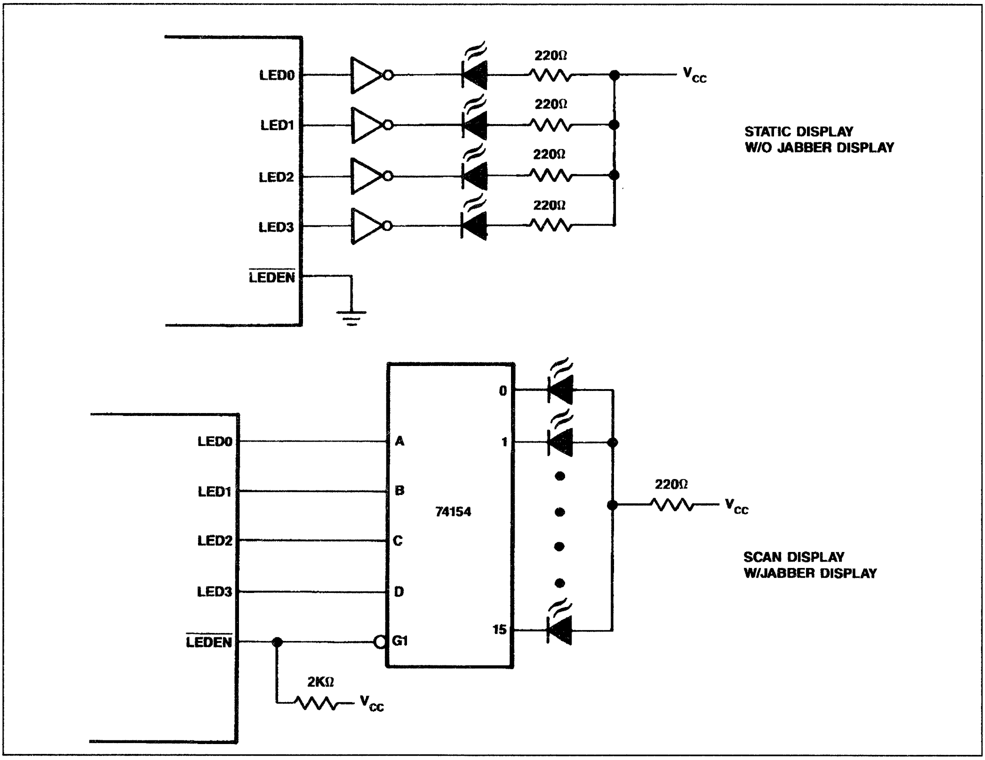


Figure 3 : Example Showing Retime and No-retime Modes.



Note : Sublevels A and B may either be on same circuit board or on separate boards located in close proximity.

Figure 4 : Status Display Modes.



OVERVIEW OF CIRCUIT DESCRIPTION

The MK5030 HUB chip consists of three sub-modules : the uplink, the downlink, and the status display module.

The uplink module multiplexes twelve inputs from stations and/or "lower" HUBS (see figure 3) and retimes the multiplexed data to remove jitter. The uplink also handles several optional features including retiming, disabled auto-preamble generation, and collision detection/transmission. Refer to figure 5, MK5030 HUB Block Diagram.

The downlink module is used only in the intermediate mode, and is nearly identical to the uplink (see figure 3).

The status display modules provide both static and scanned display of the line activity, detected collisions, and "jabbed" inputs.

THE UPLINK MODULE

The uplink module has a carrier processor which performs the following :

- Detects carrier and outputs a carrier presence signal.
- Detects collision and outputs a collision presence signal.
- Will ignore one, or more, inputs by the jabber or protection time functions.
- Provides time domain filtering to improve noise tolerance on carrier inputs.
- When in the retime mode :
  - Automatically compensates for wiring reversal.
  - Recovers clock using a DPLL (for internal HUB chip use only).
  - Passes data through a serial 9 bit FIFO buffer.

  - When in minimum frame length mode, the frame length is guaranteed to be greater than 96 bits.
  - Will perform Automatic Preamble Generation (APG) if the optional APG is selected.
  - Detects end of-frame using the DPLL.
- When NOT in the retime mode, the DPLL, APG and FIFO are bypassed, and the output is taken from the selected input without any flip-flop delays (gate delays only).

CARRIER AND DATA INPUTS

When an input has a signal present, the carrier will be detected on the appropriate pin (CO-C11). The carrier input is user selected for either external squelch (with external one-shots) or internal squelch (a 2µS pulse stretcher is added by the HUB chip). MODE2 = 0 selects external squelch, and MODE2 = 1 selects internal.

Note, a carrier input must be active for at least three clock samples for it to be recognized by the chip. Any isolated pulse less than three clock samples wide will be ignored. However, when using internal carrier squelch, the carrier must be active for at least one clock sample time every 2µS to be considered valid beyond the initial carrier recognition. When not using internal squelch, the carrier must be active during the entire frame to avoid data loss. Ignoring carrier spikes provides extra noise protection and is also referred to as time domain filtering (TDF).

RETIME MODE

The selected valid data input is fed to a digital phase locked loop (DPLL). The DPLL is implemented with a counter which clears on each transition. This gives a jitter tolerance of 167ns peak to peak (83ns peak). This is 40% more tolerance than required by the StarLAN specification.

The valid input is passed through a 9 bit FIFO. Output from the FIFO is prevented until the FIFO is 4 bits full. This is called the 4 bit watermark, and gives the FIFO 4 bits of elasticity which is more than sufficient to absorb the allowable 0.01% clock tolerance.

AUTOMATIC PREAMBLE GENERATOR

If APG is enabled, preamble generation at the outputs DXD and UXD is started as soon as the DPLL acquires lock. When the FIFO reaches the 4 bit watermark, the output data are taken from the FIFO. The APG keeps the preamble from "shrinking" as a frame is passed from HUB to HUB. Without APG, a HUB chip will lose an average of 2 preamble bits, but with APG, an average of 2 preamble bits are gained. This two bit gain should be taken into account by system designers.

AUTOMATIC COMPENSATION FOR WIRING REVERSAL

When installing twisted pair telephone wiring, it is often difficult and expensive to maintain proper polarity on the wire pairs. The MK5030 will automatically compensate for this reversal on a per port basis. Any frame that is received with inverse polarity will be detected and will be transmitted on the DXD and UXD pins with the correct polarity. This polarity compensation is active only while in the retime mode.

MINIMUM FRAME LENGTH ENFORCEMENT

When minimum frame length enforcement (FLE) is enabled (MODE6 = 1) and the MK5030 is in retime mode, the input carrier is assumed to be valid for at least 96 bit times. If either the incoming carrier goes inactive or EOF is detected prior to 96 bit times, the MK5030 will send collision presence (CP) for the remainder of the 96 bit times. This feature is important in multi-port environments where signal superposition may cause early carrier dropout. NOTE : IEEE standards committee is, at the printing, still defining FLE. The actual length guaranteed is subject to change.

NO-RETIME MODE

If the retime mode is disabled, then the DPLL, FIFO, and APG are bypassed. The outputs, DXD and UXD,are taken from the selected input without clocked delays (i.e., flip-flops). There are gate delays only. End-of-frame is detected by a counter instead of the DPLL. The advantage of the no-retime mode is that two HUBs cascaded together will appear to the network as one. See figure 3. In no-retime mode, an average of 2 bits will be lost as outputs are enabled after the first rising edge of the incoming data.

PROTECTION TIME

At the end of each frame, all carrier inputs are ignored for 20µS. This is called the protection time and insures immunity to post end-of-frame spikes caused by transformer coupling.

Figure 5 : MK5030 HUB Block Diagram.

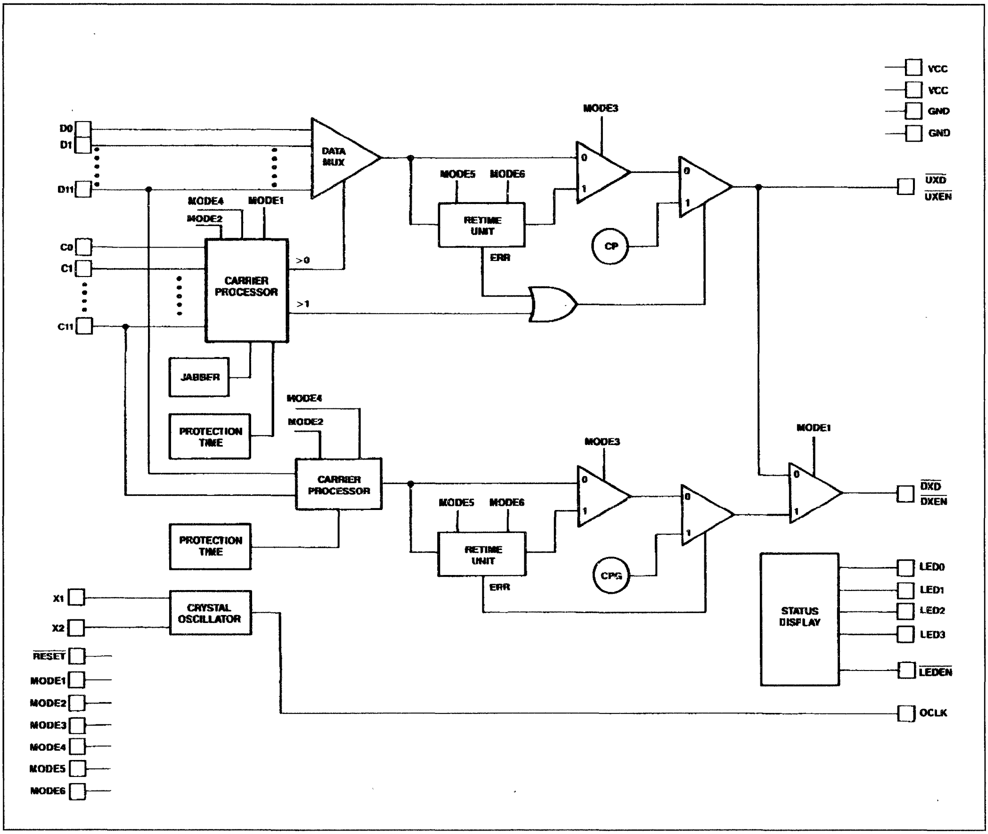
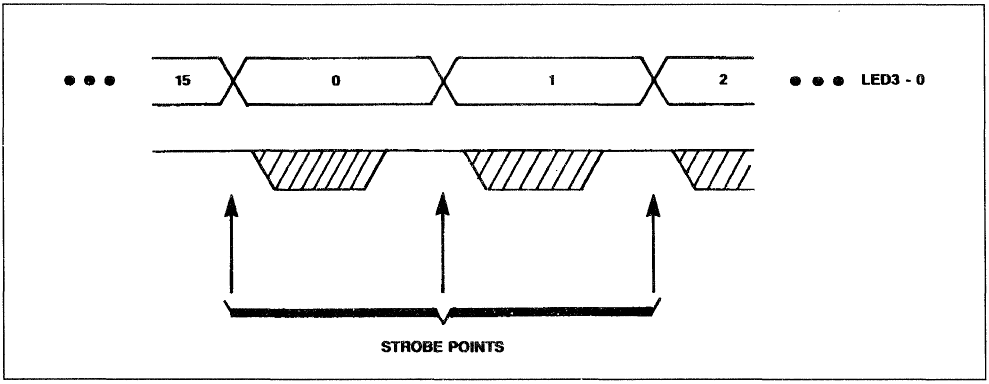


Figure 6 : Scan Display Timing.



COLLISION

A collision is defined when any one of the following five conditions exists : a. Multiple carrier inputs ; b. Manchester code violations ; c. FIFO underflow/overflow exception. d. PLL lock acquisition timeout ; e. frame length violation. An IEEE defined collision presence (CP) code is placed on the UXD output. The starting point of the CP sequence is adjusted to allow faster CP detection by the remote station or HUB. The HUB chip is fully upward compatible with the existing AT&T release 1 CP signal.

JABBER

If a station transmits data for greater than 27mS (which allows twice its normal maximum frame size) then the HUB will output a CP signal. This should correct the error in the station in most cases. However, if that station continues to transmit for up to a total time of 54mS, then the station is "jabbed". This means that the jabber function in the HUB chip will ignore that input and, in effect, remove the station from the network. If the "jabbed" station goes silent and then is the originator of new data, the station is allowed back onto the network by the HUB chip.

Thus, StarLAN networks automatically adjust as portions of the network fail and are repaired.

THE DOWNLINK MODULE

The downlink is identical to the uplink except without the multiplexer, frame length enforcement, and jabber functions. In the intermediate mode (MODE1 = 1), port 11 is a downlink input connected to the next "higher" HUB downlink output. See fig. 3. In the high-end mode (MODE1 = 0), all twelve ports feed the uplink, and DXD-DXEN are internally connected to UXD-UXEN.

STATUS DISPLAY

Two display modes are supported : static and scan. When LEDEN is externally tied to GND, then LEDO - LED3 provide static status information. When LEDEN is externally tied to VCC through a pull-up resistor, then an external demultiplexer (such as a 74154) may be used to provide 16 lines of status information. See Figures 4 and 6, and also see the description of pins LED0 - LED3.

OSCILLATOR

The MK5030 will accept two forms of clock input : a CMOS input or a crystal. If pin X2 is left unconnected, a 6.0MHz ± 0.01% CMOS clock may be applied to pin X1. Alternately, a crystal circuit may be connected between X1 and X2 to form the basis of an oscillator. Typically, a 6.0 ± 0.005% parallel resonant crystal is needed to insure the ± 0.01% frequency accuracy required for StarLAN. Refer to figure 7. A fundamental mode, parallel resonance type crystal should be used with the manufacturer's suggested load capacitance.

OCLK provides a TTL CMOS level clock output useful for cascading HUB chips or driving surrounding logic.

RESET INPUT

The reset pin is an active low Schmidt trigger input. A simple RC network may be used to insure correct operation upon power-up. In addition, if the mode inputs are changed after the application of power, reset must be reapplied. If either MODE1 or MODE3 is changed, an internal reset is generated automatically. Refer to fig. 8.

Figure 7 : Oscillator Operation.

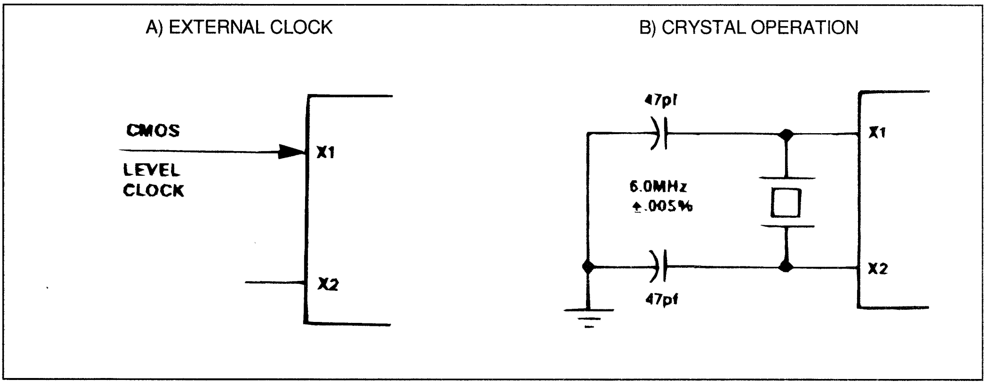
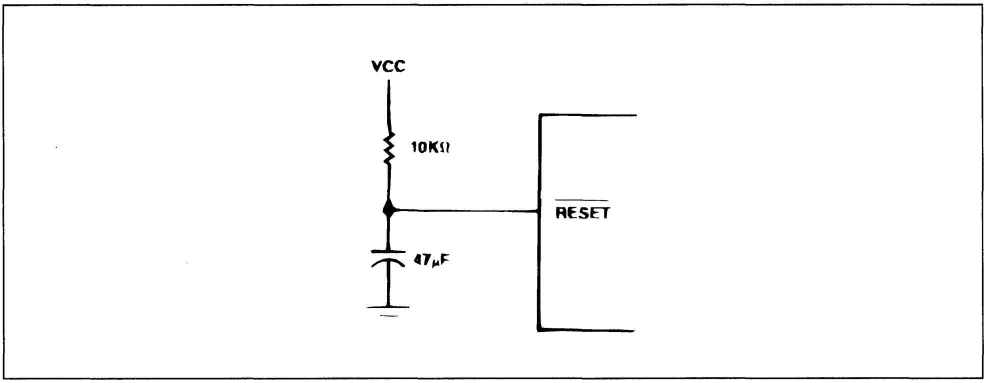


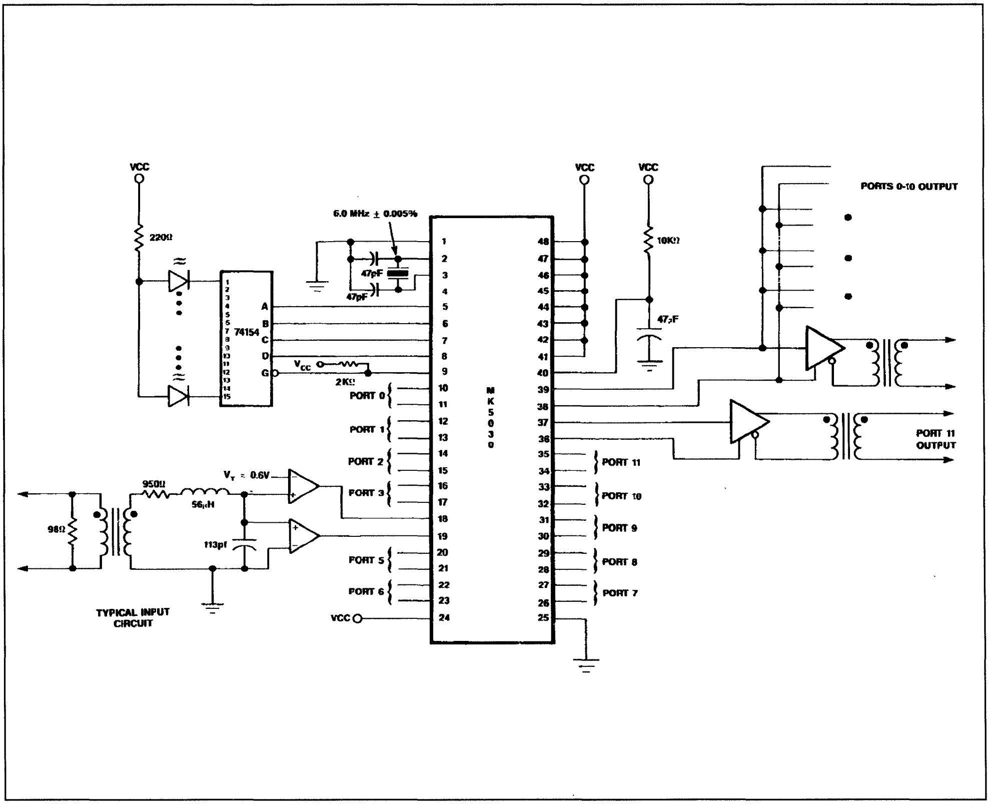
Figure 8 : Typical RC Connection for Power-On Reset.



SYSTEM PERFORMANCE CONSIDERATIONS USING THE MK5030

The MK5030 has several modes of operation. This allows designers flexibility in their design. Figure 9 shows a typical circuit diagram.

Figure 9 : MK5030 External Component Diagram.



ELECTRICAL SPECIFICATIONS

This chapter provides tabular presentations for Absolute Maximum Ratings, DC Characteristics, Capacitance, and AC Timing Specifications. In addition, illustrations are provided for an Output Load Diagram (figure 10) and HUB Timing Diagrams.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
	Temperature under Bias	– 25 to + 100	°C
	Storage temperature	– 65 to + 150	°C
	Voltage on any Pin with Respect to Ground	– 0.5 to V_{CC} + 0.5	V
	Power Dissipation (no load)	125	mW

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = + 5V ± 5% unless otherwise specified.

Symbol	Test Condition	Min.	Max.	Unit
V_{IL}		– 0.5	+ 0.8	V
V_{IH}	Except Pin X1	+ 2.0	V_{CC} + 0.5	V
V_{IH}	Pin X1	+ 3.5	V_{CC} + 0.5	V
V_{OL}	@ I_{OL} = 3.2mA		+ 0.5	V
V_{OH}	@ I_{OH} = – 0.4mA	+ 2.4		V
I_{IL}	@ V_{IN} = 0.4 to V_{CC}		± 10	µA
I_{CC}	@ T_{XI} = 6MHz		25	mA

CAPACITANCE F = 1MHz

Symbol	Test Condition	Min.	Max.	Unit
C_{IN}			10	pf
C_{OUT}			10	pf
C_{IO}			20	pf

AC TIMING SPECIFICATIONS

T_A = 0°C to 70°C, V_{CC} = + 5V ± 5% unless otherwise specified, V_{TH} = 2.0V, V_{TL} =0.8V

#	Signal	Symbol	Parameter	Min.	Typ.	Max.	Unit
1	X_1	T_{XIT}	X1 Period	160			ns
2	X_1	T_{XIL}	X1 Low Time	60			ns
3	X_1	T_{XIH}	X1 High Time	60			ns
4	X_1	T_{XIR}	Rise Time of X1	0		10	ns
5	X_1	T_{XIF}	Fall Time of X1	0		10	ns
6	X2	T_{X2}	X2 Delay from X1		30		ns
7	OCLK	T_{OCLK}	OCLK Delay from X1			45	ns
8	$\overline{UXD}$	$T_{\overline{UXD}}$	$\overline{UXD}$ Delay from X1		45	65	ns
9	$\overline{UXEN}$	$T_{\overline{UXEN}}$	$\overline{UXEN}$ Delay from X1		48	75	ns
10	$\overline{DXD}$	$T_{\overline{DXD}}$	$\overline{DXD}$ Delay from X1		48	65	ns
11	$\overline{DXEN}$	$T_{\overline{DXEN}}$	$\overline{DXEN}$ Delay from X1		51	75	ns
12	$\overline{UXD}$	J_{UXD}	Transmit Jitter : | $T_{\overline{UXD}}$ ↑ − $T_{\overline{UXD}}$ ↓ | ÷ 2		2	4	ns
13	$\overline{DXD}$	J_{DXD}	Transmit Jitter : | $T_{\overline{DXD}}$ ↑ − $T_{\overline{DXD}}$ ↓ | ÷ 2		2	4	ns
14	C0-11	T_{CS}	C0-11 Setup to X1	15			ns
15	C0-11	T_{CH}	C0-11 Hold from X1	15			ns
16	D0-11	T_{DS}	D0-11 Setup to X1	15			ns
17	D0-11	T_{DH}	D0–11 Hold from X1	15			ns
18	D0-11	J_{DIN}	D0-11 Incoming Jitter Tolerance			162	ns
19	$\overline{UXD}$, $\overline{DXD}$	T_{PXD}	Delay from any D Input		42	60	ns
20	$\overline{UXD}$, $\overline{DXD}$	J_{NR}	No Retime Jitter : | T_{XD} ↑ − T_{XD} ↓ | ÷ 2		2	4	ns
21	LED0-3	T_{LED}	Delay from X1		50		ns
22	$\overline{LEDEN}$	T_{LEDENR}	Guaranteed Release Time until LED_{0-3} Change		2600		ns
23	$\overline{LEDEN}$	T_{LEDENR}	Delay from LED_{0-3} Transition		2660		ns

Figure 10 : Oscillator Timing Diagram.

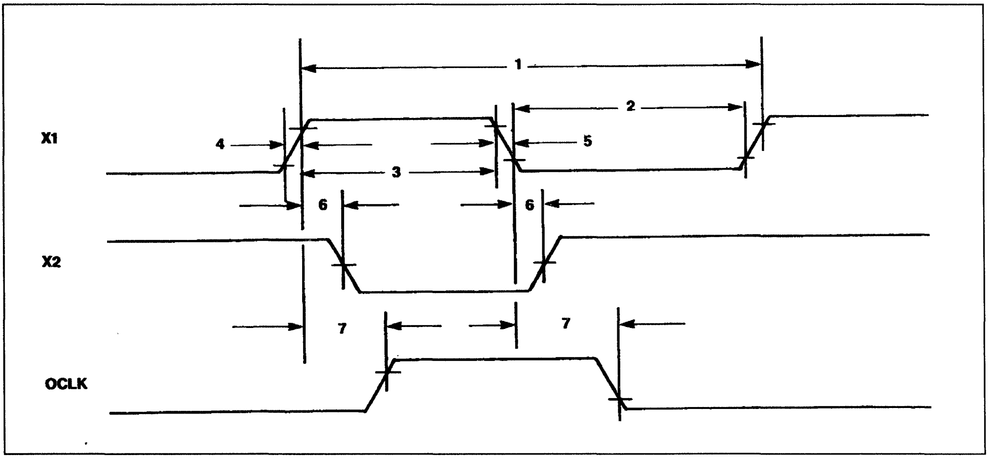


Figure 11 : Retimer Enabled (MODE3 = 1) Timing Diagram.

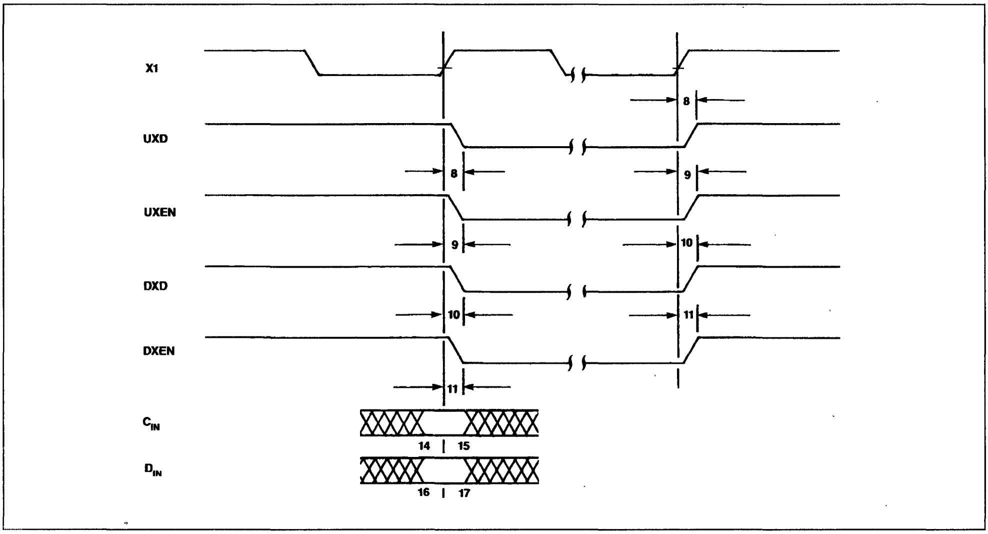


Figure 12 : Retimer Disabled (MODE3 = 0) Timing Diagram.

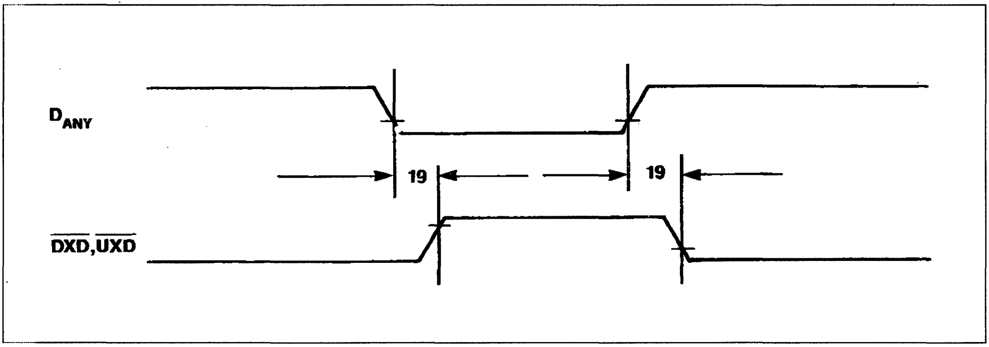


Figure 13 : Status Display Timing Diagram.

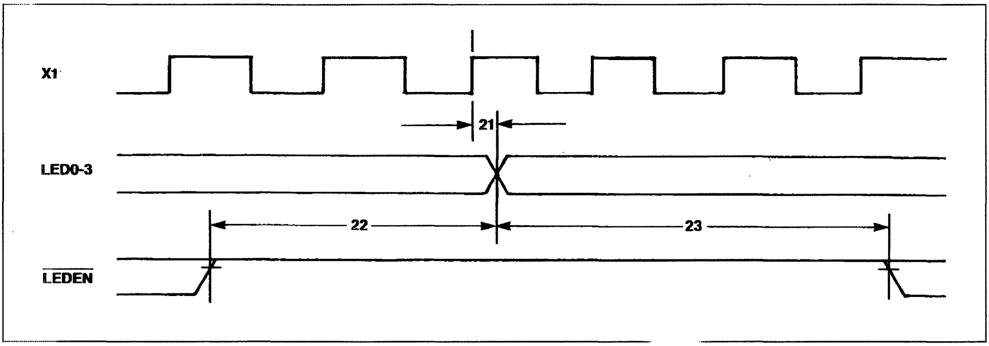
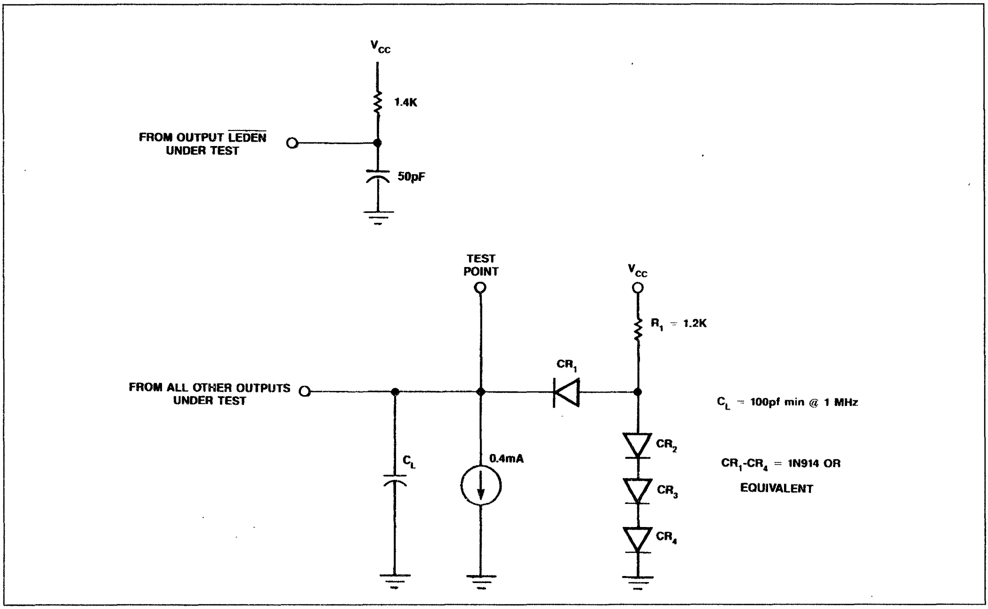


Figure 14 : Output Load Diagram.


Vcc
1.4K
FROM OUTPUT LEDEN UNDER TEST
50pF
TEST POINT
Vcc
R1 = 1.2K
CR1
FROM ALL OTHER OUTPUTS UNDER TEST
CL = 100pf min @ 1 MHz
CL
0.4mA
CR2
CR1-CR4 = 1N914 OR EQUIVALENT
CR3
CR4

GLOSSARY OF TERMS

automatic preamble generator (APG)
An optional circuit in the HUB chip which will begin preamble generation before the FIFO reaches the 4 bit watermark. The APG replaces a 2 bit loss in the preamble through a HUB with a 2 bit gain. See Retime Mode.

protection time
A 20µS period at the end of each frame where all carrier inputs are ignored. This protection insures immunity to post end-of-frame spikes caused by transformer coupling. See Protection Time.

downlink
The data path going from a "higher" HUB to the next "lower" HUB, or going from a HUB back to the stations.

high-end hub
A HUB that does not connect to another "higher" HUB. The downlink outputs, DXD and DXEN, are internally connected to the uplink outputs, UXD and UXEN.

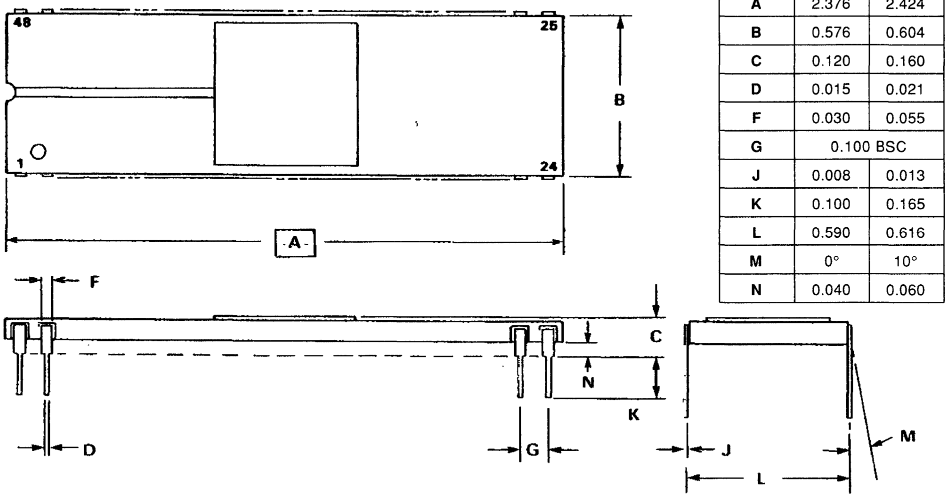
intermediate hub
A HUB that connects to another "higher" HUB. Pins C11 and D11 must be used as downlink inputs.

jabber
A circuit module inside the HUB chip which protects the network from a station which is constantly transmitting. See Jabber.

uplink
The data path going from a "lower" HUB to the next "higher" HUB, or going from the stations to a HUB.

PACKAGE DESCRIPTION

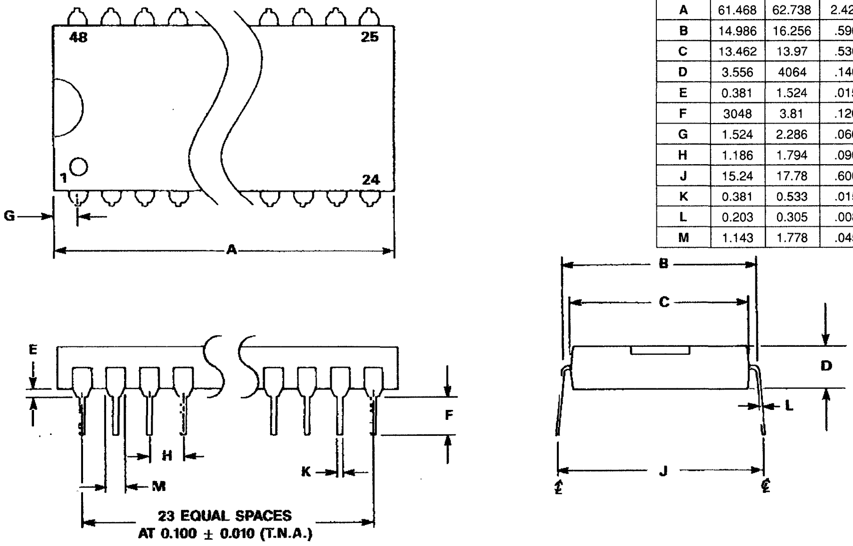
48 Pin Ceramic - MK5030P



DIM.	INCHES	
	MIN	MAX
A	2.376	2.424
B	0.576	0.604
C	0.120	0.160
D	0.015	0.021
F	0.030	0.055
G	0.100 BSC	
J	0.008	0.013
K	0.100	0.165
L	0.590	0.616
M	0°	10°
N	0.040	0.060

48 Pin Plastic - MK5030N

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	61.468	62.738	2.420	2.470
B	14.986	16.256	.590	.640
C	13.462	13.97	.530	.550
D	3.556	4064	.140	.160
E	0.381	1.524	.015	060
F	3048	3.81	.120	.150
G	1.524	2.286	.060	.090
H	1.186	1.794	.090	.110
J	15.24	17.78	.600	.700
K	0.381	0.533	.015	.021
L	0.203	0.305	.008	.012
M	1.143	1.778	.045	.070



MK5032

VARIABLE BIT-RATE (1 – 10MHz) IEEE 802.3 CONTROLLER

- SUPPORTS LAN STANDARDS : IEEE 802.3, ETHERNET, CHEAPERNET, AND STARLAN
- SUPPORTS DATA RATES FROM 1 TO 10Mbps
- SUPPORTS SYSTEM CLOCKS FROM 1 TO 10MHz
- ON-CHIP DMA WITH BUFFER MANAGEMENT USING CIRCULAR QUEUES
- COMPLETE CSMA/CD DATA LINK CONTROLLER (MAC)
- PREAMBLE INSERTION AND CHECKING
- CRC INSERTION AND STRIPPING
- GENERAL PURPOSE BUS INTERFACE COMPATIBLE WITH 8086 AND 68000 BUSES
- CABLE FAULT DETECTION
- 48 PIN DIP. + 5V ONLY. ALL INPUTS/OUTPUTS TTL COMPATIBLE
- COMPATIBLE WITH MK5033, MK50351 ENCODER/ DECODER AND WITH MK68591/2 SERIAL INTERFACE ADAPTOR

DESCRIPTION

The 5032 variable Bit-Rate LANCE is a 48-pin VLSI device that simplifies the interfacing of a microcomputer or a minicomputer to an IEEE 802.3 Local Area Network.

PIN DESCRIPTION

DAL00-DAL15

(Data/Address Bus)

Input/Output Three State. The time multiplexed Address/Data bus. These lines will be driven as a bus master and as a bus slave.

READ

Input/Output Three State. Indicates the type of operation to be performed in the current bus cycle. When it is a bus master, MK5032 drives this signal.

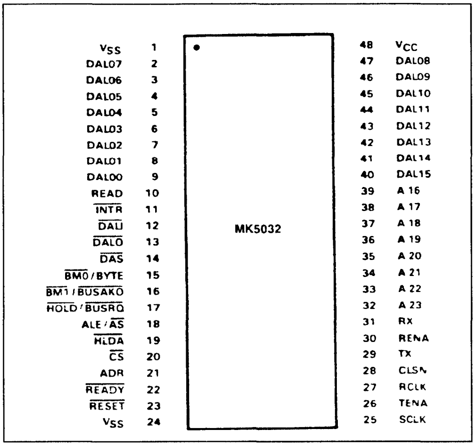
MK5032 as bus slave :

High - The chip places data on the DAL lines.
Low - The chip takes data off the DAL lines.

MK5032 as bus master :

High - The chip takes data off the DAL lines.
Low - The chip places data on the DAL lines.

Figure 1 : Pin Connections.


VSS 1
DAL07 2
DAL06 3
DAL05 4
DAL04 5
DAL03 6
DAL02 7
DAL01 8
DAL00 9
READ 10
$\overline{\text{INTR}}$ 11
$\overline{\text{DALI}}$ 12
$\overline{\text{DALO}}$ 13
$\overline{\text{DAS}}$ 14
$\overline{\text{BM0}}$/BYTE 15
$\overline{\text{BM1}}$/$\overline{\text{BUSAKO}}$ 16
$\overline{\text{HOLD}}$/$\overline{\text{BUSRQ}}$ 17
ALE/$\overline{\text{AS}}$ 18
$\overline{\text{HLDA}}$ 19
$\overline{\text{CS}}$ 20
ADR 21
$\overline{\text{READY}}$ 22
$\overline{\text{RESET}}$ 23
VSS 24
MK5032
48 VCC
47 DAL08
46 DAL09
45 DAL10
44 DAL11
43 DAL12
42 DAL13
41 DAL14
40 DAL15
39 A 16
38 A 17
37 A 18
36 A 19
35 A 20
34 A 21
33 A 22
32 A 23
31 RX
30 RENA
29 TX
28 CLSN
27 RCLK
26 TENA
25 SCLK

$\overline{\text{INTR}}$

(Interrupt)

Output Open Drain. When enabled, an attention signal that indicates the occurence of one or more of the following events : a message reception or transmission has completed or an error has occured during the transaction ; the initialization procedure has completed ; or a memory error has been encountered. Setting INEA in CSR0 (bit 06) enables $\overline{\text{INTR}}$.

DALI

(Data/Address Line In)

Output Three State. An external bus transceiver control line. When MK5032 is a bus master and reads from the DAL lines, $\overline{\text{DALI}}$ is asserted during the data portion of the transfer.

$\overline{\text{DALO}}$

(Data/Address Line Out)

Output Three State. An external bus transceiver control line. When MK5032 is a bus master and drives the DAL lines, DALO is asserted during the address portion of a read transfer of for the duration of a write transfer.

DAS

(Data Strobe)

Input/Output Three State. Defines the data portion of the bus transaction. DAS is driven is driven only as a bus master.

BM0, BM1 OR BYTE, BUSAKO

(Byte Mask)

Output Three State. Pins 15 and 16 are programmable through bit (00) of CSR3 (known as BCON). Asserting RESET clears CSR3.

CSR3(00) BCON = 0

PIN 16 = BM1 (Output Three State)

PIN 15 = BM0 (Output Three State)

BM0, BM1 Byte Mask. Indicates the byte(s) of a bus transaction to be read or written. The BM lines are ignored as a bus slave and assume word transfers only. The MK5032 drives the BM lines only when it is a bus master. Byte selection occurs as follows :

BM1	BM0	
Low	Low	Whole Word
Low	High	Byte of DAL 08 - DAL 15
High	Low	Byte of DAL 00 - DAL 07
High	High	None

CSR3(00) BCON = 1

PIN 16 = BUSAKO (Output)

PIN 15 = BYTE (Output Three State)

BYTE. An alternate byte selection line. Byte selection occurs when the BYTE and DAL (00) lines are latched during the address portion of the bus transaction. BYTE, BM0 and BM1 are ignored when MK5032 is a bus slave. There are two modes of ordering bytes depending on bit (02) of CSR3, (known as BSWP). This programmable ordering of upper and lower bytes when using BYTE and DAL (00) as selection signals is required to make the ordering compatible with various 16-bit microprocessors.

BSWP = 0		BSWP = 1		
BYTE	DAL(00)	BYTE	DAL(00)	
Low	Low	Low	Low	Whole Word
Low	High	Low	High	Illegal Condition
High	High	High	Low	Upper Byte
High	Low	High	High	Lower Byte

BUSAKO. The DAM daisy chain output.

HOLD/BUSRQ

(Bus Hold Request)

Input/Output Open Drain. MK5032 asserts this signal when it requires access to memory. HOLD is held low for the entire bus transaction. This bit is programmable through bit (00) of CSR3 (known as BCON) In the daisy chain DAM mode (BCON = 1) BUSRQ is asserted only if BUSRQ is inactive prior to assertion. Bit (00) of CSR3 is cleared when RESET is asserted.

CSR3(00) BCON = 0

PIN 17 = HOLD (Output Open Drain)

CSR3(00) BCON = 1

PIN 17 = BUSRQ (Output Open Drain)

BUSRQ will be asserted only if PIN 17 is high prior to assertion.

ALE/AS

(Address Latch Enable)

Output Three State. Used to demultiplex the DAL lines and define the address portion of the bus cycle. This pin is programmable through bit (01) of CSR3. As ALE, the signal transitions from high to low at the end of the address portion of the bus the address portion of the bus transaction and remains low during the entire data portion of the transaction. As AS, the signal transitions from low to high at the end of the address portion of the bus transaction and remains high throughout the entire data portion of the transaction. The MK5032 drives the ALE/AS line only as a bus master.

CSR3(01) ACON = 0

PIN 18 = ALE

CSR3(01) ACON = 1

PIN 18 = AS

HLDA

(Bus Hold Acknowledge)

Input. A response to HOLD indicating that the MK5032 is the Bus Master. HLDA stops its response when HOLD ends its assertion.

CS

(Chip Select)

Input. When asserted, CS indicates MK5032 is the slave device of the data transfer. CS must be valid throughout the data portion of the bus cycle.

ADR

(Register Address Port Select)

Input. When CS is asserted, ADR indicates which of the two register ports is selected. ADR must be valid throughout the data portion of the bus cycle.

DATA	PORT
Low	Register Data Port
High	Register Address Port

READY

Input/Output Open Drain. When the MK5032 is a bus master, READY is an asynchronous acknowledgement from external memory that will complete the data transfer. As a bus slave, the chip asserts READY when it has put data on the bus, or is about to take data off the bus. READY is a response to DAS. READY negates after DAS negates. Note : If DAS or CS deassert prior to the assertion of READY, READY cannot assert.

RESET

Input. Bus reset signal. Causes MK5032 to cease operation and to enter an idle state.

SCLK
(System Clock)

Input. A clock from 1 to 10MHz.

TENA
(Transmit Enable)

Output. Transmit Output Stream Enable. A level asserted with the transmit output bit stream, TX, to enable the external transmit logic.

RCLK
(Receive Clock)

Input. Normally a 6 square wave synchronized to the receive data and present only while receiving an input bit stream.

CLSN
(Collision)

Input. A logical input that indicates that a collision is occuring on the channel.

TX
(Transmit)

Output. Transmit output bit stream. This pin is programmable through bit (07) of the MODE REGISTER (MAN). When this bit is a "zero" the output data stream will be NRZ. When MAN is set to a "one", the data will be Manchester Encoded starting at a zero level and ending at the end of packet in a marking condition. (Continuous one level). This mode will function only when the data rate is programmed less than the rate of SCLK.

Three other bits in the MODE register provide a data rate division of 1, 2, 4, 6, 8, or 10. This means that the data rate of TX will be a division of SCLK. For more details on the MODE register, see the technical manual.

RENA
(Receive Enable)

Input. A logical input that indicates the presence of data on the channel.

RX
(Receive)

Input. Receive input bit stream.

A16 - A23
(High-Order Address Bus)

Output Three State. The additional address bits necessary to extend the DAL lines to produce a 24-bit address. These lines will be driven only as a bus master.

VCC

Power supply pin. + 5VDC ± 5 percent.

Filtering : A power supply filter is recommended at the MK5032 between V_{CC} (48) and V_{SS} (1, 24). This filter consists of two capacitors in parallel having the values of 10 µf and 0.47 µf respectively.

VSS

Ground. 0 VDC.

FUNCTIONAL CAPABILITIES

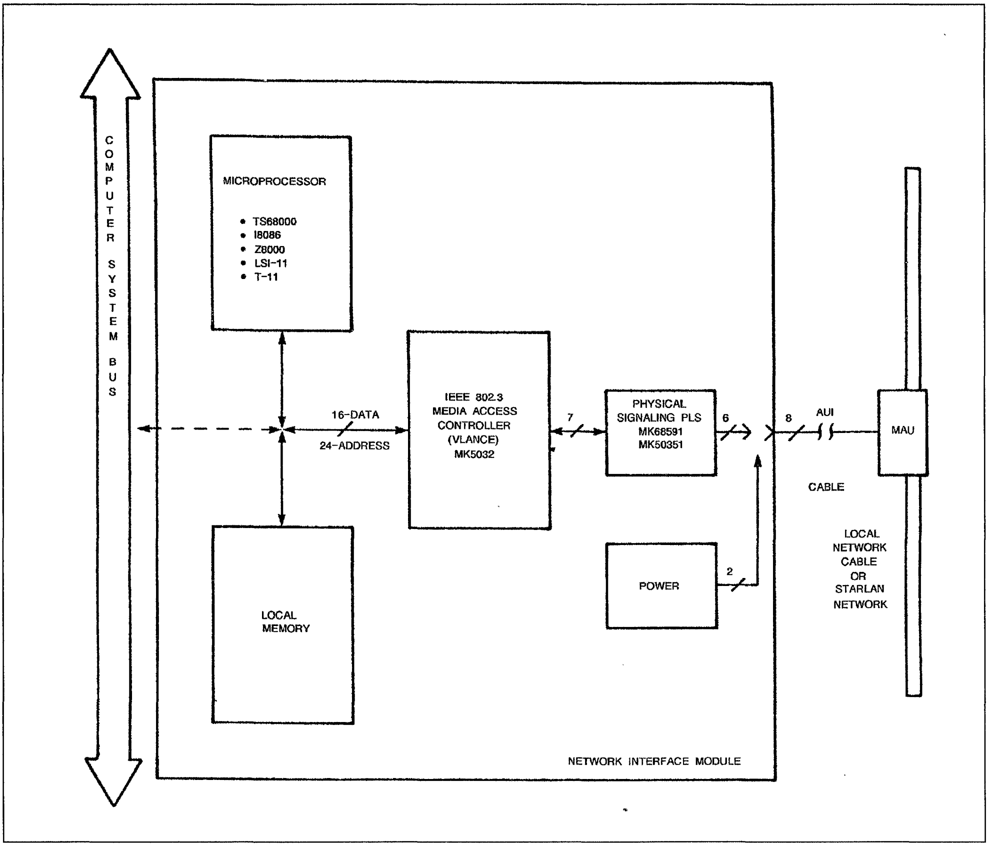
The MK5032 interfaces to a microprocessor bus characterized by time-multiplexed address and data lines. Typically, data transfers are 16 bits wide but byte transfers occur if the buffer memory address boundaries are odd. The address bus is 24 bits wide.

The IEEE 802.3 packet format consists of 64-bit preamble, a 48-bit destination address, a 48-bit source address, a 16-bit field, and from a 46 to 1500 byte data field terminated with a 32-bit CRC. The packets' variable widths accommodate both short-status command and terminal traffic packets and long data packets to printers and disks (1024-byte disk sectors, for example). Packets are spaced a minimum of 96 bit times apart to allow one node enough time to receive back-to-back packets.

The MK5032 operates in a minimal configuration that requires close coupling between local memory and a processor. The local memory provides packet buffering for the chip and serves as a communication link between chip and processor. During initialization, the control processor loads the starting address of the initialization block plus the operating mode of the chip via two ports that can access four control registers into MK5032. The host processor talks directly to MK5032 only during this initial phase. All further communications are handled via a Direct Memory Access (DMA) machine under microword control contained within MK5032. Figure 2 shows a block diagram of the MK5032 and PLS (MK68591, MK50351, or MK5033) device used to create an IEEE 802.3 interface for a computer system.

mode of the chip via two ports that can access four control registers into MK5032. The host processor talks directly to MK5032 only during this initial phase. All further communications are handled via a Direct Memory Access (DMA) machine under microword control contained within MK5032. Figure 2 shows a block diagram of the MK5032 and PLS (MK68591, MK50351, or MK5033) device used to create an IEEE 802.3 interface for a computer system.

Figure 2 : Ethernet Local Area Network System Block Diagram.



FUNCTIONAL DESCRIPTION

SERIAL DATA HANDLING

MK5032 provides the IEEE 802.3 interface as follows. In the transmit mode (since there is only one transmission path, IEEE 802.3 is a half duplex system), the MK5032 reads data from a transmit buffer by using DMA and appends the preamble, sync pattern (two ones after alternating ones and zeros in the preamble), and calculates and appends the complement of the 32-bit CRC. In the receive mode, the destination address, source address, type, data, and CRC fields are transferred to memory via DMA cycles. The CRC is calculated as data and transmitted CRC is received. At the end of the packet, if this calculated CRC does not agree with a constant, an error bit is set in RDM1 of the receiver descriptor ring. In the receive mode, MK5032 accepts packets under four modes of operation. The first mode is a full comparison of the 48-bit destination address in the packet with the node address that was programmed into the MK5032 during an initialization cycle. There are two types of logical addresses. One is a group type mas ; where the 48-bit address in the packet is put through a hash filter in order to map the 48-bit physical addresses into 1 of 64 logical ation is the so called "promiscuous mode" in which a node will accept all packets on the cable regardless of their destination address.

COLLISION DETECTION AND IMPLEMENTATION

The IEEE 802.3 CSMA/CD network access algorithm is implemented completely within MK5032. In addition to listening for a clear network cable before transmitting, IEEE 802.3 handles collisions in a predetermined way. Should two transmitters attempt to seize the network cable at the same time, they will collide, and the data on the network cable will be garbled. MK5032 is constantly monitoring the Collision (CLSN) pin. This signal is generated by the MAU when the signal level on the network cable indicates the presence of signals from two or more transmitters. If MK5032 is transmitted when CLSN is asserted, it will continue to transmit the preamble (collisions normally occur while the preamble is being transmitted), then will "jam" the network for 32 bit times. This jamming ensures that all nodes have enough time to detect the collision. The transmitting nodes then delay a random amount of time according to the "truncated binary backoff" algorithm defined in the IEEE 802.3 specification to minimize the probability of the colliding nodes having multiple collisions with each other. After 16 abortive attemps to transmit packet, MK5032 will report a RTRY error due to excessive collisions and step over the transmitter buffer. During reception, the detection of a collision causes that reception to be aborted. Depending on when the collision occurred, MK5032 will treat this packet as an error packet if the packet has an address mismatch, as a runt packet (a packet that has less than 64 bytes), or as a legal length packet with a CRC error.

Fatal error reporting is provided by the MK5032 through a microprocessor interrupt and error flags in CSR0. These error conditions are collision error (the failure of the MAU to send a signal-quality-error message at the conclusion of a normal transmission), transmitter ON longer than 1518 bytes, a missed packet, and a memory error (failure of a memory transaction to complete within 256 sys clocks).

Additional errors are reported through bits in the descriptor rings (on a buffer by buffer basis). Receive error conditions include framing, CRC and buffers errors, and overflow. Transmit descriptor rings have error bits indicating buffer, underflow, late collision, and loss of carrier. Additionally, transmit descriptor rings have a bit indicating that the transmitter has unsuccessfully tried to transmit over a busy communication link.

Transmit descriptor rings also have ten bits reserved for a Time Domain Reflectometry counter (TDR). On the occurance of a collision, the value in the TDR will give the number of system clocks until the collision, which can be used to determine the distance to the fault.

BUFFER MANAGEMENT

A key feature of the MK5032 and its DMA channel is the flexibility and speed of communication between the MK5032 and the host microprocessor through common memory locations. The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings, as shown if figure 3. These rings control both transmit and receive operations. Up to 128 tasks may be queued on a descriptor ring for execution by the MK5032. Each entry in a descriptor ring holds a pointer to a data memory buffer and an entry for the data buffer length. Data buffers can be chained or cascaded to handle a long packet in multiple data buffer areas. The MK5032 searches the descriptor rings to determine the next empty buffer. This enables it to chain buffers together or to handle back-to-back packets. As each buffer is filled, an "own" bit is reset, signaling the host processor to empty this buffer.

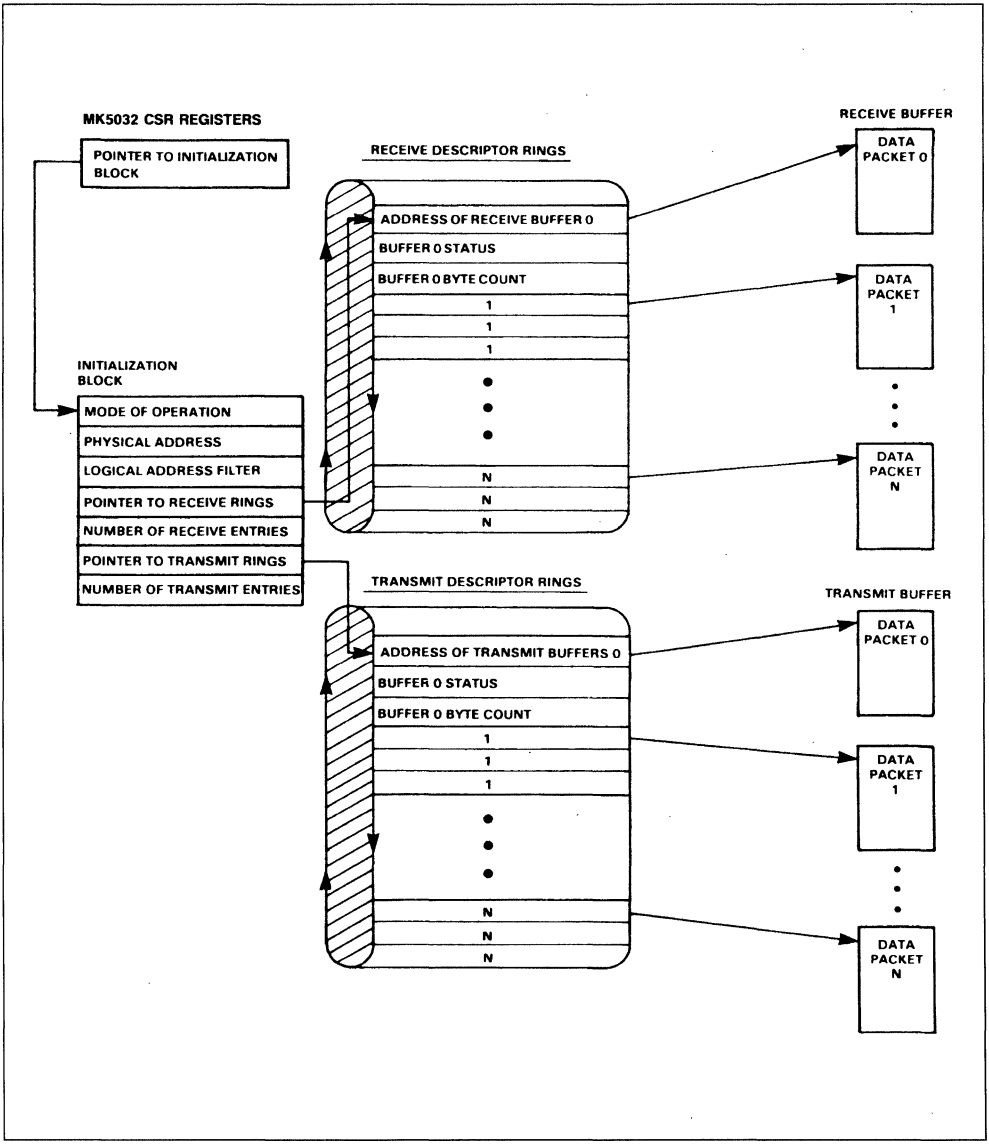
MICROPROCESSOR INTERFACE

The parallel interface of MK5032 has been designed to be "friendly", or easy to interface, to many popular 16-bit microprocessors. These microprocessors include the TS68000, Z8000, 8086, LSI-11, T-11. MK5032 has a wide 24-bit linear address space when in the Bus Master Mode, allowing it to DMA the entire address space of the above microprocessors. MK5032 uses no segmentation or paging methods. As such, MK5032 addressing is closest to MK68000 addressing, but is compatible with the other microprocessors. When MK5032 is a bus master, a programmable mode of operation allows byte addressing, either by employing a Byte/Word control signal (much like that used on the 8086 or the Z8000) or by using an Upper Data Strobe/Lower Data Strobe much like that used on the TS68000, LSI-11 microprocessors. A programmable polarity on the Address Strobe signal eliminates the need for external logic. MK5032 interfaces with multiplexed and demultiplexed data busses and features control signals for address/data bus transceivers.

After the initialization routine, packet reception or transmission, transmitter timeout error, a missed packet, or memory error, the MK5032 generates an interrupt to the host microprocessor.

The cause of the interrupt is ascertained by reading CSR0.Bit (06) of CSR0, INEA, enables or disables interrupts to the micorprocessor. In the polling mode, BIT (07) of CSR0 is sampled to determine if an interrupt causing condition has occurred.

Figure 3 : MK5032 Memory Management.



MK5032 INTERFACE DESCRIPTION

ALE, DAS and READY time all data transfers from the MK5032 in the Bus Master mode. The automatic adjustment of the MK5032 cycle by the READY signal allows synchronization with variable cycle time memory due either to memory refresh or to dual port access. Bus cycles are a minimum of 600ns long and can be increased in 100ns increments.

READ SEQUENCE

At the beginning of a read cycle, valid addresses are placed on DAL00-DAL15 and A16-A21. The BYTE Mask signals (BM0 and BM1) become valid at the beginning of this cycle as does READ, indicating the type of cycle. The trailing edge of ALE or AS strobes the addresses A0-A15 into the external latches. Approximately 100ns later, DAL00-DAL15 go into a three state mode. There is a 50ns delay to allow for tranceiver turnaround, then DAS falls low to signal the beginning of the data portion of the cycle. At this point in the cycle, the MK5032 stalls waiting for the memory device to assert READY. Upon assertion of READY, DAS makes a transition from a zero to a one, latching memory data. (DAS is low for a minimum of 200ns).

The bus transceiver controls, DALI and DALO, control the bus transceivers. DALI signals to strobe data toward the MK5032 and DALO signals to strobe data or addresses away from the MK5032. During a read cycle, DALO goes inactive before DALI goes active to avoid "spiking" of bus transceivers.

WRITE SEQUENCE

The write cycle begins exactly like a read cycle with the READ line remaining inactive. After ALE or AS pulse, the DAL00-DAL15 change from addresses to data. DAS goes active when the DAL00-DAL15 are stable. This data remains valid on the bus until the memory device asserts READY. At this point, DAS goes inactive, latching data into the memory device. Data is held for 75ns after the negation of DAS.

MK5032 INTERFACE DESCRIPTION BUS SLAVE MODE

The MK5032 enters the Bus Slave Mode whenever CS becomes active. This mode must be entered whenever writing or reading the four status control registers (CSR0, CSR1, CSR2, and CSR3) and the register address pointer (RAP). RAP and CSR0 may be read or written to at any time, but the MK5032 must be stopped (CSR0 bit 02) when CSR1, CSR2, or CSR3 is to be written to or read.

MK5032 ELECTRICAL SPECIFICATION

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
T_A	Temperature Under Bias	– 25 to + 125	°C
T_{stg}	Storage Temperature	– 65 to + 150	°C
V_I	Voltage on any Pin with Respect to Ground	– 0.3 to + 7	V
P_D	Power Dissipation	2.0	V

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = + 5V ± 5 percent unless otherwise specified.

Symbol	Parameter	Min.	Max.	Unit
V_{IL}	Input Low Voltage	– 0.5	+ 0.8	V
VI_H	Input High Voltage	+ 2.0	V_{CC} +0.5	V
V_{OL}	Output Low Voltage		+ 0.5	V
V_{OH}	Output High Voltage	+ 2.4		V
I_{IL}	Input Leakage		± 10	µA

CAPACITANCE F = 1MHz

Symbol	Parameter	Min.	Max.	Unit
C_{IN}	Input Capacitance		10	pF
C_{OUT}	Output		10	pF
C_{IO}			20	pF

AC TIMING SPECIFICATIONS (T_A = 0°C to 70°C, V_{CC} = + 5V ± 5 percent, unless otherwise specified.

N°	Signal	Symbol	Parameter	Test Conditions	Min. (ns)	Typ. (ns)	Max. (ns)
1	SCLK	T_{SCT}	SCLK Period		99		101
2	SCLK	T_{SCL}	SCLK Low Time		45		55
3	SCLK	T_{SCH}	SCLK High Time		45		55
4	SCLK	T_{SCR}	Rise Time of SCLK		0		8
5	SCLK	T_{SCF}	Fall Time of SCLK		0		8
6	TENA	T_{TEP}	TENA Propagation delay after the rising edge of SCLK.	C_L = 50pF			75
7	TENA	T_{TEH}	TENA Hold Time after the rising edge of SCLK.	C_L = 50pF	5		
8	TX	T_{TDP}	TX data propagation delay after the rising edge of SCLK. (see note 1)	C_L = 50pF			75
8A	TX	T_{TDTT}	TX Transition - Transition (see note 2)	C_L = 50pF	B_t - 7		B_t - 7
9	TX	T_{TDH}	TX data Hold Time after the rising edge of SCLK. (see note 1)	C_L = 50pF	5		
10	RCLK	T_{RCT}	RCLK Period		85		118
11	RCLK	T_{RCH}	RCLK High Time		38		
12	RCLK	T_{RCL}	RCLK Low Time		38		
13	RCLK	T_{RCR}	Rise Time of RCLK		0		8
14	RCLK	T_{RCF}	Fall time of RCLK		0		8
15	RX	T_{RDR}	RX Data Rise Time		0		8
16	RX	T_{RDF}	RX Data Fall Time		0		8
17	RX	T_{RDH}	RX Data Hold Time (RCLK to RX data change)		5		
18	RX	T_{RDS} (see note 3)	RX Data Setup Time (RX data stable to the rising edge of RCLK)		See note 3		
19	RENA	T_{DPL}	RENA Low Time		120		
20	RENA	T_{RENH}	RENA Hold Time after rising edge of RCLK.		40		
21	CLSN	T_{CPH}	CLSN High Time		80		
22	A/DAL	T_{DOFF}	Bus master driver disable after rising edge of $\overline{HOLD}$		0		50
23	A/DAL	T_{DON}	Bus master driver enable after falling edge of $\overline{HLDA}$		0		150
24	$\overline{HLDA}$	T_{HHA}	Delay to falling edge of $\overline{HLDA}$ from falling edge of $\overline{HOLD}$ (bus master)		0		

AC TIMING SPECIFICATIONS (continued)

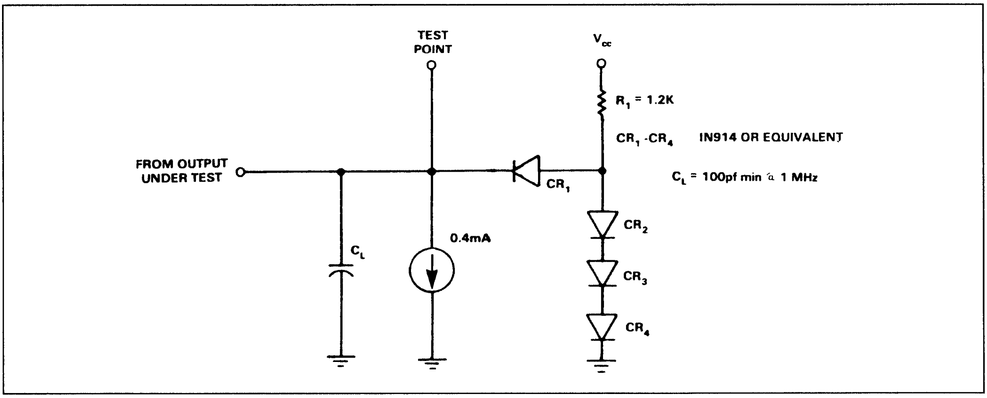
N°	Signal	Symbol	Parameter	Test Conditions	Min. (ns)	Typ. (ns)	Max. (ns)
25	$\overline{\text{RESET}}$	T_{RW}	$\overline{\text{RESET}}$ Pulse Width Low		200		
26	A/DAL	T_{CYCLE}	Read/Write, Address/Data Cycle time		600		
27	A	T_{XAS}	Address setup time to the falling edge of ALE.		75		
28	A	T_{XAH}	Address hold time after the rising edge of $\overline{\text{DAS}}$.		15		
29	DAL	T_{AS}	Address setupt time to the falling edge of ALE.		75		
30	DAL	T_{AH}	Address hold time after the falling edge of ALE.		35		
31	DAL	T_{RDAS}	Data setup time to the rising edge of $\overline{\text{DAS}}$ (bus master read).		50		
32	DAL	T_{RDAH}	Data hold time after the rising edge of $\overline{\text{DAS}}$ (bus master read).		0		
33	DAL	T_{DDAS}	Data setup time to the falling edge of $\overline{\text{DAS}}$ (bus master write).		0		
34	DAL	T_{WDS}	Data setup time to the rising edge of $\overline{\text{DAS}}$ (bus master write).		200		
35	DAL	T_{WDH}	Data hold time after the rising edge of $\overline{\text{DAS}}$ (bus master write).		35		
36	DAL	T_{SD01}	Data driver delay after the falling edge of $\overline{\text{DAS}}$ (bus slave read).	(CSR 0, 3, RAP)		400	
37	DAL	T_{SD02}	Data driver delay after the falling edge of $\overline{\text{DAS}}$ (bus slave read).	(CSR 1, 2)		1200	
38	DAL	T_{SRDH}	Data hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave read).		0		35
39	DAL	T_{SWDH}	Data hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave write).		0		
40	DAL	T_{SWDS}	Data setup time to the falling edge of $\overline{\text{DAS}}$ (bus slave write).		0		
41	ALE	T_{ALEW}	ALE Width High		120		150
42	ALE	T_{DALE}	Delay from rising edge of $\overline{\text{DAS}}$ to the rising edge of ALE.		70		
43	$\overline{\text{DAS}}$	T_{DSW}	$\overline{\text{DAS}}$ Width Low		200		
44	$\overline{\text{DAS}}$	T_{ADAS}	Delay from the falling edge of ALE to the falling edge of $\overline{\text{DAS}}$.		80		130
45	$\overline{\text{DAS}}$	T_{RIDF}	Delay from the rising edge of $\overline{\text{DALO}}$ to the falling edge of $\overline{\text{DAS}}$ (bus master read).		15		
46	$\overline{\text{DAS}}$	T_{RDYS}	Delay from the falling edge of $\overline{\text{READY}}$ to the rising edge $\overline{\text{DAS}}$.	Taryd = 300ns	75		250
47	$\overline{\text{DALI}}$	T_{ROIF}	Delay from the rising edge of $\overline{\text{DALO}}$ to the falling edge of $\overline{\text{DALI}}$ (bus master read).		15		

AC TIMING SPECIFICATIONS (continued)

N°	Signal	Symbol	Parameter	Test Conditions	Min. (ns)	Typ. (ns)	Max. (ns)
48	$\overline{\text{DALI}}$	T_{RIS}	$\overline{\text{DALI}}$ setup time to the rising edge of $\overline{\text{DAS}}$ (bus master read).		135		
49	$\overline{\text{DALI}}$	T_{RIH}	$\overline{\text{DALI}}$ hold time to the rising edge of $\overline{\text{DAS}}$ (bus master read).		0		
50	$\overline{\text{DALI}}$	T_{RIOF}	Delay from the rising edge of $\overline{\text{DALI}}$ to the falling edge of $\overline{\text{DALO}}$ (bus master read).		55		
51	$\overline{\text{DALO}}$	T_{OS}	$\overline{\text{DALO}}$ setup time to the falling edge of ALE (bus master read).		110		
52	$\overline{\text{DALO}}$	T_{ROH}	$\overline{\text{DALO}}$ hold time after the falling edge of ALE (bus master read).		35		
53	$\overline{\text{DALO}}$	T_{WDSI}	Delay from the rising edge of $\overline{\text{DAS}}$ to the rising edge of $\overline{\text{DALO}}$ (bus master write).		35		
54	$\overline{\text{CS}}$	T_{CSH}	$\overline{\text{CS}}$ hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave).		0		
55	$\overline{\text{CS}}$	T_{CSS}	$\overline{\text{CS}}$ setup time to the falling edge of $\overline{\text{DAS}}$ (bus slave).		0		
56	ADR	T_{SAH}	ADR hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave).		0		
57	ADR	T_{SAS}	ADR setup time to the falling edge of $\overline{\text{DAS}}$ (bus slave).		0		
58	$\overline{\text{READY}}$	T_{ARYD}	Delay from the falling edge of ALE to the falling edge of $\overline{\text{READY}}$ to insure a minimum bus cycle time (600ns).				80
59	$\overline{\text{READY}}$	T_{SRDS}	Data setup time to the falling edge of $\overline{\text{READY}}$ (bus slave read).		75		
60	$\overline{\text{READY}}$	T_{RDYH}	$\overline{\text{READY}}$ hold time after the rising edge of $\overline{\text{DAS}}$ (bus master).		0		
61	$\overline{\text{READY}}$	T_{SR01}	$\overline{\text{READY}}$ driver turn on after the falling edge of $\overline{\text{DAS}}$ (bus slave).	(CSR 0, 3, RAP)		600	
62	$\overline{\text{READY}}$	T_{SR02}	$\overline{\text{READY}}$ driver turn on after the falling edge of $\overline{\text{DAS}}$ (bus slave).	(CSR 1, 2)		1400	
63	$\overline{\text{READY}}$	T_{SRYH}	$\overline{\text{READY}}$ hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave).		0		35
64	READ	T_{SRH}	READ hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave).		0		
65	READ	T_{SRS}	READ setup time to the falling edge of $\overline{\text{DAS}}$ (bus slave).		0		

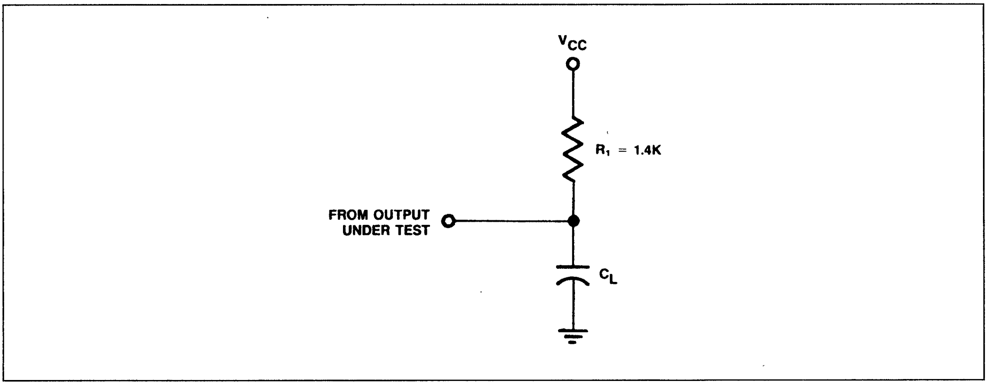
Notes : 1. This timing is for the NRZ mode only.
2. B_t = Bit time. This measurement is during preamble ; valid for Manchester Mode only.
3. $T_{RDS\ (min)} = T_{RCT} - 25$ns. Therefore, T_{RCT} = 100ns when $T_{RDS\ (min)}$ = 75ns.

Figure 4 : Output Load Diagram.



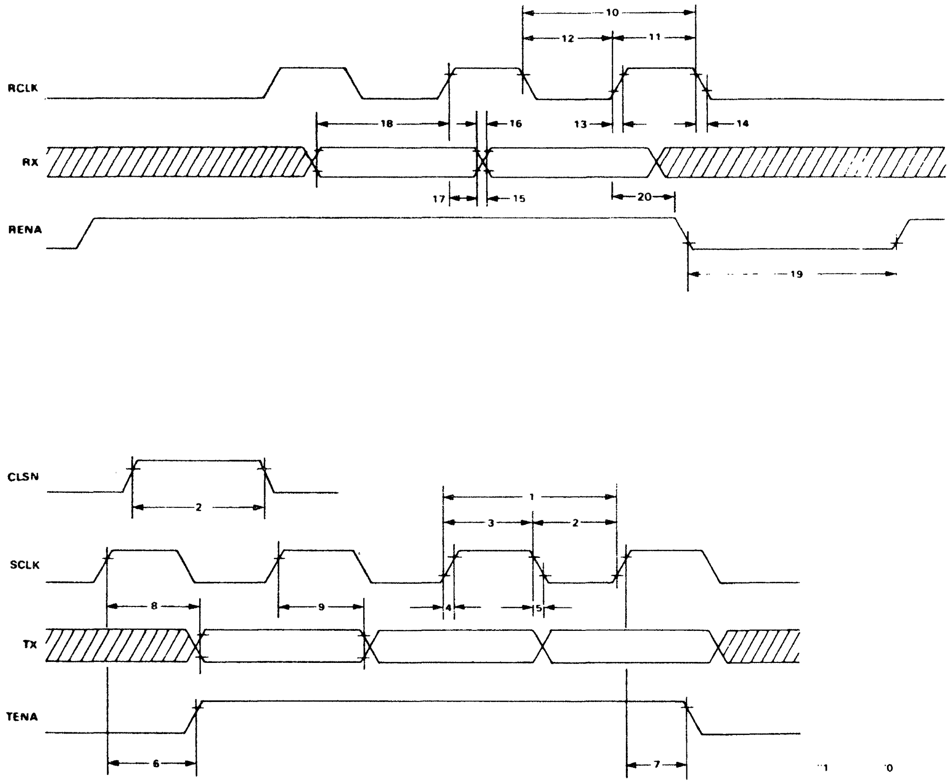
Note : This load is used on DAL00 through DAL15, READ, DALI, DALO, DAS, BM0, BM1 ALE/AS, A16 through A23, TENA, and TX.

Figure 5 : Open Drain Output Load Diagram.



Note : This load is used on open drain outputs $\overline{\text{INTR}}$, $\overline{\text{HOLD}}/\overline{\text{BUSRQ}}$, and $\overline{\text{READY}}$.

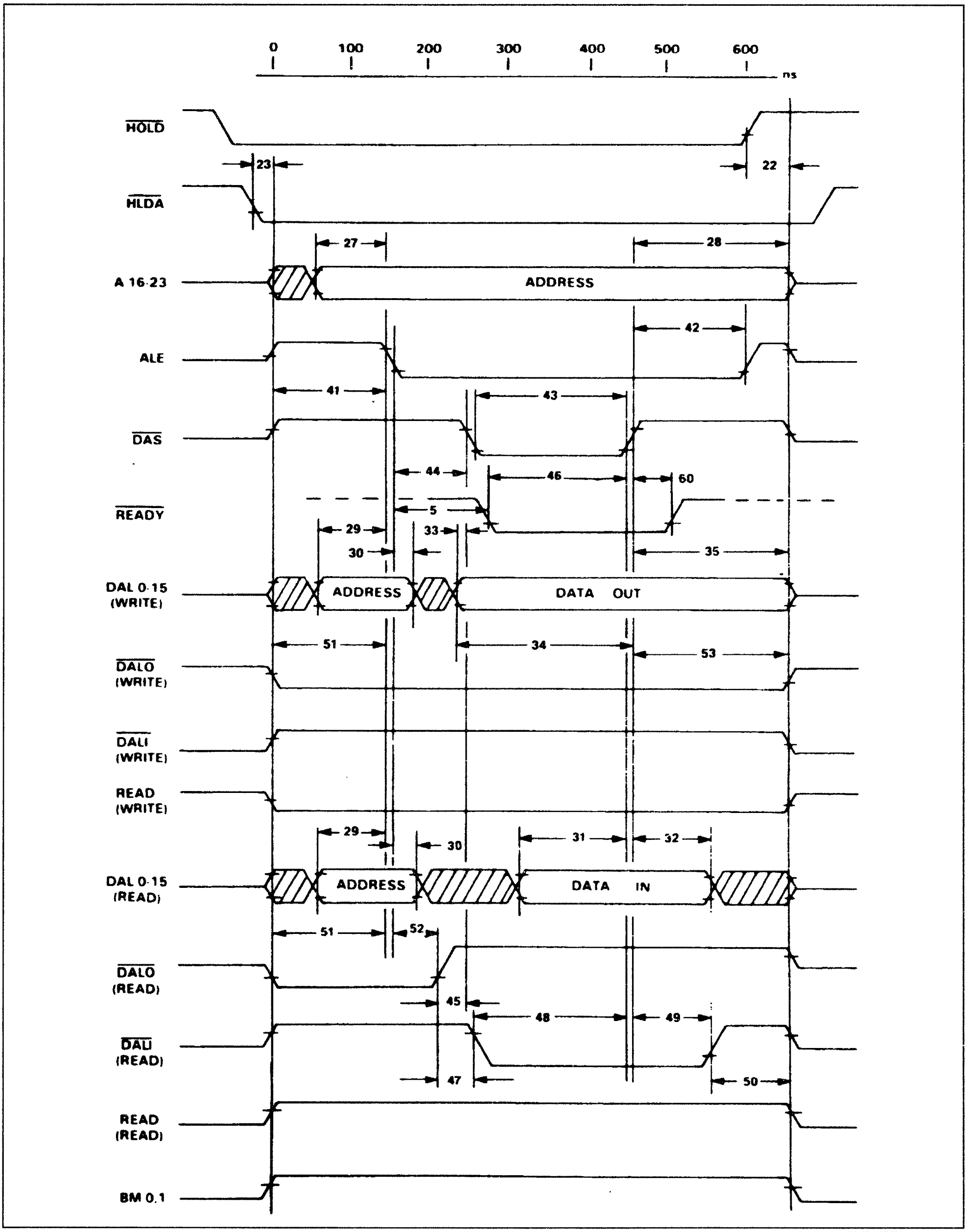
Figure 6 : Physical Link Signaling Timing Diagram - PLS-VMAC Interface Signals.



	¨1	˙0
OUTPUT	2 0 V	0 8 V
INPUT	2 0 V	0 8V
FLOAT	V	0 5 V

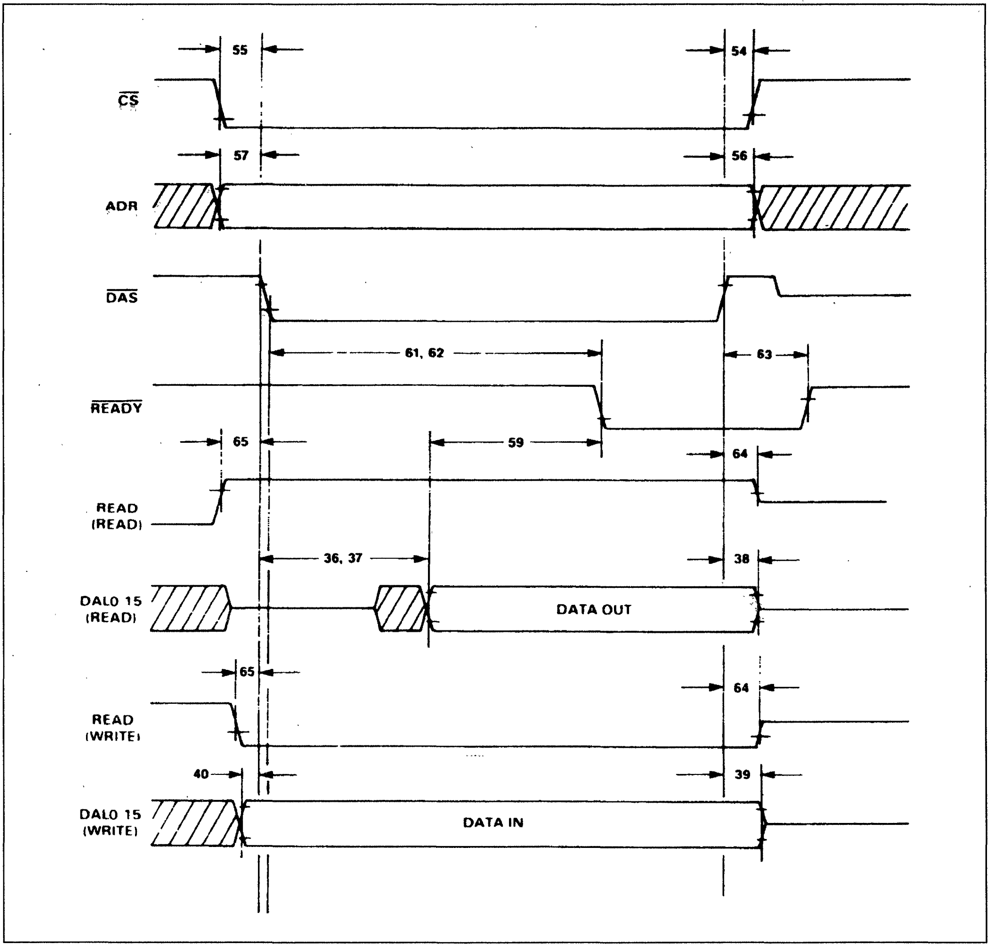
Note : Timing measurements are made at the following voltages unless otherwise specified.

Figure 7 : MK5032 Bus Master Timing Diagram.



Note : The Bus Master cycle time will increase from a minimum of 600ns in 100ns steps until the slave device returns $\overline{\text{READY}}$.

Figure 8 : MK5032 Bus Slave Timing Diagram.



SGS-THOMSON
MICROELECTRONICS

TECHNICAL MANUAL

MK5032 - VARIABLE BIT RATE LOCAL AREA NETWORK CONTROLLER FOR ETHERNET

TABLE OF CONTENTS

SECTION PAGE

Chapter 1
General Description

Chapter 2
Programming Specifications

TABLE OF CONTENTS (continued)

Chapter 3
Functional Specifications

TABLE OF CONTENTS (continued)

Chapter 4
Electrical Specifications

LIST OF ILLUSTRATIONS

PAGE

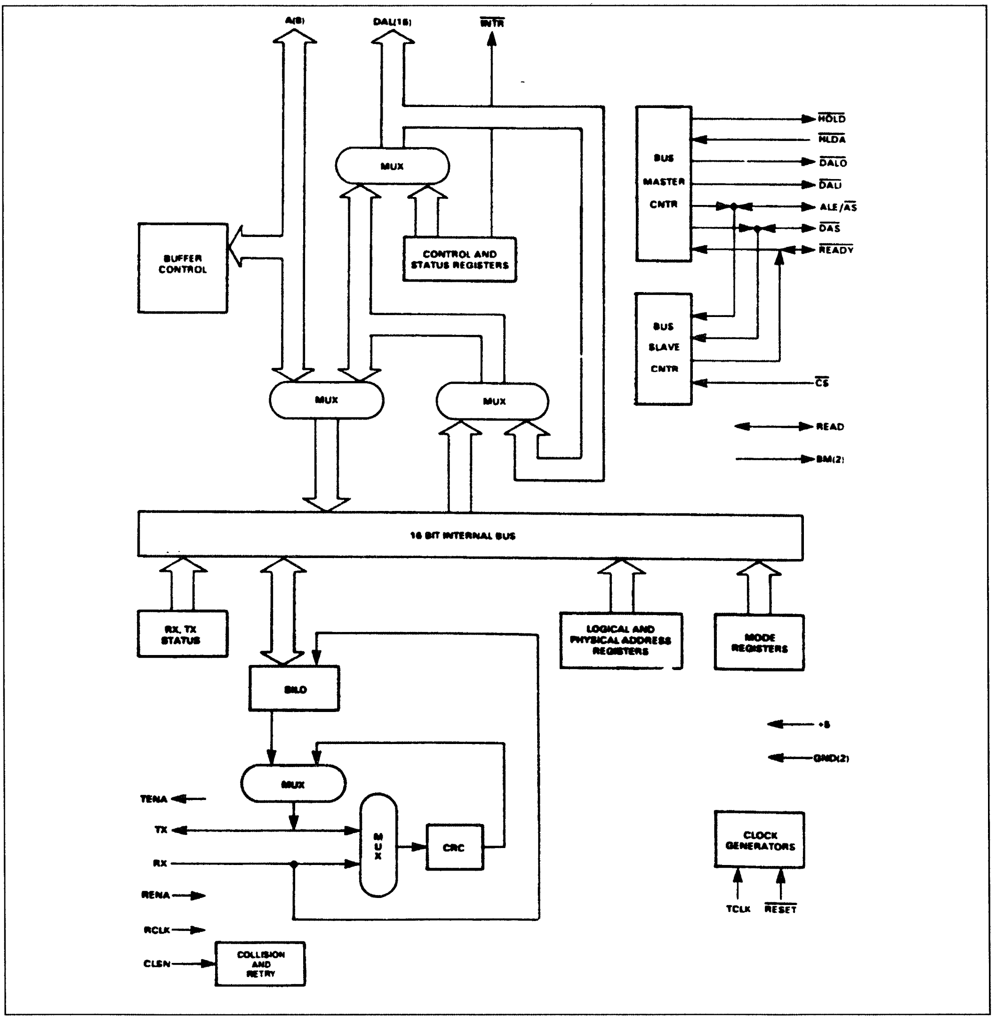
CHAPTER 1
GENERAL DESCRIPTION

1. INTRODUCTION

1.1. OVERVIEW

The MK5032 VLANCE variable Bit Rate (Local Area Network Controller for Ethernet) is a 48 pin VLSI device designed to simplify greatly the interfacing of a microcomputer or a minicomputer to an Ethernet Local Area Network. This chip is intended to operate in a local environment that includes a closely coupled memory and microprocessor. The VLANCE uses scaled N-channel MOS technology and is compatible with several microprocessors. A block diagram of the chip is shown in figure 1.

Figure 1 : VLANCE Block Diagram.



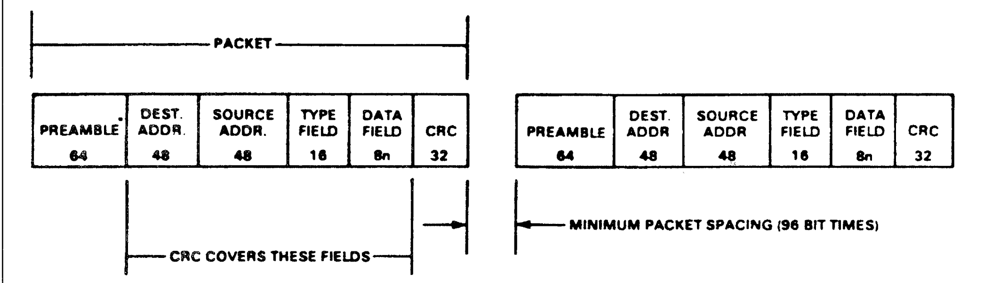
VLANCE is a trademark of SGS-THOMSON Microelectronics.

1.2. FUNCTIONAL CAPABILITIES

The Variable bit rate Local Area Network Controller For Ethernet (VLANCE) interfaces to a microprocessor bus characterized by time multiplexed address and data lines. Typically, data transfers are 16 bits wide, but byte transfers occur if the buffer memory address boundaries are odd. The address bus is 24 bits wide.

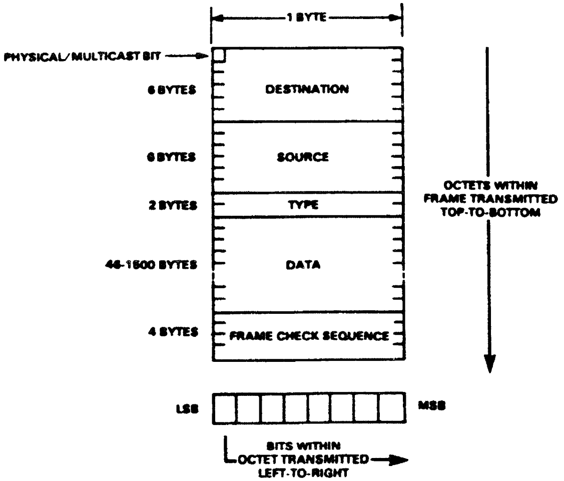
The Ethernet packet format consists of a 64-bit preamble, a 48-bit destination address, a 48-bit source address, a 16-bit type field, and a 46 to 1500 byte data field terminated with a 32-bit CRC as shown in figure 2 and figure 3. The variable widths of the packets accommodate both short status, command and terminal traffic packets, and long data packets to printers and disks (1024 bytes disk sectors for example). Packets are spaced a minimum of 9.6 µsec apart to allow one node time enough to receive backtoback packets.

Figure 2 : Ethernet and VLANCE Packet Format.



* Last Byte is Start of Frame Synchronization Byte..10101011.

Figure 3 : Ethernet and VLANCE Packet Bit Transmission Sequence.



The VLANCE is intended to operate in a minimal configuration that requires close coupling between local memory and a processor. Figure 5 shows the relationship between the chip and local memory. The local memory provides packet buffering for the chip and serves as a communication link between the chip and the processor. During initialization, the control processor loads into VLANCE the starting address of the initialization block plus the operating mode of the chip via two control registers. It is only during this initial phase that the host processor talks directly to VLANCE. All further communications are handled via a DMA machine under microword control contained within the VLANCE. Figure 4 is a block diagram of the VLANCE and SIA device used to create an Ethernet interface for a computer system.

Figure 4 : Ethernet Local Area Network System Block Diagram.

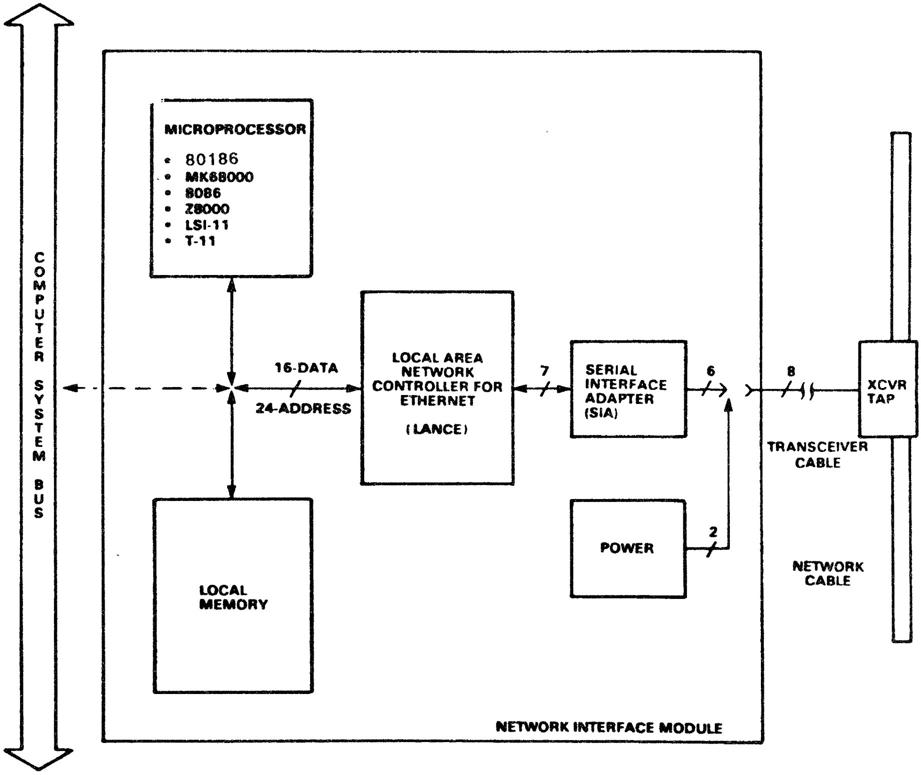
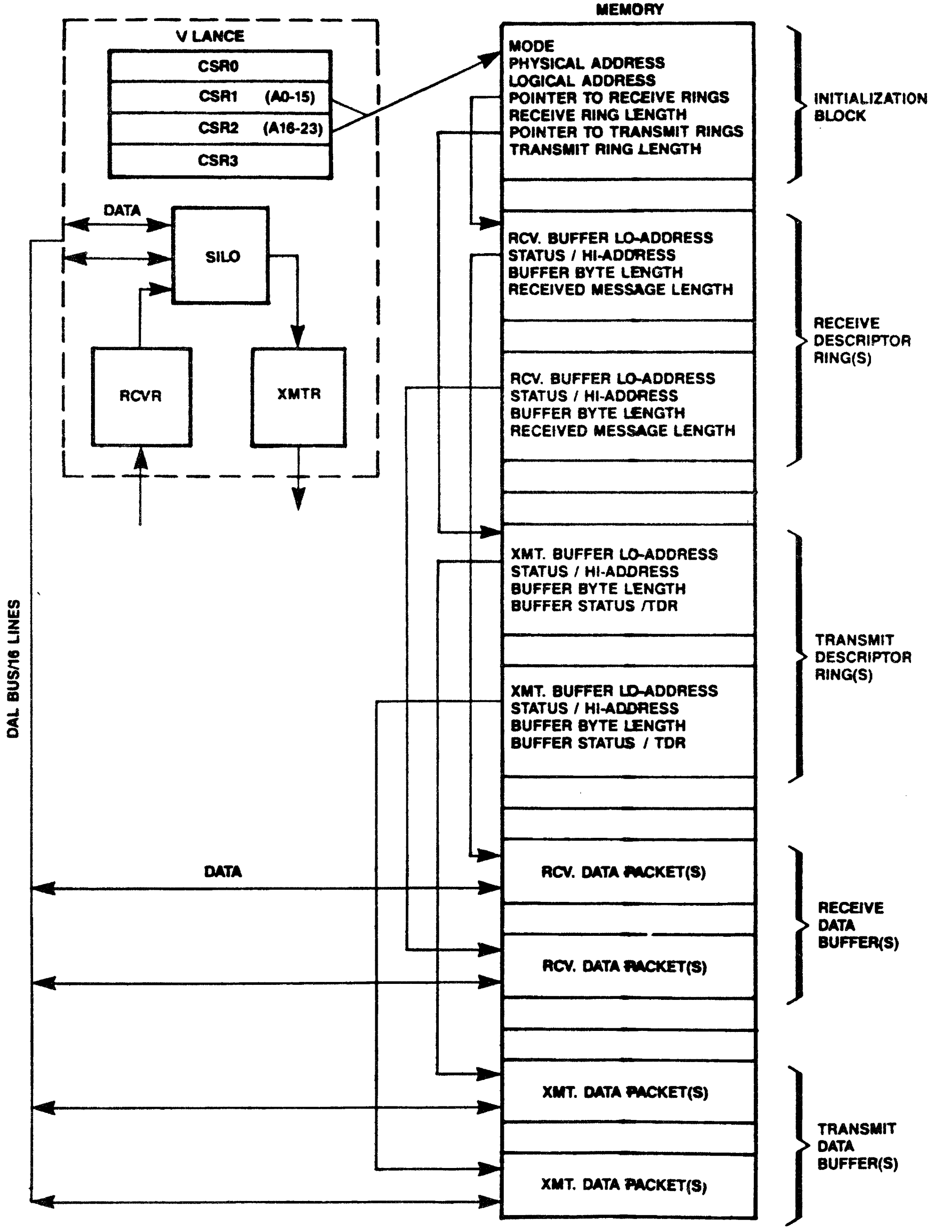


Figure 5 : VLANCE Conceptual View.



1.3. FUNCTIONAL DESCRIPTION

1.3.1. SERIAL DATA HANDLING. The basic operation of the chip set to provide the Ethernet interface is as follows. In the transmit mode (since there is only one transmission path, Ethernet is a half duplex system), the VLANCE reads data from a transmit buffer by using Direct Memory Access (DMA) and appends the preamble, sync pattern (two ones after alternating ones and zeros in the preamble), and calculatės and appens the complement of the 32-bit CRC. The first eight words of the transmit buffer must contain the destination address, source address, and type field as detailed in the Ethernet specification. In the receive mode, the destination address, source address, type, data, and CRC fields are transferred to memory via DMA cycles. The CRC is calculated as the data and transmitted CRC are received. At the end of the packet, if this calculated CRC does not agree with a constant, an error bit is set and in RMD1 of the receiver descriptor rings. In the receive mode, packets will be accepted by the VLANCE under four modes of operation. The first mode is a full comparison of the 48-bit destination address in the packet with the node address that was programmed into the VLANCE during an initialization cycle. There are two types of logical address. One is a group type mask where the 48-bit address in the packet is put through a hash filter in order to map the 48-bit physical addresses into 1 of 64 logical groups. This mode can be useful if sending packets to all of one type of a device simultaneously or the network (i.e. sending a packet to all file servers or all printer servers). The second logical address is the broadcast address where all nodes on the network receive the packet. The last receive mode of operation is the so called "promiscuous mode" in which a node will accept all packets on the network cable regardless of their destination address.

1.3.2. COLLISION DETECTION AND IMPLEMENTATION. The Ethernet CSMA/CD network access algorithm is implemented completely within the VLANCE. In addition to listening for a clear network cable before transmitting, Ethernet handles collisions in a predetermined way. Should two transmitters attempt to seize the network cable at the same time, they will collide, and the data on the network cable will be garbled. VLANCE is constantly monitoring the CLSN (collision) pin. This signal is generated by the transceiver when the signal level on the network cable indicates the presence of signals from two or more transmitters. If VLANCE is transmitting when CLSN is asserted, it will continue to transmit the preamble, (normally collisions will occur while the preamble is being transmitted) then will "jam" the network for 32 bit times (3.2 microseconds). This jamming ensures that all nodes have enough time to detect the collision. The transmitting nodes then delay a random amount of time according to the "truncated binary backoff" algorithm defined in the Ethernet specification to minimize the probability of the colliding nodes having multiple collisions with each other. After 16 abortive attemps to transmit a packet, VLANCE will report a RTRY error due to excessive collisions and step over the transmit buffer. During reception, the detection of a collision causes that reception to be aborted. Depending on when the collision occured, VLANCE will treat this packet as an error packet if the packet has an address mismatch, as a runt packet (a packet that has less than 64 bytes), or as a legal length packet with a CRC error. Extensive error reporting is provided by the VLANCE through a microprocessor interrupt and error bits in a status register. The following are the significant error conditions : CRC error on received data ; transmitter on longer than 1518 bytes ; missed packet error (meaning a packet on the network cable was missed because there were no empty buffers in memory), and memory error, in which the memory did not respond (handshake) to a memory cycle request.

1.3.3. BUFFER MANAGEMENT. A key feature of the VLANCE and its on board DMA channel is the flexibility and speed of communication between the VLANCE and the host microprocessor through common memory locations. The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings, as shown in figure 6. There are separate descriptor rings to describe transmit and receive operations. Up to 128 tasks may be queued up on a descriptor ring awaiting execution by the VLANCE. Each entry in a descriptor ring holds a pointer to a data memory buffer and an entry for the length of the data buffer. Data buffers can be chained or cascaded in order to handle a long packet in multiple data buffer areas. The VLANCE searches the descriptor rings in a "look ahead manner" to determine the next empty buffer in order to chain buffers together or to handle back to back packets. As each buffer is filled, an "own" bit is reset, signaling the host processor to empty this buffer. The minimum buffer size is 64 bytes for receive buffers and 100 bytes for transmit buffers.

1.3.4. MICROPROCESSOR INTERFACE. The parallel interface of the VLANCE has been designed to be "friendly" or easy to interface to a variety of popular 16-bit microprocessors. These microprocessors include the following : 80186, 68000, Z8000, 8086, LSI-11, T-11. The VLANCE has a wide 24-bit linear address space when it is in the Bus Master Mode, allowing it to DMA directly into the entire address space of the above microprocessors. No segmentation or paging methods are used within the VLANCE, and as such the addressing is closest to that used by the 68000 but is compatible with the others. When the VLANCE is a bus master, a programmable mode of operation allows byte addressing either by employing a Byte/Word control signal, much like that used on the 8086 or the Z8000, or by using an Upper Data Strobe/Lower Data strobe much like that used on the 68000 and LSI-11 microprocessors. A programmable polarity on the Address Strobe signal eliminates the need for external logic. The VLANCE interfaces with both multiplexed and demultiplexed data buses and features control signals for address/data bus transceivers.

Figure 6 : VLANCE Memory Management.

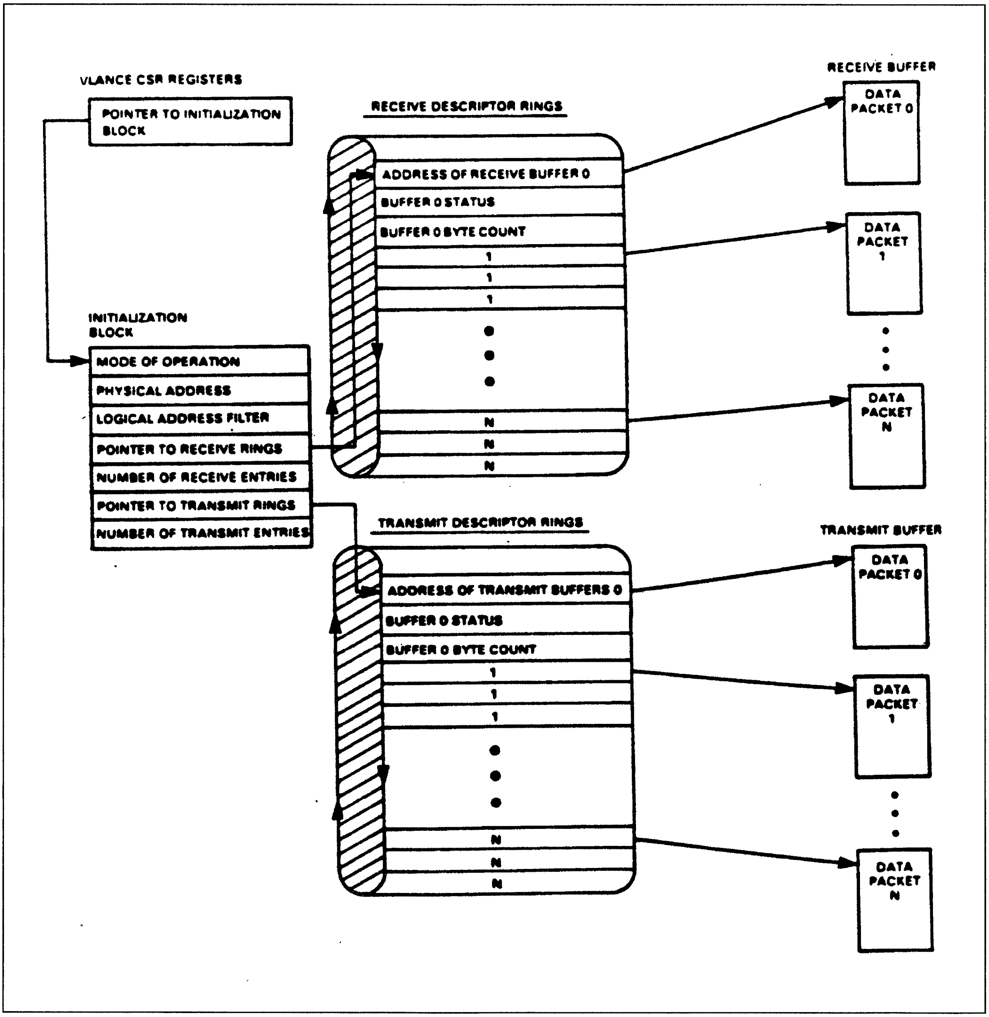
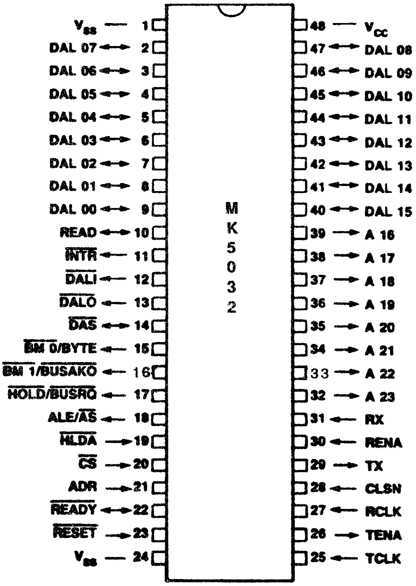


Figure 7 : VLANCE Pin Assignment.



Interrupts to the microprocessor are generated by the VLANCE upon completion of its initialization routine, the reception of a packet, the transmission of a packet, transmitter timeout error, a missed packet, or a memory error.

The cause of the interrupt is ascertained by reading the control status register (CSR0). Bit (06) of CSR0, INEA, enables or disables interrupts to the microprocessor. In a polling mode, BIT (07) of CSR0 is sampled to determine when an interrupt causing condition occured.

1.3.5. PIN DESCRIPTION.

DAL00-DAL15

(Data/Address Bus).

Input/Output Tri-State. Pins 2-9 and 40-47. The time multiplexed Addres/Data bus. During the address portion of a memory transfer, DAL <15 : 00> contains the lower 16 bits of the memory address. The upper 8 bits of address are contained in A <23 : 16>. During the data portion of a memory transfer, DAL <15 : 00> contains the read or write data, depending on the type of transfer. The VLANCE drives these lines both as a Bus Master and as a Bus Slave.

READ

Input/Output Tri-State. Pin 10. Read indicates the type of operation the bus controller is performing during a bus transaction. When it is a Bus Master, VLANCE drives READ. Read is valid during the entire bus transaction and is tri-stated at all other times.

VLANCE as Bus Slave :

High - The chip places data on the DAL lines.

Low - The chip takes data off the DAL lines.

VLANCE as Bus Master :

High - The chip takes data off the DAL lines.

Low - The chip places data on the DAL lines.

$\overline{\text{INTR}}$

(Interrupt).

Output Open Drain. Pin 11. INTERRUPT is an attention interrupt line that indicates that one or more of the following CSR0 status flags is set : BABL, MISS, MERR, RINT, TINT, OR IDON. Interrupt is enabled by CSR0 <6>, INEA = 1.

DALI

(Data/Address Line In).

Output Tri-State. Pin 12. DAL IN is an external bus transceiver control line. VLANCE drives DALI only while it is the Bus Master. When VLANCE reads the DAL lines during the data portion of a READ transfer, DALI is asserted. DALI is not asserted during a WRITE transfer.

DALO

(Data/Address Line Out).

Output Tri-State. Pin 13. DAL OUT is an external bus transceiver control line. VLANCE drives DALO only when it is a Bus Master. When VLANCE drives the DAL lines during the address portion of a READ transfer or for the duration of a WRITE transfer, DALO is asserted.

DAS

(Data/Strobe).

Input/Output Tri-State. Pin 14. Data Strobe defines the data portion of the bus transaction. By definition, data is stable and valid at the low to high transition of DAS. When it is the Bus Master, VLANCE drives this signal. At all other times, the signal is tri-stated.

BM0, BM1 **or BYTE, BUSAKO**

(Byte Mask).

Output Tri-State. Pins 15 and 16 are programmable through CSR3.

CSR3 <00> BCON = 0
PIN 15 = BM0 (Output Tri-State)
PIN 16 = BM1 (Output Tri-State)

Byte Mask <1 : 0> Indicates the byte(s) on the DAL to be read or written during this bus transaction. VLANCE drives these lines only as a Bus Master. VLANCE ignores the BM lines when it is a Bus Slave and assumes word transfers. Byte selection follows :

BM1	BM0	
LOW	LOW	Whole Word
LOW	HIGH	Byte <DAL 15 : 08>
HIGH	LOW	Byte <DAL 07 : 00>
HIGH	HIGH	None

CSR3 <00> BCON = 1
PIN 15 = BYTE (Output Tri-State)
PIN 16 = BUSAKO (Output)

Byte selection occurs by using the BYTE line and DAL <00> latched during the address portion of the bus transaction. VLANCE drives BYTE only as a Bus Master and ignores it when operating as a Bus Slave. Byte selection occurs as follows :

BYTE	DAL <00>	(During Address Portion)
LOW	LOW	WHOLE WORD
LOW	HIGH	ILLEGAL CONDITION
HIGH	LOW	LOWER BYTE
HIGH	HIGH	UPPER BYTE

BUSAKO is a bus request daisy chain output. If VLANCE is not requesting the bus and it receives HLDA, BUSAKO is driven low. If VLANCE is requesting the bus when it receives HLDA, BUSAKO remains high.

HOLD/BUSRQ

(Bus Hold Request).

Input/Output Open Drain. Pin 17. This pin is programmable through CSR3.

CSR3 <00> BCON = 0
PIN 17 = HOLD

VLANCE asserts the HOLD request when it requires a DMA cycle regardless of the HOLD pin state. HOLD is held LOW for the entire bus transaction.

CSR3 <00> BCON = 1
PIN 17 = BUSRQ

VLANCE asserts BUSRQ when it requires a DMA cycle if the prior state of the BUSRQ pin was high. BUSRQ is held low for the entire bus transaction.

ALE/AS

(Address Latch Enable).

Output Tri-State. Pin 18. The active level of Address Strobe is programmable through CSR3. The address portion of a bus transfer occurs while this signal is at its asserted level. VLANCE drives this signal while it is the Bus Master. At all other times, the signal is tri-stated.

CSR3 <01> ACON = 0
PIN 18 = ALE

Address Latch Enable is used to demultiplex the DAL lines and define the address portion of the transfer. As ALE, the signal transitions from high to low during the address portion of the transfer and remains low during the data portion. A slave device can use ALE to control a latch on the bus address lines. When ALE is high, the latch should be open and when ALE goes low, the latch should be closed.

CSR3 <01> ACON = 1
PIN 31 = AS

As AS, the signal pulses low during the address portion of the bus transfer. The low to high transition of AS can be used by a slave device to strobe the address into a register.

HLDA

(Hold Acknowledge).

Input. Pin 19. Hold Acknowledge is the response to HOLD. When HLDA is low in response to VLANCE's assertion of HOLD, the VLANCE is the Bus Master. HLDA should be deasserted after VLANCE releases HOLD.

CS

(Chip Select).

Input. Pin 20. When low, CS indicates VLANCE is the slave device for the data transfer. CS must be valid throughout the data portion of the transaction.

ADR

(Register Address Port Select).

Input. Pin 21. Address selects the Register Address Port of the Register Data Port. It must be valid throughout the data portion of the transfer and the chip only uses it when CS is low.

ADR	PORT
LOW	Register Data Port
HIGH	Register Address Port

READY

Input/Output Open Drain. Pin 22. When VLANCE is a Bus Master, READY is an asynchronous acknowledgement from the bus memory that memory will accept data in a WRITE cycle or that memory has put data on the DAL lines in a READ cycle. As a Bus Slave, VLANCE asserts READY when it has put data on the DAL lines during a READ cycle or is about to take data off the DAL lines during a WRITE cycle. READY is a response to DAS and is negated after DAS is negated. CS and DAS must remain asserted until READY is asserted or READY will not be asserted.

RESET

(Bus Reset Signal.).

Input. Pin 23. Causes VLANCE to cease operation, clear its internal logic and enter an idle state with the STOP bit of CSR0 set.

TCLK

(Transmit Clock).

Input. Pin 25. A crystal-controlled 10MHz clock. This clock is the primary VLANCE clock as well as the Transmit clock. (A0.01% clock as specified in the Ethernet Specification.

TENA

(Transmit Enable).

Output. Pin 26. A high level signal asserted with the transmit output serial bit stream, TX, to enable the external transmit logic.

RCLK

(Receive Clock).

Input. Pin 27. The 10MHz clock that is synchronous with the received data and is used for transferring the received data into the VLANCE.

CLSN

(Collision).

Input. Pin 28. A logical input that indicates, when high, that a collision is occuring on the channel.

TX

(Transmit).

Output. Pin 29. Transmit Output Bit Stream.

RENA

(Receive Enable).

Input. Pin 30. A logical input that indicate, when high, the presence of data on the channel.

RX

(Receive).

Input. Pin 31. The input for the serial receive data. The data is synchronous with the receive clock.

A16-A23

(High-Order Address Bus).

Output, Three State pins 32 thru 39. Address bits <23 : 16> used in conjunction with DAL <15 : 00> to produce a 24-bit address. VLANCE drives these lines only as a Bus Master.

Vcc

Power supply pin 48. + 5VDC ± 5%.

Vss

Ground pins 1 and 24. 0 VDC

1.3.6. VLANCE INTERFACE DESCRIPTION BUS MASTER MODE. All data transfers from the VLANCE in the Bus Master mode are timed by ALE, DAS, and READY. The automatic adjustment of the VLANCE cycle by the READY signal allows synchronization with variable cycle time memory due either to memory refresh or to dual port access. Bus cycles are a minimum of 600nsec in length and can be increased in 100nsec increments. Figure 8 and figure 9 show generalized interfaces to both multiplexed and demultiplexed bus microprocessors, and figure 10, the Bus Master Timing modes.

1.3.6.1. Read Sequence. At the beginning of a read cycle, valid addresses are placed on DAL <15 : 00> and A <23 : 16>. The BYTE Mask signals (BM0 and BM1) become valid at the beginning of this cycle as does READ, indicating the type of cycle. The trailing edge of ALE or AS is used to strobe in the addresses A <15 : 00> into the external latches. Approximately a hundred nanoseconds later, DAL <15 : 00> go into a tri-state mode. There is a fifty nanosecond delay to allow for transceiver turnaround, then DAS falls low to signal the beginning of the data portion of the cycle. At this point in the cycle, the VLANCE waits for the memory device to assert READY. Upon assertion of READY, DAS makes a transition from a zero to a one, latching memory data. (DAS) is low for a minimum of 200nsec).

The bus transceiver controls, DALI and DALO, are used to control the bus transceivers. The DALI signal is used to strobe data toward the VLANCE and the DALO signal is used to strobe data or addresses away from the VLANCE. During a read cycle, DALO goes inactive before DALI goes active to avoid the "spiking" of the bus transceivers.

1.3.6.2. Write Sequence. The write cycle begins exactly like a read cycle with the READ line remaining inactive. After ALE or AS pulse, the DAL <15 : 00> change from addresses to data. DAS goes active when the DAL <15 : 00> lines are stable. This data will remain valid on the bus until the memory device asserts READY. At this point, DAS goes inactive latching data into the memory device. Data is held for 75 nanoseconds after the deassertion of DAS.

1.3.7. VLANCE INTERFACE DESCRIPTION BUS SLAVE MODE. The VLANCE enters the Bus Slave Mode whenever CS becomes active. This mode must be entered whenever writing or reading the four status control registers (CSR0, CSR1, CSR2 and CSR3) and the register address pointer (RAP). RAP and CSR0 may be read or written to at any time, but the VLANCE must be stopped for CSR1, CSR2, and CSR3 to be written to or read.

1.3.7.1. Read Sequence. CS, READ, and DAS are asserted at the beginning of a read cycle. ADR also must be valid at this time. (If ADR is a "1", the contents of RAP are placed on the DAL lines. Otherwise the contents of the CSR register addressed by RAP are placed on the DAL lines.) After the data on the DAL lines become valid, VLANCE asserts READY. CS, READ, DAS, and ADR must remain stable throughout the read cycle. Refer to figure 11.

1.3.7.2. Write Sequence. This cycle is similar to the read cycle, except that during this cycle, READ is not asserted. The DAL buffers are tristated which configures these lines as inputs. The assertion of READY by VLANCE indicates to the memory device that the data on the DAL lines has been stored by VLANCE in its appropriate CSR register. CS, READ, DAS, ADR, and DAL <15 : 00> must remain stable throught the write cycle. Refer to figure 12.

Figure 8 : Multiplexed Bus Interface.

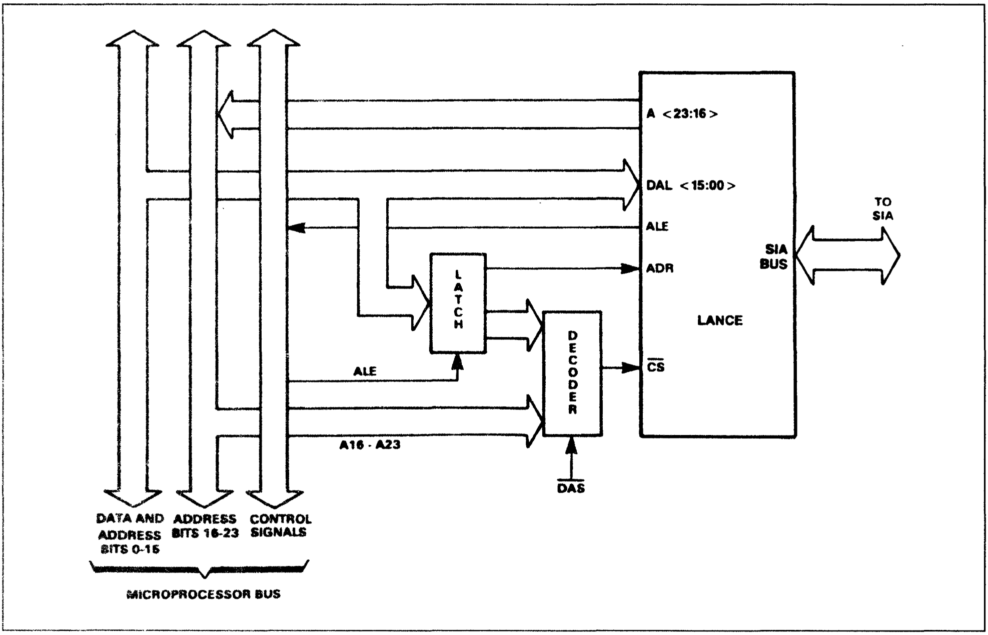


Figure 9 : Demultiplexed Bus Interface.

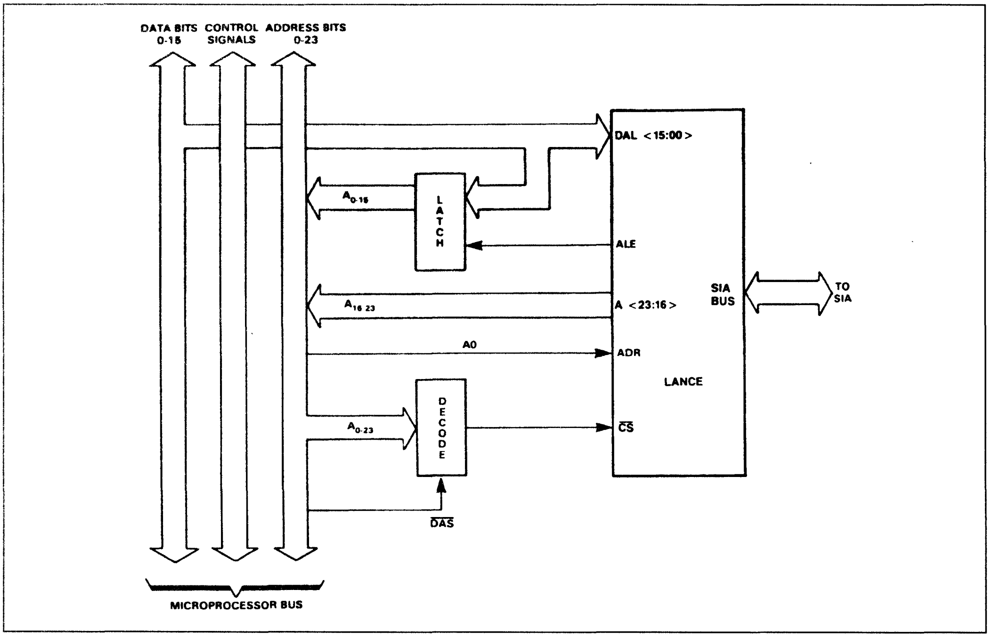


Figure 10 : Bus Master Timing.

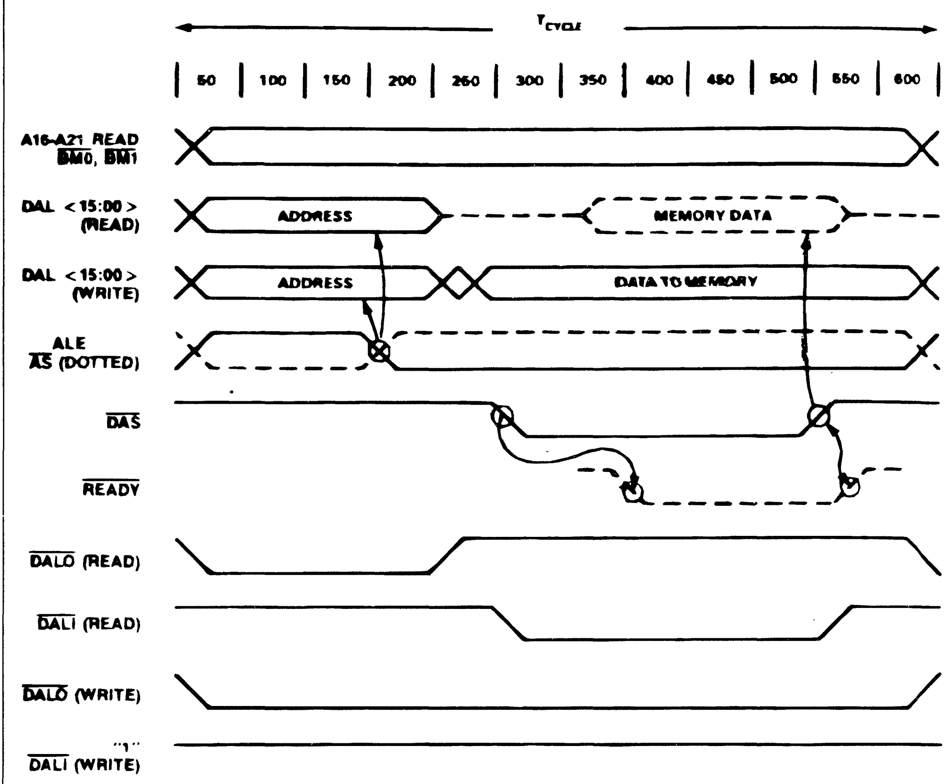
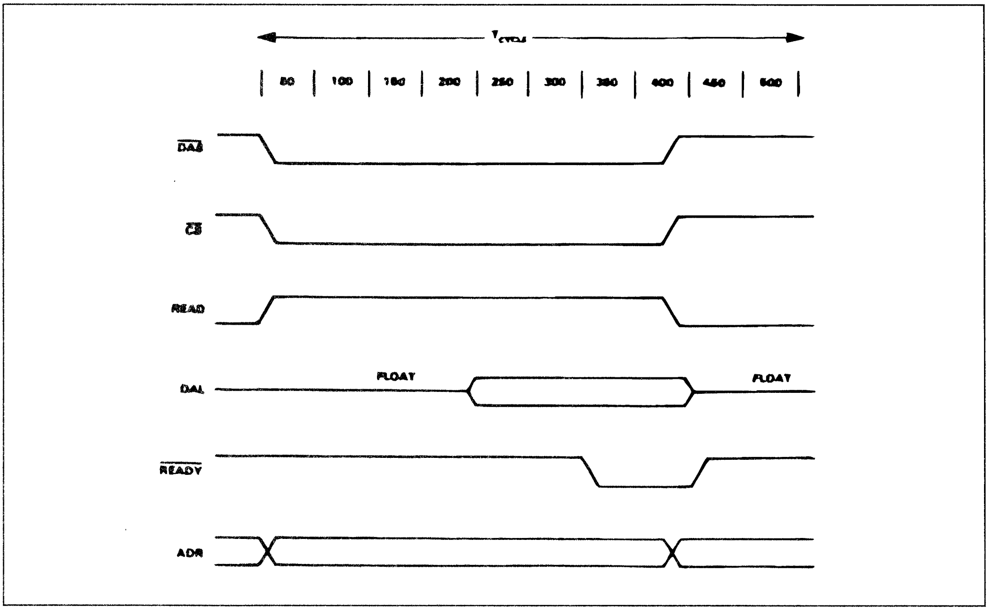
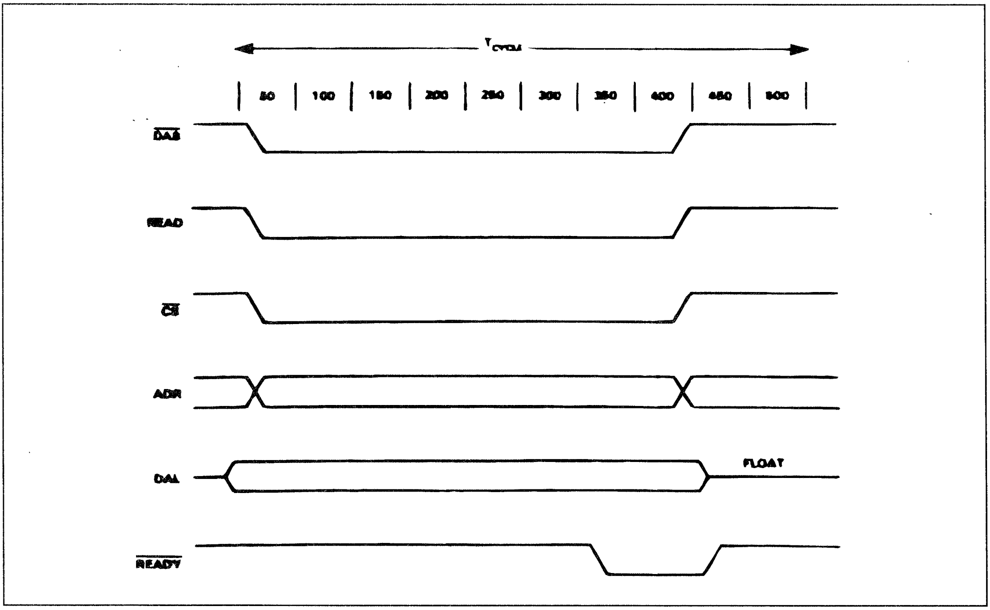


Figure 11 : Bus Slave Read Timing for CSR0, RAP, and CSR3.



Timing for CSR1 and CSR2 are similar but the DELAY to READY becomes 1250 instead of 350µs.

Figure 12 : Bus Slave Write Timing for CSR0, RAP, and CSR3.



Timing for CSR1 and CSR2 are similar but the DELAY to READY becomes 1250 instead of 350µs.

1.3.7.3. Reference Documents. The following documents provide a good overview and background for Ethernet. They can be requested from :

Ethernet
Xerox Office Systems Division
Dept. A
3333 Coyote Hills Rd.
Palo Alto, CA 94304

1. The Ethernet, a Local Area Network, Data Link Layer and Physical layer Specifications-Version 2.0, November 1982.
2. John F. Shoch, An Annotated Bibliography on Local Computer Networks, October 1979.
3. The Ethernet Local Network : Three Reports, February 1980.
4. Internet Transport Protocols, Xerox System Integration Standard, December 1981.
5. Courier : The Remote Procedure Call Protocol, Xerox System Integration Standard, December 1981.

CHAPTER 2
PROGRAMMING SPECIFICATIONS

2. INTRODUCTION

2.1. PROGRAMMING SPECIFICATIONS

This section defines the Control and Status Registers and the memory data structures required to program the VLANCE Ethernet Protocol Controller.

2.2. PROGRAMMING THE VLANCE

The VLANCE is designed to operate in an environment that includes close coupling with a local memory and a microprocessor (HOST). The VLANCE is programmed by a combination of registers and data structures resident within the VLANCE and in memory. There are four Control and Status Registers (CSR's) within the VLANCE which are programmed by the HOST device. Once enabled, the VLANCE has the ability to access external buffer memory locations to acquire additional operating parameters. VLANCE has the ability to do independent buffer management as well as transfer data packets to and from and Ethernet. There are three memory structures accessed by VLANCE, as follows :

1. Initialization Block - 12 words in contiguous memory starting on a word boundary. The initialization block is assembled by the HOST, and is accessed by VLANCE. The initialization block contains the operating parameters necessary for device operation. The initialization block is comprised of :
   1. Mode of Operation (1 word)
   2. Physical Address (3 words)
   3. Logical Address Mask (4 words)
   4. Location of Receive and Transmit Descriptor Rings (2 words)
   5. Number of Entries in Receive and Transmit Descriptor Rings (2 words)
2. Receive and Transmit Descriptor Rings - Two ring structures, one each for incoming and outgoing packets. Each entry in the rings is 4 words long. Each entry must start on a quadword boundary. The Descriptor Rings are comprised of :
   1. The address of a data buffer.
   2. The length of that buffer.
   3. Status information associated with the buffer.
3. Data Buffers - Contiguous portions of memory reserved for packet buffering. Data buffers may begin on arbitrary byte boundaries.

In general, the programming sequence of VLANCE may be summarized as :

1. Programming the VLANCE CSR's by a HOST device to locate an initialization block memory.
2. VLANCE loading itself with the information contained within the initialization block.
3. VLANCE accessing the Descriptor Rings for packet handling.

2.3. CONTROL AND STATUS REGISTERS

There are four Control and Status Registers (CRS's) resident within VLANCE. The CSR's are accessed through two bus addressable ports, an address port (RAP), and a data port (RDP).

2.3.1. ACCESSING THE CONTROL AND STATUS REGISTERS. The CSR's are read (or written) in a two step operation. The address of the CSR is written into the address port (RAP) during a bus slave transaction. During a subsequent bus slave transaction, the data being read from (or written into) the data port (RDP) is read from (or written into) the CSR selected in the RAP. Once written, the address in RAP remains unchanged until rewritten. A discrete control input pin (ADR) control input pin (ADR) is provided to distinguish the address port from the data port.

ADR Pin	Port
L	REGISTER DATA PORT (RDP)
H	REGISTER ADDRESS PORT (RAP)

2.3.1.1. Register Data Port (RDP).

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CSR DATA															

CSR DATA

Bits 15:00

Writing data to the RDP loads data into the CSR selected by RAP. Reading the data from RDP reads the data from the CSR selected by RAP. CSR1, CSR2 and CSR3 are accessible only when the STOP bit of CSR0 is set. If an attempt to access CSR1, CSR2, or CSR3 is made without the STOP bit being ser, VLANCE does not respond to the bus transfer. VLANCE will assert READY, but no data will be transferred either into or out of these registers.

2.3.1.2. Register Address Port (RAP).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES														CSR 1:0	

RES

Bits 15:02

Reserved and read as zeroes.

CSR

Bits 01:00

CSR address select bits. READ/WRITE. Selects the CSR to be accessed through the RDP. RAP is cleared by Bus RESET.

CSR <1:0>	CSR
0	CSR0
1	CSR1
2	CSR2
3	CSR3

2.3.2. CONTROL AND STATUS REGISTER DEFINITION

2.3.2.1. Control And Status Register 0 (CSR0)

RAP = 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR	BABL	CERR	MISS	MERR	RINT	TINT	IDON	INTR	INEA	RXON	TXON	TDMD	STOP	STRT	INIT

ERR

Bit 15

(Error Summary) Error Summary is set by the 'OR' of BABL, CERR, MISS and MERR. ERR remains set as long as any of the error flags are true. ERR is read only, writing it has no effect. It is cleared by RESET or by setting the stop bit.

BABL

Bit 14

(Babble) BABL is a transmitter timeout error. It indicates that the transmitter has been on longer than the time required to send the maximum length packet. BABL will be set if the number of bytes transmitted exceeds 1518. When BABL is set, an interrupt will be generated if INEA = 1. BABL is READ/CLEAR ONLY and is set by VLANCE and cleared by writting a "1" into the bit. Writting a "0" has no effect. RESET or setting the STOP bit clears it.

CERR

Bit 13

(Collision Error) Collision Error indicates that the collision input to the chip failed to activate within 2 µsec after a chip initiated transmission was completed. Collision after transmission is a transceiver test feature. CERR is READ/CLEAR ONLY. The chip sets it and clears it by writing a "1" into the bit. Writing a "0" has no effect. RESET or setting the STOP bit clears it.

MISS

Bit 12

(Missed Packet) Missed Packet is set whenever a packet arrives and passes address recognition, but is lost because the receiver does not own a receive buffer. When MISS is set, an interrupt will be generated if INEA = 1. MISS is READ/CLEAR ONLY and is set by VLANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. RESET or setting the STOP bit clears it.

MERR

Bit 11

(Memory Error) Memory Error sets when VLANCE is the Bus Master and has not received READY within 25.6 µsec after asserting the address on the DAL lines. When a Memory Error is detected, the receiver and transmitter are turned off and an interrupt is generated if INEA = 1. MERR is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. RESET or setting the STOP bit clears it.

RINT

Bit 10

(Receiver Interrupt) Receiver Interrupt is set after VLANCE updates the last entry in the Receive Descriptor Ring for the completed packet. When RINT is set, an interrupt is generated if INEA = 1. RINT is READ/CLEAR ONLY and is set by VLANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. RESET or setting the STOP bit clears it.

TINT

Bit 09

(Transmitter Interrupt) Transmitter Interrupt is sert after VLANCE updates the last entry in the Transmit Descriptor Ring for that completed packet. When TINT is set, an interrupt is generated if INEA = 1. TINT is READ/CLEAR ONLY and is set by VLANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. RESET or setting the STOP bit clears it.

IDON

Bit 08

(Initialization Done) Initialization Done indicates that VLANCE has completed the initialization procedure started by setting the INIT bit. When IDON is set, the chip has read the Initialization Block from memory and stored the new parameters. When IDON is set, an interrupt is generated if INEA = 1. IDON is READ/CLEAR ONLY and is set by VLANCE and cleared by writing a "1" into the bit. Writing a "0" has no effect. RESET or setting the STOP bit clears it.

INTR

Bit 07

(Interrupt Flag) Interrupt Flag indicates that one or more of the following interrupt causing conditions has occurred : BABL, MISS, MERR, RINT, TINT, IDON. If INEA = 1 and INTR = 1 the INTR output pin will be low. INTR is READ ONLY, writing this bit has no effect. INTR is cleared by RESET or by setting the STOP bit.

INEA

Bit 06

(Interrupt Enable) Interrupt Enable allows the INTR Output pin to be driven low when the Interrupt Flag is set. If INEA = 1 and INTR = 1 the INTR pin will be low. If INEA = 1 and INTR = 1 the INTR pin will be low. If INEA = 0 the INTR pin will be high, regardless of the state of the Interrupt Flag. INEA is READ/WRITE set by writing a "1" into this bit and is cleared by writing a "0" into this bit or by RESET or by setting the STOP bit.

RXON

Bit 05

(Receiver On) Receiver On indicates that the receiver in enabled. RXON and IDON are set at the same time, if the DRX bit in the Mode Register is a "0". RXON is cleared by MERR or STOP being set or by RESET. RXON is READ ONLY, writing this bit has no effect. RXON is gated by the STRT bit ; thus it will always be read as a "0" until STRT is set.

TXON

Bit 04

(Transmitter On) Transmitter On indicates that the transmitter is enabled. TXON and IDON are set at the same time, if the DTX bit in the Mode Register is "0". TXON is cleared by MERR, or STOP being set, a TRANSMIT UNDERFLOW, or by RESET. TXON is READ ONLY ; writing this bit has no effect. TXON is gated by the STRT bit ; thus it will always be read as a "0" until STRT is set.

TDMD

Bit 03

(Transmit Demand) When set, Transmit Demand causes VLANCE to access the Transmit Descriptor Ring without waiting for the polltime interval to elapse. (about 1.6ms). TDMD need not be set to transmit a packet, it merely hastens VLANCE's response to a Transmit Descriptor Ring entry insertion by the host. TDMD is WRITE WITH ONE ONLY and microcode clears if after it is used. It may read as a "1" for as short time after it is written because the VLANCE microcode may have been busy when TDMD was set. It is also cleared by RESET or by setting the STOP bit. Writing a "0" in this bit has no effect.

STOP

Bit 02

(Stop) STOP disables VLANCE from all external activity when set and clears the internal logic. Setting STOP is the equivalent of asserting Bus RESET. VLANCE remains inactive and STOP remains set until the STRT or INIT bit is set. If STRT, INIT and STOP are all set together, STOP will override the other bits and only STOP will be set. STOP is READ/WRITE WITH ONE ONLY and set by RESET. Writing a "0" to this has no effect.

STRT

Bit 01

(Start) Start enables VLANCE to send and receive packets, perform direct memory access and do buffer management. If STRT and INIT are set together, the INIT function will be executed first. STRT is READ/WRITE WITH ONE ONLY. Writing a "0" into this bit has no effect. STRT is cleared by RESET or by setting the STOP bit.

INIT

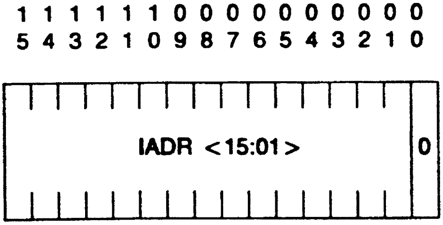
Bit 00

(Initialize) When set, Initialize causes VLANCE to begin the initialization procedure and access the Initialization Block. If STRT and INIT are set together, the INIT function is executed first. INIT is READ/WRITE WITH "1" ONLY. Writing a "0" into this bit has no effect. INIT is cleared by RESET or by setting the STOP bit.

2.3.2.2 Control and Status Register 1 (CSR1).

RAP = 1

READ/WRITE : Accessible only when the STOP bit of CSR0 is a ONE. Access at any other time will not be responded to by VLANCE. READY will be asserted but no data will be transferred. CSR1 is unaffected by RESET.



IADR

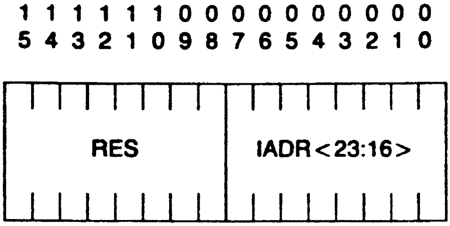
Bits 15:01

The low order 16 bits of the address of the first word (lowest address) in the Initialization Block. Bit 00 must be zero.

2.3.2.3. Control and Status Register 2 (CSR2).

RAP = 2

READ/WRITE : Accessible only when the STOP bit of CSR0 is a ONE. Access at any other time will not be responded to by VLANCE. READY will be asserted but no data will be transferred. CSR2 is unaffected by RESET.



RES

Bits 15:08

Reserved.

IADR

Bits 07:00

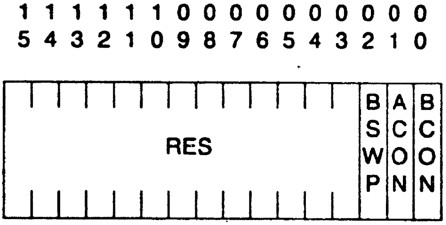
The high order 8 bits of the address of the first word (lowest address in the Initialization Block).

2.3.2.4 Control And Status Registers 3 (CSR3).

CSR3 allows redefintion of the Bus Master interface.

RAP = 3

READ/WRITE : Accessible only when the STOP bit of CSR0 is a ONE. Access at any other time will not be responded to by VLANCE. READY will be asserted but no data will be transferred. CSR3 is cleared by RESET or by setting the STOP bit in CSR0.



RES

Bits 15:03

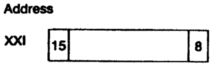
Reserved/read as "0".

BSWP

Bit 02

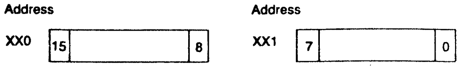
(Byte Swap) Byte Swap allows VLANCE to operate with memory organizations that have bits <07 : 00> at even addresses with bits <15 : 08> at odd addresses or vice versa.

With Byte Swap = 0 :



This memory organization is used with the LSI 11 microprocessor and the 8086 microprocessor.

With Byte Swap = 1 :



This memory organization is used with the MK68000, MK68200, and Z8000 microprocessors. Only data from SILO transfers are swapped. Initialization Block Data and Ring Descriptor entries are not swapped. BSWP is READ/WRITE and cleared by RESET or by setting the STOP bit in CSR0.

ACON

Bit 01

(ALE Control) ALE Control defines the assertive state of ALE/AS when VLANCE is a Bus Master. ACON is READ/WRITE and cleared by $\overline{\mathrm{RESET}}$ or by setting the STOP bit in CSR0.

ACON	ALE/$\overline{\mathrm{AS}}$
0	ASSERTED HIGH (ALE)
1	ASSERTED LOW ($\overline{\mathrm{AS}}$)

BCON

Bit 00

(Byte Control) Byte Control redefines the Byte Mask and Hold I/O Pins. BCON is READ/WRITE and cleared by $\overline{\mathrm{RESET}}$ or by setting the STOP bit in CSR0.

BCON	I/O PIN 16	I/O PIN 15	I/O PIN 17
0	$\overline{\mathrm{BM1}}$	$\overline{\mathrm{BM0}}$	$\overline{\mathrm{HOLD}}$
1	$\overline{\mathrm{BUSAKO}}$	$\overline{\mathrm{BYTE}}$	$\overline{\mathrm{BUSRQ}}$

2.4. INITIALIZATION

2.4.1. INITIALIZATION BLOCK. VLANCE initialization include the reading of the initialization block in memory to obtain the operating parameters. The following is a definition of the Initialization Block. The Initialization Block is read by VLANCE when the INIT bit in CSR0 is set. The INIT bit should be set before or concurrent with the STRT bit to ensure proper parameter initialization and VLANCE operation. After VLANCE has read the Initialization Block, IDON is set in CSR0 and an interrupt is generated if INEA = 1.

HIGHER ADDRESSES ↑

TLEN - TDRA <23:16>	IADR <23:00> +16
TDRA <15:00>	IADR <23:00> +14
RLEN - RDRA <23:16>	IADR <23:00> +12
RDRA <15:00>	IADR <23:00> +10
LADRF <63:48>	IADR <23:00> +E
LADRF <47:32>	IADR <23:00> +C
LADRF <31:16>	IADR <23:00> +A
LADRF <15:00>	IADR <23:00> +8
PADR <47:32>	IADR <23.00> +06
PADR <31:16>	IADR <23:00> +04
PABR <15:00>	IADR <23:00> +02
MODE	IADR <23:00> +00

BASE ADDRESS OF BLOCK

2.4.1.1. Mode. The Mode Register allows alteration of VLANCE's operating parameters. Normal operation is with the Mode Register clear.

IADR <23 : 00> + 00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROM	RES				RAT2	RAT1	RATO	MAN	INTL	DRTY	COLL	DTCR	LOOP	DTX	DRX

PROM

Bit 15

(Promiscuous Mode) When PROM = 1, all incoming addresses are accepted. This bit must be set in internal loopback if a physical address is not used.

RAT <2 : 0>

Bits 10 : 08

(Divide By Rate Control) These rate bits determine the transmit data rate as a Function of System Clock. The available divide by modes are 1, 2, 4, 6, 8, and 10.

RAT 2	RAT 1	RAT 0	DIVISIOR
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	6
1	0	0	8
1	0	1	10

MAN

Bit 01

(Manchester) For this bit to be active at least one of the "RAT" bits must be set to a "1". (Divide by two or Greater). Setting this bit changes the NRZ transmit data stream to a Manchester encoded data stream.

RES

Bit 14:11

(Reserved)

Bit 06

(Internal Loopback) Internal Loopback is used with the LOOP bit to determine where the loopback is to be done. Internal loopback allows VLANCE to receive its own transmitted packet. Since this represents full duplex operation, the packet size would be limited by the SILO size, which 48 bytes. However, a SILO full flag is generated after 32 bytes are loaded into the SILO. This limits the transmit buffer size to 32 bytes in internal or extended loopback. With transmit CRC enabled, the VLANCE generates the 4-byte CRC code and appends it to the data. Thus, the receive buffer is filled with 36 bytes and the host CPU checks the CRC result. With transmit CRC disabled, the host CPU provides 4 bytes of CRC as part of the 32 bytes in the transmit buffer. The VLANCE checks the CRC on reception and transfers only 28 bytes of "data" to the receive buffer. After each Internal Loopback packet, VLANCE should be reinitialized.

INTL is only valid if LOOP = 1, otherwise it is ignored.

LOOP	INTL	LOOPBACK
0	X	NO LOOPBACK, NORMAL OPERATION
1	0	EXTERNAL
1	1	INTERNAL

DRTY

Bit 05

(Disable Retry) When DRTY = 1, VLANCE attempts only one packet transmission If there is a collision on the first transmission attempt, a Retry Error (RTRY) is reported in Transmit Message Descriptor 3 (TMD3).

COLL

Bit 04

(Force Collision) This bit allows the collision logic to be tested. VLANCE must be in internal loopback for COLL to be valid. If COLL = 1, a collision will be forced during the subsequent transmission attempt. This will result in 1 or 16 total transmission attempts with a retry error reported in TMD3. The number of attemps depends upon the state of DRTY (Bit 05).

DTCR

Bit 03

(Disable Transmit CRC) When DTCR = 0, the transmitter generates and appends a CRC to the transmitted packet. When DTCR = 1, the CRC logic is allocated to the receiver and no CRC is generated and sent with the transmitted packet. During loopback, DTCR = 0 causes a CRC to be generated on the transmitted packet but the receiver will not perform a CRC check since the CRC logic is shared and cannot generate and check CRC at the same time. The generated CRC is written into memory with the data and can be checked by the host software. If DTCE = 1 during loopback the host software must append a CRC value to the transmit data. The receiver checks the CRC on the received data and reports any errors.

LOOP

Bit 02

(Loopback) Loopback allows VLANCE to operate in full duplex mode for test purposes. The maximum packet size is limited to 36 bytes as described above for the INTL bit. During loopback, the runt packet filter is disabled because the maximum packet is forced to be smaller than the minimum size Ethernet packet (64 bytes). LOOP = 1 allows simultaneous transmission and reception for a message constrained to fit within the SILO. The chip waits until the entire message is in the SILO before serial transmission begins. The incoming data stream fills the SILO from behind as it is being emptied. Moving the received message out of the SILO to memory does not begin until reception has ceased. In loopback mode, transmit data chaining is not possible. Receive data chaining is allowed regardless of the receive buffer length. In normal operation, the receive buffers must be 64 bytes long, to allow time for buffer lookahead.

DTX

Bit 01

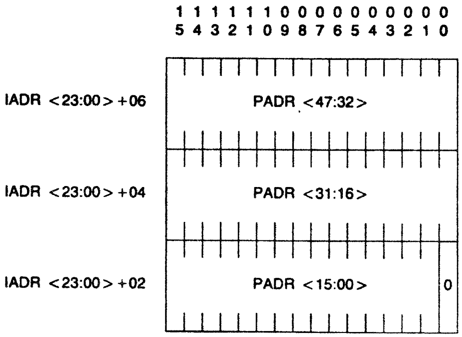
(Disable the Transmitter) Disable the Transmitter causes VLANCE not to access the Transmit Descriptor Ring and therefore no transmissions are attempted. DTX disables TXON from being set when initialization is complete.

DRX

Bit 00

(Disable the Receiver) Disable the Receiver causes VLANCE to reject all incoming packets and not access the Receive Descriptor Ring. DRX disables RXON from being set when initialization is complete.

2.4.1.2. Physical Address.



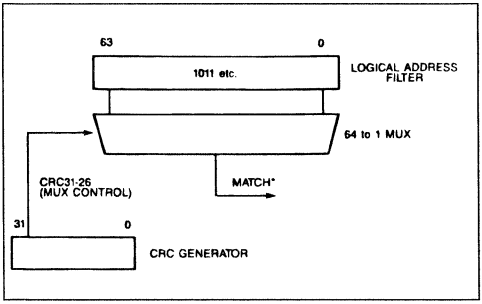
PADR

Bits 47 : 00

(Physical Address) Physical Address is the unique 48-bit physical address assigned to VLANCE. PADR <0> must be zero.

2.4.1.3. Logical Address Filter. The Logical Address Filter is a 64 bit mask composed of four sixteen bit registers LADRF <63 : 00> in the initialization block that is used to accept incoming Logical Addresses. This is an imperfect filter that requires the host processor to do the final filtering. The first bit of the incoming address must be a "1" for either the Logical Address Filter or the Broadcast Address decode to be enabled. Otherwise the incoming address is a physical address and is compared against the contents of PADR <47 : 00> that was loaded through the Initialization Block.

All incoming data goes through the CRC Generator. In the case of a logical address, the six most significant bits of the CRC Generator are strobed into the Hash Register after the 48th bit of the logical address has gone through this circuitry. This 6-bit address then selects one of the 64 bits in the Logical Address Filter. If the mask bit selected is a "1", the address is accepted and the packet will be put into the current receive buffer space. The task of mapping a logical address to one of 64 bits positions is a tedious one that requires a simple computer program to generate the CRC codes for the addresses desired. The Ethernet CRC Polynomial is CRC-32, which is : $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$. Figure 13 shows one such mapping. (This is one of 2^{26} possible mappings). Hash Address 00 will select bit 0 and Hash Address 63 will select bit 63.



* If MATCH = 1, the packet is accepted.
If MATCH = 0, the packet is rejected.

The Broadcast address, which is all ones, is decoded independent of the Logical Address Filter (Broadcast Address will also map to bit 47 of the Logical Address Filter). If the Logical Address Filter is loaded with all zeroes, all incoming logical addresses except Broadcast will be rejected.

LADRF

Bits 63 : 00

The 64bit mask used by VLANCE to accept logical addresses.

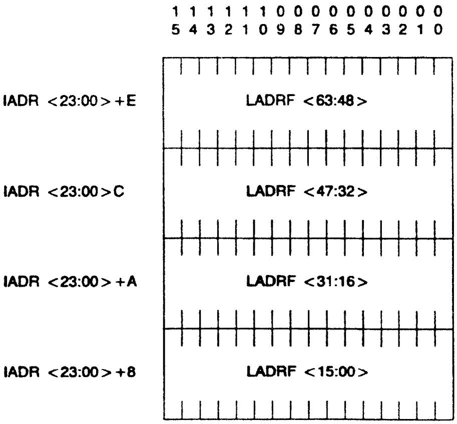
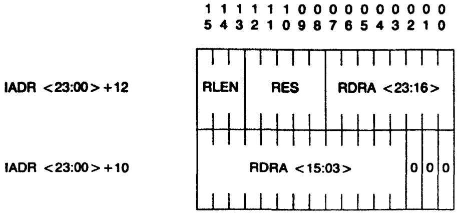


Figure 13 : Mapping of Logical Address to Filter Mask.

LAF REG. BITS SET	LAF LOC.	destination address accepted	LAF REG. BITS SET	LAF LOC.	destination address accepted
	DEC.	(HEX)		DEC.	(HEX)
0	0	FF FF FF FF FF 65	0	32	FF FF FF FF FF D1
	1	FF FF FF FF FF 55		33	FF FF FF FF FF F1
L	2	FF FF FF FF FF 15	L	34	FF FF FF FF FF B1
A	3	FF FF FF FF FF 35	A	35	FF FF FF FF FF 91
F	4	FF FF FF FF FF B5	F	36	FF FF FF FF FF 11
	5	FF FF FF FF FF 95		37	FF FF FF FF FF 31
0	6	FF FF FF FF FF D5	2	38	FF FF FF FF FF 71
	7	FF FF FF FF FF F5		39	FF FF FF FF FF 51
	8	FF FF FF FF FF DB		40	FF FF FF FF FF 7F
	9	FF FF FF FF FF FB		41	FF FF FF FF FF 4F
	10	FF FF FF FF FF BB		42	FF FF FF FF FF 1F
	11	FF FF FF FF FF 8B		43	FF FF FF FF FF 3F
	12	FF FF FF FF FF 0B		44	FF FF FF FF FF BF
	13	FF FF FF FF FF 3B		45	FF FF FF FF FF 9F
	14	FF FF FF FF FF 7B		46	FF FF FF FF FF DF
15	15	FF FF FF FF FF 5B	15	47	FF FF FF FF FF EF
0	16	FF FF FF FF FF 27	0	48	FF FF FF FF FF 93
	17	FF FF FF FF FF 07		49	FF FF FF FF FF B3
	18	FF FF FF FF FF 57		50	FF FF FF FF FF F3
	19	FF FF FF FF FF 77		51	FF FF FF FF FF D3
L	20	FF FF FF FF FF F7	L	52	FF FF FF FF FF 53
A	21	FF FF FF FF FF C7	A	53	FF FF FF FF FF 73
F	22	FF FF FF FF FF 97	F	54	FF FF FF FF FF 23
	23	FF FF FF FF FF A7		55	FF FF FF FF FF 13
1	24	FF FF FF FF FF 99	3	56	FF FF FF FF FF 3D
	25	FF FF FF FF FF B9		57	FF FF FF FF FF 0D
	26	FF FF FF FF FF F9		58	FF FF FF FF FF 5D
	27	FF FF FF FF FF C9		59	FF FF FF FF FF 7D
	28	FF FF FF FF FF 59		60	FF FF FF FF FF FD
	29	FF FF FF FF FF 79		61	FF FF FF FF FF DD
	30	FF FF FF FF FF 29		62	FF FF FF FF FF 9D
15	31	FF FF FF FF FF 19	15	63	FF FF FF FF FF BD

2.4.1.4. Receive Descriptor Ring Pointer.



Bits 15 : 13

(Receive Ring Length) Receive Ring Length is the number of entries in the Receive Ring expressed as a power of two.

RLEN	NUMBER OF ENTRIES
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

RES

Bits 12 : 08

(Reserved)

RDRA

Bits 07 : 00 and 15 : 03

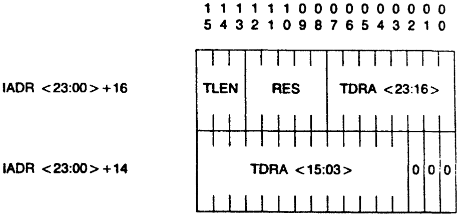
(Receive Descriptor Ring Address) Receive Descriptor Ring Address is the base address (lowest address) of the Receive Descriptor Ring.

RDRA

Bits 02 : 00

(Must Be Zeros) These bits are RDRA <02 : 00> and must be zeroes because the Receive Rings are aligned on quadword boundaries.

2.4.1.5. Transmit Descriptor Ring Pointer.



TLEN

Bits 15 : 13

(Transmit Ring Length) Transmit Ring Length is the number of entries in the Transmit Ring expressed as a power of two.

TLEN	NUMBER OF ENTRIES
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

RES

Bits 12 : 08

(Reserved)

TDRA

Bits 07 : 00 and 15 : 03

(Transmit Descriptor Ring Address) This address is the base address (lowest address) of the Transmit Descriptor Ring.

Bits 02 : 00

(Must Be Zeros) These bits must be zeroes because the Transmit Rings are aligned on quadword boundaries.

2.5. BUFFER MANAGEMENT

Buffer Management is accomplished through message descriptors organized in ring structures in memory. Each message descriptor entry is four words long. There are two rings allocated for the VLANCE : a Receive ring and a Transmit ring. The VLANCE is capable of polling each ring for buffers either to empty or fill with packets to or from the channel. The VLANCE is also capable of entering status information in the descriptor entry. When polling, VLANCE is limited to looking one ahead of the descriptor entry with which it is currently working. The speed of the data stream restricts the receiver buffer size to a minimum of 64 bytes to avoid an overflow when chaining receive buffers. The location of the descriptor rings and their length are found in the initialization block, accessed during the initialization procedure by VLANCE. Writing a "ONE" into the STRT bit of CSR0 will cause VLANCE to start accessing the descriptor rings and enable it to send and receive packets. The VLANCE communicates with a HOST device (probably a microprocessor) through the ring structures in memory. Each entry in the ring is either "owned" by VLANCE or the HOST. There is an ownerschip bit (OWN) in the message descriptor entry. Mutual exclusion is accomplished by a protocol which states that each device can only relinquish ownership of the descriptor entry to the other device ; it can never take ownership, and each device cannot change the state of any field in an entry after it has relinquished ownership. When chaining buffers, the minimum transmit buffer size is restricted to 100 bytes (to avoid mutual exclusion violations, which could occur following a collision). Otherwise, VLANCE would access a buffer to which it had relinquished ownership (to reinitialize a transmission interrupted by a collision).

2.5.1. DESCRIPTOR RINGS. Each descriptor in a ring in memory is a 4 word entry. The following is the format of the receive and the transmit descriptors.

2.5.1.1. Receive Message Descriptor Entry.

2.5.1.1.1. Receive Message Descriptor 0 (RMD0)

MEMORY ADDRESS : XXXXXXX0

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
LADR															

LADR

Bits 15 : 00

The Low Order 16 address bits of the buffer pointed to by this descriptor LADR is written by the Host and unchanged by VLANCE.

2.5.1.1.2. Receive Message Descriptor 1 (RMD1)

MEMORY ADDRESS : XXXXXXX2

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
OWN	ERR	FRAM	OFLO	CRC	BUFF	STP	ENP	HADR							

OWN

Bit 15

This bit indicates that either the Host owns the descriptor entry (OWN = 0) or VLANCE owns it (OWN = 1). The chip clears the OWN bit after filling the buffer pointed to by the descriptor entry. The Host sets the OWN bit after emptying the buffer. Once the VLANCE or Host relinquishes ownership of a buffer, it may not change any field in the four words that comprise the descriptor entry.

ERR

Bit 14

(Error Summary) Error Summary is the "OR of FRAM, OFLO, CRC or BUFF, ERR is set by VLANCE whent it releases the buffer and is cleared by the Host.

FRAM

Bit 13

(Framing Error) Framing Error indicates that the incoming packet contained both a non-integer multiple of eight (8) bits and a CRC error. In internal loopback, whenever a CRC error occurs, FRAM will always be set. FRAM is set by VLANCE when it releases the buffer and is cleared by the Host.

OFLO

Bit 12

(Overflow) Overflow error indicates that the receiver has lost all or part of the incoming packet due to an inability to store the packet in a memory buffer before the internal SILO overflowed. OFLO is set by VLANCE when it releases the buffer and is cleared by the Host.

CRC

Bit 11

CRC indicates that the receiver has detected a CRC error on the incoming packet. CRC is set by VLANCE when it releases the buffer and is cleared by the Host.

BUFF

Bit 10

(Buffer Error) Buffer Error is set either when VLANCE has utilized all its allocated buffers or if the next status is not acquired in time while data chaining a received packet. BUFF is set by VLANCE when it releases the buffer and is cleared by the Host. If a Buffer Error occurs, an Overflow Error also occurs because VLANCE tries to acquire the next buffer until the SILO overflows.

STP

Bit 09

(Start of Packet) Start of Packet indicates that this is the first buffer used by VLANCE for this packet. It is used for data chaining buffers. STP is set by VLANCE when it releases the buffer and is cleared by the Host.

ENP

Bit 08

(End of Packet) End of Packet indicates that this is the last buffer used by VLANCE for this packet. It is used for data chaining buffers. If both STP and ENP were set, the packet would fit into one buffer and there was no data chaining. ENP is set by VLANCE when it releases the buffer and is cleared by the Host.

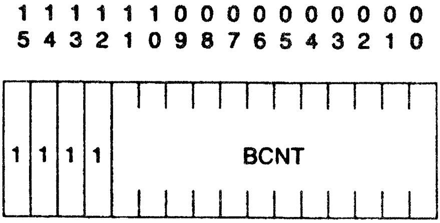
HADR

Bits 07 : 00

The High Order 8 address bits of the buffer pointed to by this descriptor. This field is written by the Host and unchanged by VLANCE.

2.5.1.1.3. Receive Message Descriptor 2 (RMD2)

MEMORY ADDRESS : XXXXXXX4



Bits 15 : 12

(Must Be Ones) This field is written by the Host and unchanged by VLANCE.

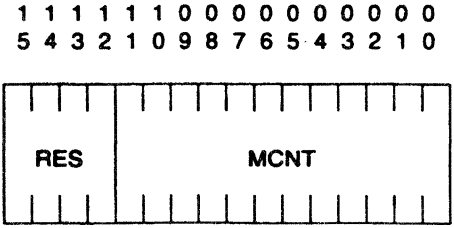
BCNT

Bits 11 : 00

(Buffer Byte Count) Buffer Byte Count is the length of the buffer pointed to by this descriptor expressed as a two's complement number. This field is written by the Host unchanged by VLANCE. The minimum buffer size is 64 bytes.

2.5.1.1.4. Receive Message Descriptor 3 (RMD3)

MEMORY ADDRESS : XXXXXXX6



RES

Bits 15 : 12

(Reserved) Read as zeroes.

MCNT

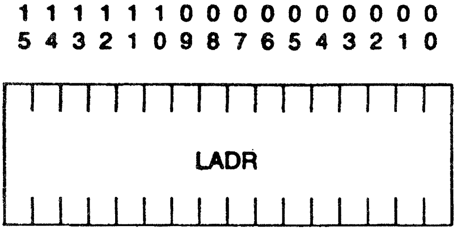
Bits 11 : 00

(Message Byte Count) Message Byte Count is the length in bytes of the received message. MCNT is valid only when ERR is clear and ENP is set. MCNT is written by VLANCE and is cleared by the Host. In data chaining, RMD3 is only written to after the last buffer status is updated. Only the status word is updated for intermediate buffers in the data chain.

2.5.1.2. Transmit Message Descriptor Entry.

2.5.1.2.1. Transmit Message Descriptor 0 (TMD0)

MEMORY ADDRESS : XXXXXXX0



LADR

Bits 15 : 00

The Low Order 16 address bits of the buffer pointed to by this descriptor. LADR is written by the Host and unchanged by VLANCE.

2.5.1.2.2. Transmit Message Descriptor 1 (TMD1)
MEMORY ADDRESS : XXXXXXX2

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
OWN	ERR	RES	MORE	ONE	DEF	STP	ENP	HADR							

OWN

Bit 15

This bit indicates that either the Host owns the descriptor entry (OWN = 0) or VLANCE it (OWN = 1). The host sets the OWN bit after filling the buffer pointed to by the descriptor. VLANCE clears the OWN bit after transmitting the contents of the buffer. Both the Host and VLANCE must not alter a descriptor entry after it has relinquished ownership.

ERR

Bit 14

(Error Summary) Error Summary is the "OR" of LCOL, LCAR, UFLO or RTRY. ERR is set by VLANCE when it releases the buffer and is cleared by the Host.

RES

Bit 13

(Reserved) VLANCE will write this bit with a "0".

MORE

Bit 12

MORE indicates that more than one retry was needed to transmit a packet. MORE is set by VLANCE when it releases the buffer and is cleared by the Host.

ONE

Bit 11

ONE indicates that exactly one retry was needed to transmit a packet. ONE is set by VLANCE when it releases the buffer and is cleared by the Host. ONE is not valid if LCOL in TMD3 is set.

DEF

Bit 10

(Deferred) Deferred indicates that VLANCE had to defer while trying to transmit a packet. This condition occurs when the channel is busy when VLANCE is ready to transmit. DEFER is set by VLANCE when it releases the buffer and is cleared by the Host.

STP

Bit 09

(Start of Packet) Start of Packet indicates that this is the first buffer to be used by VLANCE for this packet. It is used for data chaining buffers. STP is set by the Host and is unchanged by VLANCE.

ENP

Bit 08

(End of Packet) End of Packet indicates that this is the last buffer to be used by VLANCE for this packet. It is used for data chaining buffers. If both STP and ENP are set, the packet fits into one buffer and there is no data chaining. ENP is set by the Host and is unchanged by VLANCE.

HADR

Bits 07 : 00

The High Order 8 address bits of the buffer pointed to by this descriptor. This field is written by the Host and is unchanged by VLANCE.

2.5.1.2.3. Transmit Message Descriptor 2 (TMD2)
MEMORY ADDRESS : XXXXXXX4

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	BCNT											

ONES

Bits 15 : 12

(Must Be Ones) This field is set by the Host and unchanged by VLANCE.

BCNT

Bits 11 : 00

(Buffer Byte Count) Buffer Byte Count is the usable length in bytes of the buffer pointed to by this descriptor expressed as a two's complement negative number. This is the number of bytes from this buffer that will be transmitted by VLANCE. This field is written by the Host and unchanged by VLANCE. The minimum buffer size is 100 bytes when chaining or 64 bytes when not chaining.

2.5.1.2.4. Transmit Message Descriptor 3 (TMD3)

TMD3 is valid only if the ERR bit of TMD1 has been set by VLANCE.

MEMORY ADDRESS : XXXXXXX6

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
BUFF	UFLO	RES	LCOL	LCAR	RTRY	TDR									

BUFF

Bit 15

(Buffer Error) Buffer Error is set by VLANCE during transmission when VLANCE does not fing the ENP flag in the current buffer and does not own the next buffer. BUFF is set by VLANCE and cleared by the host. If a buffer Errors occurs, an Underflow Error will also occur because VLANCE tries to read memory data until the SILO is empty. Buffer Error is valid only if UFLO is set.

UFLO

Bit 14

(Underflow Error) Underflow Error indicates that the transmitter has truncated a message due to data late from memory. UFLO indicates that owing to a lack of data from memory. The SILO has emptied before the end of the packet was reached. UFLO is set by VLANCE and is cleared by the Host.

RES

Bit 13

Reserved bit. VLANCE writes this bit with a "0".

LCOL

Bit 12

(Late Collision) Late Collision indicates that a collision has occured after the slot time of the channel has elapsed. VLANCE does not retry on late collisions. LCOL is set by VLANCE and is cleared by the Host.

LCAR

Bit 11

(Loss of Carrier) Loss of Carrier is set when the carrier presence (RENA) input to VLANCE goes false during a VLANCE-initiated transmission. VLANCE does not retry upon Loss of Carrier. LCAR is set by VLANCE and is cleared by the Host.

RTRY

Bit 10

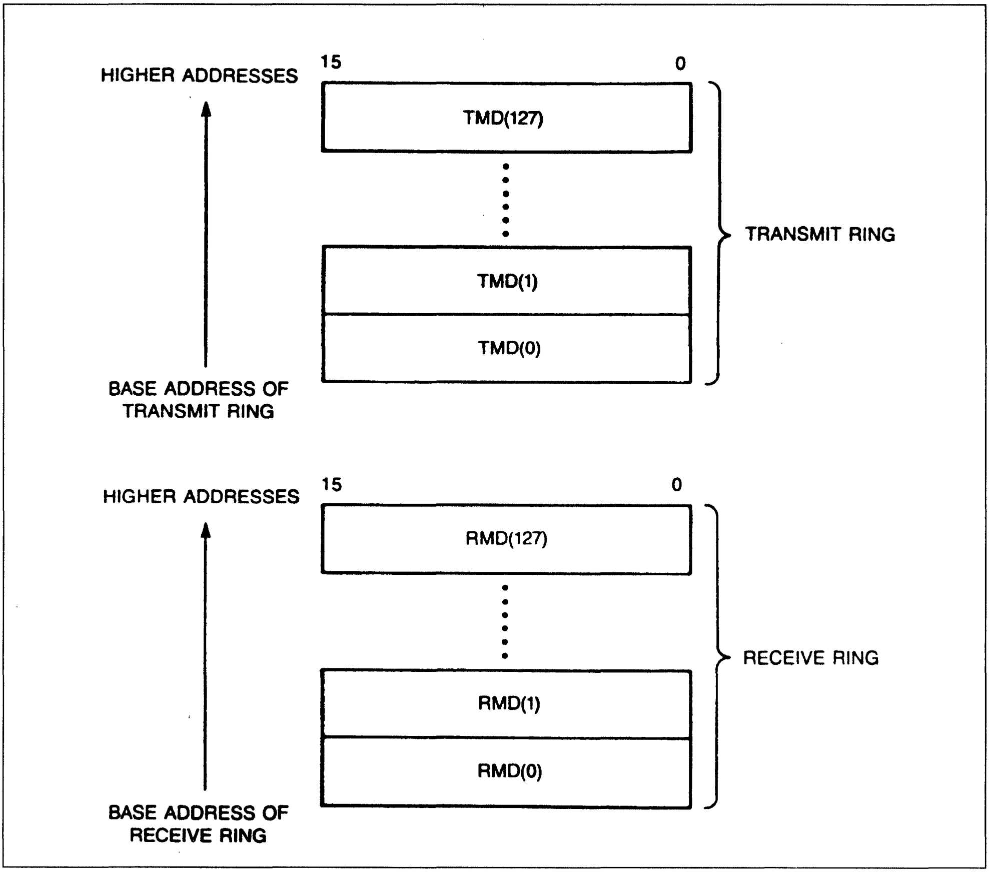
(Retry Error) Retry Error indicates that the transmitter has failed in 16 attempts to successfully transmit a message due to repeated collisions on the medium. If DRTY = 1 in the MODE register, RTRY sets after 1 failed transmission attempt. RTRY is set by VLANCE and is cleared by the Host.

TDR

Bits 09 : 00

(Time Domain Relectometry) Time Domain Reflectometry is an internal counter that counts system clocks from the start of a transmission to the occurrence of a collision. This value is useful in determining the approximate distance to a cable fault. The TDR value is written by VLANCE and is valid only RTRY is set.

DESCRIPTOR RINGS IN MEMORY



CHAPTER 3
FUNCTIONAL SPECIFICATIONS

3. INTRODUCTION

3.1. FUNCTIONAL DESCRIPTION

This section describes the logical elements used to implement the VLANCE Ethernet Controller.

3.2. LOGIC

3.2.1. CLOCK. The VLANCE has its clocks derived from a basic free running 10MHz clock presented to the input pin SCLK. Refer to Section 4 for the clock specification. The microcycle is 200 nanoseconds long, or two basic clock ticks. The microcycle is the basic unit of time in the microsequencer and the control data path. Clock suppression is the act of selectively stretching the microcyle to allow a memory transfer to complete when the Chip is operating as a bus master. Clock suppression can only occur in those microcycles that contain an asserted USUPPRESS bit in the microword register.

3.2.2. MICROSEQUENCER. VLANCE is controlled by an internal microprogram. Chained sequecing is used to advance the program address. Each microword contains the address of the next instruction plus any microbranch and trap information required in the program being executed. The microsequencer operates as a one level deep pipeline. As one micro-instruction is being executed, the next is being accessed. The basic microcycle is 200 nanoseconds long, but may be extended on 200 nanoseconds boundaries to allow memory transfers to complete. During each microcycle, an address is formed to access the program store which is clocked into the microword register at the end of each cycle.

3.2.3. CONTROL DATA PATH. The Control Data Path contains the hardware necessary to build, control, and store the information required to do buffer management and to control the block transfers of data to and from the silo. The major components in this section of logic are a 24-bit adder, a data shuffler, a constant selector, and a static memory. In this memory resides twelve 24-bit Address Registers and ten 16 bit Status/Byte Count Registers.

3.2.4. MESSAGE BYTE COUNT. The message byte count is contained in a 12-bit counter. The message byte count keeps track of the number of bytes entering or leaving the Silo under microprogram control for each transmission or reception. The value contained in the message byte count is written into memory through the MDR as part of the reception process. It is also used for the detection of runt packet on reception, and for the detection of babbling transmissions.

3.2.5. RING END FINDERS. The ring end finders, one each for the receive and transmit rings, determine whether the ring address pointers in the CDP RAM are at the end of the rings, and provide a microbranchable signal, which, when true, informs the microprogram to restore the pointers with the beginning address of the rings. The ring end finders are simply a pair of programmable modulo counters, the value of which is loaded at initialization time. The counters are independently incremented under microprogram control.

3.3. BUS CONTROL

3.3.1. BUS ADDRESS REGISTER. The Bus Address register (BA) is 27 bits wide. It is loaded directly from the Data Shuffler under control of a bit in the microword, ENA BA CLK, at the end of the microcycle. At the same time, the bus address is clocked, byte mask and read/write information, associated with the transfer is clocked into a three bit extension of the BA. Clocking of the BA initiates the bus transfer. The upper 8 bits of the BA drive A <A : 16> directly. The lower sixteen bits of the BA are multiplexed onto DAL <15 : 00> during the address portion of the bus cycle when VLANCE is the Bus Master. This is an internal register and is not directly addressable by the user.

3.3.2. MEMORY DATA REGISTER. The Memory Data Register (MDR) buffers data transfers to and from the I/O bus. The MD is clocked at the end of the microcycle. It is enabled from the ENA MD CLK bit of the microword register. I/O bus data is synchronized to VLANCE prior to loading the MDR.

3.3.3. BUS MASTER CONTROL. VLANCE becomes a Bus Master for the purposes of acquiring data from the initialization block, buffer management, and the block move of data during the transmission or reception process. The Bus Master Control works in partnership with the microprogram. The microprogram is responsible for loading the BA and MDR for a write transaction, and loading the BA and unloading the MDR for a read transaction. Clocking the BA initiates the transfer. The microprogram also provides a clock suppress enable to stall selectively a microcycle until a memory transaction completes, thus providing synchronization between the microprogram and the Bus Master Control. During block transfer (DMA) of data, memory references overlap. VLANCE performs up to 8 data transfers before relinquishing HOLD. Refer to Chapter 4 for timing specifications.

3.3.4. MEMORY TIMEOUT. As a Bus Master, VLANCE detects and recovers from non-existent memory errors. VLANCE waits for a maximum of 25 microseconds for the assertion of READY after it asserts ALE. If VLANCE does not receive READY within that time, it sets the MERR bit of CSR0, negates the RXON and TXON bits, and takes no further action unless either the RESET signal is asserted or the STOP bit of CSR0 is asserted.

3.3.5. BUS SLAVE CONTROL. The Bus Slave control is invoked when a memory transaction occurs and the CS pin is asserted. When this happens, it indicates that one of the four VLANCE CSR's is being accessed. CSR0 provides visibility into VLANCE and is accessed independently of the microprogram. CSR1 and CSR2 hold the address of the initialization block and are resident within the CDP RAM. Accessing CSR1 and CSR2 causes a microtap for access. The microprogram issues the READY signal for CSR1 and CSR2. The Bus Slave Control independently returns READY for CSR0 and CSR3. CSR3 allows the I/O pins to be programmed. CSR1, CSR2, and CSR3 are accessible only when the STOP bit of CSR0 is set. Refer to Chapter 4 for timing specifications.

3.3.6. DISCRETE USER APPARENT REGISTERS. Of the register ports and control and status registers, CSR0, CSR3 and RAP are read and written asynchronously from the parallel I/O bus. Refer to Section 2.3 for definitions of the register ports and control and status registers.

3.4. TRANSCEIVER DATA PATH

3.4.1. SERIAL DATA OUTPUT. Serial output data is presented at the TX Output pin by VLANCE. The presence of the output data stream is indicated by the assertion of the TENA level at the Output pin. TX and TENA are synchronous to the internal clock SCLK.

3.4.2. SERIAL DATA INPUT. Serial input data is presented to VLANCE at the RX Input pin. The serial input data clock is presented at the RCLK Input pin. The presence of the input data stream is indicated by the assertion of the RENA at its Input pin. RX, RCLK, and RENA are asynchronous to the internal clock TCLK. RCLK is used by VLANCE to clock in the input data stream. After the assertion of RENA, VLANCE waits 800 nanoseconds before searching for the start bit. If VLANCE detects a double ZERO prior to detecting a START bit, VLANCE rejects the rest of the packet. Once the Start bit has been detected, VLANCE frames the remaining bit stream into byte boundaries, synchronizes the bytes to the internal clock, and loads the Silo if not otherwise disabled.

3.4.3. SILO. The SILO provides buffer storage for the data being transferred between the parallel bus I/O pins and serial bus I/O pins. The capacity of the SILO is 48 bytes. The fall-through time of the SILO is 200 nanoseconds maximum. The SILO has the following capabilities :

1. SILO OPERATION - TRANSMISSION. Data is loaded into the SILO under microprogram control from the MDR. Data from the SILO goes to the serial output shift register.
2. SILO OPERATION - UNDERFLOW. Underflow occurs during Transmission when the output serial shift register requires data to continue an unbroken bit stream output, but data is not available at the output of the SILO, and the last data byte in the frame has been shifted out. Once the SILO has underflowed, the SILO locks out further reads and writes until cleared by the microprogram.
3. SILO OPERATION - RECEPTION. Data is loaded into the SILO from the serial input shift register during Reception. Data leaves the SILO under microprogram control. The destination is the MDR. Preamble is not loaded into the SILO.
4. SILO OPERATION - OVERFLOW. Overflow occurs during Reception when the SILO is filled and data needs to be transferred from the input serial shift register. Once the SILO has overflowed, the SILO locks out further reads and writes until cleared by the microprogram.
5. SILO OPERATION - RESTORE. During Reception, restoring the SILO refers to the action of discarding the 6 bytes of the destination address that have accumulated in the SILO after an address match has been tested and an address match has not occurred. The same action occurs if less than 6 bytes are received before the packet ends. Note that this is different from clearing the SILO since there may be residual data in the SILO from a previous reception which cannot be lost. During the Transmission process, restoring the SILO refers the action of discarding the accumulated Transmit bytes when bit stream transmission has not yet begun and the receiver becomes active.

6. SILO OPERATION - INDEXING. The SILO is capable of holding residual data from a received packet, and accepting data from a second packet. The SILO is able to mark the end of one packet and the beginning of another.
7. SILO OPERATION - CLEARING. The SILO is cleared as part of the recovery for overflow, underflow, and collision. SILO clearing is the action of flushing all data from the SILO unconditionally by clearing the address counters. The SILO is cleared by a discrete microprogram operation.

3.4.4. SILO - MEMORY BYTE ALIGNMENT. Memory buffers may begin and end on arbitrary byte boundaries. Parallel data is byte aligned between the SILO and the MDR. Byte alignment can be reversed by setting the Byte Swap (BSWP) bit in CSR3.

TRANSMISSION - WORD READ FROM EVEN MEMORY ADDRESS

BWSP = 0 :	SILO BYTE n	gets MDR <07 : 00>
	SILO BYTE n + 1	gets MDR <15 : 08>
BWSP = 1 :	SILO BYTE n	gets MDR <15 : 08>
	SILO BYTE n + 1	gets MDR <07 : 00>

TRANSMISSION - BYTE READ FROM EVEN MEMORY ADDRESS

BSWP = 0 :	SILO BYTE n	gets MDR <07 : 00>
BSWP = 1 :	SILO BYTE n	gets MDR <15 : 08>

TRANSMISSION - BYTE READ FROM ODD MEMORY ADDRESS

BWSP = 0 :	SILO BYTE n	gets MDR <15 : 08>
BWSP = 1 :	SILO BYTE n	gets MDR <07 : 00>

RECEPTION - WORD WRITE TO EVEN MEMORY ADDRESS

BSWP = 0 :	MDR <07 : 00> gets SILO BYTE n
	MDR <15 : 08> gets SILO BYTE n + 1
BSWP = 1 :	MDR <15 : 08> gets SILO BYTE n
	MDR <07 : 00> gets SILO BYTE n + 1

RECEPTION - BYTE WRITE TO EVEN MEMORY ADDRESS

BSWP = 0 :	MDR <07 : 00> gets SILO BYTE n
	MDR <15 : 08> don't care
BSWP = 1 :	MDR <15 : 08> gets SILO BYTE n
	MDR <07 : 00> don't care

RECEPTION - BYTE WRITE TO ODD MEMORY ADDRESS

BSWP = 0 :	MDR <07 : 00> don't care
	MDR <15 : 08> gets SILO BYTE n
BSWP = 1 :	MDR <15 : 08> don't care
	MDR <07 : 00> gets SILO BYTE n

3.4.5. CYCLIC REDUNDANCY CHECK. VLANCE utilizes the 32-bit CRC function used in the Autodin-II network. Refer to the Ethernet Specification (section 6.2.4 Frame Check Sequence Field and Appendix C ; CRC Implementation) for more detail. VLANCE requirements for the CRC logic are the following :

1. TRANSMISSION - MODE <02> LOOP = 0, MODE <03> DTCRC = 0. VLANCE calculates the CRC from the first bit following the Start bit to the last bit of the data field. The CRC value is inverted and appended onto the transmission in one unbroken bit stream.
2. RECEPTION - MODE <02> LOOP = 0. VLANCE performs a check on the input bit stream from the first bit following the Start bit to the last bit in the frame. VLANCE continually samples the state of the CRC check on framed byte boundaries, and, when the incoming bit stream stops, the last sample determines the state of the CRC error. Framing error (FRAM) is not reported if there is no CRC error.
3. LOOPBACK - MODE <02> LOOP = 1, MODE <03> DTRC = 0. VLANCE generates and appends the CRC value to the outgoing bit stream as in Transmission but does not check the incoming bit stream.
4. LOOPBACK - MODE <02> LOOP = 1, MODE <03> DTRC = 1. VLANCE performs the CRC check on the incoming bit stream as in Reception, but does not generate or append the CRC value to the outgoing bit stream.

3.5. TRANSMISSION

Serial transmission consists of sending an unbroken bit stream from the TX pin consisting of :

1. Preamble/Start bit : 64 alternating ONES and ZEROS terminating in two ONEs. The last ONE is the Start bit.
2. Data : The serialized byte stream from the Silo. Shifted out LSB first.
3. CRC : The inverted 32 bit polynomial calculated from the Data field. CRC is not transmitted if :

   1. Transmission of the Data field is truncated for any reason.

   2. CLSN becomes asserted any time during transmission.

   3. VLANCE is in Loopback mode and CRC transmission is disabled (MODE <03> = 1 and MODE <02> = 1).

   4. Mode <03> DTCRC = 1 in a normal transmission mode.

Transmission is indicated at the I/O pin by the assertion of TENA with the first bit of the preamble and the negation of TENA after the last transmitted bit. VLANCE starts transmitting the preamble when the following are satisfied.

1. There is at least one byte of data to be transmitted in the Silo.
2. The inter-packet delay has elapsed.
3. The backoff interval has elapsed, if a retransmission.

3.5.1. INTERPACKET DELAY. The interpacket delay is 9.6 to 10.6 microseconds including synchronization. The interpacket delay interval begins after the negation of the RENA signal, VLANCE continuously monitors the RENA input pin to monitor or generate an interpacket delya. If VLANCE is about to transmit (about to assert the TENA output pin) and RENA is asserted, the chip will not assert TENA until RENA has negated and the interpacket delay has elapsed. Whenever VLANCE is about to transmit and is waiting for the interpacket delay to elapse, it will begin transmission immediately after the interpacket delay interval, independent of the state of RENA.

3.5.2. COLLISION DETECTION AND COLLISION JAM. Collisions are detected by monitoring the CLSN input pin. If CLSN becomes asserted during a Frame Transmission, TENA will remain asserted for at least 32 (but not more than 48) additional bit times (including CLSN synchronization). This additional transmission after collision is referred to as COLLISION JAM. The bit pattern present at the TX output pin is unspecified during COLLISION JAM, but it may not be the 32 bit CRC value corresponding to the (partial) packet transmitted prior to the COLLISION JAM.

If CLSN becomes asserted during the reception of a packet, this reception is immediately terminated. Depending on the timing of collision detection, the following will occur. A collision that occurs within 6 byte times (4.8 microseconds) will result in the packet being rejected because of an address mismatch with the silo write pointer being reset. A collision that occurs within 64 byte times (51.2 microseconds) will result in the packet being rejected since it is a runt packet. A collision that occurs after 64 byte times will result in a truncated packet being written to the memory buffer with the CRC error bit being set in the Status Word of the Receive Ring.

3.5.3. COLLISION BACKOFF. When a transmission attempt has been terminated due to the assertion of CLSN, it is retried by VLANCE up to 15 times until successful, or something else aborts the process (memory timeout). The scheduling of the retransmissions is determined by a controlled randomized process called "truncated binary exponential backoff". Upon the negation of the COLLISION JAM interval, VLANCE calculates a delay before retransmitting. The delay is an integral multiple of the SLOT TIME. The SLOT TIME is 512 bit times. The number of SLOT TIMEs to delay before the nth retransmission attempt is chosen as a uniformly distributed random integer in the range :

$$0 \leq r < 2^k \qquad \text{where } k = \min(n,10)$$

If all 16 attempts fail, VLANCE sets the RTRY bit in the current Transmit Message Descriptor 3 in memory, and steps over the current transmit buffer.

3.5.4. COLLISION - MICROCODE INTERACTION. The microprogram uses the time provided by COLLISION JAM, INTERPACKET DELAY, and the backoff interval to restore the address and byte counts and start loading the Silo in anticipation of retransmission. It is important that VLANCE be ready to transmit when the backoff interval elapses in order to utilize the channel properly.

3.5.5. TIME DOMAIN REFLECTOMETRY. VLANCE contains a time domain reflectometry counter. The TDR counter is ten bits wide. It counts at a 10MHz rate. It is cleared by the microprogram and counts upon the assertion of RENA during transmission. Counting ceases if CLSN becomes true. The counter does not wrap around, once all ONEs are reached in the counter, that value is held until cleared. The value in the TDR is written into memory by the microprogram through the MDR. TDR is used to determine the location of suspected cable faults. Transfer from TDR counter into MDR register occurs only if RTRY is set. Normally, when RTRY is not set, the value of TDR will be all zeros.

3.5.6. HEARTBEAT. During the INTERPACKET DELAY following the negation of TENA, the CLSN input is asserted by some Version 1 and all Version 2 tranceivers as a self-test. If two microseconds of the INTERPACKET DELAY elapse without CLSN having been asserted, VLANCE will set the CERR bit in CSR0 (bit <13>). This function is gated off in the internal loopback mode.

3.6. RECEPTION

Serial reception consists of receiving an unbroken bit stream on the RX I/O pin consisting of :

1. Preamble / Start bit : Two ONES occuring a minimum of 8 bit times after the assertion of RENA. The last ONE is the Start bit.
2. Destination Address : The 48 bits (6 bytes) following the Start bit.
3. Data : The serialized byte stream following the Destination Address. The last four complete bytes of data are the CRC. The Destination Address and the Data are framed into bytes and enter the Silo.

Reception is indicated at the I/O pin by the assertion of RENA and the presence of clock on RCLK while TENA is deasserted.

3.6.1. STATION ADDRESS DETECTION. The station address detect logic checks the destination address of the incoming packet to determine if the packet is addressed to this node. A packet will be accepted if at least one of the following is true :

1. Physical address match : The destination address of the packet exactly matches the physical address of the node.
2. Logical address match : The destination address of the packet is hashed using the CRC. The hash function is used to determine a logical address match.
3. Promiscuous mode : The node accepts all packets regardless of the destination address.
4. Broadcast Detection : The destination address of the packet is the Broadcast Address ; all ones.

3.6.1.1. Physical Address Register. The physical address register is 48 bits wide and contains the physical address of VLANCE. The microprogram loads the physical address from the initialization block through three sequential memory transactions. If the first bit following the Start bit is a ZERO, VLANCE will perform a physical address compare. The following 47 bits are compared, bit for bit, for an exact match. If they don not match, VLANCE will reject the packet. Bit <00> of the physical address register corresponds to the first bit of the destination address field, and bit <47> of the physical address register corresponds to the last bit of the destination address field.

3.6.1.2. Logical Address Filter Register. The logical address filter register is 64 bits wide. The microprogram loads the logical address filter from the initialization block through four sequential memory transactions. If the first bit following the Start bit is a ONE, VLANCE will perform a logical address compare. After the last bit of the destination address is clocked into the CRC check logic, the value of CRC 31:26 is used as an index into the logical address filter register. If the bit selected in the register is not a ONE, the chip will reject the packet.

3.6.1.3. Promiscuous Mode. If MODE <15> PROM = 1, VLANCE will accept all packets, regardless of the destination address.

3.6.1.4. Broadcast Address Detection. VLANCE will always accept all packets sent to the Broadcast Address of all ones.

3.6.2. RUNT PACKET FILTRATION. If, after loading a buffer, the message byte count is less than 64 bytes, VLANCE does not update the ring descriptor entry that pointed to the buffer. Instead VLANCE retains the buffer information for use with the next incoming packet. An incoming message must be greater than 64 bytes to be considered a valid packet.

3.7. LOOPBACK

The normal operation of VLANCE is as a half duplex device. However, to provide an on-line operational test of VLANCE, a pseudo-full duplex mode is provided. In this mode, simultaneous transmission and reception of a loopback packet are enabled with the following constraints :

1. The packet length must be no longer than 32 bytes, exclusive of the CRC.
2. Serial transmission does not begin until the Silo contains the entire output packet.
3. Moving the input packet from the Silo to the memory does not begin until the serial input bit stream terminates.
4. CRC may be generated and appended to the output serial bit stream or may be checked on the input serial bit stream, but not both in the same transaction.
5. In internal loopback, the "RAT" bits must be set to "0".

Loopback is controlled by bits <06,03 : 02> INTL, DTCR, and LOOP of the MODE register. refer to Section 2.4.1.1 for detailed operation of this register.

3.8. MICROPROGRAM OVERVIEW

3.8.1. SWITCH ROUTINE. Upon power-up, the microprogram finds itself in a routine to evaluate the INIT, STRT and STOP bits of CRS0. INIT and STRT are cleared and STOP is set by the hardware by RESET. Setting either INIT or STRT through an I/O transfer to CSR0 clears STOP. Setting STOP through an I/O transfer clears INIT and STRT. After seeing STOP cleared, the microprogram tests the state of INIT. If set, it branches to the initialization routine, returns, and tests the state of STRT. If INIT is clear and STRT is set, the microprogram goes on to the Polling routine without going to the Initialization routine. If, while the STOP bit is set, an I/O transfer to CSR1 or CSR2 occurs, the microprogram traps to the CSR service routine.

3.8.2. INITIALIZATION ROUTINE. This routine is entered only from the switch routine upon the setting of the INIT bit. Its function is to load VLANCE with the data from the initialization block memory. The routine accesses the initialization block through the address loaded into the CDP RAM by a trap to CSR1 and CSR2 that should have occured prior to the INIT bit being set. This routine simply sequentially reads the initialization block and stores the information away in the appropriate elements of VLANCE. When done, the microcode returns to the switch routine.

3.8.3. POLLING ROUTINE. This routine is entered from :

1. The switch routine upon the setting of the STRT bit.
2. The receive routine after a packet has been received.
3. The transmit routine after a packet has been transmitted.
4. The transmit routine after a Transmission Abort occurs.
5. The memory error trap routine after the trap is serviced.

The routine begins by testing to see if the receiver is disabled, and, if not, tests the current receiver buffer ownership bit to see if it owns a buffer. If VLANCE had no acquired a buffer previously, the microprogram goes to the receiver polling routine to acquire one. When the microprogram returns from the receive polling routine, or if VLANCE had acquired a buffer previously, it tests to see if the transmitter is disabled, and if not, goes to the transmit polling routine to test if there is a buffer to be transmitted. When the microprogram returns from the transmit polling routine, the microprogram enters a timing loop, and repeats the routine upon timeout (above 1.6ms). Setting the TDMD bit in CSR0 overrides the timing loop. This forces the microprogram to fall through the wait loop. The TDMD bit is cleared immediately after leaving the wait loop. Therefore, to be effective, TDMD should be set after a buffer has been inserted on the transmit ring.

During this routine, should the receiver become active, the microprogram traps to the receive routine.

3.8.4. RECEIVE POLLING ROUTINE. This routine is entered if the receiver is enabled, and VLANCE needs a free buffer. The routine begins by the microprogram performing a memory transaction to get a buffer status word from the receive descriptor ring. After acquiring the word, it tests to see if it owns the buffer. If not, the microprogram returns to the polling routine. If it does, the microprogram proceeds to acquire two additional words to obtain the rest of the buffer address and byte count. If then returns to the polling routine with the three words stored in the CDP RAM. The trap to the receive routine is enabled in this routine.

3.8.5. RECEIVE ROUTINE. The receive routine is entered when the receiver is enabled and an incoming packet address has been detected as a match. The routine is divided into three sections of code, an initialization section, a buffer lookahead section, and a descriptor update section. In the initialization section, the microprogram first tests to see if it has acquired a free buffer. If not, it makes one attempt to get the status, address, and byte count from memory. VLANCE backs up the address and byte count in the CDP RAM for runt packet recovery, and proceeds to the lookahead section. In the lookahead section, the microprogram tries to acquire an additional buffer by memory transactions with the ring buffer descriptors. If it acquires one, it stores it in the CDP RAM, and waits for byte count overflow or the frame to terminate. In this section the receive DMA trap is enabled. The descriptor update section is entered when byte count overflow has occured of the message has ended. The code section begins with a test to determine if the message has completed, if data chaining needs to be done, or if a runt packet has been encountered. If a runt packet has been encountered, the microprogram restores the address and byte count and goes to the polling routine. If the incoming message has terminated, the microprogram writes the message length into the ring descriptor entry, writes the status information into the ring descriptor entry, puts the next buffer status information it acquired in the lookahead code in the current buffer status area of the CDP RAM, advances the ring pointer, and goes to polling. If the byte count has overflowed, but the message has not ended, chaining is required. The microprogram releases the buffer by writing the status information into the ring descriptor entry, puts the next buffer status information it acquired in the lookahead code in the current buffer status area of the CDP RAM, advances the ring pointer, and returns to the lookahead code section.

3.8.6. RECEIVE DMA ROUTINE. The Receive DMA routine in entered whenever there are 16 or more bytes of data in the SILO for transfer to memory during receive. The routine is also entered when there are less than 16 bytes in the SILO and the receiver has gone inactive. This is to allow the SILO to empty at thr end of reception. Once entered, the Receive DMA routine transfers 16 bytes of data to memory by doing 8 word transfers. these transfers are done on a single memory bus acquisition. This means that VLANCE will arbitrate through the HOLD-HOLD ACKNOWLEDGE sequence and then keep HOLD asserted for the duration of 8 transfers. The READY signal from the bus slave device controls the individual word transfers.

If the memory buffer starts on an odd address boundary, the first transfer is 1 byte rather than 1 word (2 bytes). This routine is also used to transfer less than 16 bytes at the end of a reception depending upon the packet size, buffer addresses and data chaining.

Note : DMA (direct memory access) is performed each time VLANCE initiates a memory transfer. However, in this document, DMA refers only to those transfers between the SILO and Bus memory.

This routine is entered through a microtrap in the lookahead section of the receive routine. The function of the routine is to move data out of the SILO to local memory. The trap is active when there are 16 or more bytes of data in the SILO and SILO overflow has not occurred or when the incoming message has terminated with data in the SILO. The routine pipelines the data through the Memory Data register while the address and byte counts are incremented in the control data path. A memory timeout will cause a trap. The routine is exited through the URETURN register to the code section that originally trapped to this routine.

3.8.7. TRANSMIT POLLING ROUTINE. The transmit polling routine is entered from the polling routine to determine if a message has been scheduled on the transmit descriptor ring. The routine begins by testing the status word of the ring descriptor entry. The routine tests the ownership of the ring buffer by reading the status word in the ring descriptor. If VLANCE does not own the buffer, the microprogram returns to the polling routine. If it does own the buffer, this indicates that a message is to be transmitted, and the microprogram performs memory transactions to acquire and store the address and byte count of the buffer in the CDP RAM. It then goes to the transmit routine to allow transmission of the buffer. The receive active trap is enabled during this routine to allow for processing of an incoming packet and termination of the transmit process.

3.8.8. TRANSMIT ROUTINE. The transmit routine is entered from the transmit polling routine when the microcode finds a buffer that it owns, indicating that a message is scheduled to be transmitted. The routine is divided into three sections of code : an initialization section, a buffer lookahead section, and a descriptor update section. Upon entering the initialization section, the first thing the microprogram does is back up the buffer address and byte count in the event of a retry. It then enables the DMA engine to start filling the SILO and send the preamble. It then enters a wait loop until the transmitter is actually sending the bit stream. It then proceeds to the lookahead section. In the lookahead section, the microprogram tests to determine if the current buffer it is transmitting has been marked with the end of packet flag. If so, data chaining is not required. The microprogram enters a wait loop for byte count overflow. If the end of packet flag is not set, the microprogram attempts to obtain the next buffer descriptor status, address, and byte count before entering the wait loop. When byte count overflow does occur, the microprogram enters the descriptor update section. In the descriptor update, section, the microprogram first determines if an error has occured or, simply, if data chaining must be performed. If an error needs to be reported, an error status word is written into the ring descriptor prior to writing the status word containing the "OWN" bit which releases the buffer. If no error is to be reported, the single word containing the "OWN" bit is written. The microprogram returns to the polling section if the "ENP" flag is found. Otherwise the microprogram returns to the lookahead section.

3.8.9. TRANSMIT DMA ROUTINE. This routine is entered through a microtrap in the lookahead section of the transmit routine. The function of the routine is to move data out of local memory into the SILO. The trap is active when there are 16 or more free locations in the SILO and SILO underflow has not occurred. The routine pipelines the data through the Memory Data register while the address and byte counts are incremented in the Control Data Path. A memory timeout will cause a trap. After a maximum of eight (8) words have been transferred into the SILO, the routine is exited through the URETURN register to the code section that originally trapped to this routine.

This routine is entered when a collision has been detected while the transmitter is active. The buffer address and byte counts are restored and the microprogram proceeds to the transmit routine to reschedule the transmission. This is the rule if less than 15 retransmissions have occured. If 15 retransmissions have occured, the microprogram goes instead to the error reporting code in the descriptor update section.

3.8.11 CSR TRAP ROUTINE. The CSR trap routine is entered only during the switch routine when the STOP bit of CSR0 is set. The function of the routine is to allow the access of CSR1 and CSR2 through an I/O transaction. The routine determines which CSR is being accessed, read or written, moves the data between the MDR and the CDP RAM, and generates a READY signal. The routine is exited through the URETURN register.

3.8.12 MEMORY TIMEOUT TRAP ROUTINE. This trap is invoked whenever a memory transfer times out. The routine sets the STOP bit of CSR0 and returns the microprogram to the start of the switch routine.

3.8.13. RETRY TRAP ROUTINE. This routine is entered when a collision has been detected. The buffer address pointer is restored and the SILO is cleared to restore the READ and WRITE pointers. If there was a TX error, it indicates that 15 retransmissions have occured (16 total attemps) or that the Disable Retry bit (DRTY) is set in the Mode register. The microprogram then writes the status into the transmit descriptor ring.

If there was no TX error, the byte count is restored and the microprogram returns to the start of the transmit routine to attempt another transmission.

3.8.14. DATA CHAIN. If Byte Count Equal 0 becomes true, it indicates that the receive buffer is full and the packet is not yet finished, which is the data chain case. The microprogram updates the receive status in the descriptor ring. It then checks the next OWN bit. If next OWN is false, which would be the case if there was only one buffer or if there was more than one buffer but the chip did not own the next one, the microprogram waits for RX Active to go false. This indicates that no more data is arriving from the Ethernet. When RX Active goes false, the current RX OWN bit is cleared because the ring entry has just been for the updated receive status. The SILO is then cleared to restore the READ and WRITE pointers, RX Clear is issued and the microprogram returns to the Polling routine.

If VLANCE owned the next buffer, the current receive buffer parameters in the CDP Ram are updated from the next receive buffer parameters that had previously been loaded into the CDP Ram. The microprogram then checks for the end of the ring and updates the address pointers accordingly. The microprogram then goes through the receive buffer lookahead microprogram once to try to acquire another receive buffer if one is available. The microprogram finally returns to the wait loop until either RX Done, SILO overflow or receive buffer overflow becomes true. When RX Done or SILO Overflow occurs, the microcode sets the RINT bit in CSR0. The flow from this point is the same as described elsewhere.

CHAPTER 4
ELECTRICAL SPECIFICATIONS

4. ELECTRICAL SPECIFICATIONS

This chapter provides tabular presentations for Absolute Maximum Ratings, DC Characteristics, Capacitance, and AC Timing Specifications. In addition, illustrations are provided for an Output Load Diagram and for Serial Link, Bus Master, and VLANCE Bus Slave Timing Diagrams.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
	Temperature under Bias	– 25 to + 100	°C
	Storage Temperature	– 65 to + 150	°C
	Voltage on any Pin with Respect to Ground	– .7 to + 7	V
	Power Dissipation	2.0	W

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = + 5V ± 5% unless otherwise specified.

Symbol	Parameter	Min.	Max.	Unit
V_{IL}		– 0.5	+ 0.8	V
V_{IH}		+ 2.0	V_{CC} +0.5	V
V_{OL}	@ I_{OL} + 3.2mA		+ 0.5	V
V_{OH}	@ I_{OH} = – 0.4mA	+ 2.4		V
I_{IL}	@ V_{in} = 0.4 to V_{CC}		± 10	µA

CAPACITANCE F = 1MHz

Symbol	Parameter	Min.	Max.	Unit
C_{IN}			10	pF
C_{OUT}			10	pF
C_{IO}			20	pF

AC TIMING SPECIFICATIONS T_A = 0°C to 70°C, V_{CC} = + 5V ± 5%, unless otherwise specified.

n°	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
1	SCLK	T_{SCT}	SLCK Period		99		101	ns
2	SCLK	T_{SCL}	SLCK Low Time		45		55	ns
3	SCLK	T_{SCH}	SCLK High Time		45		55	ns
4	SLCK	T_{SCR}	Rise Time of SCLK		0		8	ns
5	SLCK	T_{SCF}	Fall Time of SCLK		0		8	ns
6	TENA	T_{SEP}	TENA Propagation Delay after the Rising Edge of SCLK	C_L = 50pF			95	ns
7	TENA	T_{SEH}	TENA Hold Time after the Rising Edge of SCLK	C_L = 50pF	25			ns

AC TIMING SPECIFICATIONS (continued)

n°	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
8	TX	T_{SDP}	TX Data Propagation Delay after the Rising Edge of TCLK	C_L = 50pF			95	ns
8A	TX	T_{DTT}	TX Transition - Transition (see note 2)	C_L = 50pF	$B_t - 1$			ns
9	TX	T_{SDH}	TX Data Hold Time after the Rising Edge of SCLK	C_L = 50pF	25			ns
10	RCLK	T_{RCT}	RCLK Period		85		118	ns
11	RCLK	T_{RCH}	RCLK High Time		38			ns
12	RCLK	T_{RCL}	RCLK Low Time		38			ns
13	RCLK	T_{RCR}	Rise Time of RCLK		0		8	ns
14	RCLK	T_{RCF}	Fall Time of RCLK		0		8	ns
15	RX	T_{RDR}	RX Data Rise Time		0		8	ns
16	RX	T_{RDF}	RX Data Fall Time		0		8	ns
17	RX	T_{RDH}	RX Data Hold Time (RCLK to RX data change)		5			ns
18	RX (see note)	T_{RDS}	RX Data Setup Time (RX data stable to the rising edge of RCLK)		60			ns
19	RENA	T_{DPL}	RENA Low Time		120			ns
20	RENA	T_{RENH}	RENA Hold Time after Rising Edge of RCLK		40			ns
21	CLSN	T_{CPH}	CLSN High Time		80			ns
22	A/DAL	T_{COFF}	Bus Master Driver Disable after Rising Edge of $\overline{\text{HOLD}}$		0		50	ns
23	A/DAL	T_{DON}	Bus Master Driver Enable after Falling Edge of $\overline{\text{HLDA}}$		0		150	ns
24	$\overline{\text{HLDA}}$	T_{HHA}	Delay to Falling Edge of $\overline{\text{HLDA}}$ from Falling Edge of $\overline{\text{HOLD}}$ (bus master)		0			ns
25	$\overline{\text{RESET}}$	T_{RW}	RESET Pulse Width Low		200			ns
26	A/DAL	T_{CYCLE}	Read/write, Address/data Cycle Time		600			ns
27	A	T_{XAS}	Address Setup Time to the Falling Edge of ALE		75			ns
28	A	T_{XAH}	Address Hold Time after the Rising Edge of $\overline{\text{DAS}}$		15			ns
29	DAL	T_{AS}	Address Setup Time to the Falling Edge of ALE		75			ns
30	DAL	T_{AH}	Address Hold Time after the Falling Edge of ALE		35			ns
31	DAL	T_{RDAS}	Data Setup Time to the Rising Edge of $\overline{\text{DAS}}$ (bus master read)		50			ns
32	DAL	T_{RDAH}	Data Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus master read)		0			ns
33	DAL	T_{DDAS}	Data Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus master write)		0			ns
34	DAL	T_{WDS}	Data Setup Time to the Rising Edge of $\overline{\text{DAS}}$ (bus master write)		200			ns
35	DAL	T_{WDH}	Data Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave read)		35			ns

Note : T_{RDS}(min) = T_{RCT} 25ns i.e T_{RCT} = 100ns, then T_{RDS}(min) = 75ns.

AC TIMING SPECIFICATIONS (continued)

n°	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
36	DAL	T_{SDO1}	Data Driver Delay after the Falling Edge of $\overline{DAS}$ (bus slave read)	(CRS 0,3, RAP)		400		ns
37	DAL	T_{SDO2}	Data Driver Delay after the Falling Edge of $\overline{DAS}$ (bus slave read)	(CRS 1,2)		1200		ns
38	DAL	T_{SRDH}	Data Hold Time after the Rising Edge of $\overline{DAS}$ (bus slave write)		0		35	ns
39	DAL	T_{SWDH}	Data Setup Time to the Falling Edge of $\overline{DAS}$ (bus slave write)		0			ns
40	DAL	T_{SWDS}	Data Setup Time to the Falling Edge of $\overline{DAS}$ (bus slave write)		0			ns
41	ALE	T_{ALEW}	ALE Width High		120		150	ns
42	ALE	T_{DALE}	Delay from Rising Edge of $\overline{DAS}$ to the Rising Edge of ALE		70			ns
43	$\overline{DAS}$	T_{DSW}	$\overline{DAS}$ Width Low		200			ns
44	$\overline{DAS}$	T_{ADAS}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{DAS}$		80		130	ns
45	$\overline{DAS}$	T_{RIDF}	Delay from the Rising Edge of $\overline{DALO}$ to the Falling Edge of $\overline{DAS}$ (bus master read)		15			ns
46	$\overline{DAS}$	T_{RDYS}	Delay from the Falling Edge of $\overline{READY}$ to the Rising Edge of $\overline{DAS}$	Taryd=300ns	100		250	ns
47	$\overline{DALI}$	T_{ROIF}	Delay from the Rising Edge of $\overline{DALO}$ to the Falling Edge of $\overline{DALI}$ (bus master read)		15			ns
48	$\overline{DALI}$	T_{RIS}	$\overline{DALI}$ Setup Time to the Rising Edge of $\overline{DAS}$ (bus master read)		135			ns
49	$\overline{DALI}$	T_{RIH}	$\overline{DALI}$ Hold Time after the Rising Edge of $\overline{DAS}$ (bus master read)		0			ns
50	$\overline{DALI}$	T_{RIOF}	Delay from the Rising Edge of $\overline{DALI}$ to the Falling Edge of $\overline{DALO}$ (bus master read)		55			ns
51	$\overline{DALO}$	T_{OS}	$\overline{DALO}$ Setup Time to the Falling Edge of ALE (bus master read)		110			ns
52	$\overline{DALO}$	T_{ROH}	$\overline{DALO}$ Hold Time after the Falling Edge of ALE (bus master read)		35			ns
53	$\overline{DALO}$	T_{WDSI}	Delay from the Rising Edge of $\overline{DAS}$ to the Rising Edge of $\overline{DALO}$ (bus master write)		35			ns
54	$\overline{CS}$	T_{CSH}	$\overline{CS}$ Hold Time after the Rising Edge of $\overline{DAS}$ (bus slave)		0			ns
55	$\overline{CS}$	T_{CSS}	$\overline{CS}$ Setup Time to Falling Edge of $\overline{DAS}$ (bus slave)		0			ns
56	ADR	T_{SAH}	ADR Hold Time after the Rising Edge of $\overline{DAS}$ (bus slave)		0			ns
57	ADR	T_{SAS}	ADR Setup Time to the Falling Edge of $\overline{DAS}$ (bus slave)		0			ns
58	$\overline{READY}$	T_{ARYD}	Delay from the Falling Edge of ALE to the Falling Edge of READY to insure a Minimum Bus Cycle Time (600ns)				80	ns

AC TIMING SPECIFICATIONS (continued)

n°	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
59	$\overline{\text{READY}}$	T_{SRDS}	Data Setup Time to the Falling Edge of $\overline{\text{READY}}$ (bus slave read)		75			ns
60	$\overline{\text{READY}}$	T_{RDYH}	$\overline{\text{READY}}$ Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus master)		0			ns
61	$\overline{\text{READY}}$	T_{SRO1}	$\overline{\text{READY}}$ Driver Turn on after the Falling Edge of $\overline{\text{DAS}}$ (bus slave)	(CSR 0,3, RAP)		600		ns
62	$\overline{\text{READY}}$	T_{SRO2}	$\overline{\text{READY}}$ Driver turn on after the Falling Edge of $\overline{\text{DAS}}$ (bus slave)	(CSR 1,2)		1400		ns
63	$\overline{\text{READY}}$	T_{SRYH}	READY Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave)		0		35	ns
64	READ	T_{SRH}	$\overline{\text{READ}}$ Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave)		0			ns
65	READ	T_{SRS}	$\overline{\text{READ}}$ Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus slave)		0			ns

Figure 14 : Output Load Diagram.

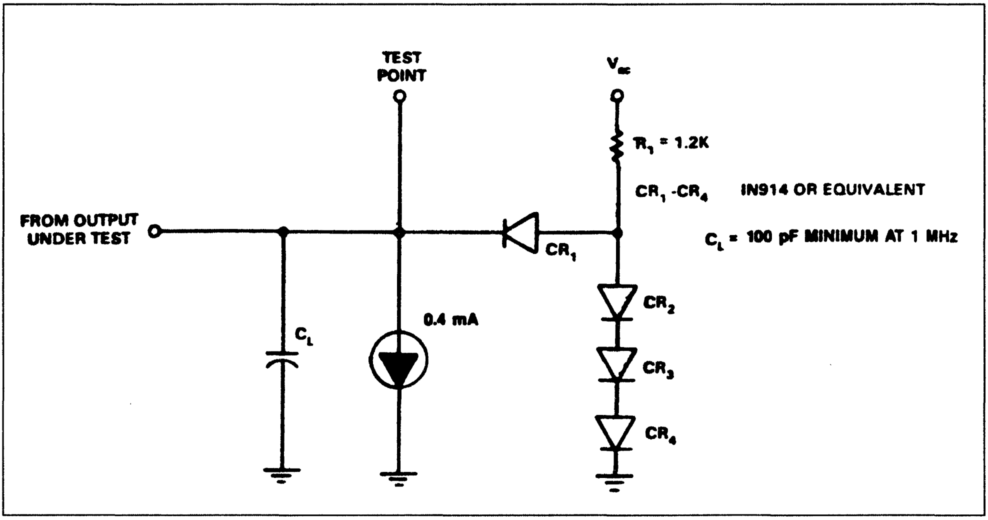
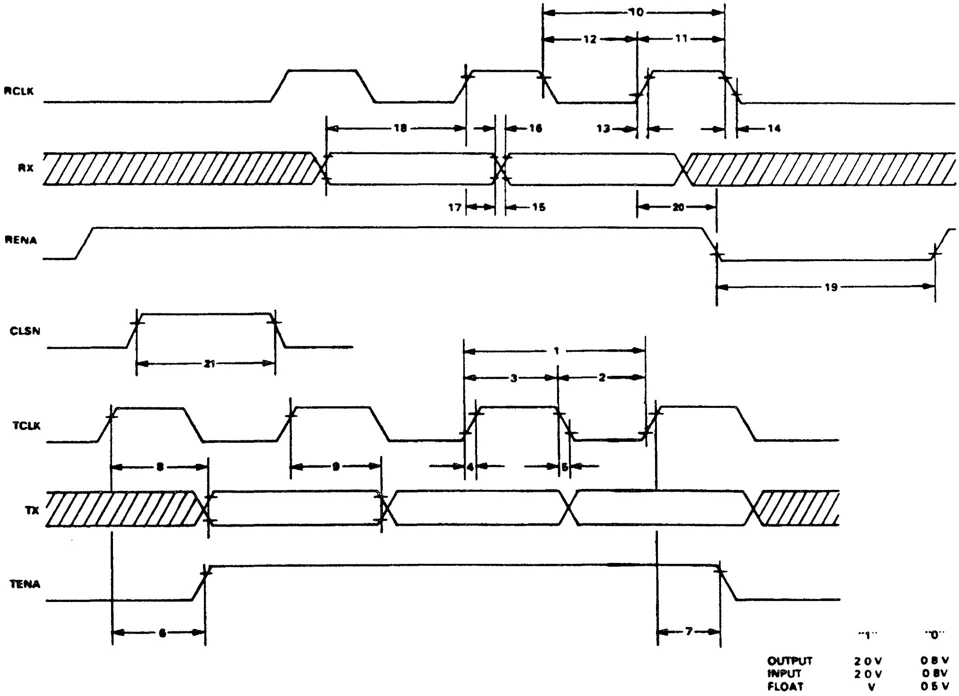
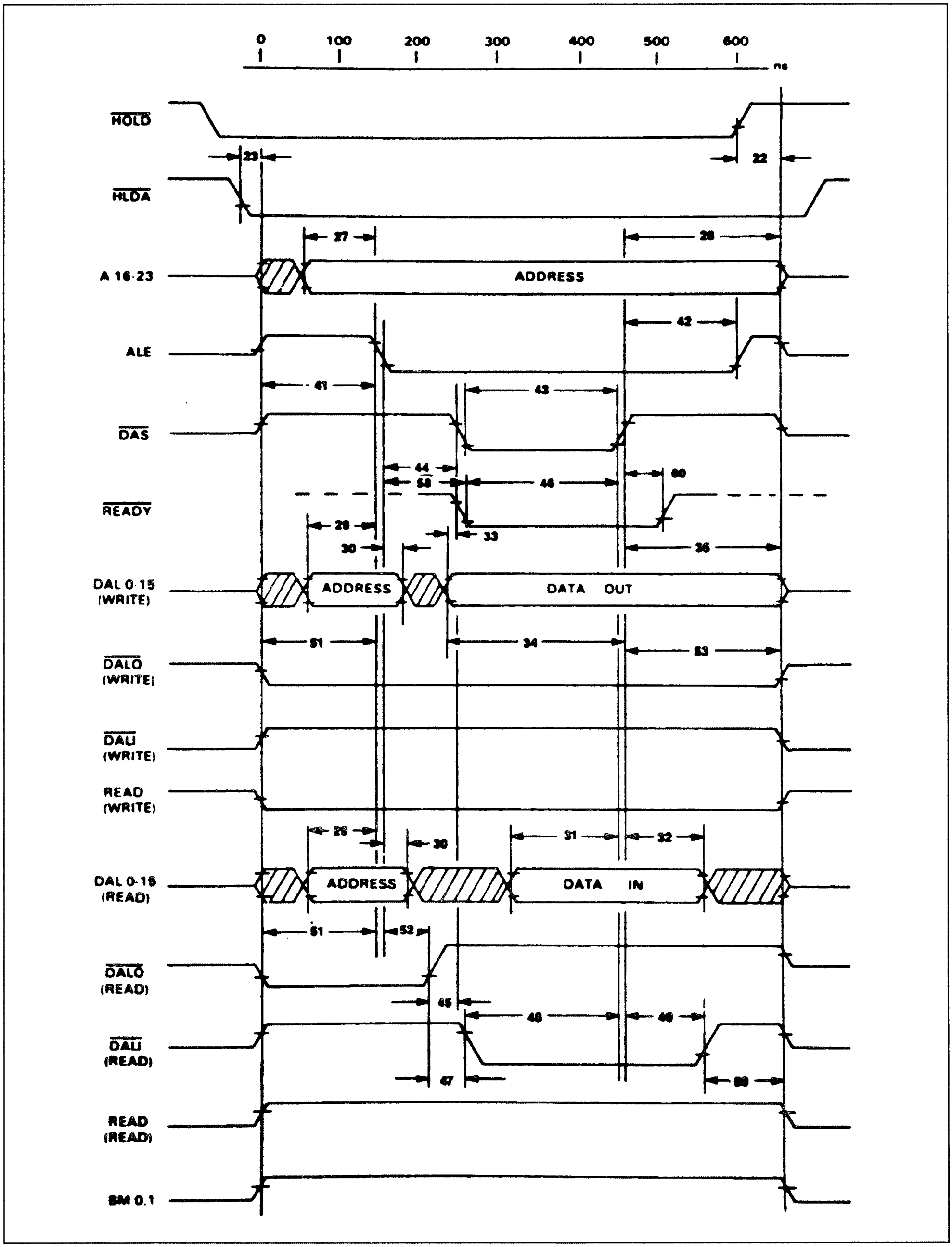


Figure 15 : Serial Link Timing Diagram - SIA Interface Signals.



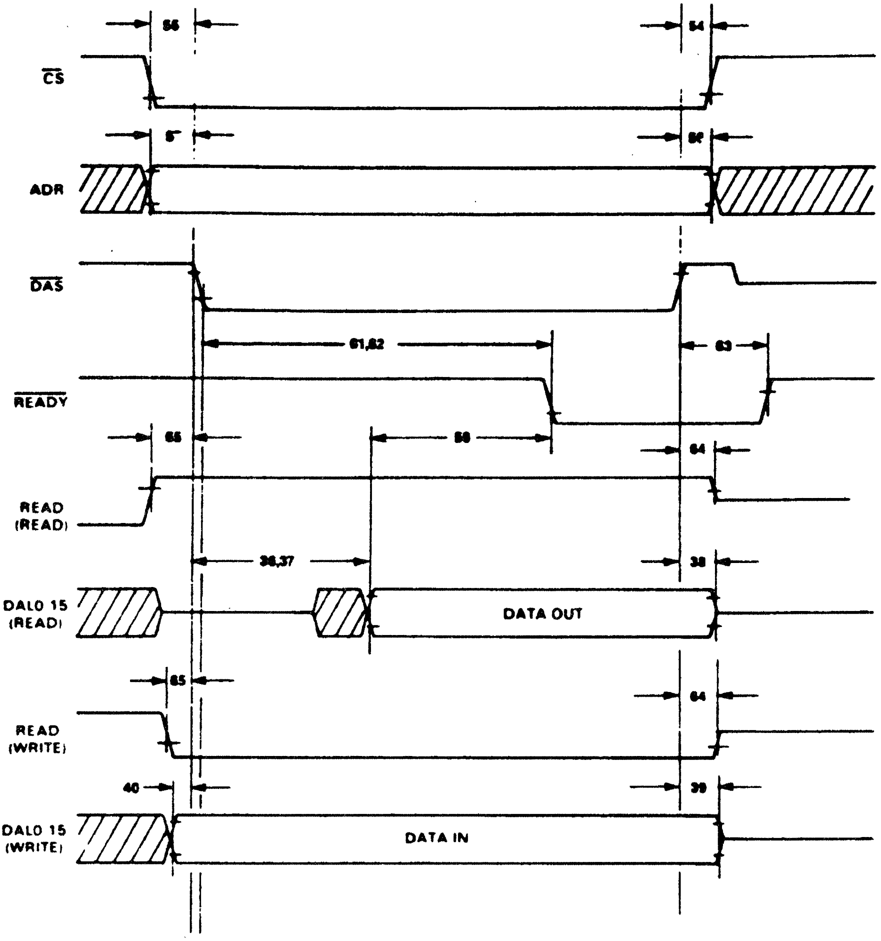
Timing measurements are mode at the following voltages,unless otherwise specified.

Figure 16 : Bus Master Timing Diagram.



Note : The Bus Master cycle time will increase from a minimum of 600ns in 100ns steps until the slave device returns $\overline{\text{READY}}$.

Figure 17 : VLANCE Bus Slave Timing Diagram.



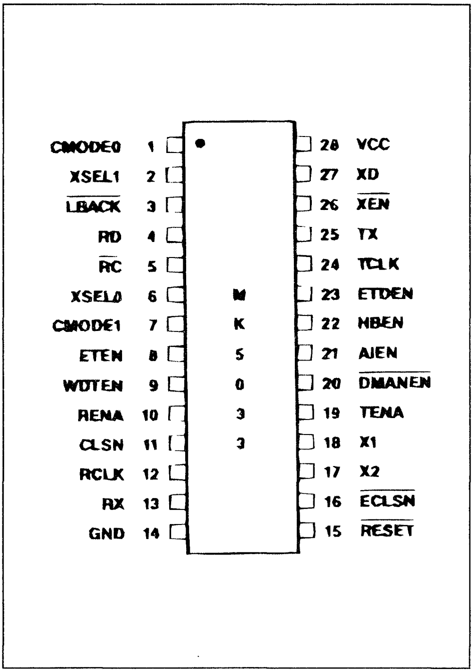
SGS-THOMSON
MICROELECTRONICS

MK5033

MANCHESTER ENCODER DECODER

- CONFORMS TO STARLAN SPECIFICATIONS
- AUTO COMPENSATION FOR LINE REVERSAL
- COMPATIBLE WITH MOST ETHERNET CONTROLLER CHIPS
- DATA RATES DC TO 2.66Mbps SUPPORTED
- MANCHESTER OR DIFFERENTIAL MANCHESTER DATA ENCODING/DECODING
- FULL DUPLEX OR HALF DUPLEX OPERATION
- SUPPORTS STAR, BUS, OR POINT-TO-POINT NETWORK TOPOLOGIES
- COLLISION DETECTION CIRCUITRY WITH THE FOLLOWING FEATURES :
  - detects missing mid-bit transitions
  - detects too close together
  - transitions too far apart
  - external collision input pin
  - carrier dropout
  - watchdog timer
  - AT&T Release 1 collision presence signal
  - echo timeout
- OPTIONAL END-OF-FRAME DETECTION
  - input protection at end-of-frame
- LOOPBACK CAPABILITY
- RECEIVE CARRIER AUTOMATICALLY CONVERTED TO A LEVEL SIGNAL
- OPTIONAL WATCHDOG TIMER TO PREVENT CONTINUOUS TRANSMISSION
- OPTIONAL ECHO TIMER TO SIGNAL ERROR IF TRANSMITTED FRAME IS NOT RECEIVED
- OPTIONAL HEARTBEAT GENERATION
- IN 82586/82588 MODE, INSENSITIVE TO EXTRA BITS AHEAD OF PREAMBLE
- DIGITAL PHASE-LOCKED LOOP
- ON CHIP CRYSTAL OSCILLATOR, 16, 10, 8, OR 6X OPERATION
- CMOS TECHNOLOGY
- 28-PIN DIP
- SINGLE 5-VOLT SUPPLY
- ALL INPUTS AND OUTPUTS TTL COMPATIBLE*

Figure 1 : MK5033 Pin Assignment.

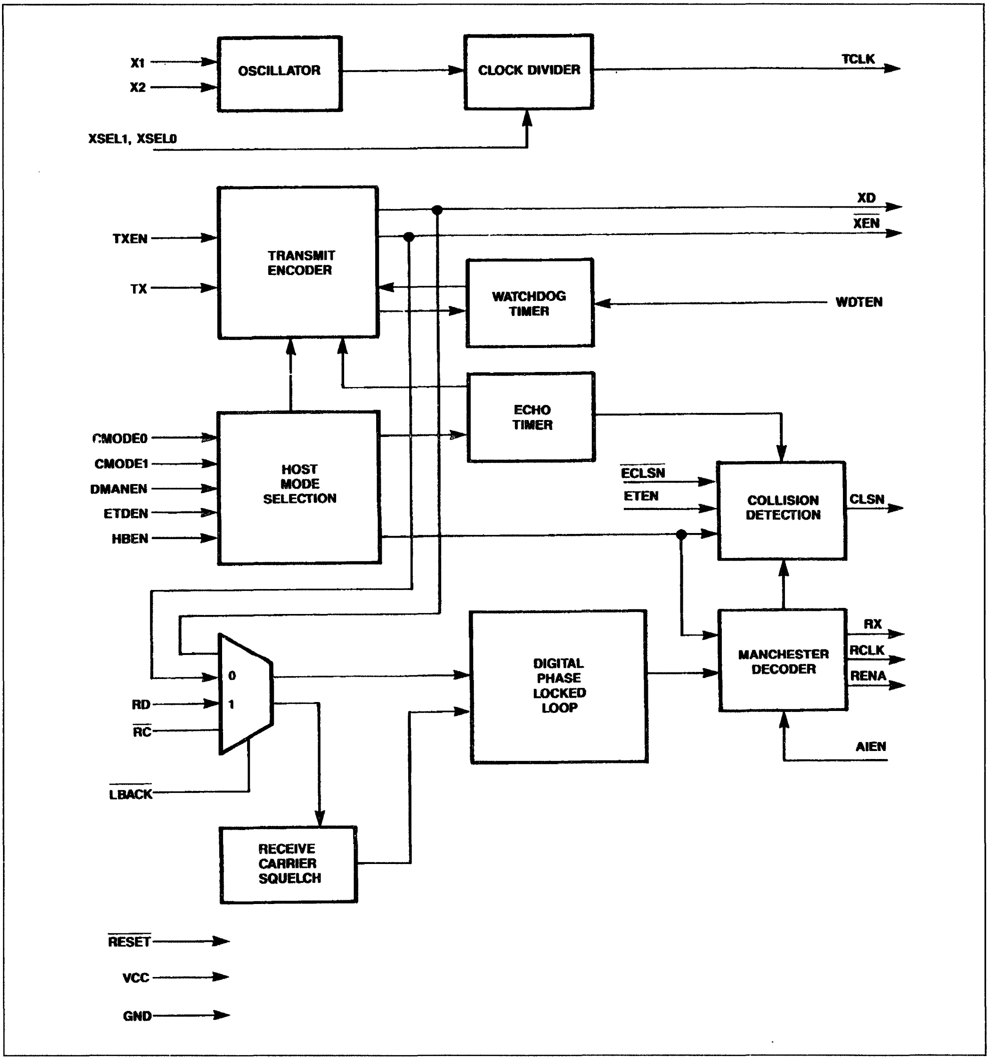


DESCRIPTION

The MK5033 is a general purpose Manchester Encoder/Decoder. It incorporates several features that make it an ideal StarLAN station chip. The MK5033 performs three functions. It encodes data from a controller chip into Manchester or Differential data. It decodes Manchester or Differential Manchester data from the line transceiver and produces NRZ data and clock for the controller chip. It also detects collision and signals the controller chip that a collision has occurred.

* Crystal inputs have CMOS thresholds.

Figure 2 : Manchester Encoder/Decoder Block Diagram.



PIN DESCRIPTION

CONTROLLER INTERFACE

RX Output RX is the serial receive data after decoding.

RENA Output This signal indicates that data is available to the controller on the RX output.

RCLK Output This is the receive data clock recovered from incoming data on the RD pin.

TX Input This is the serial data to be transmitted. It is clocked into the chip by TCLK.

TENA Input This signal indicates that data is valid on the TX input. It goes active with the first bit of transmission.

TCLK Output This is the transmit data clock. All transmit interface signals are synchronized to this clock. This clock is always active.

CLSN Output This signal is asserted when a Manchester violation is detected on the RD line or when the external collision input (ECLSN) goes active. It is also asserted if either of the timers expire. It is deasserted when line idle is detected on the RD line and TENA goes inactive.

TRANSCEIVER INTERFACE

XD Output Transmit data output.

XEN Output Transmit output enable. This signal goes low to signal XD active. It goes high at the end of transmission. NOTE : If ETDEN is active XD will remain high for 2-bit times at the end of a frame and XEN will remain low during this time.

RD Input Receive data input.

RC Input Receive carrier input. Receive carrier can be either a pulse stream or an active low signal to indicate carrier active. The chip contains internal squelch circuitry, as shown in figure 3, to convert a pulse signal to a level signal.

ECLSN Input External collision input. When this pin is held low for at least 20nS an external collision is signaled.

MISCELLANEOUS

CMODE1,
CMODE0 Inputs These two mode bits allow the chip to be used with a variety of controller chips.

CMODE1 = 1, CMODE0 = 0 (10) 82586/82588 (see note)

Transmit data (TX) is sampled on the rising edge of TCLK.

Receive data (RX) transitions on the rising edge of RCLK.

TENA - active low

RENA - active low - goes active when phase locked loop is locked

CLSN - active low

CMODE1 = 1 CMODE0 = 1 (11) SGS-THOMSON LANCE MK68590

Transmit data (TX) is sampled on the falling edge of TCLK.

Receive data (RX) transitions on the falling edge of RCLK.

TENA - active high

RENA - active high - goes active when receive carrier goes active

CLSN - active high

CMODE1 = 0 CMODE0 = 0 (00) TEST Mode

this mode is only useful for production testing.

Note : Compatibility with controller chips based on preliminary controller data sheets.

CMODE1 = 0 CMODE0 = 1 (01) SGS-THOMSON Variable Bit Rate LANCE MK5032 (see note)

TENA - active high

RENA - active high - goes active when receive carrier goes active

CLSN - active high

DMANEN Input When this pin is low the chip encodes and decodes serial data using Differential Manchester. When this pin is high it uses Manchester.

HBEN Input When this pin is high the chip will signal CLSN after TENA is deasserted at the end of transmission. CLSN remains active until link idle is received on RD.

ETDEN Input When this pin is high the chip will recognize end-of-frame as specified in the StarLAN specification. It will also ignore incoming data on RD for 20 data bits after the end of a received frame.

AIEN Input Auto Inversion Enable. If both frame recognition is enabled and Manchester is selected, then if AIEN is high the frame polarity is sensed and corrected if necessary. If AIEN is low, ETDEN is low, or DMANEN is low, then auto compensation for line reversal is disabled.

ETEN Input When ETEN is high the echo timer is activated. The echo timer starts at the beginning of a transmitted frame. If a receive carrier is not received with 510 TCLKS, then collision will be asserted.

WDTEN Input When WDTEN is high the watchdog timer is activated. The timer starts when TENA is asserted. If TENA goes inactive before the timer expires, the timer is reset. If the timer expires, transmission is aborted and collision asserted.

LBACK Input When this pin is low the chip will be put into internal loopback. The transmit data will be internally looped back into the input RD. The outputs XD and XEN will be held idle during loopback.

RESET Input When this pin is low the chip is in reset mode. All interface signals will be inhibited except TCLK. RESET must remain active for at least three TCLK periods.

XSEL0,
XSEL1 Inputs These inputs select which frequency clock or crystal is to be connected to X1 and X2.

XSEL1	XSEL0	CLOCK DIVIDOR
0	0	16X
0	1	8X
1	0	10X
1	1	6X

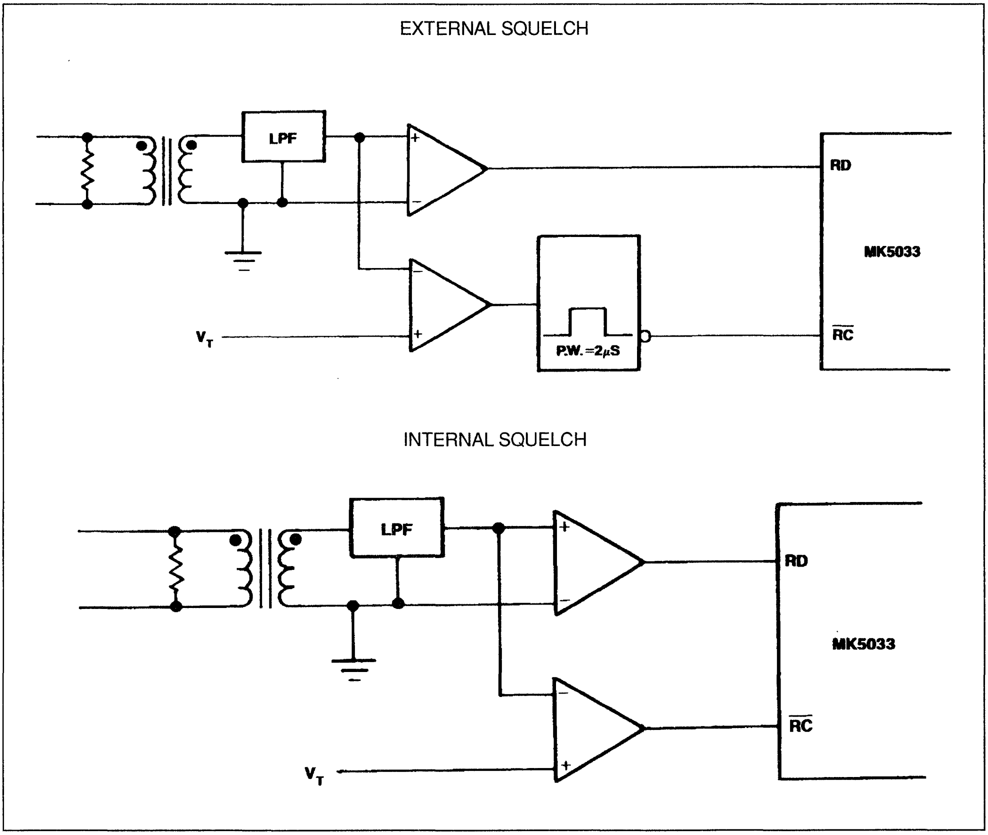
X1, X2 Inputs Crystal oscillator inputs. A crystal can be connected between these inputs, or a TTL level square wave can be connected to X1 while X2 is left unconnected.

VCC Input + 5V ± 10%

GND

Note : Compatibility with controller chips based on preliminary controller data sheets.

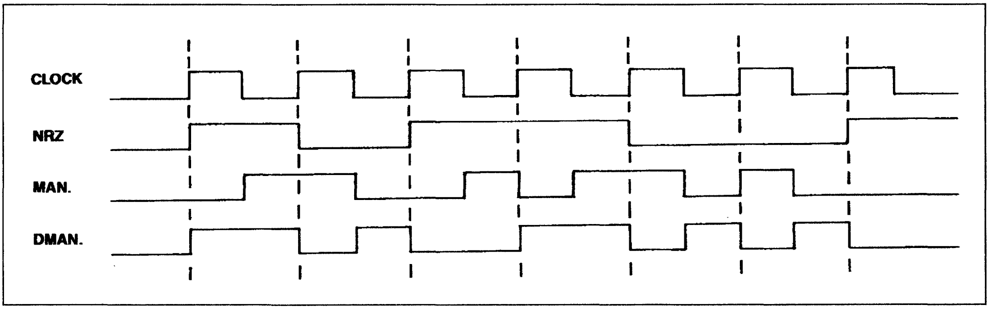
Figure 3 : Internal Versus External Squelch.



CIRCUIT DESCRIPTION

TRANSMITTER

The transmitter encodes NRZ data from the controller chip into Manchester or Differential Manchester Space data. The diagram below shows the two encoding schemes.



Data encoding and transmission begin when the controller chip brings TENA active. The start of data encoding is delayed by two bits when in 82586/82588 mode. TX data is sampled using TCLK as the clock. Data is encoded into Manchester or Differential Manchester Space, as shown above, depending upon the state of DMANEN. The encoded data is output on XD. XEN goes low with the first bit of data output on XD. The transmit delay, delay from TENA active to XEN active, is less than 1.5 TCLKs. The controller chip signals end of data by bringing TENA inactive. The pin ETDEN controls how the MK5033 handles the end transmission.

If ETDEN is high the MK5033 will add a delimiter to the data stream after the last bit is transmitted. In Manchester mode TX will be held high for 1.5 TCLKs if the last data bit is a one and for 2 TCLKs if the last data bit is a zero. During this time XEN is held active. In Differential Manchester, TX is toggled after the last data bit and held in that state for 2 TCLKs. XEN is active during this time. After the delimiter has been sent, XEN is brought inactive.

If ETDEN is low the MK5033 will not add any delimiter to the data stream. XEN will be brought inactive after the last data bit has been output.

RECEIVER

The receiver consists of four major sections :

1) Receive carrier squelch

2) Internal loopback

3) Digital phase locked loop

4) Manchester/Differential Manchester decoder

Depending on the state of DMANEN, the receiver decodes Manchester or Differential Manchester space data from pin RD into NRZ form. It also extracts timing (RCLK) from the data. The NRZ data is output to the controller on pin RX.

RECEIVE CARRIER SQUELCH

The Receive carrier pin has internal squelch logic that allows the signal to be either a level signal or a pulse train. Receive carrier is active low. The receive carrier must be present for 3 clock samples to be considered a valid carrier. Once the carrier is considered valid then it must be active for only one clock sample time every two bit times to remain valid. (see figure 3).

LOOPBACK

When loopback is enabled (LBACK low) RD and RC are ignored. Transmit data is internally looped back as receive data. The transmitter outputs XD and XEN are disabled during loopback.

DIGITAL PHASE LOCKED LOOP (DPLL)

The digital phase locked loop is implemented with a counter that clears on each transition of the receive data. The phase locked loop will declare "lock" after receiving data that has two "long transitions". A long transition occurs when the receive data does not change for at least 4/6 of a bit time in 6X mode (5/8 in 8X, 7/10 in 10X, and 11/16 in 16X). The phase locked loop generates a clock frequency that is 2 times the data rate.

AUTOMATIC COMPENSATION FOR WIRING REVERSAL

When installing twisted pair telephone wiring, it is often difficult and expensive to maintain proper polarity on the wire pairs. The MK5033 will automatically compensate for this reversal. If Manchester coding is selected with both ETDEN = 1 and AIEN = 1 then any frame that is received with inverse polarity will be detected and correct polarity established prior to data decoding.

MANCHESTER/DIFFERENTIAL MANCHESTER DECODER

The receive data (after inversion if enabled) is fed into the decoder along with the recovered 2x clock from the DPLL. The decoder changes the receive data to NRZ data. The NRZ data is output on RX. RENA signals the controller chip that data is available. (see mode pin descriptions). RCLK is a 1x clock output that is synchronous with the data on RX.

PROTECTION TIME

After the end of a received frame the receiver is disabled for 20 bit times. This protection time guarantees immunity to spikes caused by transformer coupling after the end of frame.

COLLISION

CLSN is an output to the controller chip that indicates a possible problem with the data. There are several sources of collision.

1) Transitions too close together
   Collision is signaled if the receive data stream transitions a second time in less than 2/6 bit times in 6X mode (3/8 in 8X, 3/10 in 10X, and 5/16 in 16X).

2) Transitions too far apart
   Collision is signaled if the receive data stream does not transition again within 9/6 in 6X mode (10/8 in 8X, 12/10 in 10X, and 20/16 in 16X).

3) Manchester violation
   If the data violates Manchester or Differential Manchester coding rules, depending on DMANEN, then collision is signaled.

4) Watchdog timer
   If the watchdog timer expires, then collision will be signaled, if WDTEN is high.

5) Echo timer
   If the echo timer expires, then collision will be signaled, if ETEN is high.

6) External collision
   If the external collision pin ($\overline{\text{ECLSN}}$) goes low for at least 20ns, then collision will be signaled.

7) Receive carrier lost during transmission
   If the MK5033 is transmitting and the receive carrier goes active and then inactive before it is through transmitting, then collision is signaled.

8) Heartbeat
   Heartbeat is enabled when HBEN is high. If it is enabled, then collision will be signaled 8 TCLKs after TENA goes away, and collision will remain active for at least 8 TCLKs.

Once CLSN is activated it remains active until both TENA and RENA go inactive. The exception to this is heartbeat. If heartbeat signals collision, then collision is guaranteed to remain for 8 TCLKs or until TENA or RENA go inactive, whichever is longer.

WATCHDOG TIMER

When enabled, the watchdog timer ensures that the MK5033 will not transmit for more than 101K bit times. The timer is enabled by bringing WDTEN high and disabled by bringing WDTEN low. If WDTEN is high, the timer is activated when TXEN goes active. The timer resets when TXEN goes inactive. If TXEN remains active for more than 101K bit times, then the timer will time out causing collision to be asserted and $\overline{\text{XEN}}$ to go inactive. If loopback is enabled, watchdog timeout will occur after 325 bit times. This permits run time testing of the watchdog timer.

ECHO TIMER

When the echo timer is enabled the MK5033 expects the data that it is transmitting to be received on RD within 510 bit times. The echo timer is activated when TXEN goes active. If 510 bit times elapse before RENA goes active, then the timer will time out causing collision to be activated.

Oscillator

The MK5033 will accept two forms of clock input : a CMOS input or a crystal. If pin X2 is left unconnected, a 6.0/8.0/10.0/16.0MHz ± 0.01% CMOS clock may be applied to pin X1. Alternately, a crystal circuit may be connected between X1 and X2 to form the basis of an oscillator. Typically, a 6.0/8.0/10.0/16.0 ± 0.005% parallel resonant crystal is needed to insure the ± 0.01% frequency accuracy required for StarLAN. Refer to figure 4. A fundamental mode, parallel resonance type crystal should be used with the manufacturer's suggested load capacitance.

Reset Input

The reset pin is an active low Schmidt trigger input. A simple RC network may be used to insure correct operation upon power-up. Refer to fig. 5.

Figure 4 : Oscillator Operation.

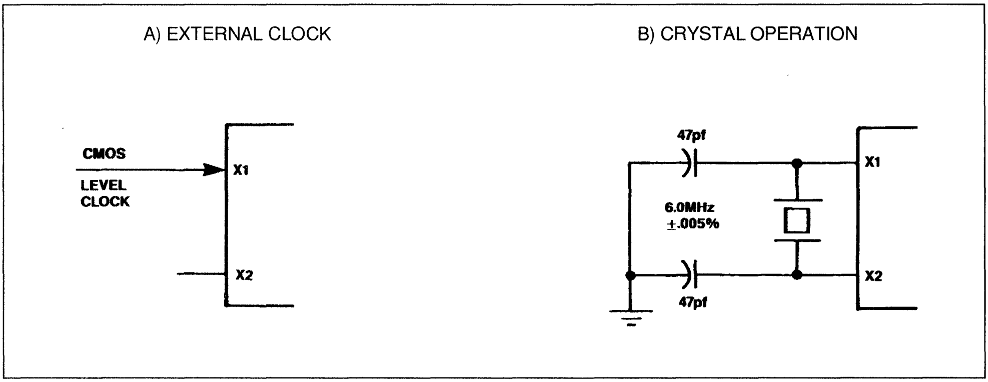
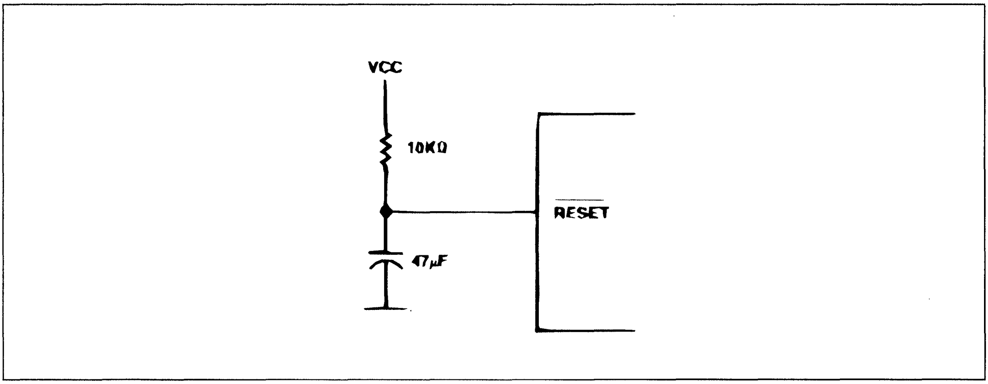


Figure 5 : Typical RC Connection for Power-On Reset.



ELECTRICAL SPECIFICATIONS

This chapter provides tabular presentations for Absolute Maximum Ratings, DC Characteristics, Capacitance and AC Timing Specifications. In addition, illustrations are provided for an Output Load Diagram (figure 9) and Station Timing Diagrams.

ABSOLUTE MAXIMUM RATINGS

	Parameter	Value	Unit
	Temperature under Bias	– 25 to + 100	°C
	Storage Temperature	– 65 to + 150	°C
	Voltage on any Pin with Respect to Ground	– 0.5 to V_{CC} + 0.5	V
	Power Dissipation (no load)	50	mW

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

CAPACITANCE : F = 1MHz

Symbol	Test Conditions	Min.	Max.	Unit
C_{IN}			10	pf
C_{OUT}			10	pf
C_{IO}			20	pf

AC TIMING SPECIFICATIONS

T_A = 0°C to 70°C, V_{CC} = + 5V, ± 5% unless otherwise specified, V_{TH} = 2.0V,V_{TL} = 0.8V

#	Signal	Symbol	Parameter	Min. ns	Typ. ns	Max. ns
1	X1	T_{X1T}	X1 Period	62		
2	X1	T_{X1L}	X1 Low Time	24		
3	X1	T_{X1H}	X1 High Time	24		
4	X1	T_{X1R}	Rise Time of X1	0		8
5	X1	T_{X1F}	Fall Time of X1	0		8
6	$\overline{XEN}$	$T_{\overline{XEN}}$	$\overline{XEN}$ Delay from X1		40	65
7	XD	T_{XD}	XD Delay from X1		40	65
8	XD	J_{XD}	Transmit Jitter $\mid T_{XD} \uparrow - T_{XD} \downarrow \mid \div 2$		4	6
9	TCLK	T_{CLK}	TCLK Delay from X1			70
10	TX	T_{TXST1}	TX Setup to Falling Edge of TCLK, CMODE = 1	90		
11	TX	T_{TXHT1}	TX Hold from Falling Edge of TCLK, CMODE = 1	15		
12	TX	T_{TXS}	TX Setup to X1	15		
13	TX	T_{TXH}	TX Hold from X1	15		
14	TENA	T_{TNAST1}	TENA Setup to Falling Edge of TCLK, CMODE = 1	90		
15	TENA	T_{TNAST1}	TENA Hold from Falling Edge of TCLK, CMODE = 1	15		
16	TENA	T_{TENAS}	TENA Setup to X1	15		
17	TENA	T_{TENAH}	TENA Hold from X1	15		
18	TX	T_{TXST0}	TX Setup to Rising Edge of TCLK, CMODE = 0	90		
19	TX	T_{TXHT0}	TX Hold from Rising Edge of TCLK, CMODE = 0	15		
20	TX	T_{TXS}	TX Setup to X1, CMODE = 0	15		
21	TX	T_{TXH}	TX Hold from X1, CMODE = 0	15		
22	TENA	T_{TNAST0}	TENA Setup to Positive Edge of TCLK, CMODE = 0	90		
23	TENA	T_{TNAHT0}	TENA Hold from Positive Edge of TCLK, CMODE = 0	15		
24	TENA	T_{TNAS}	TENA Setup to X1, CMODE = 0	15		
25	TENA	T_{TNAH}	TENA Hold from X1, CMODE = 0	15		

AC TIMING SPECIFICATIONS (continued)

#	Signal	Symbol	Parameter	Min. ns	Typ. ns	Max. ns
26	CLSN	T_{CLSN}	CLSN Delay from X1			70
27	$\overline{ECLSN}$	$T_{\overline{ECLSN}}$	Minimum Detected Pulse Width		5	20
28	$\overline{RC}$	$T_{\overline{RC}S}$	$\overline{RC}$ Setup to X1	15		
29	$\overline{RC}$	$T_{\overline{RC}H}$	$\overline{RC}$ Hold from X1	15		
30	RD	T_{RDS}	RD Setup to X1	15		
31	RD	T_{RDH}	RD Hold from X1	15		
32	RD	J_{RD6}	RD Incoming Jitter Tolerance, 6X Mode, X1 = 6MHz, $T_{X1T} - \mid T_{RDS} \uparrow - T_{RDS} \downarrow \mid$		165	161
33	RD	J_{RD8}	RD Incoming Jitter Tolerance, 8X Mode, X1 = 8MHz, $T_{X1T} - \mid T_{RDS} \uparrow - T_{RDS} \downarrow \mid$		123	119
34	RD	J_{RD10}	RD Incoming Jitter Tolerance, 10X Mode, X1 = 10MHz, $T_{X1T} - \mid T_{RDS} \uparrow - T_{RDS} \downarrow \mid$		198	194
35	RD	J_{RD16}	RD Incoming Jitter Tolerance, 16X Mode, X1 = 16MHz, $T_{X1T} - \mid T_{RDS} \uparrow - T_{RDS} \downarrow \mid$		186	182
36	RCLK	T_{RCLK}	RCLK Delay from X1, CMODE = 1		40	65
37	TX	T_{RXRCK1}	RX Delay from Falling RCLK, CMODE = 1	– 30		30
38	RX	T_{RX}	RX Delay from X1, CMODE = 1		40	65
39	RENA	$T_{RNARCK1}$	RENA Delay from Falling RCLK, CMODE = 1	– 30		30
40	RENA	T_{RENA}	RENA Delay from X1, CMODE = 1		45	65
41	CLSN	$T_{CSNRCK1}$	CLSN Delay from Falling Edge RCLK, CMODE = 1	– 30		30
42	CLSN	T_{CLSN}	CLSN Delay from X1, CMODE = 1			70
43	RCLK	T_{RCLK0}	RCLK Delay from X1, CMODE = 0		40	65
44	RCLK	P_{RCLK}	RCLK Pulse Width, CMODE = 0	$T_{X1T}-20$		$T_{X1T}+20$
45	RCLK	T_{RXCLK}	RCLK Delay from RX Stable, CMODE = 0	$T_{X1T}-20$		
46	RX	T_{CLKRX}	RX Hold from Falling Edge of RCLK, CMODE = 0	$2 * T_{X1T}-20$		
47	RCLK	T_{RNACLK}	RCLK Delay from RENA Stable, CMODE = 0	$T_{X1T}-20$		
48	RENA	T_{CLKRNA}	RENA Hold from Falling Edge of RCLK, CMODE = 0	$2 * T_{X1T}-20$		
49	RCLK	T_{CSNCLK}	Rising RCLK Delay from CLSN Stable, CMODE = 0	$2 * T_{X1T}-20$		
50	CLSN	T_{CLKCSN}	CLSN Delay from Rising Edge of RCLK, CMODE = 0	$T_{X1T}-20$		

Figure 6 : External X1 Timing Diagram.

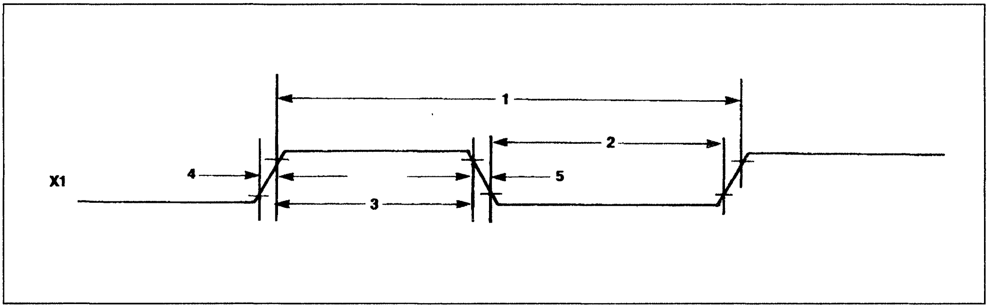


Figure 7 : Transmit Timing Diagram.

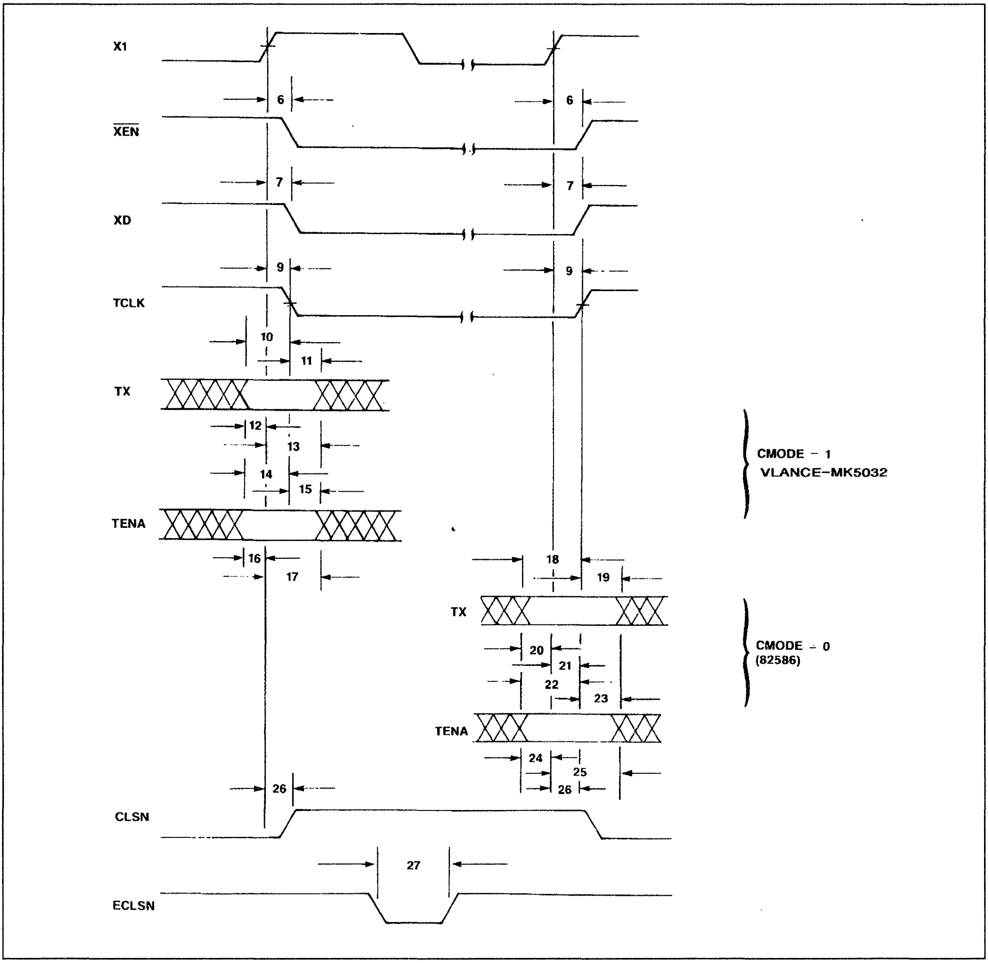


Figure 8 : Receiver Timing Diagram.

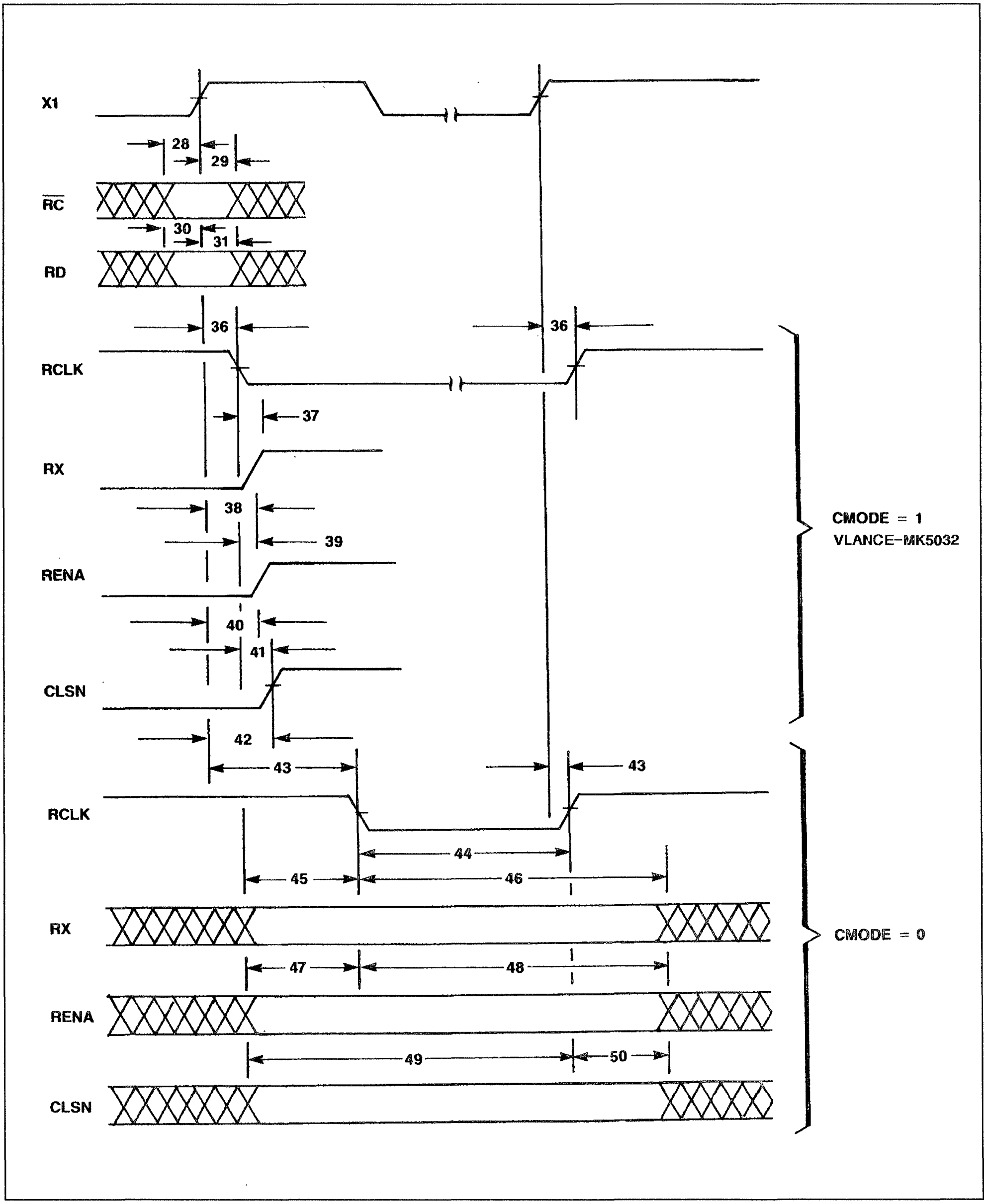
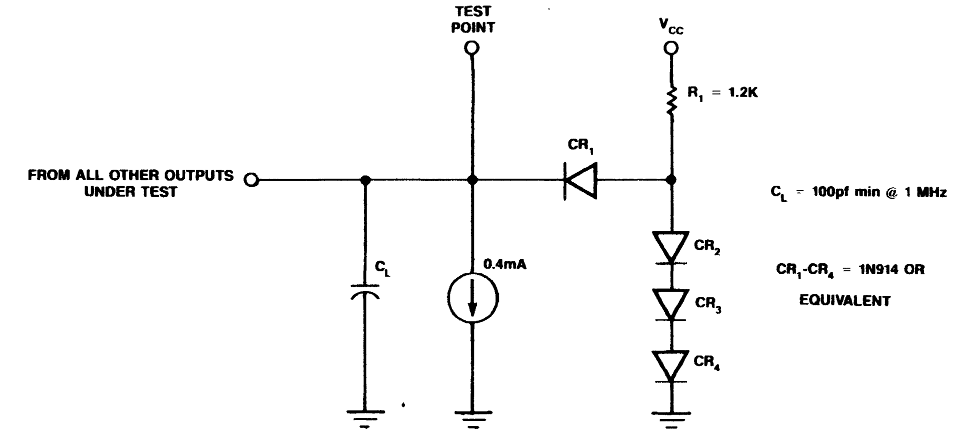
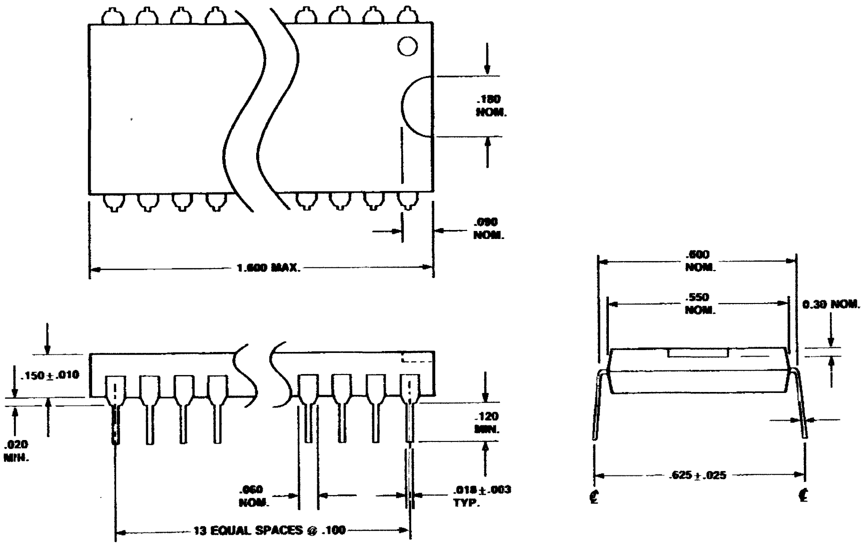


Figure 9 : Output Load Diagram.


TEST POINT
V_{CC}
R_1 = 1.2K
CR_1
FROM ALL OTHER OUTPUTS UNDER TEST
C_L = 100pf min @ 1 MHz
CR_2
C_L
0.4mA
CR_1-CR_4 = 1N914 OR EQUIVALENT
CR_3
CR_4

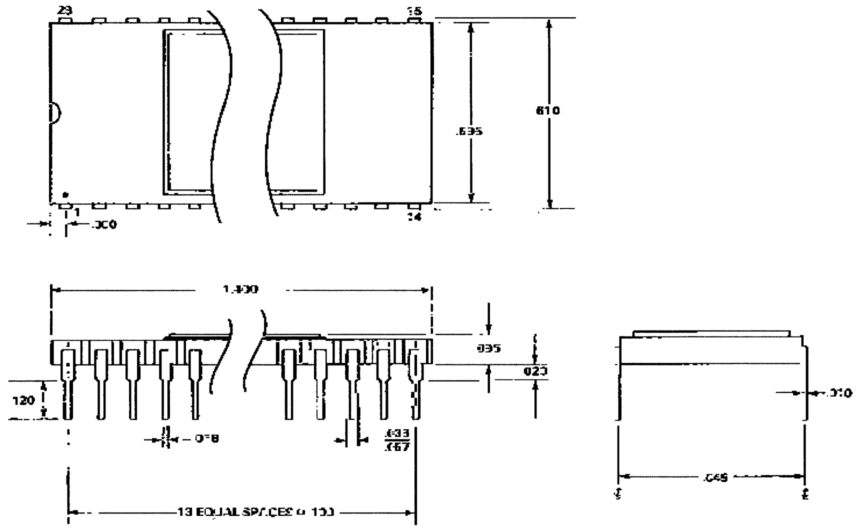
PACKAGE MECHANICAL DATA

28-Pin Plastic Dual-In-Line (N) - MK5033N



PACKAGE MECHANICAL DATA (continued)

28-Pin Ceramic - MK5033P



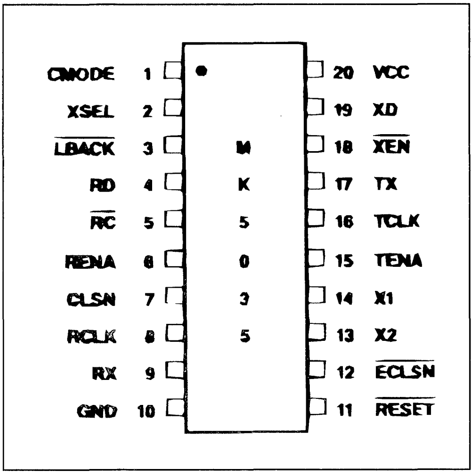
SGS-THOMSON
MICROELECTRONICS

MK5035N

StarLAN ENCODER DECODER

- CONFORMS WITH STARLAN SPECIFICATION
- SUPPORTS MULTI-POINT EXTENSION
- AUTO COMPENSATION FOR LINE REVERSAL
- COMPATIBLE WITH MK5032 VLANCE AND INTEL 82586/82588
- CLOSE PIN COMPATIBILITY WITH SEEQ8023
- DATA RATES TO 2.66Mbps SUPPORTED
- MANCHESTER DATA ENCODING/DECODING
- COLLISION DETECTION CIRCUITRY WITH THE FOLLOWING FEATURES :
  - detects missing mid-bit transitions
  - transitions too close together
  - transitions too far apart
  - external collision input pin
  - carrier dropout
  - watchdog timer
  - AT&T release 1 collision presence signal
  - echo timeout
- RECEIVE END-OF-FRAME DETECTION
  - input protection at end-of-frame
- LOOPBACK CAPABILITY
- RECEIVE CARRIER AUTOMATICALLY CONVERTED TO A LEVEL SIGNAL
- ECHO TIMER TO SIGNAL ERROR IF TRANSMITTED FRAME IS NOT RECEIVED
- HEARTBEAT GENERATION
- IN 82586 MODE, INSENSITIVE TO EXTRA BITS AHEAD OF PREAMBLE
- DIGITAL PHASE-LOCKED LOOP
- ON CHIP CRYSTAL OSCILLATOR, 8X OR 6X OPERATION
- CMOS TECHNOLOGY
- 20-PIN DIP
- SINGLE 5-VOLT SUPPLY
- ALL INPUTS AND OUTPUTS TTL COMPATIBLE*

Figure 1 : MK5035 Pin Assignment.



GENERAL DESCRIPTION

The MK5035 is a Manchester Encoder/Decoder chip incorporating several features that make it an ideal StarLAN station chip. The MK5035 performs three functions. It encodes data from a controller chip into Manchester data. It decodes Manchester data from the line transceiver and produces NRZ data and clock for the controller chip. It also detects collisions and signals the controller chip that a collision has occurred.

The MK5035 has several enhancements for StarLAN and Multi-Point extension (MPE) StarLAN. These include auto compensation for wiring reversal, echo timer, external collision detect, watchdog timer, and heartbeat, among others.

* Crystal inputs have CMOS thresholds.

PIN DESCRIPTION

CONTROLLER INTERFACE

RX Output RX is the serial receive data after decoding.

RENA Output This signal indicates that data is available to the controller on the RX output.

RCLK Output RCLK is the receive data clock recovered from the incoming data RD.

TX Input TX is the serial data to be transmitted. It is clocked into the chip by TCLK.

TENA Input This signal indicates that data is valid on the TX input. It goes active with the first bit of transmission.

TCLK Output TCLK is the transmit data clock. All transmit interface signals are synchronized to this clock.

CLSN Output This signal is asserted when a manchester violation is detected on the RD line or when the external collision input ($\overline{\text{ECLSN}}$) goes active.

TRANSCEIVER INTERFACE

XD Output Encoded transmit data output.

$\overline{\text{XEN}}$ Output Transmit output enable. This signal goes low to indicate XD active. It goes high at the end of transmission.

RD Input Encoded receive data input.

$\overline{\text{RC}}$ Input Receive carrier input. Receive carrier can be either a pulse stream or an active low signal to indicate carrier active. The chip contains internal squelch circuitry, as shown in figure 3, to convert a pulse signal to a level signal.

$\overline{\text{ECLSN}}$ Input External collision input. When this pin is held low for at least 20nS, an external collision is signaled.

OTHER PINS

CMODE Input This input allows the part to be used with either Mostek or Intel controllers :

CMODE = 0, 82586/82588 (see note)

Transmit data (TX) is sampled on the rising edge of TCLK. Receive data (RX) transitions on the rising edge of RCLK.

TENA - active low

RENA - active low - goes active when phase lock loop is locked.

CLSN - active low

CMODE = 1, SGS-THOMSON VLANCE MK5032

Transmit data (TX) is sampled on the falling edge of TCLK. Receive data (RX) transitions on the falling edge of RCLK.

TENA - active high

RENA - active high - goes active when phase lock loop is locked.

CLSN - active high

$\overline{\text{LBACK}}$ Input When this input is low the part will be put into internal loopback. The transmit data will be internally looped back as receive data. See figure 2. The outputs XD and $\overline{\text{XEN}}$ will be held inactive during loopback.

$\overline{\text{RESET}}$ Input When this pin is low the chip is in reset mode. All interface signals will be inhibited except TCLK. $\overline{\text{RESET}}$ should remain active for three TCLK periods.

XSEL Input This input selects the clock divider.

If XSEL = 0, it is 8X.

If XSEL = 1, it is 6X.

X1, X2 Inputs Crystal oscillator inputs. A crystal can be connected between these inputs, or a CMOS level square wave can be connected to X1 while X2 is left unconnected.

VCC Input + 5V ± 5%

GND

Note : Compatible with controller chips based on preliminary controller data sheets.

Figure 2 : StarLAN Encoder Decoder. MK5035.

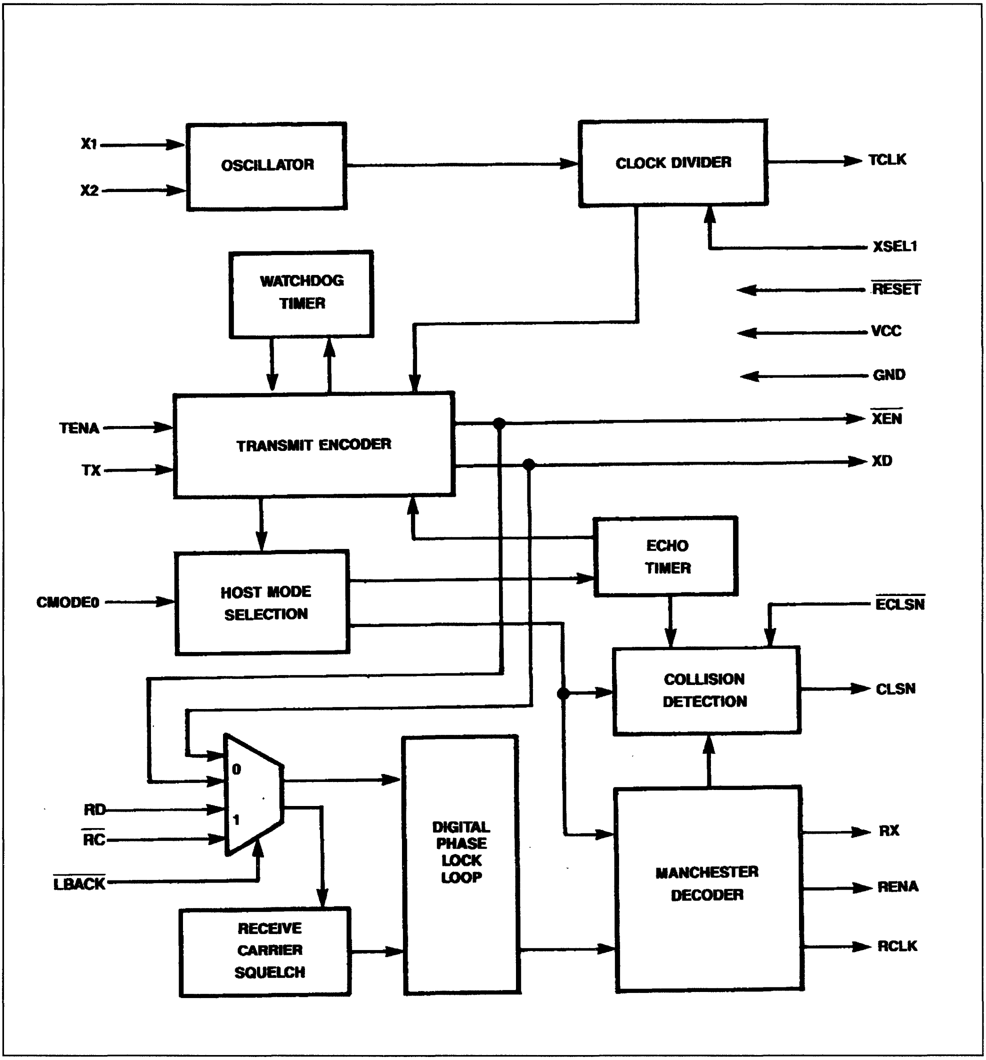
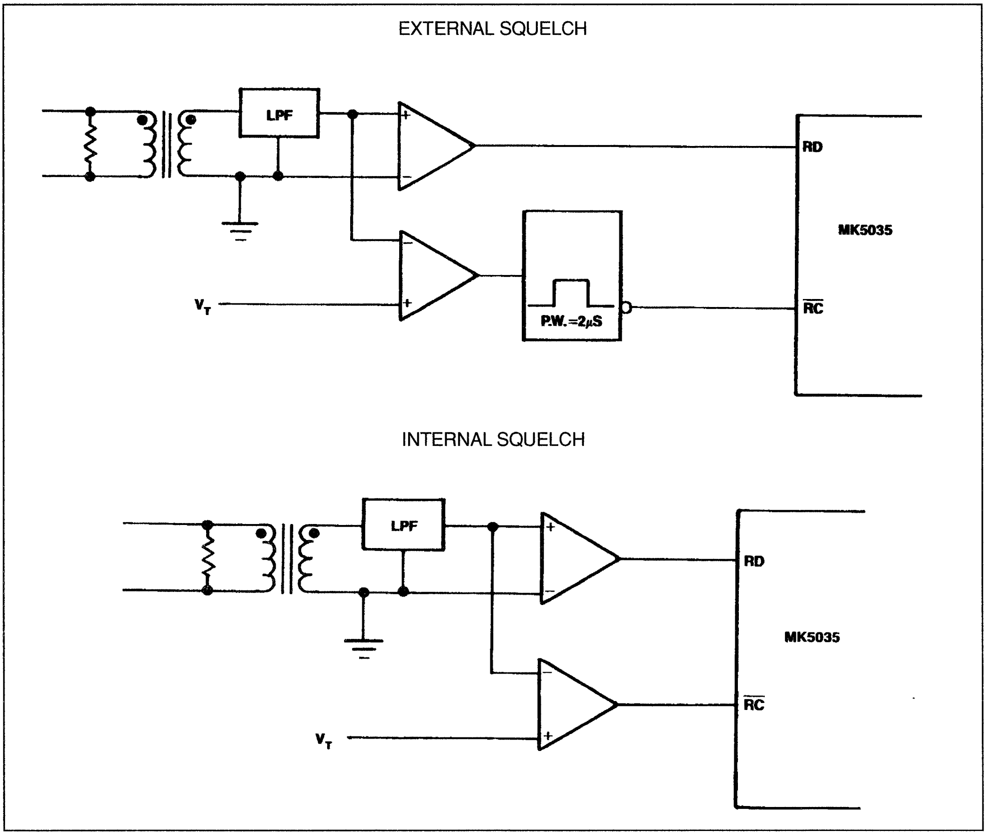


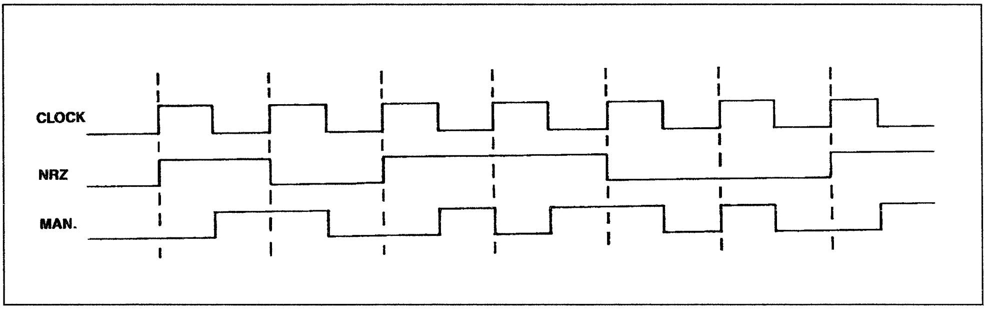
Figure 3 : Internal Versus External Time Squelch.



CIRCUIT DESCRIPTION

TRANSMITTER

The transmitter encodes NRZ data from the controller chip into the Manchester data. The diagram below shows the two encoding schemes.



Data encoding and transmission begin when the controller chip brings TENA active. The start of data encoding is delayed by two bits when in 82586/82588 mode. Tx is sampled using TCLK as the clock. The encoded data is output on XD. XEN goes low with the first bit of data output on XD. The transmit delay, delay from TENA active to XEN active, is less than 1.5 TCLKs. The controller chip signals end of data by bringing TENA inactive.

XD will be held high for an additional 1.5 TCLKs if the last data bit is a one, and for 2 TCLKs if the last data bit is a zero. During this time XEN is held active.

RECEIVER

The receiver consists of four major sections.

1) Receive carrier squelch
2) Internal loopback
3) Digital phase locked loop
4) Manchester decoder

The receiver takes Manchester data in on RD, when receive carrier (RC) is active, and decodes the data into NRZ data and also produces clock (RCLK) from the data. The NRZ data is output to the controller on RX.

RECEIVE CARRIER SQUELCH

The Receive carrier pin has internal squelch logic that allows the signal to be either a level signal or a pulse train. Receive carrier is active low. The receive carrier must be present for 3 clock samples to be considered a valid carrier. Once the carrier is considered valid then it must be active for only one clock sample time every two bit times to remain valid. (see figure 3).

Automatic Compensation For Wiring Reversal

When installing twisted pair telephone wiring, it is often difficult and expensive to maintain proper polarity on the wire pairs. The MK5035 will automatically compensate for this reversal. Any frame that is received with inverse polarity will be detected and decoded with the correct polarity.

LOOPBACK

When loopback is enabled (LBACK low), RD and RC are ignored. Transmit data is internally looped back as receive data. The transmitter outputs XD and XEN are disabled during loopback.

DIGITAL PHASE LOCKED LOOP (DPLL)

The digital phase locked loop is implemented with a counter that clears on each transition of the receive data. The phase locked loop will declare "lock" after receiving data that has two "long transitions". A long transition occurs when the receive data does not change for at least 4/6 (5/8 in 8X mode) of a bit time.

MANCHESTER DECODER

The receive data (after inversion if needed) is fed into the decoder along with the recovered 2X clock from the DPLL. The decoder changes the receive data to NRZ data. The NRZ data is output on TX. RENA signals the controller chip that data is available. (see mode pin descriptions). RCLK is a 1X clock output that is synchronous with the data on RX.

PROTECTION TIME

After the end of a received frame the receiver is disabled for 20 bit times. This protection time guarantees immunity to spikes caused by transformer coupling after the end of frame.

COLLISION

CLSN is an output to the controller chip that indicates a possible problem with the data. There are several sources of collision.

1) Transitions too close together
   Collision is signaled if the receive data stream transitions a second time in less than 2/6 (3/8 in 8X mode) bit times.
2) Transitions too far apart
   Collision is signaled if the receive data stream does not transition again within 9/6 (10/8 in 8X mode) bit times.
3) Manchester violation
   If the data violates Manchester coding rules, then collision is signaled.
   A Manchester violation is a missing mid-bit transition.
4) Watchdog timer
   If the watchdog timer expires, then collision will be signaled.

5) Echo timer
If the echo timer expires without receive carrier going active, then collision will be signaled.

6) External collision
If the external collision pin (ECLSN) goes low for at least 20ns, then collision will be signaled.

7) Receive carrier lost during transmission
If the MK5035 is transmitting and the receive carrier goes active and then inactive before it is through transmitting, then collision is signaled.

8) Heartbeat

Collision, as a result of heartbeat, will be signaled 8 TCLKs after TENA goes away, and collision will remain active for at least 8 TCLKs.

Once CLSN is activated it remains active until both TENA and RENA go inactive. The exception to this is heartbeat. If heartbeat signals collision, then collision is guaranteed to remain for 8 TCLKs.

WATCHDOG TIMER

The watchdog timer ensures that the MK5035 will not transmit for more than 101K bit times. The timer is started when TXEN goes active. The timer resets when TXEN goes inactive. If TXEN remains active for more than 101K bit times, then the timer will time-out causing collision to be asserted and XEN to go inactive. If loopback is enabled, watchdog timeout will occur after 325 bit times. This particular timer value allow StarLAN HUBs to activate their own jabber functions, thereby alerting net management.

ECHO TIMER

When the echo timer is enabled the MK5035 expects the data that it is transmitting to be received on RC/RD within 510 bit times. The echo timer is activated when TXEN goes active. If 510 bit times elapse before RENA goes active, then the timer will time out causing collision to be activated.

Oscillator

The MK5035 will accept two forms of clock input : a CMOS input or a crystal. If pin X2 is left unconnected, a 6.0/8.0MHz ± 0.01% CMOS clock may be applied to pin X1. Alternately, a crystal circuit may be connected between X1 and X2 to form the basis of an oscillator. Typically, a 6.0/8.0 ± 0.005% parallel resonant crystal is needed to insure the ± 0.01% frequency accuracy required for StarLAN. Refer to figure 4. A fundamental mode, parallel resonance type crystal should be used with the manufacturer's suggested load capacitance.

Reset Input

The reset pin is an active low Schmidt trigger input. A simple RC network may be used to insure correct operation upon power-up. Refer to fig. 5.

Figure 4 : Oscillator Operation.

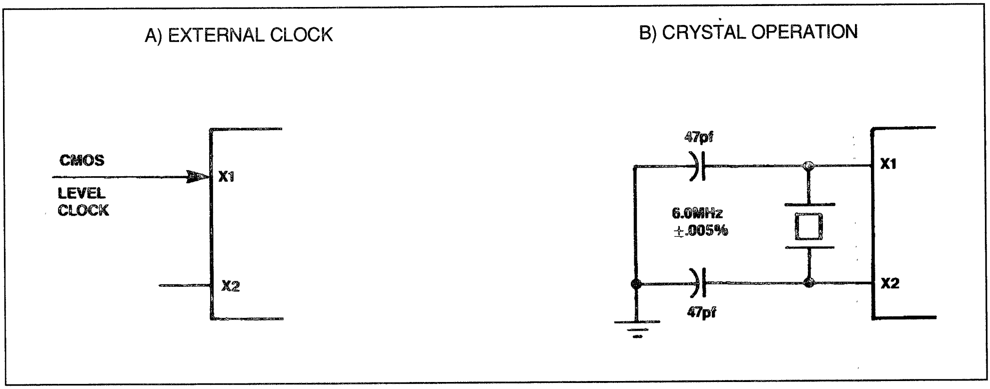
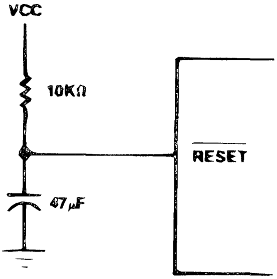


Figure 5 : Typical RC Connection for Power-On Reset.



ELECTRICAL SPECIFICATIONS

This chapter provides tabular presentations for Absolute Maximum Ratings, DC Characteristics, Capacitance and AC Timing Specifications. In addition, illustrations are provided for an Output Load Diagram (figure 9) and Station Timing Diagrams.

ABSOLUTE MAXIMUM RATINGS

Symbol	Paramter	Value	Unit
T_A	Temperature Under Bias	−25 to +100	°C
T_{stg}	Storage Temperature	−65 to +150	°C
V_I	Voltage on any Pin with Respect to Ground	−0.5 to V_{CC} +0.5	V
P_D	Power Dissipation (no load)	50	mW

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = + 5V ± 5% unless otherwise specified)

Symbol	Test conditions	Min.	Max.	Units
V_{IL}		− 0.5	+ 0.8	V
V_{IH}	Except X1	+ 2.0	V_{CC} + 0.5	V
V_{IH}	X1	+ 3.5	V_{CC} + 0.5	V
V_{OL}	@ I_{OL} = 3.2mA, except X2		+ 0.5	V
V_{OH}	@ I_{OH} = − 0.4mA, except X2	+ 2.4		V
V_{OH}	@ I_{OH} = − 40μA, except X2	+ 3.2		V
I_{IL}	@ V_{IN} = 0.4 to V_{CC}		± 10	μA
I_{CC}			8	mA

CAPACITANCE : F = 1MHz

Symbol	Conditions	Min.	Max.	Units
C_{IN}			10	pf
C_{OUT}			10	pf
C_{IO}			20	pf

AC TIMING SPECIFICATIONS

(T_A = 0°C to 70°C, V_{CC} = + 5V ± 5% unless otherwise specified, V_{TH} = 2.0V, V_{TL} = 0.8V)

#	Signal	Symbol	Parameter	Min. ns	Typ. ns	Max. ns
1	X1	T_{X1T}	X1 period	62		
2	X1	T_{X1L}	X1 low time	24		
3	X1	T_{X1H}	X1 high time	24		
4	X1	T_{X1R}	Rise time of X1	0		8
5	X1	T_{X1F}	Fall time of X1	0		8
6	$\overline{XEN}$	$T_{\overline{XEN}}$	$\overline{XEN}$ delay from X1		40	65
7	XD	T_{XD}	XD delay from X1		40	65
8	XD	J_{XD}	Transmit jitter $\lvert T_{XD}\uparrow - T_{XD}\downarrow \rvert \div 2$		4	6
9	TCLK	T_{CLK}	TCLK delay from X1			70
10	TX	T_{TXST1}	TX setup to falling edge of TCLK, CMODE = 1	90		
11	TX	T_{TXHT1}	TX hold from falling edge of TCLK, CMODE = 1	15		
12	TX	T_{TXS}	TX setup to X1	15		
13	TX	T_{TXH}	TX hold from X1	15		
14	TENA	T_{TNAST1}	TENA setup to falling edge of TCLK, CMODE = 1	90		
15	TENA	T_{TNAST1}	TENA hold from falling edge of TCLK, CMODE = 1	15		
16	TENA	T_{TENAS}	TENA setup to X1	15		
17	TENA	T_{TENAH}	TENA hold from X1	15		
18	TX	T_{TXSTO}	TX setup to rising edge of TCLK, CMODE = 0	90		
19	TX	T_{TXSTO}	TX hold to rising edge of TCLK, CMODE = 0	15		
20	TX	T_{TXS}	TX setup to X1, CMODE = 0	15		
21	TX	T_{TXH}	TX hold from X1, CMODE = 1	15		
22	TENA	T_{TNASTO}	TENA setup to positive edge of TCLK, CMODE = 0	90		
23	TENA	T_{TNAHTO}	TENA hold to positive edge of TCLK, CMODE = 0	15		
24	TENA	T_{TNAS}	TENA setup to X1, CMODE = 0	15		
25	TENA	T_{TNAH}	TENA hold from X1, cmode = 0	15		
26	CLSN	T_{CLSN}	CLSN delay from X1			70
27	$\overline{ECLSN}$	$T_{\overline{ECLSN}}$	Minimum detected pulse width		5	20
28	$\overline{RC}$	$T_{\overline{RC}S}$	$\overline{RC}$ setup to X1	15		
29	$\overline{RC}$	$T_{\overline{RC}H}$	$\overline{RC}$ hold from X1	15		

AC TIMING SPECIFICATIONS (continued)

#	Signal	Symbol	Parameter	Min. ns	Typ. ns	Max. ns
30	RD	T_{RCH}	RD setup to X1	15		
31	RD	T_{RDS}	RD hold from X1	15		
32	RD	J_{RD6}	RD Incoming Jitter Tolerance, 6X mode, X1 = 6MHz, $T_{X1T} - \lvert T_{RDS} \uparrow - T_{RDS} \downarrow \rvert$		165	161
33	RD	J_{RD8}	RD Incoming Jitter Tolerance, 8X mode, X1 = 8MHz, $T_{X1T} - \lvert T_{RDS} \uparrow - T_{RDS} \downarrow \rvert$		123	119
34	RCLK	T_{RCLK}	RCLK delay from X1, CMODE = 1		40	65
35	TX	T_{RXRCK1}	RX delay from falling RCLK, CMODE = 1	– 30		30
36	RX	T_{RX}	RX delay from X1, CMODE =1		40	65
37	RENA	$T_{RNARCK1}$	RENA delay from falling RCLK, CMODE =1	– 30		30
38	RENA	T_{RENA}	RENA delay from X1, CMODE =1		45	65
39	CLSN	$T_{CSNRCK1}$	CLSN delay from falling edge RCLK, CMODE = 1	– 30		30
40	CLSN	T_{CLSN}	CLSN delay from X1, CMODE = 1			70
41	RCLK	T_{RCLK0}	RCLK delay from X1, CMODE = 0		40	65
42	RCLK	P_{RCLK}	RCLK pulse width, CMODE = 0	T_{X1T}– 20		T_{X1T}+20
43	RCLK	T_{RXCLK}	RCLK delay from RX stable, CMODE = 0	T_{X1T}– 20		
44	RX	T_{CLKRX}	RX hold from falling edge of RCLK, CMODE = 0	2*T_{X1T}–20		
45	RCLK	T_{RNACLK}	RCLK delay from RENA stable, CMODE = 0	T_{X1T}– 20		
46	RENA	T_{CLKRNA}	RENA hold from falling edge of RCLK, CMODE = 0	2*T_{X1T}–20		
47	RCLK	T_{CSNCLK}	Rising RCLK delay from CLSN stable, CMODE = 0	2*T_{X1T}–20		
48	CLSN	T_{CLKCSN}	CLSN delay from rising edge of RCLK, CMODE = 0	T_{X1T}–20		

Figure 6 : External X1 Timing Diagram.

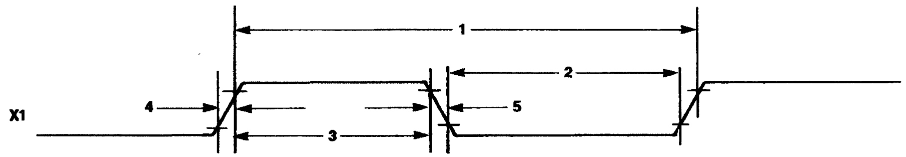


Figure 7 : Transmit Timing Diagram.

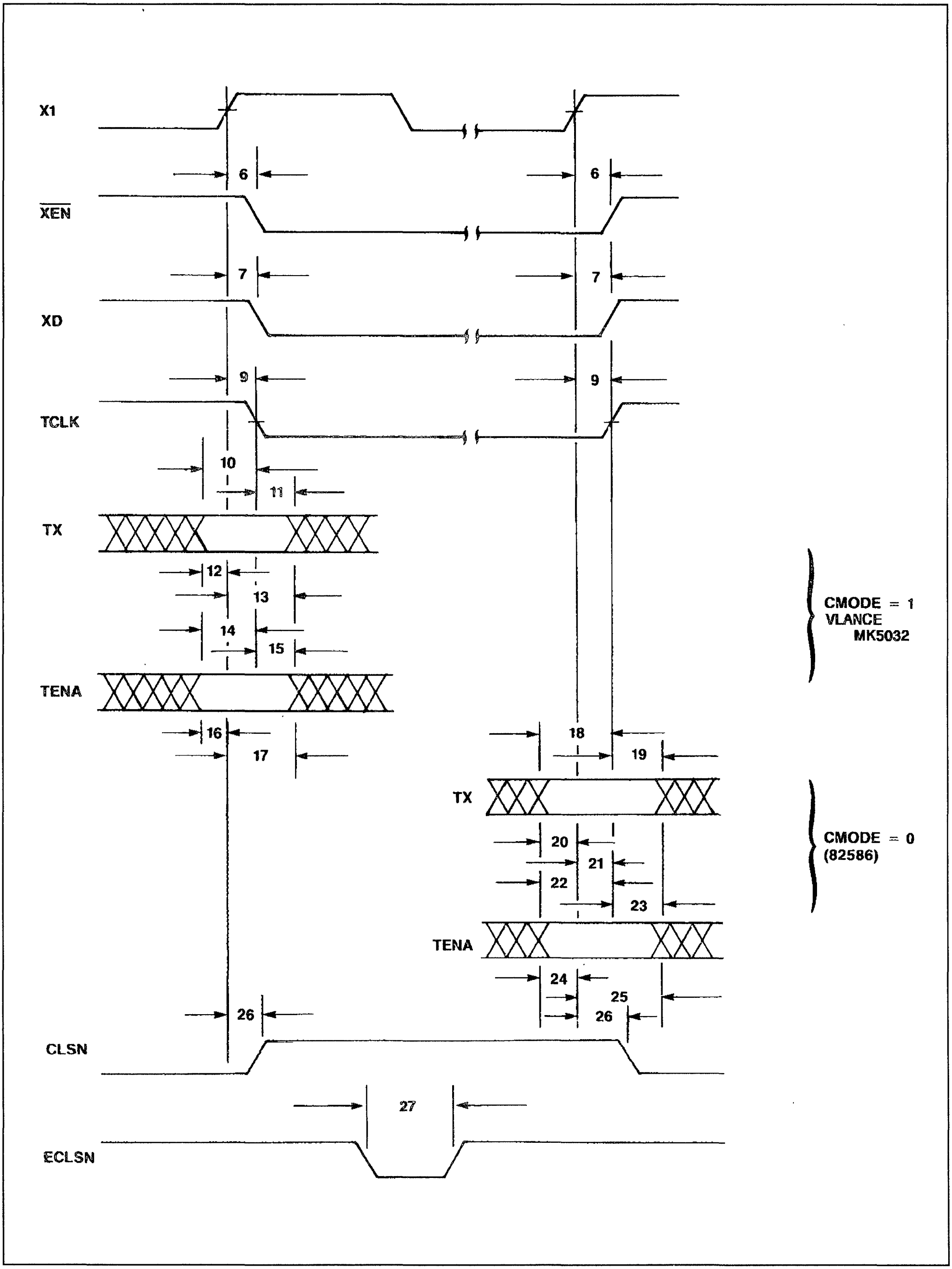


Figure 8 : Receiver Timing Diagram.

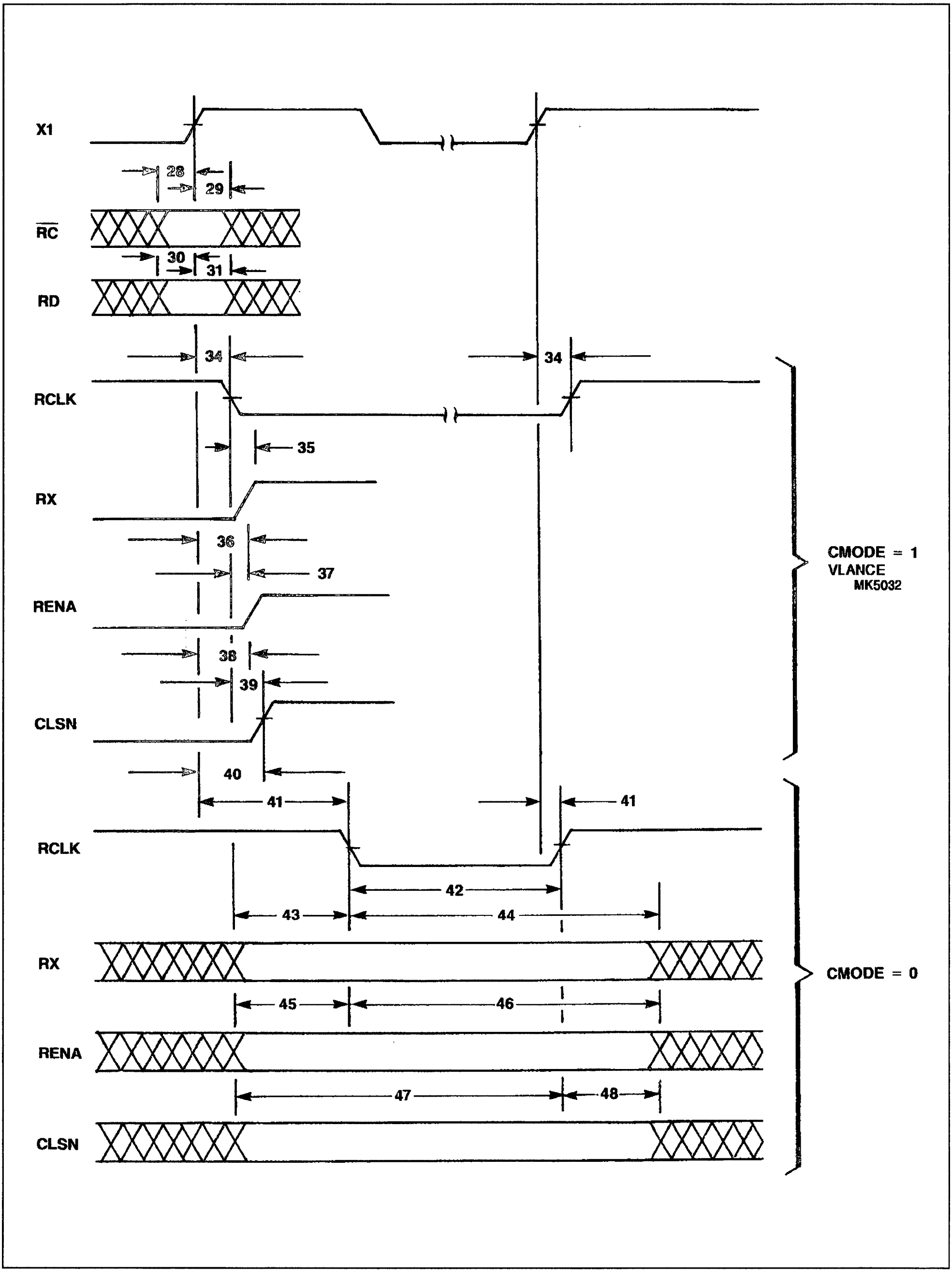
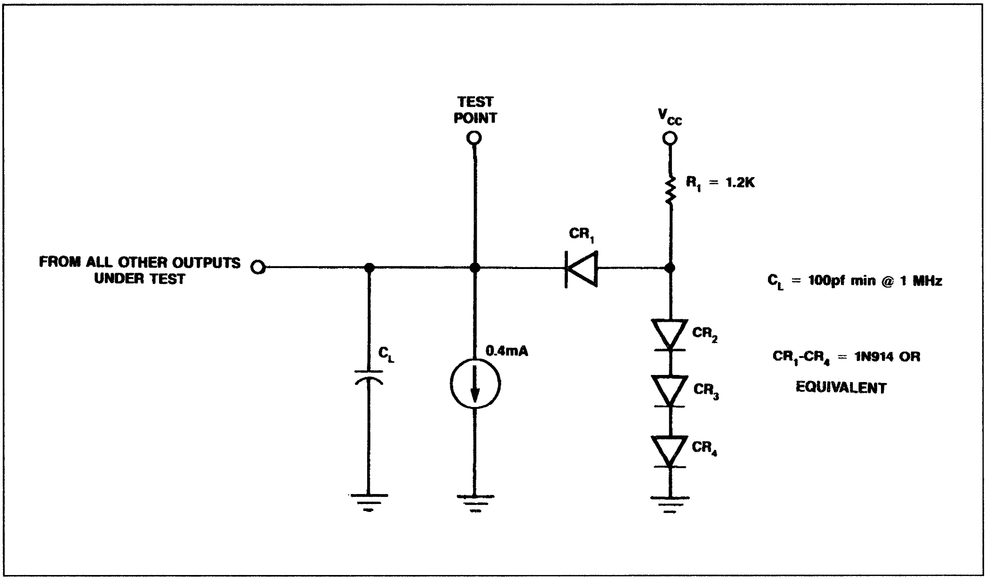
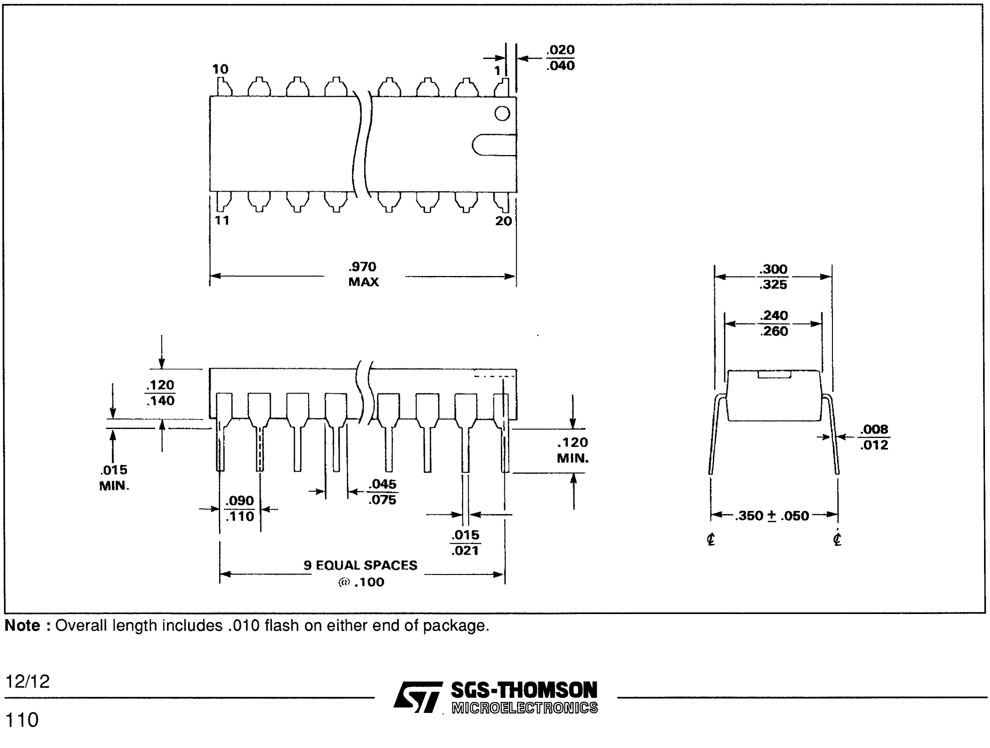


Figure 9 : Output Load Diagram.



PACKAGE DESCRIPTION

20 Pin Plastic - MK5035N



Note : Overall length includes .010 flash on either end of package.

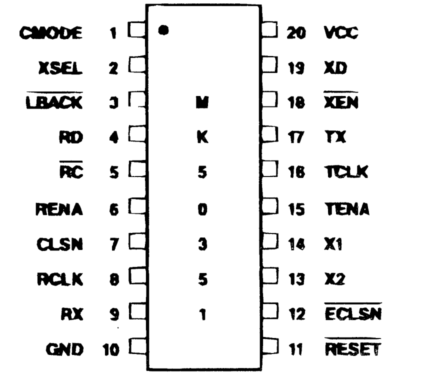
SGS-THOMSON
MICROELECTRONICS

MK50351N

MANCHESTER/STARLAN ENCODER DECODER

- COMPATIBLE WITH STARLAN SPECS
- MANCHESTER DATA ENCODING/DECODING
- SUPPORTS MULTI-POINT EXTENSION
- AUTO COMPENSATION FOR LINE REVERSAL
- COMPATIBLE WITH MOSTEK MK5032 VARIABLE BIT RATE LANCE AND INTEL 82586/82588
- DATA RATES TO 2.66Mbps SUPPORTED
- COLLISION DETECTION CIRCUITRY WITH THE FOLLOWING FEATURES :
  - detects missing mid-bit transitions
  - transitions too close together
  - transitions too far apart
  - external collision input pin
  - carrier dropout
  - watchdog timer
  - AT&T release 1 collision presence signal
  - echo timeout
- RECEIVE END-OF-FRAME DETECTION
  - input protection at end-of-frame
- LOOPBACK CAPABILITY
- RECEIVE CARRIER AUTOMATICALLY CONVERTED TO A LEVEL SIGNAL
- ECHO TIMER TO SIGNAL ERROR IF TRANSMITTED FRAME IS NOT RECEIVED
- HEARTBEAT GENERATION
- IN 82586 MODE, INSENSITIVE TO EXTRA BITS AHEAD OF PREAMBLE
- ON CHIP CRYSTAL OSCILLATOR, 8X OR 10X OPERATION
- CMOS TECHNOLOGY
- 20-PIN DIP
- SINGLE 5-VOLT SUPPLY
- ALL INPUTS AND OUTPUTS TTL COMPATIBLE*
- OUTPUTS ARE ALSO CMOS COMPATIBLE

Figure 1 : MK50351 Pin Assignment.



DESCRIPTION

The MK50351 is a Manchester Encoder/Decoder chip incorporating several features that make it an ideal StarLAN station chip. The MK50351 performs three functions. It encodes data from a controller chip into Manchester data. It decodes Manchester data from the line transceiver and produces NRZ data and clock for the controller chip. It also detects collisions and signals the controller chip that a collision has occurred.

The MK50351 has several enhancements for StarLAN and Multi-Point extension (MPE) StarLAN. These include auto compensation for wiring reversal, echo timer, external collision detect, watchdog timer, and heartbeat, among others.

* Crystal inputs have CMOS thresholds.

PIN DESCRIPTIONS :

CONTROLLER INTERFACE

RX	Output	RX is the serial receive data after decoding.
RENA	Output	This signal indicates that data is available to the controller on the RX output.
RCLK	Output	RCLK is the receive data clock recovered from the incoming data RD.
TX	Input	TX is the serial data to be transmitted. It is clocked into the chip by TCLK.
TENA	Input	This signal indicates that data is valid on the TX input. It goes active with the first bit of transmission.
TCLK	Output	TCLK is the transmit data clock. All transmit interface signals are synchronized to this clock.
CLSN	Output	This signal is asserted when a manchester violation is detected on the RD line or when the external collision input ($\overline{\text{ECLSN}}$) goes active.

TRANSCEIVER INTERFACE

XD	Output	Encoded transmit data output.
$\overline{\text{XEN}}$	Output	Transmit output enable. This signal goes low to indicate XD active. It goes high at the end of transmission.
RD	Input	Encoded receive data input.
$\overline{\text{RC}}$	Input	Receive carrier input. Receive carrier can be either a pulse stream or an active low signal to indicate carrier active. The chip contains internal squelch circuitry, as shown in figure 3, to convert a pulse signal to a level signal.
$\overline{\text{ECLSN}}$	Input	External collision input. When this pin is held low for at least 20nS, an external collision is signaled.

OTHER PINS

CMODE	Input	This input allows the part to be used with either Mostek or Intel controllers :
CMODE = 0, 82586/82588 (see note)		
		Transmit data (TX) is sampled on the rising edge of TCLK. Receive data (RX) transitions on the rising edge of RCLK. TENA - active low RENA - active low - goes active when phase lock loop is locked. CLSN - active low
CMODE = 1, MOSTEK Variable Bit Rate LANCE MK5032		
		TENA - active high RENA - active high - goes active when phase lock loop is locked. CLSN - active high
$\overline{\text{LBACK}}$	Input	When this input is low the part will be put into internal loopback. The transmit data will be internally looped back as receive data. See figure 2.. The outputs XD and $\overline{\text{XEN}}$ will be held inactive during loopback.
$\overline{\text{RESET}}$	Input	When this pin is low the chip is in reset mode. All interface signals will be inhibited except TCLK. $\overline{\text{RESET}}$ should remain active for three TCLK periods.
XSEL	Input	This input selects the clock divider. If XSEL = 0, it is 8X. If XSEL = 1, it is 10X.
X1, X2	Inputs	Crystal oscillator inputs. A crystal can be connected between these inputs, or a CMOS level square wave can be connected to X1 while X2 is left unconnected.
VCC	Input	+ 5V ± 5%
GND		

Note : Compatibility with controller chips based on preliminary controller data sheets.

Figure 2 : StarLAN Encoder Decoder. MK50351.

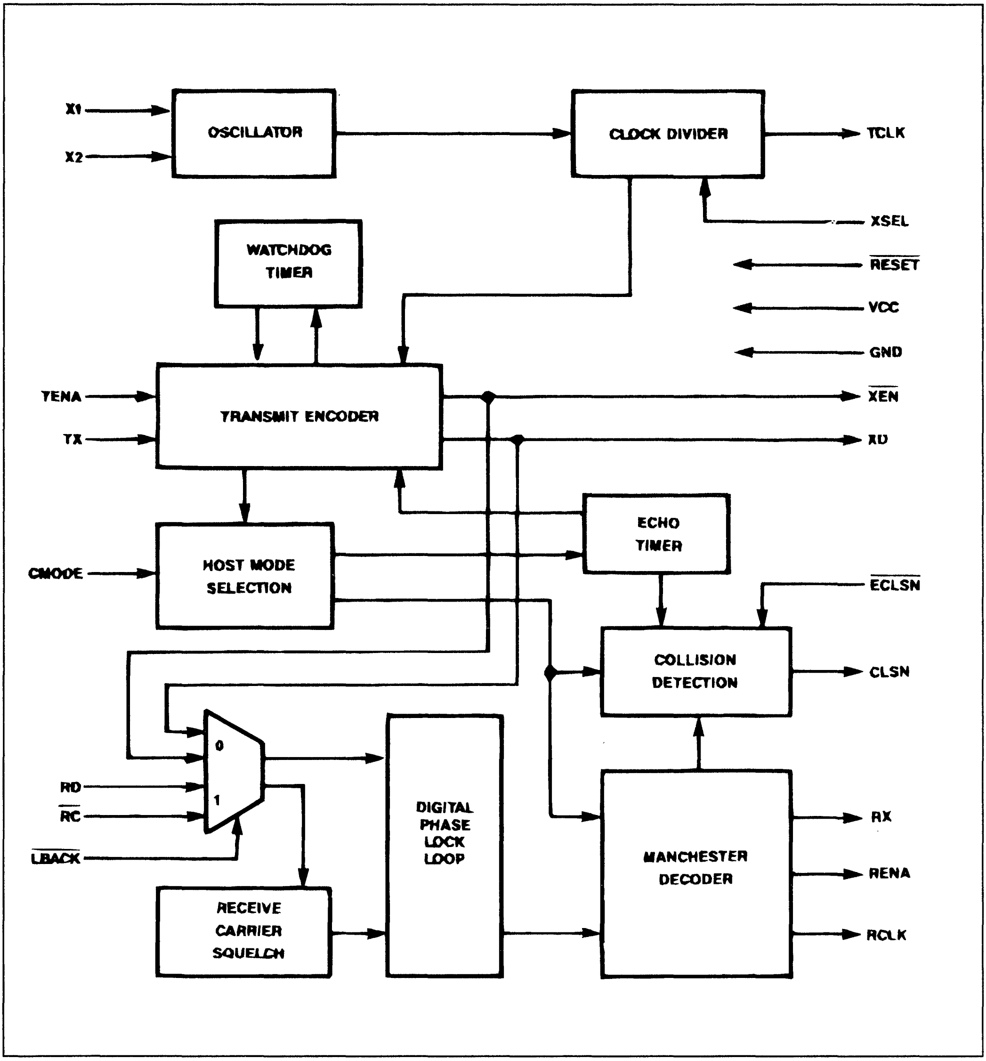
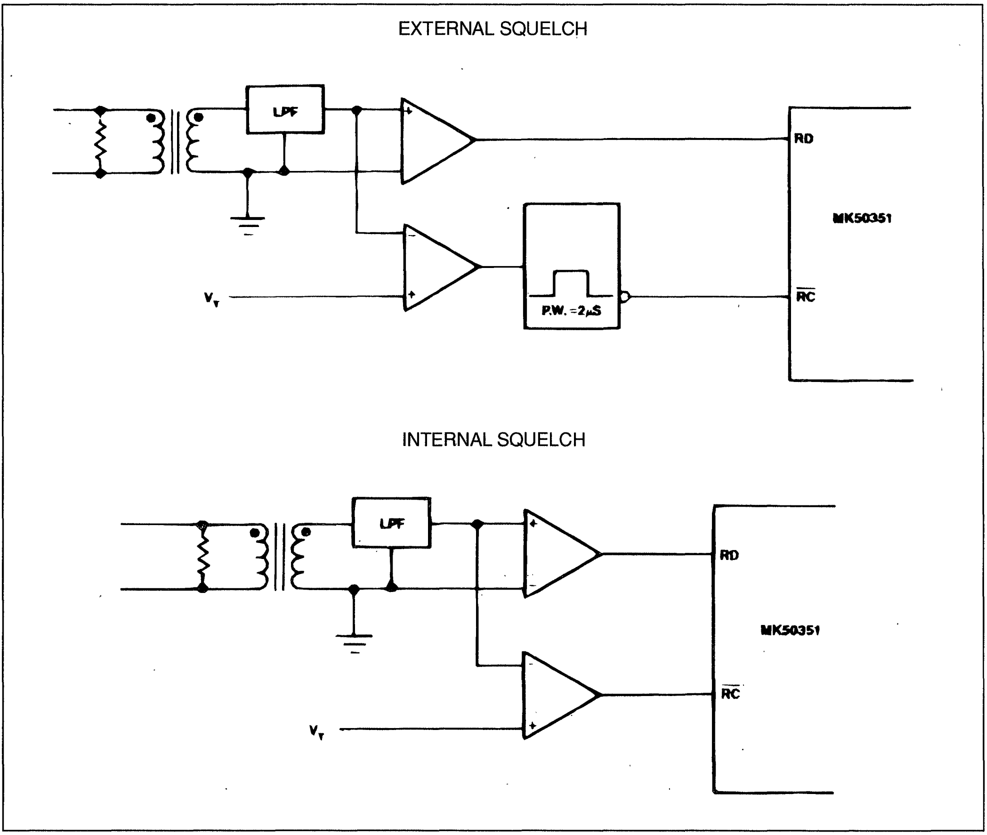


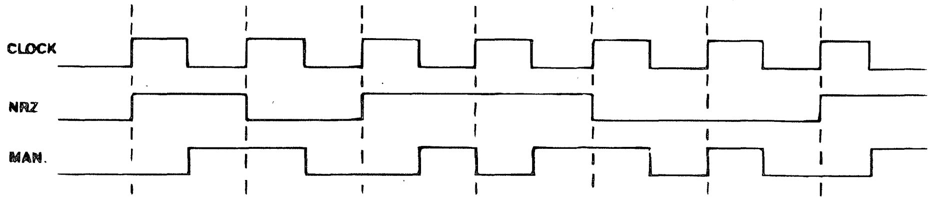
Figure 3 : Internal Versus External Time Squelch.



CIRCUIT DESCRIPTION

TRANSMITTER

The transmitter encodes NRZ data from the controller chip into the Manchester data. The diagram below shows the two encoding schemes.



Data encoding and transmission begin when the controller chip brings TENA active. The start of data encoding is delayed by two bits when in 82586/82588 mode. TX is sampled using TCLK as the clock. The encoded data is output on XD. XEN goes low with the first bit of data output on XD. The transmit delay, delay from TENA active to XEN active, is less than 2 TCLKs. The controller chip signals end of data by bringing TENA inactive.

XD will be held high for an additional 1.5 TCLKs if the last data bit is a one, and for 2 TCLKs if the last data bit is a zero. During this time XEN is held active.

RECEIVER

The receiver consists of four major sections.

1) Receive carrier squelch

2) Internal loopback

3) Digital phase locked loop

4) Manchester decoder

The receiver takes Manchester data in on RD, when receive carrier (RC) is active, and decodes the data into NRZ data and also produces clock (RCLK) from the data. The NRZ data is output to the controller on RX.

RECEIVE CARRIER SQUELCH

The Receive carrier pin has internal squelch logic that allows the signal to be either a level signal or a pulse train. Receive carrier is active low. The receive carrier must be present for 3 clock samples to be considered a valid carrier. Once the carrier is considered valid then it must be active for only one clock sample time every two bit times to remain valid. (see figure 3).

Automatic Compensation For Wiring Reversal

When installing twisted pair telephone wiring, it is often difficult and expensive to maintain proper polarity on the wire pairs. The MK50351 will automatically compensate for this reversal. Any frame that is received with inverse polarity will be detected and decoded with the correct polarity.

LOOPBACK

When loopback is enabled (LBACK low), RD and RC are ignored. Transmit data is internally looped back as receive data. The transmitter outputs XD and XEN are disabled during loopback.

DIGITAL PHASE LOCKED LOOP (DPLL)

The digital phase locked loop is implemented with a counter that clears on each transition of the receive data. The phase locked loop will declare "lock" after receiving data that has two "long transitions". A long transition occurs when the receive data does not change for at least 5/8 (7/10 in 10X mode) of a bit time.

MANCHESTER DECODER

The receive data (after inversion if needed) is fed into the decoder along with the recovered 2X clock from the DPLL. The decoder changes the receive data to NRZ data. The NRZ data is output on RX. RENA signals the controller chip that data is available. (See mode pin descriptions). RCLK is a 1X clock output that is synchronous with the data on RX.

PROTECTION TIME

After the end of a received frame the receiver is disabled for 20 bit times. This protection time guarantees immunity to spikes caused by transformer coupling after the end of frame.

COLLISION

CLSN is an output to the controller chip that indicates a possible problem with the data. There are several sources of collision.

1) Transitions too close together
Collision is signaled if the receive data stream transitions a second time in less than 3/8 (3/10 in 10X mode) bit times.

2) Transitions too far apart
Collision is signaled if the receive data stream does not transition again within 10/8 (12/10 in 10X mode) bit times.

3) Manchester violation
If the data violates Manchester coding rules, then collision is signaled.
A Manchester violation is a missing mid-bit transition.

4) Watchdog timer
If the watchdog timer expires, then collision will be signaled.

5) Echo timer
If the echo timer expires without receive carrier going active, then collision will be signaled.

6) External collision
If the external collision pin (ECLSN) goes low for at least 20ns, then collision will be signaled.

7) Receive carrier lost during transmission
If the MK50351 is transmitting and the receive carrier goes active and then inactive before it is through transmitting, then collision is signaled.

8) Heartbeat
Collision, as a result of heartbeat, will be signaled 8 TCLKs after TENA goes away, and collision will remain active for at least 8 TCLKs.

Once CLSN is activated it remains active until both TENA and RENA go inactive. The exception to this is heartbeat. If heartbeat signals collision, then collision is guaranteed to remain for 8 TCLKs.

WATCHDOG TIMER

The watchdog timer ensures that the MK50351 will not transmit for more than 101K bit times. The timer is started when TENA goes active. The timer resets when TENA goes inactive. If TENA remains active for more than 101K bit times, then the timer will time-out causing collision to be asserted and XEN to go inactive. If loopback is enabled, watchdog timeout will occur after 325 bit times. This particular timer value allow StarLAN HUBs to activate their own jabber functions, thereby alerting net management.

ECHO TIMER

The MK50351 expects the data that it is transmitting to be received on RC/RD within 510 bit times. The echo timer is activated when TENA goes active. If 510 bit times elapse before RENA goes active, then the timer will time out causing collision to be activated.

Oscillator

The MK50351 will accept two forms of clock input : a CMOS input or a crystal. If pin X2 is left unconnected, a 8.0/10.0MHz ± 0.01% CMOS clock may be applied to pin X1. Alternately, a crystal circuit may be connected between X1 and X2 to form the basis of an oscillator. Typically, a 8.0/10.0 ± 0.005% parallel resonant crystal is needed to insure the ± 0.01% frequency accuracy required for StarLAN. Refer to figure 4. A fundamental mode, parallel resonance type crystal should be used with the manufacturer's suggested load capacitance.

Reset Input

The reset pin is an active low Schmidt trigger input. A simple RC network may be used to insure correct operation upon power-up. Refer to fig. 5.

Figure 4 : Oscillator Operation.

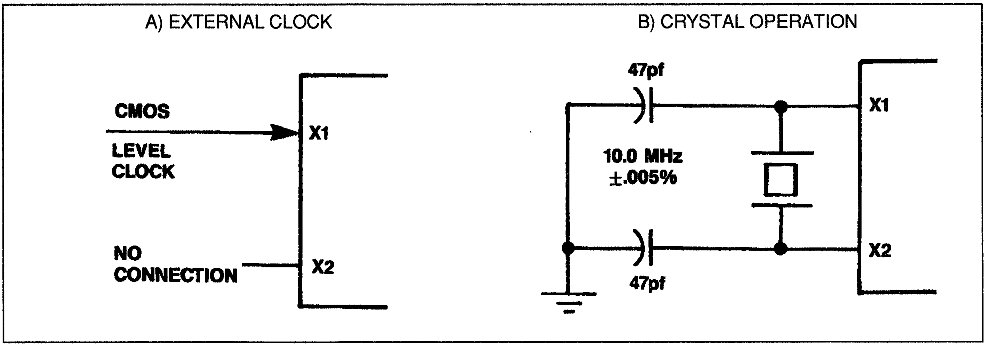
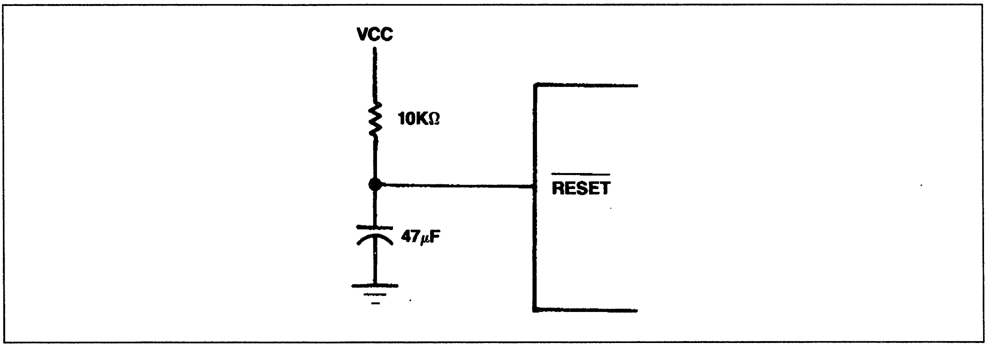


Figure 5 : Typical RC Connection for Power-On Reset.



ELECTRICAL SPECIFICATIONS

This chapter provides tabular presentations for Absolute Maximum Ratings, DC Characteristics, Capacitance and AC Timing Specifications. In addition, illustrations are provided for an Output Load Diagram (figure 9) and Station Timing Diagrams.

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
T_A	Temperature under Bias	– 25 to + 100	°C
T_{stg}	Storage Temperature	– 65 to + 150	°C
V_I	Voltage on any Pin with Respect to Ground	– 0.5 to V_{CC} + 0.5	V
P_D	Power Dissipation (no load)	50	mW

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = + 5V ± 5% unless otherwise specified

Symbol	Test Conditions	Min.	Max.	Unit
V_{IL}		– 0.5	+ 0.8	V
V_{IH}	Except X1	+ 2.0	V_{CC} + 0.5	V
V_{IH}	X1	+ 3.5	V_{CC} + 0.5	V
V_{OL}	@ I_{OL} = 3.2mA, Except X2		+ 0.5	V
V_{OH}	@ I_{OH} = – 0.4mA, Except X2	+ 2.4		V
V_{OH}	@ I_{OH} = – 40µA, Except X2	+ 3.2		V
I_{IL}	@ V_{IN} = 0.4 to V_{CC}		± 10	µA
I_{CC}			8	mA

CAPACITANCE

F = 1MHz

Symbol	Test Conditions	Min.	Max.	Unit
C_{IN}			10	pf
C_{OUT}			10	pf
C_{IO}			20	pf

AC TIMING SPECIFICATIONS

T_A = 0°C to 70°C, V_{CC} = + 5V ± 5% unless otherwise specified, V_{TH} = 2.0V, V_{TL} = 0.8V

#	Signal	Symbol	Parameter	Min.	Typ.	Max.	Unit
1	X1	T_{X1T}	X1 Period	62			ns
2	X1	T_{X1L}	X1 Low Time	24			ns
3	X1	T_{X1H}	X1 High Time	24			ns
4	X1	T_{X1R}	Rise Time of X1	0		8	ns
5	X1	T_{X1F}	Fall Time of X1	0		8	ns
6	$\overline{XEN}$	$T_{\overline{XEN}}$	$\overline{XEN}$ Delay from X1		40	65	ns
7	XD	T_{XD}	XD Delay from X1		40	65	ns
8	XD	J_{XD}	Transmit Jitter $\lvert T_{XD} \uparrow - T_{XD} \downarrow \rvert \div 2$		4	6	ns
9	TCLK	T_{CLK}	TCLK Delay from X1			70	ns
10	TX	T_{TXST1}	TX Setup to Falling Edge of TCLK, CMODE = 1	90			ns
11	TX	T_{TXHT1}	TX Hold from Falling Edge of TCLK, CMODE = 1	15			ns
12	TX	T_{TXS}	TX Setup to X1	15			ns
13	TX	T_{TXH}	TX Hold from X1	15			ns
14	TENA	T_{TNAST1}	TENA Setup to Falling Edge of TCLK, CMODE = 1	90			ns
15	TENA	T_{TNAST1}	TENA Hold from Falling Edge of TCLK, CMODE = 1	15			ns
16	TENA	T_{TENAS}	TENA Setup to X1	15			ns
17	TENA	T_{TENAH}	TENA Hold from X1	15			ns
18	TX	T_{TXST0}	TX Setup to Rising Edge of TCLK, CMODE = 0	90			ns
19	TX	T_{TXHT0}	TX Hold from Rising Edge of TCLK, CMODE = 0	15			ns
20	TX	T_{TXS}	TX Setup to X1, CMODE = 0	15			ns
21	TX	T_{TXH}	TX Hold from X1, CMODE = 0	15			ns
22	TENA	T_{TNAST0}	TENA Setup to Positive Edge of TCLK, CMODE = 0	90			ns
23	TENA	T_{TNAHT0}	TENA Hold from Positive Edge of TCLK, CMODE=0	15			ns
24	TENA	T_{TNAS}	TENA Setup to X1, CMODE = 0	15			ns
25	TENA	T_{TNAH}	TENA Hold from X1, CMODE = 0	15			ns
26	CLSN	T_{CLSN}	CLSN Delay from X1			70	ns
27	$\overline{ECLSN}$	$T_{\overline{ECLSN}}$	Minimum Detected Pulse Width		5	20	ns
28	$\overline{RC}$	$T_{\overline{RC}S}$	$\overline{RC}$ Setup to X1	15			ns
29	$\overline{RC}$	$T_{\overline{RC}H}$	$\overline{RC}$ Hold from X1	15			ns
30	RD	T_{RDS}	RD Setup to X1	15			ns
31	RD	T_{RDH}	RD Hold from X1	15			ns
32	RD	J_{RD6}	RD Incoming Jitter Tolerance, 8X Mode, X1 = 8MHz, $T_{X1T} - \lvert T_{RDS} \uparrow - T_{RDS} \downarrow \rvert$		123	119	ns
33	RD	J_{RD8}	RD Incoming Jitter Tolerance, 10X Mode, X1 = 10MHz, $T_{X1T} - \lvert T_{RDS} \uparrow - T_{RDS} \downarrow \rvert$		198	194	ns
34	RCLK	T_{RCLK}	RCLK Delay from X1, CMODE = 1		40	65	ns
35	TX	T_{RXRCK1}	RX Delay from Falling RCLK, CMODE = 1	– 30		30	ns
36	RX	T_{RX}	RX Delay from X1, CMODE = 1		40	65	ns
37	RENA	$T_{RNARCK1}$	RENA Delay from Falling RCLK, CMODE = 1	– 30		30	ns
38	RENA	T_{RENA}	RENA Delay from X1, CMODE = 1		45	65	ns
39	CLSN	$T_{CSNRCK1}$	CLSN Delay from Falling Edge RCLK, CMODE = 1	– 30		30	ns

AC TIMING SPECIFICATIONS (continued)

#	Signal	Symbol	Parameter	Min.	Typ.	Max.	Unit
40	CLSN	T_{CLSN}	CLSN Delay from X1, CMODE = 1			70	ns
41	RCLK	T_{RCLK0}	RCLK Delay from X1, CMODE = 0		40	65	ns
42	RCLK	P_{RCLK}	RCLK Pulse Width, CMODE = 0	T_{X1T}–20		T_{X1T}+20	ns
43	RCLK	T_{RXCLK}	RCLK Delay from RX Stable, CMODE = 0	T_{X1T}–20			ns
44	RX	T_{CLKRX}	RX Hold from Falling Edge of RCLK, CMODE = 0	2*T_{X1T}–20			ns
45	RCLK	T_{RNACLK}	RCLK Delay from RENA Stable, CMODE = 0	T_{X1T}–20			ns
46	RENA	T_{CLKRNA}	RENA Hold from Falling Edge of RCLK, CMODE = 0	2*T_{X1T}–20			ns
47	RCLK	T_{CSNCLK}	Rising RCLK Delay from CLSN Stable, CMODE = 0	2*T_{X1T}–20			ns
48	CLSN	T_{CLKCSN}	CLSN Delay from Rising Edge of RCLK, CMODE = 0	T_{X1T}–20			ns

Figure 6 : External X1 Timing Diagram.

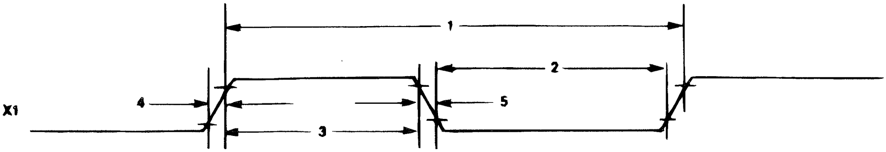


Figure 7 : Transmit Timing Diagram.

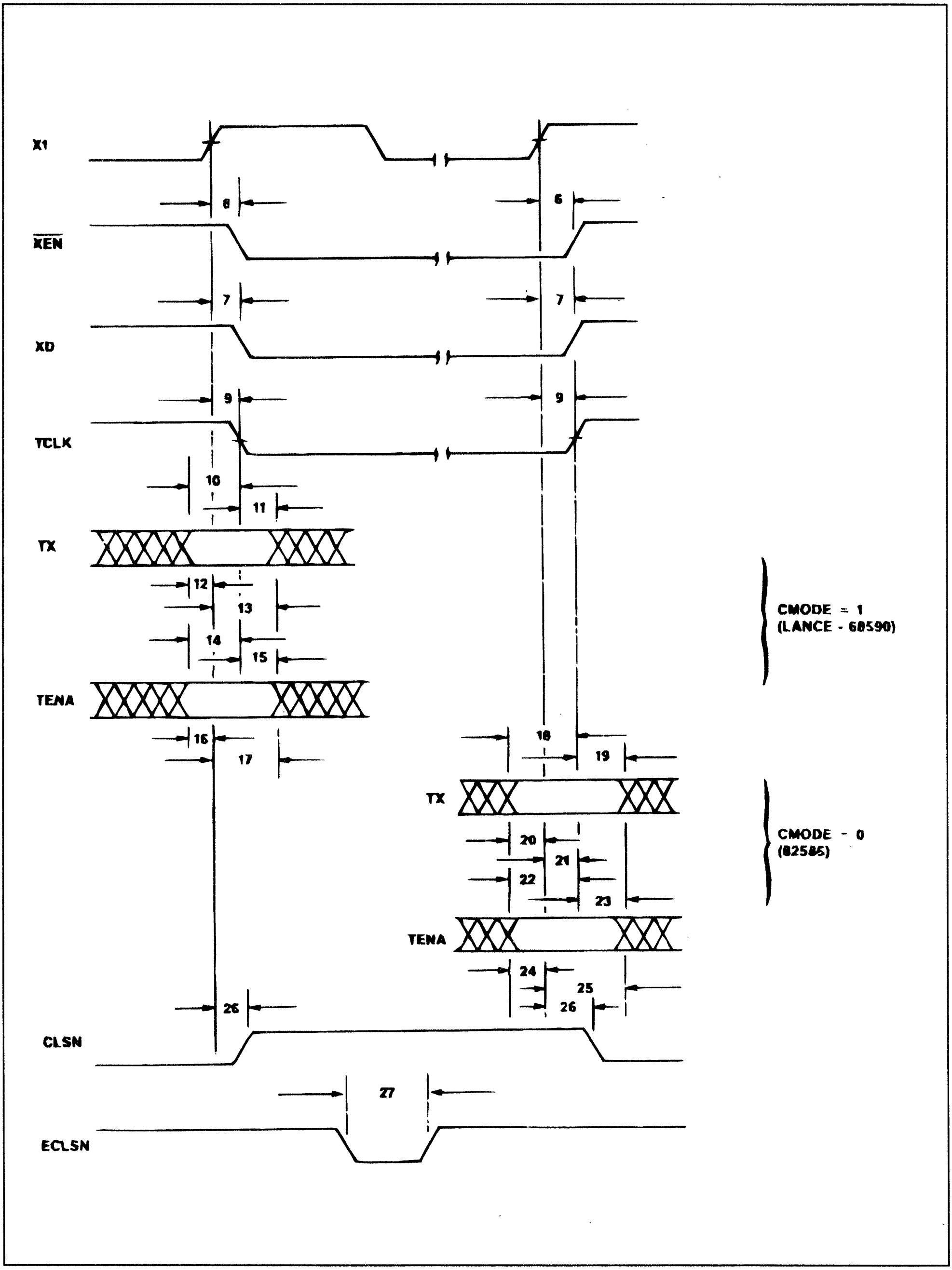


Figure 8 : Receiver Timing Diagram.

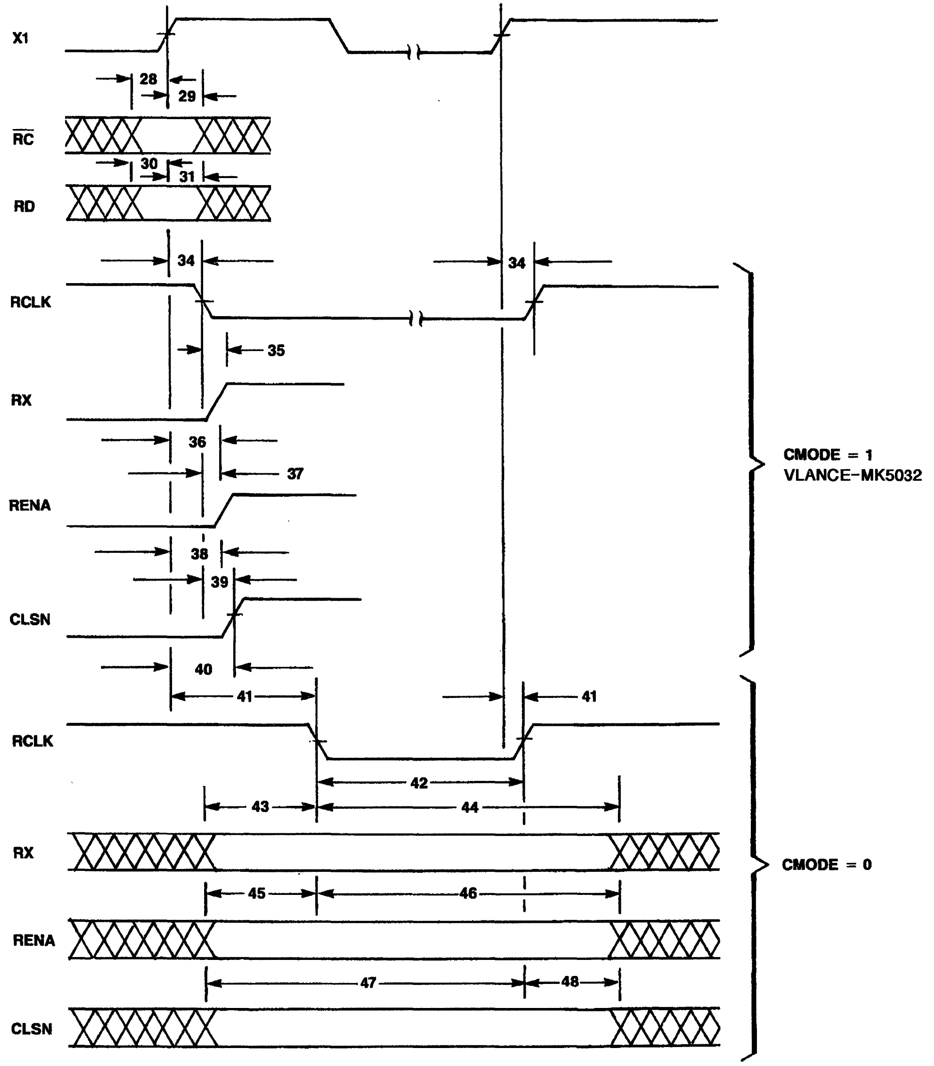
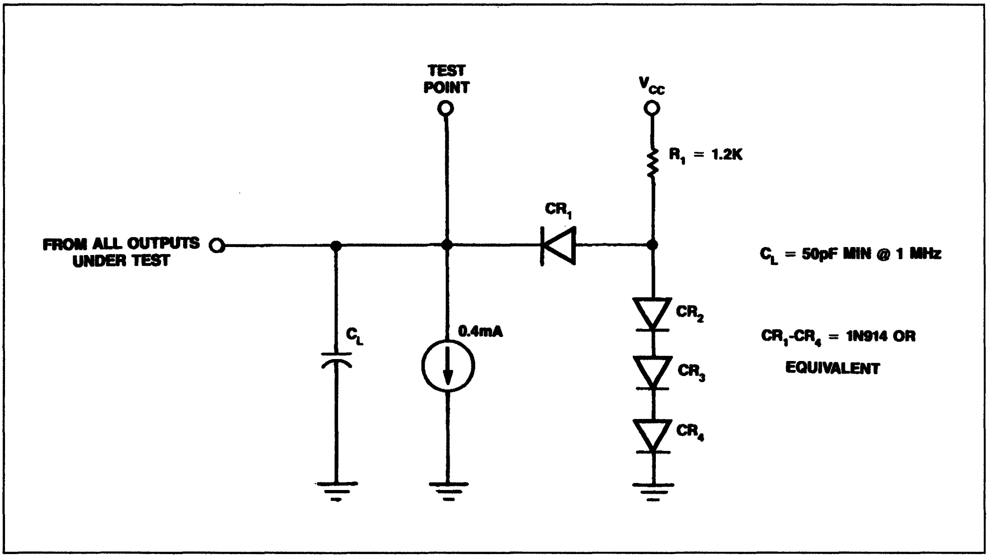
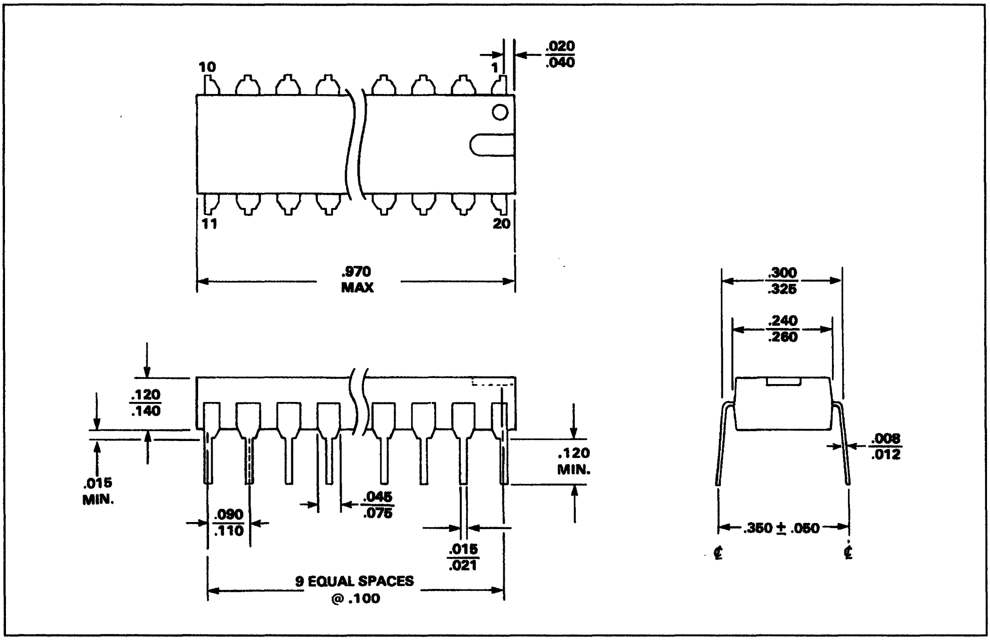


Figure 9 : Output Load Diagram.



PACKAGE DESCRIPTION

20 Pin Plastic - MK50351N

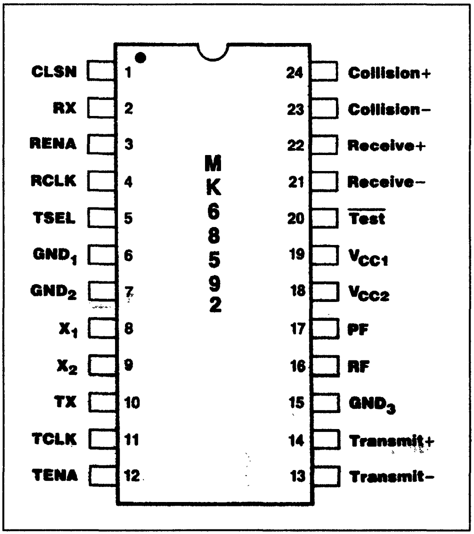


Note : Overall length includes .010 flash on either end of package.

SERIAL INTERFACE ADAPTER (SIA)

- COMPATIBLE WITH ETHERNET AND IEEE-802.3 SPECIFICATIONS
- CRYSTAL-CONTROLLED MANCHESTER ENCODER/DECODER
- MANCHESTER DECODER ACQUIRES CLOCK AND DATA WITHIN SIX-BIT TIMES WITH AN ACCURACY OF ± 3NS
- GUARANTEED CARRIER AND COLLISION DETECTION SQUELCH THRESHOLD LIMITS
  - carrier/collision detected for inputs more negative than – 275mV
  - no carrier/collision for inputs more positive than – 175mV
- INPUT SIGNAL CONDITIONING REJECTS TRANSIENT NOISE
  - transients < 10ns for collision detector inputs
  - transients < 20ns for carrier detector inputs
- RECEIVER DECODES MANCHESTER DATA WITH UP TO ± 20NS CLOCK JITTER (at 10MHz)
- TTL COMPATIBLE HOST INTERFACE
- TRANSMIT OSCILLATOR ACCURACY ± 0.01% (without adjustments)

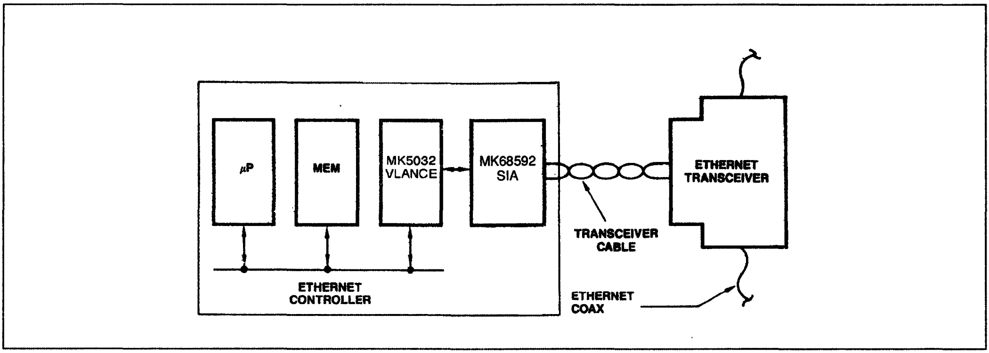
Figure 1 : Pin Assignments.



DESCRIPTION

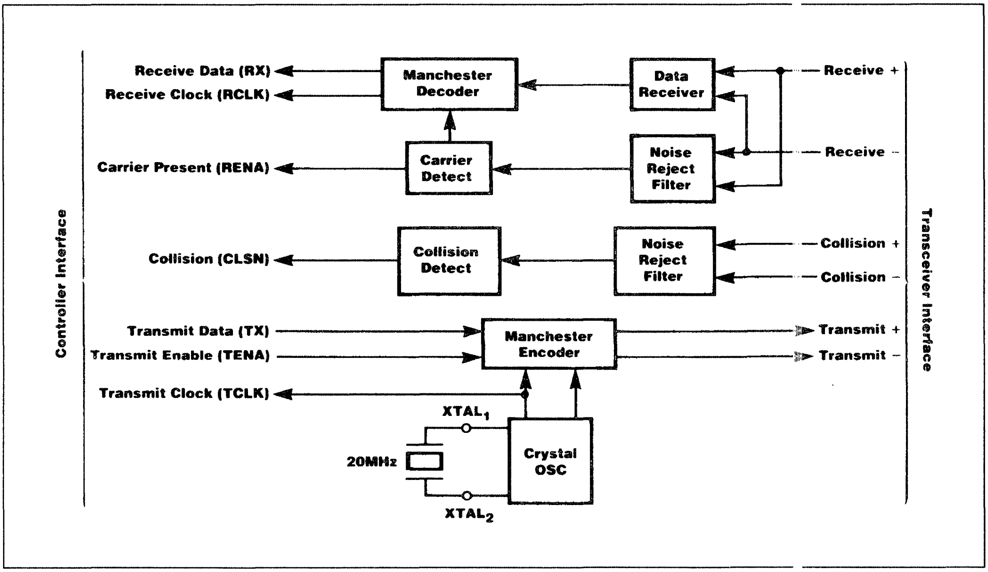
The MK68592 Serial Interface Adapter (SIA) is a Manchester Encoder/Decoder compatible with Ethernet and IEEE-802.3 specifications. In an Ethernet/IEEE-802.3 application, the MK68592 interfaces the MK5032 Variable Bit Rate Local Area Network Controller for Ethernet (VLANCE) to the Ethernet transceiver cable, acquires clock and data within 6 bit-times and decodes Manchester data up to ± 20ns phase jitter at 10MHz. SIA provides both guaranteed signal threshold limits and transient noise suppression circuitry in both data and collision paths to minimize false start conditions.

Figure 2 : Typical Ethernet Node.



VLANCE is a trademark of ST Corporation.

Figure 3: MK68592 Block Diagram.



PIN DESCRIPTION

CLSN **Collision** (output). A TTL active high output. Signals at the Collision ± terminals meeting threshold and pulse width requirements will produce a logic high at CLSN output. When no signal is present at Collision ±, CLSN output will be low.

RX **Receive Data** (output). A MOS/TTL output, recovered data. When $\overline{\text{there}}$ is no signal at Receive ± and $\overline{\text{TEST}}$ is high, RX is high. RX is actuated with RCLK and remains activated until end of message. During reception, RX is synchronous with RCLK and changes after the rising edge of RCLK.

RENA **Receive Enable** (output). A TTL active high output. When there is no signal at Receive ± and $\overline{\text{TEST}}$ is high, RENA is low. Signals at Receive ± meeting threshold and pulse width requirements will produce a logic high at RENA. When Receive ± becomes idle, RENA returns to the low state synchronous with the rising edge of RCLK.

RCLK **Receive Clock** (output). A MOS/TTL output recovered clock. When $\overline{\text{there}}$ is no signal at Receive ± and $\overline{\text{TEST}}$ is high, RCLK is low. RCLK is activated after the third negative data transition at Receive ±, and remains active until end of message. When $\overline{\text{TEST}}$ is low, RCLK is enabled.

TX **Transmit** (input). TTL compatible input. When TENA is high, signals at TX meeting setup and hold time to TLCK will be encoded as normal Manchester at Transmit + and Transmit –.

TX High : Transmit + is negative with respect to Transmit – for first half of data bit cell.

TX Low : Transmit + is positive with respect to Transmit – for first half of data bit cell.

PIN DESCRIPTION

TENA **Transmit Enable** (input). TTL compatible input. Active high data encoder enable. Signals meeting setup and hold time to TCLK allow encoding of Manchester data from TX to Transmit + and Transmit –.

TCLK **Transmit Clock** (output). MOS/TTL output. TCLK provides symmetrical high and low clock signals at data rate for reference timing of data to be encoded. It also provides clock signals for the controller chip (MK68590 VLANCE) and an internal timing reference for receive path voltage controlled oscillators.

Transmit +
Transmit – **Transmit** (outputs). A differential line output. This line pair is intended to operate into terminated transmission lines. For signals meeting setup and hold time to TCLK at TENA and TX Manchester clock and data are outputted at Transmit +/Transmit –. When operating into a 78Ω terminated transmission line, signalling meetings the required output levels and skew for both Ethernet and IEEE-802.3 drop cables.

Receive +
Receive – **Receiver** (inputs). A differential input. A pair of internally biased line receivers consisting of a carrier detect receiver with offset threshold and noise filtering to detect the signal, and a data recovery receiver with no offset for Manchester data decoding.

Collision +
Collision – **Collision** (inputs). A differential input. An internally biased line receiver input with offset threshold and noise filtering. Signals at Collision ± have no effect on data path functions.

TSEL **Transmit Mode Select**. An open collector output and sense amplifier input.

TSEL Low : Idle transmit state Transmit + is positive with respect to Transmit –.

TSEL High : Idle transmit state Transmit + and Transmit – are equal, providing "zero" differential to operate transformer coupled loads.

When connected with an RC network, TSEL is held low during transmission. At the end of transmission, the open collector output is disabled, allowing TSEL to rise and provide a smooth transmission from logic high to "zero" differential idle. Delay and output return to zero are externally controlled by the RC time constant TSEL.

X_1, X_2 **Biased Crystal Oscillator.** X_1 is the input and X_2 is the bypass port. When connected for crystal operation, the system clock which appears at TCLK is half the frequency of the crystal oscillator. X_1 may be driven from an external source of two times the data rate.

RF **Frequency Setting Voltage Controlled Oscillator** (V_{CO}) **Loop Filter.** This loop filter output is a reference voltage for the receive path phase detector. It also is a reference for timing noise immunity circuits in the collision and receive enable path. Nominal reference V_{CO} gain is 1.25 TCLK frequency MHz/V.

PF **Receive Path V_{CO} Phase Lock Loop Filter.** This loop filter input is the control for receive path loop damping. Frequency of the receive V_{CO} is internally limited to transmit frequency ± 12%. Nominal receive V_{CO} gain is 0.25 reference V_{CO} gain MHz/V.

$\overline{\text{TEST}}$ **Test Control** (input). A static input that is connected to V_{CC} for normal MK68591/2 operation and to ground for testing of receive path function. When $\overline{\text{TEST}}$ is grounded, RCLK and RX are enabled so that receive path loop may be functionally tested.

GND_1 **High Current Ground**

GND_2 **Logic Ground**

GND_3 **Voltage Controlled Oscillator Ground**

V_{CC1} **High Current and Logic Supply**

V_{CC2} **Voltage Controlled Oscillator Supply**

* Non-Return-to-Zero.

FUNCTIONAL DESCRIPTION

The MK68592 Serial Interface Adapter (SIA) has three basic functions. It is a Manchester Encoder/line driver in the transmit path, a Manchester Decoder with noise filtering and quick lock-on characteristics in the receive path, and a signal detect/converter (10MHz differential to TTL) in the collision path. In addition, the SIA provides the interface between the TTL logic environment of VLANCE and the differential signaling environment in the transceiver cable.

TRANSMIT PATH

The transmit section encodes separate clock and NRZ* data input signals meeting the set up and hold time to TCLK at TENA and TX, into a standard Manchester II serial bit stream. The transmit outputs (Transmit +/ Transmit –) are designed to operate into terminated transmission lines. When operating into a 78Ω terminated transmission line, signaling meets the required output levels and skew for both Ethernet and IEEE-802.3.

Transmitter Timing and Operation. A 20MHz fundamental mode crystal oscillator provides the basic timing reference in the SIA. It is divided by two to create the transmit clock reference (TCLK). Both 20MHz and 10MHz clocks are fed into the Manchester Encoder to generate the transitions in the encoded data stream. The 10MHz clock, TCLK, is used by the SIA to internally synchronize transmit data (TX) and transmit enable (TENA). TCLK is also used as a stable bit-rate clock by the receive section of the SIA and by other devices in the system (the MK5032 VLANCE uses TCLK to drive its internal state machine). The oscillator may use an external 0.005% crystal or an external TTL level input as a reference. Transmit accuracy of 0.01% is achieved (no external adjustments are required).

TENA is activated when the first bit of data is made available on TX. As long as TENA remains high, signals at TX will be encoded as Manchester and will appear at Transmit + and Transmit –. When TENA goes low, the differential transmit outputs go to one of the two idle states defined below :

- TSEL High : The idle state of Transmit +/ Transmit – yields "zero" differential to operate transformer coupled loads (see figure 14a).
- TSEL Low : In this idle state, Transmit + is positive to Transmit – logical high (see figure 14b).

RECEIVE PATH

The principle function of the receiver is the separation of the Manchester encoded data stream into clock and NRZ data.

Input Signal Conditioning. Before the data and clock can be separated, it must be determined whether there is "real" data or unwanted noise at the transceiver interface. The MK68591/2 SIA carrier detection receiver provides a static noise margin of – 175 to – 275mV for received carrier detection. These DC thresholds assure that no signal more positive than – 175mV is ever decoded and that signals more negative than – 275mV are always decoded. Transient noise of less than 10ns duration in the collision path and 20ns duration in the data path are also rejected.

This signal conditioning prevents unwanted idle noise on the transceiver cable from causing "false starts" in the receiver. This helps assure a valid response to "real" data.

The receiver section, shown in figure 6, consists of two data paths. The receive data path is designed to be a zero threshold, high bandwidth receiver. The carrier detection receiver has an additional bias generator. Only data amplitudes larger than the bias level are interpreted as valid data. The noise rejection filter prevents noise transients of less than 20ns from enabling the data receiver output. The collision detector similarly rejects noise transients of less than 10ns.

Figure 4 : Transmit Section.

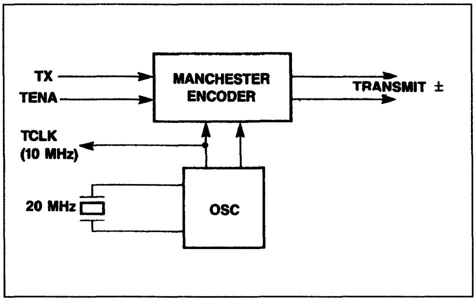
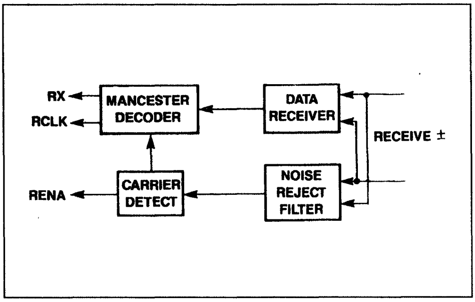


Figure 5 : Receiver.



Receiver Section Timing. Receive Enable (RENA) is the "carrier present" indication established when a signal of sufficient amplitude (V_{IDC}) and duration (t_{RPWR}) is present at the receive inputs. Receive Clock (RCLK) and Receive Data (RX) become available after the third negative data transition at Receive +/ Receive – inputs, and stay active until the end of a packet. During reception, RX is synchronous with RCLK, changing after the rising edge of RCLK.

The receiver detects the end of a packet when the normal transition on the differential inputs cease. After the last low-to-high transition, RENA goes low and RCLK completes one last cycle, storing the last data bit. It then becomes and remains low (see Receive End of Packet Timing diagrams). When TEST is low, RCLK continues to run, tracking data (if available) or synchronize with TCLK.

Receive Clock Control. To insure quick capture of incoming data, the receiver phase-locked-loop is frequency locked to the transmit oscillator and it phase locks to incoming data edges. Clock and data are available within 6 bit times (accurate to within ± 3ns). The SIA will decode jittered data of up to ± 20ns (see figure 7).

Differential I/O Terminations. The differential input for the Manchester data (receive ±) is externally terminated by two 40.2Ω ± 1% resistors and one optional common mode bypass capacitor. The differential input impedance Z_{IDF} and the common mode input Z_{ICM} are specified so that the Ethernet specification for cable termination impedance is met using standard 1% resistor terminators. The Collision ± differential input is terminated in exactly the same way as the receive input (see figure 8).

Collision Detection. The Ethernet Transceiver detects collisions on the Ethernet and generates a 10MHz signal on the transceiver cable (Collision +/ Collision –). This collision signal passes through an input stage which assures signal levels and pulse duration. When the signal is detected by the SIA, the SIA sets the CLSN line high. This condition continues for approximately 190ns after the last low-to-high transition on Collision +/ Collision –.

Figure 6 : Receiver Section Detail.

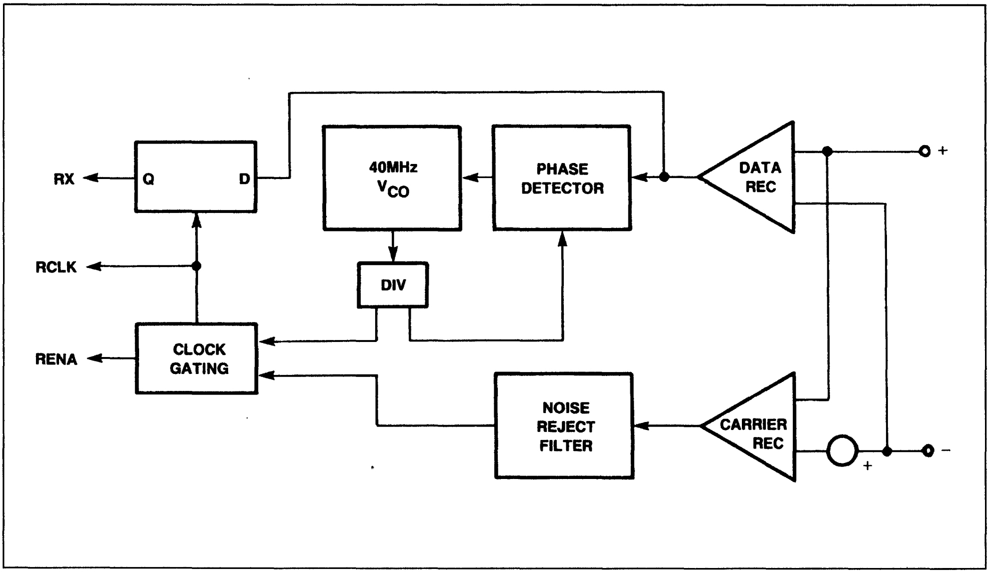


Figure 7 : Maximum Jitter Impact on Sampling.

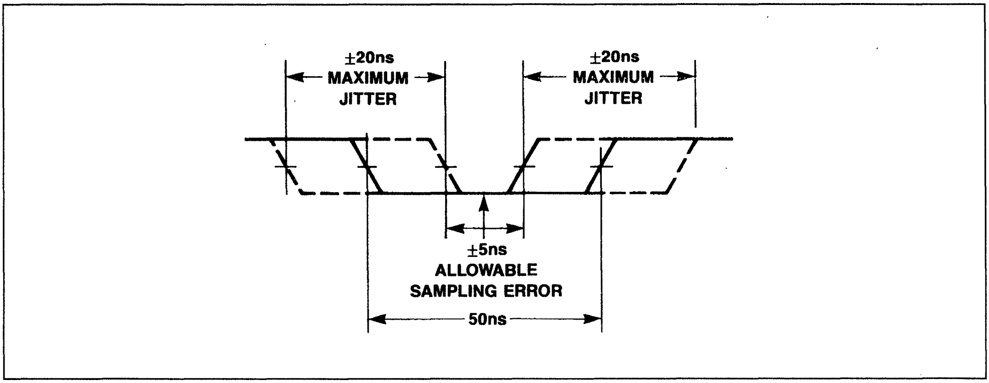
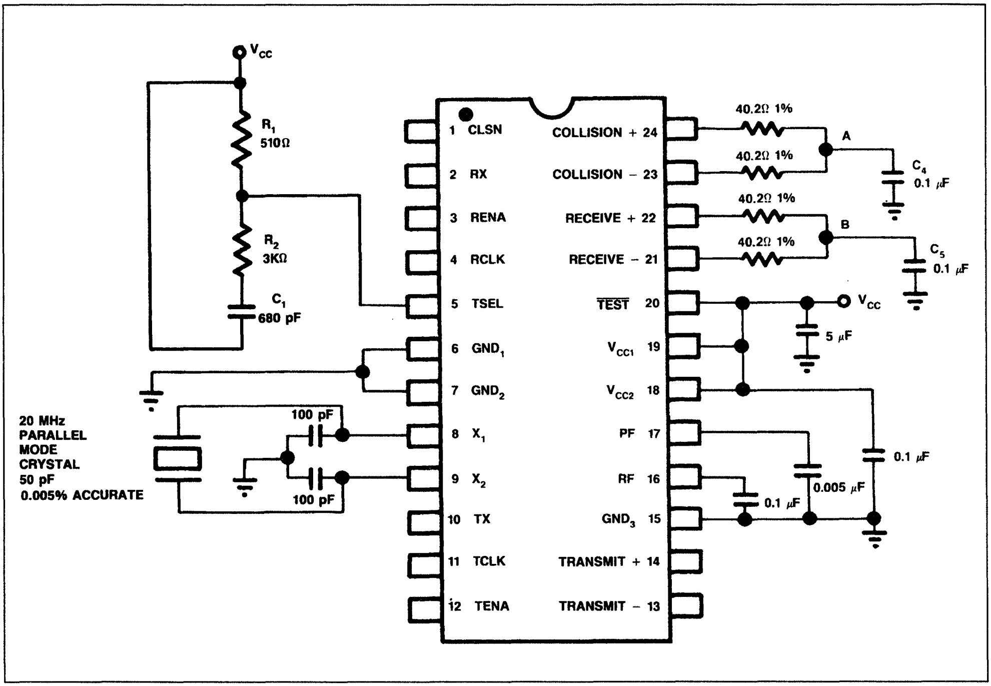
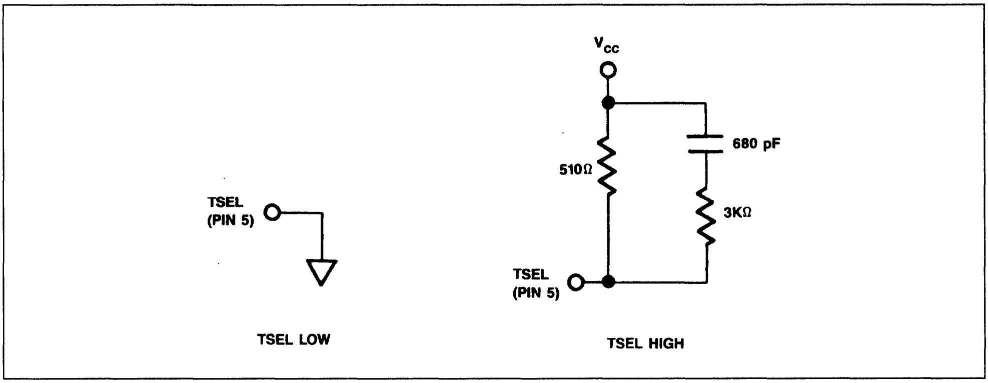


Figure 8 : MK68592 External Component Diagram.



Notes :
1. Connect R_1, R_2, C_1 for 0 differential nontransmit. Connect to ground for logic 1 differential nontransmit.
2. Pin 20 shown for normal device operation.
3. Nodes A and B may be connected directly to ground for proper decoder operations, or to the common mode bypass C_4 and C_5. Some direct coupled transceivers require C_4 and C_5 to ground for proper operation.

Figure 9 : Transmit Mode Select (TSEL) Connection.



ABSOLUTE MAXIMUM RATINGS*

Symbol	Parameter	Value	Unit
T_{stg}	Storage Temperature	– 65 to + 150	°C
T_A	Temperature (ambient) under Bias	0 to 70	°C
V_S	Supply Voltage to Ground Potential Continuous	+ 7.0	V
	DC Voltage applied to Outputs for High Output State	– 0.5 + V_{CC} Max	
	DC Input Voltage (logic inputs)	+ 5.5	V
	DC Input Voltage (receive/collision)	– 6 to + 6	V
	Transmit ± Output Current	– 50 to + 5	mA
	DC Output Current, into Outputs	100	mA
	DC Input Current (logic inputs)	± 30	mA

* Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS OVER OPERATING RANGE. The following conditions apply unless otherwise specified :
T_A = 0 to 70°C, V_{CC} = 5.0V ± 10%, MIN = 4.5V, MAX = 5.5V, period of crystal oscillator (T_{OSC}) = 50ns.

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{OH}	Output High Voltage RX, RENA, CLSN, TCLK, RCLK	I_{OH} = – 1.0mA	2.4	3.4		V
V_{OL}	Output Low Voltage RCLK, TCLK, RENA, RX, CLSN, TSEL	I_{OL} = 16mA,		0.36	0.5	V
		I_{OL} = 1mA		0.25	0.4	
V_{OD}	Differential Output Voltage (Transmit +) – (Transmit –) V_0	R_L = 78Ω Figure 19	550	670	770	mV
	V_0		– 550	– 670	– 770	
$V_{0D\ OFF}$	Transmit Differential Output Idle Voltage	R_L = 78Ω Figure 19 T_{SEL} = HIGH	– 20	0.5	20	mV
$I_{OD\ OFF}$	Transmit Differential Output Idle Current		– 0.5	± 0.1	0.5	mA
V_{CMT}	Common Mode Output Transmit Voltage	Figure 19 R_L = 78Ω	0	2.5	5	V
V_{ODI}	Differential Output Voltage Imbalance (Transmit ±) $\| \|V_0\| - \|V_0\| \|$			5	20	mV
V_{IH}	Input High Voltage TTL		2.0			V
I_{IH}	Input High Current TTL	V_{CC} = Max, V_{IN} = 2.7V			+ 50	µA
V_{IL}	Input Low Voltage TTL				0.8	V
I_{IL}	Input Low Current TTL	V_{CC} = Max, V_{IN} = 0.4V		– 270	– 400	µA
V_{IRD}	Differential Input Threshold (rec data)	Figure 20	– 25	0	+ 25	mV
V_{IDC}	Differential Input Threshold (carrier/collision ±)	Figure 20	– 175	– 225	– 275	mV
I_{CC}	Power Supply Current	t_{OSC} = 50ns		125	180	mA
		t_{OSC} = 50ns, T_A = Max			160	
V_{IB}	Input Breakdown Voltage V_I = + 5.5 (TX, TENA, $\overline{TEST}$)	I_I = 1mA	5.5			V
V_{IC}	Input Clamp Voltage	I_{IN} = – 18mA			– 1.2	V
I_{SCO}	RX, TCLK, CLSN, RENA, RCLK Short Circuit Current		– 40	– 80	– 150	mA
R_{IDF}	Differential Input Resistance	V_{CC} = 0 to Max	6	8.4	13	kΩ
R_{ICM}	Common Mode Input Resistance	V_{CC} = 0 to Max	1.5	2.1	7.5	kΩ
V_{ICM}	Receive and Collision Input Bias Voltage	I_{IN} = 0	1.5	3.5	4.2	V
I_{ILD}	Receive and Collision Input Low Current	V_{IN} = – 1V	– 0.6	– 1.06	– 1.64	mA
I_{IHD}	Receive and Collision Input High Current	V_{IN} = 6V	+ 0.4	+ 0.6	+ 1.10	mA
I_{IHZ}	Receive and Collision Input High Current	V_{CC} = 0, V_{IN} = + 6V	0.4	1.28	1.86	mA

SWITCHING CHARACTERISTICS OVER OPERATING RANGE. The following conditions apply unless otherwise specified :
T_A = 0 to + 70°C, V_{CC} = 5.0V ± 10%, MIN = 4.5V, MAX = 5.5V, T_{OSC} = 50ns

RECEIVER SPECIFICATIONS

#	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
1	RCLK	t_{RCT}	RCLK Cycle Time	C_L = 50pF Figure 17a (see note)	85	100	118	ns
2	RCLK	t_{RCH}	RCLK High Time		38	50		ns
3	RCLK	t_{RCL}	RCLK Low Time		38	50		ns
4	RCLK	t_{RCR}	RCLK Rise Time			2.5	8	ns
5	RCLK	t_{RCF}	RCLK Fall Time			2.5	8	ns
6	RX	t_{RDR}	RX Rise Time			2.5	8	ns
7	RX	t_{RDF}	RX Fall Time			2.5	8	ns
8	RX	t_{RDH}	RX Hold Time (RCLK to RX change)		5	8		ns
9	RX	t_{RDS}	RX Prop Delay (RCLK to RX stable)			8	25	ns
10	RENA	t_{DPH}	RENA Turn–on Delay (V_{IDC} Max on receive ± to $RENA_H$)	Figures 10, 16a, and 20		50	80	ns
11	RENA	t_{DPO}	RENA Turn–off Delay (V_{IDC} Min on Receive ± to $RENA_L$)	Figures 11 and 20		265	300	ns
12	RENA	t_{DPL}	RENA Low Time	Figure 11	120	200		ns
13	Rec ±	t_{RPWR}	Receive ± Input Pulse Width to Reject (input < V_{IDC} Min)	Figures 16a and 20		30	20	ns
14	Rec ±	t_{RPWO}	Receive ± Input Pulse Width to Turn–on (input > VIDC Max)	Figures 16a and 20	45	30		ns
15	RCLK	t_{RLT}	Decoder Acquisition Time	Figure 10		390	450	ns

Note : Assumes equal capacitance loading on RCLK and RX.

SWITCHING CHARACTERISTICS OVER OPERATING RANGE (continued)

COLLISION SPECIFICATION

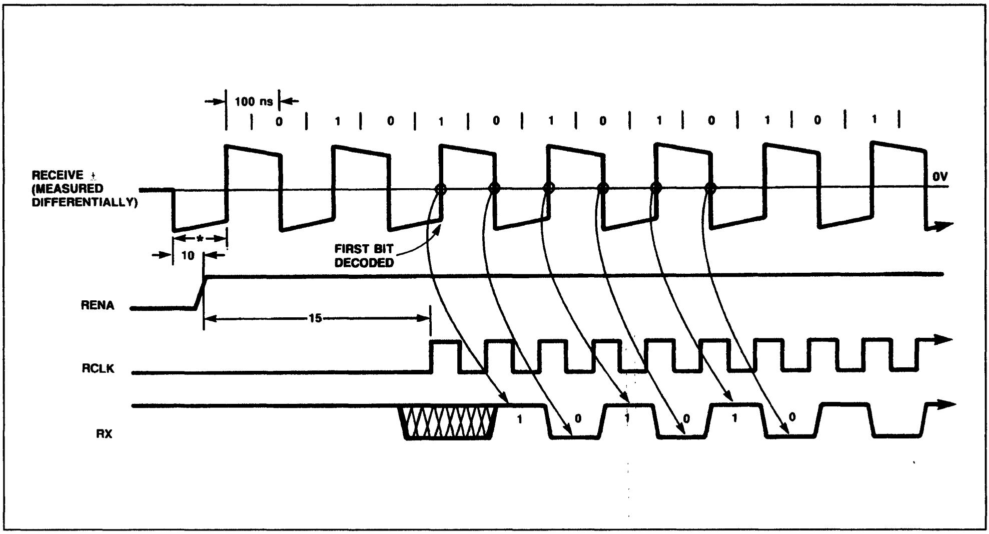
#	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
16	Coll ± 0	t_{CPWR}	Collsion Input Pulse Width to Reject (input < V_{IDC} Min)	Figures 16b and 20		18	10	ns
17	Coll ±	t_{CPWO}	Collision Input Pulse Width to Turn–on (collision ± exceeds V_{IDC} Max)		26	18		ns
18	Coll ±	t_{CPWE}	Collision Input to Turn–off CLSN (input < V_{IDC} Max)		80	117		ns
19	Coll ±	t_{CPWN}	Collision Input to not Turn–off CLSN (input > V_{IDC} Min)			117	160	ns
20	CLSN	t_{CPH}	CLSN Turn–on Delay (V_{IDC} Max on collision ± to $CLSN_H$)	Figures 15, 16b, and 20		33	50	ns
21	CLSN	t_{CPO}	CLSN Turn–off Delay (V_{IDC} Min on collision ± to $CLSN_L$)			133	160	ns

TRANSMITTER SPECIFICATION

#	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
22	TCLK	t_{TCL}	TCLK Low TIme	t_{OSC} = 50ns Figures 17b and 18	45	50	55	ns
23	TCLK	t_{TCH}	TCLK High Time		45	50	55	ns
24	TCLK	t_{TCR}	TCLK Rise Time			2.5	8	ns
25	TCLK	t_{TCF}	TCLK Fall Time			2.5	8	ns
26	TX, TENA	t_{TDS}, t_{TES}	TX and TENA Setup Time	Figures 13, 14a, 14b, and 17b	5	1.1		ns
27	TX, TENA	t_{TDH}, t_{EH}	TX and TENA Hold Time		5	– 1.1		ns
28	TX ±	t_{TOCE}	Transmit ± Output, (bit cell center to edge)	Figures 14a, 14b, and 19	49.5	50	50.5	ns
29	TCLK	t_{OD}	TCLK high to Transmit ± Output			80	100	ns
30	TX ±	t_{TO}	Transmit ± Output Rise Time	20 through 80% Figure 19		2	4	ns
31	TX ±	t_{TOF}	Transmit ± Output Fall Time			2	4	ns
32	TX ±	V_{OD}	Undershoot Voltage at Zero Differential Point to Transmit Return to Zero (end of message)	Figure 14a			– 100	mV

Note : Assumes equal capacitance loading on RCLK and RX.

Figure 10 : Receiver Timing - Start of Packet.



* Pulse width of ≥ 45ns is always recognized. However, pulse width of ≥ 20ns is rejected.

Figure 11 : Receiver Timing - End of Packet (last bit = 0).

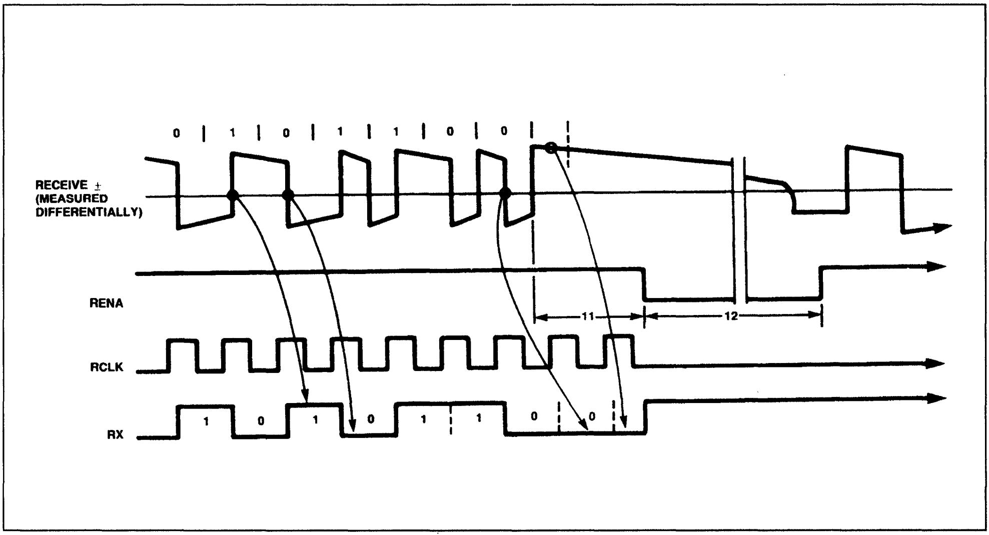


Figure 12 : Receiver Timing - End of Packet (last bit = 1).

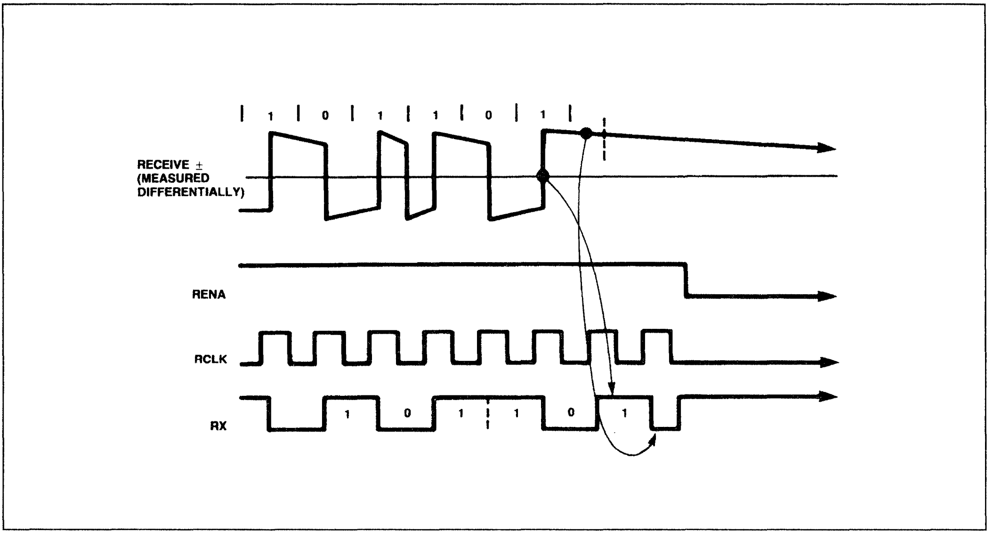


Figure 13 : Transmitter Timing - Start of Transmission (TSEL Low, TSEL High).

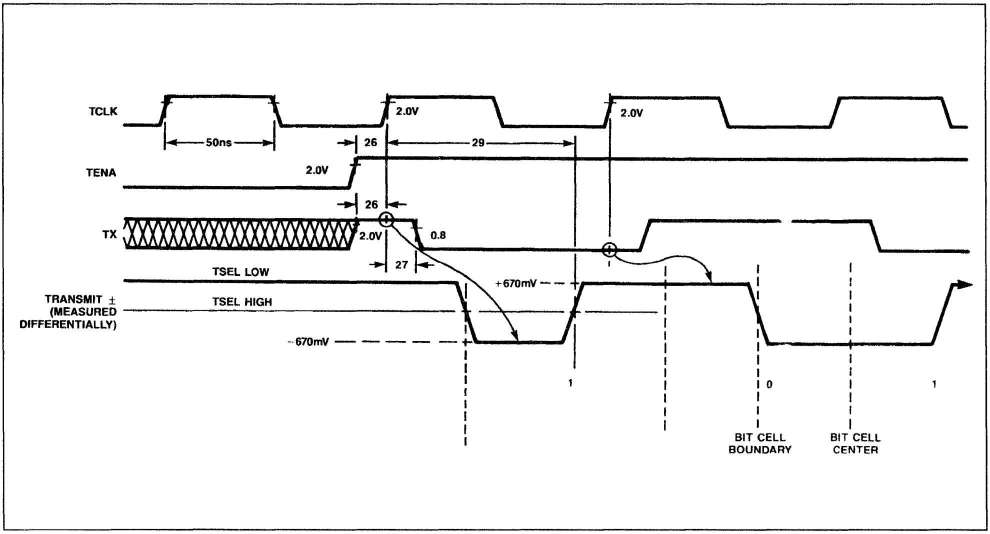


Figure 14a : Transmitter Timing - End of Transmission (TSEL High).

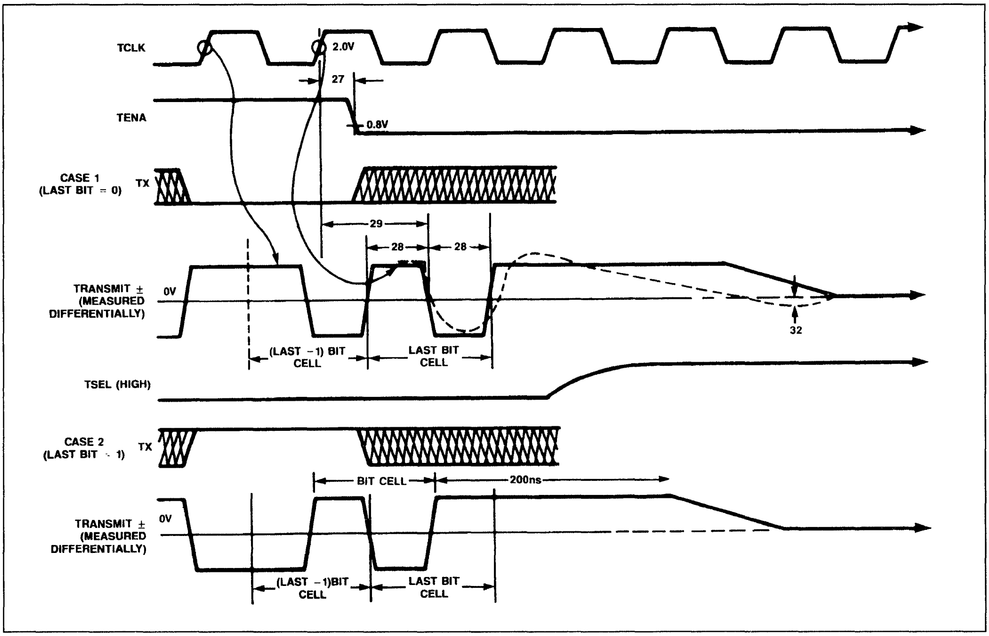


Figure 14b : Transmitter Timing - End of Transmission (TSEL Low).

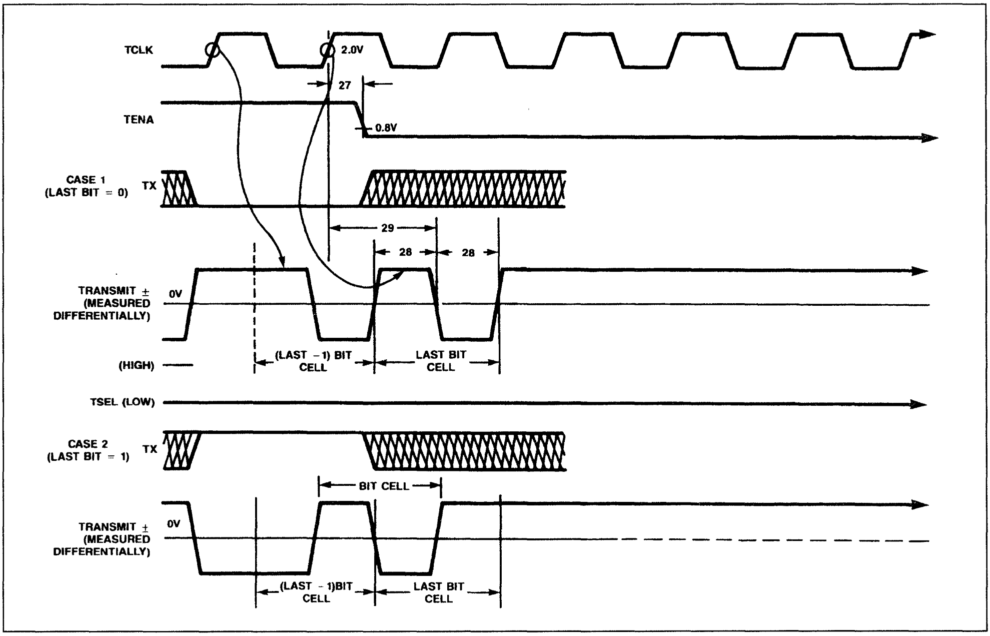
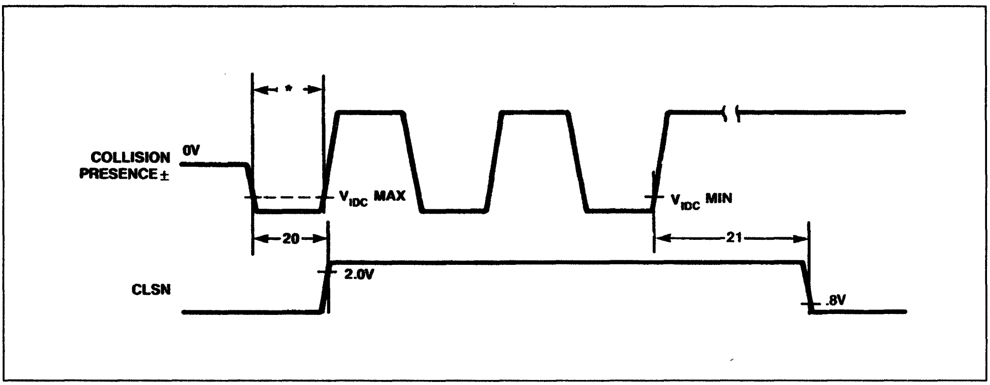


Figure 15 : Collision Timing.



* Pulse width of – 26ns is guaranteed to be recognized : however, pulse width of ≤10ns is rejected.

Figure 16a : Receive ± Input Pulse Width Timing

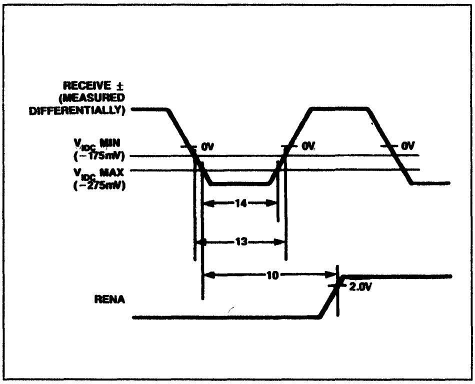


Figure 16b : Collision ± Input Pulse Width Timing.

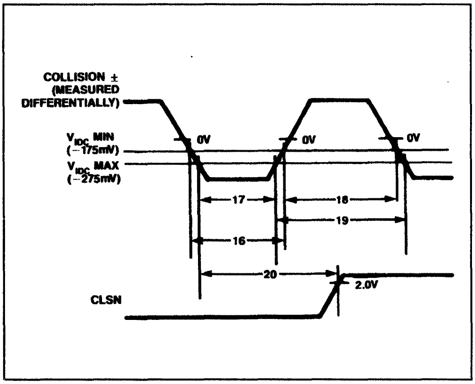


Figure 17a : RCLK and RX Timing.

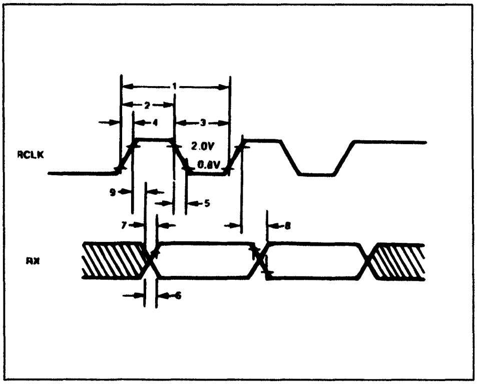


Figure 17b : TCLK and TX Timing.

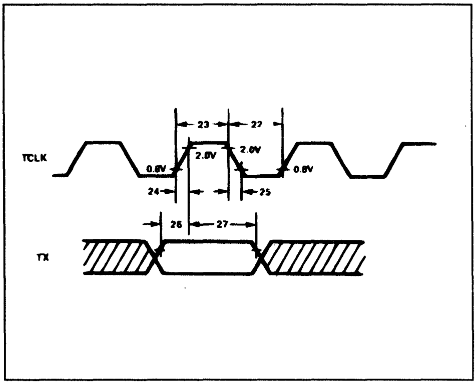


Figure 18 : Test Load for RX, RENA, and TCLK.

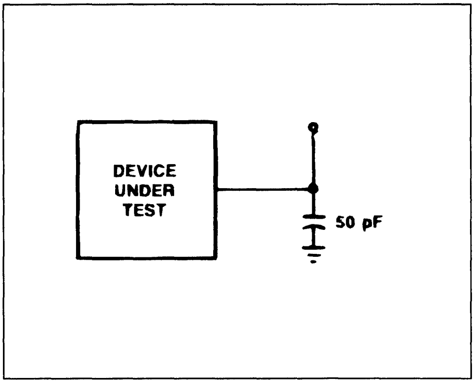


Figure 19 : Transmit ± Output Test Circuit.

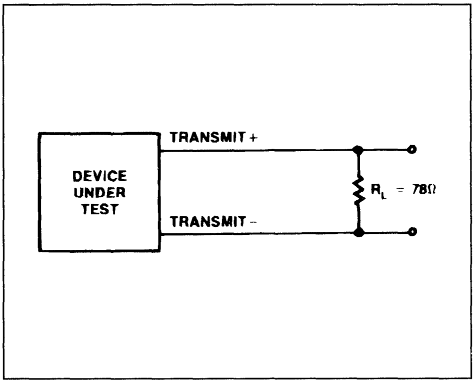
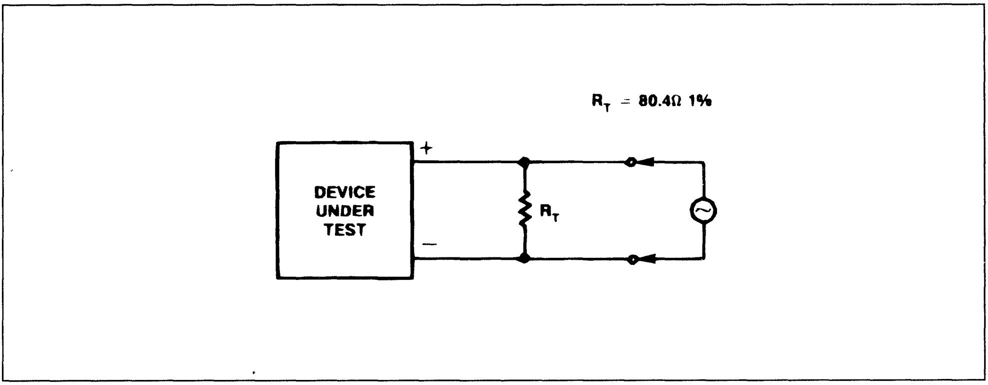
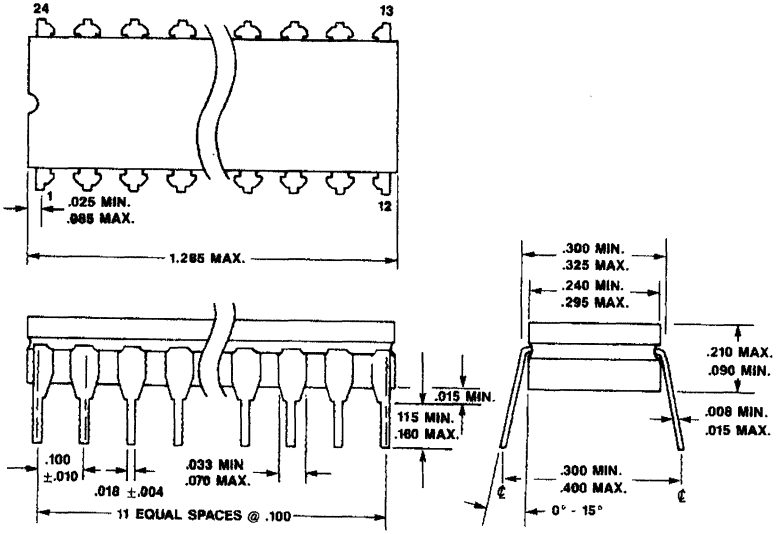


Figure 20 : Receive ± and Collision ± Input Test Circuit.



PACKAGE MECHANICAL DATA

24-Pin Cerdip Hermetic Package (J) - MK68592 (300 mil)



SGS-THOMSON MICROELECTRONICS

APPLICATION NOTE

CONVERTING TO THE MK5032 VLANCE FROM THE AM7990/MK68590 LANCE

INTRODUCTION

SGS-THOMSON Microelectronics currently provides two LAN Controllers to meet system designers' needs. These two LAN Controllers are the MK5032 Variable Bit-Rate LANCE and the MK68590 LANCE™. SGS-THOMSON Microelectronics is in the process of converting all new MK68590 designs to MK5032 designs. This application brief will discuss the easy conversion from the MK68590 to the MK5032. *Any future designs should use the MK5032 as the LAN Controller.*

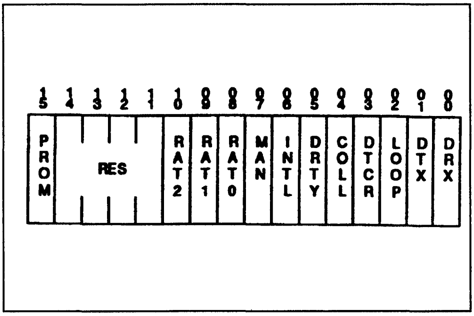
THE LAN CONTROLLERS

The MK5032 VLANCE and the MK68590 LANCE are IEEE 802.3 CSMA/CD LAN controllers designed to simplify the interfacing of a microcomputer or a minicomputer to a local area network. The MK5032 is fully compatible with Ethernet and StarLAN. The MK68590 is fully compatible with the Ethernet specification. These devices are identical in every facet except the MK5032 is a true variable bit-rate controller. Therefore, a discussion of their clock and data rate operation is warranted.

MK5032 CLOCK AND DATA RATE OPERATION

The MK5032 VLANCE supports system clocks from 1 to 10MHz and data rates from 1 to 10Mb/s. The big advantage of the MK5032 is that the data rate and system clock are independent. For example, the MK5032 can perform DMA transfers at 10MHz and transmit and receive data at 1Mb/s. This is accomplished because the MK5032 can divide the system clock rate to give the desired data rate. It has six divide-by modes - divide-by 1, 2, 4, 6, 8, and 10. The divisor is selected by programming bits 10 - 07 in the mode register in the initialization block. Figure 1 shows this 16-bit wide mode register.

Figure 1 : Mode Register for MK5032.



The bits involved in selecting the divisor are bits 10 - 07. Bits 10 - 08 are known as the 'RAT' (rate) bits. Bit 07 is known as the MAN (Manchester) bit. Table 1 shows how to configure these 'RAT' bits in order to select the desired divisor and thus desired data rate. For the MAN bit to be active, at least one of the 'RAT' Bits must be set to a '1'. When the MAN bit is set and at least one 'RAT' bit is set, the serial transmit data stream will be Manchester encoded data. (When using SGS-THOMSON Microelectronics Manchester encoder/decoders, the MAN bit should not be set). Consult the MK68590 Technical Manual for further details on the other bits in the mode register.

Table 1 : Configuration of "RAT" Bits for Divisor Selection.

RAT2	RAT1	RAT0	Divisor
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	6
1	0	0	8
1	0	1	10

LANCE is a trademark of SGS-THOMSON Microelectronics.

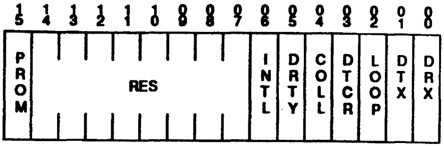
MK68590 CLOCK AND DATA RATE OPERATION

The MK68590 LANCE has dependent system clock and data rate operation. The speed at which the LANCE transmits and receives data is the same speed at which it does DMA transfers. The LANCE supports system clock rates from 1 to 10MHz and data rates from 1 to 10Mb/s. Designed for Ethernet, the LANCE performs at optimum efficiency with a 10MHz system clock and 10Mb/s data rate. In a StarLAN application, the LANCE performs DMA transfers at 1MHz and transmits and receives data at 1Mb/s.

SIMPLICITY OF CONVERSION

The mode register in the initialization block used for the MK68590 LANCE is shown in figure 2. It is identical to the mode register shown in figure 1 except bits 10 - 07 are reserved like bits 14 - 11. Since the MK68590 Technical Manual dictates these bits be set to zero for normal operation, the MK5032 can be "plugged-in" for the MK68590 and the MK5032 will be in Ethernet mode. Therefore, no software changes are necessary for the MK5032 in Ethernet mode given that the reserved bits have previously been written as '0's'.

Figure 2 : Mode Register for MK68590.



15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
PROM	RES								INTL	DRTY	COLL	DTCR	LOOP	DTX	DRX

CONCLUSION

With the multi-faceted similarity between the MK5032 VLANCE and the MK68590 LANCE, conversion from the MK68590 to the MK5032 is easy. With the MK5032, designers have a true variable bit-rate LANCE. This versatility enables users to design with one LAN controller to implement Ethernet or StarLAN.

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

MK5032 INTERFACE TO 68000 APPLICATION NOTE

TABLE OF CONTENTS

PARAGRAPH **TITLE** **PAGE**

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

PAGE

LIST OF ILLUSTRATIONS (continued)

1. INTRODUCTION

The VLANCE^{TM} Application Note describes the basic hardware and software needed to interface the MK5032 Variable Local Area Network Controller for Ethernet (VLANCE), to the 68000. It can be used as a "cookbook" for designing the basic hardware and software needed for a VLANCE-68000 interface, but is actually more useful as a design guide.

This publication contains seven sections in addition to this introduction. Section two is the Ethernet Primer, which contains background material and basic concepts involved in Ethernet communication. Section three defines terms used throughout the Application Note. Section four is an overview of the Variable Local Area Network Controller for Ethernet (VLANCE) chip. Section five is the Ethernet Node overview, listing design requirements for an intelligent Ethernet node. It deals with the block level concepts of both hardware and software. Section six is the Hardware Description. It deals with the basic hardware requirements and includes both block diagrams and schematics. Section seven describes the software used in the VLANCE-68000 interface. This section includes flowcharts, assembly code, and a step-by-step explanation of the software. The eighth and final section is a Assembly Code listing of the software needed for an intelligent Ethernet node.

The VLANCE-68000 hardware and software were designed as simply as possible. The design incorporates a minimum number of interface logic gates and no time multiplexing of devices. Structured assembly code with few subroutines and loops is used, making it easy to change and understand. The schematics and assembly code are presented in a format that easily lends itself to a self-tutorial on an existing Ethernet interface design for users. The design is intended to give users ideas on how to design or how to improve present designs. But, if so desired, users can apply the information directly to quickly generate a basic working interface.

This Application Note refers to many technical aspects of the VLANCE but they are not fully explained. For this reason, users should also study the MK5032 VLANCE Technical Manual.

2. ETHERNET PRIMER

Much debate and questioning as to the nature of celestial bodies existed in the early days of science. These questions included : What type of material exists between the heavenly bodies ? If it is not a vacuum, how does light move across it ? The conclusion agreed upon was that an Ether existed between the planets and it was the medium light used to travel from the sun to the Earth.

Xerox borrowed this terminology in their early stages of defining a local area network. Ether was the medium information would use to travel from one station to the next throughout the network specification. They called this network "Ethernet".

VLANCE is a trademark of SGS-THOMSON Microelectronics.

Ethernet was designed as a system for local communication between computing stations. A series of tapped coaxial cables between computing stations comprise a network that connects up to 1024 different stations per network. In addition, a gateway interface may be used to connect each network to other networks. The computing stations may consist of personal computers, CAD workstations, file storage devices, magnetic tape backup stations, large central computers, printers, plotters or any device that conforms to the Ethernet standard.

The objective of Ethernet was to provide a communication system that can grow with users' needs and accommodate several buildings in a local area. One of its purposes is to eliminate bottlenecks and reliability problems associated with a central controller.

Ethernet provides for one, and only one, connection between any two points on the network. Since the Ether, or the common broadcast communication channel, is passive when an active node fails, only its operation is affected - not the entire system.

Ethernet employs Carrier Sense Multiple Access with Collision Detect (CSMA/CD). The principle behind this operation is similar to human conversation. If several people are conversing while standing in a circle (assuming it is a non-controversial discussion between polite adults), only one person speaks at a time. When the first person finishes, someone else begins to speak and continues until he or she is also finished. If no one has anything else to say, silence falls over the group.

The same is true for Ethernet. Each node listens until the network becomes silent. Then, if a message needs to be transmitted the node broadcasts onto the Ether and the remaining nodes listen.

The more people in the circle, and the more they have to say, the greater the chance that two or more people will begin speaking simultaneously. If this happens in normal conversation, all the participants who have started speaking will stop and whoever begins to speak first will be able to say what he has to say. Two or more people may occasionally begin speaking again at the same time, in which case, they will again stop talking and, hopefully, the more gracious member of the group will wait. When a person starts talking, no one else begins until that person finishes. Delays and collisions occur in Ethernet just as they do in normal conversation. Due to the propagation delays encountered in the medium, a station may not sense network activity and begin transmitting. Once it begins, it keeps listening to the network ; if it detects a collision, it aborts the transmission and waits a certain amount of time determined by a random number generator.

In any given group of people, some are naturally talkers while others are listeners. If the conversation does not concern them, the listeners are less likely to pay attention. The same is true with Ethernet stations. Some stations, such as personal computers, transmit much information and others, such as printers, do much of the receiving.

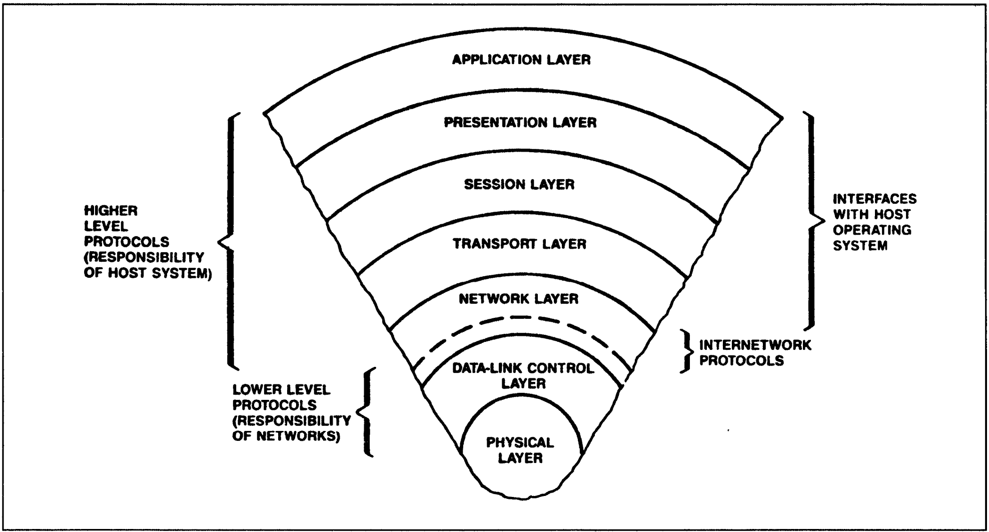
A particular station's Ethernet interface connects bit-serially through an interface to a transceiver that taps into the passive Ether. The Ethernet station, or node, broadcasts its message into the passive Ether, enabling all nodes on the network to hear it. All stations receive the message and determine if they are the desired destination. If the transmitting node's destination address matches that of the receiving node, the packet is accepted and the station digests the data. If the address does not match, the receiving node rejects the packet.

A station may use two addressing schemes when broadcasting. The first is the physical addressing scheme, whereby the transmitting station addresses one, and only one unique destination station. As long as all stations on the network are not in the promiscuous mode, only one station accepts the message.

The second addressing scheme uses a logical address. In this operation, stations receiving the message must determine if they are one of possibly many, intended recipients. An example of logical addressing is one in which all printers have the same logical address. If people want to send memos via the printers, they simply set the destination as the logical address - "printers". All printers on the network then see themselves as the destination and all accept the message. In addition, a station may set itself up in a promiscuous mode. In this mode, the station accepts all incoming messages, no matter what destination address the message has.

When Xerox defined Ethernet, their intent was to define a standard that all manufacturers who wanted access to the Ether, could use. They wanted to define a rigorous standard from the onset to avoid incompatibility. Universal acceptance of a standard is the very key to practical applications of local area networking. The International Standards Organization (ISO) has approved a layered protocol standard that specifies functions, as well as minimal rules for accessing these functions and for information exchange between devices on the network. This seven-layer architecture logically groups functions and provides conventions for connecting functions between layers. The model, shown in figure 1, is called the Open Systems Interconnection (OSI) network model.

Figure 1 : OSI Network Model.


APPLICATION LAYER
PRESENTATION LAYER
SESSION LAYER
TRANSPORT LAYER
NETWORK LAYER
DATA-LINK CONTROL LAYER
PHYSICAL LAYER
HIGHER LEVEL PROTOCOLS (RESPONSIBILITY OF HOST SYSTEM)
LOWER LEVEL PROTOCOLS (RESPONSIBILITY OF NETWORKS)
INTERFACES WITH HOST OPERATING SYSTEM
INTERNETWORK PROTOCOLS

The bottom three layers of the OSI model include the physical, data-link control, and network layers. Hardware is based on the actual definition of the two lower layers. Specifications in these layers include the transmission medium (Ether) and how the node must interface to the Ether. The physical and data-link control layers also specify how information should be formatted for error-free transmission and reception. Each layer supports another in hierarchial fashion. In other words, layer 1 serves layer 2, layer 2 serves layer 3 and so forth. The three bottom layers differ according to network architecture. The top three layers - session, presentation, and application - are the same for all networks. The transport layer is the interfacing layer between the top three and bottom three layers.

The functions of each layer of the OSI model follow.

Physical Layer :

- Handle cables, connectors and components
- Handle collision detection for CSMA/CD
- Handle voltages and electrical pulses

Data Link Control Layer :

- Make sure data is not mistaken for flags
- Add error checking algorithms
- Insert flags to indicate beginning and end of messages
- Provide access methods for local area networks

Network Layer :

- Internetworking
- Send control messages to peer layers about own status
- Set up routes for packets to travel (virtual circuit)
- May disassemble transport messages into packets and reassemble them at their destination
- Flow control
- Recognize message priorities and send messages in proper priority order
- Address network machines on the route through which the packets travel

Transport Layer :

- Multiplex end-user addresses onto network
- Monitor quality of service
- End-to-end error detection and recovery

- Address end user machines without concern for route of message or address machines in route between end user machines
- Possible disassemble and reassemble session messages
- Map address to names

Session Layer :
- Send information from one task to another
- Coordination and cooperation between end users tasks
- Start and stop tasks
- Dialog control
- Recovery from communication problems during a session without losing data

Presentation Layer :
- Encoding and decoding
- Data compaction
- Syntax transformation for character sets, text string, data display formats, graphics, file organization, data types

Application Layer :
- Log in
- Password checks
- Color control
- Graphics procedures
- Downline loading
- Creation of charts and displays
- File requests and file transfers
- Remote job entry
- Computer based message systems
- Job minipulation
- Virtual terminal service
- Data-based queries, insertions, and deletions
- User specific applications (e.g. editing, word processing, electronic funds transfer, airline reservation, and transaction processing)

The VLANCE, together with the SIA, Transceiver, and Coaxial cable satisfy the specifications in the bottom two layers of the ISO model. By adding the software shown in this Application Note, layers 1, 2, and 3a can be satisfied.

3. DEFINITION OF TERMS

DMA Capability : The ability to directly access memory or memory-mapped I/O. This includes reading, writing, and control signal generation.

Peripheral : A processing unit attached to the system bus which handles part of the processing load.

Bus Master : A CPU or DMA device that has gained control of the system bus. It can initiate data transfers on the bus by issuing an address and by driving the read/write, address strobe, and data strobe control signals.

Bus Slave : A device that decodes the address, read/write, address strobe, and data strobe control signals and responds accordingly for a read or write operation.

Bus Arbitration : In a system with more than one device capable of being the Bus Master, a bus arbitration convention must be employed to determine which device may take control of the system bus at any one time. Normally one of the Bus Master type devices is responsible for receiving and granting requests for access to the system bus.

Front End Processor : A processor microsystem, usually consisting of a microprocessor, or single-chip microcomputer, along with memory and control logic. This microsystem alleviates some of the burden placed on the main host processor. The front end processor acts as an intermediate stage of processing between the host processor and an I/O device.

Intelligent Ethernet Node : An Ethernet node that acts as a front end processor to the host processor. An intelligent Ethernet node would basically consist of microprocessor or microcomputer, memory, an Ethernet protocol device (VLANCE), transceiver interface device (SIA), and the associated firmware required to implement the lower layers of the Ethernet protocol.

4. VLANCE CHIP DESCRIPTION

The MK5032 VLANCE (Variable Bit Rate Local Area Network Controller for Ethernet) is a 48-pin VLSI device that simplifies interfacing a microcomputer or minicomputer to an Ethernet Local Area Network (LAN). This chip operates in a local environment that includes a closely coupled memory and microprocessor. The VLANCE uses scaled N-channel MOS technology and is compatible with several popular microprocessors. It interfaces to a microprocessor bus characterized by time-multiplexed address and data lines. Typically, data transfers are 16 bits wide, but byte transfers occur if the buffer memory address boundaries are odd. The address bus is 24 bits wide.

The Ethernet packet format consists of a 64-bit preamble, a 48-bit destination address, a 48-bit source address, a 16-bit type field, and a 46- to 1500-byte data field terminated with a 32-bit CRC (cyclic redundancy check) as shown in figures 2 and 3. The packet's variable widths accommodate both short status, command and terminal traffic packets, and long data packets to printers and disks (1024 byte disk sectors, for example). Packets are spaced a minimum of 9.6 usec apart to allow one node time enough to receive back-to-back packets.

Figure 2 : Ethernet and VLANCE Packet Format.

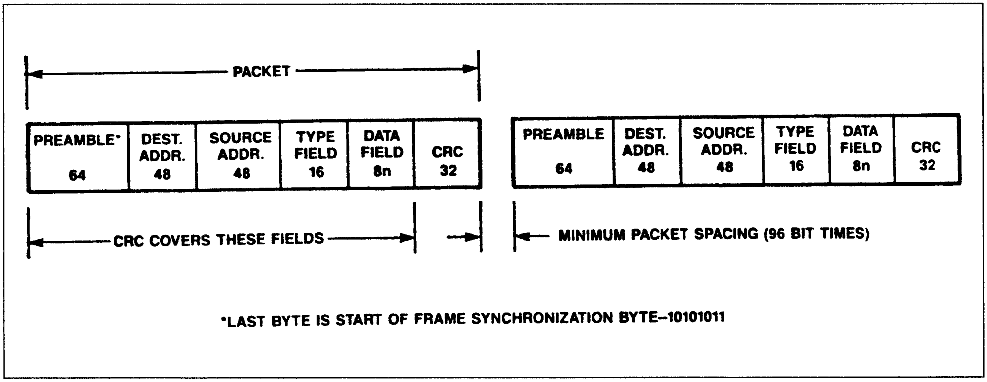
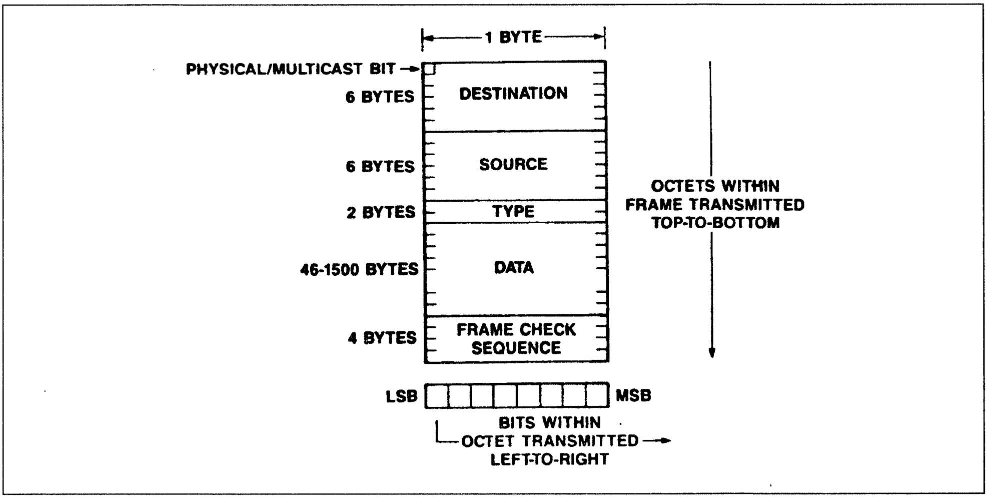


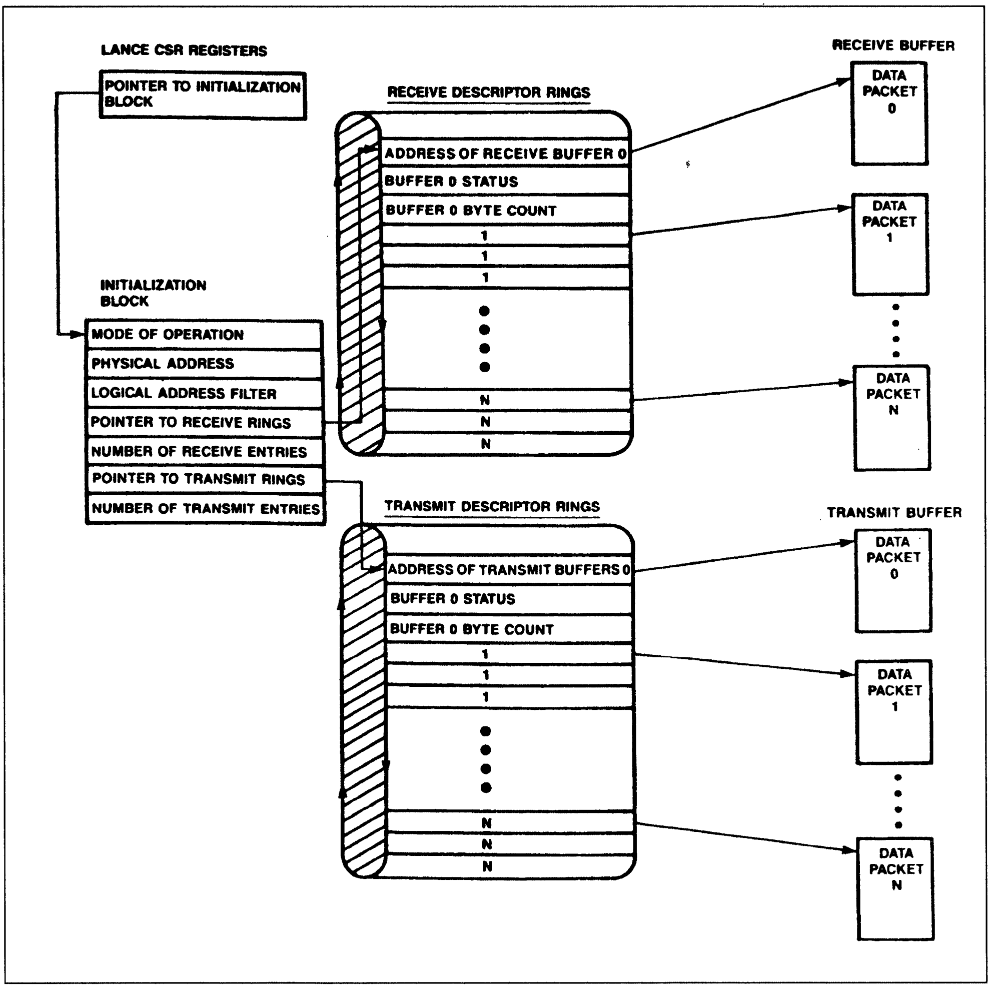
Figure 3 : Ethernet and Packet Bit Transmission Sequence.



The VLANCE operates in a minimal configuration that requires close coupling between local memory and a processor. The local memory provides packet buffering and is a communication link between the chip and processor. During initialization, the control processor loads the starting address of the initialization block plus the operation mode into the VLANCE via two control registers. The host processor talks directly to the VLANCE only during this initialization as a Bus Slave peripheral. The VLANCE's DMA machine, under microword control, handles all further communications.

The VLANCE on-chip DMA channel provides flexibility and speed by communicating with the host, or dedicated microprocessor, through common memory locations.Buffer management is organized by a circular queue of tasks in memory called "descriptor rings" (see figure 4). Separate descriptor rings describe transmit and receive operations. Up to 128 tasks may be queued on a descriptor ring for future execution. Each entry in a descriptor ring holds a pointer to a data memory buffer and an entry for the data buffer length. Data buffers can be chained or cascaded to handle a long packet in multiple data buffer areas. The VLANCE searches the descriptor rings in a "look-ahead manner" to determine the next empty buffer for chaining buffers together or handling back-to-back packets. As each buffer is filled, an "own" bit is reset, signaling the host processor to empty this buffer.

Figure 4 : VLANCE Memory Management.



5. ETHERNET NODE OVERVIEW

5.1. INTRODUCTION

The VLANCE has two basic types of micro-interfaces. The first is when the VLANCE acts as a peripheral to a host processor and the second is when the VLANCE and a dedicated microprocessor or microcontroller work together to form an intelligent node. In the second type, all processing between the processor and the VLANCE occurs on a local bus, while the entire node interfaces to the main system bus as an intelligent subsystem.

The VLANCE parallel interface is designed to be an easy or "friendly" interface to several popular 16-bit microprocessors. This Application Note addresses MK68000 interface requirements, but the concepts can be applied to other microprocessors.

5.2. OPERATIONAL DESCRIPTION

The following two sections are operational descriptions of a typical intelligent Ethernet node controller design, and the Ethernet node controller design used to generate this Application Note.

5.2.1. TYPICAL ETHERNET NODE. An intelligent Ethernet node's primary function is to unburden the host processor from many networking tasks and, at the same time, reduce system bus congestion by eliminating the need for the VLANCE to ever access the system bus.

A typical intelligent node performs node initialization and self tests, as well as transmitting and receiving messages. It also retains statistical records of error-causing conditions and generates time-out interrupts. These timeout interrupts are used for both memory refresh and upper-level protocol requirements.

The software of the intelligent node isolates the higher level user software from node details, such as memory refresh and the VLANCE interface.

In a typical system, an intelligent Ethernet controller may be designed and placed on a separate system board. The VLANCE chip permits placement of the entire Ethernet controller on a single system board because it reduces the chip count. This board would interface to a particular bus structure, i.e., the VME, Versabus^{TM}, Multibus^{TM}, etc.

The system host processor may interface to the node processor via a dual-ported memory with the use of semiphores.

5.2.2. APPLICATION-NOTE ETHERNET NODE. The Ethernet node controller design discussed in this Application Note does not have the characteristics of a typical intelligent Ethernet controller. It was designed more as a demonstration and teaching tool. The only software written for the node is the lower level software included in the Appendix.

This design uses a total of 8K bytes of memory. A greater quantity of memory would be present in a typical intelligent Ethernet node design. This design has no memory refresh requirements since static RAMs are used instead of dynamic RAMs.

This design includes a "message send request" button that users depress to send data messages to any desired node. The push button simulates a message request that the system host processor would normally generate. The button idea is only used on the breadboard for controlled message transfer for debug and demonstration purposes.

Depressing the button interrupts the local processor and calls a message routine. This message routine generates a pseudo message and updates the transmit descriptor so proper transmission can occur. Immediately after pushing the button, users can examine the memory to verify message transmission and proper hardware and software operation. The pushbutton was felt to be the best way to demonstrate message generation.

Aside from these few differences, the Ethernet node design used in this publication is functionally the same as a typical design used in industry.

5.3. HARDWARE REQUIREMENTS

As the Bus Master, the VLANCE has a wide 24-bit linear address space it can access directly via DMA. This 24-bit address bus interfaces directly to the MK68000's 24-bit bus. The only provision is that latches must be placed on the multiplexed address/data bus to latch up the address at the start of a read or write cycle when the VLANCE is the Bus Master. Only the lower 15 bits must be latched up ; the upper eight bits can be interfaced directly. When interfacing to the MK68000, BM1 and BM0 correspond to upper data strobe (UDS) and lower data strobe (LDS). Address bit A0 is not used in the basic MK68000 interface.

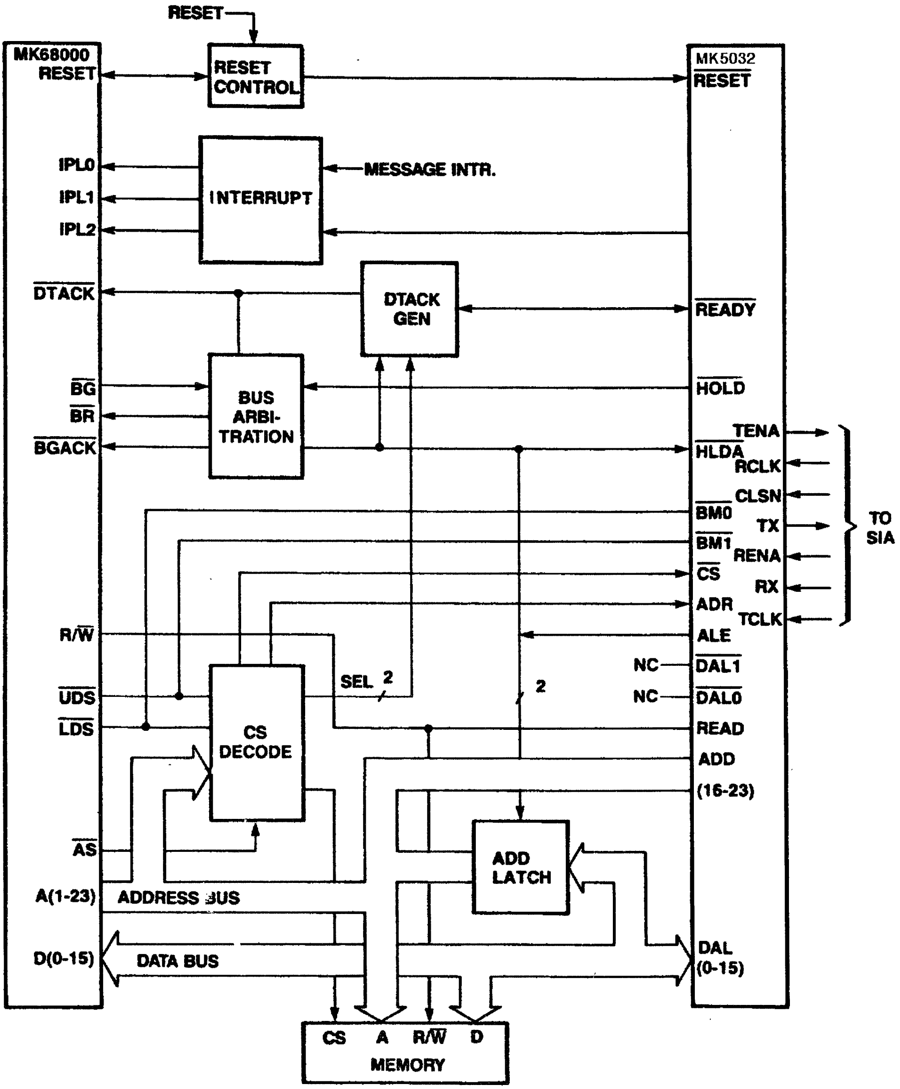
The basic blocks of this design include the VLANCE, MK68000, memory, bus arbitration circuitry, chip select circuitry, and DTACK circuitry. Figure 5 is a block diagram of this interface.

The memory includes 8K bytes of random access memory (RAM) and 4K bytes of erasable programmable read only memory (EPROM). The EPROM contains the Ethernet node program, while the RAM is used for receive and transmit rings as well as stack area.

Section 6 - Hardware Description gives a detailed hardware description.

Versabus is a trademark of Motorola, Inc.
Multibus is a trademark of INTEL Corp.

Figure 5 : 68000/5032 Block Diagram.



5.4. SOFTWARE REQUIREMENTS

The software program has three functions :

1. Initialize the VLANCE
2. Perform message ring management
3. Keep track of the error- and flag-causing conditions.

The software acts as an intermediate stage between the higher-level software protocol of the system host processor and the lower-level protocol implemented by the VLANCE. Figure 6 shows the main flow of the software.

The initialization and diagnostics are implemented upon powerup. Users may set up the diagnostics submodule to be implemented any time they desire an operational check. The two service modules - VLANCE Interrupt and Message Interrupt - are both interrupt-driven.

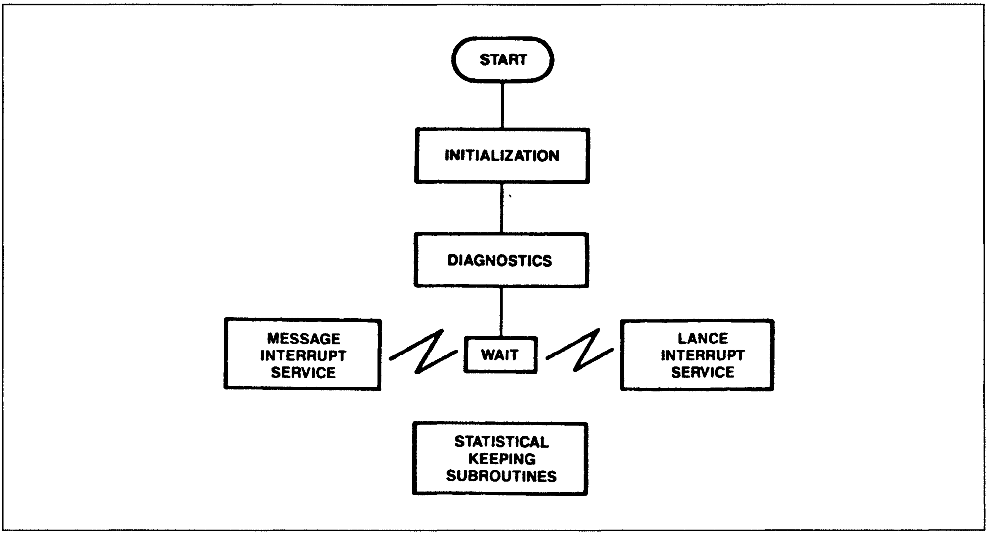
The VLANCE Interrupt service routine is called when the VLANCE interrupts the local processor. The VLANCE interrupts the processor when a message is received or transmitted, initialization is complete, or an error has occurred. The error- or flag-causing condition is determined by reading the Control and Status Registers zero (CSR0). The processor then either services the flag-raising condition or calls one of the statistical-keeping subroutines. It then simply reports the condition to the system processor.

It is not necessary for the VLANCE to operate on an interrupt-driven basis. The processor may be programmed to periodically poll the Control and Status Registers zero (CSR0) for flag-causing conditions. This Application Note performs the function on an interrupt basis to eliminate the need for a timer.

The message interrupt service routine is called when users depress the "message request" button. The routine generates a pseudo message depending on what parameters users give it. It then writes the message into a transmit message buffer and updates the transmit message descriptor.

As stated before, the "Push-button" operation is not included in a typical Ethernet node processor, but appears here to give the reader possible suggestions on how to write required software for servicing an actual system-generated message.

Figure 6 : Program Software Overview Flowgraph.



5.5. SYSTEM MEMORY MAP

All software activities take place in the low end of memory. The entire program uses less than 2K bytes of memory. Figure 7 is a memory map of the space this program uses.

Memory area from hex address $0000 to $03FF is reserved for vectors. The program resides in the memory space bounded by addresses $0400 and $1FFF. Message management memory space is between $2000 and $387B. This area contains the initialization block and the receive and transmit message descriptors, along with the receive and transmit buffers. The memory space between addresses $387C and $3FFF is the utility area. The program stack, ring management, and error flags reside here. Figure 8 gives a detailed map of the utility area. Memory location $4000 is the address of the VLANCE's register data port (RDP). Memory location $4002 is the address of the VLANCE's register address port (RAP).

The receive and transmit ring length, as well as their buffer sizes, are all variable. These variables may be altered by changing an equate statement at the beginning of the program. This Application Note uses eight receive descriptors and four transmit buffers. The size for both transmit and receive buffers is 256 bytes. There are twice as many receive buffers as transmit buffers because it is undesirable for the node to miss an incoming packet due to a lack of receive buffers.

Figure 7 : System Memory Map.

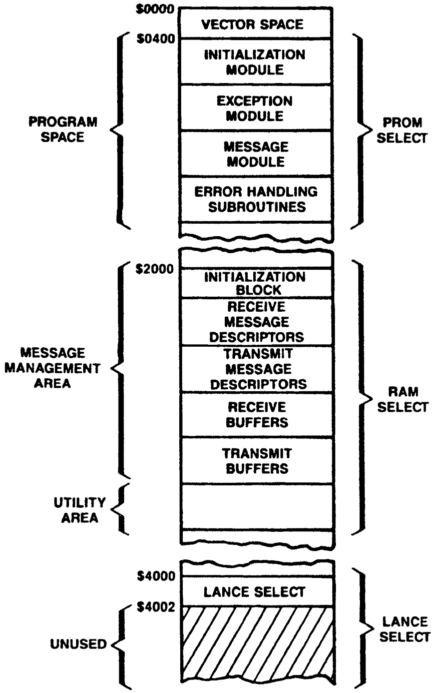
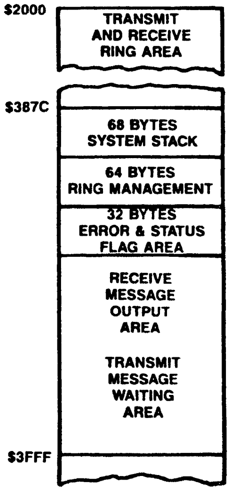


Figure 8 : Utility Area Memory Map.


$2000
TRANSMIT AND RECEIVE RING AREA
$387C
68 BYTES SYSTEM STACK
64 BYTES RING MANAGEMENT
32 BYTES ERROR & STATUS FLAG AREA
RECEIVE MESSAGE OUTPUT AREA
TRANSMIT MESSAGE WAITING AREA
$3FFF

5.6. UTILITY AREA DEFINITION

The utility area is composed of the following :

1. 68 bytes of system stack
2. 64 bytes of ring management variables
3. 32 bytes of error and status flag area
4. 620 bytes for message area.

The message storage area is to be used for receive message output and transmit message waiting area. A ring management stack or status area is set aside in memory to record all transmit and receive descriptor ring activity. Figure 9 gives the address location and describes the information in that memory location.

The addressing mode to access all ring management locations is "Address Register Indirect with Displacement". Base address $38C0 is placed in register A4 upon initialization. The displacements are all defined in the equate statements at the beginning of module number one (see software listing in the appendix).

The ring management area has several pointers. Some point to the top and bottom of the descriptor rings. Others point to the next-descriptor to be used and the last-descriptor used. The complicated task of message management is controlled by using these pointers and other status information in the ring management area.

Figure 9 : Ring Management Area Description.

ADDRESS **DESCRIPTION OF CONTENTS**

38C0 RECEIVE RING LENGTH. Number of entries in the receive ring.

38C2 TRANSMIT RING LENGTH. Number of entries in the transmit ring.

38C4 RECEIVE RING BASE ADDRESS POINTER. This points to the first receive message descriptor.

38C8 TRANSMIT RING BASE ADDRESS POINTER. This points to the first transmit message descriptor.

38CC RECEIVE RING BOTTOM ADDRESS POINTER. This points to the last receive message descriptor.

38D0 TRANSMIT RING BOTTOM ADDRESS POINTER. This points to the last transmit message descriptor.

38D4 LAST RECEIVE DESCRIPTOR USED POINTER. This points to the last receive descriptor to be used. When the VLANCE interrupts the host because of a received message, this pointer will indicate which ring corresponds to the message. If more than one buffer was required for the incoming message, than one buffer was required for the incoming message, this pointer will point to the first descriptor used.

38D8 LAST TRANSMIT DESCRIPTOR USED POINTER. This points to the last transmit descriptor turned over the host by VLANCE. When VLANCE interrupts the host because it has just transmitted a message, this pointer will indicate the descriptor just used.

38DC NEXT TRANSMIT DESCRIPTOR TO BE USED POINTER. This points to the next transmit message descriptor available for use. The "number of rings available" must be checked before the next ring is accessed. If there are "0" rings available the Next Transmit Pointer will be pointing to a ring which has not been serviced by VLANCE.

38E0 LOOPBACK MESSAGE COUNT. This contains the number of bytes contained in the loopback message. This is not used in diagnostics.

38E2 TRANSMIT DESCRIPTOR COUNT number of descriptors that are available.

38E4 RING MANAGEMENT STATUS.

BIT #1 "1" indicates that VLANCE is in LOOPBACK Mode.
BIT #2 "1" indicates that the START OF PACKET bit was set before.
BIT #3 "1" indicates that the END OF PACKET bit was set before.
BIT #4 "1" indicates that the program is in DIAGNOSTIC Mode.
BIT #5 "1" indicates that this portion of the TEST IS COMPLETE.
BIT #6 NOT USED.
BIT #7 NOT USED.

38E8 LOOPBACK TRANSMIT BUFFER ADDRESS POINTER.

38EC MORE COUNTER keeps count of the number of times more than one retry was needed to transmit a packet.

38EE ONE COUNTER keeps count of the number of times exactly one retry was needed to transmit a packet.

38F0 MESSAGE WORD used for testing and debugging.

6. HARDWARE DESCRIPTION

6.1. INTRODUCTION

The circuit described in this Application Note has a minimal amount of logic for interfacing the 68000 to the VLANCE. Figure 5 is a block diagram of the circuit.

The basic interface blocks consist of the following :

1. Chip select decode and DTACK (Data Transfer Acknowledge) generation
2. Address/Data bus interface
3. Interrupt - autovector
4. Bus arbitration blocks.

Detailed schematics of these blocks are given in figures 10 through 16.

6.2. CHIP SELECT DECODE CIRCUITRY

The Chip Select Decode circuit consists of five logic gates and a Bipolar Programmable Read Only Memory (PROM) as shown in figure 10. The circuit output enables both the RAM and EPROM, as well as enabling the VLANCE. When the VLANCE is the Bus Slave and the 68000 needs to access one of its control-status registers, the VLANCE is enabled. As Bus Masters, either the 68000 or the VLANCE can access memory. The DTACK circuitry uses the output signals, RAM SELECT and ROM SELECT. They generate the needed DTACK and READY signals for the 68000 and the VLANCE, respectively.

Figure 11 describes chip select decoding. Since the entire addressable memory used in this circuit resides below hex address $7FFF, address bits A15 thru A23 are not used. These address bits are all inputs to the OR-AND gate array. This array's output is active if address bits A15 through A23 are all at logic "0". The decode circuit's output enables different segments of memory. RAM HH enables high-address, high-word of RAM, while RAM HL will enable the high-address, low-word of RAM. RAM LH enables the low-address, high-word of RAM, and so forth. Figure 12 is a chip select decode PROM firmware map.

Figure 10 : Chip Select Decode and DTACK Circuit.

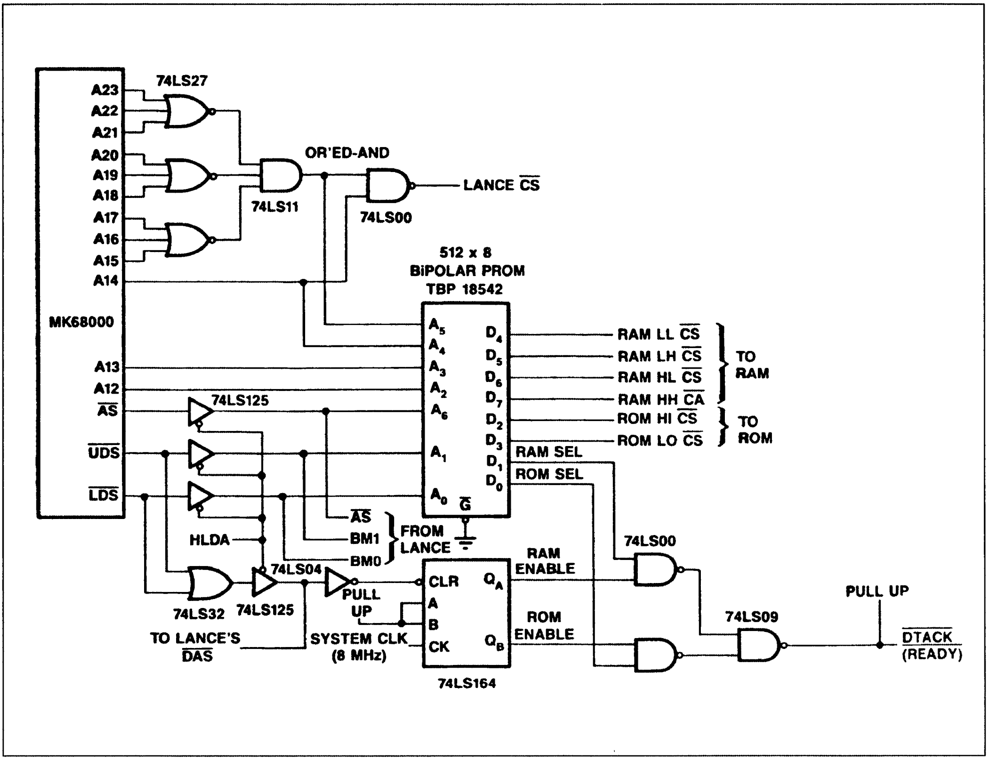
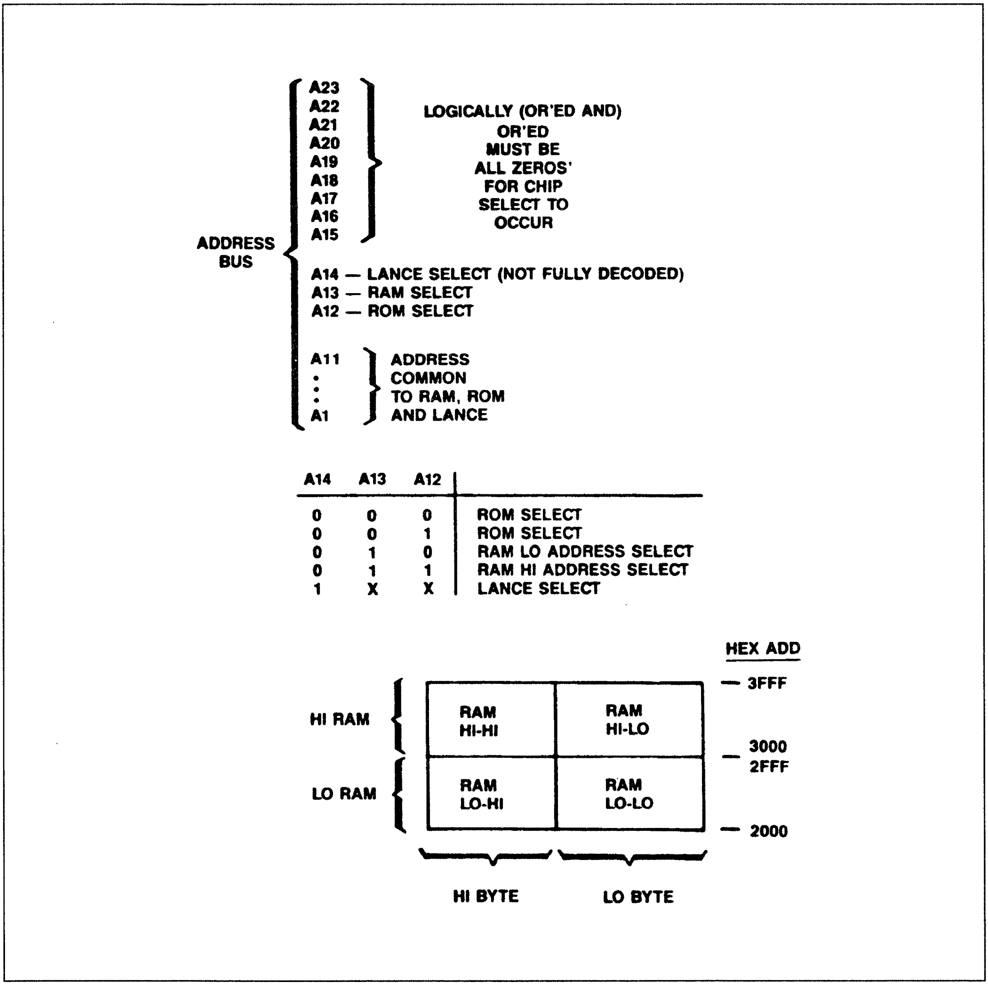


Figure 11 : Chip Select Decode Description.



A14	A13	A12	
0	0	0	ROM SELECT
0	0	1	ROM SELECT
0	1	0	RAM LO ADDRESS SELECT
0	1	1	RAM HI ADDRESS SELECT
1	X	X	LANCE SELECT

Figure 12 : Chip Select Decode PROM Firmware.

AS/	ORED AND	A14	A13	A12	UDS/	LDS/		RAM HH CS/	RAM HL CS/	RAM LH CS/	RAM LL CS/	ROM LO CS/	ROM HI CS/	ROM SEL.	RAM SEL.		
A_6	A_5	A_4	A_3	A_2	A_1	A_0	HEX ADD	D_2	D_6	D_5	D_4	D_3	D_2	D_1	D_0	HEX DAT	
0	1	0	0	0	0	0	20	1	1	1	1	0	0	1	0	F2	ROM Word Select
0	1	0	0	0	0	1	21	1	1	1	1	1	0	1	0	FA	ROM HI Byte Select
0	1	0	0	0	1	0	22	1	1	1	1	0	1	1	0	F6	ROM LO Byte Select
0	1	0	0	0	1	1	23	1	1	1	1	1	1	0	0	FC	Invalid (no select)
0	1	0	0	1	0	0	24	1	1	1	1	0	0	1	0	F2	ROM Word Select
0	1	0	0	1	0	1	25	1	1	1	1	1	0	1	0	FA	ROM HI Byte Select
0	1	0	0	1	1	0	26	1	1	1	1	0	1	1	0	F6	ROM LO Byte Select
0	1	0	0	1	1	1	27	1	1	1	1	1	1	0	0	FC	Invalid (no select)
0	1	0	1	0	0	0	28	1	1	0	0	1	1	0	1	CD	Low Address/Word Select
0	1	0	1	0	0	1	29	1	1	0	1	1	1	0	1	DD	Low Address/HI Byte Select
0	1	0	1	0	1	0	2A	1	1	1	0	1	1	0	1	ED	Low Address/LO Byte Select
0	1	0	1	0	1	1	2B	1	1	1	1	1	1	0	0	FC	Invalid (no select)
0	1	0	1	1	0	0	2C	0	0	1	1	1	1	0	1	3D	HI Address/Word Select
0	1	0	1	1	0	1	2D	0	1	1	1	1	1	0	1	7D	HI Address/HI Byte Select
0	1	0	1	1	1	0	2E	1	0	1	1	1	1	0	1	BD	HI Address LO Byte Select
0	1	0	1	1	1	1	2F	1	1	1	1	1	1	0	0	FC	Invalid (no select)
1	X	X	X	X	X	X		1	1	1	1	1	1	0	0	FC	Invalid (no select)
X	X	X	X	X	X	X		1	1	1	1	1	1	0	0	FC	Invalid (no select)

6.3. DTACK CIRCUITRY

A few logic gates and a shift register make up the DTACK circuitry (see figure 10). The shift register allows selectable delay time outputs to compensate for varying memory access speeds.

When the VLANCE is a Bus Master, READY is an asynchronous acknowledgement from the DTACK circuit that memory will accept data in a WRITE cycle, or that memory has put data on the Data/Address lines in a read cycle.

As a Bus Slave, the VLANCE asserts READY when it has put data on the Data/Address lines during a READ cycle or, is about to take data off the Data/Address lines during a WRITE cycle. Ready is a response to Data Address Strobe (DAS) and is negated after DAS is negated. The separate DTACK signals are wire ORed with the READY signal from the VLANCE. In this manner, the READY signal can be either an input or output. When the VLANCE is the Bus Master, the DTACK input on the 68000 is in the high impedance state.

6.4. ADDRESS/DATA BUS INTERFACE

Figures 13 and 14 are schematics of address and data bus interfacing. The interfacing is very straightforward. The data bus is connected directly between the 68000, the VLANCE, and memory. Address bits A16 through A23 are connected up directly between the VLANCE and the 68000. Since the VLANCE has a multiplexed Address/Data bus, tri-state latches must be placed on the DAL lines. These latches store the address during the beginning of a read or write cycle when the VLANCE is a Bus Master. HLDA enables the output of the latches, this occurs only when the VLANCE is a Bus Master.

Figure 13 : 68000-Memory Interface Circuit.

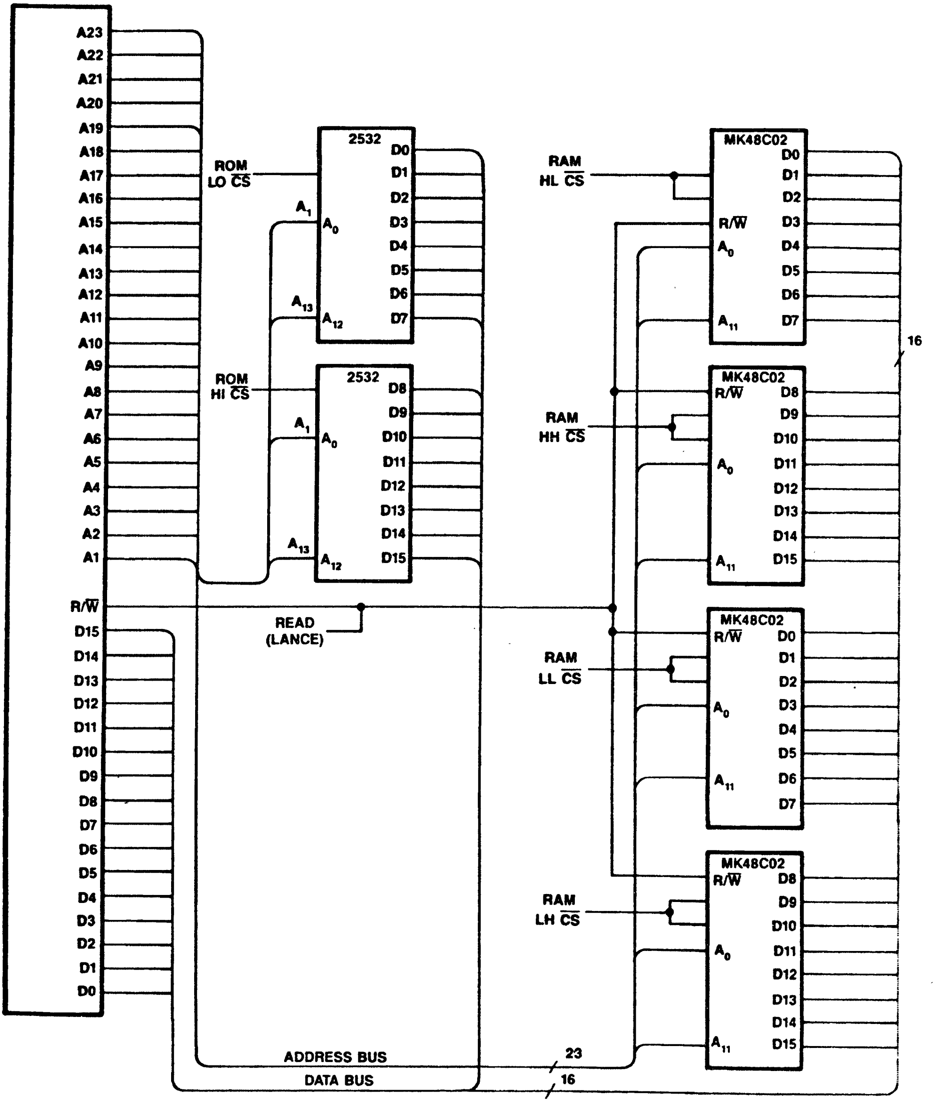
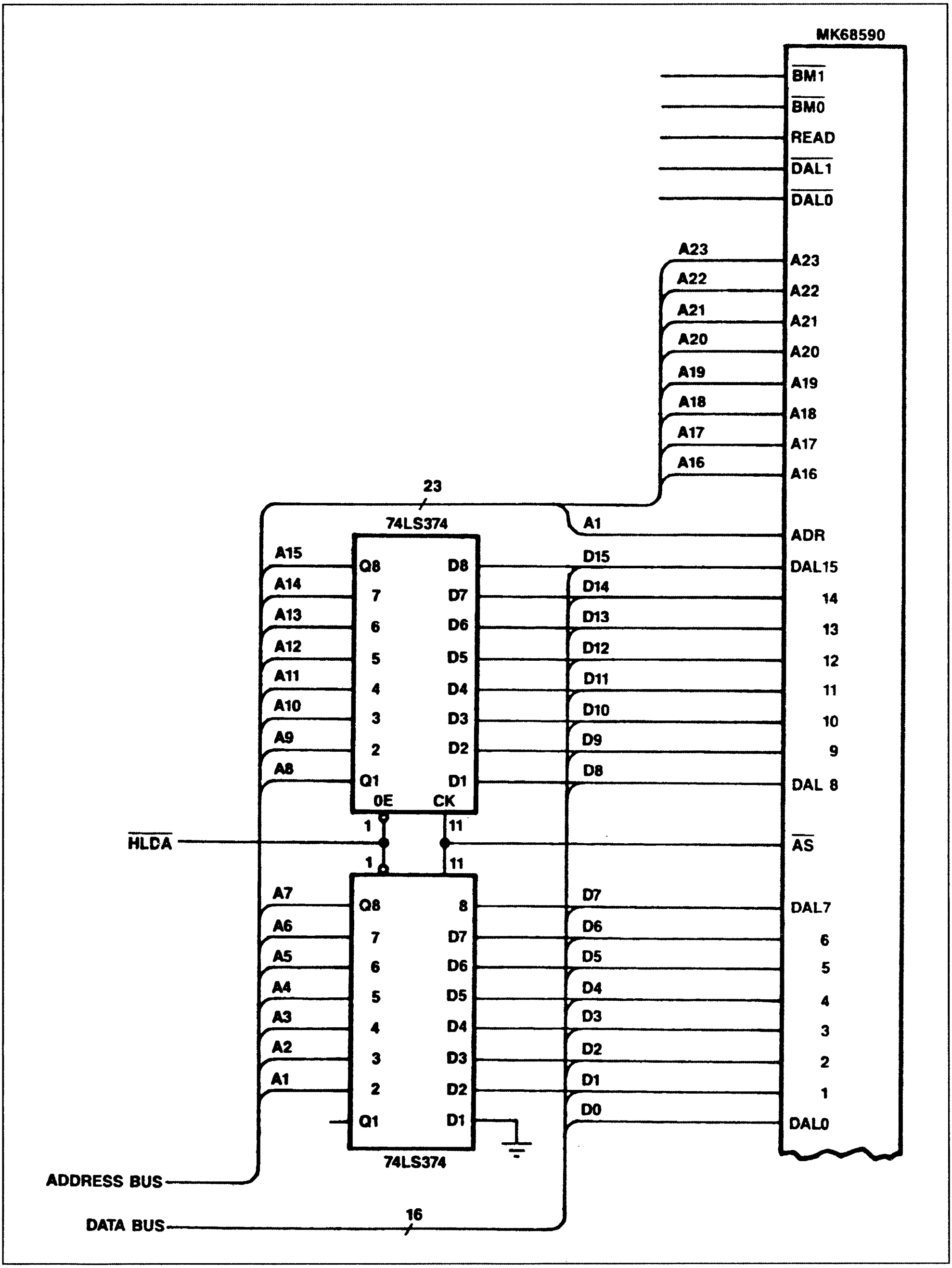


Figure 14 : Address/Data Bus Interface Circuit.



6.5. INTERRUPT CIRCUITRY

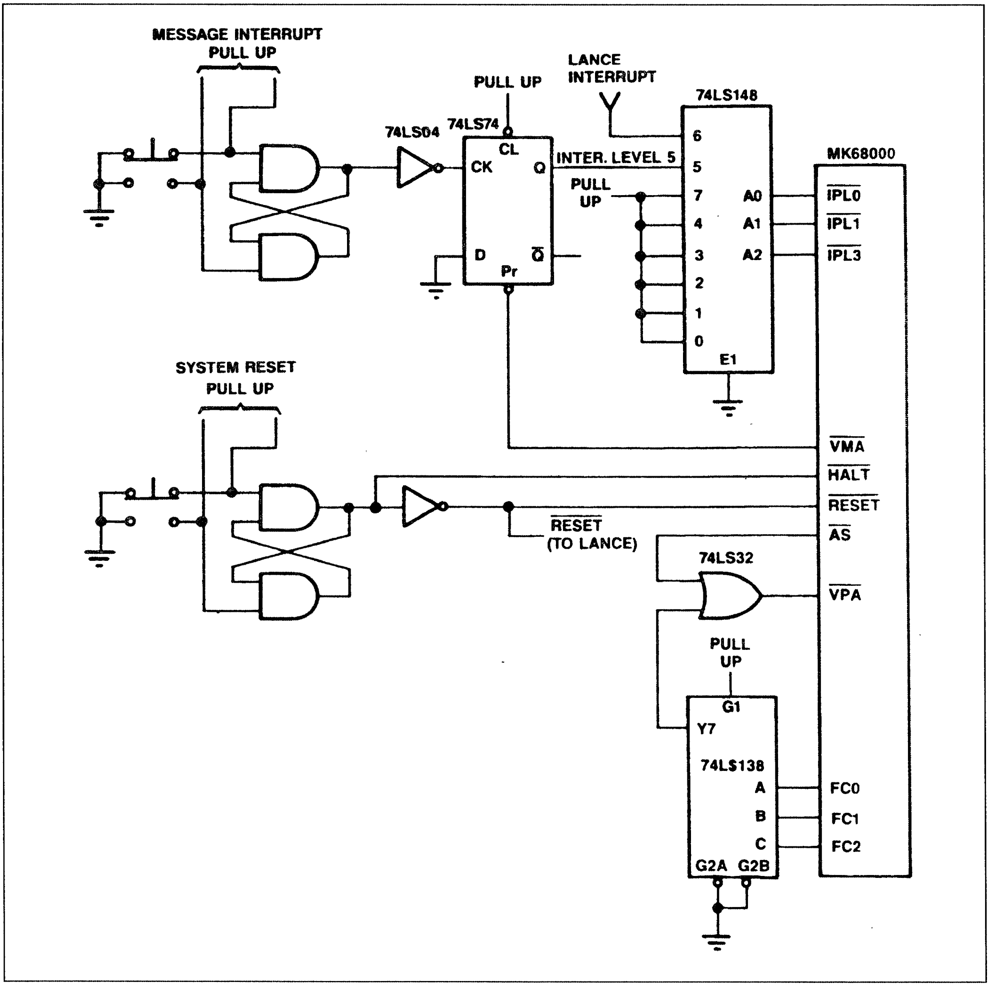
The interrupt circuitry, as shown in figure 15, handles incoming interrupts to the 68000, and also handles the interrupt acknowledgement scheme. This circuit has two interrupt sources. One can come from the VLANCE, the other can come from a push button on the breadboard. The push button informs the 68000 that a message is ready to be transmitted.

6.6. AUTOVECTORING CIRCUIT

Interrupts in this program are handled with autovectoring. The interrupt from the VLANCE is set to interrupt level six, and the interrupt that represents a "message ready to be transferred" from the system host is set at interrupt level five. Level seven interrupt is the non-maskable interrupt, and is not used in this circuit.

Upon an interrupt, the 68000 responds with binary "111'" at the Function Code Output (FC0-2). This is decoded and Valid Peripheral Acknowledge (VPA) is generated from it, which tells the 68000 to autovector (see the 68000 users manual for more details on autovectoring).

Figure 15 : Interrupt and Auto Vector Circuit.



6.7. BUS ARBITRATION CIRCUIT

The Bus Arbitration circuit, as shown in figure 16, generates the necessary control and handshake signals needed for the VLANCE to take control of the bus. The 68000 has three handshake signals for bus arbitration, while the VLANCE has only two. When the VLANCE requires control of the bus, it asserts $\overline{HOLD}$. The 68000 returns a Bus Grant ($\overline{BG}$) signal. $\overline{BG}$ along with $\overline{DTACK}$ and $\overline{AS}$ generate hold acknowledge HLDA and Bus grant acknowledge ($\overline{BGACK}$). $\overline{DTACK}$ and $\overline{AS}$ are needed to verify that the previous cycle is over before the VLANCE takes control of the bus. A timing diagram of the bus arbitration process is shown in figure 17.

Figure 16 : Bus Arbitration Circuit.

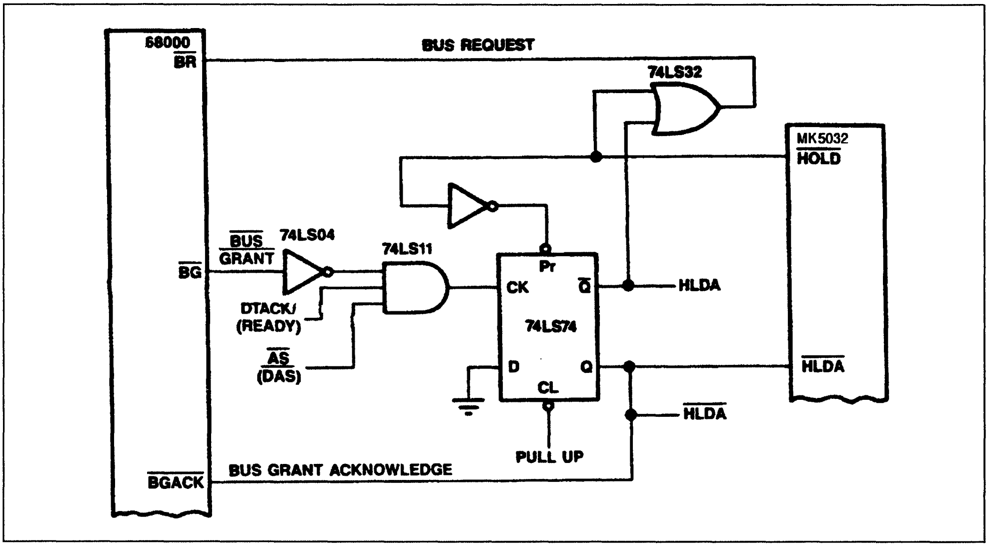
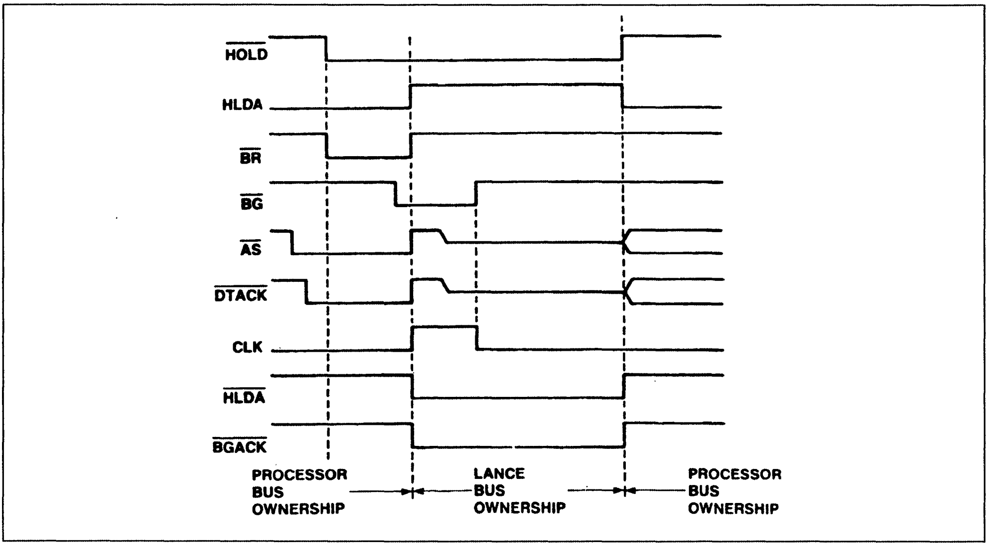


Figure 17 : Bus Arbitration Timing.



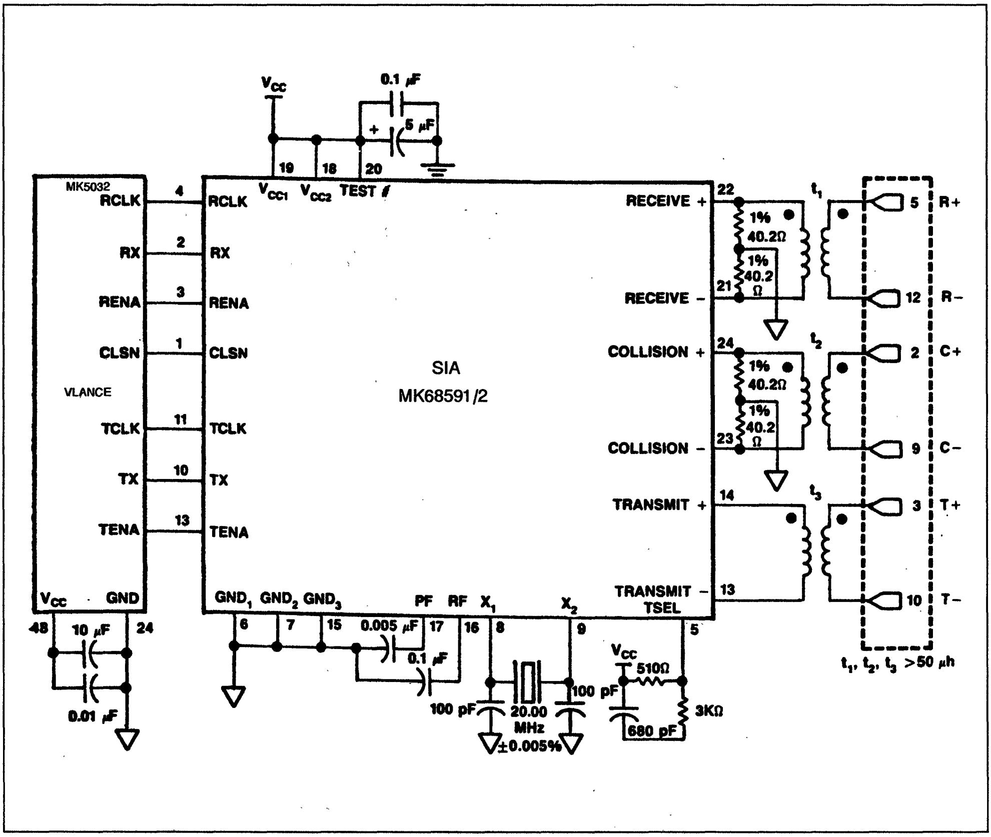
6.8. SIA-VLANCE INTERCONNECT

Users must carefully plan the interconnect layout between SIA, the VLANCE, and the output connector to the transceiver. The SIA is a very fast ECL interface device. It has an internal voltage controlled oscillator (VCO) that generates a 40MHz clock from which internal timing is derived. This device is very susceptible to noise. For this reason, the proximity of the SIA to the VLANCE and output connector is important. Very close coupling between all filter and terminating devices is required. They must be mounted directly adjacent to the SIA pins in printed circuit boards and must be attached directly to the socket pin in a breadboard situation.

Only low-profile sockets should be used for the SIA in a breadboard situation. ZIP-DIPs, or high-profile sockets result in an environment too noisy for the fast communication rates these devices generate.

Decoupling is also very important when building a breadboard or laying out a PC board. The Ground and Vcc pins (pins 1 and 48) of the VLANCE should be decoupled to reduce noise possibilities. These capacitors should be attached directly to the socket pins in a breadboard situation. Figure 18 gives suggested filter values to be used on the SIA.

Figure 18 : SIA Filter Values.



7. SOFTWARE DESCRIPTION

7.1. INTRODUCTION

The software needed to generate a basic interface between the VLANCE and the 68000 is described in three different formats : a written description of the step-by-step process, flow charts of the individual submodules, and a printout of the actual assembly code. The assembly code appears in Appendices A, B, and C.

Users should read the software description along with studying the flow charts and assembly code. This triple reinforcement should make it easier to comprehend software requirements.

The software has four basic modules, as shown in the memory map of figure 19. They include : Initialization & Diagnostics, VLANCE Interrupt, Message Interrupt, and Status Module.

7.2. INITIALIZATION & DIAGNOSTICS SOFTWARE MODULE

7.2.1. INTRODUCTION. The Initialization & Diagnostics Module, as shown in figure 20, contains six submodules :

1. Clear
2. Block-Move
3. Receive Ring Initialization
4. Transmit Ring Initialize
5. Diagnostics
6. Normal Initialize.

The diagnostics portion is actually a subroutine. Diagnostics may be performed at any time by simpling calling this subroutine.

In addition to these six submodules are two subroutines : Control and status Register Initialization Subroutine, and the Cyclic Redundancy Check Subroutine.

Figure 19 : Software Memory Map.

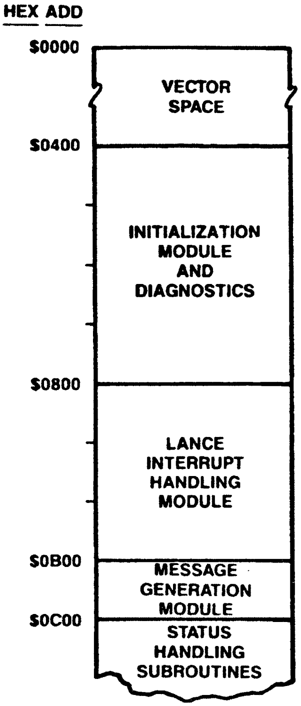
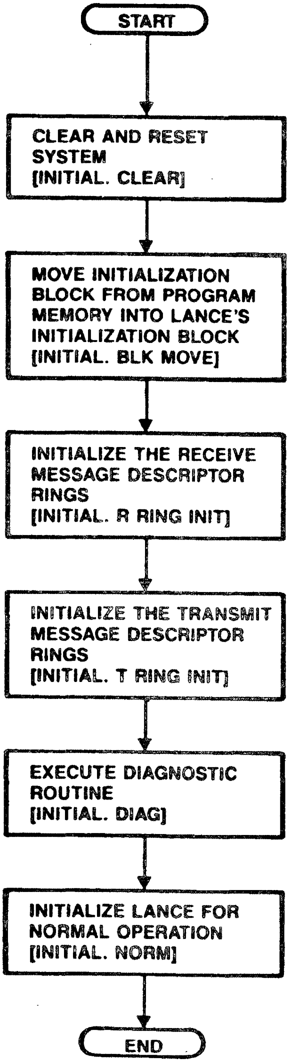

HEX ADD
$0000
VECTOR SPACE
$0400
INITIALIZATION MODULE AND DIAGNOSTICS
$0800
LANCE INTERRUPT HANDLING MODULE
$0B00
MESSAGE GENERATION MODULE
$0C00
STATUS HANDLING SUBROUTINES

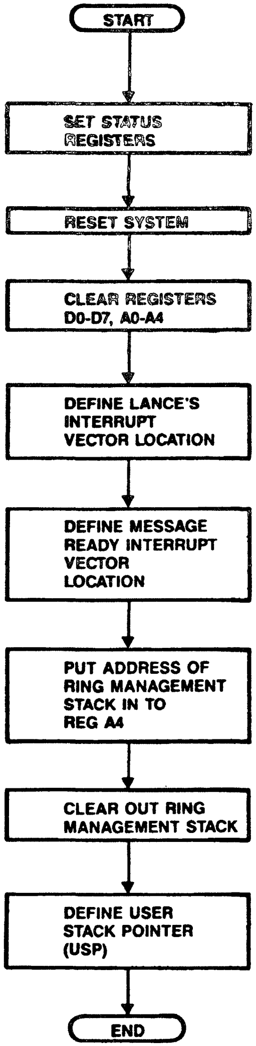
Figure 20 : Initialization Software Module (VLANCE. INITIAL).


START
CLEAR AND RESET SYSTEM [INITIAL. CLEAR]
MOVE INITIALIZATION BLOCK FROM PROGRAM MEMORY INTO LANCE'S INITIALIZATION BLOCK [INITIAL. BLK MOVE]
INITIALIZE THE RECEIVE MESSAGE DESCRIPTOR RINGS [INITIAL. R RING INIT]
INITIALIZE THE TRANSMIT MESSAGE DESCRIPTOR RINGS [INITIAL. T RING INIT]
EXECUTE DIAGNOSTIC ROUTINE [INITIAL. DIAG]
INITIALIZE LANCE FOR NORMAL OPERATION [INITIAL. NORM]
END

7.2.2. CLEAR SUBMODULE. Upon powerup, the 68000 (also referred to as the local host, or host) addresses the reset vector location $0000 that holds the address of the start of the program. The starting address of the Clear Submodule, shown in figure 21, is at location $400.

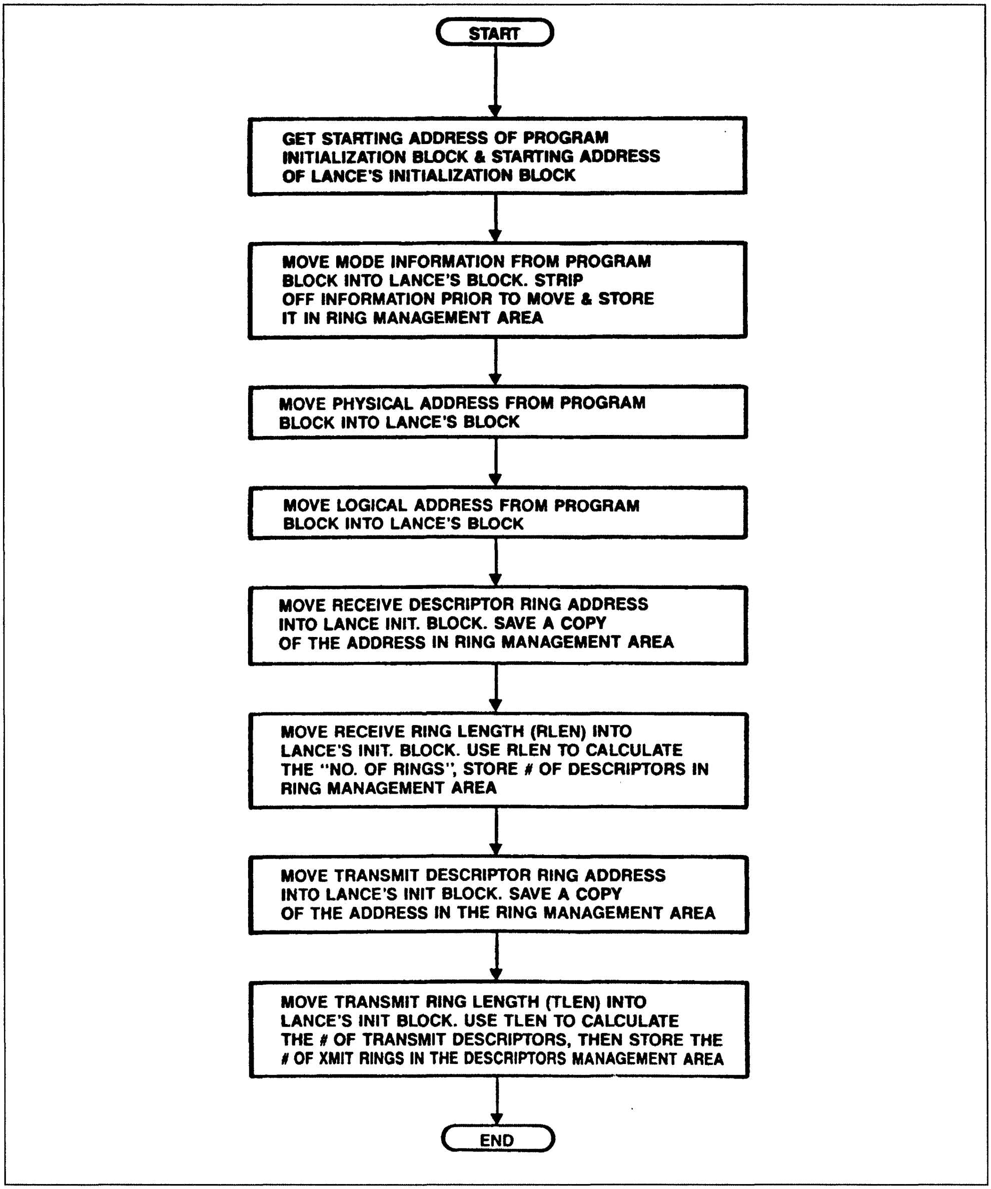
The program's first action is to reset the VLANCE and other peripherals on the local system bus. Next, it clears out the address and data registers, sets the interrupt mask, loads the interrupt autovectors and both system and ring-management stack locations. The ring management area starting address is stored in Address Register A4. Following this action, the host clears all data held in the ring status register.

Figure 21 : Clear Submodule (VLANCE. INITIAL. CLEAR).


START
SET STATUS REGISTERS
RESET SYSTEM
CLEAR REGISTERS D0-D7, A0-A4
DEFINE LANCE'S INTERRUPT VECTOR LOCATION
DEFINE MESSAGE READY INTERRUPT VECTOR LOCATION
PUT ADDRESS OF RING MANAGEMENT STACK IN TO REG A4
CLEAR OUT RING MANAGEMENT STACK
DEFINE USER STACK POINTER (USP)
END

7.2.3. BLOCK-MOVE SUBMODULE. Once the registers have all been cleared, the processor's next task is to move the VLANCE's initialization block into memory where the VLANCE can access it. Information also must be extracted from the initialization block which allows the software to determine the base addresses of the transmit and receive descriptor rings and their respective lengths. These operations take place in the "Block-Move" submodule described by the flowchart in figure 22.

Figure 22 : Block-Move Submodule (VLANCE. INITIAL. BLKMOVE).


START
GET STARTING ADDRESS OF PROGRAM INITIALIZATION BLOCK & STARTING ADDRESS OF LANCE'S INITIALIZATION BLOCK
MOVE MODE INFORMATION FROM PROGRAM BLOCK INTO LANCE'S BLOCK. STRIP OFF INFORMATION PRIOR TO MOVE & STORE IT IN RING MANAGEMENT AREA
MOVE PHYSICAL ADDRESS FROM PROGRAM BLOCK INTO LANCE'S BLOCK
MOVE LOGICAL ADDRESS FROM PROGRAM BLOCK INTO LANCE'S BLOCK
MOVE RECEIVE DESCRIPTOR RING ADDRESS INTO LANCE INIT. BLOCK. SAVE A COPY OF THE ADDRESS IN RING MANAGEMENT AREA
MOVE RECEIVE RING LENGTH (RLEN) INTO LANCE'S INIT. BLOCK. USE RLEN TO CALCULATE THE "NO. OF RINGS", STORE # OF DESCRIPTORS IN RING MANAGEMENT AREA
MOVE TRANSMIT DESCRIPTOR RING ADDRESS INTO LANCE'S INIT BLOCK. SAVE A COPY OF THE ADDRESS IN THE RING MANAGEMENT AREA
MOVE TRANSMIT RING LENGTH (TLEN) INTO LANCE'S INIT BLOCK. USE TLEN TO CALCULATE THE # OF TRANSMIT DESCRIPTORS, THEN STORE THE # OF XMIT RINGS IN THE DESCRIPTORS MANAGEMENT AREA
END

The VLANCE buffer management information is set up as equates. Equates define the following information :

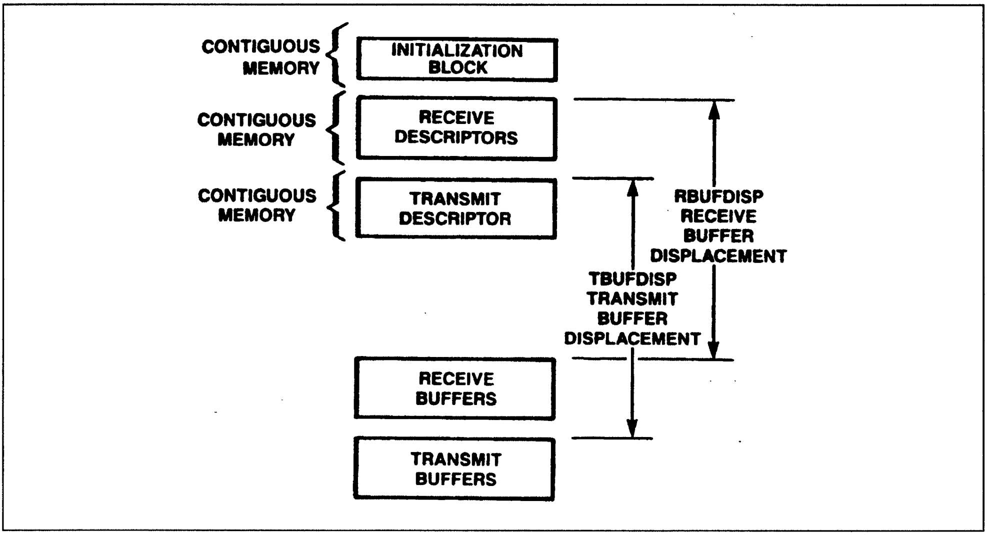
1. Starting address of the initialization block
2. Displacement needed between the initialization block's starting address to the receiver and transmit descriptors' starting address
3. Number of transmit and receive buffers desired
4. Desired buffer size.

By altering one or more equate statements, any of these buffer management areas may be relocated in memory.

This software routine makes all calculations needed to initialize the receive and transmit descriptors. The only limitation is that all receive descriptors must be in a contiguous block of memory. The same holds true for transmit descriptors. The buffers for the receive and transmit operation may be placed anywhere throughout the memory by specifying a buffer displacement from the descriptor's starting address (see figure 23).

This design was chosen to give the program more flexibility. The same piece of software can be used, independent of buffer number, size, and location.

Figure 23 : Buffer Displacement Diagram.



Information is stripped by the block-move submodule from the initialization block source and transferred to the VLANCE shared memory. If the source is EPROM, the initialization block is identical upon each powerup. If the source is downloaded from the main host, the initialization block may vary from one powerup sequence to the next. The software in this Application Note was written so the initialization block, receive and transmit descriptors, and the receive and transmit buffers are all in contiguous memory.

The "Block-Move" submodule also strips the receive and transmit ring lengths as it moves the initialization block into memory. It stores the values to be used in memory allocation calculations in the receive and transmit ring initialization routines. It also extracts mode information and sets the corresponding bits in the ring management status register. In addition, it also copies, reformats for 68000 compatability, and stores the receive and transmit ring base address for later use.

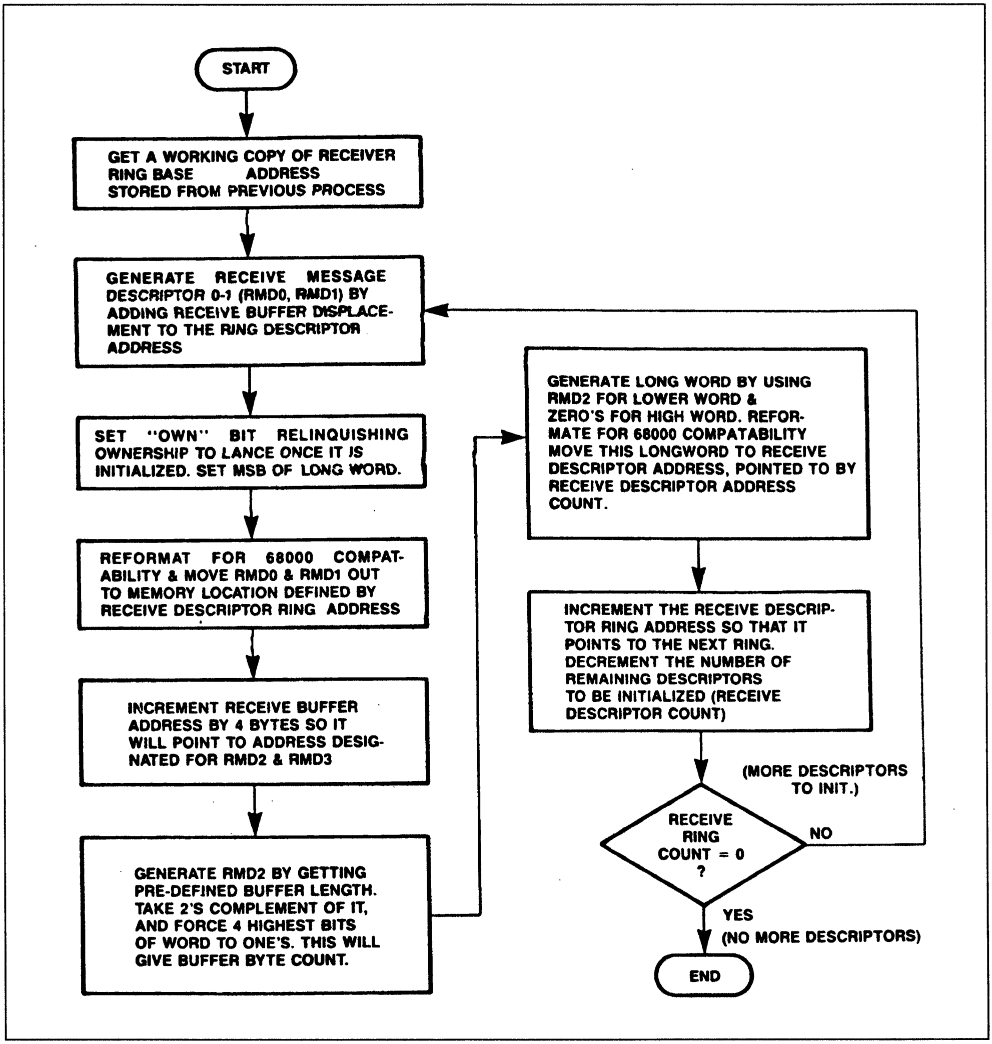
In this software design, the receive and transmit descriptor base addresses are predetermined and the program generates the buffer addresses from ring length information. The software also could have been structured to give only ring length information. The program would then determine ring and buffer locations. Another structure may have all address information pre-calculated without any calculations necessary upon initialization, although this structure would not allow flexible ring management. The structure depends on the application desired.

7.2.4. RECEIVE RING INITIALIZATION SUBMODULE. Once the entire initialization block is moved into the share memory, the "Block-Move" submodule

dule is complete and the "Receive Ring Initialization" submodule begins. This submodule initializes all receive ring descriptors by generating Receive Descriptors 0 through 3 (RMD0, RMD1, RMD2, & RMD3), as shown in figure 24. It generates each descriptor's respective buffer address by displacing it with the buffer length and displacement.

For example, if 256-byte buffers are required and the receive descriptor has a starting address of hex $5000 and a buffer displacement of hex $1000, users would first add the buffer displacement to the starting address of the descriptors. This would give a receive buffer starting address of hex $6000. Therefore, the buffer address of receive buffer number 1 would be $6000. Receive buffer number 2 would have a starting address of $6100, Receive number 3 would have a starting address of $6200 and so on, each displaced 256 bytes, or one buffer length from the previous buffer. Once the descriptor is initialized, ownership of it is given to the VLANCE to be used when an incoming message is received.

Figure 24 : Receive Ring Initialization Submodule (VLANCE. INITAL. RRINGINIT).


START
GET A WORKING COPY OF RECEIVER RING BASE ADDRESS STORED FROM PREVIOUS PROCESS
GENERATE RECEIVE MESSAGE DESCRIPTOR 0-1 (RMD0, RMD1) BY ADDING RECEIVE BUFFER DISPLACEMENT TO THE RING DESCRIPTOR ADDRESS
SET "OWN" BIT RELINQUISHING OWNERSHIP TO LANCE ONCE IT IS INITIALIZED. SET MSB OF LONG WORD.
REFORMAT FOR 68000 COMPATABILITY & MOVE RMD0 & RMD1 OUT TO MEMORY LOCATION DEFINED BY RECEIVE DESCRIPTOR RING ADDRESS
INCREMENT RECEIVE BUFFER ADDRESS BY 4 BYTES SO IT WILL POINT TO ADDRESS DESIGNATED FOR RMD2 & RMD3
GENERATE RMD2 BY GETTING PRE-DEFINED BUFFER LENGTH. TAKE 2'S COMPLEMENT OF IT, AND FORCE 4 HIGHEST BITS OF WORD TO ONE'S. THIS WILL GIVE BUFFER BYTE COUNT.
GENERATE LONG WORD BY USING RMD2 FOR LOWER WORD & ZERO'S FOR HIGH WORD. REFORMATE FOR 68000 COMPATABILITY MOVE THIS LONGWORD TO RECEIVE DESCRIPTOR ADDRESS, POINTED TO BY RECEIVE DESCRIPTOR ADDRESS COUNT.
INCREMENT THE RECEIVE DESCRIPTOR RING ADDRESS SO THAT IT POINTS TO THE NEXT RING. DECREMENT THE NUMBER OF REMAINING DESCRIPTORS TO BE INITIALIZED (RECEIVE DESCRIPTOR COUNT)
(MORE DESCRIPTORS TO INIT.)
RECEIVE RING COUNT = 0 ?
NO
YES
(NO MORE DESCRIPTORS)
END

7.2.5. TRANSMIT RING INITIALIZATION SUBMODULE. When all receive ring descriptors are initialized, the program proceeds into the "Transmit Ring Initialization" submodule shown in figure 25. This submodule initializes the transmit descriptors in much the same way it initializes the receive descriptors, with the exception that the host retains ownership of the transmit descriptors. They are used during message transmission. This submodule also generates the 2's complement of the byte count (BCNT) and places it in the address specified for TMD2.

7.2.6. DIAGNOSTIC SUBROUTINE. At this point in the program, the initialization block is in memory, which the VLANCE can access. The receive and transmit rings are initialized. A diagnostic routine now needs to be run to determine if the VLANCE and associated hardware are operating properly. (See figures 26 and 27). This is done by placing the VLANCE in four different loopback modes :

1. Internal loopback with transmit CRC enabled
2. Internal loopback with transmit CRC disabled
3. Internal loopback with transmit CRC enabled and the collision force bit set.
4. External loopback.

To test the VLANCE a test message must be generated and placed in the first transmit buffer. In addition, the first transmit descriptor must be initialized.

The test data message created in this program is 28 bytes long. The test message is all hex A's, which is alternating binary 1's and 0's. The maximum amount of data that can be transmitted in loopback mode, whether external or internal, is 32 bytes. With transmit CRC enabled, a 28-byte message is transmitted but, the actual message size is 32 bytes because the VLANCE tags four bytes of hardware CRC upon transmission. The transmit length constraint is due to the size limitation of the VLANCE's SILO.

In this Application Note, the loopback date size is 28 bytes for all four tests.

After the message is generated, a CRC is generated by calling the software CRC subroutine described in Section 7.2.6.1. In all but the second loopback test, the VLANCE's transmit CRC is enabled (bit 3 of the mode is set for transmit CRC disable). When the transmit CRC is disabled, a software CRC is generated and transmitted with the message. For the remaining loopback tests, the VLANCE generates the CRC code in hardware and tags it on the end of the transmitted message. This hardware-generated CRC is then compared to the software CRC to assure proper operation of the VLANCE.

The CRC generation is followed by initialization of the transmit descriptor. The buffer byte count (BCNT) of 28 is written into transmit message descriptor 2 (TMD2). The start and end of packet bits, as well as the own bit are set in transmit message descriptor 1 (TMD1). Following this descriptor initialization, the mode is set to : promiscuous, internal loopback, with the "disable transmit CRC" bit set to "0".

As described before, with transmit CRC enabled, a CRC is generated for the outgoing message, but is not checked for the incoming message. This is because the VLANCE has only one CRC device. The device can generate the CRC or check the CRC, but not concurrently. Verification of proper CRC occurs in the Receive Interrupt submodule, described later.

The final step in each loopback test is an initialization of the VLANCE's control and status registers (CSRINIT subroutine call). The CSRINIT subroutine initializes the control and status registers which effectively starts the VLANCE. These subroutine steps are summarized in Section 7.2.6.2.

Once the CSRINIT subroutine has executed, the VLANCE initializes itself. The program stays in a loop, waiting for completion of this part of the loopback test. This is indicated when a test-complete bit is set in the ring management status register. Once one portion of the loopback test has completed, the next portion begins.

The steps taken after the CSR initialization subroutine is called include :

1. The VLANCE is initialized by calling [CSRINIT].
2. The VLANCE requests the bus and makes DMA cycles. It reads the entire initialization block.
3. The VLANCE polls the receive and transmit rings to check for ownership (see MK68590 VLANCE Technical Manual for a more detailed description of polling routines).
4. Finding that it owns the transmit ring, it enters its transmit DMA routine.
5. Once it has completed its transmission, it immediately starts to DMA the message it concurrently received into the receive buffer.

Figure 25 : Transmit Ring Initialization Submodule (VLANCE. INITIAL. TRINGINIT).

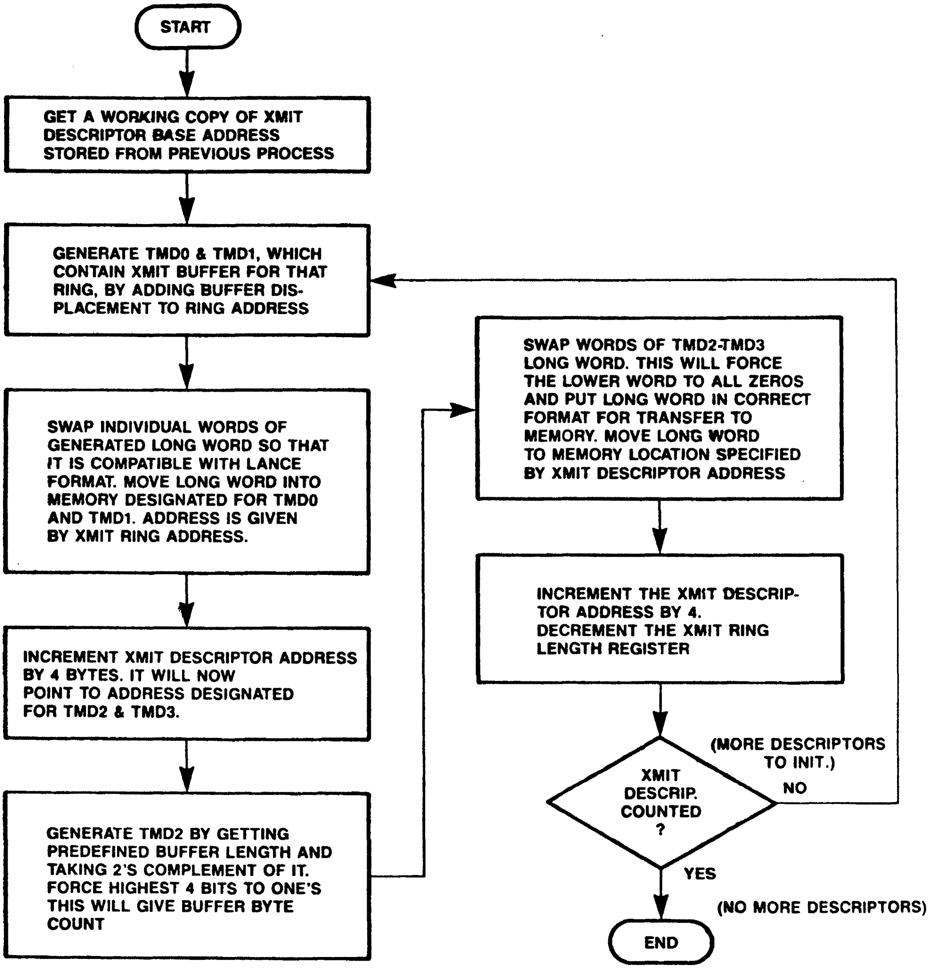

START
GET A WORKING COPY OF XMIT DESCRIPTOR BASE ADDRESS STORED FROM PREVIOUS PROCESS
GENERATE TMD0 & TMD1, WHICH CONTAIN XMIT BUFFER FOR THAT RING, BY ADDING BUFFER DISPLACEMENT TO RING ADDRESS
SWAP INDIVIDUAL WORDS OF GENERATED LONG WORD SO THAT IT IS COMPATIBLE WITH LANCE FORMAT. MOVE LONG WORD INTO MEMORY DESIGNATED FOR TMD0 AND TMD1. ADDRESS IS GIVEN BY XMIT RING ADDRESS.
INCREMENT XMIT DESCRIPTOR ADDRESS BY 4 BYTES. IT WILL NOW POINT TO ADDRESS DESIGNATED FOR TMD2 & TMD3.
GENERATE TMD2 BY GETTING PREDEFINED BUFFER LENGTH AND TAKING 2'S COMPLEMENT OF IT. FORCE HIGHEST 4 BITS TO ONE'S THIS WILL GIVE BUFFER BYTE COUNT
SWAP WORDS OF TMD2-TMD3 LONG WORD. THIS WILL FORCE THE LOWER WORD TO ALL ZEROS AND PUT LONG WORD IN CORRECT FORMAT FOR TRANSFER TO MEMORY. MOVE LONG WORD TO MEMORY LOCATION SPECIFIED BY XMIT DESCRIPTOR ADDRESS
INCREMENT THE XMIT DESCRIPTOR ADDRESS BY 4. DECREMENT THE XMIT RING LENGTH REGISTER
XMIT DESCRIP. COUNTED ?
(MORE DESCRIPTORS TO INIT.)
NO
YES
(NO MORE DESCRIPTORS)
END

Figure 26 : Diagnostics' Subroutine (VLANCE. INITIAL. DIAG).

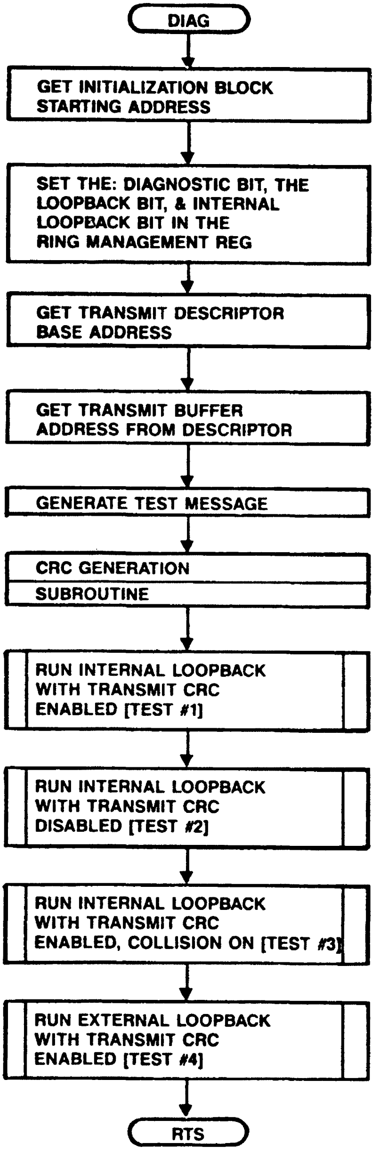
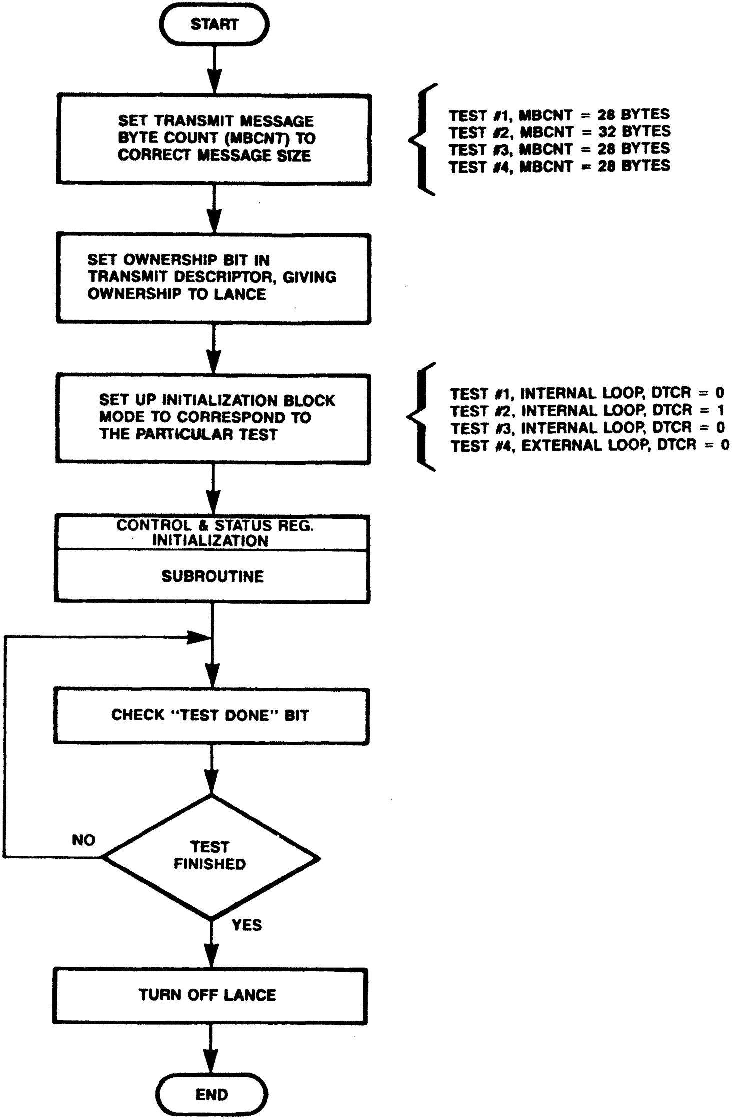

DIAG
GET INITIALIZATION BLOCK STARTING ADDRESS
SET THE: DIAGNOSTIC BIT, THE LOOPBACK BIT, & INTERNAL LOOPBACK BIT IN THE RING MANAGEMENT REG
GET TRANSMIT DESCRIPTOR BASE ADDRESS
GET TRANSMIT BUFFER ADDRESS FROM DESCRIPTOR
GENERATE TEST MESSAGE
CRC GENERATION
SUBROUTINE
RUN INTERNAL LOOPBACK WITH TRANSMIT CRC ENABLED [TEST #1]
RUN INTERNAL LOOPBACK WITH TRANSMIT CRC DISABLED [TEST #2]
RUN INTERNAL LOOPBACK WITH TRANSMIT CRC ENABLED, COLLISION ON [TEST #3]
RUN EXTERNAL LOOPBACK WITH TRANSMIT CRC ENABLED [TEST #4]
RTS

Figure 27 : Loopback Diagnostic Test Routine.


START
SET TRANSMIT MESSAGE BYTE COUNT (MBCNT) TO CORRECT MESSAGE SIZE
TEST #1, MBCNT = 28 BYTES
TEST #2, MBCNT = 32 BYTES
TEST #3, MBCNT = 28 BYTES
TEST #4, MBCNT = 28 BYTES
SET OWNERSHIP BIT IN TRANSMIT DESCRIPTOR, GIVING OWNERSHIP TO LANCE
SET UP INITIALIZATION BLOCK MODE TO CORRESPOND TO THE PARTICULAR TEST
TEST #1, INTERNAL LOOP, DTCR = 0
TEST #2, INTERNAL LOOP, DTCR = 1
TEST #3, INTERNAL LOOP, DTCR = 0
TEST #4, EXTERNAL LOOP, DTCR = 0
CONTROL & STATUS REG. INITIALIZATION
SUBROUTINE
CHECK "TEST DONE" BIT
TEST FINISHED
NO
YES
TURN OFF LANCE
END

6. The VLANCE then interrupts the 68000 to notify it that a transmission and reception has occurred.
7. The program vectors off to service the interrupt routine. During the exception processing routine, the application-dependent status subroutines are called if an error flag is found.
8. Once all exception processing is complete, the test done bit is set and a return from the exception processing occurs.
9. The "test-done" bit is constantly checked. Once set, the program continues normal execution, turning off the VLANCE and proceeding to the next portion of the diagnostics.

When loopback test number one is complete, the second loopback test proceeds in a similar manner as the first. The only difference between loopback tests is the mode. For clarity, this test is repeated four times in the software rather than tightening the code by inserting four loops.

If the diagnostics detects problems with the VLANCE, the status subroutines report the errors by setting a status bit in memory. This Application Note does not deal with error handling, since it is application-dependent and out of the realm of this document.

After all four loopback tests have completed, the VLANCE is initialized for normal operation.

7.2.6.1. CRC Code Software Generation. Before the descriptor initialization, a CRC subroutine, is called to generate a 32-bit cycle redundancy check code (CRC) to be added at the end of the message (see figure 28).

Since all hex A's is the predetermined data in the message, the CRC can also be predetermined. If the test message differed for each test, an actual rigorus software CRC subroutine would have to be designed. But, in this case, the CRC subroutine simply writes $B1109280 out to the message buffer. This data is the CRC code for the 28-byte test message.

7.2.6.2. Control and Status Register Initialization Subroutine. The Control and Status register subroutine is shown in figure 29. The first step is to move the initialization starting address into the VLANCE's Control and Status registers. The low order bits <00 : 15> are placed in CSR1, and the high order bits <16 : 23> are placed in CSR2. Next, the VLANCE is made compatible to the hardware interface by setting BSWP = 1, ACON = 1, and BCON = 0 in CSR3. Finally, a $0043 is written into CSR0, which sets the Interrupt Enable, the Start and Initialize bits. Immediately following this last write, the VLANCE starts its initialization procedure by requesting the bus and completing 12 DMA cycles. Each DMA cycle corresponds by moving one word of the initialization block into its internal registers.

Figure 28 : Cycle Redundancy Check Generation Subroutine (CRCGEN).

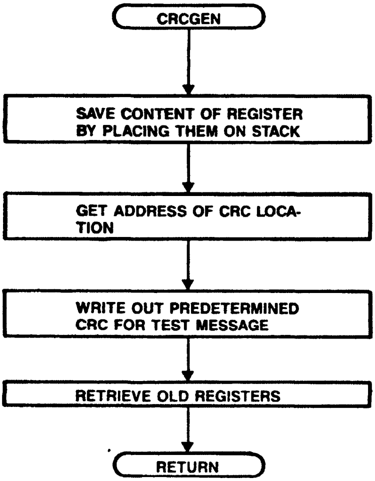
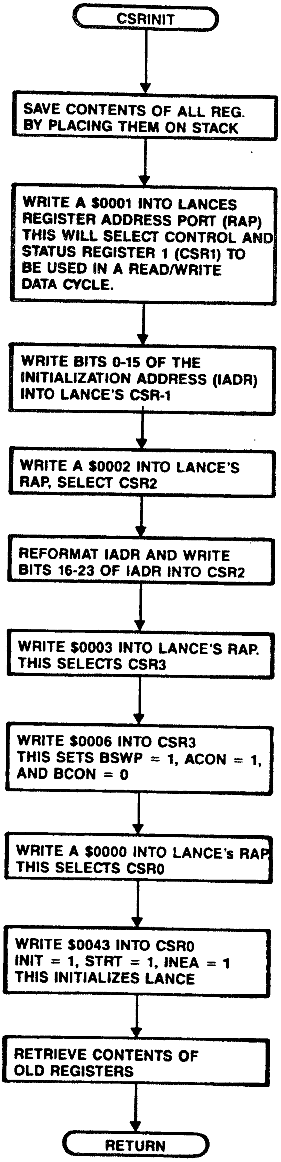

CRCGEN
SAVE CONTENT OF REGISTER BY PLACING THEM ON STACK
GET ADDRESS OF CRC LOCATION
WRITE OUT PREDETERMINED CRC FOR TEST MESSAGE
RETRIEVE OLD REGISTERS
RETURN

Figure 29 : Control and Status Initialization Subroutine (CSRINIT).


CSRINIT
SAVE CONTENTS OF ALL REG. BY PLACING THEM ON STACK
WRITE A $0001 INTO LANCES REGISTER ADDRESS PORT (RAP) THIS WILL SELECT CONTROL AND STATUS REGISTER 1 (CSR1) TO BE USED IN A READ/WRITE DATA CYCLE.
WRITE BITS 0-15 OF THE INITIALIZATION ADDRESS (IADR) INTO LANCE'S CSR-1
WRITE A $0002 INTO LANCE'S RAP, SELECT CSR2
REFORMAT IADR AND WRITE BITS 16-23 OF IADR INTO CSR2
WRITE $0003 INTO LANCE'S RAP. THIS SELECTS CSR3
WRITE $0006 INTO CSR3 THIS SETS BSWP = 1, ACON = 1, AND BCON = 0
WRITE A $0000 INTO LANCE's RAP THIS SELECTS CSR0
WRITE $0043 INTO CSR0 INIT = 1, STRT = 1, INEA = 1 THIS INITIALIZES LANCE
RETRIEVE CONTENTS OF OLD REGISTERS
RETURN

Once the DMA cycles are complete, the VLANCE interrupts to notify the host of completed initialization. If the Interrupt Enable bit in CSR0 is not set, the VLANCE does not interrupt the host, but rather, waits for the host to poll its Control and Status register.

Next, a polling routine is begun to determine which receive and transmit descriptors are owned by VLANCE. When a VLANCE-owned receive ring is found, that result is stored and polling continues to locate a relinquished transmit ring. The CSR initialization submodule is complete when the final CSR0 takes place. The program then returns from this subroutine.

7.2.7. NORMAL-OPERATION INITIALIZATION. After the diagnostics are complete, the normal-operation initialization occurs. This submodule is shown in figure 30. The predefined mode in the program initialization block is moved into the VLANCE's initialization block. The mode information is checked to determine if the desired mode is a loopback mode. If it is, the ring management status register is set up accordingly. The VLANCE is then restarted by calling the control and status register initialization subroutine, CSRINIT.

After the Normal-Operation Initialization, the initialization software module is complete and stays in a wait loop. The program is completely interrupt-driven from this point on and only moves out of the wait loop when an interrupt occurs.

7.3. VLANCE INTERRUPT EXCEPTION SOFTWARE MODULE

7.3.1. INTRODUCTION. The VLANCE Interrupt Exception Module has basically four submodules. They are :

1. Interrupt Error Determination Submodule
2. Transmit Interrupt Handling Submodule
3. Receive Interrupt Handling Submodule
4. Initialization Done Interrupt Handling Submodule

Figure 31 depicts the flow of the Exception Software Module. The VLANCE Interrupt Exception Software Module, also referred to as the VLANCE Interrupt Routine, is called when the VLANCE interrupts the host processor. The VLANCE interrupts the host if a message has been transmitted or received, the VLANCE has finished its initialization routine, or an error has occurred.

The VLANCE Interrupt is hardware set at level 6 ; the second highest priority. The only higher priority is level seven, that of the nonmaskable interrupt. When the VLANCE interrupts the 68000, it autovectors to memory location $800.

The first action the 68000 takes is to get the VLANCE's status from CSR0. This status helps determine what caused the interrupt. The error bit is the first bit checked. If this bit is not set, the Transmit Interrupt bit is checked to see if a transmitted message caused the interrupt. If an error did cause the interrupt, the software determines the type of error and the routine proceeds to the transmit interrupt check. Once the transmit bit is checked, the receive bit is checked in the same manner. Finally the Initialization Done bit is checked and the program returns to the waiting loop.

All flag-causing conditions are serviced and the error-causing conditions call application-dependent service subroutines. For the purposes of this Application Note, these error service routines simply set certain bits in memory that correspond to the error. The programmer may then examine memory for errors.

7.3.2. INTERRUPT ERROR DETERMINATION SUBMODULE. The first task in this submodule, as shown in figure 32, is to save the registers. Information in the data and address registers is placed on the stack to be retrieved after exception processing. Next, status information in the Control and Status Register zero (CSR0) is moved into one of the 68000's registers. Following this, all flags set in CSR0 and the Interrupt Enabled bit are cleared. This occurs by clearing bit 6 (Interrupt Enable bit) in the register containing a copy of CSR0, and writing this copy back into CSR0. Since the flags are all cleared by writing a "1" in their bit location, all flags set in CSR0 are cleared. The Error bit in CSR0 clears itself after all the individual error flags are cleared. The interrupt enable bit is cleared by writing a "0" in its bit location. The VLANCE's interrupt capabilities are disabled because it is not desirable to have another interrupt occur while the present interrupt from the VLANCE is being serviced.

Figure 30 : Normal Operation Initialization Submodule (VLANCE. INITIAL. NORM).

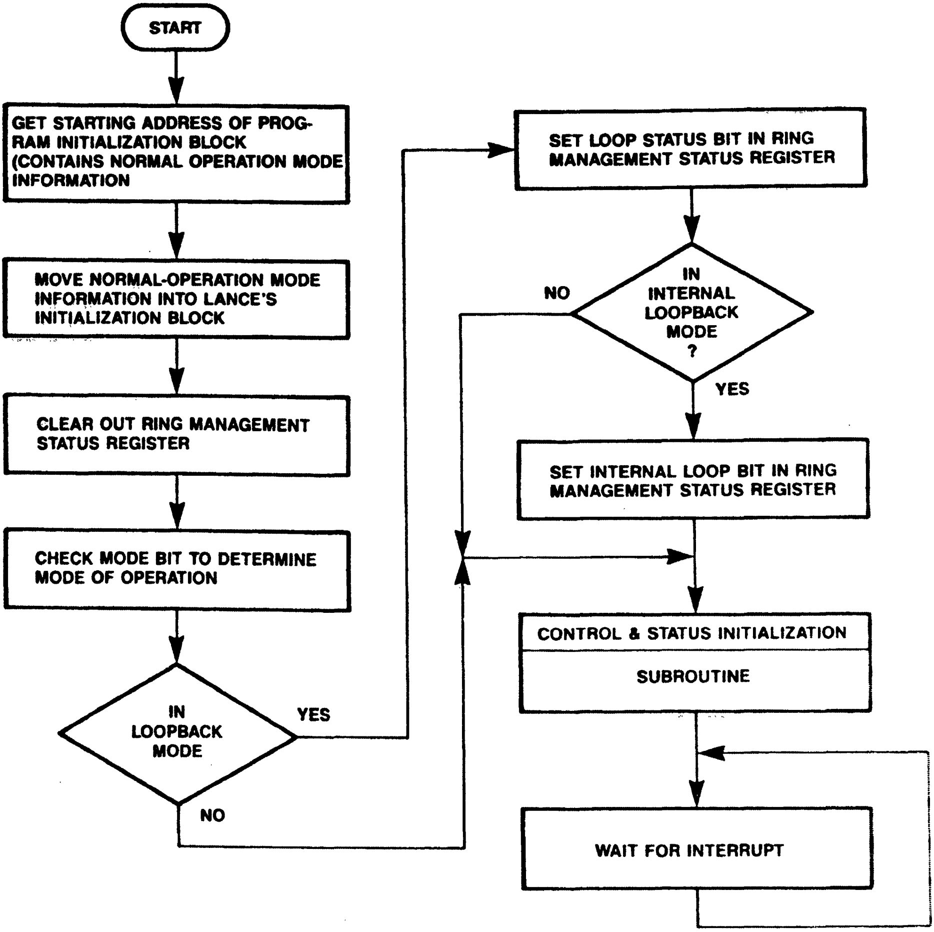

START
GET STARTING ADDRESS OF PROG-RAM INITIALIZATION BLOCK (CONTAINS NORMAL OPERATION MODE INFORMATION
MOVE NORMAL-OPERATION MODE INFORMATION INTO LANCE'S INITIALIZATION BLOCK
CLEAR OUT RING MANAGEMENT STATUS REGISTER
CHECK MODE BIT TO DETERMINE MODE OF OPERATION
IN LOOPBACK MODE
YES
NO
SET LOOP STATUS BIT IN RING MANAGEMENT STATUS REGISTER
IN INTERNAL LOOPBACK MODE ?
NO
YES
SET INTERNAL LOOP BIT IN RING MANAGEMENT STATUS REGISTER
CONTROL & STATUS INITIALIZATION
SUBROUTINE
WAIT FOR INTERRUPT

Figure 31 : VLANCE Exception Processing Module (LEXCEPT).

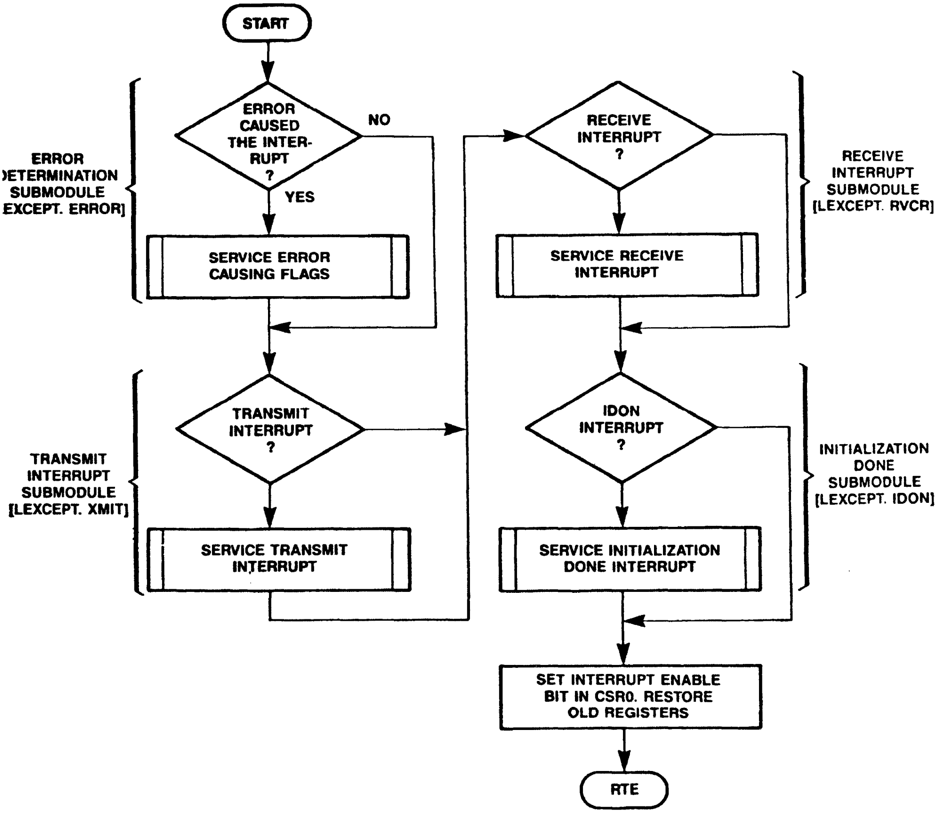
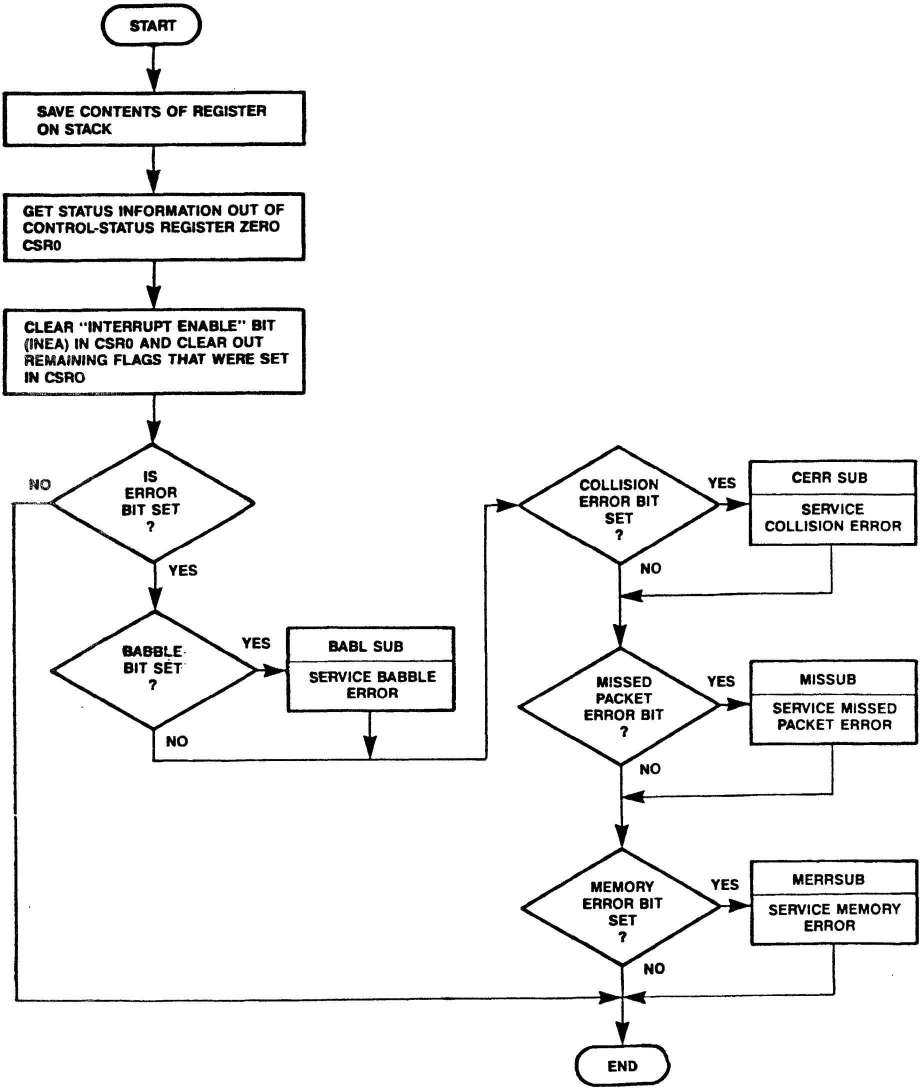

START
ERROR CAUSED THE INTER-RUPT ?
NO
YES
ERROR)ETERMINATION SUBMODULE EXCEPT. ERROR]
SERVICE ERROR CAUSING FLAGS
TRANSMIT INTERRUPT ?
TRANSMIT INTERRUPT SUBMODULE [LEXCEPT. XMIT]
SERVICE TRANSMIT INTERRUPT
RECEIVE INTERRUPT ?
RECEIVE INTERRUPT SUBMODULE [LEXCEPT. RVCR]
SERVICE RECEIVE INTERRUPT
IDON INTERRUPT ?
INITIALIZATION DONE SUBMODULE [LEXCEPT. IDON]
SERVICE INITIALIZATION DONE INTERRUPT
SET INTERRUPT ENABLE BIT IN CSR0. RESTORE OLD REGISTERS
RTE

Figure 32 : Interrupt Error Determination Submodule (VLANCE. LEXCEPT. ERROR).


START
SAVE CONTENTS OF REGISTER ON STACK
GET STATUS INFORMATION OUT OF CONTROL-STATUS REGISTER ZERO CSR0
CLEAR "INTERRUPT ENABLE" BIT (INEA) IN CSR0 AND CLEAR OUT REMAINING FLAGS THAT WERE SET IN CSRO
IS ERROR BIT SET ?
NO
YES
BABBLE BIT SET ?
YES
NO
BABL SUB
SERVICE BABBLE ERROR
COLLISION ERROR BIT SET ?
YES
NO
CERR SUB
SERVICE COLLISION ERROR
MISSED PACKET ERROR BIT ?
YES
NO
MISSUB
SERVICE MISSED PACKET ERROR
MEMORY ERROR BIT SET ?
YES
NO
MERRSUB
SERVICE MEMORY ERROR
END

The Error Summary bit is the first status bit to be checked. If this bit is not set, no error-causing conditions are present and the individual error bits need not be checked. The Transmit Interrupt bit is checked next to see if it is set.

If the Error Summary bit is set, each individual error bit is checked. If any are set, an error-handling subroutine is called to report or service the error. Again, as stated before, these are application-dependent subroutines. The user can implement error handling routines to suit system requirements.

When it is determined whether an error caused the interrupt, the Transmit Interrupt bit is checked. Since more than one condition can cause the interrupt, all bits must be checked. The interrupt pin is simply an OR of the interrupt-causing conditions. If another interrupt occurs after the interrupt pin is asserted, another transition on the interrupt pin will not occur.

7.3.3. TRANSMIT INTERRUPT HANDLING SUBMODULE. If the Transmit Interrupt bit is set, the Transmit Interrupt Handling Submodule is executed (see figure 33). First, the address pointer to the last transmit descriptor ring is retrieved from the ring management area. This address is incremented by two bytes so it points to the transmit status (TMD1). Next, the transmit status is moved into a 68000 register. The "More" bit is checked to see if more than one transmission attempts were required. If so, the More counter in the ring management area is updated. The "One" bit is then checked to see if it took exactly one attempt to transmit the message. If so, the One counter is updated in a similar manner.

Next, the transmit error bit is tested to see if a transmission error occurred. As with the Interrupt Error bit in CSR0, if this bit is not set, it is not necessary to check each individual error bit, but if this Error Summary bit is set, the program does check each individual error bit. It checks the errors in the following order : retry error, loss of carrier error, late collision error, underflow error, and finally the buffer error. If any of these errors have occurred, the error handling subroutines are called to report and/or service the error-causing conditions.

If the transmission error summary bit is not set, the program checks to see if the VLANCE is in the loopback mode. If this is true, a copy of the starting address of the transmit buffer is saved so the data in this buffer can be compared against data in the loopbacked message just received. This comparison indicates loopback mode status.

If the VLANCE is in the loopback mode, the pointer to the last transmit descriptor is not incremented. This is because the descriptor buffer must not be available for additional messages until data in the transmit buffer is compared to that in the receive message buffer. This comparison occurs in the Receive Interrupt Handling Submodule of the module that is presently executing. At this time, the Last Transmit Descriptor pointer is updated.

In all other cases, before this submodule is complete, the last descriptor address pointer is updated to point to the next available descriptor. When the last descriptor address is updated, the present address pointed to must be compared with the bottom of the ring.

If the pointer is pointing to the bottom of the ring, it must not be simply incremented but, the address of the top of the ring must be placed in the pointer register. Once the Transmit Interrupt Handling Submodule is complete, the Receive Interrupt bit is checked in the CSR0 status.

7.3.4. RECEIVE INTERRUPT HANDLING SUBMODULE. The Receive Interrupt Handling Submodule, shown in figure 34, is the most complicated piece of software in this program because it must handle loopback messages differently from a normal received message.

Two loopback situations can occur : One can occur during the diagnostic routine, and the other is the normal condition loopback. The latter loopback happens when the programmer desires to have external or internal loopback as a normal operating mode. This would most likely be the case during debut and system integration.

If during CSR0 Status Checking the Receive Interrupt bit is set, the Receive Interrupt Handling Submodule is executed the first step is to check the status of the receive message descriptor that the VLANCE has turned over to the host by retrieving the Receive Descriptor pointer from the ring management area. Next, receive message descriptor number one (RMD1) is moved into one of the 68000's registers and the Receive Error Summary bit is checked. If this bit is set, the individual error bits are checked to see which error, or combination of errors, caused the error summary bit to be set. Error bits are checked in the following order : frame error, CRC error, overflow error, and buffer error.

If the Error Summary bit is not set, and VLANCE is in the loopback mode, the loopback routine is executed (see figure 35). The loopback routine is still part of the receive interrupt routine, but is listed separately for clarity.

Figure 33 : Transmit Interrupt Handling Submodule (VLANCE. LEXCEPT. XMIT).

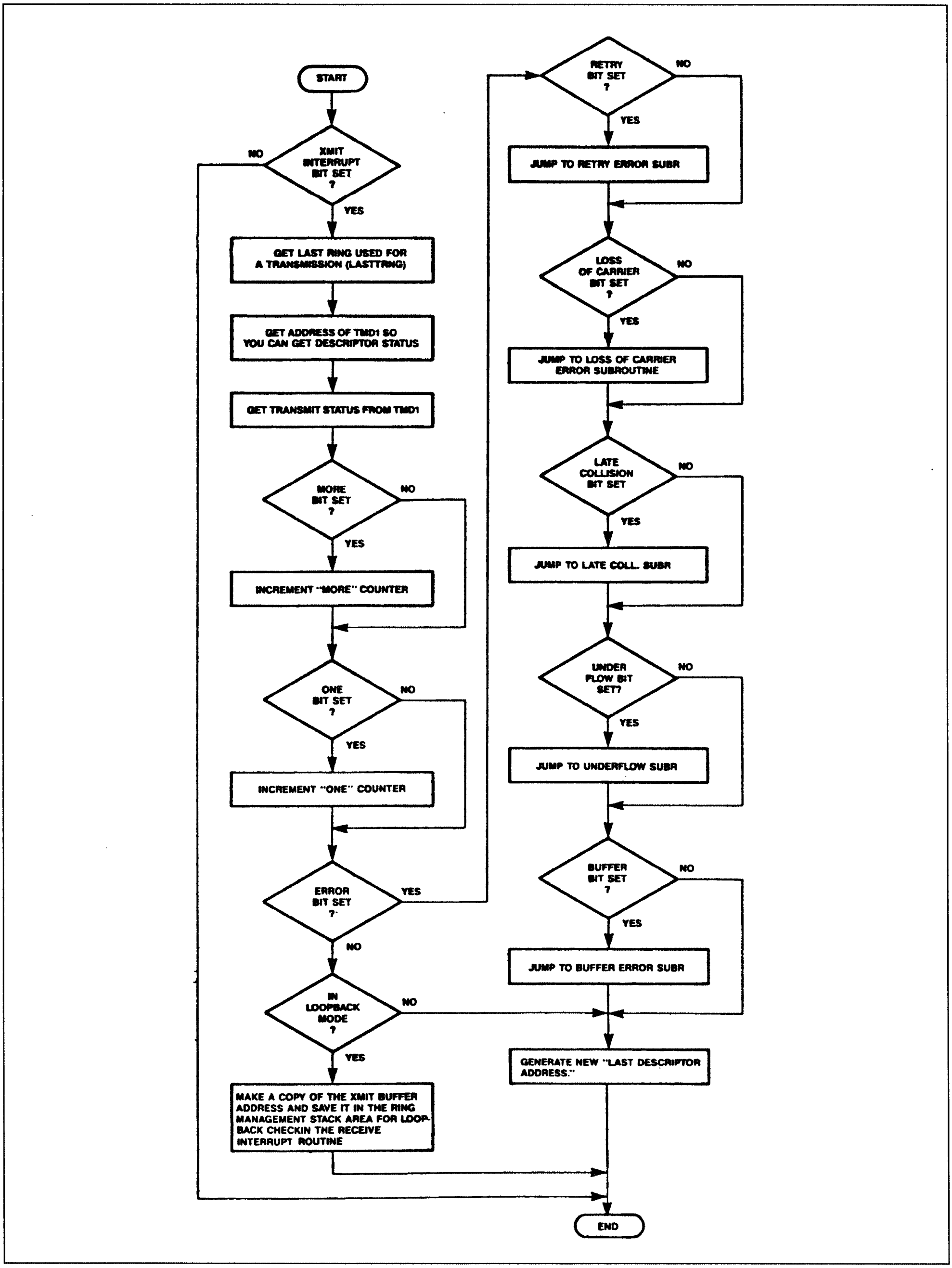

START
XMIT INTERRUPT BIT SET ?
NO
YES
GET LAST RING USED FOR A TRANSMISSION (LASTTRNG)
GET ADDRESS OF TMD1 SO YOU CAN GET DESCRIPTOR STATUS
GET TRANSMIT STATUS FROM TMD1
MORE BIT SET ?
NO
YES
INCREMENT "MORE" COUNTER
ONE BIT SET ?
NO
YES
INCREMENT "ONE" COUNTER
ERROR BIT SET ?
YES
NO
IN LOOPBACK MODE ?
NO
YES
MAKE A COPY OF THE XMIT BUFFER ADDRESS AND SAVE IT IN THE RING MANAGEMENT STACK AREA FOR LOOP-BACK CHECKIN THE RECEIVE INTERRUPT ROUTINE
RETRY BIT SET ?
NO
YES
JUMP TO RETRY ERROR SUBR
LOSS OF CARRIER BIT SET ?
NO
YES
JUMP TO LOSS OF CARRIER ERROR SUBROUTINE
LATE COLLISION BIT SET
NO
YES
JUMP TO LATE COLL. SUBR
UNDER FLOW BIT SET?
NO
YES
JUMP TO UNDERFLOW SUBR
BUFFER BIT SET ?
NO
YES
JUMP TO BUFFER ERROR SUBR
GENERATE NEW "LAST DESCRIPTOR ADDRESS."
END

Figure 34 : Receive Interrupt Handling Submodule (VLANCE. LEXCEPT. XMIT).

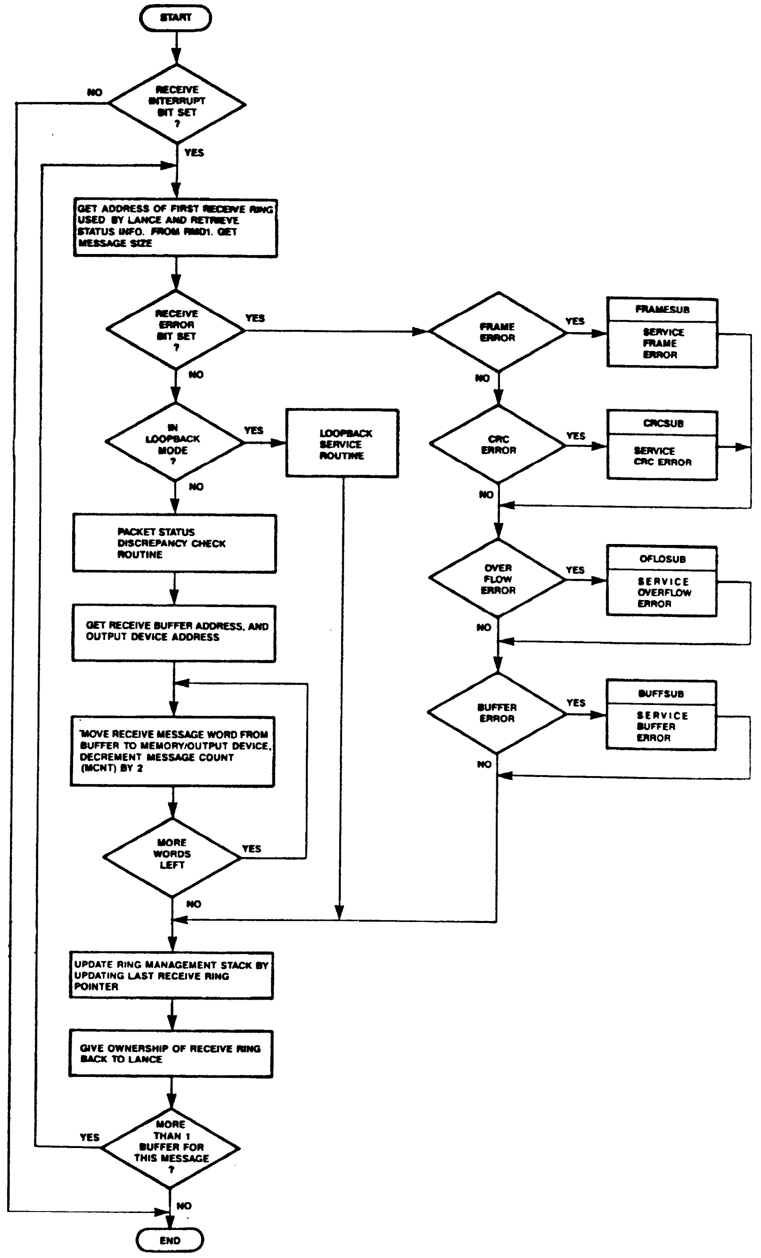
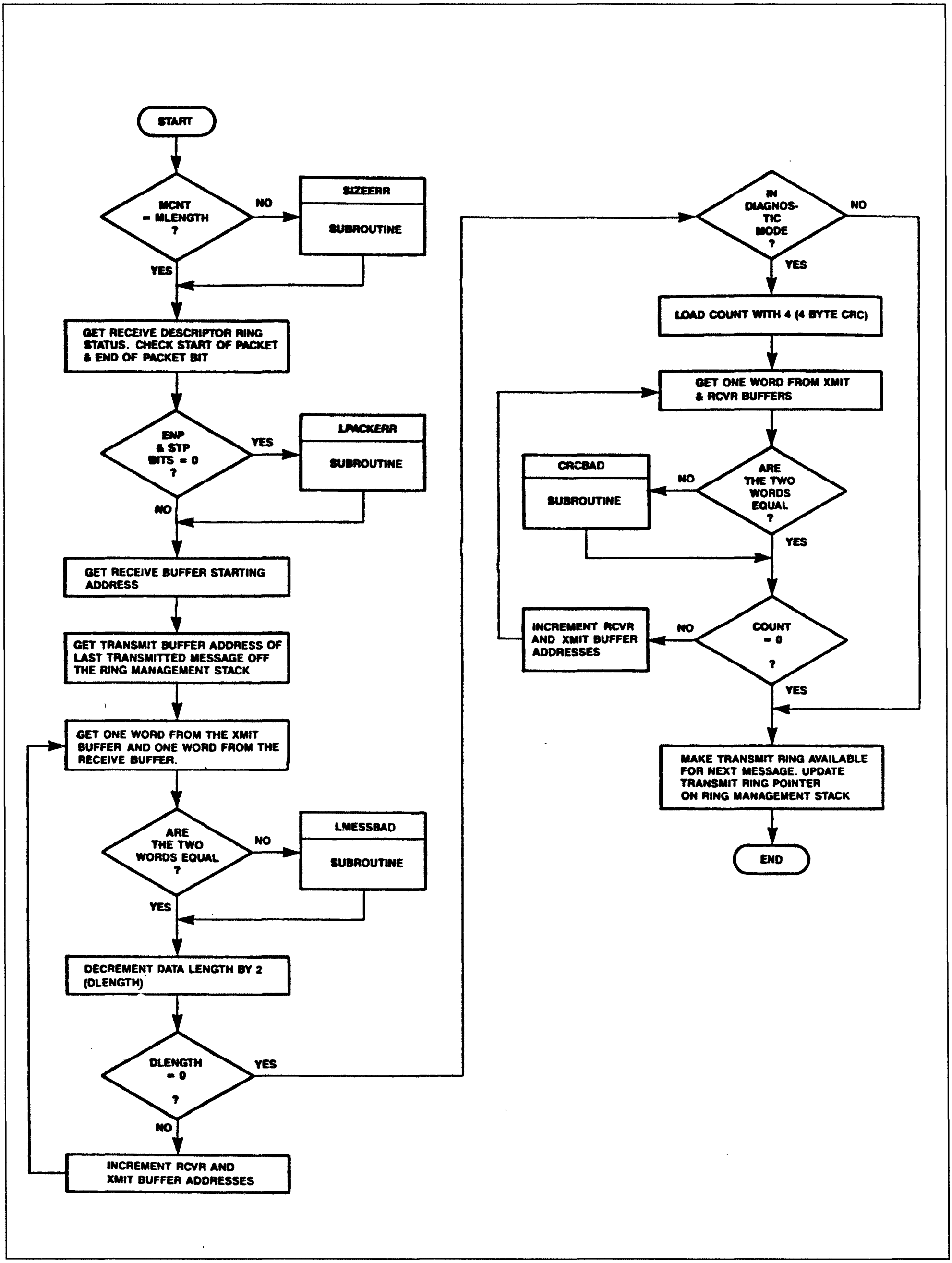

START
RECEIVE INTERRUPT BIT SET ?
NO
YES
GET ADDRESS OF FIRST RECEIVE RING USED BY LANCE AND RETRIEVE STATUS INFO. FROM RMD1. GET MESSAGE SIZE
RECEIVE ERROR BIT SET ?
YES
NO
FRAME ERROR
YES
NO
FRAMESUB
SERVICE FRAME ERROR
IN LOOPBACK MODE ?
YES
NO
LOOPBACK SERVICE ROUTINE
CRC ERROR
YES
NO
CRCSUB
SERVICE CRC ERROR
PACKET STATUS DISCREPANCY CHECK ROUTINE
OVER FLOW ERROR
YES
NO
OFLOSUB
SERVICE OVERFLOW ERROR
GET RECEIVE BUFFER ADDRESS, AND OUTPUT DEVICE ADDRESS
BUFFER ERROR
YES
NO
BUFFSUB
SERVICE BUFFER ERROR
MOVE RECEIVE MESSAGE WORD FROM BUFFER TO MEMORY/OUTPUT DEVICE, DECREMENT MESSAGE COUNT (MCNT) BY 2
MORE WORDS LEFT
YES
NO
UPDATE RING MANAGEMENT STACK BY UPDATING LAST RECEIVE RING POINTER
GIVE OWNERSHIP OF RECEIVE RING BACK TO LANCE
MORE THAN 1 BUFFER FOR THIS MESSAGE ?
YES
NO
END

Figure 35 : Loopback Service Routine (VLANCE. LEXCEPT. RVCR. LOOPBACK).


START
MCNT = MLENGTH ?
NO
YES
SIZEERR
SUBROUTINE
GET RECEIVE DESCRIPTOR RING STATUS. CHECK START OF PACKET & END OF PACKET BIT
ENP & STP BITS = 0 ?
YES
NO
LPACKERR
SUBROUTINE
GET RECEIVE BUFFER STARTING ADDRESS
GET TRANSMIT BUFFER ADDRESS OF LAST TRANSMITTED MESSAGE OFF THE RING MANAGEMENT STACK
GET ONE WORD FROM THE XMIT BUFFER AND ONE WORD FROM THE RECEIVE BUFFER.
ARE THE TWO WORDS EQUAL ?
NO
YES
LMESSBAD
SUBROUTINE
DECREMENT DATA LENGTH BY 2 (DLENGTH)
DLENGTH = 0 ?
YES
NO
INCREMENT RCVR AND XMIT BUFFER ADDRESSES
IN DIAGNOSTIC MODE ?
NO
YES
LOAD COUNT WITH 4 (4 BYTE CRC)
GET ONE WORD FROM XMIT & RCVR BUFFERS
ARE THE TWO WORDS EQUAL ?
NO
YES
CRCBAD
SUBROUTINE
COUNT = 0 ?
NO
YES
INCREMENT RCVR AND XMIT BUFFER ADDRESSES
MAKE TRANSMIT RING AVAILABLE FOR NEXT MESSAGE. UPDATE TRANSMIT RING POINTER ON RING MANAGEMENT STACK
END

7.3.4.1. Receive Interrupt Handling Submodule. The first step in this routine is to check the received message length, which must be equal to the message sent, plus four bytes if transmit CRC is enabled. This is done by checking MCNT against the transmit message word (MLENGTH). The transmit message word length is in the ring management area (see Ring Management). The size of the transmit message is moved into MLENGTH when loopback occurs during the diagnostics.

In normal operation with the application breadboard, an arbitrary message size can be written into DLENGTH, followed by pushing the message interrupt button, and a message of that data size is transmitted.

If the message sizes do not match, an error subroutine is called. Next the packet check is made. In loopback mode, transmit data chaining cannot occur and the maximum received message size is 36 bytes. The receive buffers are larger than 36 bytes and since there must be only one packet per message, the start- and end-of-packet bits must be set. If both are not set, an error routine is called.

After the packet check, the receive and transmit buffer starting addresses are retrieved. The receive buffer address is taken from the receive descriptor, while the transmit buffer address is taken out of LOOPXADD from the Ring Management Area. LOOPXADD is moved into the Ring Management Area during the transmit interrupt routine. By using these addresses, the transmit and receive message data can be compared to verify proper transmission and reception. If any part of the message has been altered, an error subroutine is called.

When the entire message has been compared, the program checks if it is in the diagnostic mode. As stated previously, a software CRC is generated for the test message, and is compared to VLANCE's hardware CRC. If any of the two do not match, an error subroutine is called. This comparison is only performed during diagnostics. If the program is in loopback, but not in diagnostics, this part of the routine is omitted.

After all comparisons are made, the transmit descriptor pointer is updated. The transmit descriptor pointer is not updated in the transmit interrupt routine because the descriptor points to the transmit buffer needed for the loopback data comparison. If the transmit buffer is made available before the data comparison, the buffer might be written over, thus invalidating the comparison.

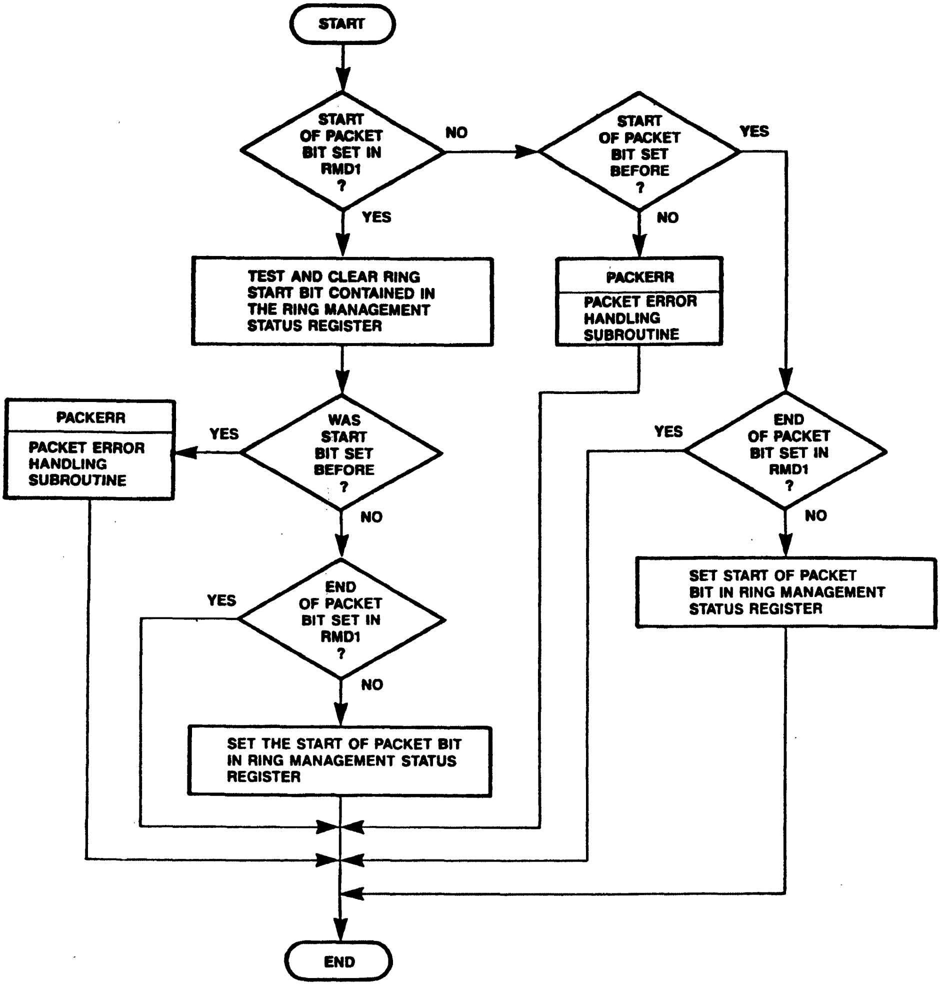
7.3.4.2. Receive Interrupt Normal Operation. If no errors are detected, and the LANCE is not in loopback, the receive interrupt routine proceeds normally. A packet check then searches for any errors with the start-of-packet and end-of-packet bits (see figure 36). The ring management status register contains information on the status of these bits. A receive message coming in with the start-of-packet bit set and the end-of-packet bit not set indicates that more than one buffer contains the message. In this situation, the start-of-packet bit is set in the ring management status register. If the following descriptor contains another start-of-packet bit, an error flag is set since an end-of-packet has not been detected between the two start-of-packets. If the end-of-packet bit is set in the descriptor, the whole packet is received and the status register is set accordingly. The same holds true for two end-of-packets detected sequentially. If two end-of-packets are detected, without a start-of-packet in between them, an error flag is raised. These error flags call application-dependent subroutines to service the error.

Once the packet check is complete, the received message is moved to an output device or to another memory location. Once moved, this layer of software no longer handles the message, instead the upper level software takes over. The upper level reformats the message and combines messages by sequence number as well as routing them to their final destination.

In this Application Note, the message is simply moved to another memory block. This can be accomplished effectively by changing the receive buffer starting address in the receive message descriptor to the address of an empty memory block. Then the used buffer address can be passed on to the next software layer. The same holds for transmitted messages. If the number of messages exceeds the number of available buffers, the host either rejects the last message and waits for a free buffer or moves the message into an empty memory block. When a descriptor is free, the host then changes the transmit buffer address in the descriptor.

The message is moved from the message buffer into the output memory a word at a time. Each time a word is moved, a count of the message length is decremented and checked to see if any messages still need to be transferred. Once the entire message is out of the buffer, the ring management area is updated by moving the last receive descriptor pointer to the next descriptor to be used by the VLANCE. The VLANCE again owns the buffer.

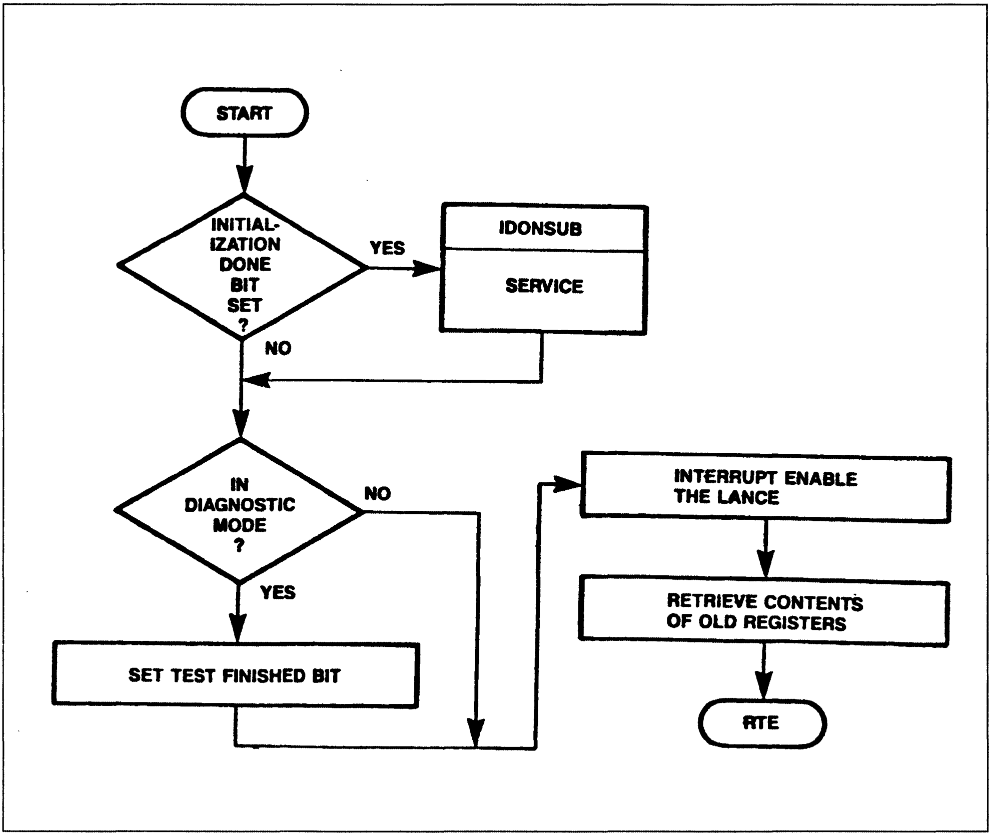
Figure 36 : Packet Status Discrepancy Check Routine (VLANCE. LEXCEPT. RVCR. PACKCHECK).


START
START OF PACKET BIT SET IN RMD1 ?
NO
YES
START OF PACKET BIT SET BEFORE ?
YES
NO
TEST AND CLEAR RING START BIT CONTAINED IN THE RING MANAGEMENT STATUS REGISTER
PACKERR
PACKET ERROR HANDLING SUBROUTINE
WAS START BIT SET BEFORE ?
YES
NO
PACKERR
PACKET ERROR HANDLING SUBROUTINE
END OF PACKET BIT SET IN RMD1 ?
YES
NO
SET START OF PACKET BIT IN RING MANAGEMENT STATUS REGISTER
END OF PACKET BIT SET IN RMD1 ?
YES
NO
SET THE START OF PACKET BIT IN RING MANAGEMENT STATUS REGISTER
END

The start bit in the ring management area undergoes a final check. A set bit indicates that this receive buffer was only one of several buffers needed for the full message. If more buffers are associated with this message, a jump back to the beginning of the receive interrupt routine is made and the routine is executed again until the entire message is processed. If the start bit is not set, this was the only buffer needed for the message, thus, the routine is completed.

7.3.5. INITIALIZATION DONE INTERRUPT HANDLING SUBMODULE. The final submodule of the VLANCE interrupt processing routine, shown in figure 37, is the Initialization Done Interrupt Handling Submodule. This submodule determines if the host was interrupted because the VLANCE had just completed initialization. CSR0 is checked to see if the IDON bit is set. If so, a flag-raising routine is called.

Figure 37 : Initialization Done Interrupt Handling Submodule (VLANCE. LEXCEPT. IDON).


START
INITIAL-IZATION DONE BIT SET ?
YES
NO
IDONSUB
SERVICE
IN DIAGNOSTIC MODE ?
NO
YES
SET TEST FINISHED BIT
INTERRUPT ENABLE THE LANCE
RETRIEVE CONTENTS OF OLD REGISTERS
RTE

The diagnostic bit in the ring management register is then checked. If the VLANCE is in the diagnostic routine, the test done bit is set. The test done bit indicates a message has been loopbacked and checked and the next portion of the diagnostics may begin. This bit is checked after both a completed interrupt service routine and a Return From Exception.

At this point, all of the VLANCE's interrupting conditions have been checked and serviced. During the exception processing, the VLANCE's interrupting capabilities are disabled because another interrupt from the VLANCE should not be performed while the first one is being serviced. Since servicing is complete the Interrupt Enable bit can again be set again in CSR0.

The module is concluded by moving the contents of the old registers off the stack and back into the register locations. A Return From Exception is then executed.

7.4. MESSAGE INTERRUPT SOFTWARE MODULE

The Message Interrupt Software Module, shown in figure 38, is the routine that services the "push-button" message generator used in Application Note's hypothetical design. This push button is used for demonstration and debugging. The software description is included in this Application Note because the service routine is general enough to be adapted to any type of interrupt. The source of the interrupt may be anything, such as another processor controlling a terminal, a file server, or, as in this case, a push-button switch.

There is, however, one major difference between this module and one used in real life. In addition to servicing the transmit message, this module also generates the message.

This module's basic function is to fill up as many transmit buffers as needed for the outgoing message, and to set the status information in the message descriptor accordingly.

The message interrupt is hardware set at interrupt level 5. Upon interrupt, the processor autovectors to memory location $0B00, the location of this module.

The software first saves the contents of the old registers and then clears the registers for use. Next, the message data length is retrieved from DLENGTH in the Ring Management Area. This message data length is manually written into before the message button is depressed.

Next, the Descriptor Count is checked for available transmit buffers. If none are available, an error subroutine is called, which sets an error bit in the status area, and the routine is over. If a buffer is available, the descriptor count is decremented and the Next Descriptor Ring address is moved into a register from the Ring Management Area. The Next Transmit Descriptor points to the descriptor of the next transmit buffer. The transmit buffer address is then moved in from the descriptor and reformatted for 68000 compatability.

A message data word is then generated. The message consists of byte values ranging from zero to the hex value specified by DLENGTH. In other words, if the value $20 is written into DLENGTH and a message is generated, it consists of 32 bytes. The value of the first byte is $00, the second, $01, the third, $03, the 32nd, $1F, and so forth.

Each time a data word is generated, the transmit buffer address is incremented along with the buffer and message counts. The buffer count records how many bytes have been moved into the present transmit buffer. If the transmit message length exceeds the buffer size, more then one transmit buffer is used. The message count keeps track of the number of generated message words. If this amount equals DLENGTH, the message generation part of this module is complete.

If the transmit buffer is full, and more data words need to be generated, the transmit descriptor for that buffer is updated, and the rest of the message is placed in the next available transmit message buffer. In this case, the first transmit descriptor has the start-of-packet bit set, but the end-of-packet bit is not set. Figure 39 shows a flow chart of the transmit descriptor update submodule.

Once the entire message has been moved into the message buffer, or buffers, the last transmit descriptor is updated with the status information, along with a 2's complement of the number of bytes placed in its buffer. Finally, the old registers are retrieved, and a Return From Exception is executed.

Figure 38 : Message Interrupt Software Module (VLANCE. MESSAGE).

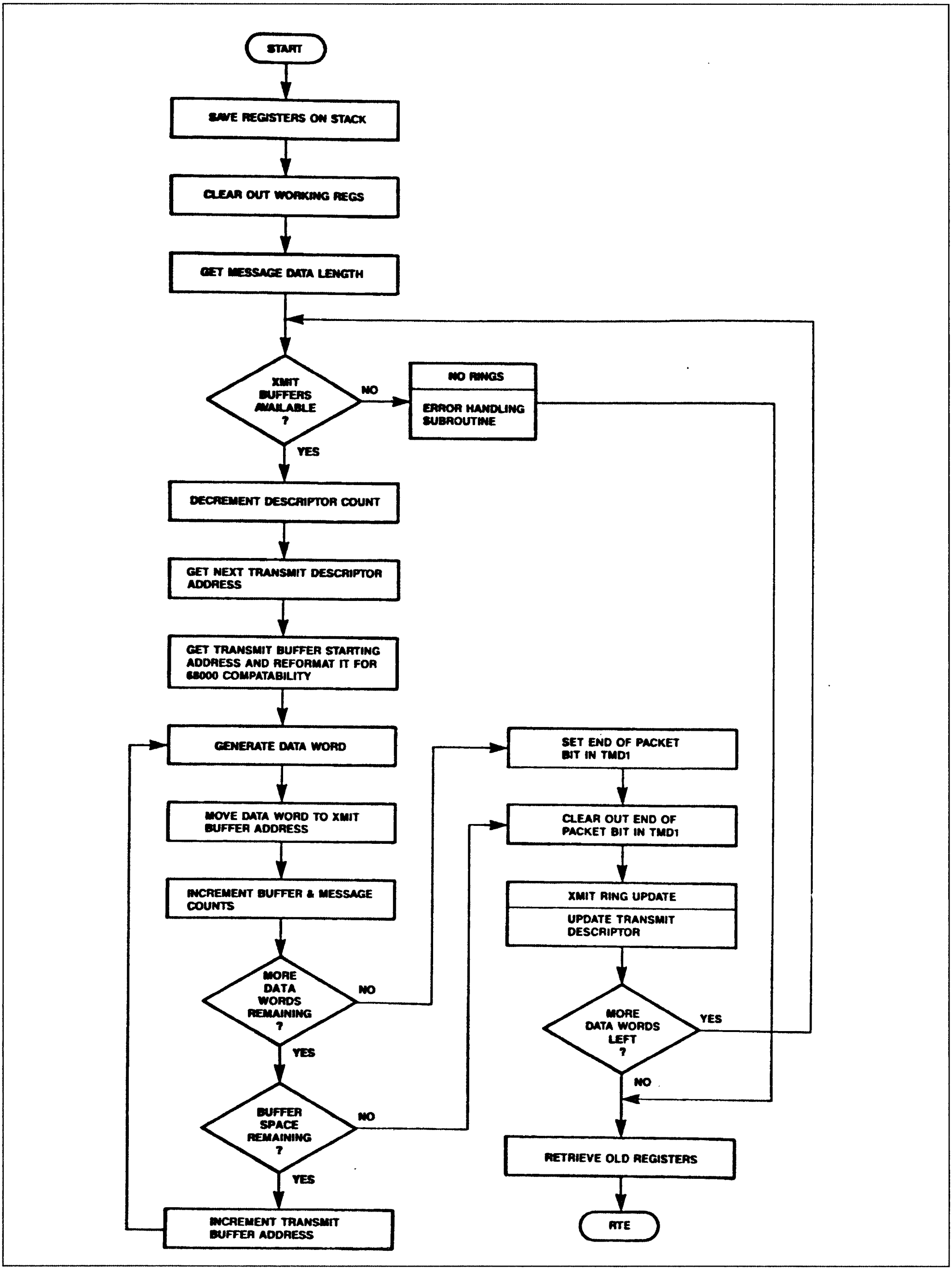
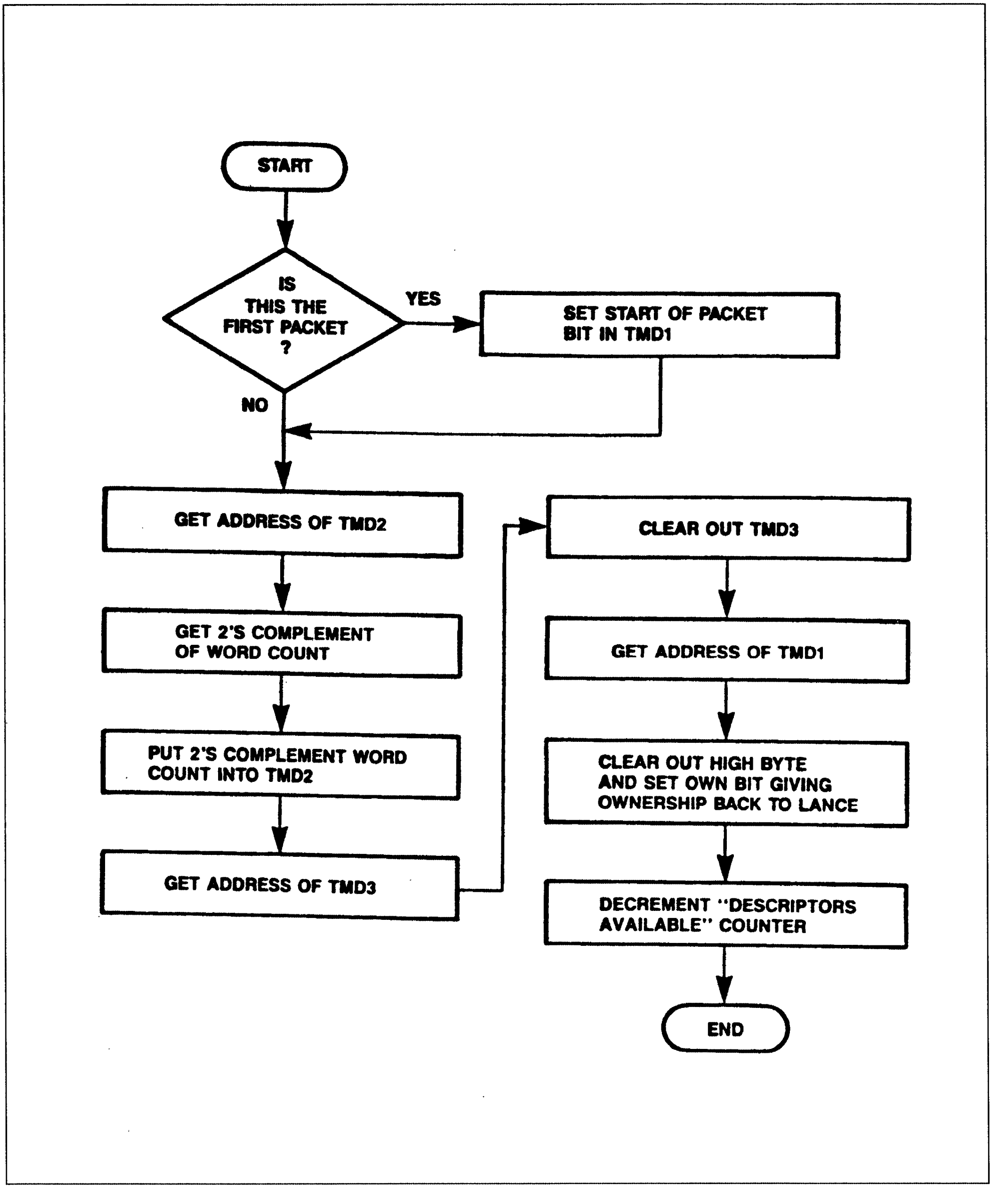

START
SAVE REGISTERS ON STACK
CLEAR OUT WORKING REGS
GET MESSAGE DATA LENGTH
XMIT BUFFERS AVAILABLE ?
NO
NO RINGS
ERROR HANDLING SUBROUTINE
YES
DECREMENT DESCRIPTOR COUNT
GET NEXT TRANSMIT DESCRIPTOR ADDRESS
GET TRANSMIT BUFFER STARTING ADDRESS AND REFORMAT IT FOR 68000 COMPATABILITY
GENERATE DATA WORD
MOVE DATA WORD TO XMIT BUFFER ADDRESS
INCREMENT BUFFER & MESSAGE COUNTS
MORE DATA WORDS REMAINING ?
NO
YES
BUFFER SPACE REMAINING ?
NO
YES
INCREMENT TRANSMIT BUFFER ADDRESS
SET END OF PACKET BIT IN TMD1
CLEAR OUT END OF PACKET BIT IN TMD1
XMIT RING UPDATE
UPDATE TRANSMIT DESCRIPTOR
MORE DATA WORDS LEFT ?
YES
NO
RETRIEVE OLD REGISTERS
RTE

Figure 39 : Transmit Ring Update Submodule (MESSAGE. XMIT RING UPDATE).


START
IS THIS THE FIRST PACKET ?
YES
SET START OF PACKET BIT IN TMD1
NO
GET ADDRESS OF TMD2
GET 2'S COMPLEMENT OF WORD COUNT
PUT 2'S COMPLEMENT WORD COUNT INTO TMD2
GET ADDRESS OF TMD3
CLEAR OUT TMD3
GET ADDRESS OF TMD1
CLEAR OUT HIGH BYTE AND SET OWN BIT GIVING OWNERSHIP BACK TO LANCE
DECREMENT "DESCRIPTORS AVAILABLE" COUNTER
END

SGS-THOMSON MICROELECTRONICS

APPLICATION NOTE

StarLAN SYSTEM SOLUTION

INTRODUCTION

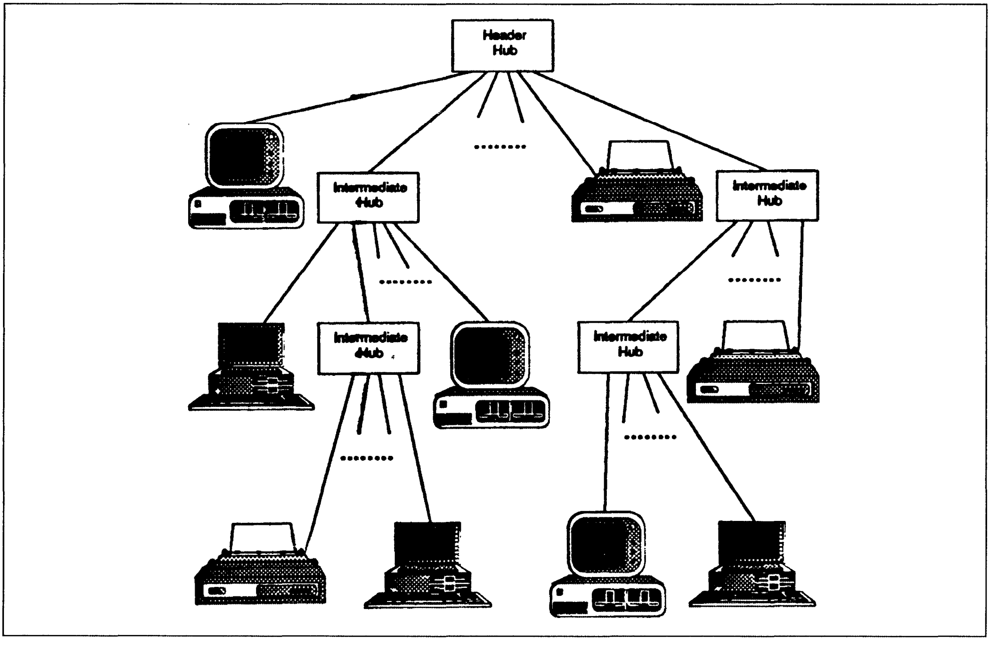
SGS-THOMSON Microelectronics is the only company to manufacture a family of devices for StarLAN and thus provide a complete StarLAN System Solution. Included in this StarLAN solution are the MK5032 - a Variable Bit-Rate LAN (Local Area Network) Controller, the MK5033 - a generic Manchester Encoder/Decoder Station chip, and the MK5030 StarLAN Hub. Each chip has a specific purpose and is of course completely compatible with the other chips. With only the addition of simple line driver and receiver circuitry, a StarLAN solution can be easily implemented.

This application brief is dedicated to showing how readily these SGS-THOMSON devices work together and how easy these chips make it for the user to implement a StarLAN solution. Not only will the functionality of, and the essential connections between, the LAN controller, the station chip, and the hub be discussed and shown, but also StarLAN will be described in general terms. In other words, this document is designed to be conceptual as well as realistic.

STARLAN SYNOPSIS

StarLAN, as the name suggests, uses a star topology. The nodes are at the extremities of a star and the central point is called a hub. There can be more than one hub in a network. The hubs are connected in a hierarchical fashion resembling an inverted tree as shown in figure 1. The hub at the top of the network is called the header hub (HH) and the others are called intermediate hubs (IH). Topologically, this entire network of nodes and hubs is equivalent to one where all the nodes are connected to a single hub. Also StarLAN does not limit the number of nodes or hubs at any given level.

Figure 1 : StarLAN Topology.



Technically, StarLAN is a 1 Mb/s network based on the CSMA/CD access algorithm (Carrier Sense, Multiple Access with Collision Detection). CSMA/CD means before a node will transmit a packet of data, it must first detect if there is an active carrier already on the line. In other words, the node listens first. This is known as "carrier sense". "Multiple access" means that multiple users can access the LAN (at different times of course), and that many can receive at the same time. Consequently, there will be occasions in which two or more nodes sense an inactive carrier at nearly the same time, and attempt to transmit data. Collisions of data may occur. When a collision happens the hub will detect the presence of a collision, and broadcast a CP (Collision Presence) signal to all nodes. This is "collision detection". When a transmitting node receives a CP, it will terminate its transmission and retry later at a randomly determined time.

For the transmission medium, StarLAN uses standard, unshielded, twisted pair telephone wiring. Typically, the telephone wiring connects each desk to a wiring closet in a star topology (from which the IEEE Task Force working on the standard derived the name StarLAN in 1984). In fact, telephone and StarLAN wiring can coexist in the same twisted pair bundle connecting the desk to the wiring closet. Abundant quantities of unused phone wiring exist in most office environments, particularly in the U.S. The StarLAN concept of wiring and networking concepts was originated by AT&T Information Systems.

StarLAN has many advantages and is attractive for many reasons. First and most obvious is the fact that StarLAN uses telephone grade twisted pair wire for its transmission medium. Telephone wire is very economical to buy and install. Second is the installed base of redundant telephone wiring in most buildings, enabling StarLAN to run along side the telephone wiring. Third is the fact that buildings are designed for star topology wiring. Fourth is the availability of low cost VLSI LAN controllers like the MK5032, station chips like the MK5033, and hub chips like the MK5030.

THE MK5032 LAN CONTROLLER

An attractive feature of StarLAN is the availability of the MK5032, a VLSI LAN controller designed to meet the needs of a StarLAN node. The MK5032 performs many necessary tasks that make it an ideal StarLAN controller. It is a fully compatible IEEE 802.3 CSMA/CD controller and is also fully compatible with the StarLAN standard. Basically the MK5032 simplifies the interfacing of a microcomputer or a minicomputer to a local area network. The parallel interface of the MK5032 has been designed to be "friendly", or easy to interface, to many popular microprocessors such as the 68000, 8086, Z8000, and the LSI-11. Furthermore, the MK5032 with its DMA channel allows for flexibility and speed of communication between the MK5032 and the host microprocessor through shared memory locations.

The MK5032 supports system clocks from 1 to 10MHz and data rates from 1 to 10Mb/s. A unique feature of the MK5032 is that it can do its DMA cycles at 10MHz and transmit on or receive from the line via the station chip at the StarLAN standard of 1Mb/s. In addition to handling data rates of 1 and 10Mb/s, it also supports data rates of 1.25, 1.67, 2.50, and 5.00Mb/s if the clock rate is 10MHz. The MK5032 has six divide-by modes and they are divide-by 1, 2, 4, 6, 8, and 10. This feature is configurable by the user and makes the MK5032 a very versatile device.

THE MK5033 STARLAN STATION CHIP

The MK5033 is a general purpose Manchester encoder/decoder that incorporates several features that make it an ideal StarLAN station chip. It provides all the logic necessary to interface a LAN controller to a StarLAN network. The MK5033 conforms to all StarLAN standards, but in addition, can support data rates up to 2.66Mb/s. Basically it encodes data from a controller chip into Manchester or Differential Manchester data. It decodes Manchester or Differential Manchester data from the line transceiver and produces NRZ data and clock for the controller chip. Also, it detects collisions and signals the controller chip that a collision has occurred.

Since the MK5033 is a generic Manchester encoder/decoder, it can be used with different controllers. It has an on chip crystal oscillator and can work in a 16, 10, 8, or 6X operation mode. Besides being compatible with the MK5032, the MK5033 is compatible with the MK68590 LANCE and the Intel 82586 & 82588. Also, the MK5033 supports different network topologies such as Star, Bus, and Point-to-Point. In addition, the MK5033 has several enhancements for StarLAN and Multi-Point-Extension (MPE) StarLAN. These include automatic compensation for wiring reversal, an echo timer, external collision detect, a watchdog timer, and heartbeat, among others.

THE MK5030 HUB

The hub is the point of concentration in StarLAN. All the nodes transmit to the hub and receive from the hub. The MK5030 provides all the digital logic necessary to implement a twelve port hub. Essentially, it is a collision detector and a multi-port repeater. The MK5030 conforms to all StarLAN standards, but in addition, can support data rates up to 2Mb/s.

The MK5030 has many attractive features and one of the most unique is the fact that it allows for wiring line polarity reversal. This automatic compensation definitely reduces installation costs because when installing telephone wiring. It is often difficult and expensive to maintain proper polarity on the wire pairs. Also, the MK5030 eases network administration by providing status display for line activity and jabber status on a per port basis. In addition, the APG (Automatic Preamble Generation) option allows the hub to replace preamble bits lost when going through hub stacks. Furthermore, the optional minimum-frame-length-enforcement permits reliable operation in an MPE (Multi-Point-Extension) environment. Finally, the retiming function using the serial FIFO may be optionally bypassed. In this option, there are no bit delays through the hub, there are only gate delays. In this way, two hubs could be stacked, but appear as only one hub to the network.

The MK5030 has six pin selectable modes plus a test mode, and these are summarized below :

MODE 0 Logic(1) = Normal Operation
(pin 47) Logic(0) = Test Mode

MODE 1 Logic(1) = MK5030 is an IH
(pin 46) Logic(0) = MK5030 is an HH

MODE 2 Logic(1) = Int. Squelch for Input Carriers
(pin 45) Logic(0) = Ext. Pulse Stretcher Required

MODE 3 Logic(1) = Retimer Circuit Enabled
(pin 44) Logic(0) = Retimer Circuit Disabled

MODE 4 Logic(1) = Carrier inputs are active high.
(pin 43) Logic(0) = Carrier inputs are active low.

MODE 5 Logic(1) = Early preamble is enabled.
(pin 42) Logic(0) = Auto preamble is disabled.

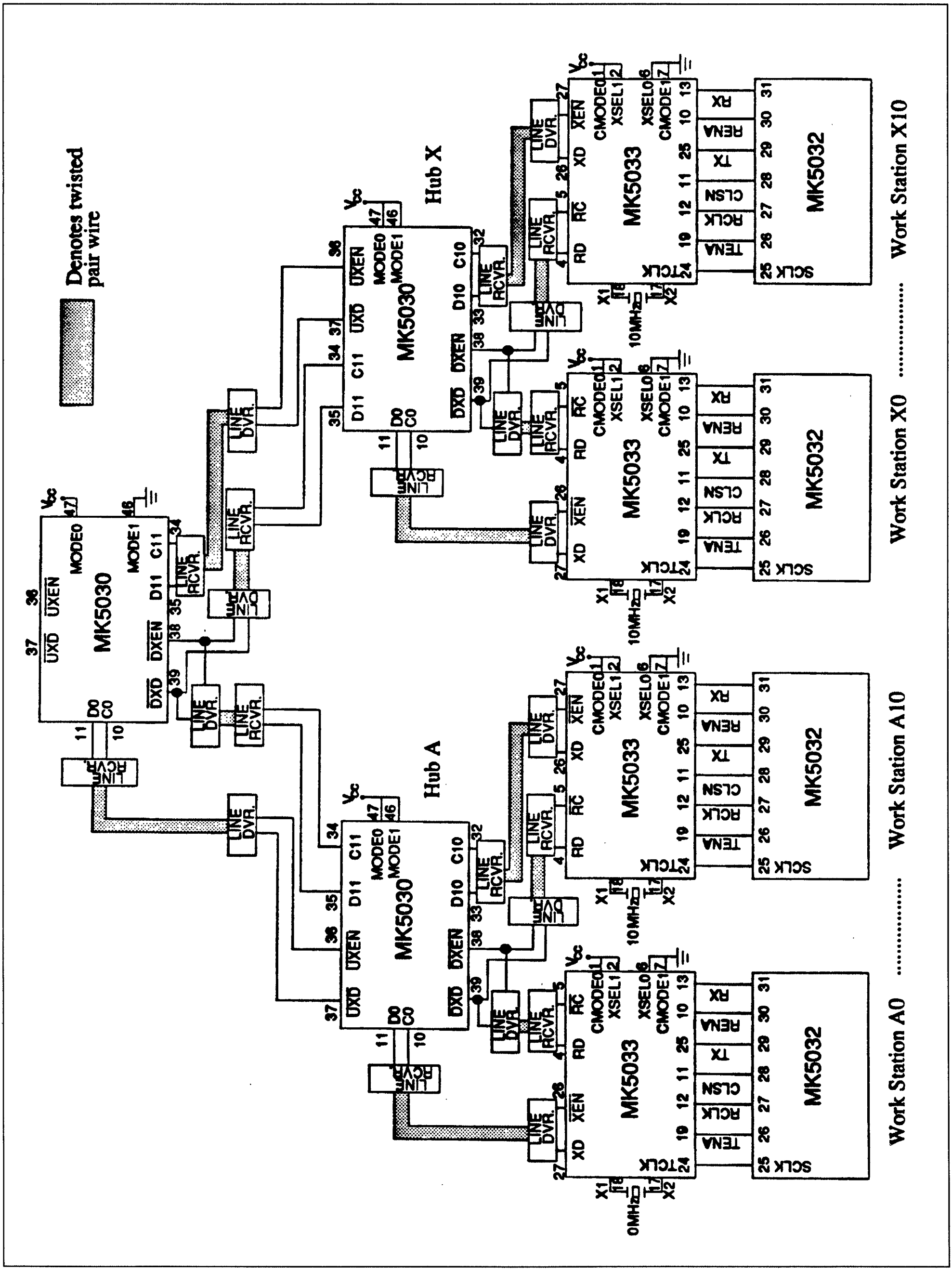
MODE 6 Logic(1) = Min. frame length enforced.
(pin 41) Logic(0) = No min. frame length enforced.

Because of these options, the MK5030 is a very versatile chip.

The MK5030 has an upstream and a downstream signal processing unit. The upstream unit has 11 + 1 inputs and 1 signal output and the downstream unit has 1 input and 12 output signals. For the upstream unit, 11 + 1 means 11 channels are always available and 1 more may be available if the hub is in header hub mode. In other words, in the intermediate hub mode, 1 channel (channel 11), is reserved for the downlink connection from the next higher hub. The inputs to the upstream unit come from the nodes or from the intermediate hubs and its output goes to a higher level hub. The downstream unit is connected the other way around : input from an upper level hub and the outputs to nodes or lower level intermediate hubs.

The functions performed by the upstream unit of the MK5030 are basically collision detection, Collision Presence signal generation, signal retiming, and execution of the jabber function. The collision detection in the hub is done by sensing the activity on the inputs. If there is activity (or transmissions) on more than one input, it is assumed that more than one node is transmitting. This is a collision. If a collision is detected, a special signal called the Collision Presence signal is generated by the hub and sent out as long as activity is sensed on any of the input lines. This signal is interpreted by every node as an occurrence of a collision. When there is activity only on one input, signal retiming is performed by the hub. The signal is cleaned up of any jitter and sent out. The jabber function performed by the hub isolates network failures. If a node transmits for greater than 27ms, (which allows twice its normal maximum frame size), the hub exercises its jabber function to disable the node from interfering with any other transmissions. It sends a CP (Collision Presence) signal. If that node continues to transmit for up to a total time of 54ms, the hub will disable its input from that node. In this manner, the faulty station is removed from the network and the node is said to be "jabbed". The jammed station can be let back onto the network, but only after it goes idle. Thus the MK5030 automatically takes care of the jabber function and makes maintenance of the network easy. Like the upstream unit, the downstream unit has its own particular function. The downstream unit essentially just retimes the signal received at the input, and sends it to all outputs. Figure 2 shows the essential connections between the MK5032 LAN Controller, the MK5033 Station chip, and the MK5030 Hub.

Figure 2 : Essential Connections between StarLAN Devices.



Figures 3A and 3B show the typical line driver and receiver circuitry needed to properly execute StarLAN. Every port must have this circuitry. SN75172 and SN75173 devices are commonly used for the drivers and receivers, respectfully. Physically, each input to the receivers and each output of the drivers consist of one twisted pair wire carrying a differential signal. The drivers and receivers associated with the MK5033 Station chip are located on the same board as the MK5033 and the MK5032 whereas the drivers and receivers associated with the MK5030 Hub are located in the wiring closet with the MK5030. The main advantage of having external line drivers and receivers as opposed to on-chip line drivers and receivers is that in the event of catastrophic conditions on the line, just the line drivers and receivers need be replaced instead of the more expensive station chips.

CONCLUSION

The MK5032 LAN Controller, the MK5033 Station chip, and the MK5030 Hub are fully compatible with one another, and make it easy to implement a StarLAN network. With the MK5032, the MK5033, and the MK5030, SGS-THOMSON Microelectronics provides a StarLAN System Solution.

Figure 3A : Driver Circuitry.

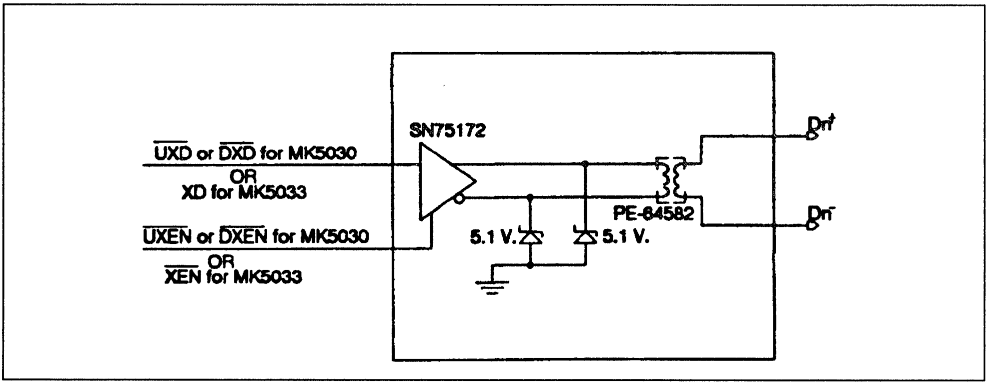
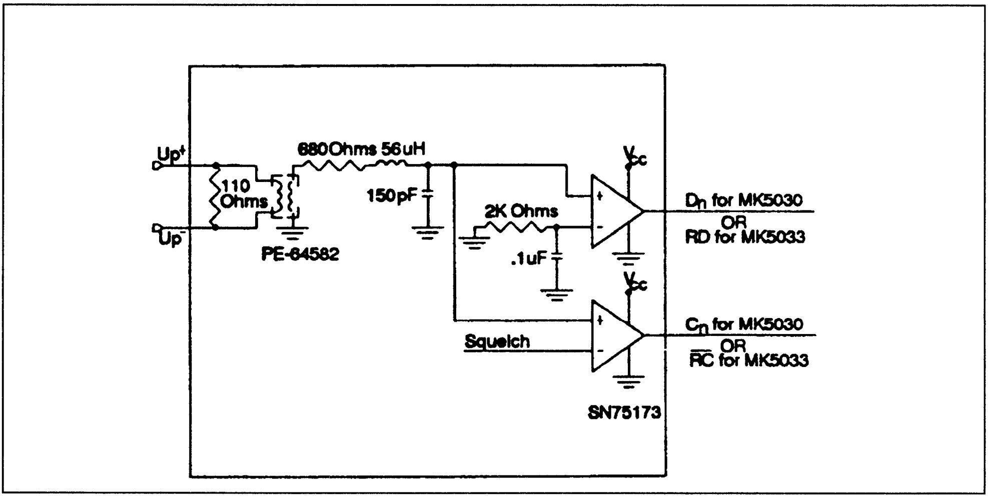


Figure 3B : Receiver Circuitry.



SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

MANCHESTER ENCODER/ DECODER INTERFACE SOLUTIONS

INTRODUCTION

SGS-THOMSON Microelectronics provides two LAN Controllers and four Manchester Encoder/Decoder chips to meet system designer's needs. The MK5032 Variable Bit-Rate LANCE and the MK68590 LANCE^{TM} are the LAN controllers. The Manchester encoder/decoder devices are the MK5033, the MK5035, the MK50351, and the MK68591/2 SIA Serial Interface Adapter). This wide selection makes implementation of Ethernet and StarLAN easy. Also, the versatility of these devices allows them to be used in many different applications.

PURPOSE

This application brief is designed to show the mainstream connections between the appropriate LAN controllers and the appropriate Manchester encoder/decoders. Since three of the Manchester encoder/decoders are fully compatible with Intel LAN controllers, these interfaces will also be shown. In all, 12 different interfaces will be presented.

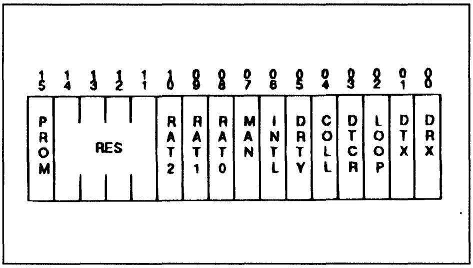
THE LAN CONTROLLERS

The MK5032 VLANCE and the MK68590 LANCE are IEEE 802.3 CSMA/CD LAN controllers designed to simplify the interfacing of a microcomputer or a mini-computer to a local area network. These devices are identical in every facet except the MK5032 is a true variable bit-rate controller. Therefore, a discussion of their clock and data rate operation is warranted.

MK5032 CLOCK AND DATA RATE OPERATION

The MK5032 VLANCE supports system clocks from 1 to 10MHz and data rates from 1 to 10Mb/s. The big advantage of the MK5032 is that the data rate and system clock are independent. For example, the MK5032 can perform DMA transfers at 10MHz and transmit and receive data at 1Mb/s. This is accomplished because the MK5032 can divide the system clock rate to give the desired data rate. It has six divide-by modes - divide-by 1, 2, 4, 6, 8, and 10. The divisor is selected by programming bits 10 - 07 in the mode register in the initialization block. Figure 1 shows this 16-bit wide mode register.

Figure 1 : Mode Register for MK5032.



The bits involved in selecting the divisor are bits 10 - 07. Bits 10 - 08 are known as the 'RAT' (rate) bits. Bit 07 is known as the MAN (Manchester) bit. Table 1 shows how to configure these 'RAT' bits in order to select the desired divisor and thus desired data rate. For the MAN bit to be active, at least one of the 'RAT' bits must be set to a '1'. When the MAN bit is set and at least one 'RAT' bit is set, the serial transmit data stream will be Manchester encoded data. (When using SGS-THOMSON Microelectronics Manchester encoder/decoders, the MAN bit does not need to be set). Consult the MK68590 Technical Manual for further details on the other bits in the mode register.

Table 1 : Configuration of "RAT" Bits for Divisor Selection.

RAT2	RAT1	RAT0	Divisor
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	6
1	0	0	8
1	0	1	10

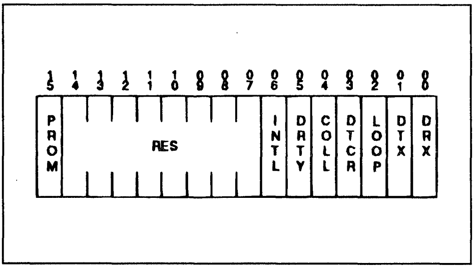
LANCE is a trademark of SGS-THOMSON Microelectronics.

MK68590 CLOCK AND DATA RATE OPERATION

The MK68590 LANCE has dependent system clock and data rate operation. The speed at which the LANCE transmits and receives data is the same speed at which it does DMA transfers. The LANCE supports system clock rates from 1 to 10MHz and data rates from 1 to 10Mb/s. Designed for Ethernet, the LANCE performs at optimum efficiency with a 10MHz system clock and 10Mb/s data rate. In a StarLAN application, the LANCE performs DMA transfers at 1MHz and transmits and receives data at 1Mb/s.

The mode register in the initialization block used for the MK68590 LANCE is shown in figure 2. It is identical to the mode register shown in figure 1 except bits 10 - 07 are reserved like bits 14 - 11. Since the MK68590 Technical Manual dictates these bits be set to zero for normal operation, the MK5032 can be "plugged-in" for the MK68590 and the MK5032 will be in Ethernet mode. Therefore, no software changes are necessary for the MK5032 in Ethernet mode if the reserved bits have previously been written as '0's'.

Figure 2 : Mode Register for MK68590.



THE MANCHESTER ENCODER/DECODERS

To interface these LAN controllers to the physical transmission media, SGS-THOMSON Microelectronics offers four Manchester encoder/decoder devices. Each device has different features and options that make it better suited for certain applications. The following is a summary of the operation of these devices.

MK5033 OPERATION

The MK5033 has the option of four different modes of operation. In addition to a production test mode, the MK5033 has three other modes which enable it to be used with different LAN controllers. By connecting the CMODE0 pin and the CMODE1 pin in the appropriate way, the MK5033 can easily interface with the MK5032 VLANCE, the MK68590 LANCE, and the Intel 82586 and 82588. These "controller mode" pins, when configured correctly, correct for the proper polarity needed for these different interfaces. Table 2 shows how to configure the CMODE0 and CMODE1 pins to select the desired mode of operation.

Table 2 : MK5033 "CMODE" Pin Values for Controller Selection.

CMODE0	CMODE1	Mode of Operation
0	0	Production Test Mode
0	1	Intel 82586/82588
1	0	MK5032
1	1	MK68590

Another option offered by the MK5033 is the selectivity of the data rate. This optional data rate selection is possible because the MK5033 has an internal clock divisor which divides the crystal (clock) rate by a factor of 16, 10, 8, or 6. By connecting the XSEL0 pin and the XSEL1 pin the appropriate manner, the desired clock divisor can be selected. Table 3 shows how to configure the XSEL0 and XSEL1 pins to give the desired clock divisor. Typically a 6.0, 8.0, 10.0, or 16.0MHz crystal (clock) rate is applied to the MK5033. As for the data rate, the MK5033 can support data rates up to 2.66Mb/s.

Table 3 : MK5033 "XSEL" Pin Values for Divisor Selection.

XSEL0	XSEL1	Divisor
0	0	16
0	1	10
1	0	8
1	1	6

MK5035 AND MK50351 OPERATION

The MK5035 and MK50351 are subsets of the MK5033 designed for more specific applications. As a result, they are very similar devices. Both have a CMODE pin and a XSEL pin which, when properly configured, allow LAN controller and divisor selection, respectively. Again, the CMODE pin, when configured properly, corrects for the proper polarity needed for different interfaces. The MK5035 is compatible with the MK68590 LANCE and the Intel 82586 and 82588. The MK50351 is compatible with the MK5032 VLANCE and the Intel 82586 and 82588. Tables 4 and 5 show how to configure the CMODE pin and the XSEL pin for the MK5035 in order to select the desired mode of operation and clock divisor, respectively.

Table 4 : MK5035 CMODE Pin Values for Controller Selection.

CMODE	Mode of Operation
0	Intel 82586/82588
1	MK68590

Table 5 : MK5035 "XSEL" Pin Values for Divisor Selection.

XSEL	Divisor
0	8
1	6

Tables 6 and 7 show how to configure the CMODE pin and the XSEL pin for the MK50351 in order to select the desired mode of operation and clock divisor, respectively.

Table 6 : MK50351 CMODE Pin Values for Controller Selection.

CMODE	Mode of Operation
0	Intel 82586/82588
1	MK5032

One difference between these parts is that the MK5035 has clock divisors of 8 and 6 whereas the MK50351 has clock divisors of 10 and 8. Typically a 6.0 or 8.0MHz crystal (clock) rate is applied to the MK5035 and an 8.0 or 10.0MHz crystal (clock) rate is applied to the MK50351. Both of these devices support data rates up to 2.66Mb/s.

Table 7 : MK50351 XSEL Pin Values for Divisor Selection.

XSEL	Divisor
0	8
1	10

MK68591/2 OPERATION

The MK68591/2 SIA uses a 20MHz fundamental mode crystal oscillator for its basic timing reference. It divides this clock by 2 to create transmit clock (TCLK) reference which in turn is used by the LANCE and VLANCE to drive their internal state machines. Also, TCLK is used by the receive section of the SIA. Designed with Ethernet in mind, the SIA provides the interface between the TTL logic environment of the VLANCE and LANCE and the differential signaling environment in the transceiver AUI cable.

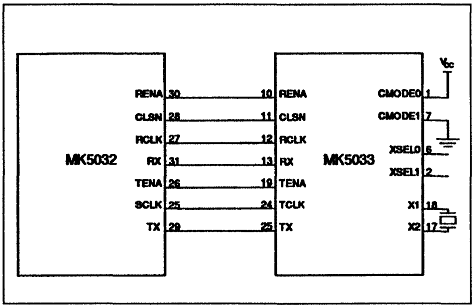
LAN CONTROLLER - MANCHESTER ENCODER/DECODER INTERFACES

The following is a summary of interfaces between the appropriate LAN controllers and the appropriate Manchester encoder/decoders. Figure 1 and table 1 should be referred to when MK5032 divisor selection is mentioned. Likewise table 3, 5, or 7 should be referred to when Manchester encoder/decoder divisor selection is mentioned.

THE MK5032 - MK5033 INTERFACE

Figure 3 shows the interface between the MK5032 VLANCE and the MK5033. In this configuration, the MK5032 performs DMA transfers at the crystal (clock) rate of the MK5033 ; therefore, either a 6.0, 8.0, or 10.0MHz crystal (clock) rate should be applied to the MK5033. For example, a 10MHz crystal (clock) can be applied to the MK5033 and XSEL0 can be set to '1' and XSEL1 can be set to '0' giving a crystal (clock) divisor of 8. This results in a data rate of 1.25Mb/s. For the MK5032 the appropriate bits in the mode register in the initialization block must be set to alert the MK5032 of the desired divisor of 8. As a result, the MK5032 will transmit and receive data at 1.25Mb/s and perform DMA transfers at 10MHz.

Figure 3 : MK5032-MK5033 Interface.

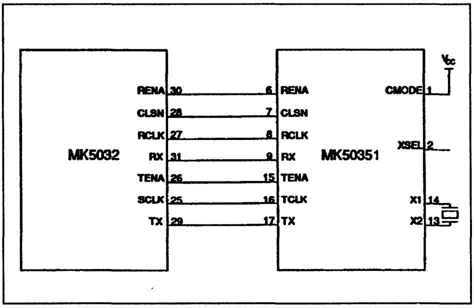


Of course the interface between the MK5032 and the MK5033 can be configured for StarLAN. In this application, the MK5032 can perform DMA transfers at 10MHz while transmitting and receiving at the StarLAN standard of 1Mb/s. The appropriate divisors must be selected.

THE MK5032 - 50351 INTERFACE

Figure 4 shows the interface between the MK5032 VLANCE and the MK50351. In this interface, the MK5032 again performs DMA transfers at the crystal (clock) rate of the Manchester encoder/decoder ; therefore, either an 8.0 or 10.0MHz crystal (clock) rate should be applied to MK50351. In a typical StarLAN application, a 10MHz crystal (clock) rate is applied to the MK50351 and the XSEL pin is set to a '1' giving a clock divisor of 10. The corresponding data rate is 1Mb/s. For the MK5032, the appropriate bits in the mode register in the initialization block are set to alert the MK5032 of the desired divisor of 10. As a result, the MK5032 performs DMA cycles at 10MHz and transmits and receives at the StarLAN standard of 1Mb/s.

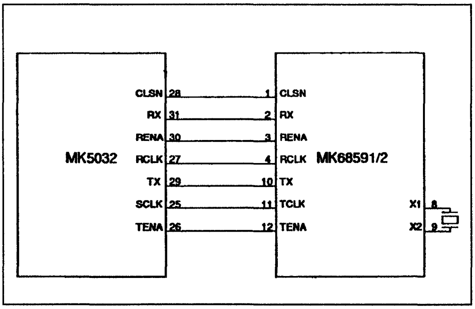
Figure 4 : MK5032-MK50351 Interface.



THE MK5032 - MK68591/2 INTERFACE

Besides being fully compatible with the StarLAN standard, the MK5032 is also completely compatible with the Ethernet standard. Therefore, the MK5032 can be used with the MK68591/2 SIA to implement Ethernet. Figure 5 shows the interface between the MK5032 and the MK68591/2. In this application, the 20MHz crystal applied to the SIA is divided by two. This 10MHz signal is fed to the MK5032 resulting in a 10MHz system clock (SCLK) rate for the MK5032. By writing '0's' to bits 10 - 07 in the mode register for the MK5032, the appropriate divisor of 1 can be selected to give the desired data rate of 10Mb/s. In this manner, the MK5032 performs DMA transfers at 10MHz and transmits and receives at 10Mb/s.

Figure 5 : MK5032-MK68591/2 Interface.

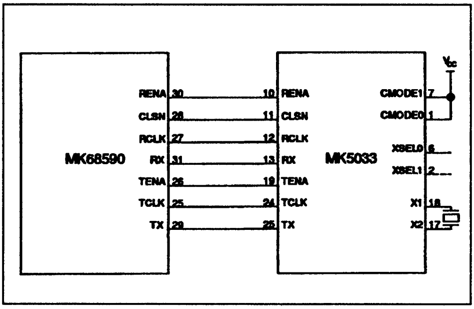


THE MK68590 - MK5033 INTERFACE

Figure 6 shows the interface between the MK68590 LANCE and the MK5033. Since the LANCE is not a true variable bit-rate controller, it performs DMA cycles at the speed at which it transmits and receives. For example, if a 16MHz crystal (clock) is applied to the MK5033 and the XSEL0 pin is set to '1' and the XSEL1 pin is also set to '1' giving a clock divisor of 6, the corresponding data rate will be 2.66Mb/s. As a result, the LANCE performs DMA cycles at 2.66MHz as well as transmits and receives data at 2.66Mb/s.

Of course this configuration can also be used for StarLAN. In this application, the appropriate divisor must be selected by properly configuring the XSEL0 and XSEL1 pins. The LANCE performs DMA transfers at 1MHz and transmits and receives data at the StarLAN standard of 1Mb/s.

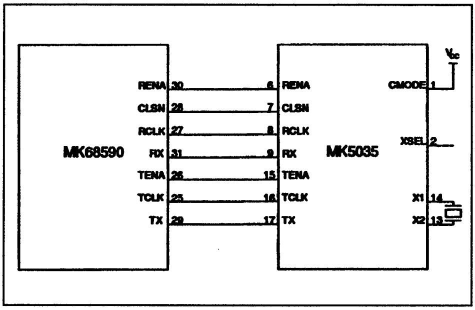
Figure 6 : MK68950-MK5033 Interface.



THE MK68590 - MK5035 INTERFACE

Figure 7 shows the interface between the MK68590 LANCE and the MK5035. This interface is similar to the LANCE-5033 interface in the fact that the LANCE performs DMA cycles at the same speed at which it transmits and receives. In a StarLAN application with this configuration, the LANCE performs DMA transfers at 1MHz and transmits and receives at the StarLAN standard of 1Mb/s. Again, the appropriate divisor must be selected.

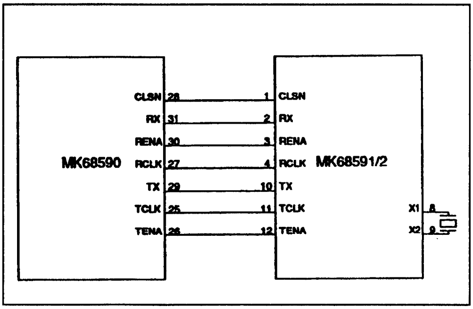
Figure 7 : MK68590-MK5035 Interface.



THE MK68590 - MK68591/2 INTERFACE

Figure 8 shows the interface between the MK68590 and the MK68591/2. This interface was designed for Ethernet. In this alignment, the 20MHz crystal applied to the SIA is divided by two. This 10MHz signal is fed to the LANCE giving the LANCE a 10MHz system clock (TCLK). This causes the LANCE to also transmit and receive at the Ethernet standard of 10Mb/s.

Figure 8 : MK68590-68591/2 Interface.



THE INTEL 82586/88 - MK5033/35/351 INTERFACES

Figures 9 - 14 show the Intel 82586/88 - MK5033/35/351 interfaces. These interfaces are selectable by the CMODE0 and CMODE1 pins on the MK5033 and the CMODE pin on the MK5035 and the MK50351. Also, the XSEL0 and XSEL1 pins on the MK5033 and the XSEL pin on the MK5035 and the MK50351 allow for easy interface. All of these interfaces can be easily configured to execute StarLAN.

Figure 9 : 82586-MK5033 Interface.

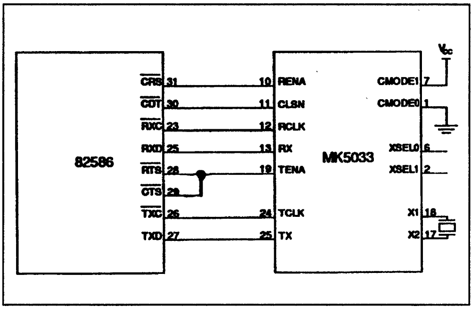


Figure 10 : 82586-MK5035 Interface.

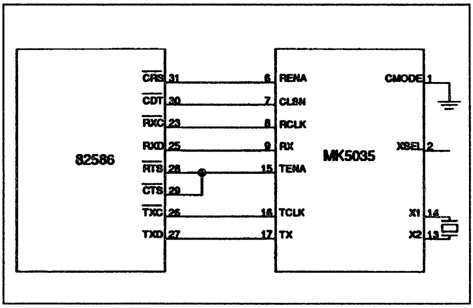


Figure 13 : 82588-MK5035 Interface.

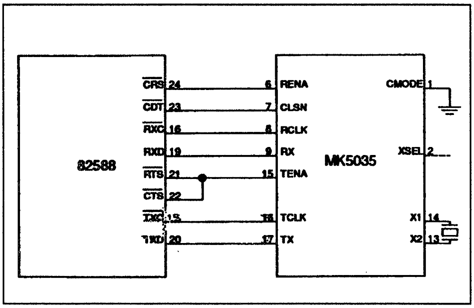


Figure 11 : 82586-MK50351 Interface.

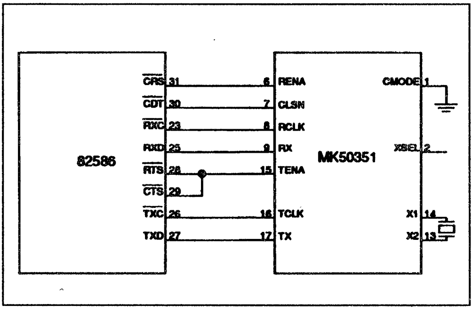


Figure 14 : 82588-MK50351 Interface.

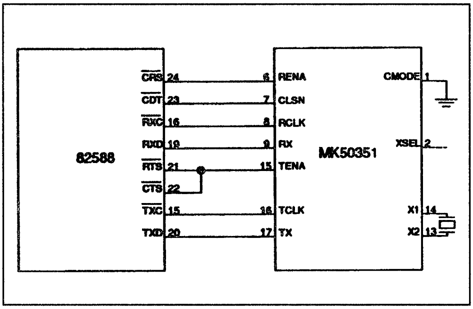


Figure 12 : 82588-MK5033 Interface.

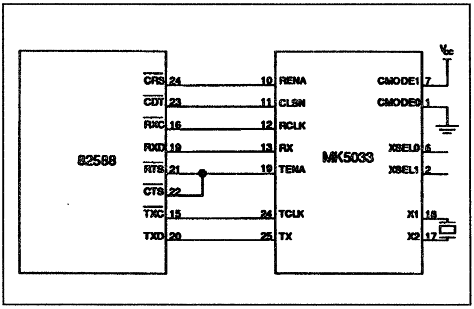
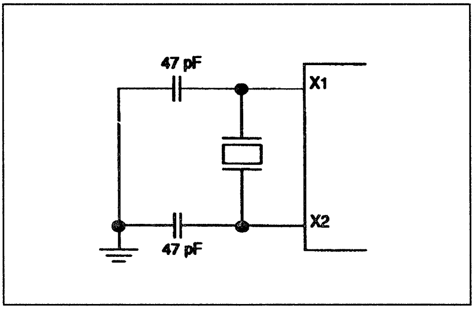


Figure 15 : Crystal Connection.



CRYSTAL CONNECTION

All of the Manchester encoder/decoder devices discussed in this application brief use a fundamental mode crystal oscillator for the basic timing reference. Figure 15 shows typical crystal interface circuitry.

If a crystal is not connected to the X1 and X2 pins, a square wave with TTL levels can be connected to X1. X2 should be left disconnected in this case.

CONCLUSION

With the selection of the LAN controllers and the Manchester encoder/decoder devices that SGS-THOMSON Microelectronics offers, system designers have a wide range of flexibility. Ethernet and StarLAN are made easy to implement. In addition, the versatility of these chips allow them to be designed-in in many applications.

WIDE AREA NETWORK DEVICES

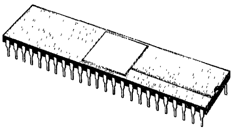
SGS-THOMSON
MICROELECTRONICS

MK5025

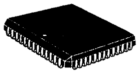
CCITT X.25 LINK LEVEL CONTROLLER

PRELIMINARY DATA

- CMOS
- FULLY COMPATIBLE WITH BOTH 8-BIT OR 16-BIT SYSTEMS
- SYSTEM CLOCK RATE TO 10MHz
- DATE RATE UP TO 7Mbps, WITH A 64-BYTE FIFO IN EACH DIRECTION
- COMPLETE DATA LINK LAYER IMPLEMENTATION
- COMPATIBLE WITH X.25 LAPB, ISDN LAPD, X.32, AND X.75 LINK LEVEL PROTOCOLS
- 48-PIN DIP NEARLY PIN-FOR-PIN COMPATIBLE WITH THE LANCE CHIP (MK5032)
- BUFFER MANAGEMENT INCLUDES :
  - initialization block
  - separate receive and transmit rings
  - variable descriptor ring and window size
- ON CHIP DMA CONTROL WITH PROGRAMMABLE BURST LENGTH
- SELECTABLE SINGLE OR EXTENDED CONTROL FIELD
- PROGRAMMABLE 1 BYTE OR 2 BYTES ADDRESS FIELD AND GLOBAL ADDRESS
- TRANSPARENT MODE WITH OR WITHOUT ADDRESS FILTERING ALLOWS DISABLING X.25 PROCESSING FOR CUSTOMIZED APPLICATIONS
- HANDLES ALL HDLC (ADCCP) FRAME FORMATTING :
  - zero bit insert and delete
  - FCS generation and detection
  - frame delimiters by flags
- FIVE PROGRAMMABLE TIMER/COUNTERS : T1, T3, TP, N1, N2
- HANDLES ALL ERROR RECOVERY, SEQUENCING, AND S AND U FRAME CONTROL
- SELECTABLE FCS OF 16 OR 32 BITS
- DATA LINK SERVICES :
  - compatible with ISO data link services
  - compatible with LAPD data link services
- TESTING FACILITIES :
  - built-in self test facility
  - 4 loopback modes
- ALL INPUTS AND OUTPUTS ARE TTL COMPATIBLE
- PROGRAMMABLE FOR FULL OR HALF DUPLEX OPERATION
- PROGRAMMABLE MINIMUM FRAME SPACING ON TRANSMIT (flags between frames)



CDIP-48



PLCC52

(Order Codes at the end of the datasheet)

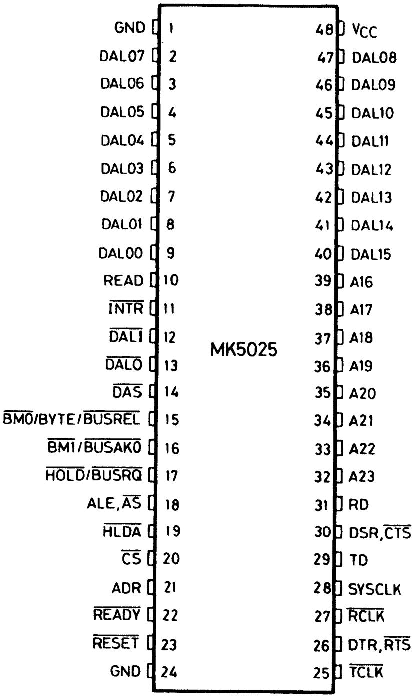
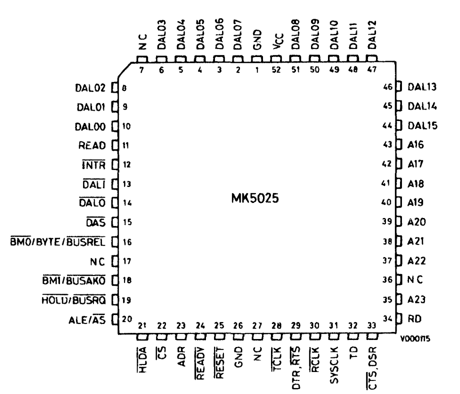


Figure 1 : MK5025 Dual in Line Pin Configuration.

GENERAL DESCRIPTION

The SGS-THOMSON X.25 Link Level Controller (MK5025) is a VLSI semiconductor device which provides a complete link level data communication control conforming to the 1984 CCITT version of X.25. This includes frame formatting, transparency (so-called "bit-stuffing"), error recovery by retransmission, sequence number control, U (unnumbered) frame control, and S (supervisory) frame control. The MK5025 also supports X.32 (XID), X.75, and ISDN LAPD. The MK5025 may be used with any of several popular 16 and 8 bit microprocessors, such as 68000, 6800, Z8000, Z80, LSI-11, 8086, 8088, 8080, etc.

Figure 2 : MK5025 Chip Carrier Pin Configuration.



PIN DESCRIPTIONS

DAL <07 : 00> (Input/Output ; 3-State). The time multiplexed Data/Address bus. During the address portion of the memory transfer, DAL <07 : 00> contains the lower 8 bits of the memory address. During the data portion of the memory address. During the data portion of a memory transfer, DAL <07 : 00> contains the read or write data, depending on the type of transfer.

READ (Input/Output ; 3-State). READ indicates the type of operation that the bus controller is performing during a bus transaction. READ is driven by the MK5025 only while it is the BUS MASTER. READ is valid during the entire bus transaction and is tristated by the MK5025 at all other times.

- MK5025 as a Bus Slave
  READ = HIGH - Data is placed on the DAL lines by the chip.
  READ = LOW - Data is taken off the DAL lines by the chip.
- MK5025 as a Bus Master
  READ = HIGH - Data is taken off the DAL lines by the chip.
  READ = LOW - Data is placed on the DAL lines by the chip.

INTR (Output ; Open Drain). INTERRUPT is an attention interrupt line that indicates that one or more of the following CSR0 status flags is set : MISS, MERR, ROR, TUR, RINT, TINT OR PINT. INTERRUPT is enabled by CSR0<09>, INEA=1.

DALI (Output ; 3-State). DAL IN is an external bus transceiver control line. DALI is driven by the MK5025 only while it is the BUS MASTER. DALI is asserted by MK5025 when it reads from the DAL lines during the data portion of a READ transfer. DALI is not asserted during a WRITE transfer.

DALO (Output ; 3-State). DAL OUT is an external bus transceiver control line. DALO is driven by the MK5025 only while it is the BUS MASTER. DALO is asserted by MK5025 when it driven the DAL lines during the data portion of a READ transfer or for the duration of a WRITE transfer.

DAS (Input/Output ; 3-State). DATA STROBE defines the data portion of a bus transaction. By definition data is stable and valid at the low to high transition of DAS. This signal is driven by the MK5025 while it is the BUS MASTER. During Bus Slave operations, this pin is used as an input. At all other times the signal is tristated.

BM0, BYTE, BUSREL (Input/Output ; 3-State). I/O pins 15 (16) and 16 (18) are programmable through CSR4. If bit 06 of CSR4 is set to a one, pin 15 (16) becomes input BUSREL and is used by the host to signal the MK5025 to terminate a DMA burst after the current bus transfer has completed. If bit 06 is clear the pin 15 (16) is the output BM0, and behaves as described below for pin 16 (18).

BM1, BUSAKO (Output ; 3-State). Pin 15 (16) and 16 (18) are programmable through bit 00 of CSR4 (BCON).

If CSR4 <00> BCON = 0,
I/O Pin 15 (16) = BM0 (Output ; 3-State)
I/O Pin 16 (18) = BM1 (Output ; 3-State)

BYTE MASK <1 : 0> Indicates the byte(s) on the DAL to be read or written during this bus transition.

PIN DESCRIPTIONS (continued)

MK5025 drives these lines only as a Bus Master. MK5025 ignores the BM lines when it is a Bus Slave.

Byte selection is done as outlined in the following table :

BM1	BM0	
LOW	LOW	Entire Word
LOW	HIGH	Upper Byte (DAL <15 : 08>)
HIGH	LOW	Lower Byte (DAL <07 : 00>)
HIGH	HIGH	None

If CSR4 <00> BCON = 1,
I/O Pin 15 (16) = BYTE (Output ; 3-State)
I/O Pin 16 (18) = BUSAKO (Output)

Byte selection is done using the BYTE line and DAL <00> latched during the address portion of the bus transaction. MK5025 drives BYTE only as a Bus Master and ignores it when a Bus Slave. Byte selection is done as outlined in the following table :

BYTE	DAL <00>	
LOW	LOW	Entire Word
LOW	HIGH	Illegal Condition
HIGH	LOW	Lower Byte
HIGH	HIGH	Upper Byte

BUSAKO is a bus request daisy chain output. If MK5025 is not requesting the bus and it receives HLDA, BUSAKO will be driven low.If MK5025 is requesting the bus when it receives HLDA, BUSAKO will remain high.

HOLD, BUSRQ (Input/Output ; Open Drain). Pin 17 (19) is configured through bit 0 of CSR4.

If CSR4 <00> BCON = 0
I/O Pin 17 (19) = HOLD

HOLD request is asserted by MK5025 when it requires a DMA cycle regardless of the previous state of the HOLD pin. HOLD is held low for the entire ensuing bus transaction.

If CSR4 <00> BCON = 1
I/O Pin 17 (19) = BUSRQ

BUSRQ is asserted by the MK5025 when it requires a DMA cycle if the prior state of the BUSRQ pin was high. BUSRQ is held low for the entire ensuring bus transaction.

ALE, AS. *Address Latch Enable, Address Strobe* (Output ; 3-State). The active level of Address Strobe is programmable through CSR4. The address portion of a bus transfer occurs while this signal is at its asserted level. This signal is driven by the MK5025 while it is the BUS MASTER. At all other times, the signal is tristated.

If CSR4 <00> ACON = 0
I/O PIN 18 (20) = ALE

Address Latch Enable is used to demultiplex the DAL lines and define the address portion of the transfer. As ALE, the signal transitions from high to low during the address portion of the transfer and remains low during the data portion.

If CSR4 <00> ACON = 1,
I/O Pin 18 (20) = AS

As AS, the signal pulses low during the address portion of the bus transfer. The low to high transition of AS can be used by a slave device to strobe the address into a register.

HLDA. *Hold Acknowledge* (Input). Hold Acknowledge is the response to HOLD. When HLDA is low in response to MK5025's assertion of HOLD, the MK5025 is the Bus Master. HLDA should be disasserted ONLY after HOLD has been released by MK5025.

CS. *Chip Select* (Input). Chip Select indicates, when low, that the MK5025 is the slave device for the data transfer. CS must be valid throughout the entire transaction.

ADR. *Address* (Input). Address selects the Register Address Port or the Register Data Port. It must be valid throughout the data portion of the transfer and is only used by the chip when CS is low.

ADR	PORT
LOW	Register Data Port
HIGH	Register Address Port

READY (Input/Output ; Open Drain). When the MK5025 is a Bus Master, READY is an asynchronous acknowledgement from the bus memory that memory will accept data in a WRITE cycle or that memory has put data on the DAL lines in a READ cycle.

As a Bus Slave, the MK5025 asserts READY when it has put data on the DAL lines during a READ cycle or is about to take data from the DAL lines during a WRITE cycle. READY is a response to DAS and it will be negated after DAS is negated.

PIN DESCRIPTIONS (continued)

$\overline{\text{RESET}}$ (Input). $\overline{\text{RESET}}$ is the Bus signal that will cause MK5025 to cease operation, clear its internal logic and enter an idle state with the STOP bit of CSR0 set.

$\overline{\text{TCLK}}$. *Transmit Clock* (Input). A 1X clock input for transmitter timing. TD changes on the falling edge of $\overline{\text{TCLK}}$. The frequency of $\overline{\text{TCLK}}$ may be up to 7Mbps.

DTR, $\overline{\text{RTS}}$. Data Terminal Ready, Request to Send (Input/Output). Modem Control Pin. Pin 26 (29) is configurable through CSR5. This pin can be programmed to behave as output $\overline{\text{RTS}}$ or as programmable I/O in DTR. If configured as $\overline{\text{RTS}}$, the MK5025 will assert this pin if it has data to send and throughout transmission of a frame.

$\overline{\text{RCLK}}$. *Receive Clock* (Input). A 1X clock input for receiver timing. RD is sampled on the rising edge of $\overline{\text{RCLK}}$. The frequency of $\overline{\text{RCLK}}$ may be up to 7MHz.

SYSCLK. *System Clock* (Input). Used for internal timing of the MK5025. SYSCLK should be a square wave, and be greater than 500KHz and less than 10MHz.

TD. *Transmit Data* (Output). Transmit serial data output.

DSR, $\overline{\text{CTS}}$. *Data Set Ready, Clear to Send* (Input/Output). Modem Control Pin. Pin 30 (33) is configurable through CSR5. This pin can be programmed to behave as input $\overline{\text{CTS}}$ or as programmable I/O pin DSR. If configured as $\overline{\text{CTS}}$, the MK5025 will transmit all 1's while $\overline{\text{CTS}}$ is high.

RD. *Receive Data* (Input). Received serial data input.

A <23 : 16> (Output ; 3-State). Address bits <23 : 16> used in conjunction with DAL <15 : 00> to produce a 24-bit address. MK5025 drives these lines only as a Bus Master.

DAL <15 : 08> (Input/Output ; 3-State). The time multiplexed Data/Address bus. For 16-bit operations, DAL <15 : 08> behaves similar to DAL <07 : 00> above for the high byte of data or the middle byte of the 24-bit address. For 8-bit operations,

DAL <15 : 08> behaves similar to A <23 : 16> for the middle byte of the 24-bit address only.

V_{SS}. Digital circuit ground pins.

V_{CC}. Main power supply pin (5V ± 5%).

OPERATIONAL DESCRIPTION

The SGS-THOMSON X.25 Link Controller (MK5025) device is a VLSI product intended for data communication applications requiring X.25 link level control (LAPB). The MK5025 will perform all frame formating, such as frame delimiting with flags, FCS generation and detection, as well as zero-bit insertion and delection for transparency. The MK5025 also includes a buffer management mechanism that allows the user to transmit and/or receive multiple packets. Contained in the buffer management is an on-chip dual channel DMA : one channel for receive and one channel for transmit. The MK5025 handles all supervisory (S) and unnumbered (U) frames as shown in table 2 and table 3.

The MK5025 is intended to be used with any popular 16 or 8 bit microprocessor. A possible system configuration for the MK5025 is shown in Figure 3.

The MK5025 will move multiple blocks of receive and transmit data directly into and out of memory through the Host's bus. An I/O acceleration processor can be used to off-load Network Level software from the Host. The I/O acceleration processor in Figure 3 is recommended, but not required.

All signal pins on the MK5025 independent of the physical interface. As shown in Figure 3, line drivers and receivers are used for electrical connection to the physical layer.

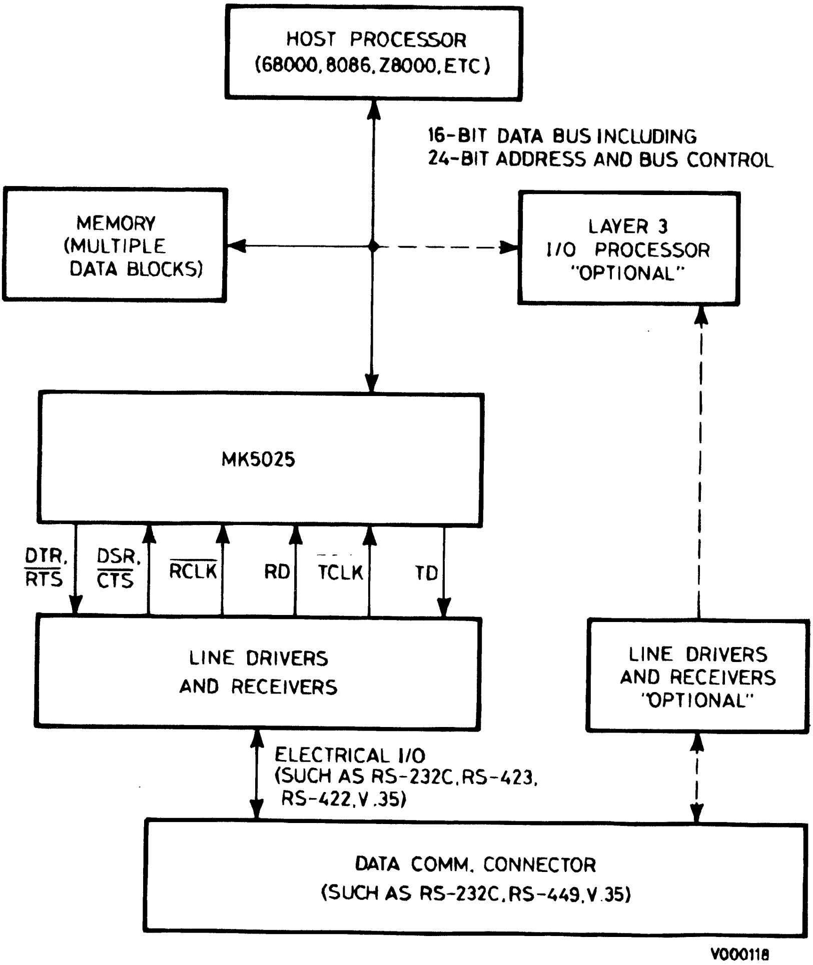
The MK5025 is mode up of functional blocks shown in Figure 4 and the MK5025 is made up of functional blocks shown in Figure 4 and described in the following paragraphs.

Microcontroller.

The microcontroller is the brain of the MK5025. It controls all of the other blocks and contains most of the protocol processing. All frame content processing as well as S and U frame processing and generation is performed by the microcontroller. All primitive processing and generation is also done here. The microcode ROM contains the control program for the microcontroller.

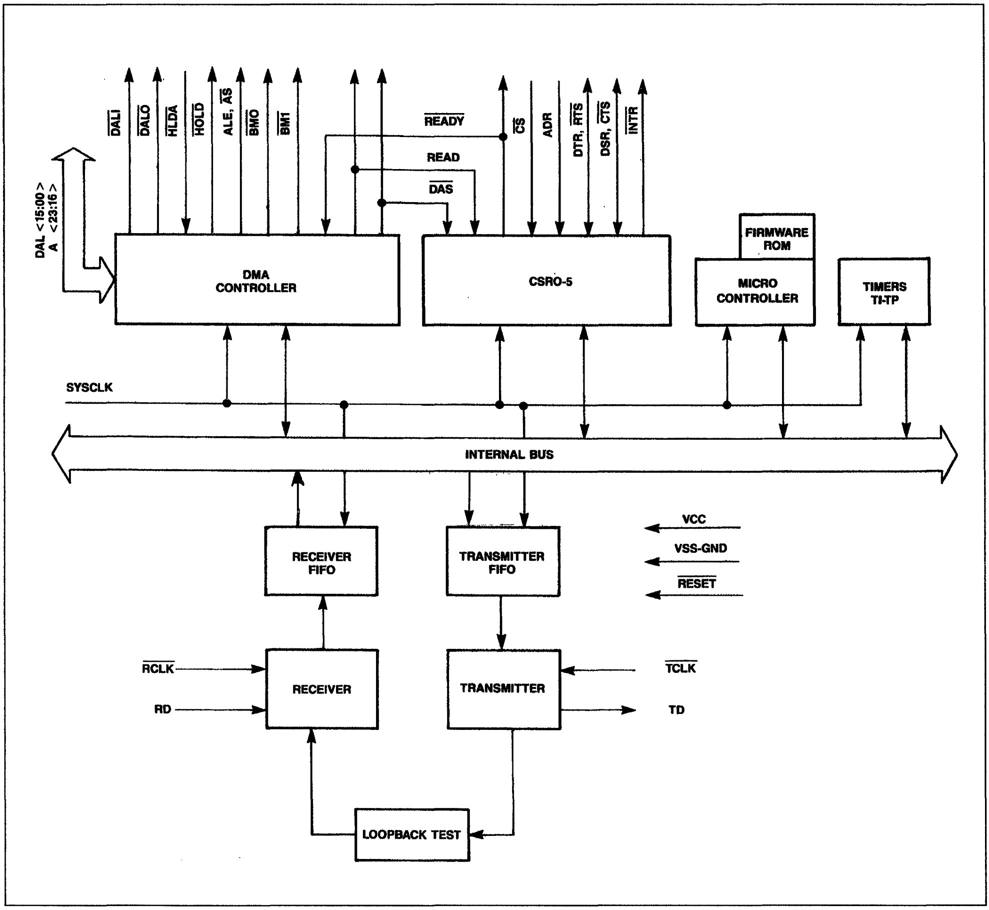
OPERATIONAL DESCRIPTION (continued)

Figure 3 : Possible System Configuration for the MK5025.



OPERATIONAL DESCRIPTION (continued)

Figure 4 : Simplified Block Diagram.



The MK5025 can interface with the host bus in two ways : either as bus master or as a bus peripheral. The MK5025 contains a dual channel DMA on chip to handle data transfers to and from the host memory. All access to the initialization block and descriptor rings is handled in this way. The address but is 24 bits wide and does not use any segmentation or paging methods. Data transfers can optionally be 8 and 16 bit operations, this allows easy interfacing with both 8 and 16 bit processors. DMA transfers can be up to 1, 8 or an unlimited number of words per-transfer under program control. In bus slave mode the MK5025 allows access to its 6 control/status registers which are used to monitor and control the chip. These registers are used to control link procedures, configure interface options, control and monitor interrupt status, and more. Bus slave mode also allows both 8 and 16-bit accesses.

DMA Controller.

The MK5025 has an on-chip DMA Controller circuit. This allows it to access memory without requiring host software intervention. Whenever the MK5025 requires access to the host memory it will negotiate

OPERATIONAL DESCRIPTION (continued)

for mastership of the bus. Upon gaining control of the bus the MK5025 will begin transferring data to or from memory. The MK5025 will perform memory transfers until either it has nothing more to transfer, it has reached its DMA burst limit (user programmable), or the BUSREL pin is driven low. In any case it will complete all bus transfers before releasing bus mastership back to the host. If, during a memory transfer, the memory does not respond within 256 SCLK cycles, the MK5025 will release ownership of the bus immediately and the MERR bit will be set in CSRO. The DMA burst limit can be programmed by the user through CSR4. In 16 bit mode th limit can be set to 1 word, 8 words, or unlimited word transfers. In 8 bit mode, it can be set to 2 bytes, 16 bytes, or unlimited byte transfers. For high speed data lines (i.e. > 1Mbps) a burst limit of 8 words, 16 bytes or unlimited is suggested to allow maximum throughput.

The byte ordering of the DMA transfers can be programmed to account for differences in processor architectures or host programming languages. Byte ordering can be programmed seperately for data and control information. Data information is defined as all contents of data buffers ; control information is defined as anything else in the shared memory space (i.e. initialization block, descriptors, etc). For more information see "Control and Status Register 4" description.

Serial Interface.

The MK5025 provides two seperate serial channels ; one for received data and one for transmitted data. These serial channels are completely seperate and may be run at different clock frequencies up to 7MHz. The receiver is responsible for recognizing frame boundaries, removal of inserted zeroes (for transparency), and checking the incoming FCS. Frames with incorrect FCS values are discarded. The receiver also parallelizes the incoming data which is placed into the receive data buffers within the receive descriptor ring. The receiver also recognizes link idle and frame abort sequences. The transmitter is responsible for framing and serializing the data frames placed in the transmit descriptor ring. The transmitter calculates the FCS of the outgoing data and appends it to the data. The transmitter generates flag sequences for interframe fill, at least two flags are transmitted between adjacent frames. The FCS calculations for both directions of serial data optionally follow either the 16-bit CRC-CCITT or the 32-bit CRC-32 algorithms. FCS generation and checking can also be optionally disabled if defined.

Receiver. Serial receive data comes into the Receiver (figure 4). The Receiver is responsible for :

1. Leading and trailing flag detection.
2. Deletion of zeroes inserted for transparency.
3. Detection of idle and abort sequences.
4. Detection of good and bad FCS (Frame Check Sequence).
5. Monitoring Receiver FIFO status.
6. Detection of Receiver-Over-Run.
7. Odd byte detection. If frames are received that have an odd number of bytes in the information field, the last byte of the frame is said to be an odd byte.
8. Detection of non-octet aligned frames, such frames are treated as invalid frames.

Transmitter. The Transmitter is responsible for :

1. Serialization of outgoing data.
2. Generating and appending the FCS.
3. Generation of interframe time-fill as either flags or idle.
4. Zero bit insertion for transparency.
5. Transmitter-Under-Run detection.
6. Transmission of odd byte.
7. RTS/CTS Control.

Frame Check Sequence. The FCS on the transmitter or receiver may be either 16 bit or 32 bit, and is user selectable. For full duplex operation, both the receiver and transmitter have individual FCS computation circuits. The characteristics of the FCS are :

- Transmitted Polarity : inverted
- Transmitted Order : High Order Bit First
- Pre-set Value : All 1's
- Polynominal 16 bit : X16 + X12 + X5 + 1
- Remainder 16 bit (if received correctly) :
  high order bit → 0001 1101 0000 1111
- Polynominal 32 bit :
  X32 + X26 + X23 + X22 + X16 + X12 + X11 + X10 + X8 + X7 + X5 + X4 + X2 + X + 1
- Remainder 32 bit (if received correctly) :
  high order bit → 1100 0111 0000 0100
  1101 1101 0111 1011

OPERATIONAL DESCRIPTION (continued)

Receive FIFO. The Receive FIFO buffers the data received by the receiver. This performs two major functions. First, it resynchronizes the data from the receive clock to the system clock. Second, it allows the microcontroller time to finish whatever it may be doing before it has to process the received data.

The receive FIFO holds the data from the receiver without interrupting the microcontroller until it contains enough data to reach the watermark level. This watermark level can be programmed in CSR4 to occur when the FIFO contains at least 2 bytes ; 18 or more bytes ; 34 or more bytes ; or 50 or more bytes. This programmability, along with the programmable burst length of the DMA controller, enables the user to define how often and for how long the MK5025 must use the host bus. For more information, see Control and Status Register 4 description.

For example, if the watermark level is set at 34 bytes and the burst length is limited to 8 word transfers at a time, the MK5025 will request control of the host bus as soon as 34 bytes are received and again after every 16 subsequent bytes.

Transmit FIFO. The Transmit FIFO buffers the data to be transmitted by the MK5025. This also performs two major functions. First, it resynchronizes the data from the system clock to the transmit clock. Second, it allows the microcontroller and DMA controller to read data from the host's memory buffers in bursts ; making both the MK5025 and the host bus more efficient.

The transmit FIFO has a watermark scheme similar to the one described for the receive FIFO above. The transmit FIFO will not interrupt the microcontroller for service until it empties enough to reach the watermark level. The watermark can be programmed in CSR4 as : any space available, 18 bytes of space available, 34 bytes of space available, or 50 bytes of space available.

Bus Slave Circuitry

The MK5025 contains a bank of internal control/status registers (CSR0-5) which can be accessed by the host as a peripheral. The host can read or write to these registers like any other bus slave. The contents of these registers are listed in the Control and Status Registers section and the bus signal timing is described in figures 9 and 10.

Buffer Management

The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings. There are separate rings to describe the transmit and receive operations. Up to 128 buffers may be queued-up on a descriptor ring awaiting execution by the MK5025. The descriptor ring has a segment assigned to each buffer. Each segment holds a pointer for the starting address of the buffer, and holds a value for the length of the buffer in words (16 bits).

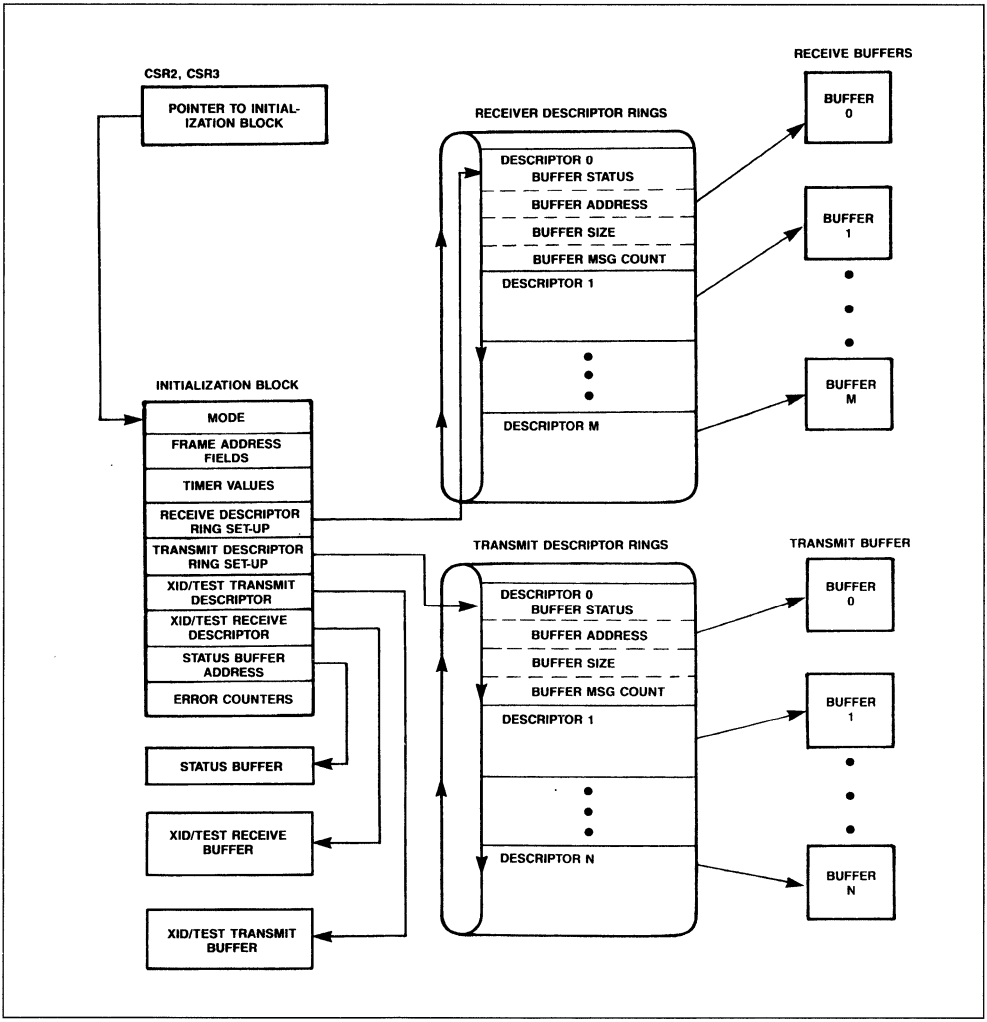
Each segment also contains two control bits called OWNA and OWNB, which denote whether the MK5025, the HOST, or the I/O ACCELERATION Processor (if present) "owns" the buffer. For transmit, when the MK5025 owns the buffer, the MK5025 is allowed and commanded to transmit the .buffer. When the MK5025 does not own the buffer, it will not transmit that buffer. For receive, when the MK5025 owns a buffer, it may place received data into that buffer. Conversely, when the Mk5025 does not own a receive buffer, it will not place received data in that buffer.

The MK5025 buffer management mechanism will handle frames which are longer than the length of an individual buffer. This is done by a chaining method which utilizes multiple buffers. The MK5025 tests the next segment in the descriptor ring in a "look ahead" manner. If the packet is too long for one buffer, the next buffer will be used after filling the first buffer ; that is, "chained". The MK5025 will then "look ahead" to the next buffer, and chain that buffer if necessary, and so on.

The operational parameters for the buffer management are defined by the user in the initialization block. The parameters defined include the basic mode of operation, the number of entries for the transmitter and receiver descriptor rings, frame Address field, and etc.

OPERATIONAL DESCRIPTION (continued)

Figure 5 : Buffer Management Organization.



The Initialization Block. Chip initialization information is located in a block of memory called the Initialization Block. The Initialization Block consists of 28 contiguous words of memory starting on a word boundary. This memory is assembled by the HOST or I/O acceleration processor, and is accessed by MK5025 during initialization. The Initialization Block is comprised of :

A. Mode of Operation.

B. Frame Address Values.

C. Timer Preset Values.

D. Location and size of Receive and Transmit Descriptor Rings.

E. Location and size of XID/TEST Buffers.

F. Location of status buffer.

G. Error Counters.

OPERATIONAL DESCRIPTION (continued)

The Circular Queue. The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings. There are separate rings to describe the transmit and receive operations. Up to 128 buffers may be queued-up on a descriptor ring awaiting execution by the MK5025. The descriptor ring has a segment assigned to each buffer. Each segment holds a pointer for the starting address of the buffer, and holds a value for the length of the buffer in bytes.

Each segment also contains two control bits called OWNA and OWNB, which denote whether the MK5025, the HOST, or the I/O ACCELERATION PROCESSOR (if present) "owns" the buffer. For transmit, when the MK5025 owns the buffer, the MK5025 is allowed and commanded to transmit the buffer. When the MK5025 does not own the buffer, it will not transmit that buffer. For receive, when the MK5025 owns a buffer, it may place received data into that buffer. Conversely, when the MK5025 does not own a receive buffer, it will not place received data in that buffer.

The MK5025 buffer management mechanism will handle frames which are longer than the length of an individual buffer. This is done by a chaining method which utilizes multiple buffers. The MK5025 tests the next segment in the descriptor ring in a "look ahead" manner. If the frames are too long for one buffer, the next buffer will be used after filling (or transmitting) the first buffer ; that is, "chained". The MK5025 will then "look ahead" to the next buffer, and chain that buffer if necessary, and so on.

The operational parameters for the buffer management are defined by the user in the initialization block. The parameters defined include the basic mode of operation, the number of entries for the transmitter and receiver descriptor rings, frame Address field, and etc. The starting address for the Initialization, IADR, is defined in the CSR2 and CSR3 registers inside the MK5025.

Frame Format. The frame format used by the MK5025 is shown below. Each frame consists of a programmable number of leading flag patterns (01111110), an address field, a control field, an information field (not in all frames), an FCS of either 16 or 32 bits, and a trailing flag pattern. The number of leading flags is programmable through the Mode Register in the Initialization Block. Received frames may have only one flag between adjacent frames.

F	•••	F	A	C	I	FCS	F
8-bits		8	8/16	8/16	8*n	16/32	8

Transmitted First

The Command/Response Repertoire. The command/response repertoire of the MK5025 is shown in tables 2 and 3. This set conforms to the ISDN LAPD, which is a super-set of X.25 Link Level. The MK5025 will process the S and U frames shown in table 1 and will handle the A and C fields for all I and UI frames.

Table 1 : Frame Types Symbols Definitions.

Name	Definition
I	Information frame
UI	Unnumbered Information
RR	Receiver Ready
DISC	Disconnect
RNR	Receiver Not Ready
UA	Unnumbered Acknoledge
REJ	Reject
FRMR	Frame Reject
SABM	Set Asynchronous Balance Mode
DM	Disconnect Mode
XID	Exchange Identification
TEST	Link Test frame

OPERATIONAL DESCRIPTION (continued)

Table 2 : Command/Response Repertoire - Module 8 Operation.

Format	Command	Resp	Encoding							
			1	2	3	4	5	6	7	8
Information Transfer	I		0	←	N(S)	→	P	←	N(R)	→
Supervisory	RR	RR	1	0	0	0	P/F	←	N(R)	→
	RNR	RNR	1	0	1	0	P/F	←	N(R)	→
	REJ	REJ	1	0	0	1	P/F	←	N(R)	→
Unnumbered	SABM		1	1	1	1	PF	1	0	0
		DM	1	1	1	1	F	0	0	0
	XID (1)	XID (1)	1	1	1	1	P/F	1	0	1
	UI (1)	UI (1)	1	1	0	0	P/F	0	0	0
	DISC		1	1	0	0	P	0	1	0
		UA	1	1	0	0	F	1	1	0
		FRMR	1	1	1	0	F	0	0	1
	TEST (2)	TEST (2)	1	1	0	0	P/F	1	1	1

Table 3 : Command/Response Repertoire - Modulo 128 Operation.

Format	Command	Resp	Encoding									
			1	2	3	4	5	6	7	8	9	10-1
Information Transfer	I		0	N(S)							P	N(R)
Supervisory	RR	RR	1	0	0	0	P/F	←	N(R)	→	P/F	N(R)
	RNR	RNR	1	0	1	0	P/F	←	N(R)	→	P/F	N(R)
	REJ	REJ	1	0	0	1	P/F	←	N(R)	→	P/F	N(R)
Unnumbered	SABME		1	1	1	1	PF	1	1	0		
		DM	1	1	1	1	F	0	0	0		
	XID (1)	XID (1)	1	1	1	1	P/F	1	0	1		
	UI (1)	UI (1)	1	1	0	0	P/F	0	0	0		
	DISC		1	1	0	0	P	0	1	0		
		UA	1	1	0	0	F	1	1	0		
		FRMR	1	1	1	0	F	0	0	1		
	TEST (2)	TEST (2)	1	1	0	0	P/F	1	1	1		

Notes : 1. XID and UI frames can be individually enabled for compatibility with X.32 and LAPD respectively.
2. TEST frames are enabled with XID frames.
3. N(S) = Transmitter Send sequence number.
4. N(R) = Transmitter Receive sequence number.
5. P/F = Poll bit when issued as a command, Final bit when issued as a response.

Protocol. The MK5025 contains a full implementation of the 1984 CCITT X.25 data link layer. It allows both basic and extended control fields, variable window sizes, and user-defined counter and timer values. Extended addressing and UI frames are optionally available for use in ISDN LAPD applications. XID and TEST frames are available for use in X.32. The interface between the MK5025 and the host (layer 3) conforms to both the ISO data link services standard and the ISDN LAPD data link services standard.

PROGRAMMING SPECIFICATIONS

This section defines the Control and Status Registers and the memory data structures required to program the MK5025.

Control and Status Registers.

There are six Control and Status Registers (CSR's) resident within MK5025. The CSR's are accessed through two bus addressable ports, an address port (RAP), and a data port (RDP). Thus the MK5025 needs only two address locations in the system memory or I/O map.

Accessing the Control and Status Registers. The CSR's are read (or written) in a two-step operation. The address of the CSR is written into the address port (RAP) during a bus slave transaction. During a subsequent bus slave transaction, the data being read from (or written into) the data port (RDP) is read from (or written into) the CSR selected in the RAP. Once written, the address in RAP remains unchanged until rewritten. A control I/O pin (ADR) is provided to distinguish the address port from the data port.

ADR	PORT
HIGH	Register Address Port (RAP)
LOW	Register Data Port (RDP)

Register Address Port (RAP)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	BM8	0	0	0	CSR <2:0>			HBYTE

Bit <15 : 08>. They are reserved and must be written as zeroes.

Bit 07, BM8. When set, places chip into 8-bit mode. CSR's, Init Block and data transfers are all 8-bit transfers, this provides compatibility with 8-bit microprocessors. When clear, all transfers are 16-bit transfers. This bit must be set to the same value each time it is written, changing this bit during normal operation will achieve unexpected results. BM8 is Read/Write and cleared on Bus Reset.

Bit <06 : 04>. They are reserved and must be written as zeroes.

Bit <03 : 01>, CSR <2 : 0>. CSR address select bits is Read/Write. Selects the CSR to be accessed through the RDP RAP is cleared by Bus Reset.

CSR<2 : 0>	CSR Selected
0	CSR0
1	CSR1
2	CSR2
3	CSR3
4	CSR4
5	CSR5

Bit 00, HBYTE. Determines which byte is addressed for 8-bit operation. If set, the high byte of the register referred to by CSR <2 : 0> is addressed, otherwise the low byte is addressed. This bit is only meaningful for 8-bit operation and must be written as zero if BM8 = 0. HBYTE is Read/Write and cleared on Bus Reset.

Register Data Port (RDP)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
CSR DATA															

Bit <15 : 00>, CSR DATA. Writing data to the RDP loads data into the CSR selected by RAP. Reading the data from RDP reads the data from the CSR selected in RAP.

Control and Status Register 0 (CSR0)

RAP <3 : 1> = 0

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TDMD	STOP	DTX	DRX	TXON	RXON	INEA	INTR	MERR	MISS	ROR	TUR	PINT	TINT	RINT	0

PROGRAMMING SPECIFICATIONS (continued)

Bit 15, TDMD. TRANSMIT DEMAND, when set, causes MK5025 to access the Transmit Descriptor Ring without waiting for the transmit polltime interval to elapse. TDMD need not be set to transmit a frame, it merely hastens MK5025's response to a Transmit Descriptor Ring entry insertion by the host. TDMD is WRITE WITH ONE ONLY and cleared by the MK5025 after it is used. It may read as a "1" for a short time after it is written because the MK5025 may have been busy when TDMD was set. It is also cleared by Bus RESET. Writing a "0" in this bit has no effect.

Bit 14, STOP. STOP, when set, indicates that MK5025 is operating in the STOPPED phase of operation. All external activity is disabled and internal logic is reset. MK5025 remains inactive except for primitive processing until a START primitive is issued. STOP IS READ ONLY and set by Bus RESET or a STOP primitive. Writing to this bit has no effect.

Bit 13, DTX. Transmitter Ring Disable prevents the MK5025 from further access to the Transmitter Descriptor Ring. No transmissions are attempted after finishing transmission of any frame in transmission at the time of DTX being set. DTX is READ/WRITE. TXON acknowledges changes to DTX, see below.

Bit 12, DRX. Receiver Ring Disable prevents the MK5025 from further access to the Receiver Descriptor Ring. No received frames are accepted after finishing reception of any frame in reception at the time of DRX being set. If DRX is set while a data link is established the MK5025 will go into the local busy condition and will send a RNR response frame to the remote station. Upon clearing DRX the MK5025 will send a RR response frame. DRX is READ/WRITE. RXON acknowledges changes to DRX, see below.

Bit 11, TXON. TRANSMITTER ON indicates that the transmitter ring access is enabled. TXON is set as the START primitive is issued if the DTX bit is "0" or afterward as DTX is cleared. TXON is cleared upon recognition of DTX being set, by issuing a STOP primitive in CSR1, or by a Bus RESET. If TXON is clear, the host may modify the Transmit Descriptor Ring entries regardless of the state of the OWNA bits. TXON is READ ONLY ; writing this bit has no effect.

Bit 10, RXON. RECEIVER ON indicates that the receiver ring access is enabled. RXON is set as the START primitive is issued if the DRX bit is "0" or afterward as DRX is cleared. RXON is cleared upon recognition of DRX being set, by sending a STOP primitive in CSR1, or by a Bus RESET. If RXON is clear, the host may modify the Receive Descriptor Ring entries regardless of the state of the OWNA bits. RXON is READ ONLY ; writing this bit has no effect.

Bit 09, INEA. INTERRUPT ENABLE allows the INTR I/O pin to be driven low when the Interrupt Flag is set. If INEA = 1 the INTR I/O pin will be low if CSR0 <08> INTR is set. If INEA = 0 the INTR I/O pin will be high, regardless of the state of the Interrupt Flag. INEA is READ/WRITE, set by writing a "1" into this bit and is cleared by writing a "0" into this bit or by Bus RESET or by issuing a STOP primitive.

Bit 08, INTR. INTERRUPT FLAG indicates that one or more of the following interrupt causing conditions has occured ; MISS, MERR, RINT, TINT, PINT, TUR or ROR. If INEA = 1 and INTR = 1 the INTR I/O pin will be low. INTR is READ ONLY, writing this bit has no effect. INTR is cleared as the specific interrupting condition bits are cleared. INTR is also cleared by Bus RESET or by issuing a STOP primitive.

PROGRAMMING SPECIFICATIONS (continued)

Bit 07, MERR. MEMORY ERROR sets when MK5025 is the Bus Master and has not received READY within 256 SYSCLKs (25.6 usec @ 10MHz) after asserting the address on the DAL lines. When a Memory Error is detected, the receiver and transmitter are turned off and an interrupt is generated if INEA = 1. MERR is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by Bus RESET or by issuing a STOP primitive.

Bit 06, MISS. MISSED PACKET is set when the receiver loses a packet because it does not own a receive buffer, indicating loss of a frame. When MISS is set, an interrupt will be generated if INEA = 1. MISS is READ/CLEAR ONLY and is set by the MK5025 and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by Bus RESET or by issuing a STOP primitive.

Bit 05, ROR. RECEIVER OVERRUN indicates that the Receiver FIFO was full when the receiver was ready to input data to the Receiver FIFO. The frame being received is lost but is recoverable according to the Link Level protocol. When ROR is set, an interrupt is generated if INEA = 1. ROR is READ/CLEAR ONLY and is set by MK5025 and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by Bus RESET or by issuing a STOP primitive.

Bit 04, TUR. TRANSMITTER UNDERRUN indicates that the MK5025 has aborted a frame since data was late from memory. This condition is reached when the transmitter and transmitter FIFO both become empty while transmitting a frame. When TUR is set, an interrupt is generated if INEA = 1. TUR is READ/CLEAR ONLY and is set by the MK5025 and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by Bus RESET or by issuing a STOP primitive.

Bit 03, PINT. PRIMITIVE INTERRUPT is set after the chip updates the primitive register to issue a provider primitive or to indicate a User Primitive Error Condition. See CSR1<15> UERR. When PINT is set, an interrupt is generated if INEA = 1. PINT is READ/CLEAR ONLY and is set by MK5025 and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by Bus RESET or by issuing a STOP primitive.

Bit 02, TINT. TRANSMITTER INTERRUPT is set after the chip updates an entry in the Transmit descriptor Ring. This occurs when a transmitted frame has been acknowledged by the remote station. When TINT is set, an interrupt is generated if INEA = 1. TINT is READ/CLEAR ONLY and is set by MK5025 and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is also cleared by Bus RESET or by issuing a STOP primitive.

Bit 01, RINT. RECEIVER INTERRUPT is set after MK5025 updates an entry in the Receive Descriptor Ring. This occurs when the MK5025 has received a correct frame from the remote station. When RINT is set, an interrupt is generated if INEA = 1. RINT is READ/CLEAR ONLY and is set by MK5025 and cleared by writing a "1" into the bit. Writing a "0" has no effect. It is cleared by Bus RESET or by issuing a STOP primitive.

Bit 00. Reserved. This bit is READ ONLY.

Control and Status Register 1 (CSR1)

RAP <3 : 1> = 1

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
UERR	UAV	UPARM <1:0>		UPRIM <3:0>				PLOST	PAV	PPARM	1 : 0	PPRIM <3:0>			

PROGRAMMING SPECIFICATIONS (continued)

BIT 15, UERR. USER PRIMITIVE ERROR is set by the MK5025 when a primitive issued by the user is in conflict with the current status of the link. UERR is READ/CLEAR ONLY and is set by MK5025 and cleared by writing a "1" into the bit. Writing a "0" in this bit has no effect. It is also cleared by Bus RESET.

Bit 14, UAV. USER PRIMITIVE AVAILABLE is set by the user after a primitive has been placed in UPRIM. It is cleared by the MK5025 after the primitive has been processed. This bit is also cleared by a Bus RESET.

Bit <13 : 12>, UPARM. UPARM is written by the host in conjunction with the user primitives in UPRIM. This user parameter field provides for the information to the MK5025 concerning the corresponding user primitive. For connect and reset primitives this field determines what the MK5025 will do with frames in the transmit descriptor ring that have been previously sent but not acknowledged. If UPARM = 0, these frames will be resent when the new link is established. If UPARM = 1, these frames are discarded by the MK5025 and their OWNA bits are cleared, releasing ownership back to the host. For all other primitives this field should be written with zeroes.

Bit <11 : 08>, UPRIM. USER PRIMITIVE is written by the user to control the MK5025 link procedures. The following values are valid :

0 Stop - Instructs MK5025 to go into STOPPED Mode. All link activity is terminated and the STOP bit is set. Transmitter outputs all "1"s. All DMA activity ceases.

1 Start-Instructs MK5025 to exit STOPPED Mode and enter the Disconnected Phase. Descriptor Rings are reset. Transmitter begins outputting flags. Valid only in STOPPED Mode.

2 Init Request - Instructs MK5025 to read the initialization block. Valid only in STOPPED Mode and Disconnected Phase. This should be performed prior to the start primitive after a bus reset or powerup.

3 Trans - Instructs MK5025 to enter the Transparent phase of operation. Data frames are transmitted and received out of the descriptor rings but no protocol processing is done. Address and Control Fields are not prepended to the frames, but FCS processing works normally. If the PROM bit is set in CSR2 then no address filtering is performed on received frames. Transparent Mode may be exited only with a stop primitive or by bus reset. Valid only in STOPPED Mode.

4 Status Request - Instructs MK5025 to write the current link status into the STATUS buffer. Valid only if INIT primitive has previously been issued.

5 Self-Test Request - Instructs MK5025 to perform a self test. Valid only in STOPPED Mode. See page 31/40 or whatever page it finally is on for self-test procedure.

6 Connect request - Instructs MK5025 to attempt to establish a logical link with the remote site. Valid only in Disconnected Phase.

7 Connect Response - Indicates willingness to establish a logical link with the remote site. Valid only in Disconnected Phase after receiving a Connect Indication primitive.

PROGRAMMING SPECIFICATIONS (continued)

8 Reset Request - If a logical link has been established, instructs MK5025 to attempt to reset the current logical link with the remote site (sends SABM/E). In Transparent Mode or Disconnected Phase, instructs MK5025 to start the T1 timer.

9 Reset Response - If a logical link has been established, indicates willingness to reset current logical link with remote site (valid only after receiving a Reset Indication primitive.) In Transparent Mode or Disconnected Phase, instructs MK5025 to stop the T1 timer.

10 XID Request - Requests MK5025 to send an XID command to the remote site. Data in the XID/TEST Transmit buffer is used for the Data Field. Invalid in STOPPED Mode.

11 XID Response - Requests MK5025 to send an XID response to the remote site. Data in the XID/TEST Transmit Buffer is used for the Data Field. Valid only after receiving an XID Indication primitive.

12 TEST Request - Requests MK5025 to send a TEST command to the remote site. Data in the XID/TEST Transmit Buffer is used for the Data Field. Invalid in STOPPED Mode.

13 TEST Response - Requests MK5025 to send a TEST response to the remote site. Data in the XID/TEST transmit buffer is used for the data field. Valid only after receiving a TEST indication primitive.

14 Disconnect Request - Requests MK5025 to disconnect the current logical link. Invalid in STOPPED Mode. A DM response with the F bit clear will be sent if the link is currently disconnected.

Bit 07, PLOST. PROVIDER PRIMITIVE LOST is set by MK5025 when a provider primitive cannot be issued because the PAV bit is still set from the previous provider primitive. PLOST is cleared when PAV is cleared and by a Bus RESET. Writing to this bit has no effect.

Bit 06, PAV. PROVIDER PRIMITIVE AVAILABLE is set by the MK5025 when a new provider primitive has been placed in PPRIM. PPRIM is READ/CLEAR ONLY and is set by the chip and cleared by writing a "1" to the bit or by Bus RESET.

Bit <05 : 04>, PPARM. PROVIDER PARAMETER provides additional information about the reason for the receipt of a disconnect, reset or error indication primitive. This field is undefined for other provider primitives. Parameters are as follows :

PPARM	Disconnect Indication	Disconnect Confirmation	Reset Indication	Error Indication
0	Remotely Initiated	UA or DM F = 1 Recvd	Remotely Initiated	Unsolicited DM/F = 0 Recvd
1	SABM Timeout	DISC Timeout		Timer Rec Timeout
2	FRMR Sent then DISC or DM Received T3 Timeout		FRMR Sent then SABM/E Recvd	FRMR Received
3		T3 Timeout		Unsolicited UA or F bit Received

PROGRAMMING SPECIFICATIONS (continued)

Bit <03 : 00>, PPRIM. PROVIDER PRIMITIVE is written by MK5025 to inform the user of link control conditions. Valid Provider Primitives are as follows :

2 Init Confirmation - Indicates that the initialization has completed.

4 Error Indication - Indicates an error condition has occurred during the Information Transfer phase of operation that requires instruction by the Host for recovery. See PPARM for specific error conditions. Either a Reset Request or Disconnect Request primitive should be issued in UPRIM after receiving an Error Indication primitive.

5 Remote Busy Indication - Indicates change in Remote Busy status. PPARM = 0 indicates receipt of RNR - Remote Busy. PPARM = 1 indicates Remote no longer busy - RR received. This primitive is only issued if RBSY = 1. RBSY (bit 15 of IADR + 16) is set by the Host in the Init block.

6 Connect Indication - Indicates an attempt by the remote site to establish a logical link. Appropriate user responses are Connect Response or Disconnect Request.

7 Connect Confirmation - Indicates the success of a previous Connect Request by the user. A logical link is now established.

8 Reset Indication - If a logical link has been established, indicates an attempt by the remote site to reset the current logical link. Appropriate user responses are Reset Response or Disconnect Request. In Transparent Mode, or-Disconnected Phase, indicates expiry of timer T1.

9 Reset Confirmation - Indicates the success of a previous Reset Request by the user. The current logical link has now been reset.

10 XID Indication - Indicates the receipt of an XID command. The Data Field of the XID command is located in the XID/TEST Receive Buffer.

11 XID Confirmation - Indicates the receipt of an XID response. The Data field of the XID response is located in the XID/TEST Receive Buffer.

12 TEST Indication - Indicates the receipt of a TEST command. The Data Field of the TEST command is located in the XID/TEST Receive buffer.

13 TEST Confirmation - Indicates the receipt of a TEST response. The Data field of the TEST response is located in the XID/TEST Receive Buffer.

14 Disconnect Indication - Indicates a request by the remote site to disconnect the current logical link or the refusal of a previous Connect or Reset Request. The chip is now in the Disconnected Phase.

15 Disconnect Confirmation - Indicates the completion of a previously requested link discon nection.

Control and Status Register 2 (CSR2)

RAP <3 : 1> = 2

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	FRMRD	T203E	X75E	PROM	UIE	XIDE	IADR<23:16>							

Bit <15 : 14>. Reserved, must be written as zeroes.

Bit 13, FRMRD. Setting this bit disables the sending of FRMR frames (used for LAPD applications) ; otherwise the MK5025 behaves as specified for X.25. This bit is READ/WRITE.

Bit 12, T203E. If this bit is set, the T3 timer is reconfigured to behave as specified for LAPD T203 timer ; otherwise it behaves as specified for X.25. This bit is READ/WRITE.

PROGRAMMING SPECIFICATIONS (continued)

Bit 11, X75E. X.75 mode is enabled if this bit is set to 1 ; otherwise X.25 mode is enabled. This bit is READ/WRITE.

Bit 10, PROM. Address filtering is disabled for transparent mode, if this bit is set. All uncorrupted incoming frames are placed in the Receive Descriptor Ring. This bit is READ/WRITE. Should be set only in Trans Mode.

Bit 09, UIE. UI frames are recognized only if this bit is set. If UIE = 0 all received UI frames will not be recognized. This bit is READ/WRITE.

Bit 08, XIDE. XID frames are recognized only if this bit is set. If XIDE = 0 all received XID frames will not be recognized. This bit is READ/WRITE.

Bit <07 : 00>, IADR. The high order 8 bits of the address of the first word (lowest address) in the Initialization Block. IADR must be written by the Host prior to issuing an INIT primitive.

Control and Status Register 3 (CSR3)

RAP <3 : 1> = 3

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IADR <15:00>															0

BIT <15 : 00>, IADR. The low order 16 bits of the address of the first word (lowest address) in the Initialization Block. Must be written by the Host prior to issuing an INIT primitive. The Initialization Block must be on an even byte boundary.

Control and Status Register 4 (CRS4). CRS4 allows redefinition of the bus master interface.

RAP <3 : 1> = 4

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	FWM <1:0>		BAEN	BUSR	BSWPC	BURST	1 : 0	BSWPD	ACON	BCON

Bit <15 : 10> Reserved, must be written as zeroes.

Bit <09 : 08>, FWM. These bits define the FIFO watermarks. FIFO watermarks prevent the MK5025 from performing DMA transfers to/from the data buffers until the FIFOs contain a minimum amount of data or space for data. For receive data, data will only be transferred to the data buffers after the FIFO has at least N 16-bit words or an end of frame has been reached. Conversely, for transmit data, data will only be transferred from the data buffers when the transmit FIFO has room for at least N words of data. N is defined as follows :

FWM :0	N
11	1word
10*	9 words
01	17 words
00	25 words

* Suggested Setting

Bit 07, BAEN. This bit should be written as "0" for standard operation as described in the timing diagrams in figures 7 and 8 of this manual. If this bit is set, the upper 4 address bits (A <23 : 20>) will be available at the time HOLD is asserted, and are never tristated. This facilitates use in multiple bus systems to identify which bus is being requested.

Bit 06, BUSR. If this bit is set, pin 15 becomes input BUSREL. If this bit is clear pin 15 is either BM0 or BYTE depending on bit 00. For more information see the description for pin 15 earlier in this document. BUSR is READ/WRITE and cleared on Bus Reset.

Bit 05, BSWPC. This bit determines the byte ordering of all "non-data" DMA transfers. "Non-data" DMA transfers refers to any DMA transfers that access memory other than the data buffers themselves. This includes the Initialization Block, Descriptors, and Status Buffer. It has no effect on data DMA transfers. BSWPC allows MK5025 to operate with memory organizations that have bits <07 : 00> ar even addresses with bits <15 : 08> at odd addresses or vice versa.

With BSWPC = 1 :

Address

XX0	0 ••• 7
XX1	8 ••• 15

This memory organization is used with the LSI 11 and the 8086 microprocessors.

With BYTE SWAP = 0 :

Address

XX0	8 ••• 15
XX1	0 ••• 7

This memory organization is used with the 68000 and Z8000 microprocessors. BSWP is Read/Write and cleared by BUS RESET.

Bit <04 : 03>, BURST. This field determines the maximum number of data transfers performed each time control of the host bus is obtained. BURST is READ/WRITE and cleared on Bus RESET.

BURST <1 : 0>	8 bit Mode	16 bit Mode
00	2 bytes	1 word
10*	16 bytes	8 words
01	unlimited	unlimited

* Suggested Setting

Bit 02, BSWPD. This bit determines the byte ordering of all data DMA transfers. Data transfers are those to or from a data buffer. BSWPD has no effect on non-data transfers. The effect of BSWPD on data transfers is the same as that of BSWPC on non-data transfers (see above). For most applications, including most 68000 based systems, this bit should be set.

Bit 01, ACON. ALE CONTROL defines the assertive state of Pin 18 when MK5025 is a Bus Master. ACON is READ/WRITE and cleared by Bus RESET.

ACON	Pin 18	Asserted
0	ALE	High
1	AS	Low

Bit 00, BCON. BYTE CONTROL redefines the Byte Mask and Hold I/O pins. BCON is READ/WRITE and cleared by Bus RESET.

BCON	PIN 16	PIN 15	PIN 17
0	BM 1	BM 0	HOLD
1	BUSAKO	BYTE	BUSRQ

Control and Status Register 5 (CSR5).CSR5 facilitates control and monitoring of modem controls.

RAP <3 : 1> = 5

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	0	RTSEN	DTRD	DSRD	DTR	DSR

Bit <15 : 05>. Reserved, must be written as zeroes.

Bit 4, RTSEN. RTS/CTS ENABLE is a READ/WRITE bit used to configure pins 26 and 30. If this bit is set pin 26 becomes RTS and pin 30 becomes CTS. RTS is driven low whenever the MK5025 has data to transmit and kept low during transmission. RTS will be driven high after the closing flag of a frame transmitted if either no other frames are in the FIFO or if the minimum frame spacing is higher than 2 (see Mode Register). The MK5025 will not begin transmission and TD will remain HIGH if CTS is high.

Bit 3, DTRD. DTR DIRECTION is a READ/WRITE bit used to control the direction of the DTR pin. If DTRD = 0, the DTR pin becomes an input pin and the DTR bit reflects the current value of the pin ; if DTRD = 1, the DTR pin is an output pin controlled by the DTR bit below.

Bit 2, DSRD. DSR DIRECTION is a READ/WRITE bit used to control the direction of the DSR pin. If DSRD = 0, the DSR pin becomes an input pin and the DSR bit reflects the current value of the pin ; if DSRD = 1, the DSR pin is an output pin controlled by the DSR bit below.

Bit 1, DTR. DATA TERMINAL READY is used to control or observe the DTR I/O pin depending on the value of DTRD. If DTRD = 0, this bit becomes READ ONLY and always equals the current value of the DTR pin. If DTRD = 1, this bit becomes READ/WRITE and any value written to this bit appears on the DTR pin.

Bit 0, DSR. DATA SET READY is used to control or observe the DSR I/O pin depending on the value of DSRD. If DSRD = 0, this bit becomes READ ONLY and always equals the current value of the DSR pin. If DSRD = 1, this bit becomes READ/WRITE and any value written to this bit appears on the DSR pin.

Initialization Block

MK5025 initialization includes the reading of the initialization block in memory to obtain the operating parameters.

The Initialization Block is read by MK5025 when receiving an INIT primitive. During normal initialization the INIT should be sent prior to sending a START primitive. The user may re-issue the INIT primitive after a START, but received frames may be lost if care is not taken. An INIT cannot be issued while a link is connected ; MK5025 will reject such an attempt.

Except for the Error Counters and XID/TEST Descriptor OWNA bits, the MK5025 will not write into the Initialization Block.

Figure 6 : Initialization Block.

BASE ADDRESS	MODE	IADR+00
	LOCAL STATION ADDRESS	IADR+02
	REMOTE STATION ADDRESS	IADR+04
	N1	IADR+06
	N2 + SCALER	IADR+08
	T1 TIMER	IADR+10
	T3 TIMER	IADR+12
	TP TIMER	IADR+14
	RLEN-RDRA <23:16>	IADR+16
	RDRA <15:00>	IADR+18
	TLEN-TDRA <23:16>	IADR+20
	TDRA <15:00>	IADR+22
	XID/TEST TRANSMIT DESCRIPTOR	IADR+24
	XID/TEST RECEIVE DESCRIPTOR	IADR+32
	STATUS BUFFER ADDRESS	IADR+40
HIGHER ADDR	SIX ERROR COUNTERS	IADR+44 thru IADR+55

Mode Register. The Mode Register allows alteration of the MK5025's operating parameters.

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MFS<4:0>					EXTCF	EXTAF	DACE	EXTC	EXTA	DRFCS	DTFCS	FCSS	LBACK <2:0>		

Bit <15 : 11>, MFS <4 :0>. Minimum Frame Spacing defines the minimum number of flag sequences transmitted between adjacent frames transmitted by the MK5025. This only affects frames transmitted by the MK5025 and does not restrict the spacing of frames received by the MK5025. When using RTS/CTS control this field defines the number of flags transmitted at the beginning of the frame after CTS is driven low (minus one for the trailing flag). See following table for encoding of this field.

Number of Flags	MFS <4:0>
2	0
4	2
6	4
8	9
10	18
12	5
14	11
16	22
18	12
20	25
22	19
24	7
26	15
28	31
30	30
32	28
34	24
36	17
38	3
40	6
42	13
44	27
46	23
48	14
50	29
52	26
54	21
56	10
58	20
60	8
62	16

Bit 10, EXTCF. Extended Control Force is useful only in transparent mode operation. If set along with EXTC, the receiver will assume the control field to be two octets long regardless of the first two bits of the control field. See EXTC below.

Bit 09, EXTAF. Extended Address Force is useful only in transparent mode operation. If set along with EXTA, the receiver will assume the address field to be two octets long regardless of the first bit of the address. See EXTA below.

Bit 08, DACE. Address and control field extraction are disabled when DACE is set. Address and control fields are treated as normal data. DACE must be written as "0" for normal operation in non-transparent mode.

Bit 07, EXTC. Extended Control Field is enabled when EXTC = 1. The control fields of all S and I frames become two octets in length, instead of one. The numbering for I frames becomes modulo 128, instead of modulo 8. The control field of U frames remains one octet in length.

Bit 06, EXTA. Extended address is enabled when EXTA = 1. The length of address fields is determined by the first bit of the address. If the first bit is set then the address field is 1 octet long otherwise it is 2 octets.

Bit 05, DRFCS. Disable Receiver FCS. When DRFCS = 0, the receiver will extract and check the FCS field at the end of each frame. When DRFCS = 1, the receiver continues to extract the last 16 or 32 bits of each frame, depending on FCSS, but no check is performed to determine whether the FCS is correct.

Bit 04, DTFCS. Disable Transmitter FCS. When DTFCS = 0, the transmitter will generate and append the FCS to each frame. When DTFCS = 1, the FCS logic is disabled, and no FCS is generated with transmitted frames.

Setting DTFCS = 1 is useful in loopback testing for checking the ability of the receiver to detect an incorrect FCS.

Bit 03, FCSS. FCS Select. When FCSS = 1, a 16 bit FCS is selected otherwise a 32-bit FCS is used.

Bit <02 : 00>, LBACK. Loopback Control puts the MK5025 into one of several loopback configurations.

LBACK0. Normal operation. No loopback.

LBACK4. Simple loopback. Receive data and clock are driven internally by transmit data and clock.

LBACK5. Clockless loopback. Receive data is driven internally by transmit data. Transmit and receive clocks are driven by SYSCLK divided by 8.

LBACK6. Silent loopback. Same as simple loopback with TD pin forced to all ones.

LBACK7. Silent Clockless loopback. Combination of Silent and Clockless loopbacks. Receive data is driven internally by transmit data, transmit and receive clocks are driven by SYSCLK divided by 8. TD pin is forced to all ones.

Station Addresses. The Local and Remote station addresses may be either one or two octets according to the EXTA control bit described in the MODE register. If extended address mode is selected bit 0 should be set to a zero for adherence to ADCCP/HDLC. If extended address mode is not selected, the command and response frame addresses should be located in the lower order byte of their respective fields.

Timers. There are 5 independent counter-timers. The lower 8 bits of IADR + 08 are used as a scaler for T1, T3, and TP. The scaler is driven by a clock which is 1/32 of SYSCLK. N1 is a 16 bit counter and is used to count the number of bytes in an I-frame. N2 is an 8 bit counter.

The Host will write the period of N1, N2, T1, T3, and TP into the Initialization Block.

	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
IADR + 06	COUNTER N1	
IADR + 08	COUNTER N2	SCALER
IADR + 10	TIMER T1	
IADR + 12	TIMER T3	
IADR + 14	TIMER TP	

N1. MAXIMUM FRAME LENGTH. This field must contain the two's complement of one less than the maximum allowable frame length, in bytes. Any frame received that exceeds this count will be discarded.

N2. MAXIMUM RETRANSMISSION COUNT. This field must contain the two's complement of one less than the maximum number of retransmissions that will be made following the expiration of T1.

SCALER. TIMER PRESCALER. Timers T1, T3, and TP are scaled by this number. The prescaler incremented once every 32 system clock pulses. When it reaches zero the timers are incremented and the prescaler is reset. This field is interpreted as the two's complement of the prescaler period. Note : a prescale value of one gives the smallest amount of scaling to the timers (64 clock pulses), zero gives the largest (8224 clock pulses).

T1. RETRANSMISSION TIMER. Link control frames will be retransmitted upon the expiration of the T1 timer if the appropriate response is not received. These frames will be retransmitted up to N2 (see above) times, at which time the link will be disconnected or reset by MK5025 according to the X.25 protocol. This field must contain the two's complement of the period of timer T1. The scaled (see SCALER) value of T1 should be made large enough to allow the remote station to receive the control frame and send its response.

T3. LINK IDLE TIMER. The link idle timer determines the amount of link idle time necessary to consider the link disconnected. This field must contain the two's complement of the period of timer T3. T3 is disabled if CSR5 RTSEN = 1 or if the MK5025 is in transparent mode.

TP. TRANSMIT POLLING TIMER. This scaled timer determines the length of time between transmit frame checks. Unless TDMD (see CSR0) is set or a frame is received on the link, no attempt to transmit a frame in the transmit descriptor ring is made until TP expires. At TP expiration all transmit frames in the transmit descriptor ring will be sent. This field must contain the two's complement of the period of timer TP.

Receive Descriptor Ring Pointer

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IADR + 16	0	RLEN			RBSY	0	0	0	RDRA<23:16>							
IADR + 18	RDRA<15:00>															0

Bit 15. Reserved, must be written as a zero.

Bit <14 : 12>, RLEN. RECEIVE RING LENGTH is the number of entries in the Receive Ring expressed as a power of two.

RLEN	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

Bit 11 RBSY Setting this bit enables the generation of the Remote Busy Indication primitive (PPRIM = 5) whenever there is a change in the Remote Busy status.

Bit <10 : 08>. Reserved, must be written as zeroes.

Bit <07 : 00>/ <15 : 00>, RDRA. RECEIVE DESCRIPTOR RING ADDRESS is the base address of (lowest address) of the Receive Descriptor Ring. The Receive Ring must be aligned on a word boundary.

Transmit Descriptor Ring Pointer

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IADR + 20	0	TLEN			0	TWD			TDRA<23:16>							
IADR + 22	TDRA<15:00>															0

Bit 15. Reserved, must be written as a zero.

Bit <14 : 12>, TLEN. TRANSIT RING LENGTH is the number of entires in the Transmit Ring expressed as a power of two.

Bit 11. Reserved, must be written as a zero.

Bit <10 : 08>, TWD. TRANSMIT WINDOW is the window size of the Transmitter expressed as a power of two less one. TWD is the maximum number of I frames which may be transmitted without an acknowledgement. TWD is not allowed to be greater than 127.

TLEN	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

TWD	Window Size	
	EXTC = 0	EXTC = 1
0	NA	1
1	1	3
2	2	7
3	3	15
4	4	31
5	5	63
6	6	127
7	7	127

Bit <07 : 00>/ <15 : 00>, TDRA. TRANSMIT DESCRIPTOR RING ADDRESS is the base address of (lowest address) of the Transmit Descriptor Ring. The Transmit Ring must be aligned on a word boundary.

XID/TEST Descriptors. The XID/TEST Descriptors contain pointers to the buffers used to receive and transmit XID and TEST frames, as well as the buffer lengths. The exact format of these descriptors can be seen below under Receive and Transmit Message Descriptor Entry descriptions. They are used the same as other descriptors except that no data chaining is allowed.

Status Buffer Address

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IADR + 40	0	0	0	0	0	0	0	0	SBA<23:16>							
IADR + 42	SBA<15:00>															0

Bit <15 : 08>. Reserved, must be written as zeroes.

Bit <07 : 00>/ <15 : 00>, SBA. STATUS BUFFER ADDRESS points to a 7 word buffer into which link status information is placed upon the issuance of the STAT primitive by the HOST. The contents of the Status Buffer are described later in this document. The Status Buffer must begin on a word boundary.

Error Counters. Six locations in the initialization buffer are reserved for use as error counters which the MK5025 will increment. These are intended for use of the Host CPU for statistical analysis. The MK5025 will only increment the counters ; it is up to the user to clear and preset these counters. The error counters are :

MEMORY ADDRESS	ERROR COUNTER
IADR + 44	Bad frames received - Bad FCS - Non-Octet Aligned
IADR + 46	Number of FRMR frames received
IADR + 48	Number of T1 timeouts
IADR + 50	Number of REJ frames received
IADR + 52	Number of REJ frames transmitted
IADR + 54	Frames shorter than minimum length received

Receive and Transmit Descriptor Rings. Each descriptor ring in memory is a 4 word entry. The following is the format of the receive and transmit descriptors.

Receive Message Descriptor 0 (RMD0)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
OWNA	OWNB	SLF	ELF	UIR	FRMRR	RADR	RPF	RBADR<23:16>							

Bit 15, OWNA. When this bit is a zero either the HOST or the I/O ACCELERATION PROCESSOR owns this descriptor. When this bit is a one the MK5025 owns this descriptor. The chip clears the OWNA bit after filling the buffer pointed to by the descriptor entry provided the received frame had a good FCS, N(r), and N(s). The Host sets the OWNA bit after emptying the buffer. Once the MK5025, Host, or I/O acceleration processor has relinquished ownership of a buffer, it may not change any field in the four words that comprise the descriptor entry.

Bit 14, OWNB. This bit determines whether the HOST or the SLAVE PROCESSOR owns the buffer when OWNA is a zero. The MK5025 never uses this bit. This bit is provided to facilitate use of an I/O acceleration processor.

Bit 13, SLF. Start of Long Frame indicates that this is the first buffer used by MK5025 for this frame. It is used for data chaining buffers. SLF is set by the chip.

***Note :** A "Long Frame" is any frame which needs data chaining. Usually this will be an I frame, but it could also be a UI or FRMR frame.*

Bit 12, ELF. End of Long Frame indicates that this the last buffer used by MK5025 for this frame. It is used for data chaining buffers. If both SLF and ELF were set the frame would fit into one buffer and no data chaining would be required. ELF is set by the MK5025.

Bit 11, UIR. UI Frame Received indicates that a UI frame has been received and is stored in this buffer.

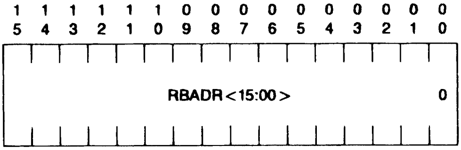
Bit 10, FRMRR. FRMR Received indicates that the I-field of a FRMR is stored in the buffer referenced by this Message Descriptor.

Bit 09, RADR. Valid only for frames received while in Transparent Mode with address filtering enabled. RADR indicates which of the 2 programmable addresses the frame matched. If set, the received address field matched the value in the Local Address field of the Initialization Block. Otherwise it matched the value in the Remote Address field.

Bit 08, RPF. Valid only for UI, XID and TEST frames. RPF equals the state of the P or F bit for the recvd frame.

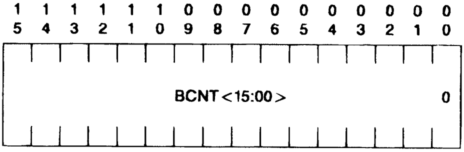
Bit <07 : 00>, RBADR. The High Order 8 address bits of the buffer pointed to by this descriptor. This field is written by the Host and unchanged by MK5025.

Receive Message Descriptor 1 (RMD1)



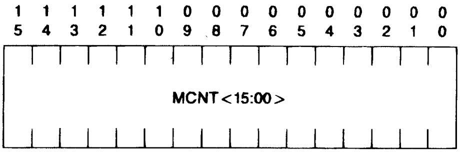
Bit <15 : 00>, RBADR. The low order 16 address bits of the receive buffer pointed to by this descriptor. RBADR is written by the Host CPU and unchanged by MK5025. The receive buffers must be word aligned.

Receive Message Descriptor 2 (RMD2)



Bit <15 : 00>, BCNT. Buffer Byte Count is the length of the buffer pointed to by this descriptor expressed in two's complement. This field is written to by the Host and unchanged by MK5025. Buffer size must be even.

Receive Message Descriptor 3 (RMD3)



Bit <15 : 00>, MCNT. Message Byte Count is the length, in bytes, of the contents of the buffer expressed in two's complement. If ELF = 0, MCNT will equal the two's complement of BCNT since the MK5025 will fill a buffer before chaining to the next descriptor.

Transmit Message Descriptor 0 (TMD0)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
OWNA	OWNB	SLF	ELF	TUI	TINTD	0	XPF	TBADR<23:16>							

Bit 15, OWNA. When this bit is a zero either the HOST or the SLAVE PROCESSOR owns this descriptor. When this bit is a one the MK5025 owns this descriptor. The host should set the OWNA bit after filling the buffer pointed to by the descriptor entry. The MK5025 releases the descriptor after transmitting the buffer and receiving the proper acknowledgement from the remote station. After the MK5025, Host, or I/O acceleration processor has relinquished ownership of a buffer, it may not change any field in the four words that comprise the descriptor entry.

Bit 14, OWNB. This bit determines whether the HOST or the I/O ACCELERATION PROCESSOR owns the buffer when OWNA is a zero. The MK5025 never uses this bit. This bit is provided to facilitate use of an I/O acceleration processor.

Bit 13, SLF. Start of Long Frame indicates that this is the first buffer used by MK5025 for this frame. It is used for data chaining buffers. SLF is set by the Host.

Note : A "Long Frame" is any frame which needs data chaining. Usually this will be an I frame, but it could also be a UI frame or others.

Bit 12, ELF. End of Long Frame indicates that this the last buffer used by MK5025 for this frame. It is used for data chaining buffers. If both SLF and ELF were set the frame would fit into one buffer and no data chaining would be required. ELF is set by the Host.

Bit 11, TUI. Transmit a UI frame indicates that a UI frame is to be transmitted from the transmit buffer instead of a normal I frame. This bit must also be set for anything transmitted while the MK5025 is in Transparent Mode.

Bit 10, TINTD Transmit Interrupt Disable. If this bit is set, no transmit interrupt is generated when ownership of this descriptor is released back to the host. This allows users to limit the number of transmit interrupts.

Bit 09 Reserved, must be written as zeroes.

Bit 8, XPF. Transmit P/F bit instructs the MK5025 to send the corresponding frame with a particular value for the P/F bit. This bit should equal the desired value of the transmitted P/F bit. This bit is valid only for UI, XID and TEST frames and should be written with zero otherwise.

Bit <07 : 00>, TBADR. The High Order 8 address bits of the buffer pointed to by this descriptor. This field is written by the Host and unchanged by MK5025.

Transmit Message Descriptor 1 (TMD1)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
TBADR <15:00>															0

Bit <15 : 00>, TBADR. The Low Order 16 address bits of the buffer pointed to by this descriptor. TBADR is written by the Host and unchanged by MK5025. The least significant bit is zero since the descriptor must be word aligned.

Transmit Message Descriptor 2 (TMD2)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
BCNT <15:00>															0

Bit <15 : 00>, BCNT. Buffer Byte Count is the usable length, in bytes, of the buffer pointed to by this descriptor in two's complement. This field is not used by the MK5025.

Transmit Message Descriptor 3 (TMD3)

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
MCNT<15:00>															

Bit <15 : 00>, MCNT. Message byte count is the length, in bytes, of the contents of the buffer associated with this descriptor expressed as a two's complement.

Status Buffer

	15 14 13 12 11 10 09 08	07 06 05 04 03 02 01 00
SBA + 00	V(r)	V(s)
SBA + 02	Local State	Remote State
SBA + 04	Phase	V(A)
SBA + 06	Unused	CURRD<23:16>
SBA + 08	CURRD<15:00>	
SBA + 10	Unused	CURXD<23:16>
SBA + 12	CURXD<15:00>	

V(r). Current value of the Receive Count Variable. 0 ≤ V(r) ≤ 7 for non-extended control ; 0 ≤ V(r) ≤ 127 for extended control.

V(s). Current value of the Transmit Count Variable. 0 ⇐ V(s) ⇐ 7 for non-extended control ; 0 ⇐ V(s) ⇐ 127 for extended control.

Local State. Current state of local station.

Value	Description
0	Normal Data Transfer state
1	Local Busy state
2	REJ sent state
3	DISC sent state
4	Normal Disconnected state
5	SABM/E sent for link connection
6	FRMR sent state
7	SABM/E sent for link reset
8	Error Indication issued

Remote State. Current state of remote station.

Value	Description
0	Normal Data Transfer state
1	Remote Busy state

Phase. Current phase of operation.

Value	Description
– 1	Stopped Mode
0	Information Transfer phase
1	Disconnected phase
2	Resetting phase
3	Transparent Data Transfer phase

V(A). Current value of Transmit Acknowledge Count.. This field contains the value of the N(r) of the most recently received S or I frame. The modulo difference between V(A) and V(s) determines the number of outstanding I frames that have not been acknowledged by the remote station.

CURRD <23 : 00>. Current Receive Descriptor. This pointer indicates the position of the descriptor for the next receive buffer to be filled.

CURRXD <23 : 00>. Current Transmit Descriptor. This pointer indicates the position of the descriptor for the next transmit buffer to be transmitted.

Data Link Services

The MK5025 is consistent with the ISO Data Link Service Definition in providing services to the HOST. The following section is a brief description of this interface.

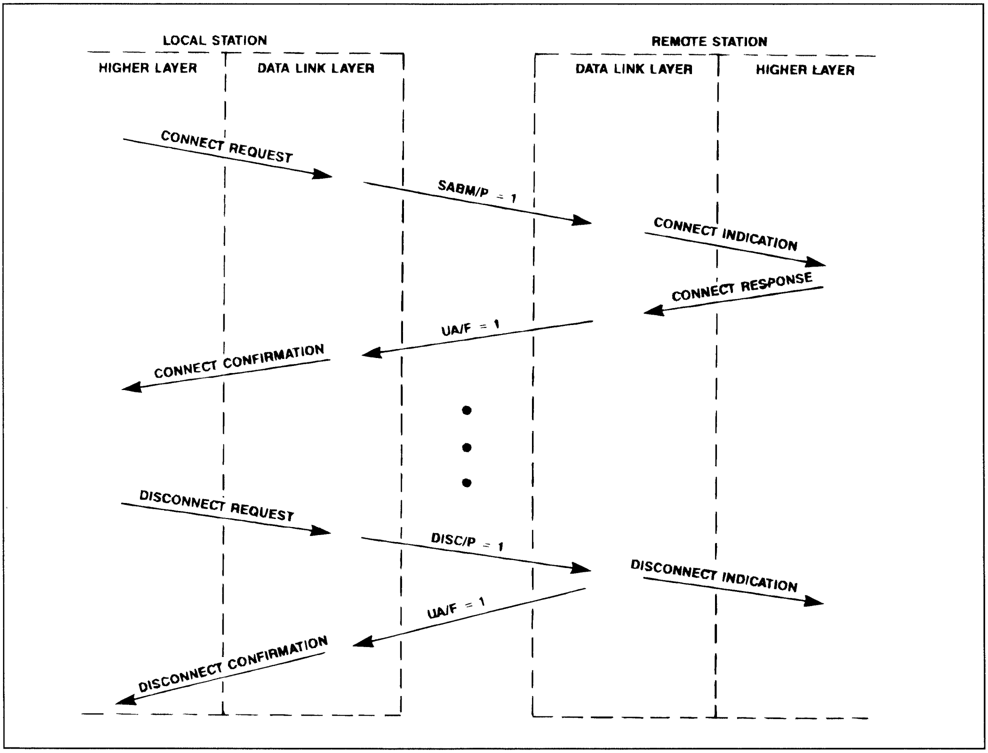
All link oriented services are provided through the exchange of Data Link Service Primitives. These primitives provide services to the HOST. Each primitive falls into one of the following categories :

1. Link Establishment (Connection)
2. Link Resetting
3. Link Disconnection
4. Data Transfer

A primitive is also one of the following types :

1. Request
2. Response
3. Indication
4. Confirmation

Figure 7 : Examples of Confirmed Data Link Services.



Requests and Responses are issued by the HOST and Indications and Confirmations are issued by the MK5025.

A Request will be issued by the HOST when a service is required. An Indication will be issued by the MK5025 when the remote system is attempting to change the data link status. A Response is issued by the HOST when receiving an indication for a confirmed service. A confirmation is issued by the MK5025 when the remote system has responded to a previously requested service.

In the MK5025, primitives are exchanged two ways : through the CSR1 and through the OWN bits in the descriptor rings. Connection, disconnection, and link reset primitives are exchanged through CSR1. Data transfer primitives are handled transparently by the OWN bit handshaking in the Descriptor Rings.

Nine additional primitives have been included in the MK5025 to handle services not mentioned in the ISO Data Link Service definition. These primitives include :

STOP - Disables the MK5025 from link operation. MK5025 transmits 1's.
INIT - Instructs the MK5025 to read the initialization block.
START - Enables the MK5025 for link operation. MK5025 starts sending flags.
TRANS - Enables the MK5025 for transparent operation. MK5025 starts sending flags.
STAT - Instructs the MK5025 to write chip status in the status buffer.
STEST - Instructs the MK5025 to perform an internal self test.
ERROR - Indicates the occurence of a link error requiring higher level action.
XID - Confirmed exchange of identification (optional).
TEST - Provides a full remote loopback test facility (optional).

For examples of the use of primitives, see the section on detailed programming procedures below.

Detailed Programming Procedures

Initialization. The following procedures should be followed to initialize the MK5025 :

1. Setup bus control information in CSR4.
2. Setup Initialization Block and Descriptor Rings and load the address of the initialization block in CSR's 2 and 3.
3. Issue the INIT primitive through CSR1 instructing the MK5025 to read the initialization block information.
4. Wait for INIT Confirmation primitive from the MK5025.
5. Issue the START Primitive through CSR1 to enable the MK5025 for link operation.
6. Enable interrupts in CSR0 if desired.

Active Link Setup. The following procedures should be followed to actively establish a link :

1. Issue the Connect Request primitive through CSR1. The MK5025 will attempt to establish a logical link.
2. Wait for a Connect Confirm primitive from the MK5025.
3. If a Connect Confirm primitive is received, a link has been established.
4. If a Disconnect Indication primitive is received, the MK5025 has been unable to establish a link. The reason will be in the PPARM field of CSR1.

Passive Link Setup. The following procedures should be followed to passively establish a link :

1. Issue a Disconnect Request primitive. A DM frame with F bit clear will be sent to the remote station requesting link setup. This step is optional.
2. Wait for a Connect Indication primitive from the MK5025.
3. If a Connect Indication primitive is received, issue a Connect Response primitive to indicate willingness to establish the link. The link is now established.
4. If no Connect Indication primitive is received, the remote site is not trying to actively setup a link.

Refusing Link Setup. The following procedure should be followed when refusing link establishment :

1. A Connect Indication will be received indicating a request by the remote station to establish a link.
2. Issue a Disconnect Request to refuse the link establishment request.

Sending Data. The following procedure should be followed to send a data frame :

1. Wait for OWNA bit of current transmit descriptor to be cleared, if not already.
2. Fill buffer associated with current transmit descriptor with data to be sent, or set descriptor buffer address to any already filled buffer.
3. Repeat steps 1 and 2 for next buffer if chaining is necessary, setting SLF, ELF and MCNT appropriately.
4. Set the OWNA bit for each descriptor used.
5. Go on to next descriptor, these OWNA bits will be cleared when the data has been sucessfully sent and acknowledged. In Transparent Mode, OWNA bits are cleared immediately after frame transmission.

Receiving Data. The following procedure should be followed when receiving a data frame :

1. Make sure that the OWNA bit of the current receive descriptor is clear.
2. Read data out of buffer associated with current receive descriptor.
3. Set the OWNA bit of current descriptor.
4. If ELF bit of current descriptor is clear, go on to next descriptor and repeat above steps appending data from each buffer until a descriptor with the ELF bit set is reached.

Link Disconnection. The following procedure should be followed to disconnect an established link :

1. Issue the Disconnect Request primitive to the MK5025. The MK5025 will disconnect the link.
2. A Disconnect Confirmation will be issued after successful disconnection.

Link Reset. The following procedure should be followed to reset an established link :

1. Issue a Reset Request primitive to the MK5025.
2. Wait for a Reset Confirm primitive from the MK5025.
3. If a Reset Confirm primitive is received, the link has been reset.
4. If a Disconnect Indication is received, the MK5025 was unable to reset and has disconnected. The reason for failure is in the PPARM field of CSR1. Link connection procedures must now be performed to re-establish the link.

Receiving Link Reset. The following procedure should be followed when receiving a request for link reset :

1. A Reset Indication will be received from the MK5025 indicating the remote station has requested a link resetting.
2. If able to reset, issue a reset response to indicate willingness to reset the link.
3. If unable to reset, issue a Disconnect Request to disconnect the link.

Receiving FRMR Frame. The following procedure should be followed when receiving a FRMR :

1. An Error Indication will be received from MK5025 indicating an error condition. PPARM will indicate a FRMR frame has been received. The I-field of the FRMR has been placed in the next Receive Descriptor.
2. If able to reset, issue a Reset Request to MK5025 and wait for either a Reset Indication or a Disconnect Indication as described above for Link Reset.
3. If unable to reset, issue a Disconnect Request to disconnect the current link. Link setup procedures should now be performed to re-establish a link.

Exchanging Identification. The following procedure should be followed to exchange identification with the remote :

1. XIDE in CSR3 must be set prior to any identification exchange.
2. Place identification information in the XID/TEST Transmit Buffer.
3. Issue an XID Request primitive.
4. If an XID Confirm primitive is received, the identication exchange has been performed and the remote response is located in the XID/TEST receive buffer.
5. If a Disconnect Indication is received, the identification exchange was unsuccessful.

Receiving an Identification Request. The following procedure should be performed when receiving a request for identification :

1. An XID Indication primitive will be received from the MK5025 to indicate the request for identification. The remote identification information will be located in the XID/TEST receive buffer.
2. To respond, place identification information in the XID/TEST send buffer and issue an XID Response primitive.
3. To refuse, issue a Disconnect Request primitive.

***Note :** An XID Indication will only be issued if the XIDE bit in CSR3 has been set. Otherwise, all identification requests will automatically be refused and XID frames will not be recognized.*

Disabling the MK5025. The following procedure should be followed to disable the MK5025 :

1. Issue the STOP primitive. This will disable the MK5025 from receiving or transmitting. The TD pin will be held high while the MK5025 is in stopped mode. The STOP bit in CSR0 will be set and interrupts disabled. If a link is currently established, data may be lost.

Re-enabling the MK5025. The same procedure should be followed for re-enabling the MK5025 as was used to initialize upon power-up. If the Initialization Block and the hardware configuration have not changed then steps 1 and 2 may be omitted.

MK5025 Internal Self Test. The MK5025 contains an easy to use internal built-in self test designed to test, with a high fault coverage, all of the major blocks of the device except for the DMA controller. It is suggested that a loopback test also be performed to more completely test the DMA controller.

The following procedure should be followed to execute internal self test :

1. Reset the device using the $\overline{\text{RESET}}$ pin.
2. Set bit 04 of CSR4.
3. Issue a Self Test Request request through CSR1.
4. Poll CSR1, waiting for the PAV bit in CSR1 to be set by the MK5025.
5. After the PAV bit is set, read CSR1. The success or failure of the test is indicated in the PPRIM and PPARM fields as follows :

PPARM	PPRIM	RESULT
0	0	Passed self test
1	1	Failed the reset test of the self test
1	2	Failed self test in the micro controller RAM
1	3	Failed self test in the ALU
1	4	Failed self test in the timers
1	5	Failed self test in the transmitter and/or receiver
1	6	Failed self test in the CSRs and/or bus master
Otherwise		failed device.

6. If the PAV bit is not set within 75 msec (SYSCLK = 10MHz), the MK5025 is unable to respond to the Self Test Request and will not complete it successfully.

If the self test passes, then it may be immediately re-executed from step 3, otherwise re-execution should proceed from step 1.

ELECTRICAL SPECIFICATION

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_I	Voltage on any Pin Relative to GND	– 0.5 to V_{CC} + 0.5	V
T_A	Ambient Operating Temperature Under Bias	– 25 to + 100	°C
T_{stg}	Ambient Storage Temperature	– 65 to + 150	°C
P_D	Total Device Power Dissipation	0.5	W

Note : Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = + 5V ± 5%, unless otherwise specified)

Symbol	Conditions	Min.	Typ.	Max.	Unit
V_{IL}		– 0.5		+ 0.8	V
V_{IH}		+ 2.0		V_{CC}+0.5	V
V_{OL}	@ I_{OL} = 3.2mA			+ 0.5	V
V_{OH}	@ I_{OH} = – 0.4mA	+ 2.4			V
I_{IL}	@ V_{in} = 0.4 to V_{CC}			± 10	µA
I_{CC}	T_{SCT} = 100ns		50		mA

CAPACITANCE (Frequency = 1MHz)

Symbol	Conditions	Min.	Max.	Unit
C_{IN}			10	pF
C_{OUT}			10	pF
C_{IO}			20	pF

AC TIMING SPECIFICATIONS (T_A = 0°C to 70°C, V_{CC} = + 5V ± 5%, unless otherwise specified).

N°.	Signal	Symbol	Parameter	Test Conditions	Min. ns	Typ. ns	Max. ns
1	SYSCLK	T_{SCT}	SYSCLK period		100		2000
2	SYSCLK	T_{SCL}	SYSCLK low time		45		
3	SYSCLK	T_{SCH}	SYSCLK high time		45		
4	SYSCLK	T_{SCR}	Rise time of SYSCLK		0		8
5	SYSCLK	T_{SCF}	Fall time of SYSCLK		0		8
6	$\overline{\text{TCLK}}$	T_{TCT}	$\overline{\text{TCLK}}$ period		140		
7	$\overline{\text{TCLK}}$	T_{TCL}	$\overline{\text{TCLK}}$ low time		63		
8	$\overline{\text{TCLK}}$	T_{TCH}	$\overline{\text{TCLK}}$ high time		63		
9	$\overline{\text{TCLK}}$	T_{TCR}	Rise time of $\overline{\text{TCLK}}$		0		8
10	$\overline{\text{TCLK}}$	T_{TCF}	Fall time of $\overline{\text{TCLK}}$		0		8
11	TD	T_{TDP}	TD data propagation delay after the falling edge of $\overline{\text{TCLK}}$	C_L = 50pF			40
12	TD	T_{TDH}	TD data hold time after the falling edge of $\overline{\text{TCLK}}$	C_L = 50pF	5		

AC TIMING SPECIFICATIONS (T_A = 0°C to 70°C, V_{CC} = + 5V ± 5%, unless otherwise specified)

N°.	Signal	Symbol	Parameter	Test Conditions	Min. ns	Typ. ns	Max. ns
13	$\overline{\text{RCLK}}$	T_{RCT}	$\overline{\text{RCLK}}$ Period		140		
14	$\overline{\text{RCLK}}$	T_{RCH}	$\overline{\text{RCLK}}$ High Time		63		
15	$\overline{\text{RCLK}}$	T_{RCL}	$\overline{\text{RCLK}}$ Low Time		63		
16	$\overline{\text{RCLK}}$	T_{RCR}	Rise Time of $\overline{\text{RCLK}}$		0		8
17	$\overline{\text{RCLK}}$	T_{RCF}	Fall Time of $\overline{\text{RCLK}}$		0		8
18	RD	T_{RDR}	RD Data Rise Time		0		8
19	RD	T_{RDF}	RD Data Fall Time		0		8
20	RD	T_{RDH}	RD Hold Time after Rising Edge of $\overline{\text{RCLK}}$		5		
21	RD	T_{RDS}	RD setup Time Prior to Rising Edge of $\overline{\text{RCLK}}$		30		
22	A/ADL	T_{DOFF}	Bus Master Driver Disable after Rising Edge of $\overline{\text{HOLD}}$		0		50
23	A/DAL	T_{DON}	Bus Master Driver Enable after Falling Edge of $\overline{\text{HLDA}}$	T_{SCT} = 100nS	0		200
24	$\overline{\text{HLDA}}$	T_{HHA}	Delay to Falling Edge of $\overline{\text{HLDA}}$ from Falling Edge of $\overline{\text{HOLD}}$ (bus master)		0		
25	$\overline{\text{RESET}}$	T_{RW}	$\overline{\text{RESET}}$ Pulse Width		30		
26	A/DAL	T_{CYCLE}	Read/write, Address/data Cycle Time	T_{SCT} = 100nS	600		
27	A	T_{XAS}	Address Setup Time to Falling Edge of ALE		100		
28	A	T_{XAH}	Address Hold Time after the Rising Edge of $\overline{\text{DAS}}$		50		
29	DAL	T_{AS}	Address Setup Time to the Falling Edge of ALE		75		
30	DAL	T_{AH}	Address Hold Time after the Falling Edge of ALE		20		
31	DAL	T_{RDAS}	Data Setup Time to the Rising Edge of $\overline{\text{DAS}}$ (bus maaster read)		55		
32	DAL	T_{RDAH}	Data Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus master read)		0		
33	DAL	T_{DDAS}	Data Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus master write)		0		
34	DAL	T_{WDS}	Data Setup Time to the Rising Edge of $\overline{\text{DAS}}$ (bus master write)		250		
35	DAL	TWDH	Data Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus mater write)		35		
36	DAL	TSRDH	Data Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave read)	T_{SCT} = 100nS	0		35
37	DAL	TSWDH	Data Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave write)		0		
38	DAL	TSWDS	Data Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus slave write)		0		
39	ALE	TALEW	ALE width high		110		

AC TIMING SPECIFICATIONS (T_A = 0°C to 70°C, V_{CC} = + 5V ± 5%, unless otherwise specified).

N°.	Signal	Symbol	Parameter	Test Conditions	Min. ns	Typ. ns	Max. ns
40	$\overline{\text{ALE}}$	T_{DALE}	Delay from rising edge of $\overline{\text{DAS}}$ to the rising edge of ALE		70		
41	$\overline{\text{DAS}}$	T_{DSW}	DAS width low		200		
42	$\overline{\text{DAS}}$	T_{ADAS}	Delay from the falling edge of ALE to the falling edge of $\overline{\text{DAS}}$		80		
43	$\overline{\text{DAS}}$	T_{RIDF}	Delay from the rising edge of $\overline{\text{DALO}}$ to the falling edge of $\overline{\text{DAS}}$ (Bus master read)		35		
44	$\overline{\text{DAS}}$	T_{RDYS}	Delay from the falling edge of $\overline{\text{READY}}$ to the rising edge of $\overline{\text{DAS}}$	T_{ARYD} = 300nS T_{SCT} = 100nS	120		200
45	$\overline{\text{DALI}}$	T_{ROIF}	Delay from the rising edge of $\overline{\text{DALO}}$ to the falling edge of $\overline{\text{DALI}}$ (Bus master read)		70		
46	$\overline{\text{DALI}}$	T_{RIS}	$\overline{\text{DALI}}$ setup time to the rising edge of $\overline{\text{DAS}}$ (Bus master read)		150		
47	$\overline{\text{DALI}}$	T_{RIH}	$\overline{\text{DALI}}$ hold time after the rising edge of $\overline{\text{DAS}}$ (Bus master read)		0		
48	$\overline{\text{DALI}}$	T_{RIOF}	Delay from the rising edge of $\overline{\text{DALI}}$ to the falling edge of $\overline{\text{DALO}}$ (Bus master read)		70		
49	$\overline{\text{DALO}}$	T_{OS}	$\overline{\text{DALO}}$ setup time to the falling edge of $\overline{\text{ALE}}$ (Bus master read)		110		
50	$\overline{\text{DALO}}$	T_{ROH}	$\overline{\text{DALO}}$ hold time after the falling edge of ALE (Bus master read)		35		
51	$\overline{\text{DALO}}$	T_{WDIS}	Delay from rising edge of $\overline{\text{DAS}}$ to the rising edge of DALO (bus master write)		50		
52	CS	T_{CSH}	$\overline{\text{CS}}$ hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave)		0		
53	$\overline{\text{CS}}$	T_{CSS}	$\overline{\text{CS}}$ setup time to the falling edge of $\overline{\text{DAS}}$ (bus slave)		0		
54	ADR	T_{SAH}	$\overline{\text{ADR}}$ hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave)		0		
55	ADR	T_{SAS}	ADR setup time to the falling edge of $\overline{\text{DAS}}$ (bus slave)		0		
56	$\overline{\text{READY}}$	T_{ARYD}	Delay from the falling edge of ALE to the falling edge of $\overline{\text{READY}}$ to insure a minimum bus cycle (600nS)	T_{SCT} = 100nS			150
57	$\overline{\text{READY}}$	T_{SRDS}	Data setup time to the falling edge of $\overline{\text{READY}}$ (bus slave read)		75		
58	$\overline{\text{READY}}$	T_{RDYH}	$\overline{\text{READY}}$ hold time after the rising edge of $\overline{\text{DAS}}$ (bus master)		0		
59	$\overline{\text{READY}}$	T_{SRYH}	$\overline{\text{READY}}$ hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave)	T_{SCT} = 100nS	0		35
60	READ	T_{SRH}	$\overline{\text{READ}}$ hold time after the rising edge of $\overline{\text{DAS}}$ (bus slave)		0		
61	$\overline{\text{READ}}$	T_{SRS}	READ setup time to the falling edge of $\overline{\text{DAS}}$ (bus slave)		0		
62	$\overline{\text{READY}}$	T_{RDYD}	Delay from falling edge of $\overline{\text{DAS}}$ to falling edge of $\overline{\text{READY}}$ (bus slave read)	T_{SCT} = 100nS		200	

Figure 8 : Output Load Diagram.

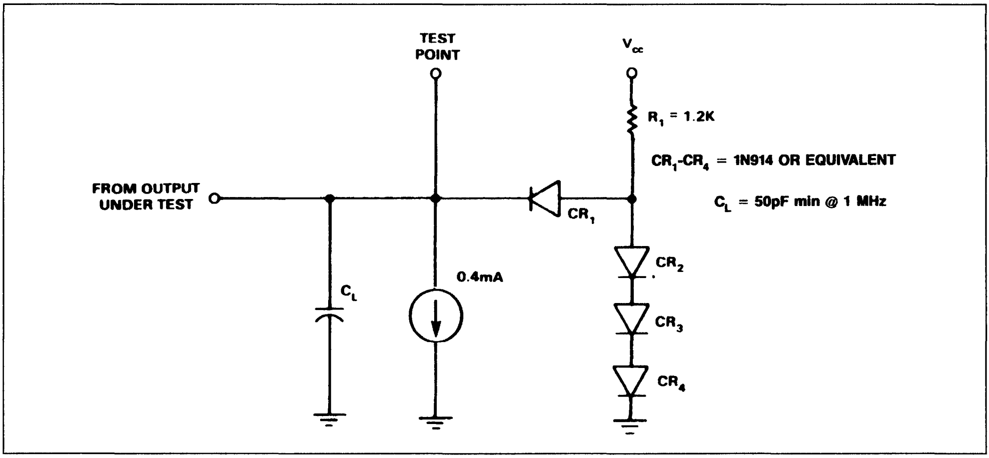
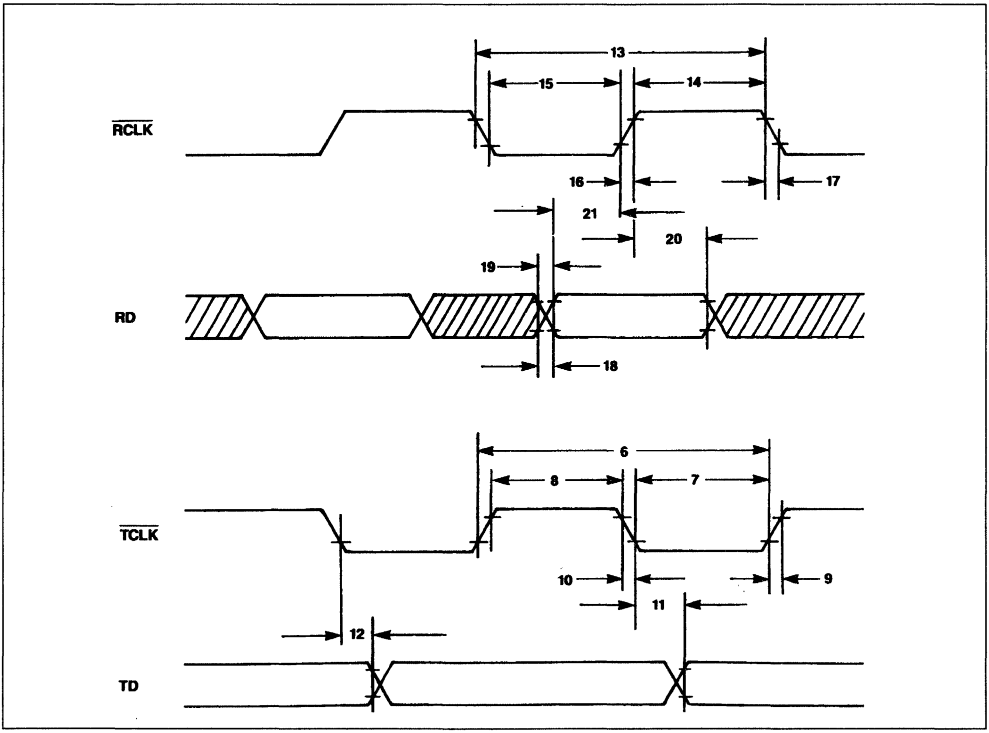


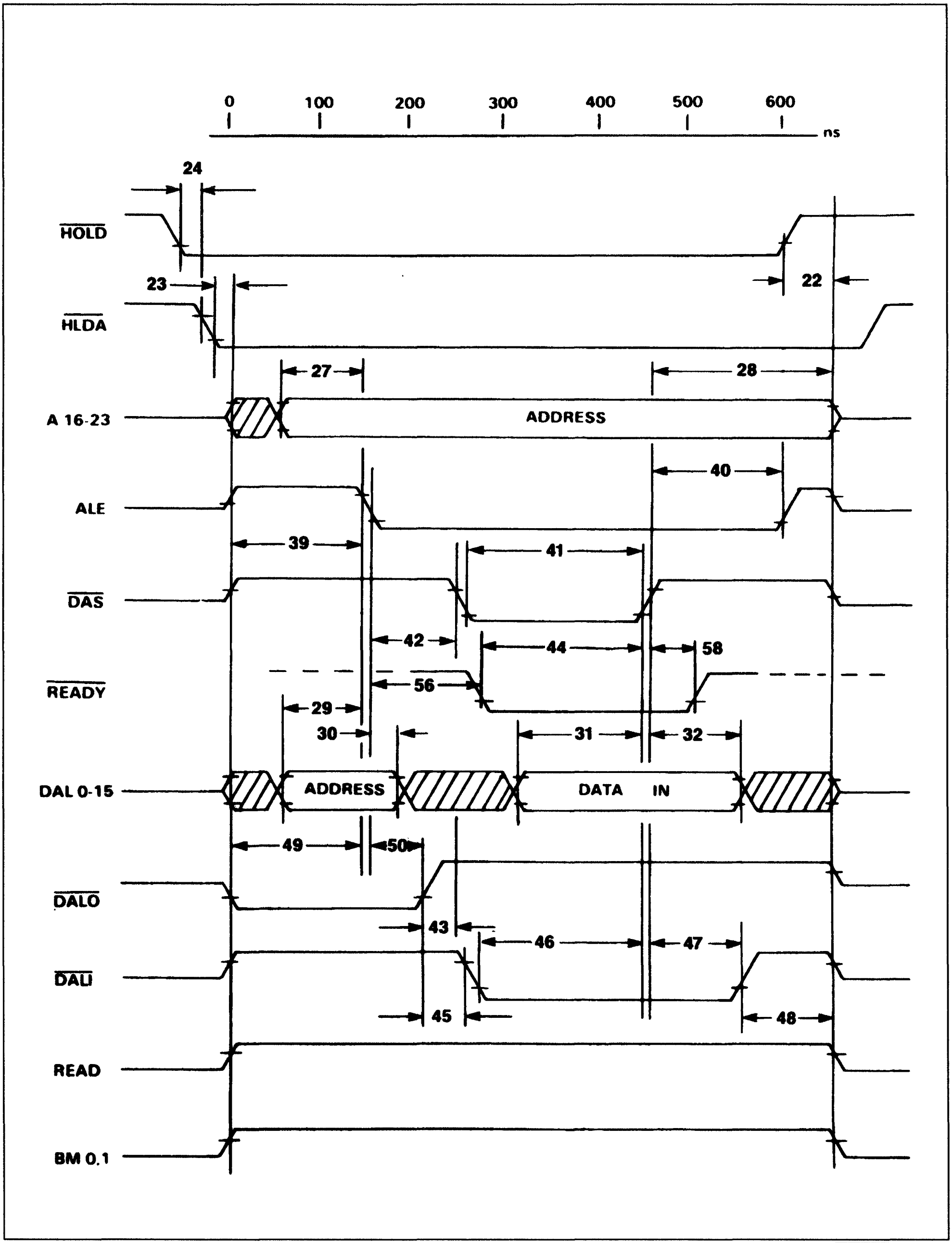
Figure 9 : Serial Link Timing Diagram.



Note : Timing Measurements are made at the following voltages, unless otherwise specified :

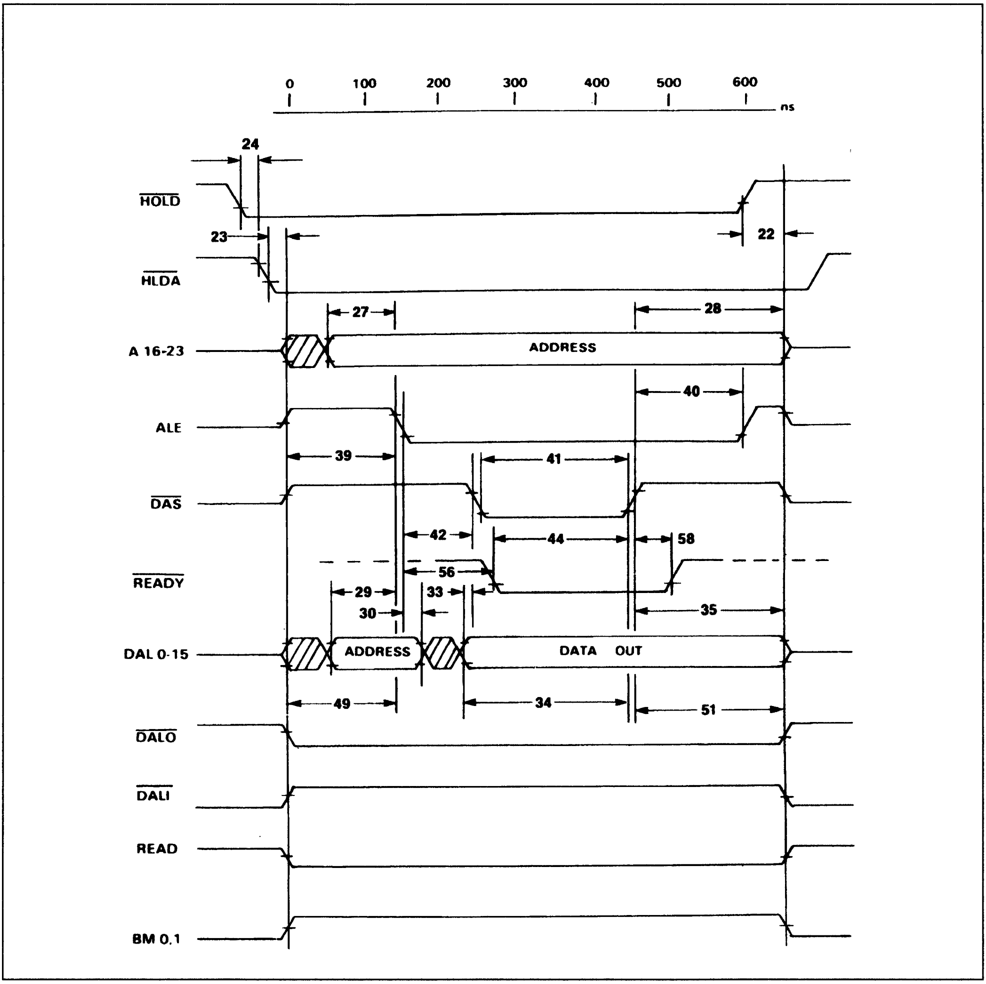
	"1"	"0"
OUTPUT	2.0V	0.8V
INPUT	2.0V	0.8V
FLOAT	90% V_{OH}	10% V_{OL}

Figure 10 : Bus Master Timing Diagram (Read).



Note : The Bus Master cycle time will increase from a minimum of 600ns in 100ns steps until the slave device returns READY.

Figure 11 : Bus Master Timing Diagram (Write).



Note : The Bus Master cycle time will increase from a minimum of 400ns in 100ns steps until the slave device returns READY.

Figure 12 : MK5025 Bus Slave Timing Diagram (Read).

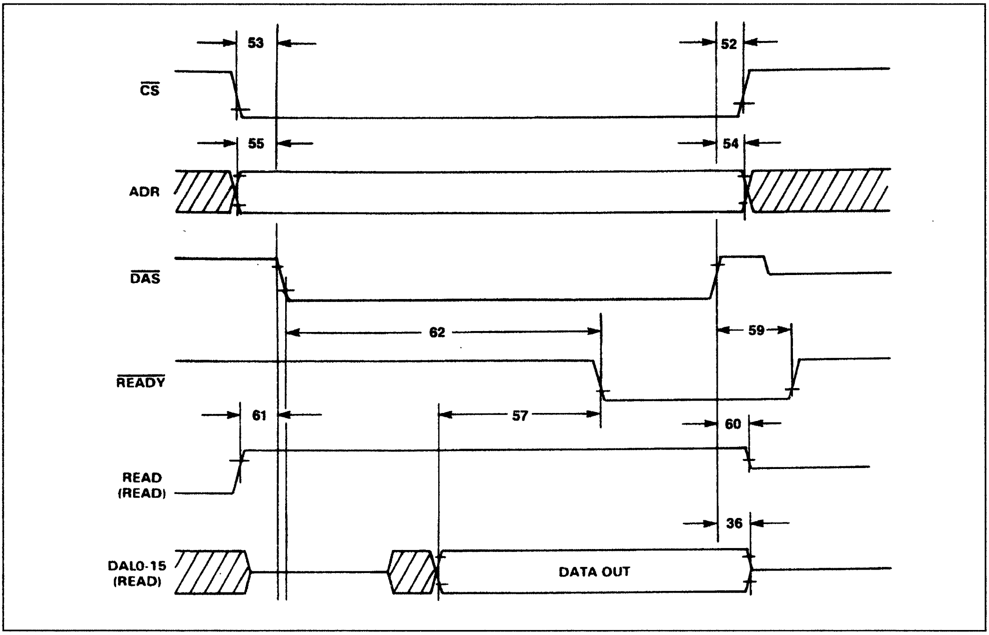
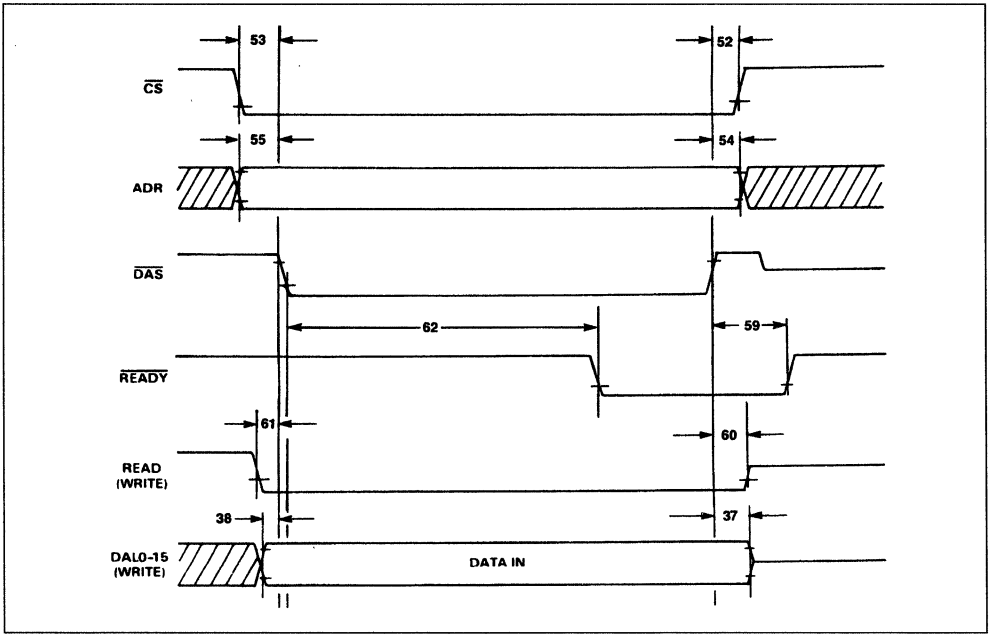
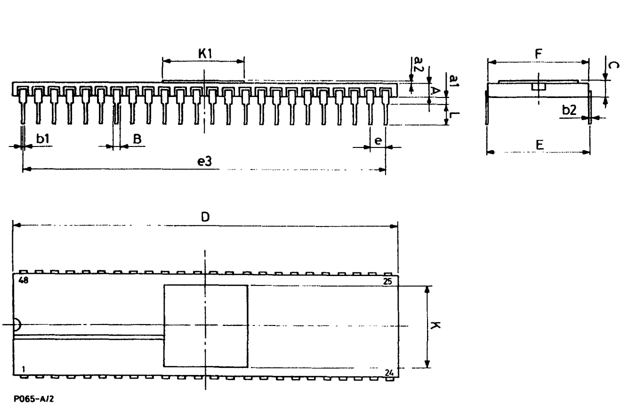


Figure 13 : MK5025 Bus Slave Timing Diagram (Write).



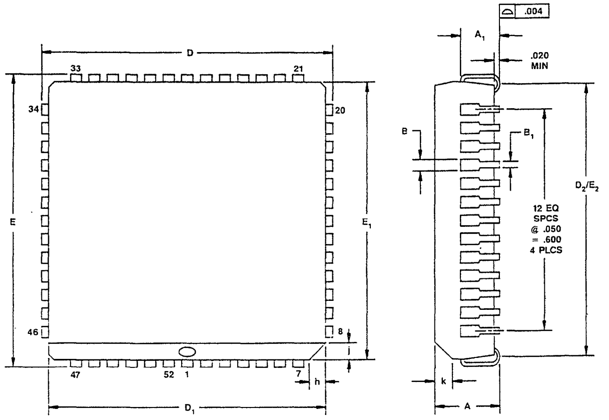
PACKAGES MECHANICAL DATA

Figure 14: MK5025P 48-Pin Ceramic Dual in Line Package.



Dim.	mm			Inches		
	Min.	Typ..	Max.	Min.	Typ.	Max.
A						
a1		1.27			.050	
a2						
B		1.02			.040	
b1		0.45			.017	
b2		0.25			.009	
C		2.46			.096	
D			61.59			2.424
E		15.24			.600	
e		2.54			.100	
e3		58.42			2.300	
F			15.34		.600	
L		3.30			.129	
K		12.80			.500	
K1		12.80			.500	

Figure 15 : MK5025Q 52-Lead Plastic Leaded Chip Carrier.



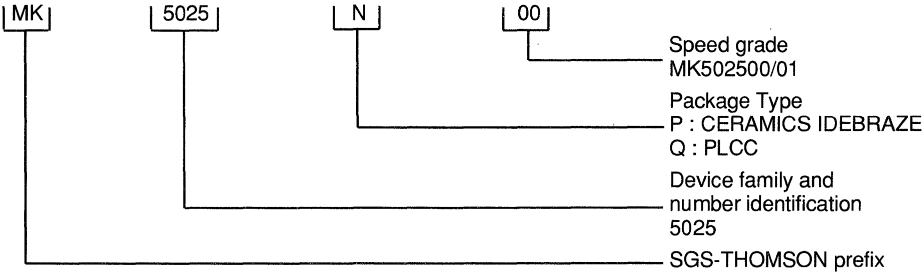
Dim.	Inches		Notes
	Min.	Typ..	
A	.165	.185	2
A_1	.090	.120	2
B	.026	.032	2
B_1	.013	.021	2
D	.785	.795	
D_1	.750	.756	
D_2	.690	.730	
E	.785	.795	
E_1	.750	.756	
E_2	.690	.730	
h	.042	.060	
j	.042	.060	
k	.042	.056	

PIN OUT :		17	DO NOT CONNECT	* = ACTIVE LOW		37	ADDRESS 22
1	VSS	18	BM1*/BUSAKO*	21	HLDA*	38	ADDRESS 21
2	DAL07	19	HOLD*/BURSQ*	22	CS*	39	ADDRESS 20
3	DAL06	20	ALE/AS*	23	ADR	40	ADDRESS 19
4	DAL05			24	READY*	41	ADDRESS 18
5	DAL04			25	RESET*	42	ADDRESS 17
6	DAL03			26	VSS (GROUND)	43	ADDRESS 16
7	DO NOT CONNECT			27	DO NOT CONNECT	44	DAL15
8	DAL02			28	TCLK*	45	DAL14
9	DAL01			29	DTR, RTS*	46	DAL13
10	DAL00			30	RCLK*	47	DAL12
11	READ			31	SYSCLK	48	DAL11
12	INTR*			32	TD	49	DAL10
13	DALI*			33	DSR, CTS*	50	DAL09
14	DALO*			34	RD	51	DAL08
15	DAS*			35	ADDRESS 23	52	VCC
16	BM0*BYTE/BUSREL*			36	DO NOT CONNECT		

ORDER CODES

Part Number	I/O	Data Rate	Clock Frequency	Temperature Range	Package Type
MK5025P-00/01		7MB/s	10MHz	0°C to 70°C	CDIP-48 600-MIL
MK5025P-I			10MHz	0°C to 70°C	CDIP-48 600-MIL
MK5025P-00/01		7MB/s	10MHz	0°C to 70°C	CDIP-48 600-MIL
MK5025Q-00/01		7MB/s	10MHz	0°C to 70°C	PLCC-52
MK5025Q			10MHz	0°C to 70°C	PLCC-52
MK5025Q-00/01		7MB/s	10MHz	0°C to 70°C	PLCC-52

Note : CDIP = Ceramic Multilayer DIP, PLCC = Plastic leaded Chip carrier, PDIP = Plastic DIP.



APPLICATION NOTE

MK5025 TRANSPARENT MODE

INTRODUCTION

The SGS-THOMSON X.25 Link Level Controller (MK5025) is a VLSI device which provides a complete link level data communication control conforming to the 1984 CCITT version of X.25. The MK5025 also supports X.32 (XID) and X.75 as well as single channel LAPD for ISDN with its UI frames and extended addressing capabilities.

One of the outstanding features of the MK5025 is its buffer managment which includes on-chip DMA. This feature allows users to handle multiple frames of receive and transmit data at a time. In order to utilize these buffer managment and DMA features with protocols not directly supported by the MK5025, a transparent mode is available for customized protocols using HDLC framing. This transparent mode provides an HDLC transport mechanism without link layer support. Extended addressing and control are optionally supported within transparent mode. Address filtering is also optional in this mode.

PURPOSE

The purpose of this application brief is to provide a detailed description of the MK5025 transparent mode and its options. Please refer to the MK5025 Technical Manual for more detailed information concerning the overall operation of the MK5025.

TRANSPARENT MODE ENTRY AND EXIT

To enter the transparent mode of operation, a user primitive *Trans* (UPRIM 3) is written to Control Status Register 1 (CSR1) after the completion of the first 4 steps of the *Initialization* procedure described in page 4-24 of the MK5025 Technical Manual. (The *Trans* Primitive is substituted for the *Start* primitive). The transmitter then begins to output flags, and data frames are transmitted and received via the descriptor rings, but no protocol processing is done. Address and Control Fields are not prepended to the frames, and FCS (Frame Check Sequence) processing may be enabled or disabled by the DRFCS and DTFCS bits in the mode register, as described on page 4-12 of the Technical Manual. Transparent mode may be exited only with a *Stop* primitive (UPRIM 0) or by bus reset.

CONTROL AND STATUS REGISTER OPERATION

In transparent mode all the control and status register mechanisms are still functional, but there are several bits that pertain only to the protocols directly supported by the MK5025 and are not valid in transparent mode. The following is a list of the validity of the bits in each CSR and the mode register when in transparent mode.

CSR0 - All bits in this register are valid in Transparent mode. It should be noted that interrupts can only be set (by setting bit 09, INEA = 1) once the device is in start or transparent mode.

CSR1 - All bits except PPARM are valid, although only for a few non-protocol related conditions. Since in transparent mode only a Stop User Primitive would have been valid, User Primitives 8 and 9 have been redefined. When in transparent mode, issuing User Primitive 8 (UPRIM 8) will start T1 timer and User Primitive 9 (UPRIM 9) will stop it. In this case Provider Primitive 8 (PPRIM 8) has been redefined to indicate expiry of T1 timer.

CSR2 - Aside from the IADR bits, only the PROM and XIDE bits have valid meaning in transparent mode. The XIDE bit can be set to 1 to enable global address recognition, and the PROM bit is used to disable address filtering in transparent mode.

CSR3 - As in CSR2, the IADR bits contain the address of the first word in the Initialization Block. Bits 00 - 07 in CSR2 contain the high order 8 bits, and bits 00 - 15 of CSR3 contain the low order 16 bits of the address of the first word of the Initialization Block.

CSR4 - All bits in this register are valid in transparent mode, including the FIFO watermarks and bursting operations.

CSR5 - All bits are valid in transparent mode.

MODE REGISTER OPERATION

All bits in the Mode Register are valid including MFS (Minimum Frame Spacing) and LBACK (Loopback). However, bits 09 (EXTAF) and 10 (EXTCF) are useful mainly in transparent mode, for forcing extended address and control field filtering. The DACE bit also offers further flexibility in transparent mode by allowing address and control fields to be treated as normal data when DACE = 1, as shown in table 2.

It should be noted that although DTFCS and DRFCS (bits 04 & 05) may be used to disable FCS generation and checking, the value in the FCS field of the received frame will not be stored in memory, even in transparent mode.

DESCRIPTOR RING OPERATION

The buffer management and descriptor ring operation is the same in transparent as non-transparent mode. It should be noted however that for the Transmit Message Descriptor, TUI (bit 11 of TMD0) should be set to 1 for anything transmitted in transparent mode. This is done because data transmitted in transparent mode is considered much the same as a UI frame rather than a normal I frame.

In transparent mode as in non-transparent mode, the Transmit Window size (TWD) in the Transmit Descriptor Ring Pointer must be set to a value greater than 0 for any transmission to occur. In fact, if TWD = 0 the MK5025 will not poll the Transmit Descriptor Ring.

ADDRESS FIELD FILTERING AND CONTROL FIELD OPERATION

The frame structure for HDLC is as follows :

F	A	C	I	FCS	F

where : F = Flag
A = Address field (A-field)
C = Control field (C-field)
FCS = Frame Check Sequence

According to HDLC rules, the A-field may be one or more octets in length. If the LSB of the first octet is 0, then the second octet is also part of the A-field. If the LSB of the second is 0, then the third octet is part of the A-field, and so on until an octet has an LSB = 1. The MK5025 allows the A-field to be one or two octets, depending upon the EXTA bit (Mode Register bit 10).

The C-field is one octet for modulo 8 for all frames. For modulo 128, the C-field is said to be extended, and is two octets for S (Supervisory) and I (Information) frames and one octet for U (Unnumbered) frames.

In the MK5025, address filtering and control field handling applies only to octet aligned frames received with good FCS. Any frame not meeting both of these conditions is discarded and the "Bad Frames Received" error counter (located at IADR + 44 of the Initialization Block) is incremented.

Table 1. MK5025 Address Filtering Options.

EXTA	EXTAF	XIDE	PROM	DACE	ADDRESS FILTERING
0	0	0	0	0	Single Octet Filtering L &R (Local & Remote adresses)
X	X	X	1	X	No Address Filtering, all Frames Accepted
0	0	1	0	0	Single Octet Filtering L & R and Global
0	X	X	X	1	Not Allowed
1	0	0	0	0	Double Octet Filtering L & R per HDLC Rules
1	0	0	0	1	Double Octet Filtering L & R per HDLC Rules
1	1	0	0	0	Double Octet Filtering L & R Regardless of A - Field LSB
0	1	X	X	0	Not Allowed

Notes : 1. EXTA = Extended address, Mode register bit 06. EXTAF = Extended address force, Mode register bit 09.
2. XIDE = XID enabled, CSR2 bit 08. PROM = Promiscuous mode, CSR2 bit 10.
3. DACE = Disable address and control field extraction for load to memory, Mode register bit 08.
4. L&R = Local and Remote addresses. X = Do not care.

In the transparent mode, address filtering is supported if the PROM bit (CSR2, bit 10) is 0. In this case, frames are accepted if the received A-field matches either the Local Station Address or the Remote Station Address as specified in the Initialization Block. This is a one octet compare if the extended address bit, EXTA is 0 (Mode register bit 06), or follows the HDLC rules for extended addressing if EXTA is 1. Frames not matching either address are ignored.

Extended control is also valid in transparent mode using the EXTC bit (Mode Register bit 07), as shown in table 1 and table 2. If EXTC is 0 then the C-field is one octet for all frames. If however EXTC is set to 1, the MK5025 will look to see if either of the two least significant bits of the C-field is 0. If so, the frame is said to have an extended control field which is two octets. In addition, bits EXTAF and EXTCF (Mode Register bit 09 & 10) are useful in transparent mode to force extended address and control. If EXTAF is set along with EXTA, the receiver will assume the address field to be two bytes long regardless of the first bit of the address field. If EXTCF is set along with EXTC, the receiver will assume the control field to be two bytes long regardless of the first two bits of that field.

For global addresses, the XIDE bit is valid in transparent mode, depending upon the settings of the other bits in the Mode Register, as shown in table 1. If XID is enabled by setting bit XIDE (CSR2 bit 08) to 1, then all frames with address "11111111" are accepted. Even frames which are not XID are accepted. In this case, a global address is considered as a command frame.

Address and control field extraction can be disabled in transparent mode, through use of the if PROM bit (CSR2, bit 10) and DACE bit (Mode Register bit 08), as shown in table 2. If the PROM bit is 1, all frames are accepted. When both the PROM bit and the DACE bit are set to 1, the device is considered to be in total transparent mode. In this mode no protocol processing is done and all data after the opening flag and before the FCS is loaded into memory.

CONCLUSION

The MK5025 offers great flexibility to the data communications system designer. The on-chip protocol processing may be used to save the designer much time in implementing standard protocols such as X.25, LAPB, ISDN LAPD, X.32, and X.75, while still allowing the flexibility of implementing alternate or customized HDLC based protocols using the MK5025's transparent mode.

Table 2: Address and Control Field Handling by the MK5025 Receiver.

DACE	PROM	EXTA	EXTAF	EXTC	EXTCF	Address Field Handling	Control Field Handling
0	0	0	0	0	0	A Filtered	CC → MEM1
0	0	0	0	1	0	A Filtered	CC or EC → MEM1
0	0	1	0	0	0	A or EA Filtered	CC → MEM1
0	0	1	1	0	0	EA Filtered	CC → MEM1
0	0	1	0	1	0	A or EA Filtered	CC or EC → MEM1
0	0	1	1	1	1	EA Filtered	EC → MEM1
0	1	0	0	0	0	Not Filtered, AA → MEM1	CC → MEM2
0	1	0	0	1	0	Not Filtered, AA → MEM1	CC or EC → MEM2
0	1	1	0	0	0	Not Filtered, AA or EA→MEM1	CC → MEM2
0	1	1	1	0	0	Not Filtered, EA → MEM1	CC → MEM2
0	1	1	0	1	1	Not Filtered, AA or EA→MEM1	EC → MEM2
1	0	X	X	X	X	First 2 Octets Always Filtered	EC → MEM1
1	1	X	X	X	X	Total Transparent Mode	All Data after Opening Flag & before FCS → Memory

Notes :

1. MEM1 is the first location and MEM2 is the second location where received data is loaded. MEM1 and MEM2 are each 16 bits wide.
2. C is the received, single octet, control field. CC ⇒ MEMx means the single octet control field C is loaded into both bytes of a 16 bit memory location. Similarly, A is a single octet address field, and AA ⇒ MEMx means the single octet address field A is loaded into both bytes of a 16 bit memory location.
3. EC is an extended control field (2 octets) for received S and I frames. For received U frames, the control field is not extended (1 octet). This determines whether CC or EC ⇒ MEMx. However, when EXTCF is set to 1, the control field is always extended (EC = 2 octets).
4. EA is an extended address field (2 octets). "A or EA filtered" means that one octet of the A-field is filtered if the LSB = 1, or two octets are filtered if the LSB = 0. Similarly "AA or EA ⇒ MEM1" means that AA is loaded into memory if the LSB = 0 ; else, EA is loaded. This conforms to HDLC rules for extended address. However, if EXTAF is set to 1, two octets are filtered regardless of the LSB, and EA will be loaded into memory.
5. EXTCF = Extended control force, Mode register bit 10.
6. DACE, PROM, EXTA, EXTAF, and EXTC are as defined in the notes for table 1. X = Do not care.

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

MK5025
SYNCHRONOUS TIMING

INTRODUCTION

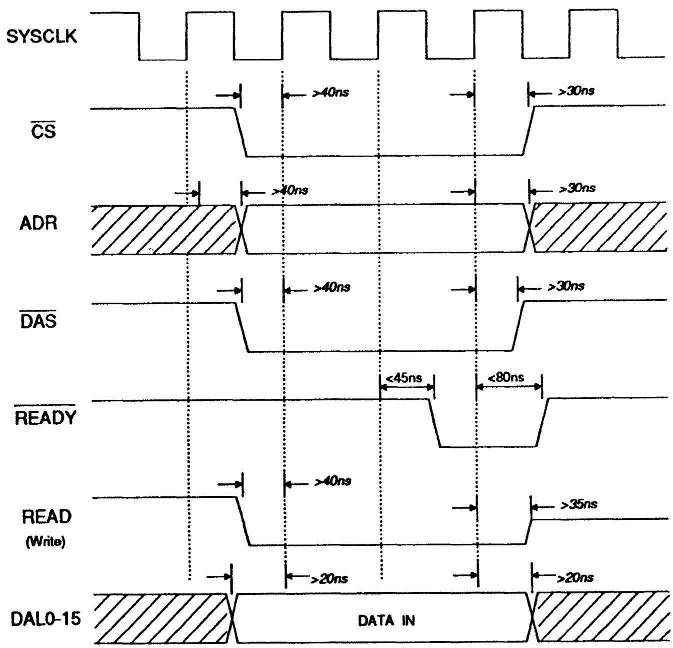
The SGS-THOMSON MK5025 X.25 Link Level Controller is a VLSI device which provides a complete link level data communication control conforming to the 1984 CCITT version of X.25. The MK5025 also supports X.32 (XID) and X.75 as well as single channel LAPD for ISDN with its UI frames and extended addressing capabilities.

PURPOSE

Although the MK5025 Data Sheet and Technical Manual provide detailed asynchronous timing diagrams that specify the relationships of the host interface signals to one another, the designer may often find it helpful to know how these timing values relate to the system clock. The purpose of this application brief is to provide a description of the MK5025 host interface as related to SYSCLK (MK5025 pin 28).

Because of the asynchronous nature of the use of this device, the MK5025 production testing is performed to ensure compliance with the asynchronous timing specifications stated in the Data Sheet and Technical Manual. It should be noted that although the synchronous timing diagrams in this document are provided to facilitate the design process, **the timing requirements in the Data Sheet must still be met to ensure proper operation.**

Figure 1 : MK5025 Bus Slave Write Cycle.



Notes : Input setup and hold times are in italics. These times are the typical minimum values required to or from the specified edge to be recognized within that clock cycle.
Output delay times are the typical maximum from the specified edge to a valid output.

SYNCHRONOUS TIMING

The synchronous timing data contained within this document was derived from a sample of MK5025 devices and guard-banded to allow for process variations. Although these values are not guaranteed or tested in the manufacturing process, the typical MK5025 device performance should meet or exceed these timing values.

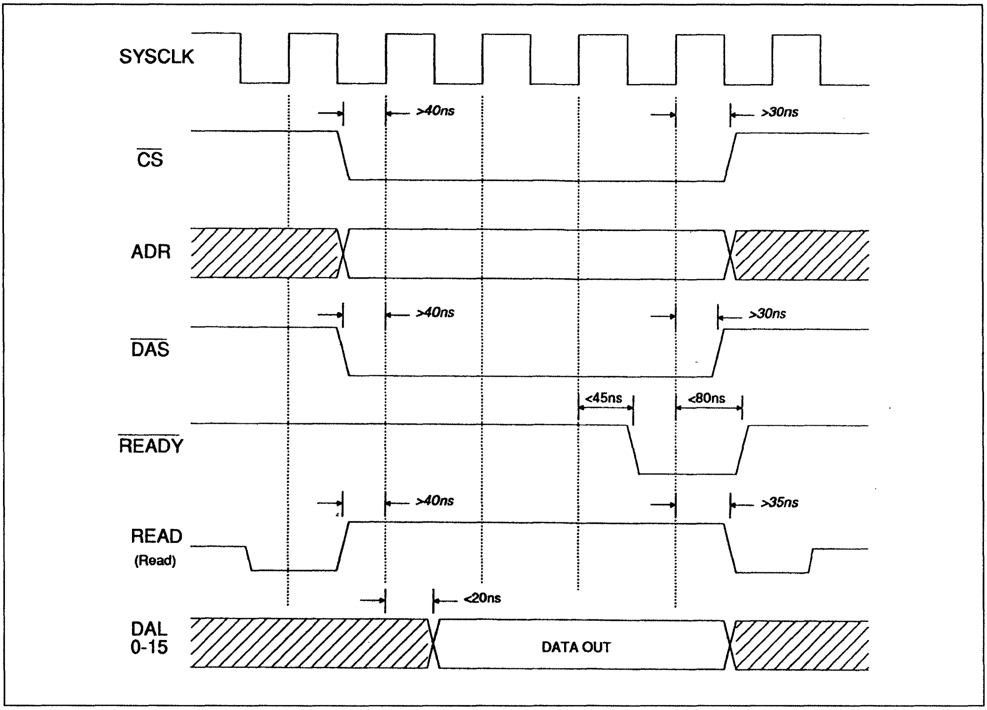
Figures 1 and 2 provide the typical timing relationships, with respect to SYSCLK, for the MK5025 operating as a bus slave. Figures 3 and 4 provide the typical timing relationships, with respect to SYSCLK, for the MK5025 operating as a bus master.

The input setup and hold times given in this document are typical minimum values required to or from the SYSCLK edge indicated in order to be recognized within that SYSCLK cycle. The output delay times are the typical maximum delay from the indicated edge to a valid output state.

CONCLUSION

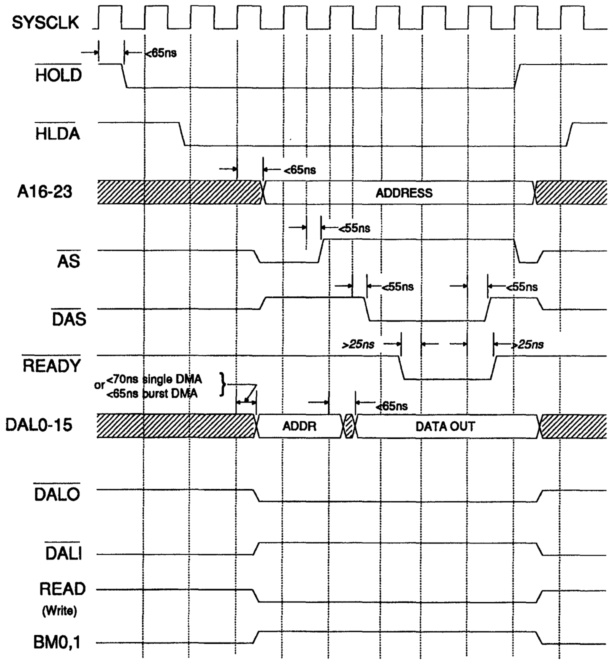
The MK5025 offers great flexibility to the data communications system designer. The on-chip protocol processing may be used to save the designer much time in implementing standard protocols such as X.25, LAPB, ISDN LAPD, X.32, and X.75, and these synchronous timing diagrams are provided to further facilitate the design process.

Figure 2 : MK5025 Bus Slave Read Cycle.



Notes : Input setup and hold times are in italics. These times are the typical minimum values required to or from the specified edge to be recognized within that clock cycle.
Output delay times are the typical maximum from the specified edge to a valid output.

Figure 3 : MK5025 Bus Master Write Cycle.

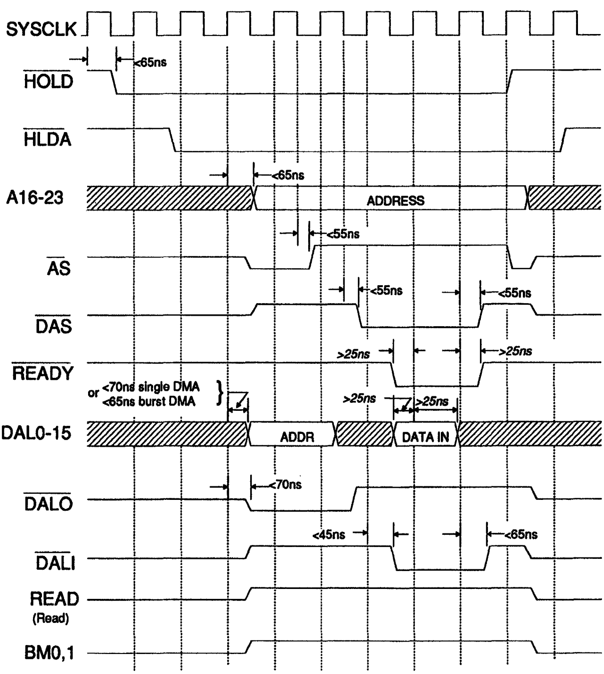


Notes : The typical MK5025 device performance should meet or exceed the typical timing values given above. Asynchronous timing requirements given in the Technical Manual still apply.

Input setup and hold times are in italics. These times are typical minimum values required to or from the particular edge specified in order to be recognized in that cycle.

Output delay times are the typical maximum delay from the specified edge to a valid output.

Figure 4 : MK5025 Bus Master Read Cycle.



Notes : The typical MK5025 device performance should meet or exceed the typical timing values given above. Asynchronous timing requirements given in the Technical Manual still apply.

Input setup and hold times are in italics. These times are typical minimum values required to or from the particular edge specified in order to be recognized in that cycle.

Output delay times are the typical maximum delay from the specified edge to a valid output.

CCS# 7 DEVICES

SGS-THOMSON
MICROELECTRONICS

MK5027

SS7 SIGNALLING LINK CONTROLLER

- CMOS
- FULLY COMPATIBLE WITH BOTH 8 OR 16 BIT SYSTEMS
- SYSTEM CLOCK RATE TO 10MHz
- DATA RATE UP TO 2.5Mbps FOR SS7 PROTOCOL PROCESSING , 7Mbps FOR TRANSPARENT HDLC MODE
- COMPLETE LEVEL 2 IMPLEMENTATION
- COMPATIBLE WITH 1988 CCITT, AT&T, ANSI, AND BELLCORE SIGNALLING SYSTEM NUMBER 7 LINK LEVEL PROTOCOLS
- 52 PIN PLCC AND 48-PIN DIP PIN-FOR-PIN COMPATIBLE WITH THE SGS-THOMSON X.25 CHIP (MK5025) AND NEARLY PIN-FOR-PIN COMPATIBLE WITH THE SGS-THOMSON VLANCE CHIP (MK5032)
- BUFFER MANAGEMENT INCLUDES :
  - Initialization Block
  - Separate Receive and Transmit Rings
  - Variable Descriptor Ring and Window Sizes.
- ON CHIP DMA CONTROL WITH PROGRAMMABLE BURST LENGTH
- SELECTABLE BEC OR PCR RETRANSMISSION METHODS, INCLUDING FORCED RETRANSMISSION FOR PCR
- HANDLES ALL 7 SS7 TIMERS
- HANDLES ALL SS7 FRAME FORMATTING :
  - Zero bit insert and delete
  - FCS generation and detection
  - Frame delimiting with flags
- PROGRAMMABLE MINIMUM SIGNAL UNIT SPACING (number of flags between SU's)
- HANDLES ALL SEQUENCING AND LINK CONTROL.
- SELECTABLE FCS OF 16 OR 32 BITS
- TESTING FACILITIES :
  - Internal Loopback
  - Silent Loopback
  - Optional Internal Data Clock Generation
  - Self Test.
- ALL INPUTS AND OUTPUTS ARE TTL COMPATIBLE
- PROGRAMMABLE FOR FULL OR HALF DUPLEX OPERATION

DESCRIPTION

The SGS-Thomson Signalling System #7 Signalling Link Controller (MK5027) is a VLSI semiconductor device which provides a complete link control function conforming to the 1988 CCITT version of SS7. This includes frame formatting, transparency (so-called "bit-stuffing"), error recovery by two types of retransmission, error monitoring, sequence number control, link status control, and FISU generation. One of the outstanding features of the MK5027 is its buffer management which includes on-chip DMA. This feature allows users to handle multiple packets of receive and transmit data at a time. (A conventional data link control chip plus a separate DMA chip would handle data for only a single block at a time.) The MK5027 may be used with any of several popular 16 and 8 bit microprocessors, such as 68000, 6800, Z8000, Z80, LSI-11, 8086, 8088, 8080, etc.

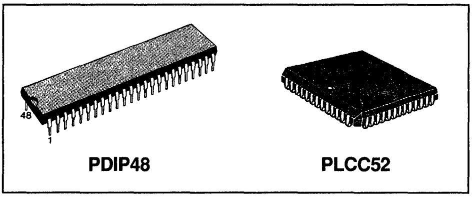


Figure 1 : Pin Connection.

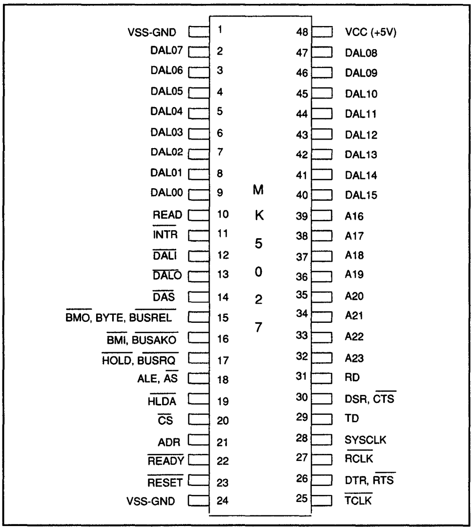


Table 1 : Pin Description.

LEGEND :

I	Input only	O	Output only
IO	Input/Output	3S	3-State
OD	Open Drain (no internal pull-up)		

Signal Name	Pin(s)	Type	Description
DAL<15:00>	2-9 40-47	IO/3S	The time multiplexed Data/Address bus. During the address portion of a memory transfer, DAL<15:00> contains the lower 16 bits of the memory address. During the data portion of a memory transfer, DAL<15:00> contains the read or write data, depending on the type of transfer.
READ	10	IO/3S	READ indicates the type of operation that the bus controller is performing during a bus transaction. READ is driven by the MK5027 only while it is the BUS MASTER. READ is valid during the entire bus transaction and is tristated at all other times. MK5027 as a Bus Slave : READ = HIGH - Data is placed on the DAL lines by the chip. READ = LOW - Data is taken off the DAL lines by the chip. MK5027 as a Bus Master : READ = HIGH - Data is taken off the DAL lines by the chip. READ = LOW - Data is placed on the DAL lines by the chip.
$\overline{\text{INTR}}$	11	O/OD	INTERRUPT is an attention interrupt line that indicates that one or more of the following CSR0 status flags is set : MISS, MERR, RINT, TINT or PINT. INTERRUPT is enabled by CSR0<09>, INEA = 1.
$\overline{\text{DALI}}$	12	O/3S	DAL IN is an external bus transceiver control line. $\overline{\text{DALI}}$ is driven by the MK5027 only while it is the BUS MASTER. $\overline{\text{DALI}}$ is asserted by the MK5027 when it reads from the DAL lines during the data portion of a READ transfer. $\overline{\text{DALI}}$ is not asserted during a WRITE transfer.
$\overline{\text{DALO}}$	13	O/3S	DAL OUT is an external bus transceiver control line. $\overline{\text{DALO}}$ is driven by the MK5027 only while it is the BUS MASTER. $\overline{\text{DALO}}$ is asserted by the MK5027 when it drives the DAL lines during the address portion of a READ transfer or for the duration of a WRITE transfer.
$\overline{\text{DAS}}$	14	IO/3S	DATA STROBE defines the data portion of a transaction. By definition, data is stable and valid at the low to high transition of $\overline{\text{DAS}}$. This signal is driven by the MK5027 while it is the BUS MASTER. During the BUS SLAVE operation, this pin is used as an input. At all other times the signal is tristated.
$\overline{\text{BMO}}$ $\overline{\text{BYTE}}$ $\overline{\text{BUSREL}}$	15	IO/3S	I/O pins 15 and 16 are programmable through CSR4. If bit 06 of CSR4 is set to a one, pin 15 becomes input $\overline{\text{BUSREL}}$ and is used by the host to signal the MK5027 to terminate a DMA burst after the current bus transfer has completed. If bit 06 is clear the pin 15 is an output and behaves as described below for pin 16.

Note : Pin out shown is for 48 pin dip.

Table 1 : Pin Description (continued).

<table>
<tr><th>Signal Name</th><th>Pin(s)</th><th>Type</th><th>Description</th></tr>
<tr><td>$\overline{\text{BM1}}$
$\overline{\text{BUSAKO}}$</td><td>16</td><td>O/3S</td><td>Pins 15 and 16 are programmable though bit 00 of CSR4 (BCON).
If CSR4<00> BCON = 0,
I/O PIN 15 = BMO (O/3S)
I/O PIN 16 = BM1 (O/3S)
BYTE MASK<1:0> indicates the byte(s) on the DAL to be read or written during this bus transaction. MK5027 drives these lines only as a Bus Master. MK5027 ignores the BM lines when it is a Bus Slave.
Byte selection is done as outlined in the following table.
<table>
<tr><th>$\overline{\text{BM1}}$</th><th>$\overline{\text{BM0}}$</th><th>TYPE OF TRANSFER</th></tr>
<tr><td>LOW</td><td>LOW</td><td>ENTIRE WORD</td></tr>
<tr><td>LOW</td><td>HIGH</td><td>UPPER BYTE
(DAL<15:08>)</td></tr>
<tr><td>HIGH</td><td>LOW</td><td>LOWER BYTE
(DAL<07:00>)</td></tr>
<tr><td>HIGH</td><td>HIGH</td><td>NONE</td></tr>
</table>
If CSR4<00> BCON = 1,
I/O PIN 15 = $\overline{\text{BYTE}}$ (O/3S)
I/O PIN 16 = $\overline{\text{BUSAKO}}$ (O)
Byte selection is done using the BYTE line and DAL<00> latched during the address portion of the bus transaction. MK5027 drives BYTE only as a Bus Master and ignores it when a Bus Slave. Byte selection is done as outlined in the following table.
<table>
<tr><th>BYTE</th><th>DAL<00></th><th>TYPE OF TRANSFER</th></tr>
<tr><td>LOW</td><td>LOW</td><td>ENTIRE WORD</td></tr>
<tr><td>LOW</td><td>HIGH</td><td>ILLEGAL CONDITION</td></tr>
<tr><td>HIGH</td><td>LOW</td><td>LOWER BYTE</td></tr>
<tr><td>HIGH</td><td>HIGH</td><td>UPPER BYTE</td></tr>
</table>
$\overline{\text{BUSAKO}}$ is a bus request daisy chain output. If MK5027 is not requesting the bus and it receives $\overline{\text{HLDA}}$, $\overline{\text{BUSAKO}}$ will be driven low. If MK5027 is requesting the bus when it receives $\overline{\text{HLDA}}$, $\overline{\text{BUSAKO}}$ will remain high.
Note : All transfers are entire word unless the MK5027 is configured for 8 bit operation.</td></tr>
</table>

Table 1 : Pin Description (continued).

Signal Name	Pin(s)	Type	Description
$\overline{\text{HOLD}}$ $\overline{\text{BUSRQ}}$	17	IO/OD	Pin 17 is configured through bit 0 of CSR4. If CSR4<00> BCON = 0, I/O PIN 17 = $\overline{\text{HOLD}}$ $\overline{\text{HOLD}}$ request is asserted by MK5027 when it requires a DMA cycle, if $\overline{\text{HLDA}}$ is inactive, regardless of the previous state of the $\overline{\text{HOLD}}$ pin. $\overline{\text{HOLD}}$ is held low for the entire ensuing bus transaction. If CSR4<00> BCON = 1, I/O PIN 17 = $\overline{\text{BUSRQ}}$ $\overline{\text{BUSRQ}}$ is asserted by MK5027 when it requires a DMA cycle if the prior state of the $\overline{\text{BUSRQ}}$ pin was high and $\overline{\text{HLDA}}$ is inactive. $\overline{\text{BUSRQ}}$ is held low for the entire ensuing bus transaction.
ALE $\overline{\text{AS}}$	18	O/3S	The active level of ADDRESS STROBE is programmable through CSR4. The address portion of a bus transfer occurs while this signal is at its asserted level. This signal is driven by MK5027 while it is the BUS MASTER. At all other times, the signal is tristated. If CSR4<01> ACON = 0, I/O PIN 18 = ALE ADDRESS LATCH ENABLE is used to demultiplex the DAL lines and define the address portion of the transfer. As ALE, the signal transitions from high to low during the address portion of the transfer and remains low during the data portion. If CSR4<01> ACON = 1, I/O PIN 18 = $\overline{\text{AS}}$ As $\overline{\text{AS}}$, the signal pulses low during the address portion of the bus transfer. The low to high transition of $\overline{\text{AS}}$ can be used by a slave device to strobe the address into a register. $\overline{\text{AS}}$ is effectively the inversion of ALE.
$\overline{\text{HLDA}}$	19	I	HOLD ACKNOWLEDGE is the response to $\overline{\text{HOLD}}$. When $\overline{\text{HLDA}}$ is low in response to MK5027's assertion of $\overline{\text{HOLD}}$, the MK5027 is the Bus Master. $\overline{\text{HLDA}}$ should be desasserted ONLY after $\overline{\text{HOLD}}$ has been released by the MK5027.

Table 1 : Pin Description (continued).

Signal	Pin(s)	Type	Description
$\overline{\text{CS}}$	20	I	CHIP SELECT indicates, when low, that the MK5027 is the slave device for the data transfer. $\overline{\text{CS}}$ must be valid throughout the entire transaction.
ADR	21	I	ADDRESS selects the Register Address Port or the Register Data Port. It must be valid throughout the data portion of the transfer and is only used by the chip when $\overline{\text{CS}}$ is low. ADR PORT LOW REGISTER DATA PORT HIGH REGISTER ADDRESS PORT
$\overline{\text{READY}}$	22	IO/OD	When the MK5027 is a Bus Master, $\overline{\text{READY}}$ is an asynchronous acknowledgement from the bus memory that memory will accept data in a WRITE cycle or that memory has put data on the DAL lines in a READ cycle. As a bus Slave, the MK5027 asserts $\overline{\text{READY}}$ when it has put data on the DAL lines during a READ cycle or is about to take data from the DAL lines during a WRITE cycle. $\overline{\text{READY}}$ is a response to $\overline{\text{DAS}}$ and it will be released after $\overline{\text{DAS}}$ or $\overline{\text{CS}}$ is negated.
$\overline{\text{RESET}}$	23	I	$\overline{\text{RESET}}$ is the Bus signal that will cause MK5027 to cease operation, clear its internal logic and enter an idle state with the Power Off bit of CSR0 set.
$\overline{\text{TCLK}}$	25	I	TRANSMIT CLOCK. A 1x clock input for transmitter timing. TD changes on the falling edge of $\overline{\text{TCLK}}$. The frequency of $\overline{\text{TCLK}}$ may not be greater than the frequency of SYSCLK.
DTR $\overline{\text{RTS}}$	26	IO	DATA TERMINAL READY, REQUEST TO SEND. Modem control pin. Pin 26 is configurable through CSR5. This pin can be programmed to behave as output $\overline{\text{RTS}}$ or as programmable IO pin DTR. If configured as $\overline{\text{RTS}}$, the MK5027 will assert this pin if it has data to send and throughout the transmission of a signal unit.
$\overline{\text{RCLK}}$	27	I	RECEIVE CLOCK. A 1x clock input for receiver timing. RD is sampled on the rising edge of $\overline{\text{RCLK}}$. The frequency of $\overline{\text{RCLK}}$ may not be greater than the frequency of SYSCLK.
SYSCLK	28	I	SYSTEM CLOCK. System clock used for internal timing of the MK5027. SYSCLK should be a square wave, of frequency up to 10 MHz.
TD	29	O	TRANSMIT DATA. Transmit serial data output.
DSR $\overline{\text{CTS}}$	30	IO	DATA SET READY, CLEAR TO SEND. Modem Control Pin. Pin 30 is configurable through CSR5. This pin can be programmed to behave as input $\overline{\text{CTS}}$ or as programmable IO pin DSR. If configured as $\overline{\text{CTS}}$, the MK5027 will transmit all ones while $\overline{\text{CTS}}$ is high.
RD	31	I	RECEIVE DATA. Received serial data input.
A <23:16>	32-39	O/3S	Address bits <23:16> used in conjunction with DAL <15:00> to produce a 24 bit address. MK5027 drives these lines only as a Bus Master. A23-A20 may be driven continuously as described in the CSR4 <7> BAEN bit.
VSS-GND	1, 24		Ground Pins
VCC	48		Power Supply Pin + 5.0 VDC ± 5%

Figure 2 : Possible System Configuration for the MK5027.

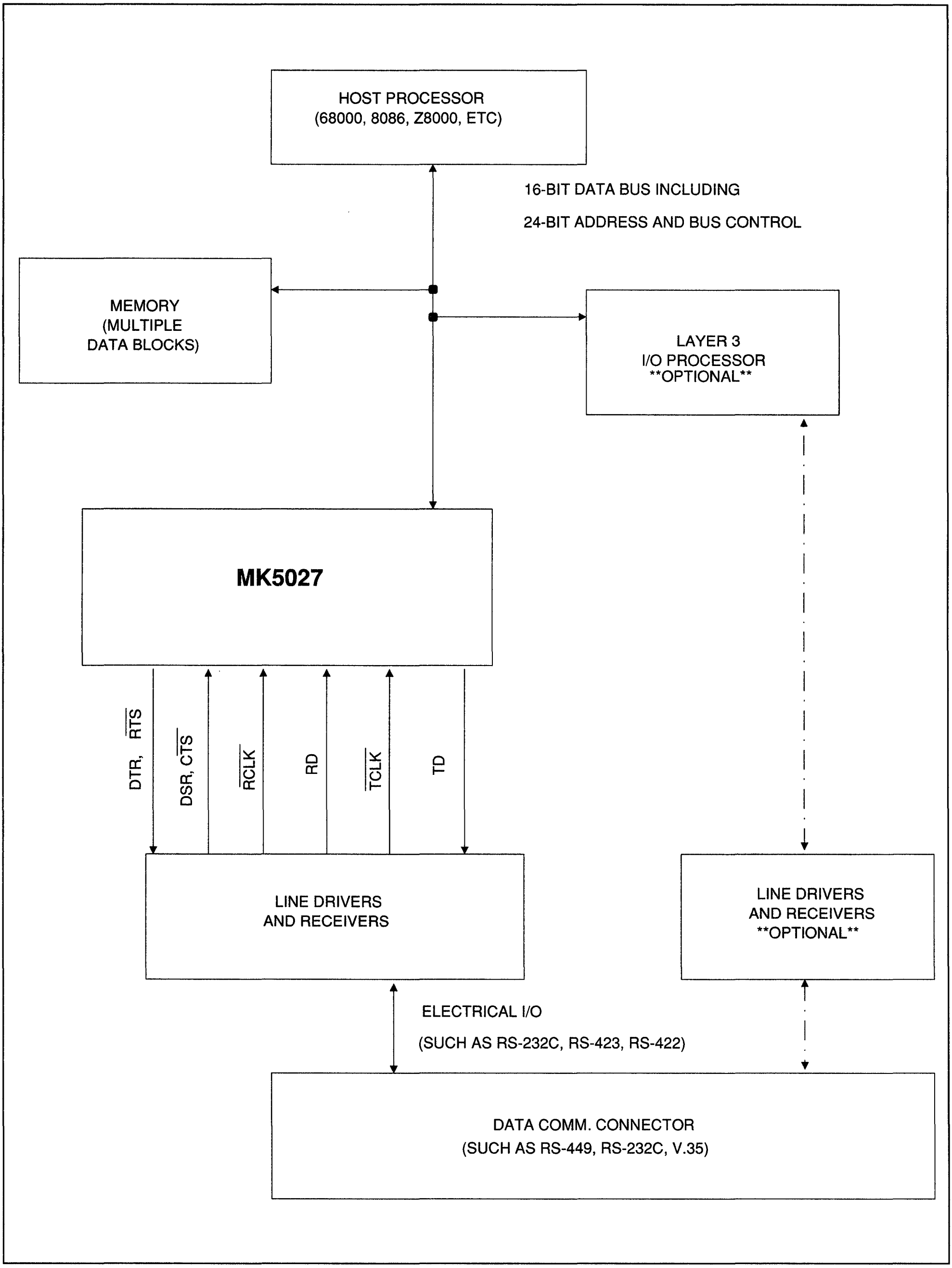
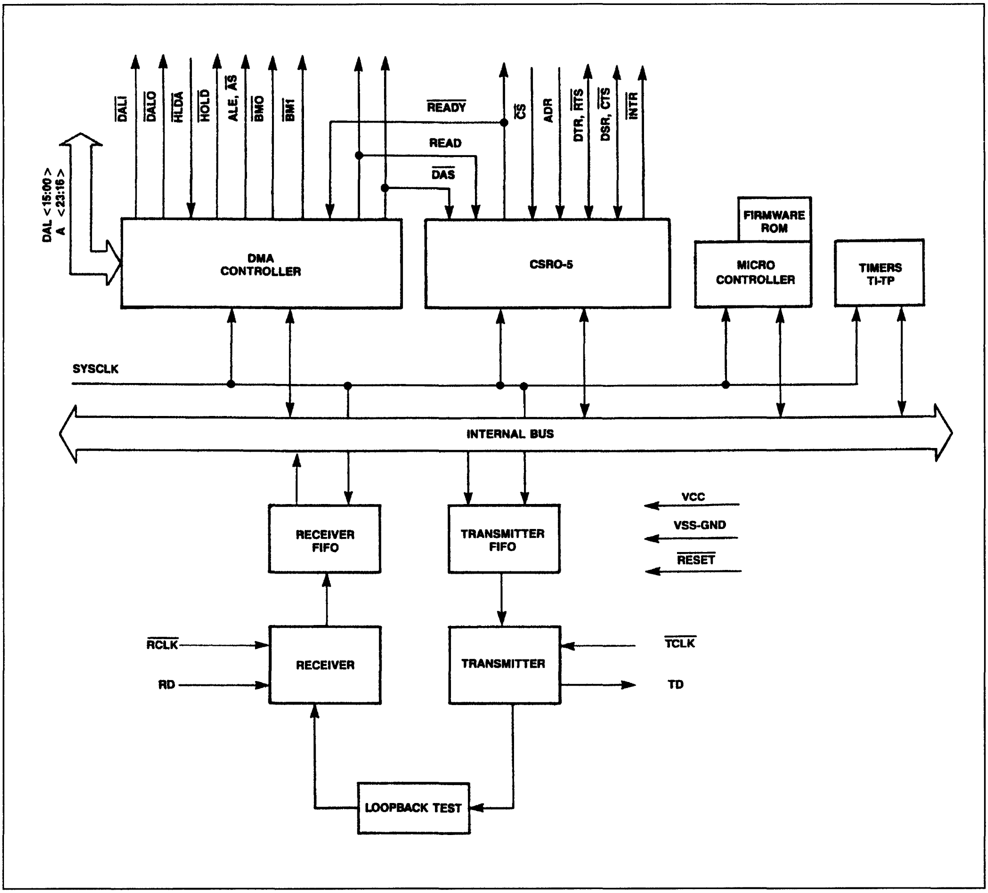


Figure 3 : MK5027 Simplified Block Diagram.



OPERATIONAL DESCRIPTION

The SGS-Thomson Signalling System #7 Signalling Link Controller (MK5027) device is a VLSI product intended for data communication applications requiring SS7 link level control. The MK5027 will perform all frame formatting, such as : frame delimiting with flags, FCS generation and detection. It will also perform all error recovery and link control. The MK5027 also includes a buffer management mechanism that allows the user to transmit and/or receive mutilple MSU's. Contained in the buffer management is an on-chip dual channel DMA : one channel for receive and one channel for transmit. The MK5027 handles error recovery and link status signalling.

The MK5027 is intended to be used with any popular 16 or 8 bit microprocessor. Possible system configuration for the MK5027 is shown in Figure 2. The MK5027 will move multiple blocks of receive and transmit data directly into and out of memory through the host's bus. An I/O acceleration processor could be used to off-load Higher Level software from the Host. The I/O acceleration processor in Figure 2 is recommended, but not required.

All signal pins on the MK5027 are TTL compatible. This has the advantage of making the MK5027 independent of the physical interface. As shown in Figure 2, line drivers and receivers are used for electrical connection to the physical layer.

SERIAL INTERFACE

The MK5027 provides two seperate serial channels; one for received data and one for transmitted data. These serial channels are completely seperate and may be run at different clock frequencies. The receiver is responsible for recognizing frame boundries, removal of inserted zeroes (for transparency), and checking the incoming FCS. Signal units with incorrect FCS values are discarded. The receiver also parallelizes the incoming data which is placed into the receive data buffers within the receive descriptor ring. The transmitter is responsible for framing and serializing the data frames placed in the transmit descriptor ring. The transmitter calculates the FCS of the outgoing data and appends it to the data. The transmitter generates flag sequences for inter-signal unit fill, at least two flags are transmitted between adjacent signal units. The FCS calculations for both directions of serial data optionally follow either the 16-bit CRC-CCITT or the 32-bit CRC-32 algorithms. FCS generation and checking can also be optionally disabled if necessary.

MICROPROCESSOR INTERFACE

The MK5027 contains a dual channel DMA on chip to handle data transfers to and from the host memory. All access to the initialization block and descriptor rings is handled in this way. The address bus is 24 bits wide and does not use any segmentation or paging methods. Data transfers can optionally be 8 and 16 bit operations, this allows easy interfacing with both 8 and 16 bit processors. DMA transfers can be up to 1, 8 or an unlimited number of words per transfer under program control. During bus slave operation the MK5027 allows access to its 6 control/status registers which are used to monitor and control the chip. These registers are used to control link procedures, configure interface options, control and monitor interrupt status, and more. Bus slave mode also allows both 8 and 16 bit accesses.

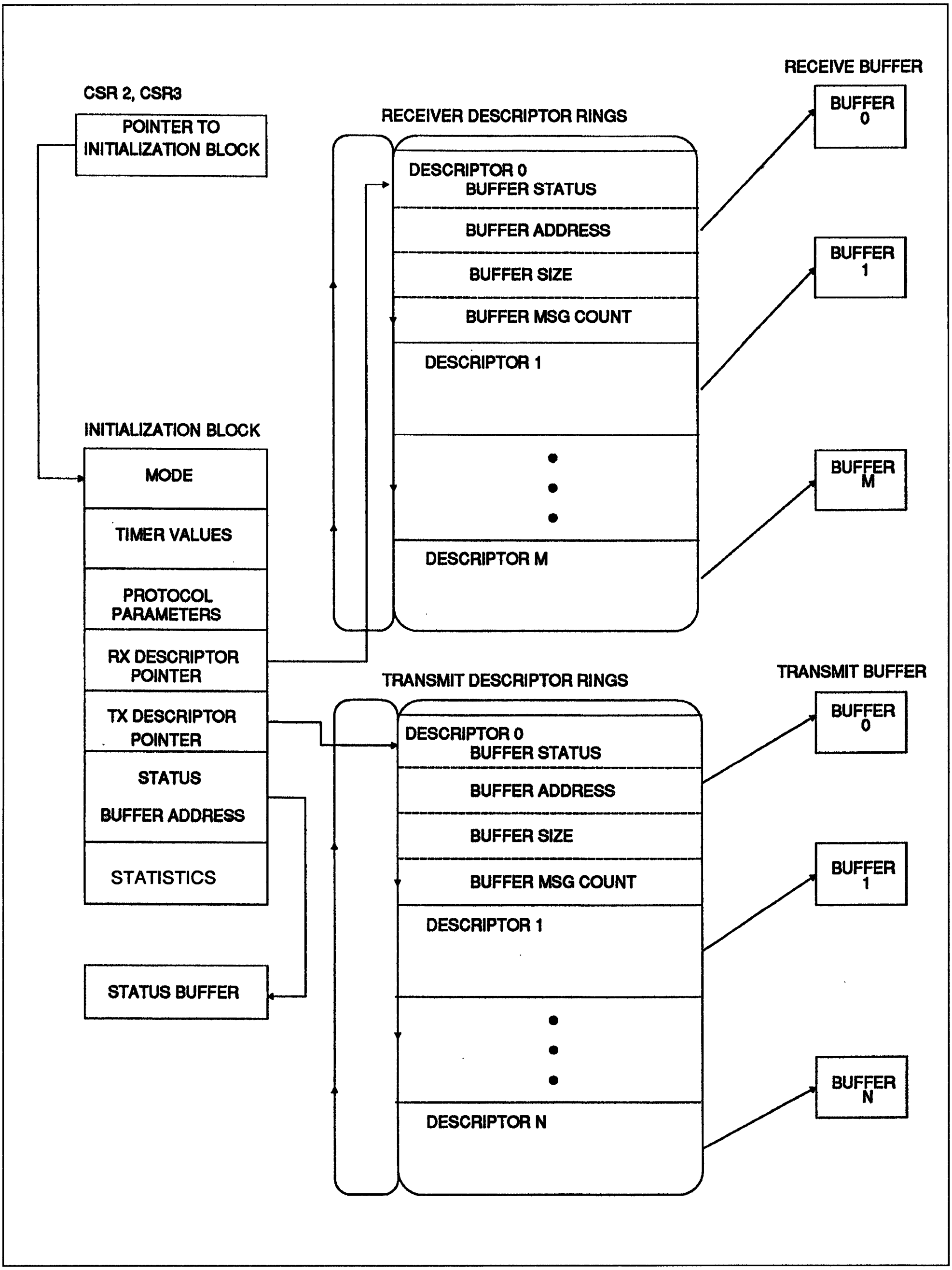
BUFFER MANAGEMENT

The basic organization of the buffer management is a circular queue of tasks in memory called descriptor rings. There are seperate rings to describe the transmit and receive operations. Up to 128 buffers may be queued-up on a descriptor ring awaiting execution by the MK5027. The descriptor ring has a segment assigned to each buffer. Each segment holds a pointer for the starting address of the buffer, and holds a value for the length of the buffer in bytes.

Each segment also contains two control bits called OWNA and OWNB, which denote whether the MK5027, the HOST, or the I/O ACCELERATION PROCESSOR (if present) "owns" the buffer. For transmit, when the MK5027 owns the buffer, the MK5027 is allowed and commanded to transmit the buffer. When the MK5027 does not own the buffer, it will not transmit that buffer. For receive, when the MK5027 owns a buffer, it may place received data into that buffer. Conversely, when the MK5027 does not own a receive buffer, it will not place received data in that buffer.

The MK5027 buffer management mechanism will handle signal units which are longer than the length of an individual buffer. This is done by a chaining method which utilizes multiple buffers.The MK5027 tests the next segment in the descriptor ring in a "look ahead" manner.If the packet is too long for one buffer, the next buffer will be used after filling the first buffer ; that is, "chained". The MK5027 will then "look ahead" to the next buffer, and chain that buffer if necessary, and so on. The operational parameters for the buffer management are defined by the user in the initialization block. The parameters defined include the basic mode of operation, the number of entries for the transmitter and receiver descriptor rings, etc.

Figure 4 : MK5027 Buffer Management.



SIGNAL UNIT REPERTOIRE

The signal unit repertoire of the MK5027 is shown in Table 1. This set conforms to the 1988 CCITT specification for level 2 of Signalling System #7.

The definitions for the symbols for the frame types are :

Name	Definition
F	Flag Sequence
FSN	Forward Sequence Number
BSN	Backward Sequence Number
FIB	Forward Indicator Bit
BIB	Backward Indicator Bit
LI	Length Indicator
X	Programmed As Zeroes
SIO	Signalling Information Octet
SIF	Service Information Field
SF	Status Field
FCS	Frame Check Sequence

Table 1 : MK5027 Signal Unit Repertoire.

Message Signal Unit (MSU)

F	FCS	SIF	SIO	X	LI	FIB	FSN	BIB	BSN	F
8	16/32	8n,n>=2	8	2	6	1	7	1	7	8

Link Status Signal Unit (LSSU)

F	FCS	SF	X	LI	FIB	FSN	BIB	BSN	F
8	16/32	8/16	2	6	1	7	1	7	8

Values for SF:

0 - SIO,	Out of alignment	3 - SIOS,	Out-of-service
1 - SIN,	Normal alignment	4 - SIPO,	Processor outage
2 - SIE,	Emergency	5 - SIB,	Congestion (Busy)

Fill-in Signal Unit (FISU)

F	FCS	X	LI	FIB	FSN	BIB	BSN	F
8	16/32	2	6	1	7	1	7	8

Right-most fields are transmitted first

MK5027 ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Temperature under Bias	– 25°C to + 100°C
Storage Temperature	– 65°C to + 150°C
Voltage on Any Pin with Respect to Ground	– 0.5V to VCC + 0.5V
Power Dissipation	0.50W

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the above device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect

DC CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = + 5V ± 5 percent unless otherwise specified.

Symbol	Conditions	Min.	Typ.	Max.	Unit
V_{IL}		– 0.5		+ 0.8	V
V_{IH}		+ 2.0		V_{CC} +0.5	V
V_{OL}	@ I_{OL} = 3.2mA			+ 0.5	V
V_{OH}	@ I_{OH} = – 0.4mA	+ 2.4			V
I_{IL}	@ V_{IN} = 0.4 to V_{CC}			± 10	µA
I_{CC}	@ T_{SCT} = 100ns		50		mA

CAPACITANCE

F = 1MHz

Symbol	Parameter	Min.	Max.	Unit
C_{IN}	Capacitance on Input Pins		10	pf
C_{OUT}	Capacitance on Output Pins		10	pf
C_{IO}	Capacitance on I/O Pins		20	pf

AC TIMING SPECIFICATIONS

T_A = 0°C to 70°C, V_{CC} = + 5V ±5 percent, unless otherwise specified.

N°	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
1	SYSCLK	T_{SCT}	SYSCLK Period		100		2000	ns
2	SYSCLK	T_{SCL}	SYSCLK Low Time		45			ns
3	SYSCLK	T_{SCH}	SYSCLK High Time		45			ns
4	SYSCLK	T_{SCR}	Rise Time of SYSCLK		0		8	ns
5	SYSCLK	T_{SCF}	Fall Time of SYSCLK		0		8	ns
6	$\overline{\text{TCLK}}$	T_{TCT}	$\overline{\text{TCLK}}$ Period		140			ns
7	$\overline{\text{TCLK}}$	T_{TCL}	$\overline{\text{TCLK}}$ Low Time		63			ns
8	TCLK	T_{TCH}	$\overline{\text{TCLK}}$ High Time		63			ns
9	$\overline{\text{TCLK}}$	T_{TCR}	Rise Time of $\overline{\text{TCLK}}$	CL = 50pF	0		8	ns
10	$\overline{\text{TCLK}}$	T_{TCF}	Fall Time of $\overline{\text{TCLK}}$		0		8	ns
11	TD	T_{TDP}	TD Data Propagation Delay after the Falling Edge of $\overline{\text{TCLK}}$	CL = 50pF			40	ns
12	TD	T_{TDH}	TD Data Hold Time after the Falling Edge of $\overline{\text{TCLK}}$		5			ns
13	$\overline{\text{RCLK}}$	T_{RCT}	$\overline{\text{RCLK}}$ Period		140			ns

AC TIMING SPECIFICATIONS (continued)

T_A = 0°C to 70°C, V_{CC} = + 5V ±5 percent, unless otherwise specified.

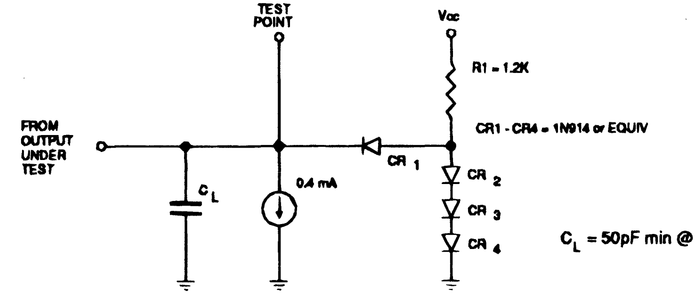
N°	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
14	$\overline{\text{RCLK}}$	T_{RCH}	$\overline{\text{RCLK}}$ High Time		63			ns
15	$\overline{\text{RCLK}}$	T_{RCL}	$\overline{\text{RCLK}}$ Low Time		63			ns
16	$\overline{\text{RCLK}}$	T_{RCR}	Rise Time of $\overline{\text{RCLK}}$		0		8	ns
17	$\overline{\text{RCLK}}$	T_{RCF}	Fall Time of $\overline{\text{RCLK}}$		0		8	ns
18	RD	T_{RDR}	RD Data Rise Time		0		8	ns
19	RD	T_{RDF}	RD Data Fall Time		0		8	ns
20	RD	T_{RDH}	RD Hold Time after Rising Edge of $\overline{\text{RCLK}}$		5			ns
21	RD	T_{RDS}	RD Setup Time Prior to Rising Edge of $\overline{\text{RCLK}}$		30			ns
22	A/DAL	T_{DOFF}	Bus Master Driver Disable after Rising Edge of $\overline{\text{HOLD}}$		0		50	ns
23	A/DAL	T_{DON}	Bus Master Driver Enable after Falling Edge of $\overline{\text{HLDA}}$	TSCT=100nS	0		200	ns
24	$\overline{\text{HLDA}}$	T_{HHA}	Delay to Falling Edge of $\overline{\text{HLDA}}$ from Falling Edge of $\overline{\text{HOLD}}$ (bus master)		0			ns
25	$\overline{\text{RESET}}$	T_{RW}	$\overline{\text{RESET}}$ Pulse Width		30			ns
26	A/DAL	T_{CYCLE}	Read/write, address/data Cycle Time	TSCT=100nS	600			ns
27	A	T_{XAS}	Address Setup Time to Falling Edge of ALE		100			ns
28	A	T_{XAH}	Address Hold Time after the Rising Edge of $\overline{\text{DAS}}$		50			ns
29	DAL	T_{AS}	Address Setup Time to the Falling Edge of ALE		75			ns
30	DAL	T_{AH}	Address Hold Time after the Falling Edge of ALE		20			ns
31	DAL	T_{RDAS}	Data Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus master read)		55			ns
32	DAL	T_{RDAH}	Data Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus master read)		0			ns
33	DAL	T_{DDAS}	Data Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus master write)		0			ns
34	DAL	T_{WDS}	Data Setup Time to the Rising Edge of $\overline{\text{DAS}}$ (bus master write)		250			ns
35	DAL	T$\overline{\text{WDH}}$	Data Hold Time to the Rising Edge of $\overline{\text{DAS}}$ (bus master write)		35			ns
36	DAL	T_{SRDH}	Data Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave read)	TSCT=100nS	0		35	ns
37	DAL	T_{SWDH}	Data Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave write)		0			ns
38	DAL	T_{SWDS}	Data Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus slave write)		0			ns
39	ALE	T_{ALEW}	ALE width High		110			ns
40	ALE	T_{DALE}	Delay from Rising Edge of $\overline{\text{DAS}}$ to the Rising Edge of ALE		70			ns
41	$\overline{\text{DAS}}$	T_{DSW}	$\overline{\text{DAS}}$ width Low		200			ns

AC TIMING SPECIFICATIONS (continued)

T_A = 0°C to 70°C, V_{CC} = + 5V ±5 percent, unless otherwise specified.

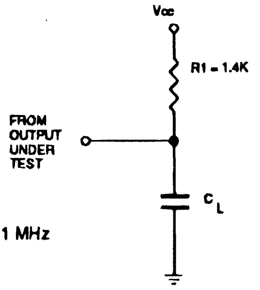
N°	Signal	Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
42	$\overline{\text{DAS}}$	T_{ADAS}	Delay from the Falling Edge of Ale to the Falling Edge of $\overline{\text{DAS}}$		80			ns
43	$\overline{\text{DAS}}$	T_{RIDF}	Delay from the Rising Edge of $\overline{\text{DALO}}$ to the Falling Edge of $\overline{\text{DAS}}$ (bus master read)		35			ns
44	$\overline{\text{DAS}}$	T_{RDYS}	Delay from the Falling Edge of $\overline{\text{READY}}$ to the Falling Edge of $\overline{\text{DAS}}$	T_{ARYD}=300ns T_{SCT}=100ns	120		200	ns
45	$\overline{\text{DALI}}$	T_{ROIF}	Delay from the Rising Edge of $\overline{\text{DALO}}$ to the Falling Edge of $\overline{\text{DALI}}$ (bus master read)		70			ns
46	$\overline{\text{DALI}}$	T_{RIS}	$\overline{\text{DALI}}$ Setup Time to the Rising Edge of $\overline{\text{DAS}}$ (bus master read)		150			ns
47	$\overline{\text{DALI}}$	T_{RIH}	$\overline{\text{DALI}}$ Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus master read)		0			ns
48	$\overline{\text{DALI}}$	T_{RIOF}	Delay from the Rising Edge of $\overline{\text{DALI}}$ to the Falling Edge of $\overline{\text{DALO}}$ (bus master read)		70			ns
49	$\overline{\text{DALO}}$	T_{OS}	$\overline{\text{DALO}}$ Setup Time to the Falling Edge of ALE (bus master read)		110			ns
50	$\overline{\text{DALO}}$	T_{ROH}	$\overline{\text{DALO}}$ Hold Time after the Falling Edge of ALE (bus master read)		35			ns
51	$\overline{\text{DALO}}$	T_{WDSI}	Delay from the Rising Edge of $\overline{\text{DAS}}$ to the Rising Edge of $\overline{\text{DALO}}$ (bus master write)		50			ns
52	$\overline{\text{CS}}$	T_{CSH}	$\overline{\text{CS}}$ Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave)		0			ns
53	$\overline{\text{CS}}$	T_{CSS}	$\overline{\text{CS}}$ Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus slave)		0			ns
54	$\overline{\text{ADR}}$	T_{SAH}	$\overline{\text{ADR}}$ Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave)		0			ns
55	$\overline{\text{ADR}}$	T_{SAS}	ADR Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus slave)		0			ns
56	$\overline{\text{READY}}$	T_{ARYD}	Delay from the Falling Edge of ALE to the Falling Edge of $\overline{\text{READY}}$ to Insure a Minimum Bus Cycle Time (600nS)	T_{SCT}=100nS			150	ns
57	$\overline{\text{READY}}$	T_{SRDS}	Data Setup Time to the Falling Edge of $\overline{\text{READY}}$ (bus slave read)		75			ns
58	$\overline{\text{READY}}$	T_{RDYH}	$\overline{\text{READY}}$ Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus master)		0			ns
59	$\overline{\text{READY}}$	T_{SRYH}	$\overline{\text{READY}}$ Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave)	T_{SCT}=100nS	0		35	ns
60	$\overline{\text{READY}}$	T_{RSH}	READ Hold Time after the Rising Edge of $\overline{\text{DAS}}$ (bus slave)		0			ns
61	READ	T_{SRS}	READ Setup Time to the Falling Edge of $\overline{\text{DAS}}$ (bus slave)		0			ns
62	$\overline{\text{READY}}$	T_{RDYD}	Delay from Falling Edge of $\overline{\text{DAS}}$ to Falling Edge of $\overline{\text{READY}}$ (bus slave)	T_{SCT}=100ns		200		ns

Figure 5A : TTL Output Load Diagram.



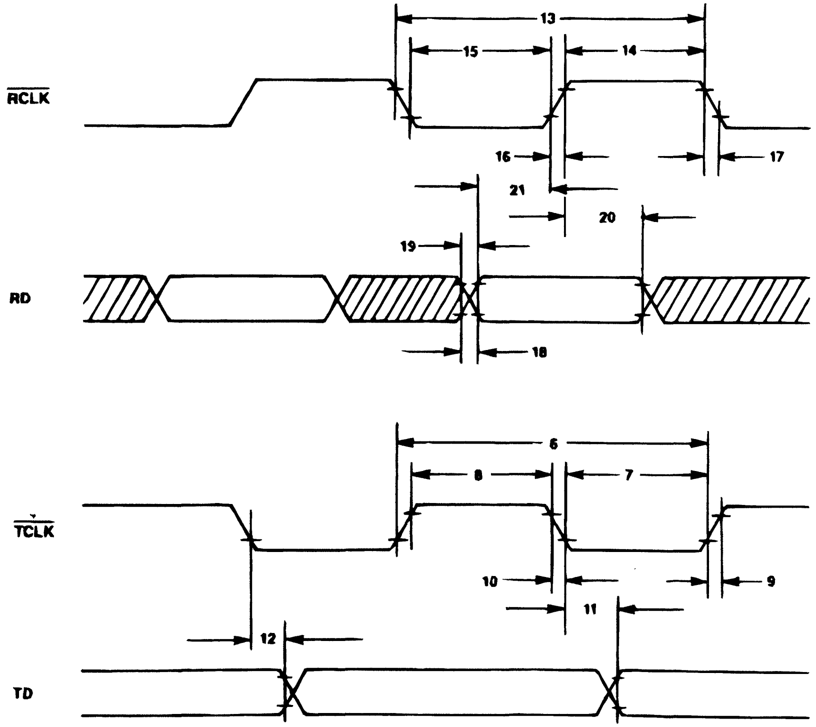
Note : This load is used on all outputs except $\overline{\text{INTR}}$, HOLD, READY.

Figure 5AB : Open Drain Output Load Diagram.



Note : This load is used on open drain outputs $\overline{\text{INTR}}$, $\overline{\text{HOLD}}$, READY.

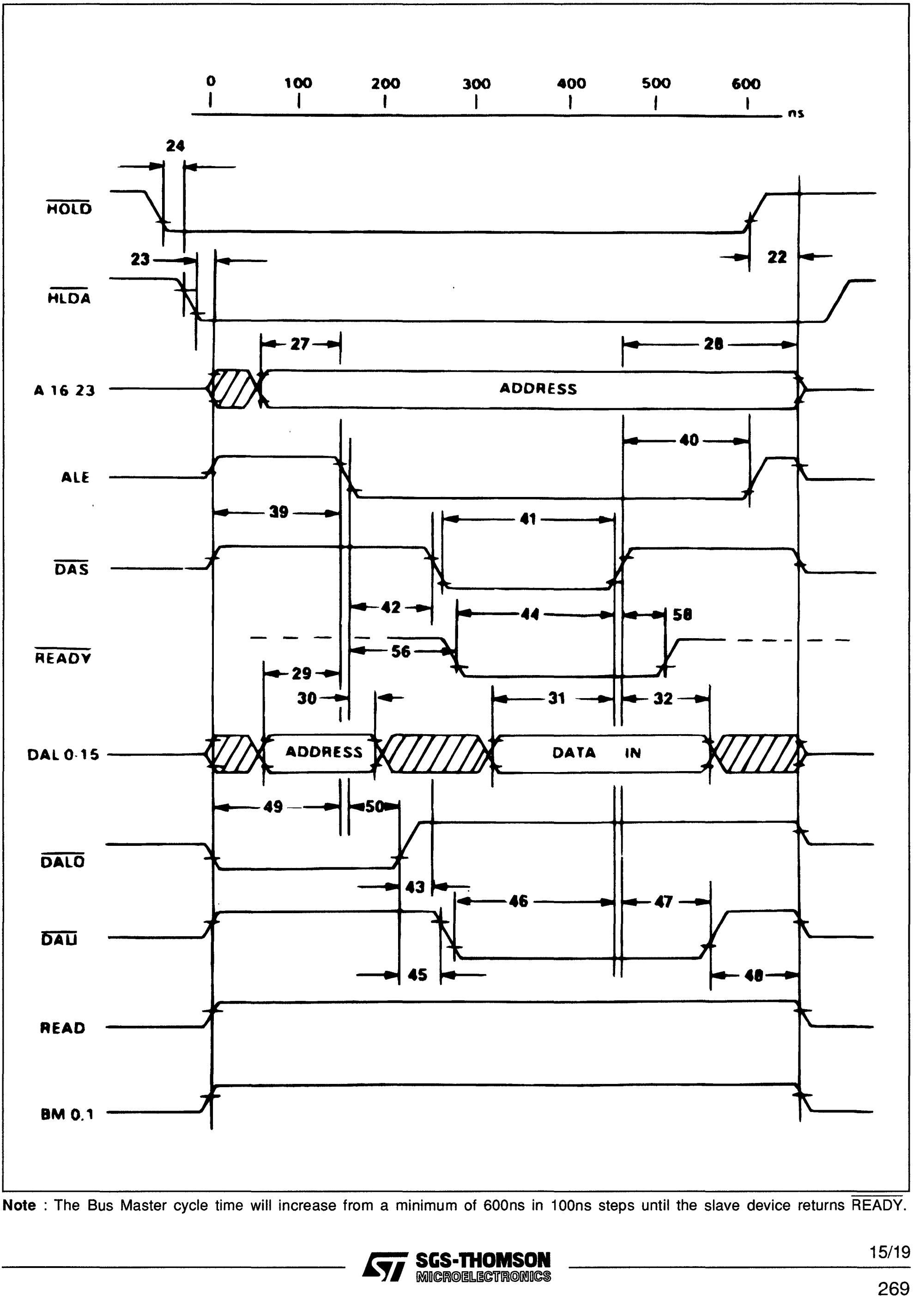
Figure 6 : MK5027 Serial Link Timing Diagram.



Timing measurements are made at the following voltages, unless otherwise specified :

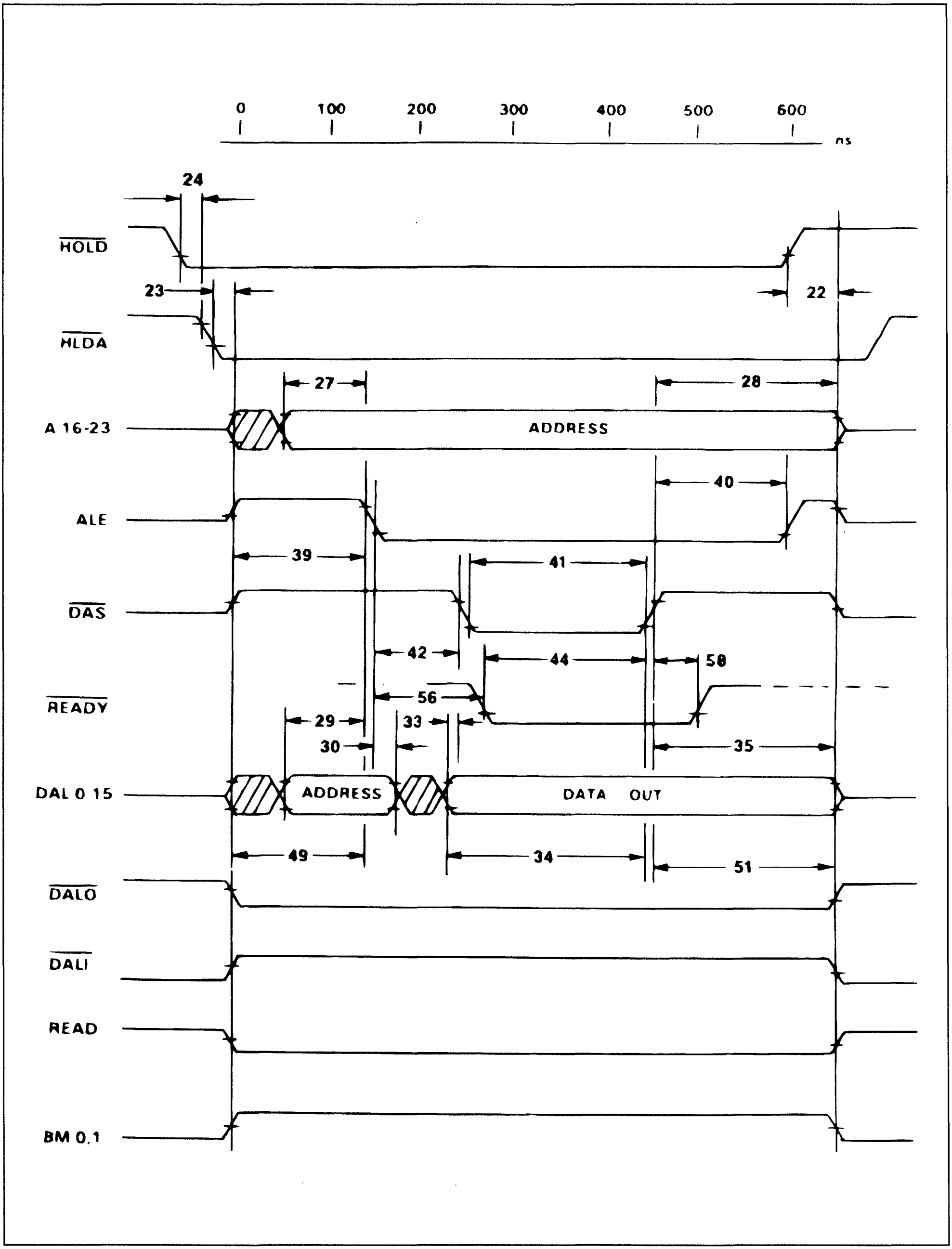
	"1"	"0"
Output	2.0V	0.8V
Input	2.0V	0.8V
Float	10%	90%

Figure 7 : MK5027 Bus Master Timing Diagram (read).



Note : The Bus Master cycle time will increase from a minimum of 600ns in 100ns steps until the slave device returns READY.

Figure 8 : MK5027 Bus Master Timing Diagram (write).



Note : The Bus Master cycle time will increase from a minimum of 600ns in 100ns steps until the slave device returns $\overline{\text{READY}}$.

Figure 9 : MK5027 Bus Slave Timing Diagram (read).

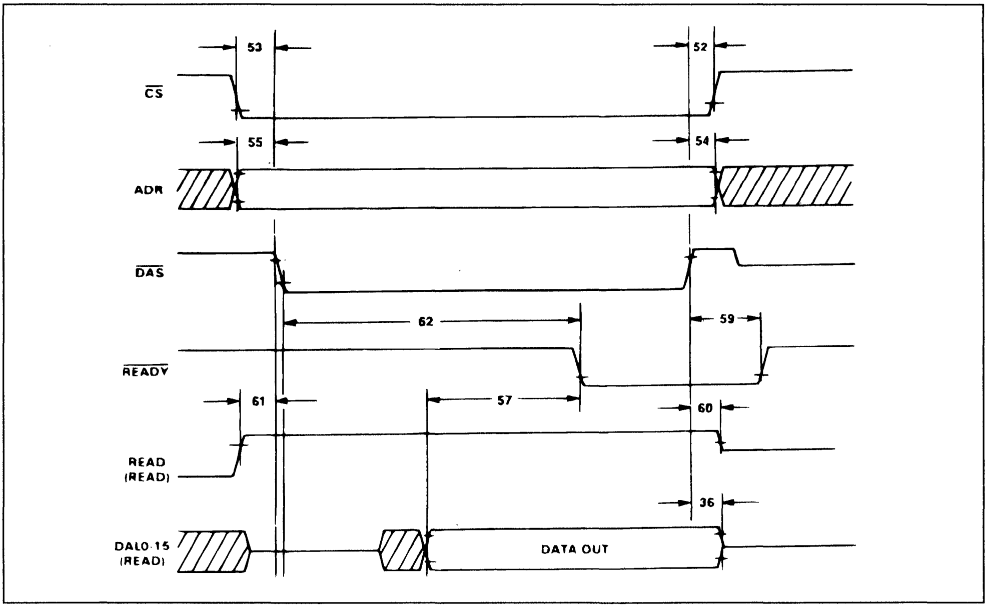
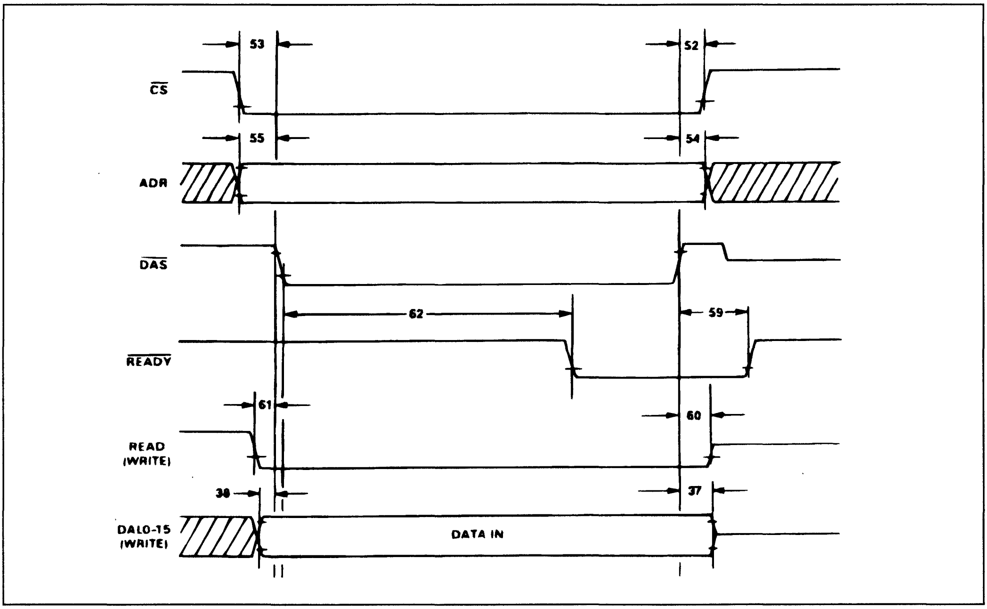
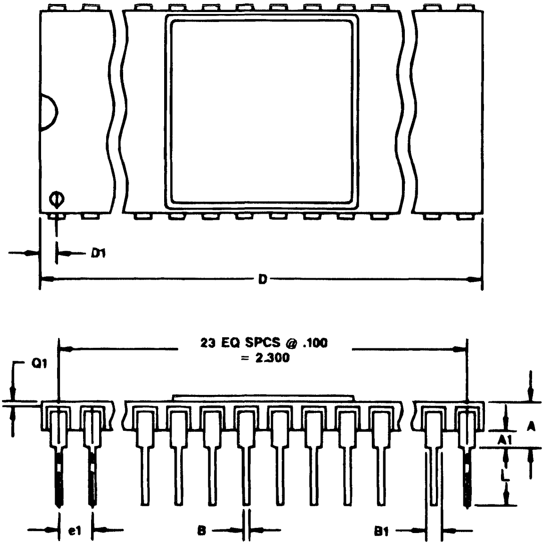


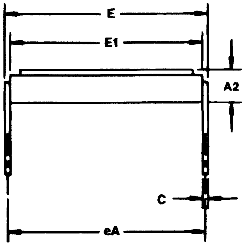
Figure 10 : MK5027 Bus Slave Timing Diagram (write).



PACKAGE DESCRIPTION
5027P
SIDE BRAZED CERAMIC

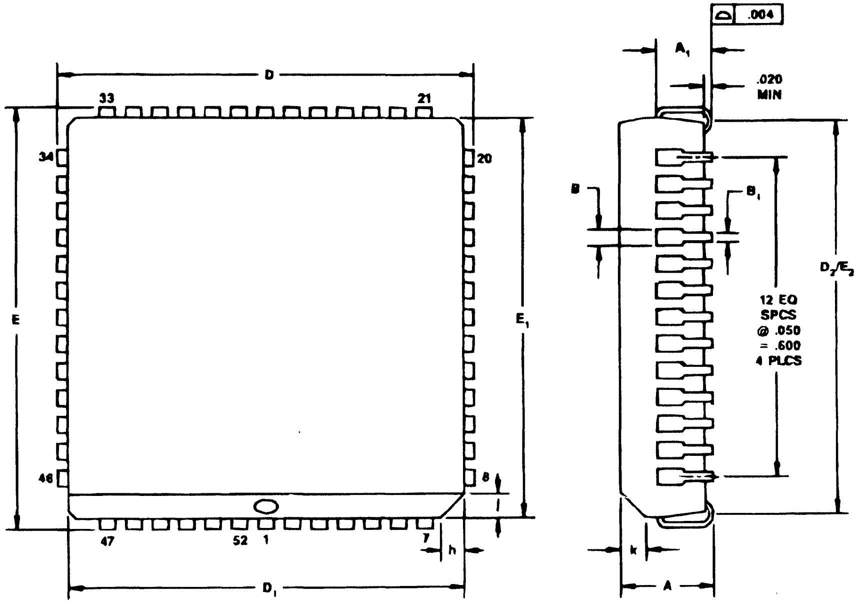
Dim.	Inches		Notes
	Min.	Max.	
A		.175	1
A1	.020		1
A2	.080	.110	
B	.015	.021	2
B1	.038	.057	
C	.008	.012	2
D	2.370	2.430	
D1	.035	.065	
E	.595	.625	
E1	.580	.610	
e1	.090	.110	
eA	.590	.665	
L	.125		
Q	.005		3





Notes : 1. Package standoff to be measured per JEDEC requirements.
2. The maximum limit shall be increased by.003 in. when solder lead finish is specified.
3. Measured from top of ceramic to nearest metallization.

PACKAGE DESCRIPTION
MK5027Q
52 PIN PLCC



Dim.	Inches		Notes
	Min.	Max.	
A	.165	.185	2
A_1	.090	.120	2
B	.026	.032	2
B_1	.013	.021	2
D	.785	.795	
D_1	.750	.756	
D_2	.690	.730	
E	.785	.795	
E_1	.750	.756	
E_2	.690	.730	
h	.042	.060	
j	.042	.060	
k	.042	.056	

PINOUT :

1 V_{SS}
2 DAL07
3 DAL06
4 DAL05
5 DAL04
6 DAL03
7 DO NOT CONNECT
8 DAL02
9 DAL01
10 DAL00
11 READ
12 INTR*
13 DALI*
14 DALO*
15 DAS*
16 BM0*BYTE/BUSREL*
17 DO NOT CONNECT
18 BM1*/BUSAKO
19 HOLD*/BURSQ*
20 ALE/AS*

* = ACTIVE LOW

21 HLDA*
22 CS*
23 ADR
24 READY*
25 RESET*
26 VSS (GROUND)
27 DO NOT CONNECT
28 TCLK*
29 DTR,RTS*
30 RCLK*
31 SYSCLK
32 TD
33 DSR,CTS*
34 RD
35 ADDRESS 23
36 DO NOT CONNECT
37 ADDRESS 22
38 ADDRESS 21
39 ADDRESS 20
40 ADDRESS 19

41 ADDRESS 18
42 ADDRESS 17
43 ADDRESS 16
44 DAL15
45 DAL14
46 DAL13
47 DAL12
48 DAL11
49 DAL10
50 DAL09
51 DAL08
52 VCC

T1 DEVICES

SGS-THOMSON
MICROELECTRONICS

MK2180A

T1 TRANSCEIVER

- SINGLE CHIP DS1 RATE TRANSCEIVER
- PIN AND SOFTWARE COMPATIBLE WITH DS2180A
- SYNCHRONIZES TO FRAMING STANDARDS
  - ➥ 193S - 12 FRAMES/SUPERFRAME
  - ➥ 193E - 24 FRAMES/SUPERFRAME
- ALLOWS COMBINATION OF "CLEAR" AND "ROBBED BIT" DS0 CHANNELS ON THE SAME DS1 LINK
- SELECTABLE PROCESSOR MODE WITH SERIAL PORT INTERFACE FOR CHIP CONFIGURATION AND STATUS MONITORING
- "HARDWIRED" OPTION FOR STAND ALONE APPLICATIONS
- ALARM GENERATION AND DETECTION
- RECEIVED BIT, FRAME, AND MULTIFRAME ERROR DETECTION AND COUNTING FOR MONITORING TRANSMISSION QUALITY
- 5V SUPPLY, LOW POWER CMOS TECHNOLOGY
- SURFACE MOUNT PACKAGE AVAILABLE
- INDUSTRIAL TEMPERATURE RANGE AVAILABLE

DESCRIPTION

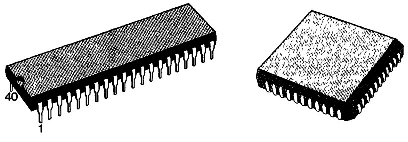
The MK2180A is a monolithic CMOS device designed as a primary rate (1.544Mbs) T-carrier transmitter and receiver. It is characterized by three main functional blocks ; the transmitter, the receiver, and the serial port interface.

The transmitter accepts a serial bit stream generated by 24 codec channels (each channel at 64kbps), then inserts alarm, signaling, and synchronization information. This combined data (1.544Mbps) is processed to provide bipolar encoded pulses to the T1 line interface IC.

The receiver decodes the bipolar 1.544 Mbps bit stream to establish frame and multiframe boundaries and extracting alarm, signaling, and synchronization information for the microprocessor and codec to use in data conditioning and decoding.

The serial port interface eliminates the external glue logic that would otherwise be required to monitor status or set operational mode of the MK2180A. This is easily accomplished by communicating with the device's internal registers using a processor such as the 8051. The serial port pins may be reconfigured as "hardwired" select pins for use without a processor.

The receive synchronizer establishes frame and multi-frame boundaries by identifying frame signaling bits, extracts signaling data, reports alarms and transmission errors, and provides output clocks useful for data conditioning and decoding.



PDIP40 PLCC44

PIN CONNECTIONS

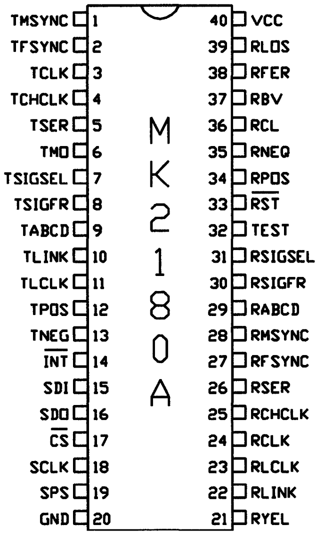


Figure 1 : MK2180A Block Diagram.

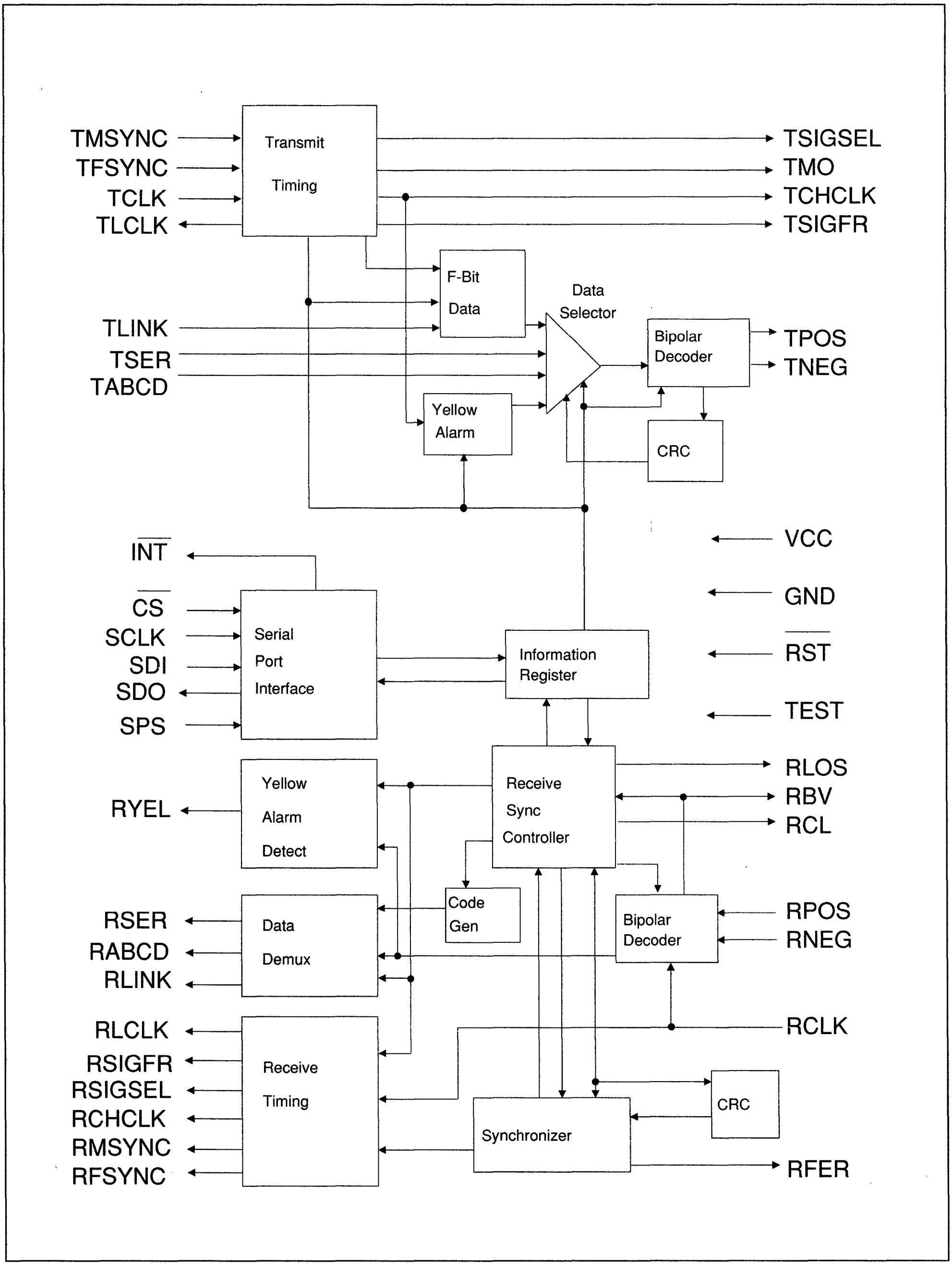


Table 1 : Transmit Pin Description.

Pin	Symbol	Type	Description
1	TMSYNC	I	**Transmit Multiframe Sync.** Rising edge indicates multiframe boundary ; may be pulsed high at multiframe boundaries to reinforce multiframe alignment, or tied low, which allows internal multiframe counter to free run.
2	TFSYNC	I	**Transmit Frame Sync.** Rising edge indicates frame boundary ; may be pulsed every frame to reinforce internal frame counter, or tied low (allowing TMSYNC to establish frame and multiframe alignment).
3	TCLK	I	**Transmit Clock.** 1.544MHz primary clock.
4	TCHCLK	O	**Transmit Channel Clock.** Rising edge of this 192kHz clock indicates channel boundaries. Useful for serial conversion and multiplexing of channel data.
5	TSER	I	**Transmit Serial Data.** NRZ data input ; sampled on falling edge of TCLK.
6	TMO	O	**Transmit Multiframe Out.** Output of internal multiframe counter, indicates multiframe boundaries. 50% duty cycle.
7	TSIGSEL	O	**Transmit Signaling Select.** 0.667kHz clock which indicates signaling frames A and C in 193E framing. 1.33kHz clock in 193S. Useful for multiplexing signaling bits.
8	TSIGFR	O	**Transmit Signaling Frame.** High during signaling frames, low otherwise.
9	TABCD	I	**Transmit ABCD Signaling.** Sampled at LSB time of each channel during signaling frames on falling edge of TCLK, enabled via TCR.4.
10	TLINK	I	**Transmit Link Data.** Sampled on falling edge of TCLK at specified F-bit times for insertion into the outgoing data stream. 193S mode : Sampled on even frames for external insertion of FS bits. 193E mode : Sampled on odd frames for external insertion of FDL bits.
11	TLCLK	O	**Transmit Link Clock.** 4kHz demand clock for TLINK input.
12 13	TPOS TNEG	O	**Transmit Bipolar Data Outputs.** CMOS-level outputs for connection to T1 LIU (Line Interface Unit). Updated on rising edge of TCLK.

Table 2 : Port Pin Description.

Pin	Symbol	Type	Description
14	$\overline{\text{INT}}$[1]	O	**Receive Alarm Interrupt.** Signals host processor during alarm conditions, B8ZS detect, or receive bit errors. Active low, open drain output.
15	SDI[1]	I	**Serial Data In.** Data input to on-board registers. Sampled on rising edge of SCLK
16	SDO[1]	O	**Serial Data Out.** Data output from on-board registers. Updated on falling edge of SCLK, tristated during serial port write or when $\overline{\text{CS}}$ is high. May be tied to SDI.
17	$\overline{\text{CS}}$[1]	I	**Chip Select.** Falling edge indicates start of serial port exchange. Must be held low throughout address/data transaction.
18	SCLK[1]	I	**Serial Data Clock.** Used to write or read the serial port registers.
19	SPS	I	**Serial Port Select.** Tie to VDD to select host processor serial port. Tie to VSS to select hardware mode.

Note : 1. Multifunction pins, see hardware mode description.

Table 3 : Power and Test Pin Description.

Pin	Symbol	Type	Description
20	GND		**Ground.** 0.0V.
32	TEST	I	**Test Mode.** Tie to VSS for normal operation.
40	VDD		**Positive Supply.** 5.0V.

Table 4 : Receive Pin Description.

Pin	Symbol	Type	Description
21	RYEL	O	**Receive Yellow Alarm.** Transitions high when yellow alarm detected, returns low when alarm condition no longer received.
22	RLINK	O	**Receive Data Link.** Updated with extracted link data one RCLK before start of specified frames (channel 24, bit position 8). Held until next update. 193S mode : Updated before even frames with extracted FS bits. 193E mode : Updated before odd frames with extrated FDL bits.
23	RLCLK	O	**Receive Link Clock.** 4kHz demand clock for RLINK.
24	RCLK	I	**Receive Clock.** 1.544MHz primary clock.
25	RCHCLK	O	**Receive Channel Clock.** Rising edge of this 192kHz clock indicates channel boundaries. Useful for parallel conversion and demultiplexing of serial channel data.
26	RSER	O	**Receive Serial Data.** Received NRZ serial data, updated on rising edges of RCLK.
27	RFSYNC	O	**Receive Frame Sync.** Extracted 8kHz pulse indicates F-bit position of each frame, one RCLK wide.
28	RMSYNC	O	**Receive Multiframe Sync.** Extracted multiframe sync ; rising edge indicates beginning of each multiframe, 50% duty cycle.
29	RABCD	O	**Receive ABCD Signaling.** Extracted signaling data output, valid for each channel time in signaling frames. In non-signaling frames, RABCD outputs the LSB of each channel word.
30	RSIGFR	O	**Receive Signaling Frame.** High during signaling frames, low during resync and non-signaling frames.
31	RSIGSEL	O	**Receive Signaling Select.** In 193E framing a 0.667kHz clock which indicates signaling frames A and C. A 1.33kHz clock in 193S.
33	$\overline{\text{RST}}$	I	**Reset.** A high-low transition clears all internal registers and resets receive side counters. A high-low transition will initiate a receive resync.
34 35	RPOS RNEG	I	**Receive Bipolar Data Inputs.** Sampled on falling edge of RCLK. Tie together to receive NRZ data and disable bipolar violation monitoring circuitry.
36	RCL	O	**Receive Carrier Loss.** High if 32 consecutive "0's" are received at RPOS and RNEG, goes low after next "1".
37	RBV	O	**Receive Bipolar Violation.** High during accused bit time at RSER if bipolar violation detected, low otherwise.
38	RFER	O	**Receive Frame Error.** High during F-bit time when F_T or F_S errors occur (193S), or when FPS or CRC errors occur (193E). Low during resync.
39	RLOS	O	**Receive Loss of Resync.** Indicates sync status ; high when internal resync in progress, low otherwise.

Table 5 : Register Summary.

Register	Address	T/R[1]	Description/Function
RSR	0000	R[2]	**Receive Status Register.** Reports all receive alarm conditions.
RIMR	0001	R	**Receive Interrupt Mask Register.** Allows masking of individual alarm and receive error generated interrupts.
BVCR	0010	R	**Bipolar Violation Count Register.** 8 bit presettable counter which records individual bipolar violations. Received B8ZS is counted as violation if disabled.
ECR	0011	R	**Error Count Register.** 2 independent 4-bit counters. Upper nibble counts OOF events, lower nibble counts individual F-bit (193S) or CRC (193E) errors.
CCR[3]	0100	T/R	**Common Control Register.** Controls operating modes common to both receive and transmit sides.
RCR	0101	R	**Receice Control Register.** Controls operating modes unique to receive side.
TCR	0110	T	**Transmit Control Register.** Controls operating modes unique to transmit side.
TIR1 TIR2 TIR3	0111 1000 1001	T T T	**Transmit Idle Registers.** Specify which outgoing channels are to be replaced with an idle code (7F or FF hex, controlled by TCR.3).
TTR1 TTR2 TTR3	1010 1011 1100	T T T	**Transmit Transparent Registers.** Specify which outgoing channels are to be transparent. (No robbed bit signaling or bit 7 zero suppression.)
RMR1 RMR2 RMR3	1101 1110 1111	R R R	**Receive Mark Registers.** Specify which incoming channels are to be replaced with an idle or digital milliwatt code (7F hex or digital milliwatt, controlled by RCR.4).

Notes :
[1] Transmit or receive side register.
[2] RST is a read only register, all other registers are read/write.
[3] Reserved bit location in this register should be programmed to 0 for proper operation.

CONTROL

The MK2180A can be controlled in one of two ways : with a processor via the serial port or by reconfiguring the serial port pins as hardwired select pins. Either method permits control of frame, zero suppression, alarm and signaling formats. Processor mode permits complete device configuration and status monitoring. Features such as selective transparent channels, idle code insertion, and alteration of sync algorithm are unavailable in hardware mode.

SERIAL PORT INTERFACE

Pins 14 through 18 of the MK2180A serve as a microprocessor/microcontroller compatible serial port. Sixteen on-board registers allow the host processor to configure the device and monitor receive status, minimizing hardware interfaces. Port read/write timing is unrelated to transmit and receive timing, thus permitting asynchronous reads and/or writes by the host.

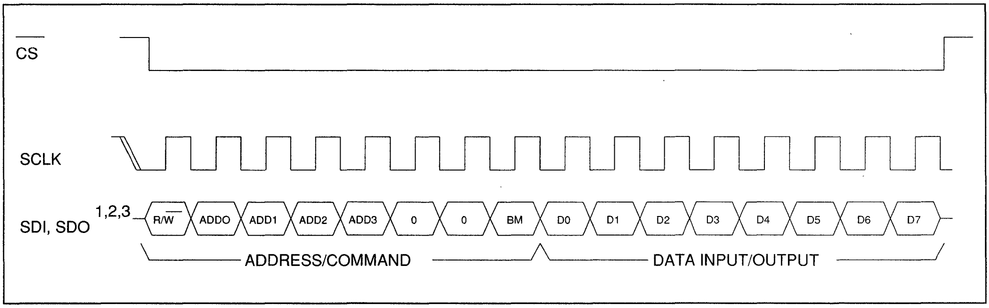
Figure 2 : ACB : Address Command Byte.

(MSB) (LSB)

BM	0	0	ADD3	ADD2	ADD1	ADD0	R/W

Symbol	Position	Name and Description
BM	ACB.7	Burst Mode; if set (and ACB.1 through ZCA.4 = 0) burst read or write is enabled
0	ACB.6	Reserved, must be 0 for proper operation.
0	ACB.5	Reserved, must be 0 for proper operation.
ADD3	ACB.4	MSB of register address.
ADD0	ACB.1	LSB of register address.
R/W	ACB.0	**Read/Write Select** 0 = write addresses register ; 1 = read addessed register.

Figure 3 : Serial Port Read/Write.



Notes : 1. Transmit or receive side register.
2. RST is a read only register, all other registers are read/write.
3. Reserved bit location in this register should be programmed to 0 for proper operation.

ADDRESS/COMMAND

Communication with the control/status registers requires writing one address/command byte prior to the transfer of register data. The first bit of the address/command byte indicates register read or write. The following four bits specify the register address. The next two bits are reserved and must be set to zero when communicating with the MK2180A. (The device ignores all serial port activity until chip select transitions high and then low again if either of these two bits are set. This feature will allow other devices to be directly wired to the serial bus while residing in a different register address space, thus eliminating external chip decoding.) The last bit of the address/command byte enables burst mode when set ; burst mode causes all registers to be consecutively written or read. *Data is writen to and read fromthe transceiver LSB first.*

Figure 2 : ACB : Address Command Byte.

(MSB) (LSB)

0	COFA	EFYEL	FM	YELMD	B8ZS	B7	LPBK

Symbol	Position	Name and Description
0	ACB.7	Reserved, must be 0 for proper operation.
COFA	ACB.6	**Replace B8ZSD with COFA (RSR.2).** 0 = Detected B8ZS code words reported at RSR.2. 1 = COFA (Change-of-Frame Alignment) reported at RSR.2 when last resync resulted in change of frame or multiframe alignment.
EFYE	ACB.5	**Extended Framing Yellow Alarm Mode Select.** Determines yellow alarm type to be transmitted and detected while in 193E framing. 0 = Yellow alarm is a repeating pattern of 00 and FF hex on the 4kHz data link. 1 = Yellow alarm is a "0" in bit 2 position of all channels.
FM	ACB.4	**Frame Mode Select.** 0 = D4 (193S, 12 frames/superframe). 1 = Extended (193E, 24 frames/superframe).
YELMD	ACB.3	**193S Yellow Mode Select.** Determines yellow alarm to be transmitted and detected while in 193S framing. 0 = Yellow alarm is a "0" in bit 2 of all channels. 1 = Yellow alarm is a "1" in the S-bit position of frame 1.

B8ZS	CCR.2	**Bipolar Eight Zero Substitution.** 0 = No B8ZS ; 1 = B8Zs enabled. *Note : B8Zs and B7 should not be enabled at the same time.*
B7	CCR.1	**Bit Seven Zero Suppression.** 0 = No bit 7 zero suppression 1 = Channels with all zero content will be transmitted with bit 7 forced to "1".
LPBK	CCR.0	**Loopback.** 0 = Isolated operation of transmit and receive sides. 1 = Device internally loops output transmit data into the incoming receive data buffers. TCLK is internally substituted for RCLK.

CHIP SELECT AND CLOCK CONTROL

All data transfers are initiated by driving the $\overline{CS}$ input low. Only one serial port exchange is acknowledged after $\overline{CS}$ goes low. Input data is latched on the rising edge of SCLK and *must be valid during the previous low period of SCLK to prevent momentary corruption of register data during writes*. Data is output on the falling edge of SCLK and held to the next falling edge. All data transfers are terminated if the $\overline{CS}$ input transitions high. Port control logic is disabled and SDO is tristated when $\overline{CS}$ is high.

DATA I/O

Following the 8 SCLK cycles that input an address/command byte to write, a data byte is strobed into the addressed register on the rising edges of the next 8 SCLK cycles. Following an address/command byte to read, contents of the selected register are output on the falling edges of the next 8 SCLK cycles. The SDO pin is tristated during device write, and may be tied to SDI in applications where the host processor has a bidirectional I/O pin.

Figure 5 : TCR : Transmit Control Register.

(MSB) (LSB)

ODF	TCP	tcp	RBSE	TIS	193SI	TBL	TYEL

Symbol	Position	Name and Description
ODF	TCR.7	**Output Data Format.** 0 = Bipolar data at TPOS & TNEG. 1 = NRZ data at TPOS ; TNEG = 0.
FPT	TCR.6	**FT/FPS Pass Through.** 0 = FT/FPS sourced internally. 1 = FT/FPS sampled at TSER during F-bit time.
TCP	TCR.5	**Transmit CRC Pass-through.** 0 = Transmit CRC code internally generated. 1 = TSER sampled at CRC F-bit time for external CRC insertion.
RBSE	TCR.4	**Robbed Bit Signaling Enable.** 0 = No signaling is inserted. 1 = Signaling is inserted. Use TTR registers to select individual clear channels.
TIS	TCR.3	**Transmit Idle Code Select.** Determines idle code format to be inserted into channels marked by the TIR registers. 0 = Insert 7F (Hex) into marked channels. 1 = Insert FF (Hex) into marked channels.
193SI	TCR.2	**193S S-bit Insertion.** Determines source of transmitted S-bit. 0 = Internal S-bit generator. 1 = External (sampled at TLINK input). No effect in 193E mode.

TBL	TCR.1	**Transmit Blue Alarm.** 0 = Disabled. 1 = Enabled.
TYEL	TCR.0	**Transmit Yellow Alarm.** 0 = Disabled. 1 = Enabled.

BURST MODE

Burst mode allows all on-board registers to be consecutively read or written by the host processor. A burst read is used to poll all registers ; RSR contents will be unaffected. This feature minimizes device initialization time on power-up or system reset. Burst mode is initiated when ACB.7 is set and the address nibble is 0000. Burst is terminated by a low-high transition on $\overline{CS}$ or end of registers.

LOOPBACK

Enabling loopback will typically induce an out-of-frame "OOF" condition. If appropriate bits are set in the receive control register, the receiver will resync to the looped transmit frame alignment. During the looped condition, the transmit outputs (TPOS, TNEG) will transmit unframed all "1's". All operating modes (B8ZS, alarm, signaling, etc.) except for blue alarm transmission are available in loopback.

Figure 6 : TTR1 - TTR3 : Transmit Transparency Registers.

(MSB) (LSB)

CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	TTR1
CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	TTR2
CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	TTR3

Symbol	Position	Name and Description
CH24	TTR3.7	**Transmit Transparency Registers.** Each of these bit positions represents a DSO channel in the outgoing frame. When set, the corresponding channel is transparent. When cleared, robbed-bits and bit 7 stuffing may be inserted.
CH1	TTR1.0	

Figure 7 : TTR1 - TTR3 : Transmit Idle Registers.

(MSB) (LSB)

CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	TIR1
CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	TIR2
CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	TIR3

Symbol	Position	Name and Description
CH24	TTR3.7	**Transmit Idle Registers.** Each of these bit positions represents a DSO channel in the outgoing frame. When set, the corresponding channel will output an idle code specified by TCR.3. When cleared, no idle code will be inserted.
CH1	TTR1.0	

BIT SEVEN STUFFING

Existing systems meet one's density requirements by forcing bit 7 of all zero channels to 1. Bit 7 stuffing is "globally" enabled by asserting bit CCR.1, and may be disabled on an individual channels basis by setting appropriate bits in TTR1-TTR3.

B8ZS

The MK2180A supports existing and emerging zero suppression formats. Selection of B8ZS coding maintains system one's density requirements without disturbing data integrity as required in clear channel applications. B8ZS coding replaces any 8 consecutive outgoing zeros with a B8ZS code word. Any B8ZS code word received is replaced with all zeros.

TRANSMIT BLUE ALARM

The blue alarm (also known as the AIS, Alarm Indication Signal) is an unframed, all 1's sequence enabled by asserting TCR.1. Blue alarm overrides all other transmit data patterns and is disabled by clearing TCR.1. Use of the TIR registers allows a framed, all 1's alarm transmission if required by the network.

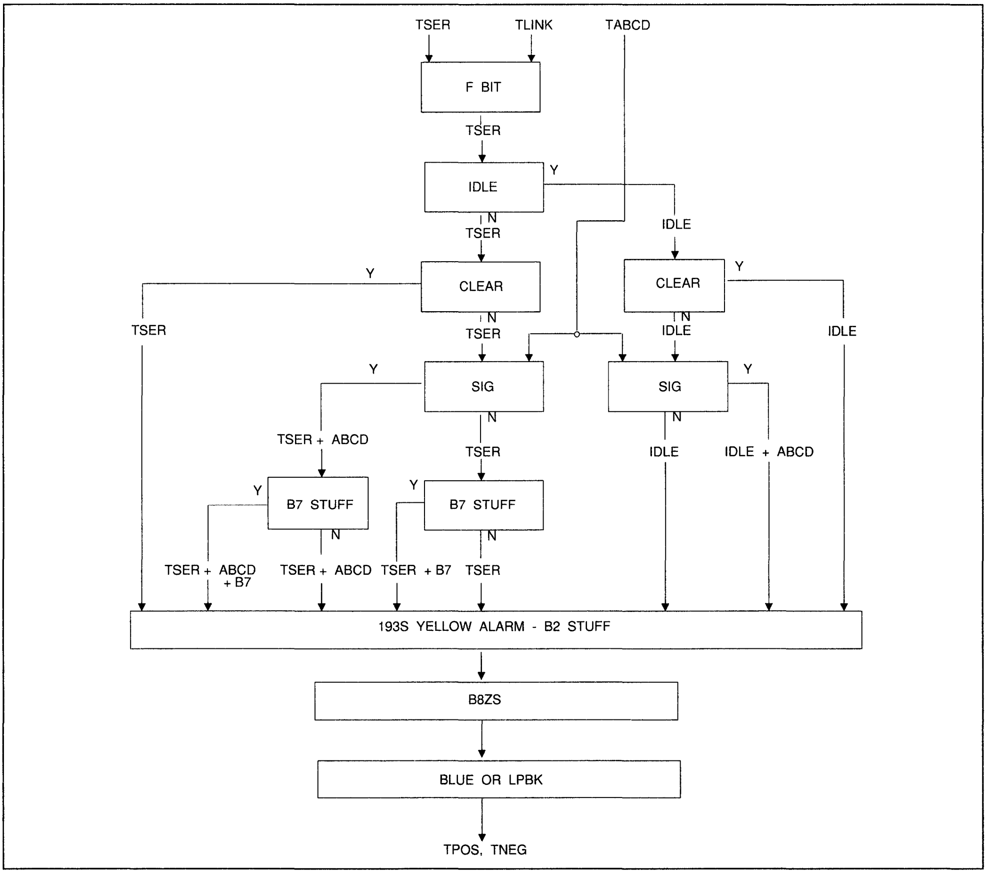
TRANSMIT YELLOW ALARM

In 193E framing, yellow alarm type is determined by the state of CCR.5. In 193S framing, yellow alarm type is determined by the state of CCR.3. In both modes, transmission of yellow alarm is enabled by asserting TCR.0 and disabled by clearing TCR.0.

TRANSMIT SIGNALING

When enabled (via TCR.4) channel signaling is inserted in frames 6 and 12, (193S) or 6, 12 and 18 and 24 (193E) in the 8th bit position of every channel word. Signaling data is sampled at TABCD on the falling edge of TCLK during 8 of each input word during signaling frames. Logical combination of clocks TMO, TSIGFR and TSIGSEL allow external multiplexing of seperate serial links for A, B, or A, B, C, D signaling sources.

Figure 8 : Transmit Insertion Hierarchy.



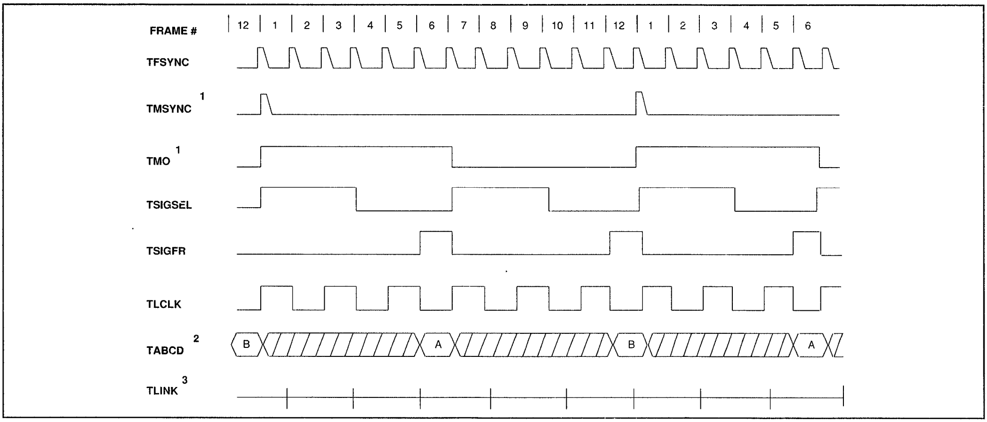
TRANSMIT CHANNEL TRANSPARENCY

Individual DSO channels in the T1 frame may be programmed clear (no inserted robbed bit signaling and no bit 7 zero suppression) by setting the appropriate bits in the transmit transparency registers. Channel tranparency is required in mixed voice/data or data-only environments such as ISDN, where data integrity must be maintained.

TRANSMIT IDLE CODE INSERTION

Individual outgoing channels in the frame can be programmed with idle code by asserting the appropriate bits in the transmit idle registers. One of two idle code formats, 7F (Hex) and FF (Hex) may be selected by the user via TCR.3. If enabled, robbed bit signaling data is inserted into the idle channel, unless the appropriate TTR bit is set for that channel. This feature eliminates external hardware currently required to intercept and stuff unoccupied channels in the DS1 bit stream.

Figure 9 : 193S Transmit Multiframe Timing.



Notes : 1. Transmit frame and multiframe timing may be established in one of four ways :

a) With TFSYNC tied low, TMSYNC may be pulsed high once every multiframe period to establish multiframe boundaries, allowing internal counters to determine frame timing.

b) TFSYNC may be pulsed every 125 microseconds ; pulsing TMSYNC once establishes multiframe boundaries.

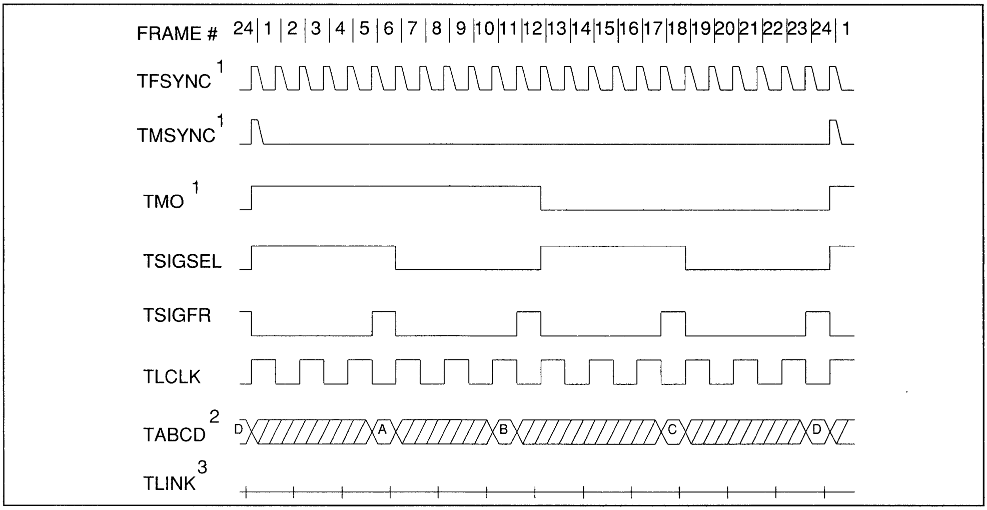
c) TMSYNC and TFSYNC may be continuously pulsed to establish and reinforce frame and multiframe timing.

d) If TMSYNC is tied low and TFSYNC is pulsed at frame boundaries, the transmitter will establish an arbitrarymultiframe boundary as indicated by TMO.

2. Channels in which robbed bit signaling is enabled will sample TABCD during the LSB bit (bit position 8) timing in frames indicated.

3. When external S-bit insertion is enabled, TLINK will be sampled during the F-bit time of even frames and inserted into the outgoing data stream.

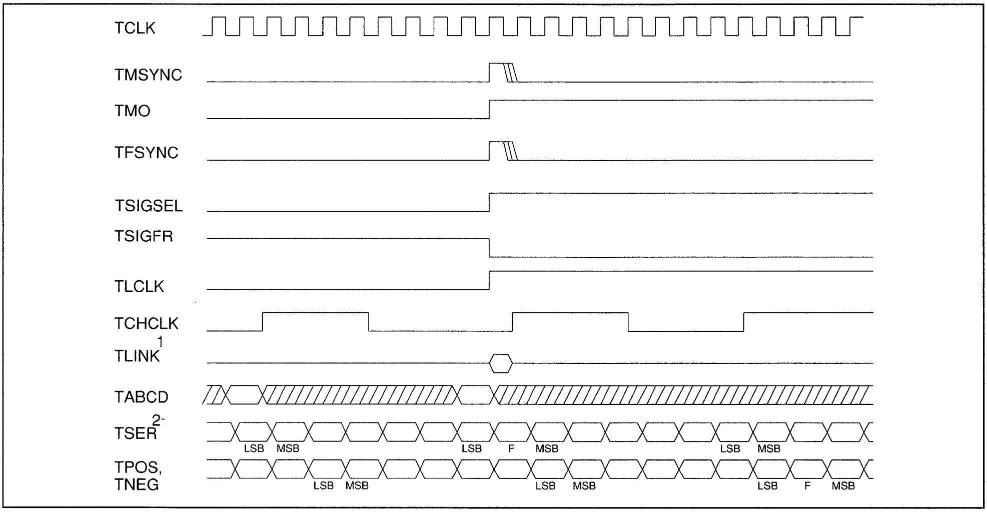
Figure 10 : 193E Transmit Multiframe Timing.



Notes : 1. Transmit frame and multiframe timing may be established in one of four ways :
a) With TFSYNC tied low, TMSYNC may be pulsed high once every multiframe period to establish multiframe boundaries, allowing internal counters to determine frame timing.
b) TFSYNC may be pulsed every 125 microseconds ; pulsing TMSYNC once establishes superframe boundaries.
c) TMSYNC and TFSYNC may be continuously pulsed to establish and reinforce frame and multiframe timing.
d) If TMSYNC is tied low and TFSYNC is pulsed at frame boundaries, the transmitter will establish an arbitrarymultiframe boundary as indicated by TMO.

2. Channels in which robbed bit signaling is enabled will sample TABCD during the LSB bit (bit position 8) timing in frames indicated.
3. TLINK is sampled during the F-bit time of odd frames and inserted into the outgoing data stream (FDL data).

Figure 11 : Transmit Multiframe Boundary Timing.



Notes : 1. TLINK timing shown is for 193E framing ; in 193E framing, TLINK is sampled as indicated for insertion into F-bit position of odd frames. When S-bit insertion is enabled in 193S, TLINK is sampled during even frames.
2. If TCR.5 = 1 ; TSER is sampled during the F-bit time of CRC frames for insertion into outgoing data stream (193E framing only).

Figure 12 : Receive Control Register.

(MSB) (LSB)

ARC	OOFCD	RCI	RCS	SYNCC	SYNCT	SYNCE	RESYNC

Symbol	Position	Name and Description
ARC	DCR.7	**Auto Resync Criteria.** 0 = Resync on OOF or RCL events. 1 = Resync on OOF only.
OOFCD	DCR.6	**Out-of-Frame (OOF) Condition Detected.** 0 = OOF declared when 2 of 4 framing bits are in error. 1 = OOF declared when 2 of 5 framing bits are in error.
RCI	DCR.5	**Receive Code Insert.** 0 = No receive code inserted. 1 = Receive code inserted into channels marked by RMR registers. Code type selected by RCR.4.
RCS	DCR.4	Receive Code Select. 0 = Idle code (7F Hex). 1 = Digital milliwatt.
SYNCC	DCR.3	**Sync Criteria.** Determines the type of algorithm utilized by the receive synchronizer and differs for each frame mode. **193S Framing (CCR.4 = 0).** 0 = Synchronize to frame boundaries using FT patterns, then search for multiframe by using FS. 1 = Cross couple FT and FS patterns in sync algorithm. **193E Framing (CCR.4 = 1).** 0 = Normal sync (utilizes FPS only). 1 = Validate new alignment with CRC before declaring sync.
SYNCT	DCR.2	**Sync Time.** 0 = 10 consecutive F-bits are qualified before sync is declared. 1 = 24 consecutive F-bits are qualified before sync is declared..
SYNCE	DCR.1	**Sync Enable.** 0 = Automatically resync is performed when ARC (RCR.7) conditions are met. 1 = No auto resync is performed (RCR.0 must be used to initiate resync).
RESYNC	DCR.0	**Resync.** When toggled low to high, the transceiver will initiate resync immediately. The bit must first be cleared prior to subsequent resyncs.

RECEIVE CODE INSERTION

Incoming receive channels can be replaced with idle (7F Hex) or digital milliwatt (u-LAW format) codes. The receive mark registers indicate which channels are inserted. Bit RCR.5 serves as a "global" enable for marked channels, and bit RCR.4 selects inserted code format : 0 = idle code, 1 = digital milliwatt.

RECEIVE SYNCHRONIZER

Bits RCR.0 through RCR.3 allow the user to control operational characteristics of the synchronizer. Sync algorithm, candidate qualify testing, auto resync, and command resync modes may be altered at any time in response to changing span conditions.

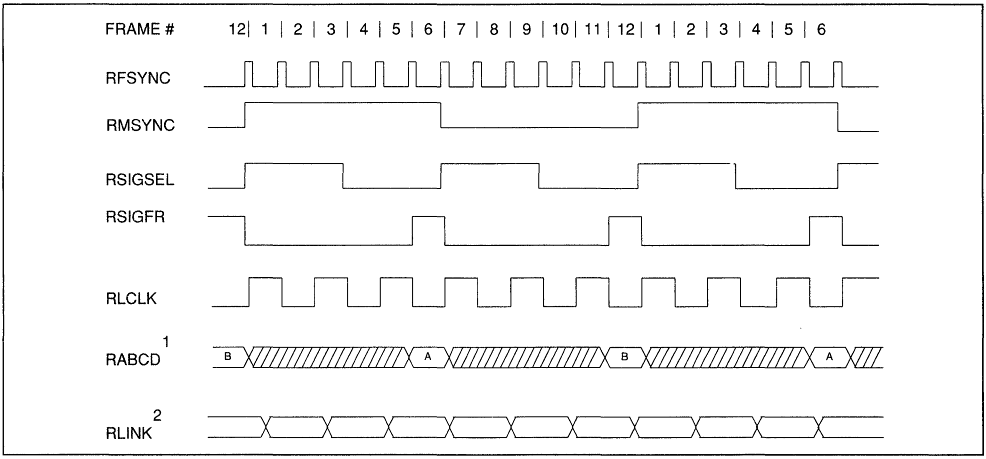
Figure 13 : RMR1 - RMR3 : Receive Mark Registers.

(MSB) (LSB)

CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	RMR1
CH16	CH15	CH14	CH13	CH12	CH11	CH10	CH9	RMR2
CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	RMR3

Symbol	Position	Name and Description
CH24	RMR3.7	**Receive Mark Registers.** Each of these bit positions represents a DSO channel in the incoming T1 frame. When set, the corresponding channel will output codes determined by RCR.4 and RCR.5.
CH1	RMR1.0	

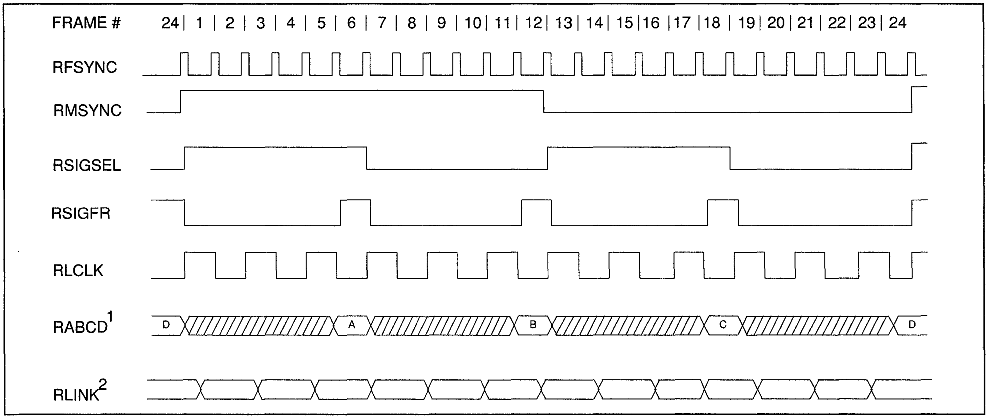
Figure 14 : 193S Receive Multiframe Timing.



Notes : 1. Signaling data is updated during signaling frames on channel boundaries. RABCD is the LSB of each channel word in non-signaling frames.

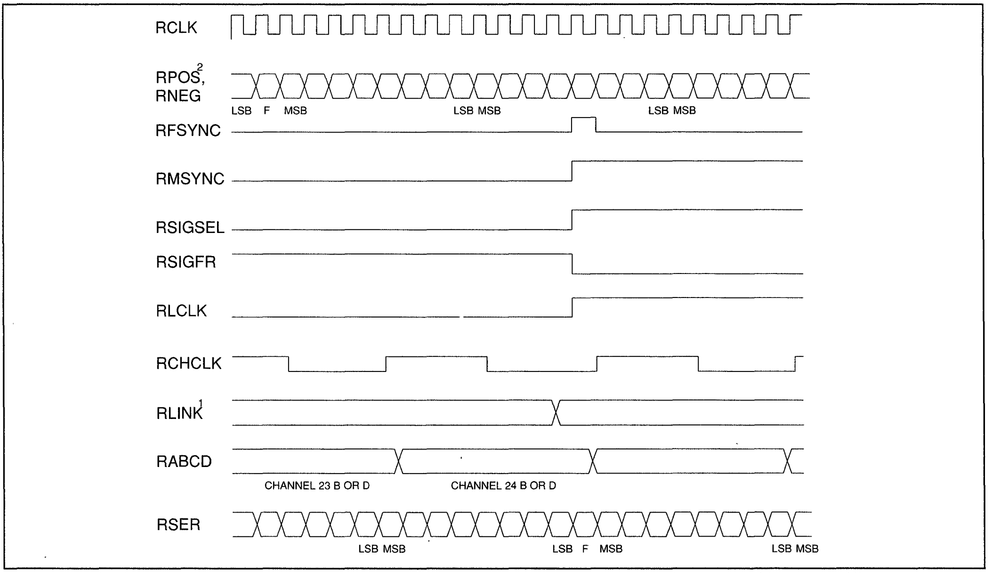
2. RLINK data (S-bit) is updated one bit time prior to odd frames and held for two frames.

Figure 15 : 193E Receive Multiframe Timing.



Notes : 1. Signaling data is updated during signaling frames on channel boundaries. RABCD is the LSB of each channel word in non-signaling frames.
2. RLINK data (FDL-bit) is updated one bit time prior to odd frames and held for two frames.

Figure 16 : Receive Multiframe Boundary Timing.



Notes : 1. RLINK timing is shown for 193E ; in 193S RLINK is updated on even frame boundaries and is held across multiframe edges..
2. Total delay from RPOS and RNEG to RSER output is 13 RCLK periods.

Figure 17 : RSR : Receive Status Register.

(MSB) (LSB)

BVCS	ECS	RYEL	RCL	FERR	B8ZSD	RBL	RLOS

Symbol	Position	Name and Description
BVCS	RSR.7	**Bipolar Violation Count Saturation.** Set when the 8-bit bipolar violation counter saturates to FF (hex).
ECS	RSR.6	**Error Count Saturation.** Set when either of the 4-bit error count register saturates to F (hex).
RYEL	RSR.5	**Receive Yellow Alarm.** Set when yellow alarm detected. (Detected yellow alarm format determined by CCR.5, CCR.4 and CCR.3).
RCL	RSR.4	**Receive Carrier Loss.** Set when 32 consecutive "0's" appear ar RPOS and RNEG.
FERR	RSR.3	**Frame Bit Error.** Set when FT (193S) or FPS (193E) bit error occurs.
B8ZSD	RSR.2	**Bipolar Eight Zero Substitution Detect.** Set when B8ZS code word detected . (Not effected by CCR.2).
RBL	RSR.1	**Receive Blue Alarm**. Set when 2 consecutive Frames have less than 3 zeros (total) in the data stream (F-bit positions not tested).
RLOS	RSR.0	**Receive Loss of Sync.** Set when resync is in process ; if RCR.1 = 0, RLOS transistions high on an OOF event or carrier loss, indicating auto resync.

RECEIVE SIGNALING

Robbed bit signaling data is presented at RABCD during each channel time in signaling frames for all 24 incoming channels. Logical combination of clocks RMSYNC, RSIGFR and RSIGSEL allow the user to identify and extract AB or ABCD signaling data.

RECEIVE ALARM REPORTING

Incoming serial data is monitored by the transceiver for alarm occurences. Alarm conditions are reported in two ways : via transitions on the alarm output pins and registered interrupt, in which the host controller reads the RSR in response to an alarm driven interrupt. Interrupts may be direct, in which the transceiver demands service for a real time alarm, or count-overflow triggered, in which an on-board alarm event counter exceeds a user-programmed threshold. The user may mask individual alarm condiditons by clearing the appropriate bits in the receive interrupt mask register (RIMR).

ALARM SERVICING

The host controller must service the transceiver in order to clear an interrupt condition. Clearing appropriate bits in the RIMR will unconditionally clear an interrupt. Direct interrupts (those driven from real-time alarms) will be cleared when the RSR is directly read, unless the alarm condition still exists. Count-overflow interrupts (BVCS, ECS) are not cleared by a direct read of the RSR. They will be cleared only when the user presets the appropriate count register to a value other than all "1's." A burst read of the RSR will not clear an interrupt condition

Figure 18 : RIMR : Receive Interrupt Mask Register.

(MSB)							(LSB)
BVCS	ECS	RYEL	RCL	FERR	B8ZSD	RBL	RLOS

Symbol	Position	Name and Description
BVCS	RIMR.7	**Bipolar Violation Count Saturation Mask.** 1 = Interrupt enabled ; 0 = Interrupt Masked
ECS	RIMR.6	**Error Count Saturation Mask.** 1= Interrupt enabled ; 0 = Interrupt Masked
RYEL	RIMR.5	**Receive Yellow Alarm Mask.** 1= Interrupt enabled ; 0 = Interrupt Masked
RCL	RRIM.4	**Receive Carrier Loss Mask.** 1= Interrupt enabled ; 0 = Interrupt Masked
FERR	RIMR.3	**Frame Bit Error Mask.** 1= Interrupt enabled ; 0 = Interrupt Masked

Symbol	Position	Name and Description
B8ZSD	RIMR.2	**Bipolar Eight Zero Substitution Detect Mask.** 1 = Interrupt enabled ; 0 = Interrupt masked
RBL	RIMR.1	**Receive Blue Alarm Mask.** 1= Interrupt enabled ; 0 = Interrupt masked
RLOS	RIMR.0	**Receive Loss of Sync. Mask.** 1= Interrupt enabled ; 0 = Interrupt masked

ALARM COUTERS

The three on-board alarm event counters allow the transceiver to monitor and record error events without processor intervention on each occurrence. All of these counters are presettable by the user, establishing an event count interrupt threshold. As each counter saturates, the next error event occurrence will set a bit in the RSR and generate an interrupt unless masked. The user may read these registers at any time, in many systems, the host will periodically poll these registers to establish link error rate performance.

OOF EVENTS AND ERRORED SUPERFRAMES

Out of frame is declared when two (or more) of four consecutive framing bits are in error. FT bits are tested for OOF occurrence in 193S, the FPS bits are tested in 193E. OOF events are recorded by the 4-bit OOF counter in the error count register. In the 193E framing mode, the OOF event is logically "OR'ed" with an on-chip generated CRC checksum. This event, known as errored superframe, is recorded by the 4-bit ESF error counter in the error count register. In the193S framing mode, the 4-bit ESF error counter records individual FT and FS errors when RCR.3 = 1, or FT errors only when RCR.3= 0.

ALARM OUTPUTS

The transceiver also provides direct alarm outputs for applications when additional decoding and demuxing are required to supplement the on-board alarm logic.

RLOS OUTPUT

The receiver loss of sync output indicates the status of the receiver synchronizer circuitry: when high, an off-line resynchronization is in progress and a high-low transition indicates resync is complete. The RLOS bit (RSR.0) is a "latched" version of the RLOS output. If the auto-resync mode is selected (RCR.1= 0) RLOS is a real time indication of a carrier loss or OOF event occurrence.

Figure 19 : Bipolar Violation Count Register.

(MSB) (LSB)

BVD7	BVD6	BVD5	BVD4	BVD3	BVD2	BVD1	BVD0

Symbol	Position	Name and Description
BVD7	BVCR.7	MSB of bipolar violation count.
BVD0	BVCR.0	LSB of bipolar violation count

This 8 bit binary up counter saturates at 255 and will generate an interrupt for each occurance of a bipolar violation once saturated (RMR.7 = 1). Presetting this register allows the user to establish specific count interrupt thresholds. The counter will count "up" to saturation from the preset value, and may be read at any time. Counter increments occur at all times and are not disabled by resync.

Figure 20 : ECR : Error Count Register.

(MSB) (LSB)

OOFD3	OOFD2	OOFD1	OOFD0	ESFD3	ESFD2	ESFD1	ESFD0
OFF COUNT				ESF ERROR COUNT			

Symbol	Position	Name and Description
OOFD3	ECR.7	MSB of OOF event count.
OOFD3	ECR.4	LSB of OOF event count
ESFD3	ECR.3	MSD of extended superframe error count.
ESFD0	ECR.0	LSB of extended superframe error count.

These separate 4-bit binary up counters saturate at a count of 15 and will generate an interrupt for each occurence of an OOF event or an ESF error event after saturation (RIMR.6 = 1). Presetting these counters allows the user to establish specific count interrupt thresholds. The counters will count "up" to saturation from the preset value, and may be read at any time. These counters share the same register address, and must be written to or read from simultaneously.

The OOF counter records out-of-frame events in both 193S and 193E. The ESF error counter records errored superframes in 193E. In 193S the ESF counter records individual FT and FS errors when RCR.3 = 1. FT errors only RCR.3 = 0. ECR counter increments are disabled when resync is in progress (RLOS high).

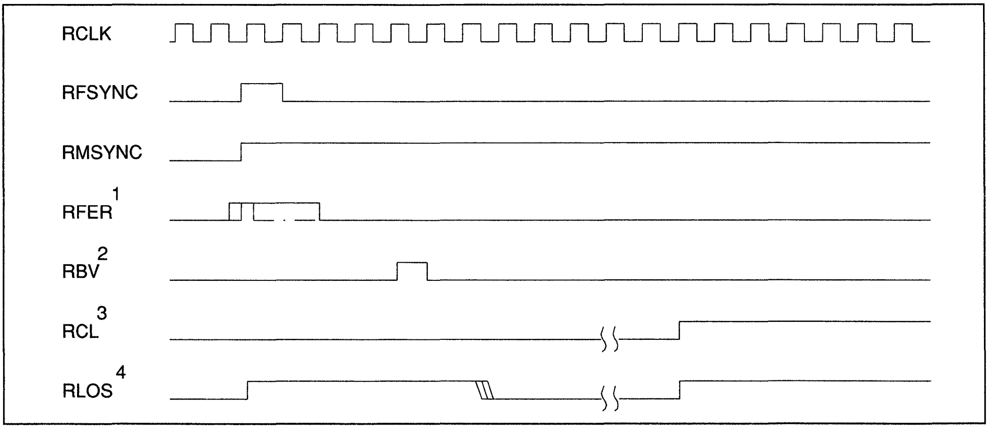
RYEL OUTPUT

The yellow alarm output transitions high when a yellow alarm is detected. A high-low transition indicates the alarm condition has been cleared. The RYEL bit (RSR.5) is a "latched" version of the RYEL output. In 193E mode, the type of yellow alarm detected is determined by CCR.5. If CCR.5 is cleared, a yellow alarm is 16 sets of 00 (hex) and FF (hex) received at RLINK. If CCR.5 is set, a yellow alarm is 256 or more consecutive channels which have a "0" in bit position 2. In 193S mode, the type of yellow alarm detected is determined by CCR.3. If CCR.3 is cleared, a yellow alarm is 256 or more consecutive channels which have a "0" in bit position 2.If CCR.3 is set, a yellow alarm is a "1" in the S-bit position of frame 12.

RBV OUTPUT

The bipolar violation output transitions high when accused bit emerges at RSER. RBV will go low at the next bit time if no additional violations are detected.

Figure 21 : Alarm Output Timing.



Notes : 1. RFER transistions high during F-bit time if received framing pattern bit is in error. (Frame 12 F-bits in 193S are ignored if CRC.3 = 1). Also, in 193E, RFER transitions 1/2 bittime before the rising edge of RMSYNC to indicate a crc error for the previous multiframe.
2. RBV indicates received bipolar violation and transistions high when accused bit emerges from RSER. If B8ZS is enabled, RBV will not report the zero replacement code.
3. RCL transitions high (during 32nd bit time) when 32 consecutive "0's" are received : RCL transitions low when the next "1" is received.
4. RLOS transitions high duri ng the F-bit time that caused an OOF event (any 2 of 4 consecutive Find or FPS bits are i n error) if autoresync mode is selected (RCR.1 = 0). Resync will also occur when loss of carrier is detected (RCL = 1). When RCR. = 1, RLOS remains low until resync occurs, regardless of OOF or carrier loss flags. In this situation, resync is initiated only when RCR.0 transitions low-to-high or the RST pin transitions high-low-high.

RFER OUTPUT

The receive frame error output transitions high at the F-bit time and is held high for two bit periods when a frame bit error occurs. In 193S framing FT and FS patterns are tested. The FPS pattern is tested in the 193E framing. Additionally, in the 193E framing, RFER reports a CRC error by a low-high transition (one bit period wide) one half RCLK period before a low-high transition on RMSYNC.

RESET

A high-low transition on RST clears all registers and forces immediate receive resync when RST returns high. This reset has no effect on transmit frame, multiframe, or channel counters. RST must be held low on system power-up to insure proper initialization of transceiver counters and registers. Following reset, the host processor should restore all control modes by writing appropriate registers with control data.

Table 6 : Hardware Mode.

Pin Number	Register Bit Location	Name and Description
14 (INT)	TCR-Bit 2	**193S mode S-bit Insertion**[3]. 1 = External ; 0 = Internal (No effect in 193E mode)
15 (SDI)	CCR-Bit 4	**Framing Mode Select.** 1 = 193E ; 0 = 193S
16 (SDO)	TCR-Bit 0	**Transmit Yellow Alarm**[2,3]. 1 = Enabled ; 0 = Disabled
17 (CS)	CCR-Bit 1	**Zero Suppression**[1]. 1 = Bit 7 stuffing ; 0 = Transparent
18 (SCLK)	CCR-Bit 2	**B8ZS**[1]. 1 = Enabled ; 0 = Disabled

Notes : 1. Tying pins 17 and 18 high enables loopback in hardware mode.
2. Bit 2 (193S) and data link (193E) yellow alarms are supported.
3. S-bit yellow alarm (193S) is not internally supported ; however, the user may elect to insert external S-bits for alarm purposes.

HARWARE MODE

For preliminary system prototyping or applications which do not require the features offered by the serial port, the transceiver can be reconfigured by the SPS pin. Tying SPS to VSS disables the serial port, clears all internal registers except CCR and TCR and redefines pins 14 through 18 as mode control inputs. The hardware mode allows device retrofit into existing applications where mode control and alarm conditioning hardware is often designed with discrete logic.

HARDWARE COMMON CONTROL

In the hardware mode bits TCR.2, CCR.4, TCR.0, CCR.1 and CCR.2 map to pins 14 through 18. The loopback feature (bit CCR.0) is enabled by tying pins 17 (zero suppression) and 18 (B8ZS) to 1. (The last states of pins 17 and 18 are latched when both pins are taken high, preserving the current zero suppression mode). Robbed bit signaling (bit TCR.4) is enabled for all channels. The user may tie TSER to TABCD externally to disable signaling if so desired. Bit CCR.3 is forced to 0, which selects bit 2 yellow alarm in 193S framing. Contents of the RCR, as well as the remaining bit locations in the CCR and TCR, are cleared in the hardware mode. The RST input may be used to force immediate receiver resync, and has no effect on transmit.

T1 OVERVIEW

FRAMING STANDARDS

The MK2180A is compatible with the existing Bell System D4 framing standard described in ATT PUB 43801 and the new extended superframe format (ESF) as described in ATT C.B. #142. In this documentation, D4 framing is referred to as 193S, and ESF (also known as Fe) is referred to as 193E. Programmable features of the MK2180A allow support of other framing standards which are derivatives of 193E and 193S. The salient differences between the 193S and 193E formats are the number of frames per superframe and use of the F-bit position. In 193S, 12 frames make up a superframe ; in 193E, 24. A frame consists of 24 channels (timeslots) of 8-bit data preceded by an F-bit. Channel data is transmitted and received MSB first.

F - BITS

The use of the F-bit position in 193S is split between the terminal framing pattern (known as FT-bits) which provides frame alignment information, and the signaling framing pattern (known as FS-bits) which provides multiframe alignment information. In 193E framing, the F-bit position is shared by the framing pattern sequence (FPS), which provides frame and multiframe alignment information, a 4 kHz data link (facility data link) known as FDL, and CRC (cyclic redundancy check) bits. The FDL bits are used for control and maintenance (inserted by the user at TLINK) and the CRC bits are an indicator of link quality and may be monitored by the user to establish error performance.

SIGNALING

During frames 6 and 12 in 193S, A and B signaling information is inserted into the LSB of all channels transmitted. In 193E, A and B data is inserted into frames 6 and 12, and C and D data is inserted into frames 18 and 24. This allows a maximum of 4 signaling states to be transmitted per superframe in 193S and 16 states in 193E.

ALARMS

The MK2180A supports all alarm pattern generation and detection required in typical Bell System applications. These alarm modes are explained in ATT PUB 43801, ATT C.B. #142 and elsewhere in this document.

LINE CODING

T1 line data is transmitted in a bipolar alternative mark inversion line format ; ones are transmitted as alternating negative and postive pulses and zeros are simply the absence of pulses. This technique minimizes DC voltage on the T1 span and allows clock to be extracted from data. The network currently has a one's density constraint to keep clock extraction circuitry functioning, which is usually met by forcing bit 7 of any channel consisting of all 0's to 1. The use of Bipolar Eight Zero Substitution (B8ZS) also satisfies the one's density requirement, while allowing data traffic to be transmitted without corruption. This feature is known as clear channel and is explained more completely in ATT C.B. #144. When the B8ZS feature is enabled, any outgoing stream of eight consecutive zeros is replaced with a B8ZS code word. If the last "one" transmitted was positive, the inserted code is 000+ - 0 - + ; if negative, the code word inserted is 000- + 0 + -. Bipolar violations occur in the fourth and seventh bit positions, which are ignored by the MK2180A error monitoring logic when B8ZS is enabled. Any received B8ZS code word is replaced with all 0's if B8ZS is enabled. Also, the receive status register will report any occurrence of B8ZS code words to the host controller. This allows the user to monitor the link for upgrade to clear channel capability, and respond to it. The B8ZS monitoring feature works at all times and is independent of the state of CCR.2.

TRANSMIT SIDE OVERVIEW

The transmit side of the MK2180A is made up of 6 major functional blocks : timing and clock generation, data selector, bipolar coder, yellow alarm, F-bit data and CRC. The timing and clock generation circuit develops all on-board and output clocks to the system from inputs TCLK, TFSYNC, and TMSYNC. The yellow alarm circuitry generates mode dependent yellow alarms. The CRC block generates checksum results utilitzed in 193E framing. F-bit data provides mode dependent framing patterns and alows insertion of link or S-bit data externally. All of these blocks feed into the data selector, where under control of the CCR, TCR, TIR's and TTR's, the contents of the outgoing data stream are established by bit selection and insertion. The bipolar coder formats the output of the data selector to make it compatible with bipolar transmission techniques and inserts zero suppression codes. The bipolar coder also supports the on-board loopback feature. *Input to output delay of the transmitter is 10 TCLK cycles.*

Table 7 : 193E Framing Format.

Frame Number	**F-Bit Use**			**Bit Use In Each Channel**		**Signaling-Bit Use**		
	FPS[1]	**FDL[2]**	**CRC[3]**	**Data**	**Signaling[4,5]**	**2 State**	**4 State**	**16 State**
1		M		BITS 1-8				
2			C1	BITS 1-8				
3		M		BITS 1-8				
4	0			BITS 1-8				
5		M		BITS 1-8				
6			C2	BITS 1-7	**BIT 8**	A	A	A
7		M		BITS 1-8				
8	0			BITS 1-8				
9		M		BITS 1-8				
10			C3	BITS 1-8				
11		M		BITS 1-8				
12	1			BITS 1-7	**BIT 8**	A	B	B
13		M		BITS 1-8				
14			C4	BITS 1-8				
15		M		BITS 1-8				
16	0			BITS 1-8				
17		M		BITS 1-8				
18			C5	BITS 1-7	**BIT 8**	A	A	C
19		M		BITS 1-8				
20	1			BITS 1-8				
21		M		BITS 1-8				
22			C6	BITS 1-8				
23		M		BITS 1-8				
24	1			BITS 1-7	**BIT 8**	A	B	D

Notes : 1. FPS - Framing Pattern Sequence.
2. FDL - 4kHz Facility Data Link ; M = message bits.
3. CRC - Cycle Redundancy Check bits. The CRC code will be internally generated by the device when TCR.5 = 0. When TCR.5 = 1, externally supplied CRC data will be sampled at TSER during the F-time of frames 2, 6, 10, 14, 18, 22.
4. The user may program any individual channels clear, in which case Bit 8 will be used for data, not signaling.
5. Depending on applications, the user can support 2-state, 4-dstate, or 16-state signaling by the appropriate decodes of TMO, TSIGFR, TSIGSEL.

RECEIVE SIDE OVERVIEW

SYNCHRONIZER

The heart of the receiver is the synchronizer/sync monitor. This circuit serves two purposes : 1) monitoring the incoming data stream for loss of frame or multiframe alignment, and 2) searching for new frame alignment pattern when sync loss is detected. When sync loss is detected, the synchronizer begins an offline search for the new alignment ; all output timing signals remain at the old alignment with the exception of RSIGFR, which is forced low during resync. When one and only one candidate is qualified, the output timing will move to the new alignment at the beginning of the next multiframe. One frame later, RLOS will transition low, indicating valid sync and the resumption of the normal sync monitoring mode. Several bits in the RCR allow tailoring of the resync algorithm by the user. These bits are described below.

Table 8 : 193S Framing Format.

Frame Number	F-Bit Use		Bit Use In Each Channel		Signaling-Bit Use
	F_T[1]	F_S[2]	Data	Signaling[4]	
1	1		BITS 1-8		
2		0	BITS 1-8		
3	0		BITS 1-8		
4		0	BITS 1-8		
5	1		BITS 1-8		
6		1	BITS 1-7	**BIT 8**	A
7	0		BITS 1-8		
8		1	BITS 1-8		
9	1		BITS 1-8		
10		1	BITS 1-8		
11	0		BITS 1-8		
12		0[3]	BITS 1-7	**BIT 8**	B

Notes : 1. FT (terminal framing) bits provide frame aligment information.
2. FS (signaling frame) bits provide multiframe aligment information.
3. The S-bit in frame 12 may be used for yellow alarm transmission and detection in some applications.
4. The user may program any individual channels clear, in which case Bit 8 will be used for data, not signaling.

SYNC CRITERIA (RCR.2)

Bit RCR.2 determines the number of consecutive framing pattern bits to be qualitfied before SYNC is declared. If RCR.2 = 1, the aligorithm will validate 24 bits ; if RCR.2 = 0, 10 bits are validated. 24 bit testing results in superior false framing protection, while 10 bit testing minimizes reframe time (although in either case, the synchronizer will only establish resync when one and only one candidate is found).

RESYNC (RCR.0)

A zero-to-one transition of RCR.0 causes the synchronizer to search for the framing pattern sequence immediately, regardless of the internal sync status. In order to initiate another resync command, this bit must be cleared and then set again.

SYNC ENABLE (RCR.1)

When RCR.1 is cleared, the receiver will inititate automatic resync if any of the following events occur: 1) an OOF event (out-of-frame), or 2) carrier loss (32 consecutive 0's appear at RPOS and RNEG). An OOF event occurs any time that 2 of 4 FT or FPS bits are in error. When RCR.1 is set, the automatic resync circuitry is disabled ; in this case, resync can only be initated by setting RCR.0 to 1, or externally via low-high transition on RST. Note that using RST to initiate resync resets the receive output timing while RST is low ; use of RCR.1 does not affect output timing until the new alignment is located.

SYNC CRITERIA (RCR.3)

193E

Bit RCR.3 determines which sync algorithm is utilized when resync is in progress (RLOS = 1). In 193E framing, when RCR.3 = 0, the synchronizer will lock only to the FPS pattern and will move to the new frame and multiframe alignment after the framing candidate is qualified. RLOS will go low one frame after the move to the new alignment. When RCR.3 = 1, the new alignment is further tested by a CRC code match. RLOS will transition low after a CRC match occurs. If no CRC match occurs in three attempts (three multiframes), the algorithm will reset and a new search for the framing pattern begins. It takes 9 ms for the synchronizer to check the first CRC code after the new alignment has been loaded. Each additional CRC test takes 3 msec. Regardless of the state of RCR.3, if more than one candidate exists after about 24 milliseconds, the synchronizer will begin eliminating emulators by testing their CRC codes on-line in order to find the true framing candidate.

193S

In 193S framing, when RCR.3 = 1, the synchronizer will cross check the FT pattern with the FS pattern to help eliminate false framing candidates such as digital milliwatts. The FS patterns are compared to therepeatingpattern ...00111000111000...(00111 X0 if CCR.3-YELMD- is equal to a 1). In this mode, FT and FS patterns must be correctly identified by the synchronizer before sync is declared. Clearing RCR.3 causes the synchronizer to search for FT patterns (101010...) without cross-coupling the FS pattern. Frame sync will be established using the FT information, while multiframe sync will be established only ifvalid FS information is present. If no valid FS pattern is identified, the synchronizer will move to the FT alignment, RLOS will go low, and a false multiframe position may be indicated by RMSYNC. RFER will indicate when the received S-bit pattern does not match the assumed internal multiframe alignment. This mode will be used in applications where non-standard S-bit patterns exist, in such applications multiframe alignment information can be decoded externally by using the S-bits present at RLINK.

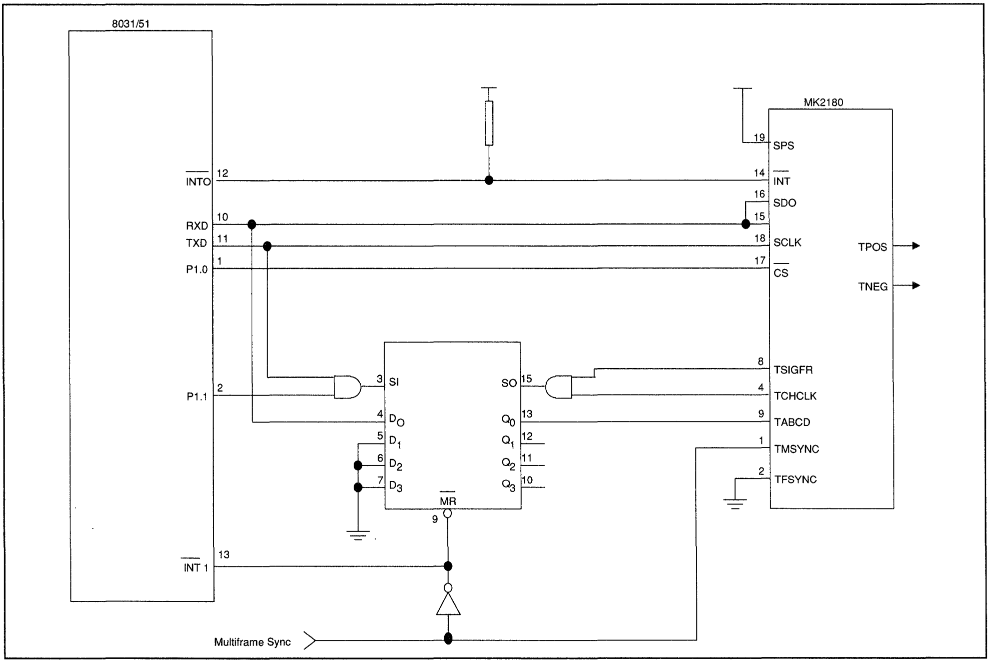
PROCESSOR-BASED SIGNALING

Many robbed bit signaling applications utilize a microprocessor to insert transmit signaling data into the outgoing data stream. The circuit shown in figure 22 "decouples" the processor timing from that of the MK2180A by use of a small FIFO memory. The processor writes to the FIFO (6 bytes are written : 3 for A data, 3 for B data) only when signaling updates are required. The FIFO automatically retransmits old data when no updates occur. The system is interrupt-driven from the transmit multiframe sync input ; the processor must update the FIFO prior to frame 6 (625 us) after interrupt to prevent data corruption. The application circuit shown supports 193S framing ; additional hardware is required for 193E applications.

Table 9 : Average Reframe Time[1].

Frame Mode	RCR.2 = 0			RCR.2 = 1			Unit
	Min.	AVG	Max.	Min.	AVG	Max.	
193S 193E	3.0 6.0	3.75 7.5	4.5 9.0	6.5 13.0	7.25 14.5	8.0 16.0	msec

Figure 22 : Processor-based Transmit Signaling Insertion.



ABSOLUTE MAXIMUM RATING*

Voltage on any Pin Relative to Ground	–0.5V to VDD + 0.5V
Operating Temperature	0˚C to 70˚C
Storage Temperature	–40˚C to 150˚C

* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

RECOMMENDED D.C. OPERATING CONDITIONS (0°C to 70°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{IH}	Logic 1	2.0		V_{DD} +.5	V
V_{IL}	Logic 0	–0.3		+0.8	V
V_{CC}	Supply	4.5	5.0	5.5	V

CAPACITANCE (t_A = 25°C)

Symbol	Parameter	Max.	Unit
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	7	pF

D.C. ELECTRICAL CHARACTERISTICS (0°C to 70°C VDD = 5V ±10%)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Notes
I_{DD}	Supply Current		3	10	mA	1,2
I_{IL}	Input Leakage			1	µA	
I_{LO}	Output Leakage			1	µA	3
V_{OH}	Output High Voltage @ - 1mA	2.4			V	4
V_{OL}	Output Low Voltage @ 4mA			0.4	V	5

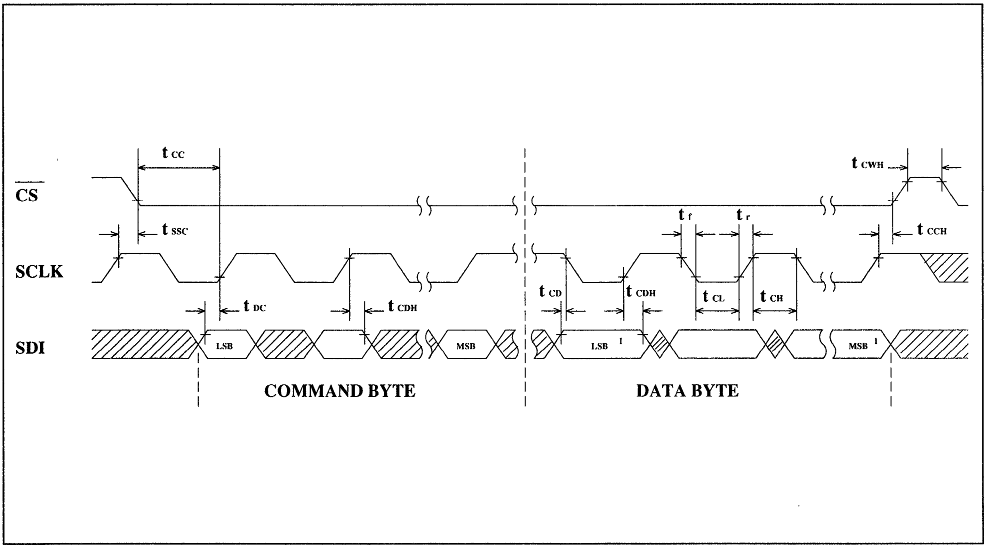
Notes : 1. TCLK = RCLK = 1.544MHz
2. Outputs open
3. Applies to SDO when tristated
4. All outputs except $\overline{\text{INT}}$, which is open collector.
5. All outputs.

A.C. ELECTRICAL CHARACTERISTICS[1] - SERIAL PORT (0°C to 70°C VCC = 5V ±10%)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{DC}	SDI Set up to SCLK Rising	50			ns
t_{CDH}	SDI Hold from SCLK Rising	50			ns
t_{CD}	SDI Set up to SCLK Falling (during data writes)	50			ns
$t_{CL.}$	SCLK Low Time	125			ns
t_{CH}	SCLK High Time	125			ns
t_R, t_F	SCLK Rise & Fall Time			500	ns
t_{CC}	$\overline{\text{CS}}$ Falling (set up) to SCLK Rising	50			ns
t_{CCH}	$\overline{\text{CS}}$ Rising (hold) from SCLK Rising	50			ns
t_{CWH}	$\overline{\text{CS}}$ High (inactive) Time	125			ns
t_{CDV}	SCLK Falling to SDO Valid[2]			100	ns
t_{CDZ}	$\overline{\text{CS}}$ to SDO High Z			75	ns
t_{SSC}	SCLK Set up to $\overline{\text{CS}}$ Falling	50			ns

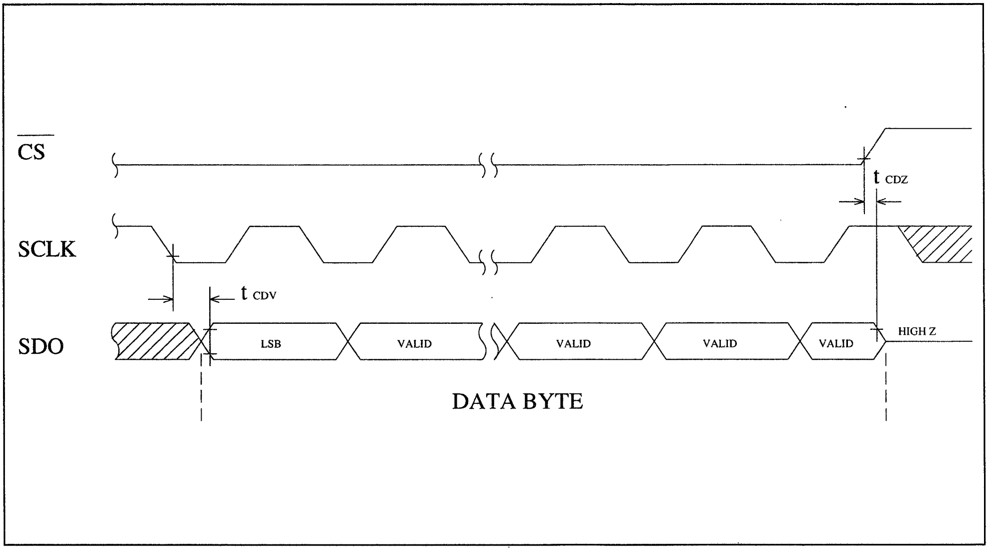
Notes : 1. Measured at V_{IH} = 2.0V, V_{IL} = .8V and 10ns maximum rise and fall time.
2. Output load capacitance = 100pF.

SERIAL PORT WRITE A.C. TIMING DIAGRAM



Notes : 1. Data byte bits must be valid across clock periods to prevent transients in operating modes.
2. Shaded regions indicated don't care states of input data.

SERIAL PORT READ^1 A.C. TIMING



Note : 1. Serial port write must precede a port read to provide address information.

A.C. ELECTRICAL CHARACTERISTICS[1] - TRANSMIT (0°C to 70°C, V_{CC} = 5V ± 10%)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_P	TCLK Period	250	648		ns
t_{WL}, t_{WH}	TCLK Pulse Width	125	324		ns
t_F, t_R	TCLK, RCLK Rise & Fall Time		20		ns
t_{STD}	TSER, TABCD, TLINK Set up to TCLK Falling	50			ns
t_{HTD}	TSER, TABCD, TLINK Hold from TCLK Falling	50			ns
t_{STS}	TFSYNC, TMSYNC Set Up to TCLK Rising	−125		125	ns
t_{PTS}	Propagation Delay[3] TCLK to TMO, TSIGEL, TSIGFR, TLCLK			100	ns
t_{PTCH}	Propagation Delay TCLK to TCHCLK			100	ns
t_{TSP}	TFSYNC, TMSYNC Pulse Width	100			ns
t_{PTD}	Propagation Delay TCLK to TPOS, TNEG			100	ns

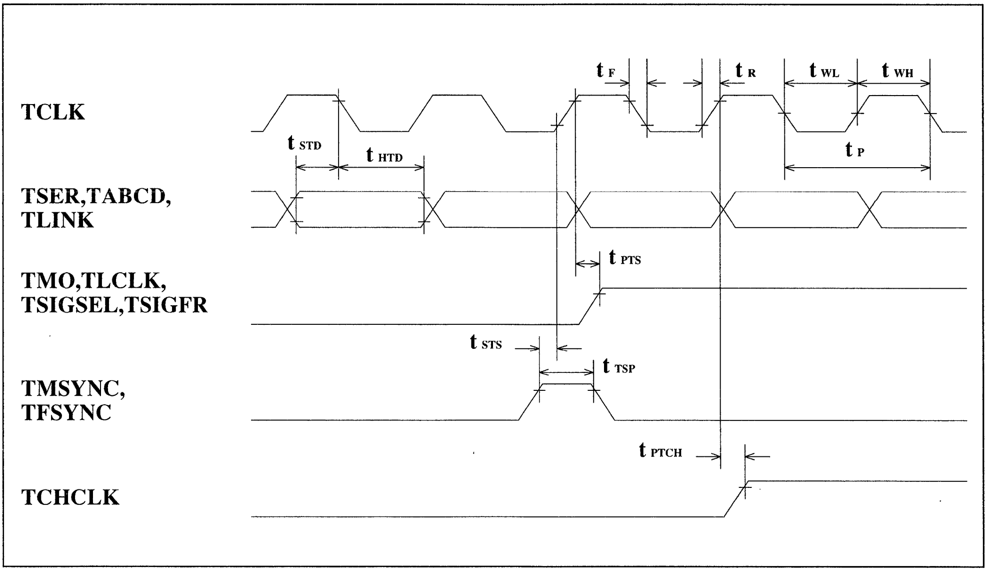
Notes : 1. Measured at V_{IH} = 2.0V, V_{IL} = .8V and 10ns maximum rise and fall time.
2. Output load capacitance = 100pF.
3. Propagation delay of these signals is measured from the rising edge of TMSYNC (instead of TCLK) if TMSYNC occurs after the rising edge of TCLK on a non-frame boundary. Maximum delay from rising edge of TMSYNC = 75ns.

A.C. ELECTRICAL CHARACTERISTICS[1] - RECEIVE (0°C to 70°C, V_{CC} = 5V ± 10%)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{PRS}	Propagation Delay RCLK to RMSYNC, RFSYNC, RSIGEL, RSIGFR, RLCLK, RCHCLK			100	ns
t_{PRD}	Propagation Delay RCLK to RSER, RABCD, RLINK			100	ns
t_{TTR}	Transition Time All Receive Outputs[2]			20	ns
t_P	RCLK Period	250	648		ns
t_{WL}, t_{WH}	RCLK Pulse Width	125	324		ns
t_R, t_F	RCLK Rise & Fall Times		20		ns
t_{SRD}	RPOS, RNEG Set Up to RCLK Falling	50			ns
t_{HRD}	RPOS, RNEG Hold to RCLK Falling	50			ns
t_{PRA}	Propagation Delay RCLK to RYEL, RCL, RFER, RLOS, RBV			100	ns
t_{RST}	Minimum $\overline{\text{RST}}$.Pulse Width on System Power Up or Restart	1			µs

Notes : 1. Measured at V_{IH} = 2.0V, V_{IL} = .8V and 10ns maximum rise and fall time.
2. Output load capacitance = 100pF.

TRANSMIT A.C. TIMING DIAGRAM



RECEIVE A.C. TIMING

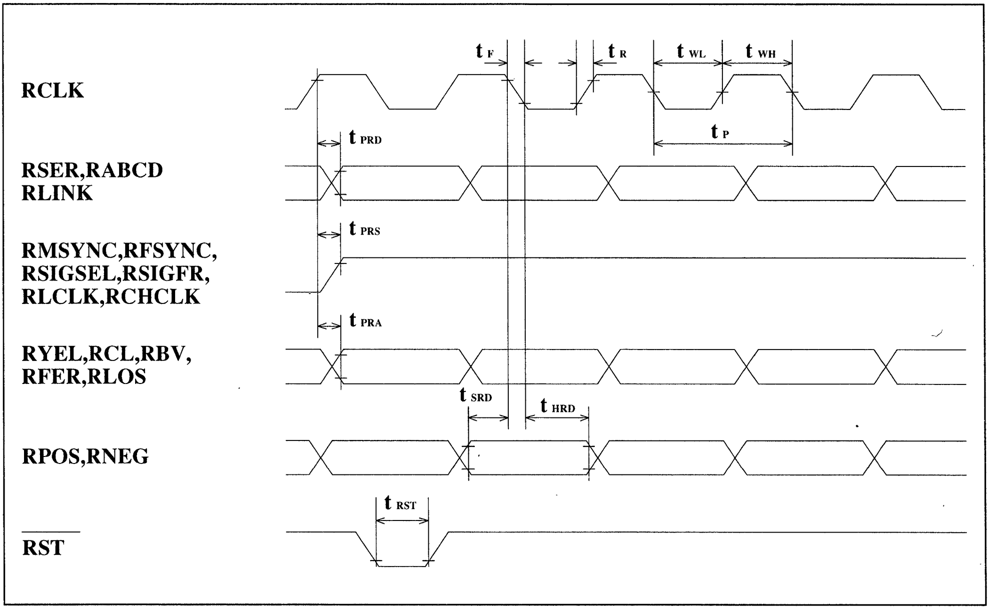


Figure 23 : Output Test Load.

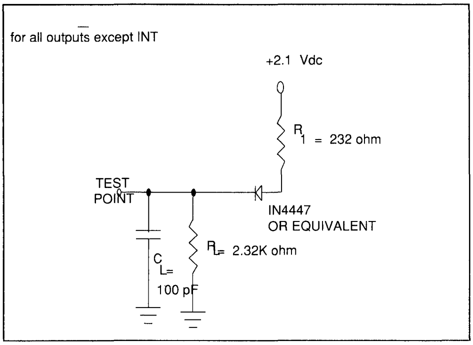
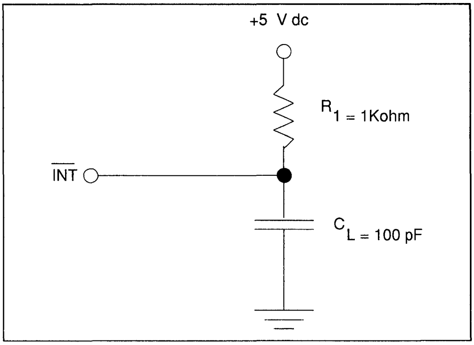
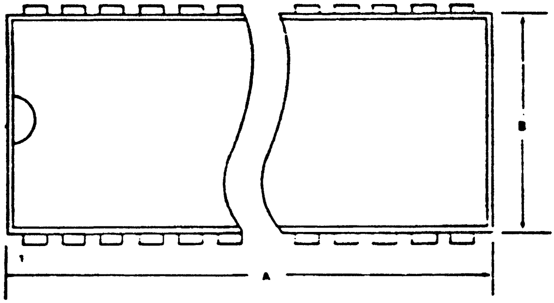


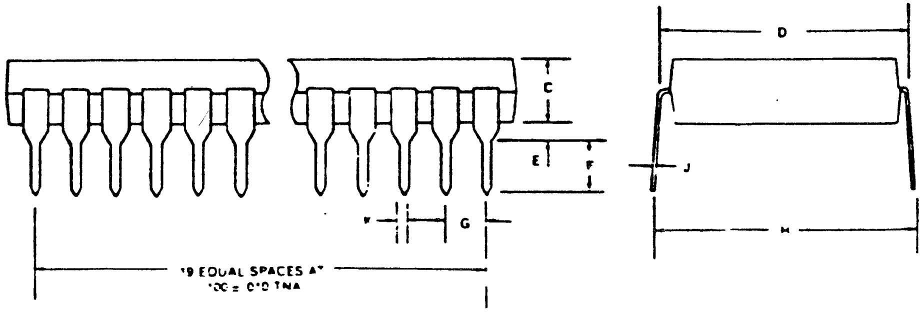
Figure 24 : INT Test Load.



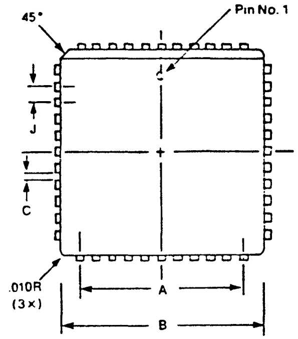
MK2180AN SERIAL T1 TRANSCEIVER 40 PIN PLASTIC DIP



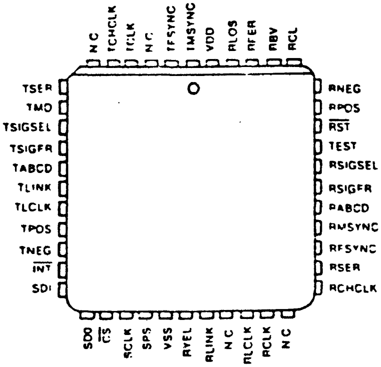
DIM.	INCHES	
	MIN.	MAX.
A	2.040	2.080
B	.540	.560
C	.140	.160
D	.590	.610
E	.020	.040
F	.110	.130
G	.090	.110
H	.600	.680
J	.008	.012
K	.015	.021

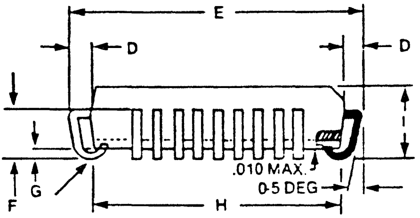


MK2180AQ SERIAL TRANSCEIVER 44 PIN PLCC



DIM.	INCHES	
	MIN.	MAX.
A	.490	.510
B	.590	.630
C	.020	.024
D	.018	.022
E	.688	.692
F	.118	.130
G	.020	.030
H	.590	.630
I	.167	.173
J	.048	.051





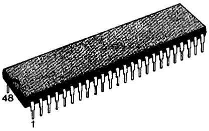
SERIAL COMMUNICATION CONTROLLER DEVICES

SGS-THOMSON
MICROELECTRONICS

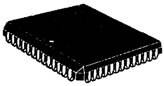
MK68564

SERIAL INPUT OUTPUT

- COMPATIBLE WITH MK68000 CPU
- COMPATIBLE WITH MK68000 SERIES DMA's
- TWO INDEPENDENT FULL-DUPLEX CHANNELS
- TWO INDEPENDENT BAUD-RATE GENERATORS
  - Crystal oscillator input
  - Single-phase TTL clock input
- DIRECTLY ADDRESSABLE REGISTERS (all control registers are read/write)
- DATA RATE IN SYNCHRONOUS OR ASYNCHRONOUS MODES
  - 0-1.25M bits/second with 5.0MHz system clock rate
- SELF-TEST CAPABILITY
- RECEIVE DATA REGISTERS ARE QUADRUPLY BUFFERED ; TRANSMIT REGISTERS ARE DOUBLY BUFFERED
- DAISY-CHAIN PRIORITY INTERRUPT LOGIC PROVIDES AUTOMATIC INTERRUPT VECTORING WITHOUT EXTERNAL LOGIC
- MODEM STATUS CAN BE MONITORED
  - Separate modem controls for each channel
- ASYNCHRONOUS FEATURES
  - 5, 6, 7, or 8 bits/character
  - 1, 1 1/2, or 2 stop bits
  - Even, odd, or no parity
  - x1, x16, x32, and x64 clock modes
  - Break generation and detection
  - Parity, overrun, and framing error detection
- BYTE SYNCHRONOUS FEATURES
  - Internal or external character synchronization
  - One or two sync characters in separate registers
  - Automatic sync character insertion
  - CDC-16 or CRC-CCITT block check generation and checking
- BIT SYNCHRONOUS FEATURES
  - Abort sequence generation and detection
  - Automatic zero insertion and deletion
  - Automatic flag insertion between messages
  - Address field recognition
  - I-field residue handling
  - Valid receive messages protected from overrun
  - CRC-16 or CRC-CCITT block check generation and checking



PDIP48
(Plastic Package)



PLCC52
(Chip Carrier)

DESCRIPTION

The MK68564 SIO (Serial Input Output) is a dual-channel, multi-function peripheral circuit, designed to satisfy a wide variety of serial data communications requirements in microcomputer systems. Its basic function is a serial-to-parallel, parallel-to-serial converter/controller ; however within that role, it is systems software configurable so that its "personality" may be optimized for any given serial data communications application.

The MK68564 is capable of handling asynchronous protocols, synchronous byte-oriented protocols (such as IBM Bisync), and synchronous bit-oriented protocols (such as HDLC and IBM SDLC). This versatile device can also be used to support virtually any serial protocol for applications other than data communications (cassette or floppy disk interface, for example).

The MK68564 can generate and check CRC codes in any synchronous mode and may be programmed to check data integrity in various modes. The device also has facilities for modem controls in each channel. In applications where these controls are not needed, the modem controls may be used for general-purpose I/O.

SIO PIN DESCRIPTION

GND :	Ground
V_{CC}:	+ 5 Volts (± 5%)
$\overline{\text{CS}}$:	Chip Select (input, active low). $\overline{\text{CS}}$ is used to select the MK68564 SIO for accesses to the internal registers. $\overline{\text{CS}}$ and $\overline{\text{IACK}}$ must not be asserted at the same time.
R/$\overline{\text{W}}$:	Read/write (input). R/$\overline{\text{W}}$ is the signal from the bus master, indicating wether the current bus cycle is a Read (high) or Write (low) cycle.
$\overline{\text{DTACK}}$:	Data Transfer Acknowledge (output, active low, three stateable). $\overline{\text{DTACK}}$ is used to signal the bus master that data is ready or that data has been accepted by the MK68564 SIO.
A1-A5 :	Address Bus (inputs). The address bus is used to select one of the internal registers during a read or write cycle.
D0-D7	Data Bus (bidirectional, threee-stateable). The data bus is used to transfer data to or from the internal registers during a read or write cycle. It is also used to pass a vector during an interrupt acknowledge cycle.
CLK :	Clock (input). This input is used to provide the internal timing for the MK68564 SIO.
$\overline{\text{RESET}}$:	Device Reset (input, active low). $\overline{\text{RESET}}$ disables both receivers and transmitters, forces TxDA and TxDB to a marking condition, forces the modem controls high and disables all interrupts. With the exception of the status registers, data registers, and the vector register, all internal registers are cleared. The vector register is reset to "0FH".
$\overline{\text{INTR}}$:	Interrupt Request (output, active low, open drain). $\overline{\text{INTR}}$ is asserted when the MK68564 SIO is requesting an interrupt. INTR is negated during an interrupt acknowledge cycle or by clearing the pending interrupt(s) through software.
$\overline{\text{IACK}}$:	Interrupt acknowledge (input, active low). $\overline{\text{IACK}}$ is used to signal the MK68564 SIO that the CPU is acknowledging an interrupt. $\overline{\text{CS}}$ and $\overline{\text{IACK}}$ must not be asserted at the same time.
$\overline{\text{IEI}}$:	Interrupt Enable In (input, active low). $\overline{\text{IEI}}$ is used to signal the MK68564 SIO that no higher priority device is requesting interrupt service.
$\overline{\text{IEO}}$:	Interrupt Enable Out (output, active low). $\overline{\text{IEO}}$ is used to signal lower priority peripherals that neither the MK68564 SIO nor another higher priority peripheral is requesting interrupt service.
XTAL1, XTAL2 :	Baud Rate Generator inputs. A crystal may be connected between XTAL1 and XTAL2, or XTAL1 may be driven with a TTL level clock. When using a crystal, external capacitors must be connectd. When driving XTAL1 with a TTL level clock, XTAL2 must be allowed to float.
$\overline{\text{RxRDYA}}$, $\overline{\text{RxRDYB}}$:	Receiver Ready (outputs, active low). Programmable DMA output for the receiver. The RxRDY pins pulse low when a character is available in the receive buffer.
$\overline{\text{TxRDYA}}$, $\overline{\text{TxRDYB}}$:	Transmitter Ready (outputs, active low). Programmable DMA output for the transmitter. The $\overline{\text{TxRDY}}$ pins pulse low when the transmit buffer is empty.
$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$:	Clear to Send (inputs, active low). If Tx Auto Enables is selected, these inputs enable the transmitter of their respective channels. If Tx Auto Enables is not selected, these inputs may be used as general purpose input pins. The inputs are Scmit-trigger buffered to allow slow rise-time input signals.
$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$:	Data Carrier Detect (inputs, active low). If Rx Auto Enables is selected, these inputs enable the receiver of their respective channels. If Rx Auto Enables is not selected, these inputs may be used as general purpose input pins. The inputs are Schmit-trigger buffered to allow slow rise-time input signals.
RxDA, RxDB :	Receive Data (inputs, active high). Serial data input to the receiver.
TxDA, TxDB :	Transmit Data (outputs, active high). Serial data output of the transmitter.

SIO PIN DESCRIPTION (continued)

RxCA, RxCB :	Receiver Clocks (input/output). Programmable pin, receive clock input, or baud rate generator output. The inputs are Schmit-trigger buffered to allow slow rise-time input signals.
TxCA, TxCB :	Transmitter Clocks (input/output). Programmable pin, transmit clock input, or baud rate generator output. The inputs are Schmit-trigger buffered to allow slow rise-time input signals.
RTSA, RTSB :	Request to Send (outputs, active low). These outputs follow the inverted state programmed into the RTS bit. When the RTS bit is reset in the asynchronous mode, the output will not change until the character in the transmitter is completely shifted out. These pins may be used as general purpose outputs.
$\overline{\text{DTRA}}$, $\overline{\text{DTRB}}$:	Data Terminal Ready (outputs, Active low). These outputs follow the inverted state programmed into the DTR bit. These pins may also be used as general purpose outputs.
$\overline{\text{SYNCA}}$, $\overline{\text{SYNCB}}$:	Synchronization (input/output, active low). The $\overline{\text{SYNC}}$ pin is an output when Monosync, Bisync, or SDLC mode is programmed. It is asserted when a sync/flag character is detected by the receiver. The SYNC pin is a general purpose input in the Asynchronous mode and an input to the receiver in the External Sync Mode.

Figure 1a : Dual In Line Pin Configuration.

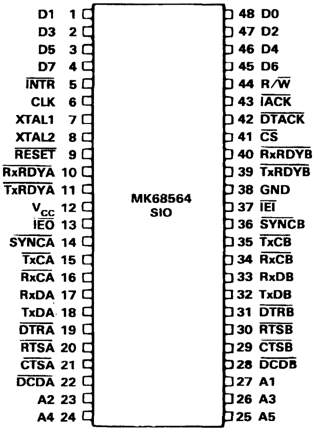
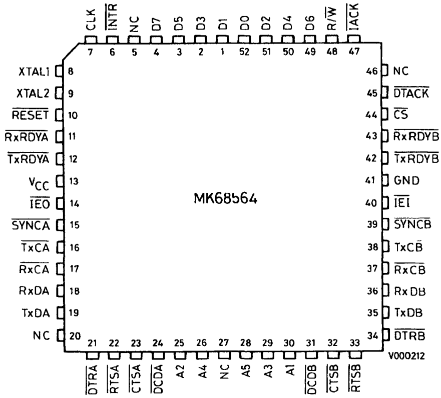


Figure 1b : Chip Carrier Pin Configuration.



NC = NO CONNECTION

SIO SYSTEM INTERFACE

INTRODUCTION

The MK68564 SIO is designed for simple and efficient interface to a MK68000 CPU system. All data transfers between the SIO and the CPU are asynchronous to the system clock. The SIO system timing is derived from the chip select input (CS) during normal read and write sequences, and from the interrupt acknowledge input (IACK) during an exception processing sequence. CS is a function of address decode and (normally) lower data strobe (LDS). IACK is a function of the interrupt level on address lines A1, A2, and A3, an interrupt acknowledge function code (FC0-FC2), and LDS.

Note : CS and IACK can never be asserted at the same time.

Note : Unused inputs should be pulled up or down, but never left floating.

READ SEQUENCE

The SIO will begin a read cycle if, on the falling edge of CS, the read-write (R/W) pin is high. The SIO will respond by decoding the address bus (A1-A5) for the register selected, by placing the contents of that register on the data bus pins (D0-D7), and by driving the data transfer acknowledge (DTACK) pin low. If the register selected is not implemented on the SIO, the data bus pins will be driven high, and then DTACK will be asserted. When the CPU has acquired the data, the CS signal is driven high, at which time the SIO will drive DTACK high and then three-state DTACK and D0-D7.

WRITE SEQUENCE

The SIO will begin a write cycle if, on the falling edge of CS, the R/W pin is low. The SIO will respond by latching the data bus, by decoding the address bus for the register selected, by loading the register with the contents of the data bus, and by driving DTACK low. When the CPU has finished the cycle, the CS input is driven high. At this time, the SIO will drive DTACK high and will then three-state DTACK. If the register selected is not implemented on the SIO, the normal write sequence will proceed, but the data bus contents will not be stored.

INTERRUPT SEQUENCE

The SIO is designed to operate as an independent, interrupting peripheral, or, when interconnected with other components, an interrupt priority daisy chain can be formed.

Independent Operation. Independent operation requires that the interrupt enable in pin (IEI) be connected to ground. The SIO starts the interrupt sequence by driving the interrupt request pin (INTR) low. The CPU responds to the interrupt by starting an interrupt acknowledge cycle, in which the SIO IACK pin is driven low. The highest priority interrupt request in the SIO, at the time IACK goes low, places its vector on the data bus pins. The SIO releases the INTR pin and drives DTACK low. When the CPU has acquired the vector, the IACK signal is driven high. The SIO responds by driving DTACK to a high level and then three-stating DTACK and D0-D7. If more than one interrupt request is pending at the start of an interrupt acknowledge sequence, the SIO will drive the INTR pin low following the completion of the interrupt acknowledge cycle. This sequence will continue until all pending interrupts are cleared. If the SIO is not requesting an interrupt when IACK goes low, the SIO will not respond to the IACK signal ; DTACK and the data bus will remain three-stated.

Daisy Chain Operation. The interrupt priority chain is formed by connecting the interrupt enable out pin (IEO) of a higher priority part to IEI of the next lower priority part. The highest priority part in the chain should have IEI tied to ground. The Daisy Chaining capability (figures 2 and 3) requires that all parts in a chain have a common IACK signal. When the common IACK goes low, all parts freeze and prioritize interrupts in parallel. Then priority is passed down the chain, via IEI and IEO, until a part which has a pending interrupt, once IEI goes low, passes a vector, does not propagate IEO, and generates DTACK.

The state of the IEI pin does not affect the SIO interrupt control logic. The SIO can generate an interrupt request any time its interrupts are enabled. The IEO pin is normally high ; it will only go low during an IACK cycle if IEI is low and no interrupt is pending in the SIO. The IEO pin will be forced high whenever IACK or IEI goes high.

Figure 2 : Conceptual Circuit of the MK68564 SIO Daisy Chaining Logic.

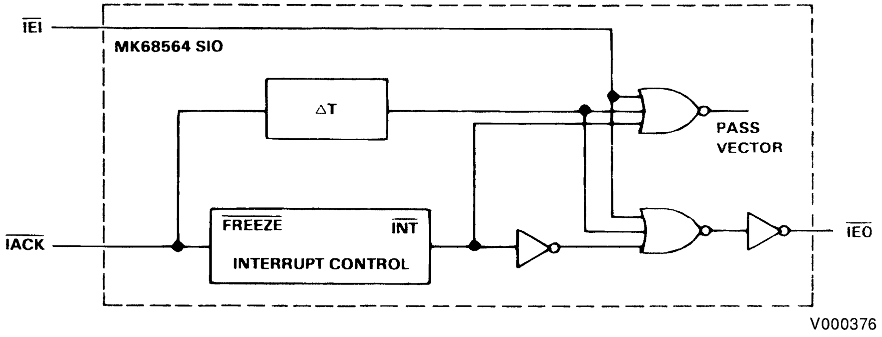


Figure 3 : Daisy Chaining.

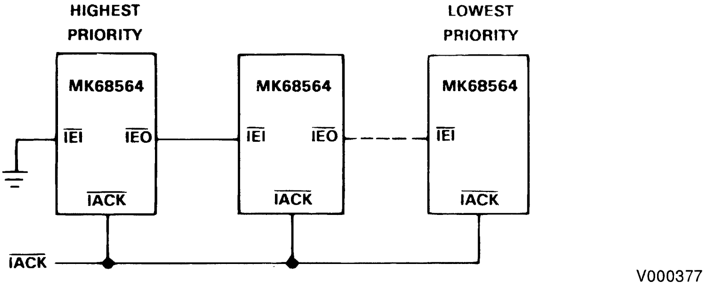
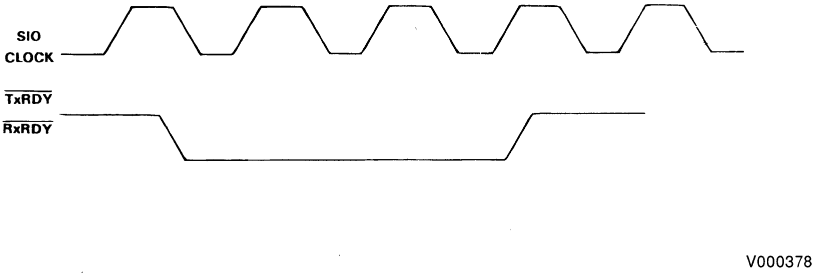


Figure 4 : DMA Interface Timing.



DMA INTERFACE

The SIO is designed to interface to the 68000 family DMA's as a 68000 compatible device, using the cycle steal mode. The SIO provides four outputs (TxRDYA, RxRDYA, TxRDYB, RxRDYB) for requesting service from the DMA. The SIO issues a request for service by pulsing the RDY pin low for three clock (CLK) cycles (see figure 4). TxRDY (when enabled) will be active when the transmit buffer becomes empty. RxRDY (when enabled) will be active when a character is available in the receive buffer. If Receive Interrupt On First Character Only is enabled during a DMA operation and a special receive condition is detected, the RxRDY pin will not become active. Instead, a special receive condition interrupt will be generated by the channel.

RESET

There are two ways of resetting the SIO : an individual, programmable channel reset and an external hardware reset.

The individual channel reset is generated by writing "18H" to the Command Register for the channel selected. All outputs associated with the channel are reset high, TxC and RxC are inputs, SYNC is an output, and TxD is forced marking. All R/W registers for the channel are reset to "00H", except the vector register and the data register, which are not affected.

Read only status register 1 is reset to "01H" (All Sent set). Break/Abort, Interrupt Pending, and Rx Character Available bits in read only status register 0 are reset ; Underrun/EOM, Hunt/Sync, and Tx Buffer Empty are set ; CTS and DCD bits are set to the inverted state of their respective input pins. Any interrupts pending for the channel are reset (any pending interrupts in the other channel will not be affected).

An external hardware reset occurs when the RESET pin is driven low for at least one clock (CLK) cycle. Both channels are reset as listed above, and the vector register is reset to "0FH".

ARCHITECTURE

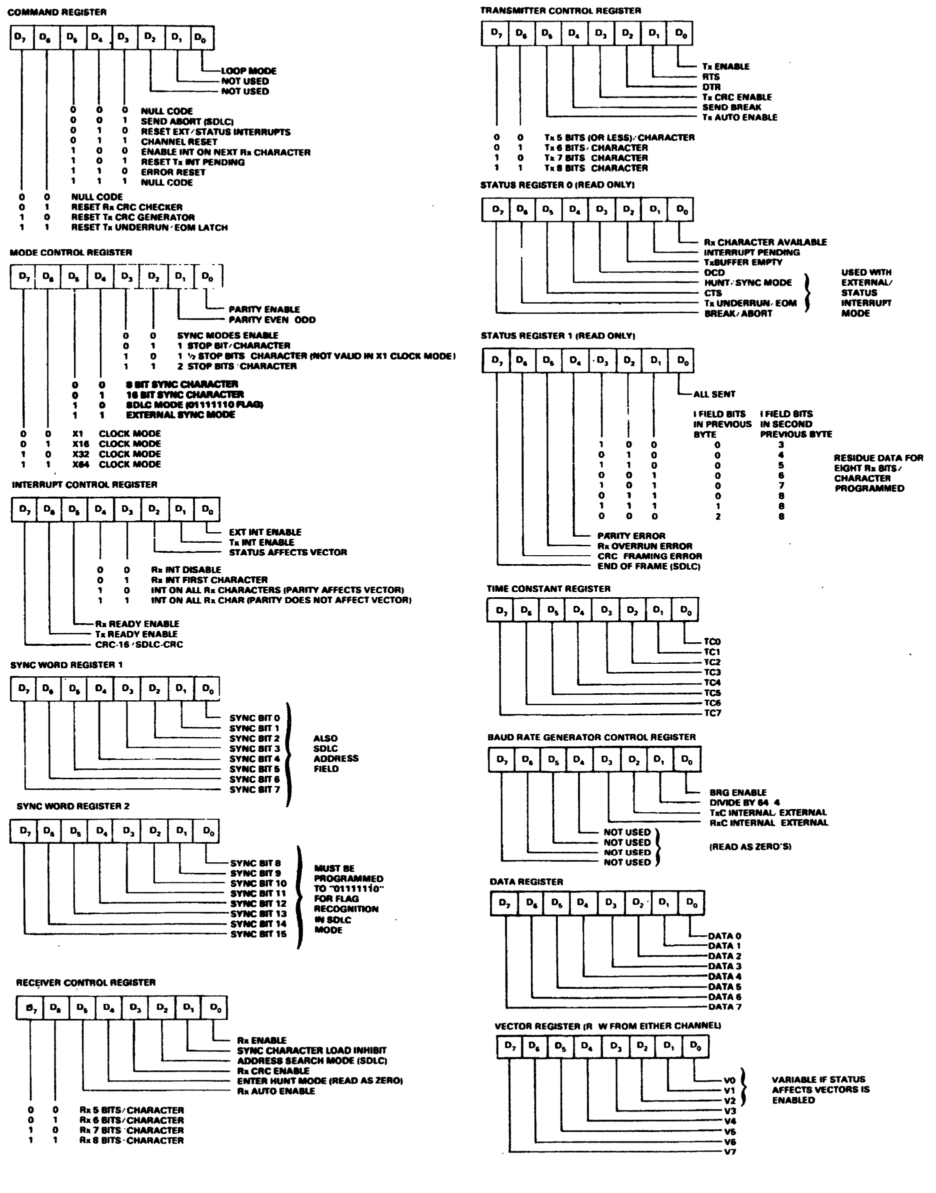
The MK68564 SIO contains two independent, full-duplex channels. Each channel contains a transmitter, receiver, modem control logic, interrupt control logic, a baud rate generator, ten Read/Write registers, and two read only status registers. Each channel can communicate with the bus master using polling, interrupts, DMA, or any combination of these three techniques. Each channel also has the ability to connect the transmitter output into the receiver without disturbing any external hardware.

Register Set. The register set is the heart of each channel. A channel is configured for different communication protocols and interface options by programming the registers. Table 1 lists all the registers available in the SIO and their addresses.

Data Register. The Data Register is composed of two separate registers : a write only register, which is the Transmit Buffer, and a read only register, which is the Receive Buffer. The Receive Buffer is also the top register of a three register stack called the receive data FIFO.

Vector Register. The Vector Register is different from the other 24 registers, because it may be accessed through either Channel A or Channel B during a R/W cycle. During an Interrupt Acknowledge cycle, the contents of the Vector Register are passed to the CPU to be used as a pointer to an interrupt service routine. If the Status Affects Vector bit is Low in the Interrupt Control Register, any data written to the Vector Register will be returned unmodified during a Read Cycle or an IACK cycle. If the Status Affects Vector bit is High, the lower three bits of the vector returned during a Read or IACK cycle are modified to reflect the highest priority interrupt pending in the SIO at that time. The upper five bits written to the Vector Register are unaffected. After a hardware reset only, this register contains a "0FH" value, which is the MK68000's uninitialized interrupt vector assignment.

Figure 5 : Register Bit Functions.



SIO INTERNAL REGISTERS

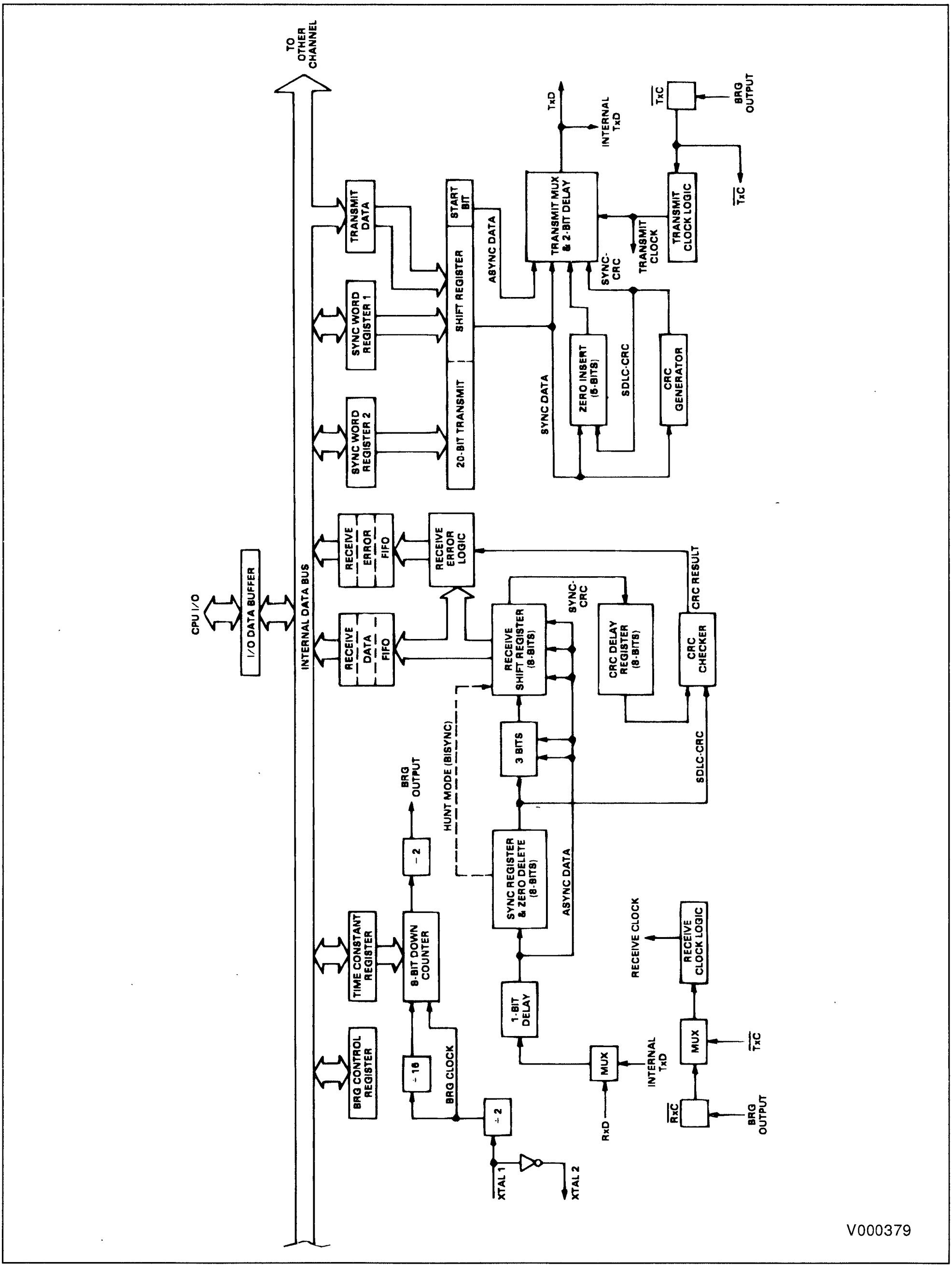
The MK68564 SIO has 25 internal registers. Each channel has ten R/W registers and two read only registers associated with it. The vector register may be accessed through either channel.

Table 1 : Register Map.

Address					Abbreviation	Channel	Register Name	Access	
5	4	3	2	1				Read/ write	Read Only
0	0	0	0	0	CMDREG	A	Command Register	X	
0	0	0	0	1	MODECTL	A	Mode Control Register	X	
0	0	0	1	0	INTCTL	A	Interrupt Control Register	X	
0	0	0	1	1	SYNC 1	A	Sync Word Register 1	X	
0	0	1	0	0	SYNC 2	A	Sync Word Register 2	X	
0	0	1	0	1	RCVCTL	A	Receiver Control Register	X	
0	0	1	1	0	XMTCTL	A	Transmitter Control Register	X	
0	0	1	1	1	STAT 0	A	Status Register 0		X
0	1	0	0	0	STAT 1	A	Status Register 1		X
0	1	0	0	1	DATARG	A	Data Register	X	
0	1	0	1	0	TCREG	A	Time Constant Register	X	
0	1	0	1	1	BRGCTL	A	Baud Rate Generator Control Reg	X	
0	1	1	0	0	VECTRG	A/B	Interrupt Vector Register (note 2)	X	
0	1	1	0	1		A	(note 1)	X	
0	1	1	1	0		A	(note 1)	X	
0	1	1	1	1		A	(note 1)	X	
1	0	0	0	0	CMDREG	B	Command Register	X	
1	0	0	0	1	MODECTL	B	Mode Control Register	X	
1	0	0	1	0	INTCTL	B	Interrupt Control Register	X	
1	0	0	1	1	SYNC 1	B	Sync Word Register 1	X	
1	0	1	0	0	SYNC 2	B	Sync Word Register 2	X	
1	0	1	0	1	RCVCTL	B	Receiver Control Register	X	
1	0	1	1	0	XMTCTL	B	Transmitter Control Register	X	
1	0	1	1	1	STAT 0	B	Status Register 0		X
1	1	0	0	0	STAT 1	B	Status Register 1		X
1	1	0	0	1	DATARG	B	Data Register	X	
1	1	0	1	0	TCREG	B	Time Constant Register	X	
1	1	0	1	1	BRGCTL	B	Baud Rate Generator Control Reg	X	
1	1	1	0	0	VECTRG	A/B	Interrupt Vector Register (note 2)	X	
1	1	1	0	1		B	(note 1)	X	
1	1	1	1	0		B	(note 1)	X	
1	1	1	1	1		B	(note 1)	X	

Notes : 1. Not Used, Read as "FFH".
2. Only One Vector Register, Accessible through Either Channel.

Figure 6 : Transmit and Receive Data Paths.



DATA PATH

The transmit and receive data paths for each channel are shown in figure 6. The receiver has three 8-bit buffer registers in a FIFO arrangement (to provide a 3-byte delay) in addition to the 8-bit receive shift register. This arrangement creates additional time for the CPU to service an interrupt at the beginning of a block of high-speed data. The receiver error FIFO stores parity and framing errors and other types of status information for each of the three bytes in the receive data FIFO. The receive error FIFO is loaded at the same time as the receive data FIFO. The contents of the receive error are read through the upper four bits in Status Register 1.

Incoming data is routed through one of several paths, depending on the mode and character length. In the Asynchronous modes, serial data is entered into the 3-bit buffer, if it has a character length of seven or eight bits, or the data is entered into the 8-bit receive shift register, if it has a length of five or six bits.

In the Synchronous mode, the data path is determined by the phase of the receive process currently in operation. A Synchronous Receive operation begins with the receiver in the Hunt phase, during which time the receiver searches the incoming data stream for a bit pattern that matches the preprogrammed sync characters (or flags in the SDLC mode). If the device is programmed for Monosync Hunt, a match is made with a single sync character stored in Sync Word Register 2. In Bisync Hunt, a match is made with the dual sync characters stored in Sync Word Registers 1 and 2. In either case, the incoming data passes through the receive sync register and is compared against the programmed sync characters in Sync Word Registers 1 and 2.

In the Monosync mode, a match between the sync character programmed into Sync Word Register 2 and the character assembled in the receive sync register establishes synchronization.

In the Bysync mode, incoming data is shifted to the receive shift register, while the next eight bits of the message are assembled in the receive sync register. The match between the assembled character in the sync register and the programmed character in Sync Word Register 2, and between the character in the shift register and the programmed character in Sync Word Register 1 establishes synchronization. Once synchronization is established, incoming data bypasses the receive sync register and directly enters the 3-bit buffer.

In the SDLC mode, all incoming data passes through the receive sync register, which continuously monitors the receive data stream and performs zero deletion when indicated. Upon receiving five contiguous ones, the sixth bit is inspected. If the sixth bit is a 0, it is deleted from the data stream. If the sixth bit is a 1, the seventh bit is inspected. If the seventh bit is a 0, a Flag sequence has been received ; if the seventh bit is a 1, an Abort sequence has been received.

The reformatted data from the receive sync register enters the 3-bit buffer and is transferred to the receive shift register. Note that the SDLC receive operation also begins in the Hunt Phase, during which time the SIO tries to match the assembled character in the receive sync register with the flag pattern in Sync Word Register 2. Once the first flag character is recognized, all subsequent data is routed through the path described above, regardless of character length.

Although the same CRC checker is used for both SDLC and synchronous data, the path taken for each mode is different. In Bisync protocol, the byte-oriented operation requires that the CPU decide whether or not to include the data character in the CRC calculation. To allow the CPU ample time to make this decision, the SIO provides an 8-bit delay before the data enters the CRC checker. In the SDLC mode, no delay is provided, since CRC is calculated on all data between the opening and closing flags.

The transmitter has an 8-bit transmit data register, which is loaded from the internal bus, and a 20-bit transmit shift register, which can be loaded from Sync Word Register 1, Sync Word Register 2, and the transmit data register. Sync Word Registers 1 and 2 contain sync characters in the Monosync, Bisync, or External Sync modes, or address field (one character long) and flag, respectively, in the SDLC mode. During Synchronous modes, information contained in Sync Word Registers 1 and 2 is loaded into the transmit shift register at the beginning of the message and, as a time filler, in the middle of the message if a Transmit Underrun condition occurs. In SDLC mode, the flags are loaded into the transmit shift register at the beginning and end of the message.

Asynchronous data in the transmit shift register is formatted with start and stop bits, and it is shifted out to the transmit multiplexer at the selected clock rate.

Synchronous (Monosync, Bisync, or External Sync) data is shifted out to the transmit multiplexer and also the CRC generator at the x1 clock rate.

SDLC/HDLC data is shifted out through the zero insertion logic, which is disabled while flags are being sent. For all other fields (address, control, and frame check), a 0 is inserted following five contiguous ones in the data stream. Note that the CRC generator result (frame check) for SDLC data is also routed through the zero insertion logic.

I/O CAPABILITIES

The SIO offers the choice of Polling, Interrupt (vectored or non-vectored), and DMA Transfer modes to transfer data, status, and control information to and from the CPU or other bus master.

Polling. The Polled mode avoids interrupts. Status Registers 0 and 1 are updated at appropriate times for each function being performed (for example, CRC Error status valid at the end of the message). All the interrupt modes of the SIO must be disabled to operate the device in a polled environment.

While in its Polling sequence, the CPU examines the status contained in Status Register 0 for each channel. The state of the status bits in Status Register 0 serves as an acknowledge to the Poll inquiry. Status bits D0 and D2 indicate that a receive or transmit data transfer is needed. The rest of the status bits in Status Register 0 indicate special status conditions. The receiver error condition bits in Status Register 1 do not have to be read until the Rx Character Available status bit in Status Register 0 is set to a one.

Interrupts. The SIO offers an elaborate interrupt scheme to provide fast interrupt response in realtime applications. The interrupt vector points to an interrupt service routine in the memory. To service operations in both channels and to eliminate the necessity of writing a status analysis routine (as required for a polling scheme), the SIO can modify the interrupt vector so it points to one of eight interrupt service routines. This is done under program control by setting the Status Affects Vector bit in the Interrupt Control Register of channel A or channel B, to a one. When this bit is set, the interrupt vector is modified according to the assigned priority of the various interrupting conditions.

Note : If the Status Affects Vector bit is set in either channel, the vector is modified for both channels. This is the only control bit that operates in this manner in the SIO.

Transmit interrupts, Receive interrupts, and External/Status interrupts are the sources of interrupts. Each interrupt source is enabled under program control with Channel A having a higher priority than Channel B, and with Receiver, Transmitter, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted by the transmit buffer becoming empty. This implies that the transmitter must have had a data character written into it so t can become empty. When enabled, the receiver can interrupt the CPU in one of three ways :

Interrupt On First Character Only

Interrupt On All Receive Characters

Interrupt On A Special Receive Condition.

Interrupt On First Character Only.This mode is normally used to start a software Polling loop or a DMA transfer routine using the RxRDY pin. In this mode, the SIO generates an interrupt on the first character received after this mode is selected and, thereafter, only generates an interrupt if a Special Receive Condition occurs. The Special Receive Conditions that can cause an interrupt in this mode are : Rx Overrun Error, Framing Error (in Asynchronous modes), and End Of Frame (in SDLC mode). This mode is reinitialized by the Enable Interrupt On Next Rx Character command. If a Special Receive Condition interrupt occurs in this interrupt mode, the data with the special condition is held in the receive data FIFO until an Error Reset Command is issued.

Interrupt On All Receive Characters. In this mode, an interrupt is generated whenever the receive data FIFO contains a character or a Special Receive Condition occurs. The Special Receive Conditions that can cause an interrupt in this mode are : Rx Overrun Error, Framing Error (in Asynchronous modes), End of Frame (in SDLC mode), and Parity Error (if selected).

Interrupt On A Special Receive Condition. The Special Receive Condition interrupt is not, as such, a separate interrupt mode. Before a Special Receive Condition can cause an interrupt, either the Interrupt On First Character Only or Interrupt On All Receive Characters mode must be selected. The Special Receive Condition interrupt will modify the receive interrupt vector if Status Affects Vector is enabled. The Special Receive Condition status is displayed in the upper four bits of Status Register 1. Two of the conditions causing a special receive interrupt are latched when they occur ; they are : Parity Error and Rx Overrun Error. These status bits may only be reset by an Error Reset command. When either of these conditions occur, a read of Status Register 1 will reflect any errors in the current word in the receive buffer plus any parity or overrun errors since the last Error Reset command was issued.

External/Status Interrupts. The main function of the External/Status interrupt is to monitor the signal transitions of the CTS, DCD, and SYNC pins ; however, an External/Status interrupt is also caused by a Transmit Underrun condition or by the detection of a Break (Asynchronous mode) or Abort

(SDLC mode) sequence in the received data stream. When any one of the above conditions occur, the external/status logic latches the current state of all five input conditions, and generates an interrupt. To reinitialize the external/status logic to detect another transition, a Reset External/Status Interrupts command must be issued. The Break/Abort condition allows the SIO to generate an interrupt when the Break/Abort sequence is detected and terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Break/Abort condition in external logic.

DMA Transfer

The SIO provides two output signals per channel for connection to a DMA controller ; they are $\overline{\text{TxRDY}}$ and $\overline{\text{RxRDY}}$. The outputs are enabled under software control by writing to the Interrupt Control Register. Both outputs will pulse Low for three system clock cycles when their input conditions are active. $\overline{\text{TxRDY}}$ will be active when the Transmit Buffer becomes empty. $\overline{\text{RxRDY}}$ will be active when a character is available in the Receive Buffer. If a Special Receive Condition occurs when Interrupt On First Character Only mode is selected, a receiver interrupt will be generated and $\overline{\text{RxRDY}}$ will not become active. This will automatically inform the CPU of a discrepancy in the data transfer.

SELF TEST

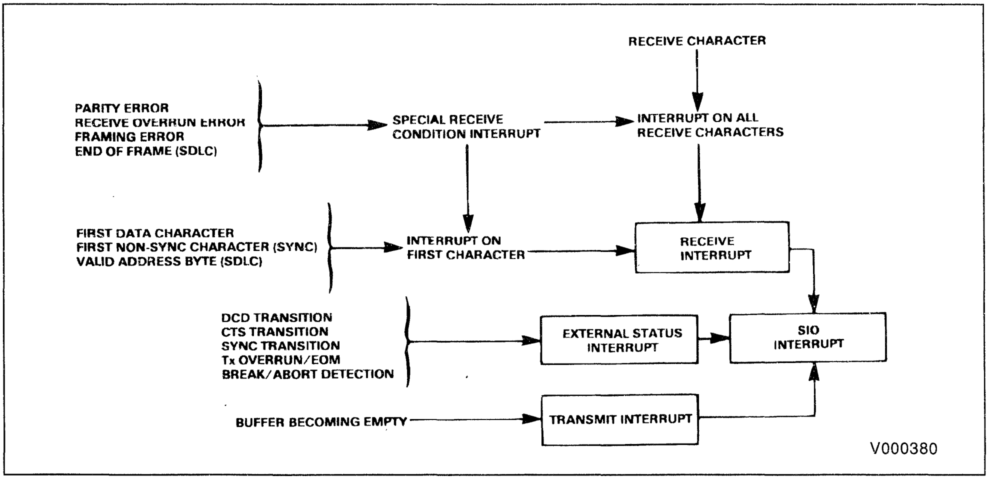
When the Loop Mode bit is set in the Command Register, the receiver shift clock input pin ($\overline{\text{RxC}}$) and the receiver data input pin (RxD) are electrically disconnected from the internal logic. The transmit data output pin (TxD) is connected to the internal receiver data logic, and the transmit shift clock pin (TxC) is connected to the internal receiver shift clock logic. All other features of the SIO are unaffected.

BAUD RATE GENERATORS

Each channel in the SIO contains a programmable baud rate generator (BRG). Each BRG consists of an 8-bit time constant register, an 8-bit down counter, a control register, and a flip-flop on the output to provide a square wave signal out. In addition to the flip-flop on the output, there is also a flip-flop on the input clock ; therefore, the maximum output frequency of the BRG is one-forth of the input clock frequency. This maximum output frequency occurs when divide by four mode is selected, and the time constant register is loaded with the minimum count of "01H". The equation to determine the output frequency is :

$$\text{Output Frequency} = \frac{\text{Input Frequency}}{\text{(divide by selected) X (time constant value in decimal)}}$$

Figure 7 : Interrupt Structure.



For example, when the time constant register is loaded with "01H" and divide by four is selected, one output clock will occur for every four input clocks. If the time constant value loaded is "00H" (256 decimal) instead of "01H" and divide by 64 is selected, one output clock will occur for every 16384 input clocks. Note that the minimum count value is "01H" (1 decimal), and the maximum count value is "00H" (256 decimal).

The output of the baud rate generator may be programmed to drive the transmitter (BRG output on TxC), the receiver (BRG output on RxC), both (BRG output on TxC and RxC), or neither (TxC and RxC are inputs). After a reset, the baud rate generator is disabled, divide by four is selected, and TxC and RxC are inputs.

The baud rate generator should be disabled before the CPU writes to the time constant register. This is necessary because no attempt was made to synchronize the loading of a new time constant with the clock used to drive the BRG.

Figure 8 indicates the external components needed to connect a crystal oscillator to the SIO XTAL inputs. The allowed crystal parameters are also listed.

For a 3.6864MHz input signal to the baud rate generator, the time constants, listed in table 2, are loaded to obtain the desired baud rates (in x1 clock mode).

ASYNCHRONOUS OPERATION

INTRODUCTION

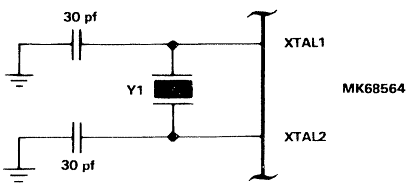
Many types of Asynchronous operations are performed by the MK68564 SIO. Figure 9 represents a typical Asynchronous message format and some of the options available on the SIO. The transmit process inserts start, stop, and parity bits to a variable data format and supplies a serial data stream to the Transmit Data output (TxD). The receiver takes the data from the Receive Data input (RxD) and strips away expected start and stop bits at a programmed clock rate. It provides error checking for overrun, parity, and carrier-loss errors, and, if desired, provides interrupts for these conditions.

To set up the SIO for Asynchronous operation, the following registers need to be initialized : Mode Control Register, Interrupt Control Register, Receiver Control Register, and Transmitter Control Register. The Mode Control Register must be programmed before the other registers to assure proper operation of the SIO. The following registers are used to transfer data or to communicate status between the SIO and the CPU or other bus master when operating in Asynchronous modes : Command Register, Status Register 0, Status Register 1, Data Register, and the Vector Register.

Table 2 : Time-Constant Values.

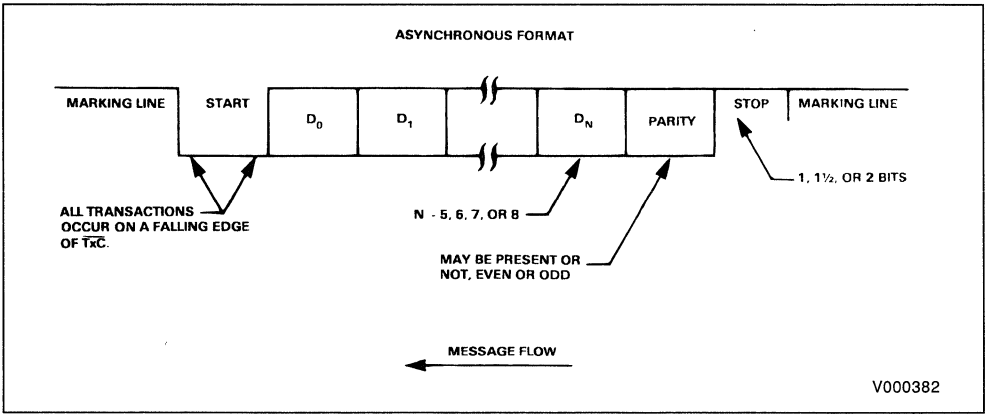
Rate	Time Constant	Divide By	Error
19200	48	4	
9600	96	4	
7200	128	4	
4800	192	4	
3600	256	4	
2400	24	64	
2000	29	64	69 %
1800	32	64	
1200	48	64	
600	96	64	
300	192	64	

Figure 8 : SIO External Oscillator Components.



CRYSTAL PARAMETERS :
Parallel Resonance, Fundamental Mode AT Cut
Rs â 150Ω (Fr = 2.8 - 5.0MHz)
Rs â 300Ω (Fr = 2.0 - 2.7MHz)
Cl = 18pf ; Cm = 0.02pF ; Ch = 5pF ; Lm = 96MHz
Fr (typ) = 2.457MHz

Figure 9 : Asynchronous Message Format.



The SIO provides five I/O lines that may be used for modem control, for external interrupts, or as general purpose I/O. The Request To Send (RTS) and Data Terminal Ready (DTR) pins are outputs that follow the inverted state of their respective bits in the Transmitter Control Register. The RTS pin can also be used to signal the end of a message in Asynchronous modes, as explained below in the transmitter section. The Data Carrier Detect (DCD), Clear To Send (CTS), and SYNC pins are inputs to the SIO in Asynchronous modes. DCD and CTS can be used as auto enables to the receiver and transmitter, respectively, or if External/Status Interrupts are enabled all three input pins will be monitored for a change of status. If these inputs change for a period of time greater than the minimum specified pulse width, an interrupt will be generated.

In the following discussion, all interrupt modes are assumed enabled.

ASYNCHRONOUS TRANSMIT

Start of Transmission. The SIO will start transmitting data when the Transmit Enable bit is set to a one, and a character has been loaded into the transmit buffer. If the TxAuto Enables bit is set, the SIO will wait for a Low on the Clear To Send input (CTS) before starting data transmission. The Tx Auto Enables feature allows the programmer to send the first data character of the message to the SIO without waiting for CTS to go Low. In all cases, the Transmit Enable bit must be set before transmission can begin. The transitions on the CTS pin will generate External/Status interrupt requests and also latch up the external/status logic. The external/status logic should be rearmed by issuing a Reset External/Status Interrupts command.

Transmit Characteristics. The SIO automatically inserts a start bit, the programmed parity bit (odd, even, or no parity), and the programmed number of stop bits to the data character to be transmitted. The transmitter can transmit from one to eight data bits per character. All characters are transmitted least-significant bit first. When the character length programmed is six or seven bits, the unused bits of the transmit buffer are automatically ignored. When a character length of five bits or less is programmed, the data loaded into the transmit buffer must be formatted as described in the Transmitter Control Register part of the Register Description section. Serial data is shifted out of the TxD pin on the falling edge of the Transmit Clock (TxC) at a rate equal to 1, 1/16th, 1/32nd, or 1/64th of TxC.

Data Transfer. The SIO will signal the CPU or other bus master with a transmit interrupt request and set the Tx Buffer Empty bit in Status Register 0, every time the contents of the transmit buffer are loaded into the transmit shift register. The interrupt request will be cleared when a new character is loaded into the transmit buffer, or a Reset Tx Interrupt Pending command (Command 5) is issued. If Command 5 is issued, the transmit buffer will have to be loaded before any additional transmit interrupt requests are generated. The Tx Buffer Empty bit is reset when a new character is loaded into the transmit buffer.

The All Sent bit in Status Register 1 is used to indicate when all data in the shift register has been transmitted, and the transmit buffer is empty. This bit is Low, while the transmitter is sending characters, and it will go High one bit time after the transmit clock that clocks out the last stop bit of the character on the TxD pin. No interrupts are generated by the All Sent bit transitions. The Request To Send

(RTS) bit in the Transmitter Control Register may also be used to signal the end of transmission. If this bit is set to a one, its associated output pin (RTS) will go Low. When this bit is reset to a zero, the RTS pin will go High one bit time after the transmit clock that clocks out the last stop bit, only if the transmit buffer is empty.

The Transmit Data output (TxD) is held marking (High) after a reset or when the transmitter has no data to send. Under program control, the Send Break command can be issued to hold TxD spacing (Low) until the command is cleared, even if the transmitter is not enabled.

ASYNCHRONOUS RECEIVE

Asynchronous operation begins when the Receiver Enable bit in the Receiver Control Register is set to a one. If the Rx Auto Enables bit is also set, the Data Carrier Detect (DCD) input pin must be Low as well. The receiver will start assembling a character as soon as a valid start bit is detected, if a clock mode other than x1 is selected. A valid start bit is a High-to-Low transition on the Receive Data input (RxD) with the Low time lasting at least one-half bit time. The High-to-Low transition starts an internal counter and, at mid-bit time, the counter output is used to sample the input signal to detect if it is still Low. When this condition is satisfied, the following data bits are sampled at mid-bit time until the entire character is assembled. The start bit detection logic is then rearmed to detect the next High-to-Low transition. If the x1 clock mode is selected, the start bit detection logic is disabled, and bit synchronization must be accomplished externally. Receive data is sampled on the rising edge of the Receiver Clock (RxC).

The receiver may be programmed to assemble five to eight data bits, plus a parity bit, into a character. The character is right-justified in the shift register and then transferred to the receive data FIFO. All data transfers to the FIFO are in eight-bit groups. If the character length assembled is less than eight bits, the receiver inserts ones in the unused bits. If parity is enabled, the parity bit is transferred with the character, unless eight bits per character is programmed, in which case, the parity bit is stripped from the character before transfer.

A Receiver Interrupt request is generated every time a character is shifted to the top of the receive data FIFO, if Interrupt On All Receive Characters mode is selected. The Rx Character Available bit in Status Register 0 is also set to a one every time a character is shifted to the top of the receive data FIFO. The Rx Character Available bit is reset to a zero when the receive buffer is read.

After a character is received, it is checked for the following error conditions :

Parity Error. If parity is enabled, the Parity Error bit in Status Register 1 is set to a one whenever the parity bit of the received character does not match the programmed parity. Once this bit is set, it remains set (latched), until an Error Reset command (Command 6) is issued. A Special Receive Condition interrupt is generated when this bit is set, if parity is programmed as a Special Receive Condition.

Framing Error. The CRC/Framing Error bit in Status Register 1 is set to a one, if the character is assembled without a stop bit (a Low level detected instead of a stop bit). This bit is set only for the character on which the framing error occurred ; it is updated at every character time. Detection of a framing error adds an additional one-half of a bit time to the character time, so the framing error is not interpreted as a new start bit. A Special Receive Condition interrupt is generated when this bit is set..

Overrun Error. If four or more characters are received before the CPU (or other bus master) reads the receive buffer, the fourth character assembled will replace the third character in the receive data FIFO. If more than four characters have been received, the last character assembled will replace the third character in the data FIFO. The character that has been written over is flagged with an overrun error in the error FIFO.

When this character is shifted to the top of the receive data FIFO, the Receive Overrun Error bit in Status Register 1 is set to a one ; the error bit is latched in the status register, and a Special Receive Condition interrupt is generated. Like Parity Error, this bit can only be reset by an Error Reset Command.

Break Condition. A break character is defined as a start bit, an all zero data word, and a zero in place of the stop bit. When a break character is detected in the receive data stream, the Break/Abort bit in Status Register 0 is set to a one, and an External/Status interrupt is requested. This interrupt is then followed by a Framing Error interrupt request when the CRC/Framing Error bit in Status Register 1 is set. A Reset External/Status Interrupts command (Command 2) should be issued to reinitialize the break detection interrupt logic. The receiver will monitor the data stream input for the termination of the break sequence. When this condition is detected, the Break/Abort bit will be reset, if Command 2 has been issued, and another External/Status interrupt request will be generated. This interrupt should also be handled by issuing Command 2 to reinitialize the external/status logic. At the end of the break sequence, a single null character will be left in the receive data FIFO. This character should be read and discarded.

Because Parity Error and Receive Overrun Error flags are latched, the error status that is read from Status Register 1 reflects an error in the current word in the receive data FIFO, plus any parity or overrun errors received since the last Error Reset command. To keep correspondence between the state of the error FIFO and the contents of the receive data FIFO, Status Register 1 should be read before the receive buffer. If the status is read after the data and more than one character is stacked in the data FIFO during the read of the receive buffer, the status flags read will be for the next word. Keep in mind that when a character is shifted up to the top of the data FIFO (the receive buffer), its error flags are shifted into Status Register 1

.An exception to the normal flow of data through the receive data FIFO occurs when the Receive Interrupt On First Character Only mode is selected. A Special Receive Condition interrupt in this mode holds the error data, and the character itself (even if read from the data FIFO) until the Error Reset command (command 6) is issued. This prevents further data from becoming available in the receiver, until Command 6 is issued, and allows CPU intervention on the character with the error even if DMA or block transfer techniques are being used.

SYNCHRONOUS OPERATION

INTRODUCTION

Before describing byte-oriented, synchronous transmission and reception, the three types of character synchronization - Monosync, Bysync, and External Sync - require some explanation. These modes use the x1 clock for both Transmit and Receive operations. Data is sampled on the rising edge of the Receive Clock input (RxC). Transmitter data transitions occur on the falling edge of the Transmit Clock input (TxC).

The differences between Monosync, Bisync, and External Sync are in the manner in which initial receive character synchronization is achieved. The mode of operation must be selected before sync characters are loaded, because the registers are used differently in the various modes. Figure 10 shows the formats for all three synchronous modes.

MONOSYNC. In the Monosync mode (8-bit sync mode), the transmitter transmits the sync character in Sync Word Register 1. The receiver compares the single sync character with the programmed sync character stored in Sync Word Register 2. A match implies character synchronization and enables data transfer. The SYNC pin is used as an output in this mode and is active for the part of the receive clock that detects the sync character.

BISYNC. In the Bisync mode (16-bit sync mode), the transmitter transmits the sync character in Sync Word Register 1 followed by the sync character in Sync Word Register 2. The receiver compares the two contiguous sync characters with the programmed sync characters stored in Sync Word Registers 1 and 2. A match implies character synchronization and enables data transfer. The SYNC pin is used as an output in this mode and is active for the part of the receive clock that detects the sync characters.

External Sync. In the External Sync mode, the transmitter transmits the sync character in Sync Word Register 1. Character synchronization for the receiver is established externally. The SYNC pin is an input that indicates that external character synchronization has been achieved. After the sync pattern is detected, the external logic must wait for two full Receive Clock cycles to activate the SYNC input pin (see figure 11). The SYNC input pin must be held Low until character synchronization is lost. Character assembly begins on the rising edge of the Receive Clock that precedes the falling edge of the SYNC input pin.

In all cases, after a reset (hardware or software), the receiver is in the Hunt phase, during which time the SIO looks for character synchronization. The Hunt phase can begin only when the receiver is enabled, and data transfer can begin only when character synchronization has been achieved. If character synchronization is lost, the Hunt phase can be reentered by setting the Enter Hunt Mode bit in the Receiver Control Register. In the transmit mode, the transmitter always sends the programmed number of sync bits (8 or 16), regardless of the bits per character programmed.

In the Monosync, Bisync, and External Sync modes, assembly of received data continues until the SIO is reset, or until the receiver is disabled (by command or the DCD pin in the Rx Auto Enables mode), or until the CPU sets the Enter Hunt Mode bit.

After initial synchronization has been achieved, the operation of the Monosync, Bisync, and External Sync modes is quite similar. Any differences are specified in the following text.

To set up the SIO for Synchronous operations, the following registers need to be initialized : Mode

Control Register, Interrupt Control Register, Receiver Control Register, Transmitter Control Register, Sync Word 1, and Sync Word 2. The Mode Control Register must be programmed before other registers to assure proper operation of the SIO. The following registers are used to transfer data or communicate status between the SIO and the CPU or other bus master : Command Register, Status Register 0, Status Register 1, Data Register, and the Vector Register.

The SIO provides four I/O lines in Synchronous modes that may be used for modem control, for external interrupts, or as general purpose I/O. The Request To Send (RTS) and Data Terminal Ready (DTR) pins are outputs that follow the inverted state of their respective bits in the Transmit Control Register. The Data Carrier Detect (DCD) and Clear To Send (CTS) pins are inputs that can be used as auto enables to the receiver and transmitter, respectively. If External/Status Interrupts are enabled, the DCD and CTS pins will be monitored for a change of status. If these inputs change for a period of time greater than the minimum specified pulse width, an interrupt will be generated.

In the following discussion, all interrupt modes are assumed enabled.

SYNCHRONOUS TRANSMIT

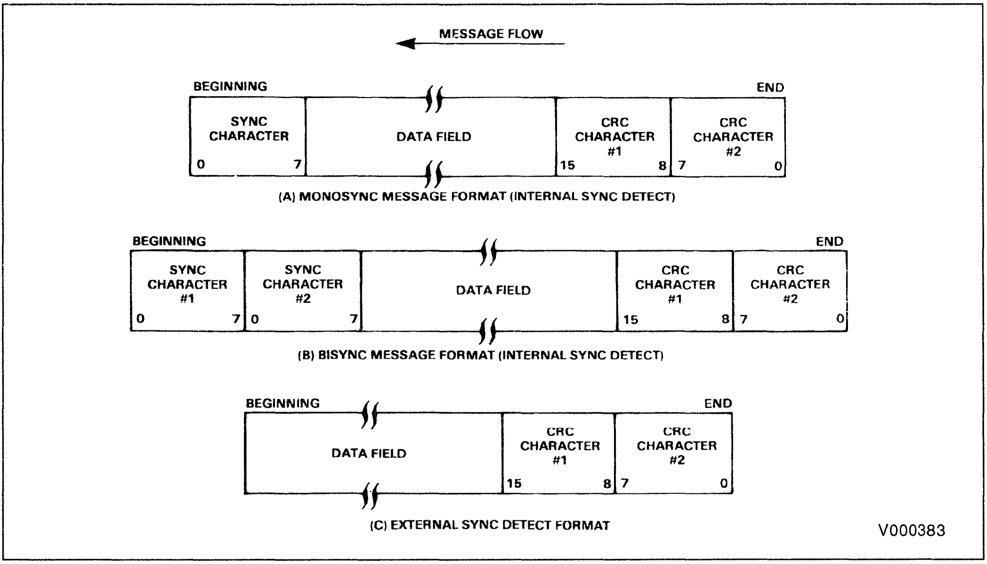
Initialization. Byte-oriented transmitter programs are usually initialized with the following parameters : odd-even or no parity, x1 clock mode, 8- or 16-bit sync character(s), CRC polynomial, Transmit Enables, interrupt modes, and transmit character length. If Parity is enabled, the transmitter will only add a parity bit to a character that is loaded into the transmit buffer ; it will not add a parity bit to the automatically inserted sync character(s) or the CRC characters.

One of two polynomials may be used with Synchronous modes, CRC-16 ($X^{16} + X^{15} + X^2 + 1$) or SDLC-CRC ($X^{16} + X^{12} + X^5 + 1$). For either polynomial (SDLC mode not selected), the CRC generator and checker are reset to all zeros. Both the receiver and transmitter use the same polynomial.

After reset (hardware or software), or when the transmitter is not enabled, the Transmit Data (TxD) output pin is held High (marking). Under program control, the Send Break bit in the Transmitter Control Register can be set to a one, forcing the TxD output pin to a Low level (spacing), even if the transmitter is not enabled. The spacing condition will persist until the Send Break bit is reset to a zero. A programmed break is effective as soon as it is written into the Transmit Control Register ; any characters in the transmit buffer and transmit shift register are lost.

If the transmit buffer is empty when the Transmit Enable bit is set to a one, the transmitter will start sending 8- or 16-bit sync characters. Continuous syncs will be transmitted on the TxD output pin, as long as no data is loaded into the transmit buffer. Note, if a

Figure 10 : Synchronous Formats.



character is loaded into the transmit buffer before enabling the transmitter, that character will be sent in place of the sync character(s).

Start of Transmission. Transmission will begin with the loading of the first data character into the transmit buffer, if the transmitter is already enabled. For CRC to be calculated correctly on each message, the CRC generator must be reset to all zeros before the first data character is loaded into the transmit buffer. This is accomplished by issuing a Reset Tx CRC Generator command in the Command Register.

Synchronous Transmit Characteristics. In all Synchronous modes, characters are sent with the least-significant bits first. All data is shifted out of the Transmit Data pin (TxD) on the falling edge of the Transmit Clock (TxC). The transmitter can transmit from one to eight data bits per character. This requires right-hand justification of data written to the transmit buffer, if the selected word length is less than eight bits per character. When the programmed character length is six or seven bits, the unused bits in the transmit buffer are ignored. If a word length of five bits per character or less is selected, the data loaded into the transmit buffer must be formatted as described in the Transmit Control register part of the Register Description section.

The number of bits per character to be transmitted can be changed on the fly. Any data written to the transmit buffer, after the bits per character field is changed, are affected by the change. The same is true of any characters in the buffer at the time the bits per character field is changed. The change in the number of bits per character does not affect the character in the process of being shifted out.

A transmitted message can be terminated by CRC and sync characters, by sync characters only, or by pad characters (replacing the sync character(s) in the Sync Word Registers with pad characters). How a message is terminated is controlled by the Tx Underrun/EOM latch in Status Register 0.

Figure 11a : External Sync Timing.

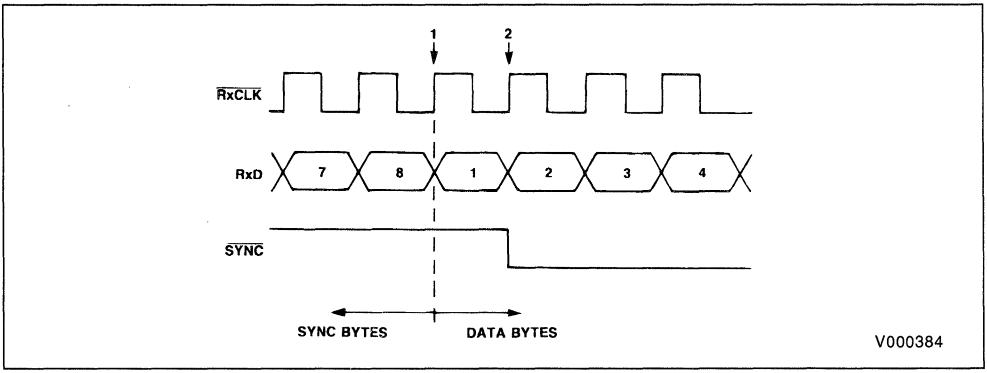
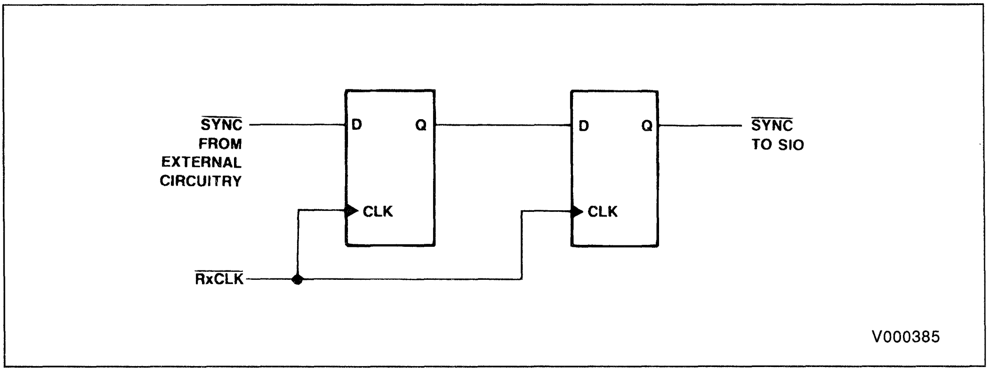


Figure 11b : Simple External Sync Delay.



Data Transfer. A Transmit Interrupt is generated each time the transmit buffer becomes empty. The interrupt may be satisfied either by writing another character into the transmit buffer or by resetting the Transmit Interrupt Pending latch with a Reset Tx Interrupt Pending command. If the interrupt is satisfied with this command, and nothing more is written into the transmit buffer, there can be no further Transmit Interrupts due to a Buffer Empty condition, because it is the process of the buffer becoming empty that causes the interrupts. This situation does cause a Transmit Underrun condition when the data in the shift register is shifted out.

Another way of detecting when the transmitter requires service is to poll the Tx Buffer Empty bit in Status Register 0. This bit is set to a one every time the data in the transmit buffer is downloaded into the transmit shift register. When data is written to the transmit buffer, this bit is reset to zero.

The SIO has all the signals and controls necessary to implement a DMA transfer routine for the transmitter. The routine may be configured to enable the DMA controller, after the first character is written to the transmit buffer, and then using the $\overline{\text{TxRDY}}$ output pin to signal the DMA that the transmitter requires service. If a data character is not loaded into the transmit buffer by the time the transmit shift register is empty, the SIO enters the Transmit Underrun condition.

Transmit Underrun/End of Message. When the transmitter has no further data to transmit, the SIO inserts filler characters to maintain synchronization. The SIO has two programmable options for handling this situation : sync characters can be inserted, or the CRC characters generated so far can be sent, followed by sync characters. These options are controlled by the state of the Transmit Underrun/EOM Latch in Status Register 0.

Following a hardware or software reset, the Transmit Underrun/EOM Latch is set to a one. This allows sync characters to be inserted when there is no data to send. CRC is not calculated on the automatically inserted sync characters. To allow CRC characters to be sent when the transmitter has no data, the Transmit Underrun/EOM Latch must be reset to zero. This latch is reset by issuing a Reset Tx Underrun/EOM Latch command in the Command Register. Following the CRC characters, the SIO sends sync characters to terminate the message.

There is no restriction as to when, in the message, the Transmit Underrun/EOM Latch can be reset, but once the reset command is issued, the 16-bit CRC is sent and followed by sync characters the first time the transmitter has no data to send. A Transmit Underrun condition will cause an External/Status Interrupt to be generated whenever the Transmit Underrun/EOM Latch is set.

For sync character insertion only, at the termination of a message, a Transmit Interrupt is generated only after the first automatically inserted sync character is loaded into the transmit shift register. The status bits in Status Register 0 indicate that the Transmit Underrun/EOM Latch and the Tx Buffer Empty bit are set.

For CRC insertion, followed by sync characters, at the termination of a message, the Transmit Underrun/EOM Latch is set, and the Tx Buffer Empty bit is reset while the CRC characters are being sent. When the CRC characters are completely transmitted, the Tx Buffer Empty status bit is set, and a Transmit Interrupt is generated, indicating to the CPU that another message can begin. This Transmit Interrupt occurs when the first sync character following the CRC characters is loaded into the transmit shift register. If no more messages are to be transmitted, the program can terminate transmission by disabling the transmitter.

CRC Generation. Setting the Tx CRC Enable bit in the Transmit Control Register initiates CRC accumulation when the program sends the first data character to the SIO. To ensure CRC is calculated correctly on each message, the Reset Tx CRC Generator command should be issued before the first data character of the message is sent to the SIO.

The Tx CRC Enable bit can be changed on the fly at any point in the message to include or exclude a particular data character from CRC accumulation. The Tx CRC Enable bit should be in the desired state when the data character is loaded from the transmit data buffer into the transmit shift register. To ensure this bit is in the proper state, the Tx CRC Enable bit should be loaded before sending the data character to the SIO.

Transmit Termination. The SIO is equipped with a special termination feature that maintains data integrity and validity. If the transmitter is disabled (by resetting the Transmit Enable bit or using the Tx Auto Enable signal) while a data or sync character is being transmitted, the character is transmitted as usual but is followed by a marking line instead of sync or CRC characters. When the transmitter is disabled, a character in the transmit buffer remains in the buffer. If the transmitter is disabled while CRC characters are being transmitted, the 16-bit transmission is completed, but the remaining bits of the CRC characters are replaced by sync characters.

Bisync Protocol Transmission. In a Bisync Protocol operation, once synchronization is achieved between the transmitter and receiver, fill characters are inserted to maintain that synchronization when the transmitter has no more data to send. The different options available in the SIO are described in the Transmit Underrun/End Of Message part of this section. If pad characters are to be sent in place of sync characters following the transmission of the CRC, the program can set the SIO transmitter to eight bits per character and then load "FFH" to the transmit buffer while the CRC characters are being sent. Alternatively, the sync characters in Sync Word Registers 1 and 2 can be redefined to be pad characters during this time. The following example is included to clarify this point :

The SIO interrupts the CPU with a Transmit Interrupt when the Tx Buffer Empty bit is set.

The CPU recognizes that the last character (ETX) of the message has already been sent to the SIO transmit buffer by examining the internal program status.

To force the SIO to send CRC, the CPU issues the Reset Tx Underrun/EOM Latch command and clears the current Transmit Interrupt with the Reset Tx Interrupt Pending command. Resetting the interrupt with this command prevents the SIO from requesting more data. The SIO then begins to send CRC (because the transmitter is in an underrun condition) and sets the Transmit Underrun/EOM Latch, which causes an External/Status Interrupt.

The CPU satisfies the External/Status Interrupt by loading pad characters into the transmit buffer and clears the interrupt by issuing the Reset External/Status Interrupt command.

The pad character will follow the CRC characters in this sequence, instead of the usual sync characters. A Transmit Interrupt is generated when the pad character is loaded into the transmit shift register.

From this point on, the CPU can send more pad characters or sync characters.

The transparent mode of operation in Bisync Protocol is made possible with the SIO's ability to change the Tx CRC Enable bit at any time during program sequencing and with the additional capability of inserting 16-bit sync characters. Exclusion of DLE (Data Link Escape) characters from CRC calculation can be achieved by disabling CRC calculations immediately preceding the DLE character transfer to the transmit buffer. In the case of a transmit underrun condition in the transparent mode, a pair of DLE-SYN characters is sent. The SIO can be programmed to send the DLE-SYNC sequence by loading a DLE character into Sync Word Register 1 and a SYNC character into Sync Word Register 2.

The SIO always transmits two sync characters (16 bits) in Bisync mode. If additional sync characters are to be transmitted before a message, the CPU can delay loading data to the transmit buffer until the required number of syncs have been sent. No CRC calculations are done on any automatically inserted sync characters. An alternate method of sending additional sync characters is to load the sync characters into the transmit buffer, in which case the transmitter will treat the characters as data. The Tx CRC Enable bit should not be set, until true data is going to be loaded into the buffer, to avoid performing CRC calculations on the additional sync characters.

SYNCHRONOUS RECEIVE

Initialization. Byte-oriented receive programs are usually initialized with the following parameters : odd-even or no parity, x1 clock mode (necessary because of the start bit detection logic), 8- or 16-bit sync character(s), CRC polynomial, Receiver Enables, interrupt modes, and receive character length. Care must be taken if Parity is enabled. The receiver will usually detect a Parity Error on all sync characters, after synchronization is achieved, and on the CRC characters.

Receiver Hunt Mode. After the SIO is initialized for a Synchronous receive operation, the receiver is in the Hunt phase. During the Hunt phase, the receiver does a bit-by-bit comparison of the incoming data stream and the sync character(s) stored in the Sync Word Register 2 (for Monosync mode) and Sync Word Registers 1 and 2 (for Bisync mode). When a match occurs, the Hunt phase is terminated, and the following data bits are assembled into the programmed character length and loaded into the receive data FIFO.

Receive Characteristics. The receiver may be programmed to assemble five to eight data bits into a character. The character is right-justified in the shift register and transferred to the receive data FIFO. All data transfers to the FIFO are in 8-bit groups. When the programmed character length is less than eight bits, the most significant bit(s) transferred with a character will be the least significant bit(s) of the next character. The programmed character length may be changed on the fly during a message ; however, care must be taken to assure that the change is effective before the number of bits specified for the character length have been assembled.

When the Sync Character Load Inhibit bit in the Receiver Control Register is set, all characters in the receive data stream that match the byte loaded into Sync Word Register 1 will be inhibited from loading into the receive data FIFO. The comparison between Sync Word Register 1 and the incoming data occurs at a character boundary time. This is an 8-bit comparison, regardless of the bits per character programmed. CRC calculations will be performed on all bytes, even if the characters are not transferred to the receive data FIFO, as long as the Rx CRC Enable bit is set.

Data Transfer and Status Monitiring. After character synchronization is achieved, the assembled characters are transferred to the receive data FIFO, and the status information for each character is transferred to the receive error FIFO. The following four modes are available to transfer the received data and its associated status to the CPU.

No Receive Interrupts Enabled. This mode is used either for polling operations or for off-line conditions. When transferring data, using a polling routine, the Rx Character Available bit in Status Register 0 should be checked to determine if a receive character is available for transfer. Only when a character is available should the receive buffer and Status Register 1 be read. The Rx Character Available bit is set when a character is loaded to the top of the receive data FIFO. This bit is reset during a read of the receive buffer.

Interrupt On First Character Only. This interrupt mode is normally used to start a DMA transfer routine or, in some cases, a polling loop. The SIO will generate an interrupt the first time a character is shifted to the top of the receive data FIFO after this mode is selected or reinitialized. An interrupt will be generated thereafter only if a Special Receive Condition is detected. This mode is reinitialized with the Enable Interrupt On Next Receive Character command. Parity Errors do not cause interrupts in this mode ; however, a Receive Overrun Error will.

Interrupt On Every Character. This interrupt mode will generate a Receiver Interrupt every time a character is shifted to the top of the receive data FIFO. A Special Receive Condition interrupt on a parity error is optional in this mode.

Special Receive Condition Interrupt. The special condition interrupt mode is not an interrupt mode as such, but works in conjunction with Interrupt On Every Character or Interrupt On First Character Only modes. When the Status Affects Vector bit in either channel is set, a Special Receive condition will modify the Receive Interrupt vector to signal the CPU of the special condition. Receive Overrun Error and Parity Error are the only Special Receive Conditions in Synchronous receive mode. The overrun and parity error status bits in Status Register 1 are latched when they occur ; they will remain latched until an Error Reset command is issued. As long as either one of these bits is set, a Special Receive Condition Interrupt will be generated at every character available time. Since these two status bits are latched, the error status in Status Register 1, when read, will reflect an error in the current word in the receive buffer, in addition to any Parity or Overrun errors received since the last Error Reset command.

CRC Error Checking and Receiver Message Termination. A CRC error check on the received message can be performed on a per character basis under program control. The Rx CRC Enable bit must set/reset by the program before the next character is transferred from the receive shift register to the receive data FIFO. This ensures proper inclusion or exclusion of data characters in the CRC check.

There is an 8-bit delay between the time a character is transferred to the receive data FIFO and the time the same character starts to enter the CRC checker. An additional 8-bit times are needed to perform CRC calculations on the character. Due to this serial nature of CRC calculations, the Receive Clock (RxC) must cycle 16 times after the second CRC character has been loaded into the receive data FIFO or 20 times (the previous 16 plus 3-bit buffer delay and 1-bit input delay) after the last bit is at the RxD input, before CRC calculation is complete. The CRC Framing Error bit in Status Register 1 will contain the comparison results of the CRC checker. The comparison results should be zero, indicating error-free transmission. The results in the status bit are valid only at the end of CRC calculation. If the result is examined before this time, it usually indicates an error (the bit is High). The comparison is made at each character available time and is valid until the character is read from the receive data FIFO.

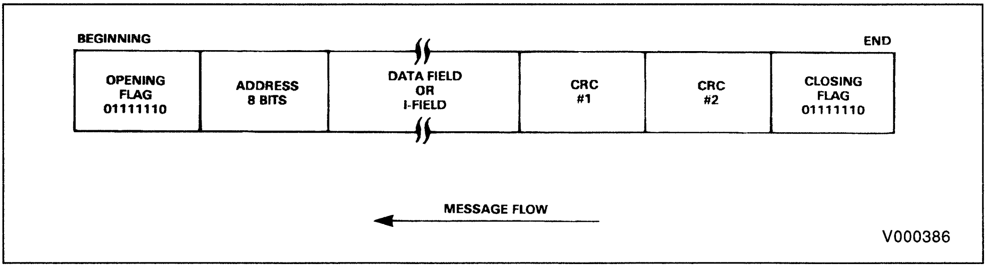
SDLC/HDLC OPERATION

INTRODUCTION

The MK68564 SIO is capable of handling both High-level Synchronous Data Link Control (HDLC) and IBM Synchronous Data Link Control (SDLC) protocols. In the following discussion, only SDLC is referenced because of the high degree of similarity between SDLC and HDLC.

The SDLC mode is considerably different from Monosync and Bisync protocols, because it is bit oriented rather than character oriented. Bit orientation makes SDLC a flexible protocol in terms of mes-

Figure 12 : Transmit/Receive SDLC/HDLC Message Format.



sage length and bit patterns. The SIO has several built-in features to handle variable message length. Detailed information concerning SDLC protocol can be found in literature on this subject, such as IBM document GA27-3093.

The SDLC message, called the frame (figure 12), is opened and closed by flags, which are similar to the sync characters used in other Synchronous protocols. The SIO handles the transmission and recognition of the flag characters that mark the beginning and end of the frame. Note that the SIO can receive shared-zero flags but cannot transmit them. The 8-bit address field of a SDLC frame contains the secondary station address. The SIO receiver has an Address Search mode, which recognizes the secondary station so that it can accept or reject a frame.

The control field of the SDLC frame is transparent to the SIO ; it is simply transferred to the CPU. The SIO handles the Frame Check sequence in a manner that simplifies the program by incorporating features such as initializing the CRC generator to all ones, resetting the CRC checker when the opening flag is detected in the receive mode, and sending the Frame Check/Flag sequence in the transmit mode. Controller hardware is simplified by automatic zero insertion and deletion logic, contained in the SIO.

To set up the SIO for SDLC operation, the following registers need to be initialized : Mode Control Register, Interrupt Control Register, Receiver Control Register, Transmitter Control Register, Sync Word Register 1, and Sync Word Register 2. The Mode Control Register must be programmed before the other registers to assure proper operation of the SIO. The following registers are used to transfer data or communicate status between the SIO and the CPU or other bus master when operating in SDLC mode : Command Register, Status Register 0, Status Register 1, Data Register, and the Vector Register.

Sync Word Register 1 contains the secondary station address, and Sync Word Register 2 stores the flag character and must be programmed to "01111110".

The SIO provides four I/O lines in SDLC mode that may be used for modem control, for external interrupts, or as general purpose I/O. The Request To Send (RTS) and Data Terminal Ready (DTR) pins are outputs that follow the inverted state of their respective bits in the Transmit Control Register. The Data Carrier Detect (DCD) and Clear To Send (CTS) pins are inputs that can be used as auto enables to the receiver and transmitter, respectively. If External/Status Interrupts are enabled, the DCD and CTS pins will be monitored for a change of status. If these inputs change for a period of time greater than the minimum specified pulse width, an interrupt will be generated.

In the following discussion, all interrupt modes are assumed enabled.

SDLC TRANSMIT

Initialization. The SIO is initialized for SDLC mode by selecting these parameters in the Mode Control Register : x1 Clock Mode, SDLC Mode, and Sync Modes Enabled. Parity is normally not used in SDLC mode, because the transmitter will not add parity to the flag character or the CRC characters, thus causing Parity Errors in the receiver. If CRC is to be calculated on the transmitted data, the SDLC-CRC polynomial must be selected in the Interrupt Control Register (CRC-16 polynomial in SDLC Mode will produce unknown results).

After reset (hardware or software), or when the transmitter is not enabled, the Transmit Data (TxD) output pin is held High (marking). Under program control, the Send Break bit in the Transmit Control Register can be set to a one, forcing the TxD output to a Low level (spacing), even if the transmitter is not enabled. The spacing condition will persist until the Send Break bit is reset to a zero. If the transmit buffer is empty when the Transmit Enable bit is set to a one, the transmitter will start sending flag characters. Continuous flags will be transmitted on the TxD output pin as long as no data is loaded into the transmit buffer.

Note : If a character is loaded into the transmit buffer before enabling the transmitter, that character will be sent in place of a flag.

An abort sequence may be transmitted at any time by issuing the Send Abort command (command 1). This causes at least eight, but less than fourteen, ones to be sent before the output reverts back to continuous flags. It is possible that the Abort sequence (eight 1's) could follow up to five continuous ones (allowed by the zero insertion logic) and, thus, cause as many as thirteen ones to be sent. Any data being transmitted and any data in the transmit buffer is lost when an abort is issued.

The zero insertion logic in the transmitter will automatically insert a 0 after five continuous ones in the data stream. This does not apply to flags or aborts.

Start of Transmission. Transmission will begin with the loading of the first character into the transmit buffer if the transmitter is already enabled. For CRC to be calculated correctly on each frame, the CRC generator must be initialized to all ones before the first character is loaded. This is accomplished by issuing a Reset Tx CRC Generator command in the Command Register. The first non-flag character transmitted is the address field. The SIO does not automatically transmit a station address, this is left to the programmer. The SIO will only transmit flags and CRC characters automatically.

SDLC Transmit Characteristics. Any length SDLC frame can be transmitted. All characters are transmitted with the least-significant bits first. All data is shifted out of the Transmit Data pin (TxD) on the falling edge of the Transmit Clock (TxC). The transmitter transmit from one to eight data bits per character. This requires right-hand justification of data written to the transmit buffer, if the word length selected is less than eight bits per character. When the programmed character length is six or seven bits, the unused bits in the transmit buffer are ignored. If a word length of five bits per character or less is selected, the data loaded into the transmit buffer must be formatted as described in the Transmit Control Register part of the Register Description section.

The number of bits per character to be transmitted can be changed on the fly. Any data, written to the transmit buffer after the bits per character field is changed, are affected by the change. The same is true of any characters in the buffer at the time the bits per character field is changed. The change in the number of bits per character does not affect the character in the process of being shifted out. Flag characters are always eigth bits in length, and CRC is always 16 bits in length, regardless of the programmed bits per character. A transmitted frame can be terminated by CRC and a flag, by a flag only, or by an abort. This is controlled by the Tx Underrun/EOM Latch and the Send Abort command.

Data Transfers. A Transmit Interrupt is generated each time the transmit buffer becomes empty. The interrupt may be satisfied either by writing another character into the transmit buffer or by resetting the Transmit Interrupt Pending latch with a Reset Tx Interrupt Pending command. If the interrupt is satisfied with this command, and nothing more is written into the transmit buffer, there are no further transmitter interrupts, and a Transmit Underrun condition will occur when the data in the shift register is shifted out. When another character is written to the buffer and loaded into the shift register, the transmit buffer can again become empty and interrupt the CPU. Following the flags in an SDLC operation, the 8-bit address field, control field, and information field may be sent to the SIO, using the Transmit Interrupt mode. The SIO transmits the frame check sequence using the Transmit Underrun feature.

When the transmitter is first enabled, the transmit buffer is already empty and obviously cannot then become empty. Therefore, no transmit interrupt can occur until after the first data character is written to the transmit buffer.

Another way of detecting when the transmitter requires service is to poll the Tx Buffer Empty bit in Status Register 0. This bit is set to a one every time the data in the transmit buffer is downloaded into the transmit shift register. When data is written to the transmit buffer, this bit is reset to zero.

The SIO has all the signals and controls necessary to implement a DMA transfer routine for the transmitter. The routine may be configured to enable the DMA controller, after the first character is written into the transmit buffer, using the TxRDY output pin to signal the DMA that the transmitter requires service. The DMA transfer can be terminated, when the DMA block count is reached, using the Tx Underrun/EOM interrupt.

Transmit Underrun/End of Message. SDLC-like protocols do not have provisions for fill characters within a message. The SIO, therefore, automatically terminates an SDLC frame when the transmit data buffer is empty, and the output shift register has no more bits to send. It does this by first sending the two bytes of CRC and the following these with one or more flags. This technique allows very high-speed transmission under DMA or CPU control, without requiring the CPU to respond quickly to the end of message situation.

The action that the SIO takes in the underrun situation depends on the state of the Transmit Underrun/EOM status bit in status Register 0. Following a reset, the Transmit Underrun/EOM bit is set to a one and prevents the insertion of CRC characters during the time there is no data to send. Consequently, flag characters are sent. If the Transmit Underrun/EOM status bit is zero when the underrun condition occurs, the 16-bit CRC character is sent, followed by one or more flag characters. The Transmit Underrun/EOM bit is reset to zero by issuing the Reset Tx Underrun/EOM Latch command in the Command Register.

The SIO begins to send a frame when data is written into the transmit buffer. Between the time the first data byte is written and the end of the message, the Reset Tx Underrun/EOM Latch command must be issued. The Transmit Underrun/EOM status bit will then be in the reset state at the end of the message (when underrun occurs), and CRC characters will automatically be sent. The transmission of the first CRC bit set the Transmit Underrun/EOM status bit to a one and generates an External/Status interrupt. Also, while CRC is being sent, the Tx Buffer Empty bit in Status Register 0 is reset to indicate that the transmit shift register is full of CRC data. When CRC has been completely sent, the Tx Buffer Empty status bit is set, and a Transmit Interrupt is generated to indicate that another message may begin. This interrupt occurs because CRC has been sent, and a flag has been loaded into the shift register. If no more messages are to be sent, the program can terminate transmission by disabling the transmitter.

Although there is no restriction as to when the Transmit Underrun/EOM bit can be reset within a message, it is usually reset after the first data character (secondary address field) is sent to the SIO. By resetting the status bit early in the message, the CPU has additional time (16 bits of CRC) to recognize if an unintentional transmit underrun situation has occured and to respond with an Abort command. Issuing the Abort command stops the flags from going on the line prematurely and eliminates the possibility of the receiver accepting the frame as valid data. This situation can happen if, at the receiving end, the data pattern immediately preceding the automatic flag insertion matches the CRC checker, giving a false CRC check result.

CRC Generation. The CRC generator must be reset to all ones at the beginning of each frame before CRC accumulation can begin. Actual accumulation begins on the first data character (address field) loaded into the transmit buffer. The Tx CRC Enable bit in the Transmit Control Register should be set to a one before the first character is loaded into the transmit buffer. In SDLC mode, all characters between the opening and the closing flags are included in CRC accumulation. The output of te CRC generator is inverted before it is transmitted.

Transmit Termination. The normal sequence at the end of a frame is

A Transmit Interrupt occurs when the last data character written to the transmit buffer is downloaded into the transmit shift register. This interrupt may be cleared by issuing a Reset Tx Interrupt Pending command.

An External/Status Interrupt occurs when the first bit of the CRC character is transmitted. This interrupt condition should first be tested to see if the interrupt was caused by the Tx Underrun/EOM bit going High and then reset by issuing a Reset External/Status Interrupts command.

A Transmit Interrupt occurs when the first bit of the flag is transmitted. This interrupt may be cleared by issuing a Reset Tx Interrupt Pending command, by loading the first character of the next message, or by disabling the transmitter.

If the transmitter is disabled while a character is being sent, that character (data or flag) is sent in the normal fashion but is followed by a marking line rather than CRC or more flag characters. If CRC characters are being sent at the time the transmitter is disabled, all 16 bits will be transmitted, followed by a marking line ; however, flags are sent in place of CRC. A character in the buffer when the transmitter is disabled remains in the buffer.

SDLC RECEIVE

Initialization. The receiver is enabled only after all of the receive parameters are initialized. After the Receiver Enable bit in the Receiver Control Register is set to a one, the receiver will be in the Hunt phase and will remain in this phase until the first flag is received. While in the SDLC mode, the receiver never re-enters the Hunt phase, unless specifically instructed to do so by the program or when an Abort character is detected in the incoming data stream.

Receiver Characteristics. The receiver may be programmed to assemble five to eight data bits into a character. The character is right-justified in the shift register and transferred to the receive data FIFO. All data transfers to the FIFO are in 8-bit groups. When the character length programmed is less than eight bits, the most significant bit(s) transferred with a character, will be the least-significant bit(s) of the next character. The character length programmed may be changed on the fly during the reception of a frame ; however, care must be taken to assure that the change is effective, before the number of bits specified for the character length has been assembled.

The address field in the SDLC frame is defined as an 8-bit field. When the Address Search Mode is selected, the receiver will compare the 8-bit character following the flag (first non-flag character) against the address programmed in Sync Word Register 1 or the hardwired global address (11111111). When the address field of the SDLC frame matches either address, data transfer will begin with the address character being loaded into the receive data FIFO. If the frame address does not match either address, the receiver will remain idle and continue checking every frame received for an address match. The address comparison is always done on the first eight bits following a flag, regardless of the bits per character programmed.

The SIO receiver is capable of matching only one address character. Once a match occurs, all data is transferred to the receive data FIFO at the programmed bits per character rate. If SDLC extended address field recognition is used (two or more address characters), the CPU program must be capable of determining whether or not the frame has a correct address field. If the correct address field is not received, the Hunt bit can be set to suspend reception and start searching for the next frame. The control field of an SDLC frame is transparent to the SIO ; it is transferred to the data FIFO as a data character. All extra zeros, inserted in the data stream by the transmitter, are automatically deleted in the receiver.

Data Transfer and Status Monitoring. After receipt of a valid flag, the assembled characters are transferred to the receive data FIFO, and the status information for each character is transferred to the receive error FIFO. The following four modes are available to transfer the received data and its associated status to the CPU.

No Receiver Interrupts Enabled. This mode is used for polling operations or for off-line conditions. When transferring data, using a polling routine, the Rx Character Available bit in Status Register 0 should be checked to determine whether or not a receive character is available for transfer. Only when a character is available should the receive buffer and Status Register 1 be read. The Rx Character Available bit is set to a one every time a character is shifted to the top of the receive data FIFO. This bit is reset when the receive buffer is read.

Interrupt On First Character Only. This interrupt mode is normally used to start a DMA transfer routine, or in some cases, a polling loop. The SIO will generate an interrupt the first time a character is shifted to the top of the receive data FIFO after this mode is selected or reinitialized. An interrupt will be generated thereafter only if a Special Receive Condition is detected. This mode is reinitialized with the Enable Interrupt On Next Received Character command. Parity Errors do not cause interrupts in this mode, but a Receive Overrun Error or an End Of Frame condition will.

Interrupt On Every Character. This interrupt mode will generate a Receiver Interrupt every time a character is shifted to the top of the receive data FIFO. A Special Receive Condition interrupt on a Parity error is optional in this mode.

Special Receive Condition Interrupt. The special condition interrupt mode is not an interrupt mode, as such, but works in conjunction with Interrupt On Every Character or Interrupt On First Character Only modes. When the Status Affects Vector bit in either channel is set, a Special Receive Condition will modify the Receive Interrupt vector to signal the CPU of the special condition. Receive Overrun Error, Parity Error, and End Of Frame are the Special Receive Conditions in SDLC mode. The Overrun and Parity error status bits in Status Register 1 are latched when they occur ; the End Of Frame bit is not latched. The two bits that are latched will remain latched and will generate a Special Receive Condition Interrupt at every character available time until an Error Reset command is issued. Since the two status bits are latched, the error status in Status Register 1, when read, will reflect an error in the current word in the receive buffer, in addition to any Parity or Overrun errors received since the last Error Reset command.

SDLC Receive CRC Checking. Control of the receiver CRC checker is automatic. It is reset by the leading flag, and CRC is calculated up to the final flag. The byte that has the End Of Frame bit set is the byte that contains the result of the CRC check. If the CRC/Framing Error bit is not set (zero), the CRC indicates a valid received message. A special check sequence is used for the SDLC check, because the transmitted CRC character is inverted. The final check must be 0001110100001111. he 2-byte CRC check characters should be read and discarded by the CPU, because the last two bits of the 2-byte SDLC CRC check characters are not transferred to the receive data FIFO due to the internal timing associated with detecting the closing flag.

Unlike Synchronous modes, the logic path in SDLC mode does not have an 8-bit delay between the time a character is transferred to the receive data FIFO and the time a character enters the CRC checker. This delay is not needed, because in SDLC, all characters between the opening and closing flags are included in the CRC calculations. When the second CRC character (six bits only) is loaded into the receive buffer, CRC calculation is complete.

SDLC Receive Termination. An SDLC frame is terminated when the closing flag is detected. The detection of the flag sets the End Of Frame bit in Status Register 1 and generates a Special Receive Condition Interrupt. In addition to the End Of Frame bit being set and the results of the CRC check, Status Register 1 has three bits of Residue code valid at this time. The Residue bits indicate the boundary between the CRC check bits and the I-field bits in the frame. A detailed description of the Residue code bits is given in the Register Description section, under Status Register 1.

Any frame can be prematurely aborted by an Abort sequence. Aborts are detected if seven or more continuous ones occur in the received data stream. This condition will cause an External/Status Interrupt to be generated with the Break/Abort bit in Status Register 0 set. After the Reset External/Status Interrupts command has been issued, a second interrupt will occur when the continuous ones condition has been cleared. This second interrupt can be used to distinguish between the Abort and Idle line conditions.

REGISTER DESCRIPTION

The following sections describe the MK68564 SIO registers. Each register is detailed in terms of bit configuration, the active states of each bit, their definitions, their functions, and their effects upon the internal hardware and external pins.

COMMAND REGISTER (CMDREG)

This register contains command and reset functions used in the programming of the SIO. This register is reset to "00H" by a channel or hardware reset. All bits, except Loop Mode, will be read as zeros during a read cycle.

D7	D6	D5	D4	D3	D2	D1	D0
CRC 1	CRC 0	CMD 2	CMD 1	CMD 0			LOOP MODE

D7, D6 : Reset Codes 1 and 0

CRC 1	CRC 0	
0	0	Null Code (no effect)
0	1	Reset Receiver CRC Checker
1	0	Reset Transmit CRC Generator
1	1	Reset Tx Underrun/End of Message Latch

Null Code. The null code has no effect on the MK68564 SIO. It is used when writing to the Command Register for some reason other than a CRC Reset.

Reset Receiver CRC Checker. It is necessary in Synchronous modes (except SDLC) to reset the receiver CRC circuitry between received messages. The CRC circuitry may be reset by one of the following : disabling the receiver, setting the Enter Hunt Mode bit in the Receiver Control Register, or issuing this Reset command. The CRC circuitry is reset automatically in SDLC mode when the End Of Frame flag is detected. This Reset command will initialize the CRC checker circuit to all ones in SDLC mode and all zeros in the other Synchronous modes.

Reset Transmit CRC Generator. This command resets the CRC generator to all ones in SDLC mode and all zeros in the other Synchronous modes. This command should be issued after the transmitter is enabled but before the first character of a message is loaded in the transmit buffer.

Reset Transmit Underrun/EOM Latch. This command resets the Underrun/EOM latch in Status Register 0 if the transmitter is enabled. The Underrun/EOM latch controls the transmission of CRC at the end of a message in Synchronous modes. When a transmit underrun occurs and this latch is low, CRC will be appended to the end of the transmission.

D5, D4, D3 : Command Codes

Command	CMD2	CMD1	CMD0	
0	0	0	0	Null Command (no effect)
1	0	0	1	Send Abort (SDLC mode)
2	0	1	0	Reset External/ Status Interrupts
3	0	1	1	Channel Reset
4	1	0	0	Enable Interrupt On Next Rx Character
5	1	0	1	Reset Tx Interrupt Pending
6	1	1	0	Error Reset
7	1	1	1	Null Command (no effect)

Command 0 (Null). The Null command has no effect on the MK68564 SIO.

Command 1 (Send Abort). This command is used in SDLC mode to transmit a sequence of eight to thirteen ones. This command always empties the transmit buffer ans sets the Tx Underrun/EOM Latch in Status Register 0 to a one

Command 2 (Reset External/Status Interrupts). After an External/Status interrupt (a change on a modem line or a Break condition, for example), the upper five bits in Status Register 0 are latched. This command reenables these bits and allows interrupts to occur again as a result of a status change. Latching the status bits captures short pulses, until the CPU has time to read the change. This command should be issued prior to enabling External/Status Interrupts.

Command 3 (Channel Reset). This command disables both the receiver and transmitter, forces TxD to a marking state ("1"), forces the modem control signals high, resets any pending interrupts from this channel, and resets all control registers. See the Reset section in the SIO System Interface Description for a more detailed list. All control registers for the channel must be rewritten after a Channel Reset command.

Command 4 (Enable Interrupt On Next Rx Character). This command is used to reactivate the Receive Interrupt On First Character Only interrupt mode. This command is normally issued after the present message is completed but before the next message has started to be assembled. The next character to enter the receive data FIFO after this command is issued will cause a receiver interrupt request.

Note : If the data FIFO has more than one character stored when this command is issued, the first previously stored character will cause the receiver interrupt request.

Command 5 (Reset Tx Interrupt Pending). When the Transmit Interrupt Enable mode is selected, the transmitter requests an interrupt when the transmit buffer becomes empty. In those cases, where there are no more characters to be sent (at the end of message, for example), issuing this command resets the pending transmit interrupt and prevents any further transmitter interrupt requests until the next character has been loaded into the transmit buffer or until CRC has been completely sent.

Command 6 (Error Reset). This command resets the upper seven bits in Status Register 1. Anytime a Special Receive Condition exists when Receive Interrupt On First Character Only mode is selected, the data with the special condition is held in the receive data FIFO until this command is issued.

Command 7 (Null). The Null command has no effect on the MK68564 SIO.

D2, D1 : Not Used (read as zeros)

D0 : Loop Mode

When this bit is set to a 1, the transmitter output is connected to the receiver input and TxC is connected to the receiver clock. RxC and RxD pins are not used by the receiver ; they are bypassed internally. RxC may still be used as the baud rate generator output in Loop Mode.

MODE CONTROL REGISTER (MODECTL)

The Mode Control Register contains control bits that affect both the receiver and the transmitter. This register must be initialized before loading the Interrupt, Tx, and Rx Control Registers, and the Sync Word Registers. This register is reset to "00H" by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
CLOCK RATE 1	CLOCK RATE 0	SYNC MODE 1	SYNC MODE 0	STOP BITS 1	STOP BITS 0	PARITY E/O	PARITY ON/OFF

D7, D6 : Clock Rate 1 and 0

These bits specify the multiplier between the input shift clock rates (TxC x RxC) and data rate. The same multiplier is used for both the transmitter and receiver, although the input clock rates may be different. In x16, x32, and x64 clock modes, the receiver start bit detection logic is enabled ; therefore, for Synchronous modes, the x1 clock rate must be specified. Any clock rate may be specified for Asynchronous mode ; however, if the x1 clock rate is selected, synchronization between the receive data and the receive clock must be accomplished externally.

CLOCK RATE 1	CLOCK RATE 0	Multiple	
0	0	x1	Clock Rate = Data Rate
0	1	x16	Clock Rate = 16 x Data Rate
1	0	x32	Clock Rate = 32 x Data Rate
1	1	x64	Clock Rate = 64 x Data Rate

D5, D4 : Sync Modes 1 and 0

These bits select the various options for character synchronization. These bits are ignored, unless Sync modes is selected in the Stop Bits filed of this register.

SYNC MODE 1	SYNC MODE 0	
0	0	8-bit Programmed Sync
0	1	16-bit Programmed Sync
1	0	SDLC Mode (01111110 flag pattern)
1	1	External Sync Mode

D3, D2 : Stop Bits 1 and 0

These bits determine the number of stop bits added to each Asynchronous character that is transmitted. The receiver always checks for one stop bit in Asynchronous mode. A special code (00) signifies that a Synchronous mode is to be selected. 1 1/2 stop bits is not allowed if x1 clock rate is selected, because it will lock up the transmitter.

STOP BIT 1	STOP BIT 0	
0	0	Sync Modes
0	1	1 Stop Bit per Character
1	0	11/2 Stop Bits per Character
1	1	2 Stop Bits per Character

D1 : Parity Even/Odd

If the Parity Enable bit is set, this bit determines whether parity is checked as even or as odd. (1 = even, 0 = odd). This bit is ignored if the Parity Enable bit is reset.

D0 : Parity Enable

If this bit is set to a one, one additional bit position beyond those specified in the bits/character control field is added to the transmitted data and is expected in the receive data. The received parity bit is transferred to the CPU as part of the data character, unless eight bits per character is selected in the Receiver Control Register.

INTERRUPT CONTROL REGISTER
(INTCTL)

This register contains the control bits for the various interrupt modes and the DMA handshaking signals. This register is reset to "00H" by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
CRC16/ SDLC	CTX RDY ENABLE	RX RDY ENABLE	RX INT MODE 1	RX INT MODE 0	STATUS AFFECTS	TX INT ENABLE	EXT INT ENABLE

D7 : CRC-16/SDLC-CRC

This bit selects the CRC polynomial used by both the transmitter and receiver. When set to a one, the CRC-16 polynomial (x16 + x15 + x2 + 1) is used ; when reset to a zero, the SDLC-CRC polynomial (x16 + x12 + x5 + 1) is used. If the SDLC mode is selected, the CRC generator and checker are preset to all ones and a special check sequence is used.

The SDLC-CRC polynomial must be selected in SDLC mode. Failure to do so will result in receiver CRC errors. When a Synchronous mode, other than SDLC, is selected, the CRC generator and checker are preset to all zeros (for both polynomials). This bit must be programmed before CRC is enabled in the receiver and transmitter control registers, to assure valid CRC generation and checking. This bit is ignored in Asynchronous modes.

D6 : Tx Ready Enable

When this bit is set to a one, the TxRDY output pin will pulse Low for three clock cycles (CLK) when the transmit buffer becomes empty. When this bit is zero, the TxRDY pin is held High.

D5 : Rx Ready Enable

When this bit is set to a one, the TxRDY output pin will pulse Low for three clockcycles (CLK) when a character is available in the receive buffer. If a Special Receive Condition is detected when the Receive Interrupt On First Character Only interrupt mode is selected, the RxRDY pin will not become active ; instead, a special Receive Condition interrupt will be generated. When this bit is zero, the RxRDY pin will be held High

D4, D3 : Receive Interrupt Modes 1 and 0

Together, these two bits specify the various character-avalaible conditions that will cause interrupt requests. When receiver interrupts are enabled, a Special Receive Condition can cause an interrupt request and modify the interrupt vector. Special Receive conditions are : Rx Overrun Error, Framing Error (in async mode), End Of Frame (in SDLC mode), and Parity Error (when selected). The Rx Overrun Error and the Parity Error conditions are latched in Status Register 1 when they occur ; they are cleared by an Error Reset command (Command 4) or by a hardware or channel rest.

Rx INT MODE 1	Rx INT MODE 0	
0	0	Receive Interrupts Disabled
0	1	Receive Interrupt On First Character Only
1	0	Interrupt On All Receive Characters-parity Error is a Special Receive Condition
1	1	Interrupt On All Receive Characters-parity Error is not a Special Receive Condition

Receive Interrupts Disabled. This mode prevents the receiver from generating an interrupt request and clears any pending receiver interrupts. If a character is avalaible in the receiver data FIFO, or if a Special Receive Condition exists before or during the time receiver interrupts are disabled, and receiver interrupts are then enabled without clearing these conditions, an interrupt request will immediately be generated.

Receive Interrupt On First Character Only. The receiver requests an interrupt in this mode on the first available character (or stored FIFO character), or on a Special Receive Condition. If a Special Receive Condition occurs, the data with the special condition is held in the receive data FIFO until an Error Reset command (Command 6) is issued.

The receive Interrupt On First Character Only mode can be re-enabled by the Enable Interrupt On Next Rx Character command (Command 4). If this interrupt mode was terminated by a Special Receive Condition, the Error Reset command must be issued, before Command 4, for proper operation to resume.

Interrupt On All Receive Characters. This mode ammows an interrupt for every character received (or character in the receive data FIFO) and provides a unique vector (if Status Affects ector is enabled) when a Special Receive Condition exists. When the interrupt request is due to a special condition, the data containing that condition, the data containing data FIFO.

D2 : Status Affects Vector

When this bit is zero, the value programmed into the Vector Register is returned during a read cycle or an interrupt acknowledge cycle. If the Vector Register has not been programmed following a hardware reset, then "0FH" is returned.

When this bit is a one, the vector returned during a read cycle or an interrupt acknowledge cycle is variable. The variable field returned depends on the highest-priority pending interrupt at the start of the cycle.

The Status Affects Vector control bits from both channels are logical "or" ed together ; therefore, if either is programmed to a one, its operation affects both channels. This is the only control bit that functions in this manner on the MK68564.

V2	V1	0	Interrupt Condition
0	0	0	Ch B Transmit Buffer Empty
0	0	1	Ch B External/status Change
0	1	0	Ch B Receive Character Available
0	1	1	Ch B Special Receive Condition*
1	0	0	Ch A Transmit Buffer Empty
1	0	1	Ch A External/status Change
1	1	0	Ch A Receive Character Available
1	1	1	Ch A Special Receive Condition*

* Special Receive Conditions : Parity Error, Rx Overrun Error, Framing Error (Async), End of Frame (SDLC).

D1 : Transmit Interrupt Enable

When this bit is set to a one, the transmitter will request an interrupt whenever the transmit buffer becomes empty. When this bit is zero, no transmitter interrupts will be requested.

D0 : External/Status Interrupt Enable

When this bit is set to a one, an interrupt will be requested by the external/status logic on any of the following occurrences : a transition (high-to-low or low-to-high) on the DCD, CTS, or SYNC input pins, a break/abort condition that has been detected and terminated, or at the beginning of CRC transmission when the Transmit Underrun/EOM latch in Status Register 0 becomes set. When this bit is zero, no External/Status interrupts will occur.

If this bit is set when an External/Status condition is pending, an interrupt will be requested. It is recommended that a Reset External/Status Interrupts command (Command 2 in the Command Register) be issued prior to enabling External/Status interrupts.

SYNC WORD REGISTER 1 (SYNC 1)

This register is programmed to contain the transmit sync character in the Monosync mode, the first eight bits of the 16-bit sync character in the Bysinc mode, or the transmit sync character in the External Sync mode. This register is not used in Asynchronous mode. In the SDLC mode, this register is programmed to contain the secondary address field used to compare against the address field of the SDLC frame. The SIO does not automatically transmit the station address at the beginning of a response frame. This register is reset to "00H" by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
SYNC/ SDLC7	SYNC/ SDLC6	SYNC/ SDLC5	SYNC/ SDLC4	SYNC/ SDLC3	SYNC/ SDLC2	SYNC/ SDLC1	SYNC/ SDLC0

SYNC WORD REGISTER 2 (SYNC 2)

This register is programmed to contain the receive sync character in the Monosync mode, the last eight bits of the 16-bit sync character in the Bisync mode, or a flag character (01111110) in the SDLC mode. This register is not used in the External Sync mode and the Asynchronous mode. This register is reset to "00H" by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
SYNC/ SDLC 15	SYNC/ SDLC 14	SYNC/ SDLC 13	SYNC/ SDLC 12	SYNC/ SDLC 11	SYNC/ SDLC 10	SYNC/ SDLC 9	SYNC/ SDLC 8

RECEIVER CONTROL REGISTER (RCVCTL)

This register contains the control bits and parameters for the receiver logic. This register is reset to "00H" by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
RX BITS CHAR 1	RX BITS CHAR 0	RX AUTO ENAB.	HUNT MODE	RX CRC ENAB.	ADDR. SEARCH	STRIP SYNC	RX ENABLE

D7, D6 : Receiver Bits/Character 1 and 0

The state of these two bits determines the number of bits to be assembled as a character in the received serial data stream. If Parity is enabled, one additional bit will be added to each character. The number of bits per character can be changed while a character is being assembled but only before the number of bits currently programmed is reached. All data is right-justified in the shift register and transferred to the receive data FIFO in 8-bit groups.

In Asynchronous mode, transfers are made at character boundaries, and all unused bits of character are set to a one. In Synchronous modes and SDLC mode, an 8-bit segment of the serial data stream is transferred to the data FIFO when the internal counter reaches the number of bits per character programmed. For less than eight bits per character, no parity, the MSB bit(s) of the first transfer will be the LSB bit(s) of the next transfer.

RX BITS CHAR 1	RX BITS CHAR 0	Bits/character (no parity)	Bits/character (parity)
0	0	5	6
0	1	6	7
1	0	7	8
1	1	8	9

D5 : Receiver Auto Enables

When this bit is set to a one, and the Receiver Enable bit is also set, a Low on the DCD input pin becomes the enable for the receiver. When this bit is zero, the DCD pin is simply an input to the SIO, and its status is displayed in Status Register 0.

D4 : Enter Hunt Mode

This bit, when written to a one, rearms the receiver synchronization logic and forces the comparison of the received bit stream to the ontents of Sync Word Register 1 and/or Sync Word Register 2, depending upon which Synchronous mode is selected, until bit synchronization is achieved. The SIO automatically enters the Hunt mode after a channel or hardware reset, after an Abort condition is detected, or when the receiver is disabled. When the Hunt mode is entered, the Hunt/Sync bit in Status Register 0 is set to a one. When synchronization is achieved, the Hunt/Sync bit is reset to a zero. If External/Status interrupts are enabled, an interrupt request will be generated on both transitions of the Hunt/Sync bit. Enter Hunt Mode has no affect in Asynchronous modes. This bit is not latched and will always be read as a zero.

D3 : Receiver CRC Enable

This bit, when set to a one in a Synchronous mode other than SDLC, is used to initiate CRC calculation at the beginning of the last byte transferred from the receiver shift register to the receive data FIFO. This operation occurs independently of the number of bytes in the receive data FIFO. As long as this bit is set, CRC will be calculated on all characters received (data or sync). When a particular byte is to be excluded from CRC calculation, this bit should be reset to a zero before the next byte is transferred to the receive data FIFO. If this feature is used, care-must be taken to ensure that eight bits per character are selected in the reciever because of an inherent eight-bit delay from the receiver shift register to the CRC checker.

When this bit is set to a one in SDLC mode, the SIO will calculate CRC on all bits between the opening and closing flags. There is no delay from the receiver shift register to the CRC checker in SDLC mode. This bit is ignored in Asynchronous modes.

D2 : Address Search Mode

Setting this bit to a one in SDLC mode forces the comparison of the first non-flag character of a frame with the address programmed in Sync Word Register 1 or the global address (11111111). If a match does not occur, the frame is ignored, and the receiver remains idle until the next frame is detected. No receiver interrupts can occur in this mode, unless there is an address match. This bit is ignored in all modes except SDLC.

D1 : Sync Character Load Inhibit

When this bit is set to a one in any Synchronous mode except SDLC, the SIO compares the byte in Sync Word Register 1 with the byte about to be loaded into the receiver data FIFO. If the two bytes are equal, the load is inhibited, and no receiver interrupt will be generated by this character. CRC calculation is performed on all bytes, whether they are loaded into the data FIFO or not, when the receiver CRC is enabled. Note that the register used in the comparison contains the transmit sync character in Monosync and External sync modes. This bit is ignored in SDLC mode because all flag characters are automatically striped in this mode without performing CRC calculations on them.

If this bit is set to a one in Asynchronous modes, any character received matching the contents of Sync Word Register 1 will not be loaded into the receive data FIFO, and no receiver interrupt will be generated for the character.

D0 : Receiver Enable

When this bit is set to a one, receiver operation begins if Rx Auto Enables mode is not selected. This bit should be set only after all receiver parameters are established, and the receiver is completely initialized. When this bit is zero, the receiver is disabled ; the receiver CRC checker is reset, and the receiver is in the Hunt mode.

TRANSMITTER CONTROL REGISTER (XMTCTL)

This register contains the control bits and parameters for the transmitter logic. This register is reset to "00H" by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
TX BITS CHAR 1	TX BITS CHAR 0	TX AUTO ENABLES	SEND BREAK	TX CRC ENABLE	DTR	RTS	TX ENABLE

D7, D6 Transmit Bits/Character 1 and 0

The state of these two bits determine the number of bits in each byte transferred from the transmit buffer to the transmit shift register. All data written to the transmit buffer must be right-justified with the least-significant bits first. The Five Or Less mode allows transmission of one to five bits per character ; however, the CPU should format the data characters as shown. If Parity is enabled, one additional bit per character will be transmitted.

TX BITS/ CHAR 1	TX BITS/ CHAR 0	Bits/character (no parity)
0	0	Five or Less
0	1	6
1	0	7
1	1	8

D7	D6	D5	D4	D3	D2	D1	D0	Five or Less
1	1	1	1	0	0	0	D	Sends One Data Bit
1	1	1	0	0	0	D	D	Sends Two Data Bits
1	1	0	0	0	D	D	D	Sends Three Data Bits
1	0	0	0	D	D	D	D	Sends Four Data Bits
0	0	0	D	D	D	D	D	Sends Five Data Bits

D5 : Transmit Auto Enables

When this bit is set to a one, and the Transmit Enable bit is also set, a Low on the CTS input pin will enable the transmitter. When this bit is zero, the CTS pin is simply an input to the SIO, and its status is displayed in Status Register 0.

D4 : Send Break

When set to a one, this bit immediately forces the Transmit Data output pin (TxD) to a spacing condition (continuous 0's), regardless of any data being transmitted at the time. This bit functions, whether the transmitter is enabled or not. When this bit is reset to zero, the transmitter will continue to send the contents of the transmit shift register. The shift register may contain sync characters, data characters, or all ones.

D3 : Transmitter CRC Enable

This bit determines if CRC calculations are performed on a transmitted data character. If this bit is a one at the time a character is loaded from the transmit buffer to the transmit shift register, CRC is calculated on the character. CRC is not calculated on any automatically inserted sync characters. CRC is not automatically appended to the end of a message unless this bit is set, and the Transmit Underrun/EOM status bit in Status Register 0 is reset when a Transmit Underrun condition occurs. If this bit is a zero when a character is loaded from the transmit buffer into the transmit shift register, no CRC calculations are performed on the character. This bit is ignored in Asynchronous modes.

D2 : Data Terminal Ready (DTR)

This is the control bit for the DTR output pin. When this bit is set to a one, the DTR pin goes Low : when this bit is reset to a zero, the DTR pin goes High.

D1 : Request To Send (RTS)

This is the control bit for the RTS output pin. In Synchronous modes, when this bit is set to a one, the RTS pin goes Low ; when this bit is reset to a zero, the RTS pin goes High. In Asynchronous modes, when this bit is set, the RTS pin goes Low ; when this bit is reset, the RTS pin will go High only after all the bits of the character are transmitted, and the transmit buffer is empty.

D0 : Transmitter Enable

Data is not transmitted until this bit is set to a one, until the Send Break bit is reset and, if Tx Auto Enables mode is selected, until the CTS pin is Low. To transmit sync or flag characters in Synchronous modes, this bit has to be set when the transmit buffer is empty. Data or sync characters in the process of being transmitted are completely sent if this bit is reset to zero after transmission has started. If this bit is reset during the transmission of a CRC character, sync or flag characters are sent instead of the CRC character.

STATUS REGISTER 0 (STAT 0) READ ONLY

This register contains the status of the receive and transmit buffers and the status bits for the five sources of External/Status interrupts.

D7	D6	D5	D4	D3	D2	D1	D0
BREAK/ ABORT	UNDERRUN /EOM	CTS	HUNT/ SYNC	DCD	TX BUFR EMPTY	INTERPT PENDING	RX CHAR AVAIL

D7 : Break/Abort

This bit is reset by a channel or hardware reset. In Asynchronous modes, this bit is set when a Break sequence (null character plus framing error) is detected in the received data stream. An External/Status interrupt, if enabled, is generated when Break is detected. The interrupt service routine must issue a Reset External/Status Interrupt command (Command 2) to the SIO, so the break detection logic can recognize the termination of the Break sequence.

The Break/Abort bit is reset to a zero when the termination of the Break sequence is detected in the incoming data stream. The termination of the Break sequence also causes the generation of an External/Status interrupt. Command 2 must be issued to enable the break detection logic to look for the next Break sequence. A single extraneous null character is present in the receiver after the termination of a break ; it should be read and discarded.

In SDLC mode, this bit is set by the detection of an Abort sequence (seven or more ones) in the received data stream. The External/Status Interrupt is handled the same way as in the case of a Break sequence. The Break/Abort bit is not used in the other Synchronous modes.

D6 : Transit Underrun/EOM

This bit is set to a one following a hardware or channel reset, when the transmitter is disabled or when a Send Abort command (Command 1) is issued. This bit can only be reset by the Reset Transmit Underrun/EOM Latch command in the Command Register. This bit is used to control the transmission of CRC at the end of a message in Synchronous modes. When a transmit underrun condition occurs and this bit is low. CRC will be appended to the end of the transmission, and this bit will be set. Only the 0-to-1 transition of this bit causes an External/Status interrupt, when enabled. This bit is not used in Asynchronous modes.

D5 : Clear To Send (CTS)

This bit indicates the inverted state of the CTS input pin at the time of the last change of any of the five External/Status bits. Any transition of the CTS input causes the CTS bit to be latched and generates an External/Status interrupt request, if enabled. To read the current state of the CTS pin, this bit must be read immediately following a Reset External/Status Interrupts command (command 2).

D4 : Hunt/Sync

In Asynchronous modes, this bit indicates the inverted state of the SYNC input pin at the time of the last change of any of the five External/Status bits. Any transition of the SYNC input causes the Hunt/Sync bit to be latched and generates an External/Status interrupt request, if enabled. To read the current state of the SYNC pin, this bit must be read immediately following a Reset External/Status Interrupt command (command 2).

In External sync mode, the SYNC pin is used by external logic to signal character synchronization is achieved, the SYNC pin is driven Low on the second rising edge of the Receive Clock (RxC) on which the last bit of the sync character was received. Once the SYNC pin is Low, it should be held Low until the end of the message and the driven back High. Both transitions on the SYNC pin cause External/Status interrupt requests, if enabled. The inverted state of the SYNC pin is indicated by this bit.

In Monosync, Bisync, and SDLC modes, this bit indicates when the receiver is in the Hunt mode. This bit is set to a one following a hardware ir channel reset, after the Enter Hunt Mode bit is written High, when the receiver is disabled, or when an Abort sequence (SDLC mode) is detected. This bit will remain in this state until character synchronization is achieved. External/Status interrupt requests will be generated on both transitions of the Hunt/Sync bit.

D3 : Data Carrier Detect (DCD)

This bit indicates the inverted state of the DCD input pin at the time of the last change of any of the five External/Status bits. Any transition of the DCD input causes the DCD bit to be latched and generates an External/Status interrupt request, if enabled. To read the current state of the DCD pin, this bit must be read immediately following a Reset External/Status Interrupts command (command 2).

D2 : Transmit Buffer Empty

This bit is set to a one, when the transmit buffer becomes empty, and when the last CRC bit is transmitted in Synchronous or SDLC modes. This bit is reset when the transmit buffer is loaded or while the CRC character is being sent in Synchronous or SDLC modes. This bit is set to a one following a hardware or channel reset.

D1 : Interrupt Pending

Any interrupt condition, pending in the interrupt control logic for this channel, will set this bit to a one. This bit is reset to zero by a hardware channel reset, or when all the interrupt conditions are cleared.

D0 : Receive Character Available

This bit is set to a one when a character becomes available in the receive data FIFO. This bit is reset to zero when the receive data FIFO (receive buffer) is read, or by a hardware or channel reset.

STATUS REGISTER 1 (STAT 1)
READ ONLY

This register contains the Special Receive Condition status bits and the Residue codes for the I-field in the SDLC receive mode. The All Sent bit is set High, and all other bits are reset to a Low by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
END OF FRAME	CRC/ FRAME ERROR	RX OVER-RUN ERR	PARITY ERROR	RESIDUE CODE 2	RESIDUE CODE 1	RESIDUE CODE 0	ALL SENT

D7 : End Of Frame (SDLC)

This bit is used only in SDLC mode. When set to a one, this bit indicates that a valid closing flag has been received and that the CRC/Framing Error bit and Residue codes are valid. If receiver interrupts are enabled, a Special Receive Condition interrupt will also be generated. This bit can be reset by issuing an Error Reset command (command 6). This bit is also updated by the first character of the following frame. This bit is a zero in all modes except for SDLC.

D6 : CRC/Framing Error

In Asynchronous modes, if a Framing Error occurs, this bit is set to a one for the receive character in which the framing error occurred. When this bit is set to a one, a Special Receive Condition interrupt will be requested, if receiver interrupts are enabled.

Detection of a Framing Error adds an additional one-half bit time to the character time, so that the Framing Error is not interpreted as a new start bit.

In Synchronous and SDLC modes, this bit indicates the result of comparing the received CRC value to the appropriate check value. A zero indicates that a match has occurred. This bit is usually set since most bit combinations result in a non-zero CRC, except for a correctly completed message. Receiver interrupts are not requested by the CRC Error bit.

The CRC/Framing bit is not latched in any receiver mode. It is always updated when the next character is received. An Error Reset command (command 6) will always reset this bit to zero.

D5 : Receive Overrun Error

This bit indicates that the receive data FIFO has overflowed. Only the character that has been written over is flagged with this error. When the character is read, the error condition is latched until reset by the Error Reset command (command 6). If receiver interrupts are enabled, the overrun character and all subsequent characters received, until the Error Reset command is issued, will generate a Special Receive Condition interrupt request.

D4 : Parity Error

When parity is enabled, this bit is set to a one for those characters whose parity does not match the programmed sense (even/odd). This bit is latched so that once an error occurs, it remains set until the Error Reset command (command 6) is issued. If parity is a Special Receive Condition, a Parity is a Special Receive Condition, a Parity Error will cause a Special Receive Condition interrupt request on the character containing the error and on all subsequent characters until the Error Reset command is issued.

D3, D2, D1 : Residue Codes 2, 1, and 0

In those cases of the SDLC receive mode, where the I-field is not an integral multiple of the character length, these three bits indicate the length of the residual I-field read in the previous bytes. These codes are meaningful only for the transfer in which the End Of Frame bit is set. This field is set to 000 by a channel or hardware reset and can leave this state only if SDLC mode is selected, and a character is received.

FOR EIGHT BITS PER CHARACTER

Residue Code 2	Residue Code 1	Residue Code 0	I-Field Bits In Previous Byte	I-Field Bits In Second Previous Byte
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

I-Fiel Bits are Right-justified in all Cases.

If a receive character length, different from eight bits, is used for the I-field, a table similar to the previous one may be constructed for each different character length. For no residue (that is, the last character boundary coincides with the boundary of the I-field and CRC field), the Residue codes are as follows :

Bits Per Character	Residue Code 2	Residue Code 1	Residue Code 0
8 Bits Per Character	0	1	1
7 Bits Per Character	0	0	0
6 Bits Per Character	0	1	0
5 Bits Per Character	0	0	1

D0 : All Sent

This bit is only active in Asynchronous modes ; it is always High in Synchronous or SDLC modes. This bit is Low while the transmitter is sending characters : it will go High only after all the bits of the character are transmitted, and the transmit buffer is empty.

DATA REGISTER (DATARG)

The Data Register is actually two separate registers ; a write only register that is the Transmit Buffer, and a read only register that is the Receiver Buffer. The Receiver Buffer is also the top register of a three register stack called the receive data FIFO. The Data Register is not affected by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
DATA 7	DATA 6	DATA 5	DATA 4	DATA 3	DATA 2	DATA 1	DATA 0

TIME CONSTANT REGISTER (TCREG)

This register contains the time constant used by the down counter in the baud rate generator. The time constant may be changed at any time, but the new value does not take effect until the next time the time constant is loaded into the down counter. It is recommended that the BRG be disabled before writing to this register, as no attempt was made to synchronize the loading of a new time constant with the clock used to drive the BRG. This register is reset to "00H" by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0

BAUD RATE GENERATOR CONTROL REGISTER (BRGCTL)

This register contains the control bits used to program the baud rate generator and to select the BRG output mode. This register is reset to "00H" by a channel or hardware reset.

D7	D6	D5	D4	D3	D2	D1	D0
				RxC INT/EXT	TxC INT/EXT	DIVIDE BY 64/4	BRG ENABLE

D7, D6, D5, D4 : Not Used (read as zeros)

D3 : Receiver Clock, Internal/External

This bit determines the direction of the RxC pin. When this bit is set to a one, the RxC pin is the output of the baud rate generator. If this bit is a zero, the RxC pin is an input, and an external source must supply the receiver clock. The receiver clock is always the signal on the RxC pin, except in Loop Mode, when the transmitter clock is connected internally to the receiver clock.

D2 : Transmitter Clock, Internal/External

This bit determines the direction of the TxC pin. When this bit is set to a one, the TxC pin is the output of the baud rate generator. If this bit is a zero, the TxC pin is an input, and an external source must supply the transmitter clock. The transmit clock is always the signal on the TxC pin.

D1 : Divide By 64/4

This bit specifies the minimum BRG input clock cycles to output clock cycle. This minimum occurs when the Time Constant Register is loaded with a "01H" value. When this bit is set to a one, 64 input clocks are required for every output clock. When this bit is a zero, four input clocks are required for every output clock.

D0 : Baud Rate Generator Enable

This bit controls the operation of the baud rate generator. When this bit is set to a one, the BRG will start counting down from the value left in the down counter when this bit was last reset to zero. If the Time Constant Register is loaded while this bit is reset, the new time constant value is loaded immediately into the down counter. The baud rate generator is disabled from counting when this bit is reset.

INTERRUPT VECTOR REGISTER (VECTRG)

This register is used to hold a vector that is passed to the CPU during an interrupt acknowledge cycle. This register can also be accessed through a read/write cycle. If the Status Affects Vector bit in the Interrupt Control Register is disabled, the value programmed into the Vector Register will be passed to the CPU during an interrupt acknowledge cycle or a read cycle. If the Status Affects Vector bit in either channel is enabled, the lower three bits of this register are modified, according to the table listed in the Interrupt Control Register description. With Status Affects Vector on, and no interrupt pending in the SIO, the lower three bits will be read as 011. Only one Vector Register exists in the SIO, but it can be accessed through either channel. This register is reset to "0FH" by a hardware reset only.

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2 *	V1 *	V0 *

* Variable if Status Affects Vectors is Enabled.

MK68564 ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
T_A	Temperature Under Bias	– 25 to 100	°C
T_{stg}	Storage Temperature	– 65 to 150	°C
V_I	Voltage on Any Pin with Respect to Ground	– 3 to 7	V
P_D	Power Dissipation	1.5	W

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS
(V_{CC} = 5.0V ± 5%, T_A = 0 to 70°C)

Symbol	Parameter	Min.	Max.	Unit.
V_{IH}	Input High Voltage ; all Inputs	V_{SS} + 2.0	V_{CC}	V
V_{IL}	Input Low Voltage ; all Inputs	V_{SS} – 0.3	V_{SS} + 0.8	V
I_{LL}	Power Supply Current ; Outputs Open		190	mA
I_{IN}	Input Leakage Current (V_{IN} = 0 to 5.25)		± 10	µA
I_{TSI}	Three-state Input Current $\overline{\text{DTACK}}$, D0-D7, $\overline{\text{SYNC}}$, $\overline{\text{TxC}}$, $\overline{\text{RxC}}$ 0 < V_{IN} < V_{CC}, $\overline{\text{INTR}}$		20 ± 10	µA µA
V_{OH}	Output High Voltage (I_{LOAD} = – 400 µA, V_{CC} = MIN) $\overline{\text{DTACK}}$, D0-D7 (I_{LOAD} = – 150 µA, V_{CC} = MIN) All Other Outputs (except XTAL2 & $\overline{\text{INTR}}$)*	V_{SS} + 2.4		V
V_{OL}	Output Low Voltage (I_{LOAD} = 5.3mA, V_{CC} = MIN) $\overline{\text{INTR}}$, $\overline{\text{DTACK}}$, D0-D7 (I_{LOAD} = 2.4mA, V_{CC} = MIN) All Other Outputs (except XTAL2)*		0.05	V

* XTAL2 Special

INTR (Open drain)

CAPACITANCE
T_A = 25°C, F = 1MHz Unmeasured Pins Returned to Ground.

Symbol	Parameter	Test Conditions	Max.	Unit.
C_{IN}	Input Capacitance $\overline{\text{CS}}$, $\overline{\text{IACK}}$ All Others	Unmeasured Pins Returned to Ground	15 10	pF pF
C_{OUT}	Tri-state Output Capacitance		10	pF

AC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5.0 VDC ± 5%, GND = 0 VDC, T_A = 0 to 70°C)

Number	Parameter	4.0 MHz		5.0 MHz		Unit	Notes
		Min.	Max.	Min.	Max.		
1	CLK Period	250	1000	200	1000	ns	
2	CLK Width High	105		80		ns	
3	CLK Width Low	105		80		ns	
4	CLK Fall Time		30		30	ns	
5	CLK Rise Time		30		30	ns	
6	$\overline{\text{CS}}$ Low to CLK High (setup time)	0		0		ns	1
7	A1-A5 Valid to $\overline{\text{CS}}$ Low (setup time)	0		0		ns	
8	DATA Valid to $\overline{\text{CS}}$ Low (write cycle)	0		0		ns	
9	$\overline{\text{CS}}$ Width High	50		50		ns	1
10	$\overline{\text{DTACK}}$ Low to A1-A5 Invalid (hold time)	0		0		ns	
11	$\overline{\text{DTACK}}$ Low to DATA Invalid (write cycle hold time)	0		0		ns	
12	$\overline{\text{CS}}$ High to $\overline{\text{DTACK}}$ High (delay)		55		50	ns	
13	CLK High to $\overline{\text{DTACK}}$ Low		320		295	ns	
14	R/$\overline{\text{W}}$ Valid to $\overline{\text{CS}}$ Low (setup time)	0		0		ns	
15	$\overline{\text{DTACK}}$ Low to R/$\overline{\text{W}}$ Invalid (hold time)	0		0		ns	
16	CLK Low to DATA Out		450		450	ns	
17	$\overline{\text{CS}}$ High to DATA Out Invalid (hold time)	0		0		ns	11
18	$\overline{\text{CS}}$ High to $\overline{\text{DTACK}}$ High Impedance		105		100	ns	
19	$\overline{\text{DTACK}}$ Low to $\overline{\text{CS}}$ High	0		0		ns	
20	DATA Valid to $\overline{\text{DTACK}}$ Low	70		70		ns	
21	$\overline{\text{IACK}}$ Width High	50		50		ns	1
22	$\overline{\text{IACK}}$ Low to CLK High (setup time)	0		0		ns	1
23	CLK Low to $\overline{\text{INTR}}$ Disabled		410		410	ns	2
24	CLK Low to DATA Out		330		330	ns	2
25	$\overline{\text{DTACK}}$ Low to $\overline{\text{IACK}}$, $\overline{\text{IEI}}$, High	0		0		ns	
26	$\overline{\text{IACK}}$ High to $\overline{\text{DTACK}}$ High	55		50		ns	
27	$\overline{\text{IACK}}$ High to $\overline{\text{DTACK}}$ High Impedence		105		100	ns	
28	$\overline{\text{IACK}}$ High to DATA Out Invalid (hold time)	0		0		ns	
29	DATA Valid to $\overline{\text{DTACK}}$ Low	195		195		ns	2
30	CLK Low to $\overline{\text{IEO}}$ Low		220		220	ns	3
31	$\overline{\text{IEI}}$ Low to $\overline{\text{IEO}}$ Low		140		140	ns	3
32	$\overline{\text{IEI}}$ High to $\overline{\text{IEO}}$ High		190		190	ns	4
33	$\overline{\text{IACK}}$ High to $\overline{\text{IEO}}$ High		190		190	ns	4
34	$\overline{\text{IACK}}$ High to $\overline{\text{INTR}}$ Low		200		200	ns	5
35	$\overline{\text{IEI}}$ Low to CLK Low (setup time)	10		10		ns	
36	$\overline{\text{IEI}}$ Low to $\overline{\text{INTR}}$ Disabled		425		425	ns	6
37	$\overline{\text{IEI}}$ Low to DATA Out Valid		225		225	ns	6
38	DATA Out Valid to $\overline{\text{DTACK}}$ Low	55		55		ns	6
39	$\overline{\text{IACK}}$ High to DATA Out High Impedence		120		90	ns	

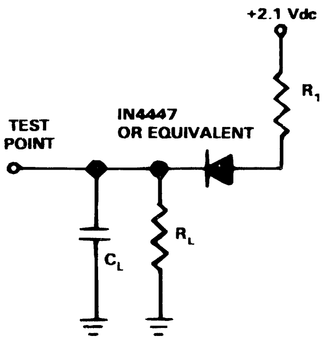
AC ELECTRICAL CHARACTERISTICS (continued)
(V_{CC} = 5.0 VDC ± 5%, GND = 0 VDC, T_A = 0 to 70°C)

Number	Parameter	4.0 MHz		5.0 MHz		Unit	Notes
		Min.	Max.	Min.	Max.		
40	$\overline{\text{CS}}$ HIGH TO DATA Out High Impedence		120		90	ns	
41	$\overline{\text{CS}}$ or $\overline{\text{IACK}}$ High to CLK Low	100		100		ns	7
42	$\overline{\text{TxRDY}}$ or $\overline{\text{RxRDY}}$ Width Low		3		3	CLK's	8, 10
43	CLK High $\overline{\text{TxRDY}}$ or $\overline{\text{RxRDY}}$ Low		300		300	ns	
44	CLK High to $\overline{\text{TxRDY}}$ or $\overline{\text{RxRDY}}$ High		300		300	ns	
	$\overline{\text{IACK}}$ High to $\overline{\text{CS}}$ Low or $\overline{\text{CS}}$ High to IACK Low (not shown)	50		50		ns	1
45	$\overline{\text{CTS}}$, $\overline{\text{DCD}}$, $\overline{\text{SYNC}}$ Pulse Width High	200		200		ns	
46	$\overline{\text{CTS}}$, $\overline{\text{DCD}}$, $\overline{\text{SYNC}}$ Pulse Width Low	200		200		ns	
47	$\overline{\text{TxC}}$ Period	1000	DC	800	DC	ns	9
48	$\overline{\text{TxC}}$ Width Low	180	DC	180	DC	ns	
49	$\overline{\text{TxC}}$ Width High	180	DC	180	DC	ns	
50	$\overline{\text{TxC}}$ Low to TxD Delay (X1 Mode)		300		300	ns	
51	$\overline{\text{TxC}}$ Low to $\overline{\text{INTR}}$ Low Delay	5	9	5	9	CLK's	10
52	$\overline{\text{RxC}}$ Period	1000	DC	800	DC	ns	9
53	$\overline{\text{RxC}}$ Width Low	180	DC	180	DC	ns	
54	$\overline{\text{RxC}}$ Width High	180	DC	180	DC	ns	
55	RxD to $\overline{\text{RxC}}$ High Setup Time (X1 mode)	0		0		ns	
56	$\overline{\text{RxC}}$ High to RxD Hold Time (X1 mode)	140		140		ns	
57	$\overline{\text{RxC}}$ High to $\overline{\text{INTR}}$ Low Delay	10	13	10	13	CLK's	10
58	$\overline{\text{RxC}}$ High to $\overline{\text{SYNC}}$ Low Delay (output modes)	4	7	4	7	CLK's	10
59	$\overline{\text{RESET}}$ Low	1		1		CLK	10
60	XTAL 1 Width High (TTL in)	100		80		ns	
61	XTAL 1 Width Low (TTL in)	100		80		ns	
62	XTAL 1 Period (TTL in)	250	2000	200	2000	ns	
63	XTAL 1 Period (crystal in)	250	1000	200	1000	ns	

Notes :

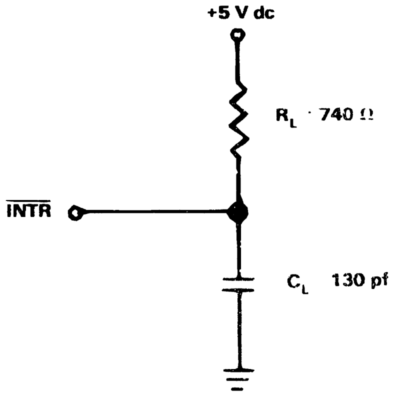
1. This specification only applies if the SIO has completed all operations initiated by the previous bus cycle, when $\overline{\text{CS}}$ or IACK was asserted. Following a read, write, or interrupt acknoledge cycle, all operations are complete within two CLK cycles after the rising edge of CS or $\overline{\text{IACK}}$. If CS or $\overline{\text{IACK}}$ is asserted prior to the completion of the internal operations, the new bus cycle will be postponed.
2. If $\overline{\text{IEI}}$ meets the setup time to the falling edge of CLK, 1 1/2 cycles following the clocking in of $\overline{\text{IACK}}$.
3. No internal interrupt request pending at the start of an interrupt acknoledge cycle.
4. Time starts when first signal goes invalid (high).
5. If an internal interrupt is pending at the end of the interrupt acknoledge cycle.
6. If Note 2 timing is not met.
7. If this spec is met, the delay listed in Note 1 wili be one CLK cycle instead of two.
8. Ready signals will be negated asynchronous to the CLK, if the condition causing the assertion of the signals is cleared.
9. If RxC and TxC are asynchronous to the System Clock, the maximum clock rate into $\overline{\text{RxC}}$ and $\overline{\text{TxC}}$ should be no more than one-fifth the System Clock rate. If $\overline{\text{RxC}}$ and $\overline{\text{TxC}}$ are synchronized to the falling edge of the System Clock, the maximum clock rate into $\overline{\text{RxC}}$ and $\overline{\text{TxC}}$ can be one-fourth the System Clock rate.
10. System Clock.
11. Due to the dynamic nature of the internal data bus, if $\overline{\text{CS}}$ is held low for more than a few hundred milliseconds the read data may go to OOH before the end of the cycle.

Figure 13 : Output Test Load.



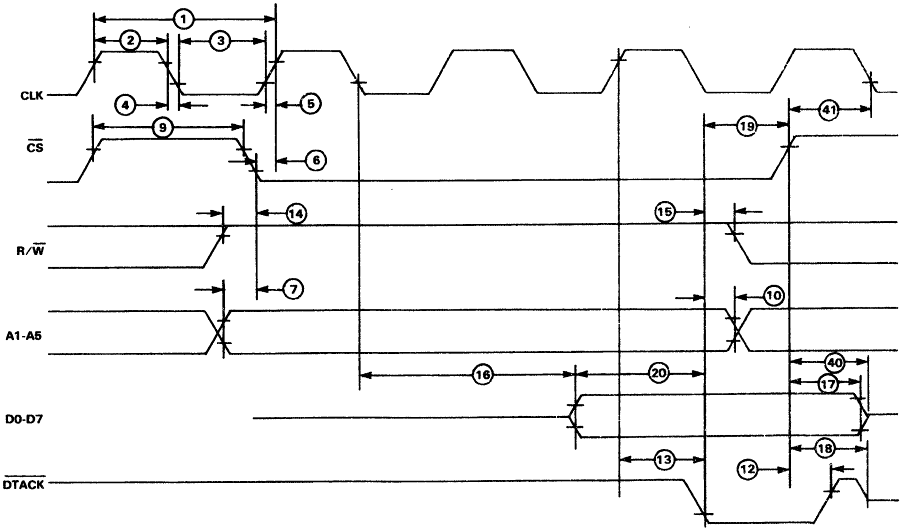
For all Outputs Except	$\overline{\text{DTACK}}$, D0-D7 $\overline{\text{INTR}}$, XTAL2 C_L = 130pf R_L = 16KΩ R_1 = 450Ω
for	$\overline{\text{DTACK}}$, D0-D7 C_L = 130pf R_L = 6KΩ R_1 = 200Ω

Figure 14 : INTR Test Load.



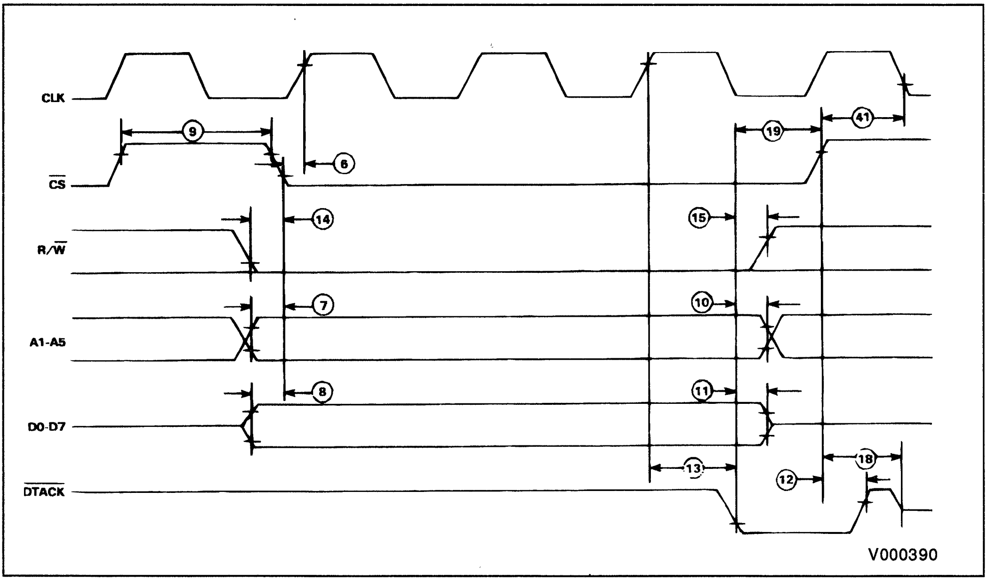
Note : XTAL2 Output Test Load is a Crystal.

Figure 15 : Read Cycle.



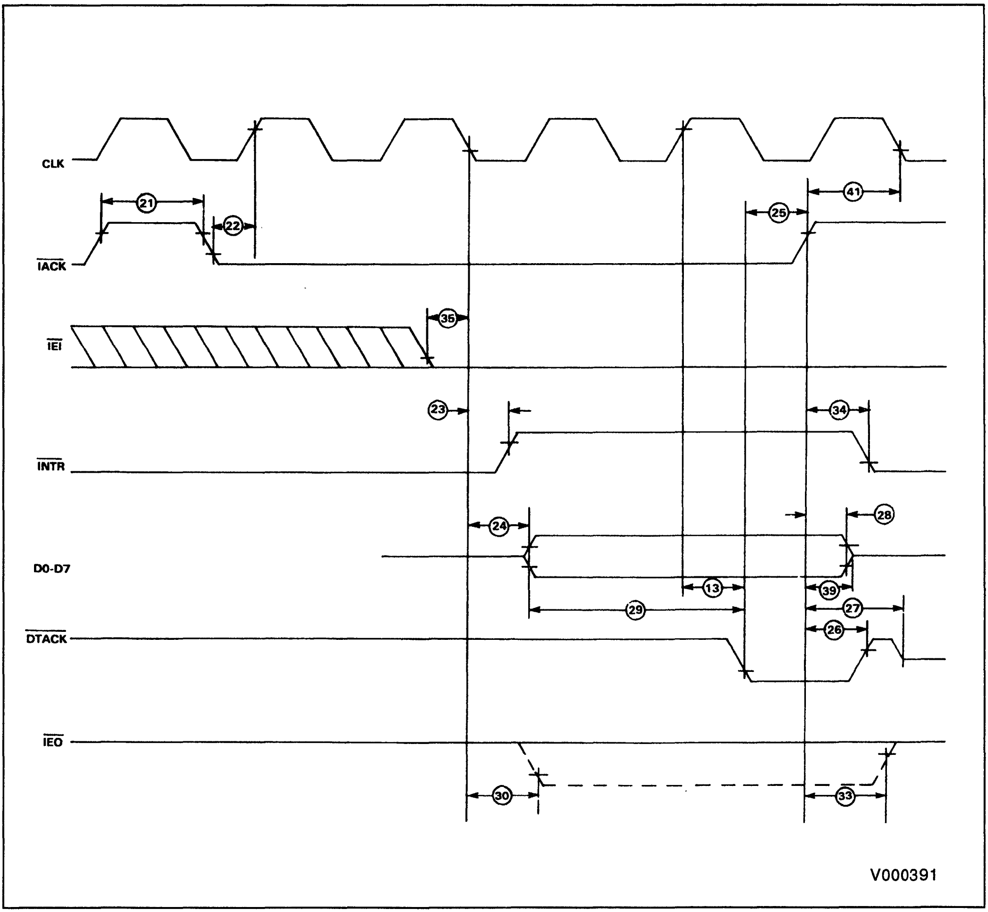
Note : Waveform Measurement for all Inputs and Outputs are Specified at Logic High = 2.0 Volts, Logic Low = 0.8 Volts.

Figure 16 : Write Cycle.



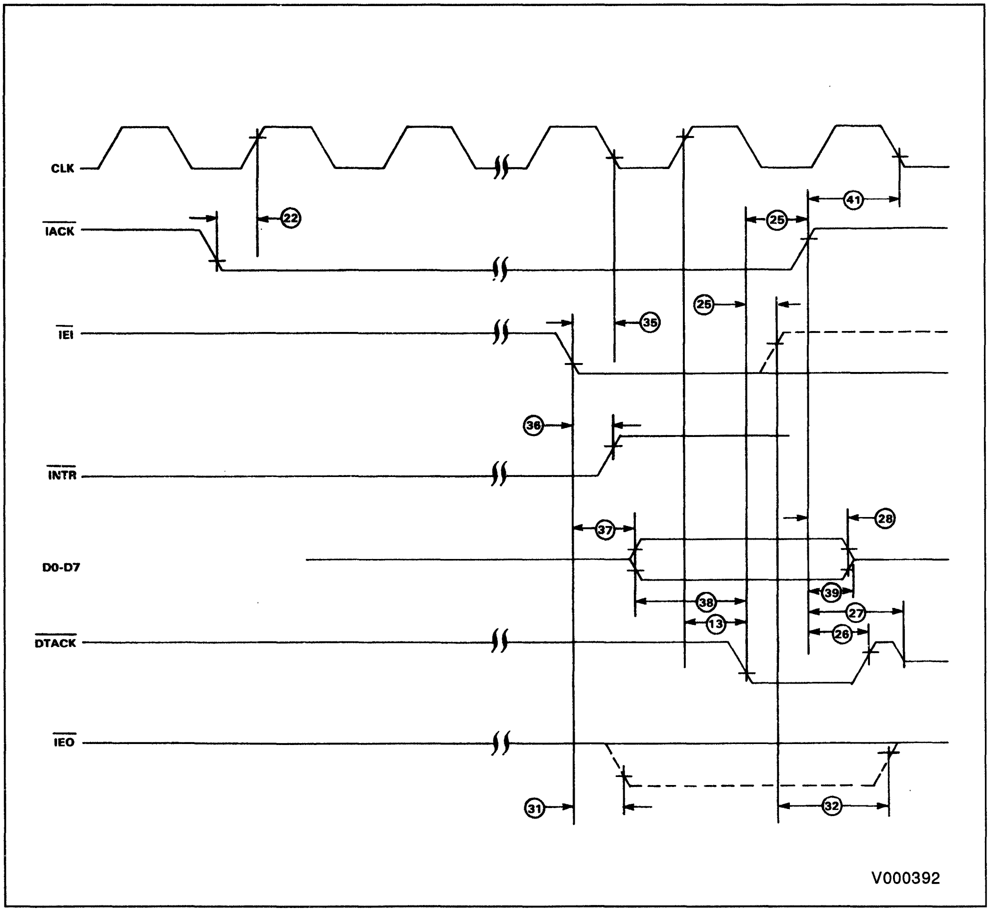
Note : Waveform Measurements for all Inputs and Outputs are Specified at Logic High = 2.0 Volts, Logic Low = 0.8 Volts.

Figure 17 : Interrupt Acknoledge Cycle ($\overline{\text{IEI}}$ low).



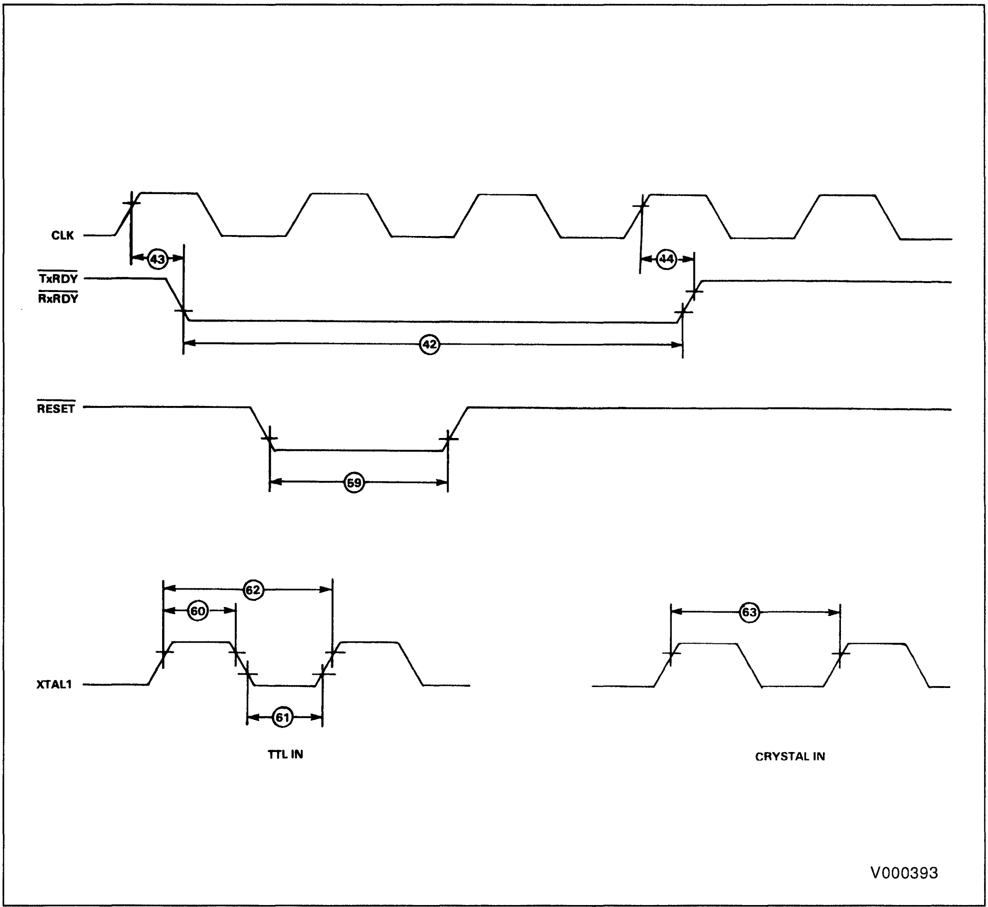
Note : Waveform Measurements for all Inputs and Outputs are Specified at Logic High = 2.0 Volts, Logic Low = 0.8 Volts.

Figure 18 : Interrupt Acknoledge Cycle ($\overline{\text{IEI high}}$).



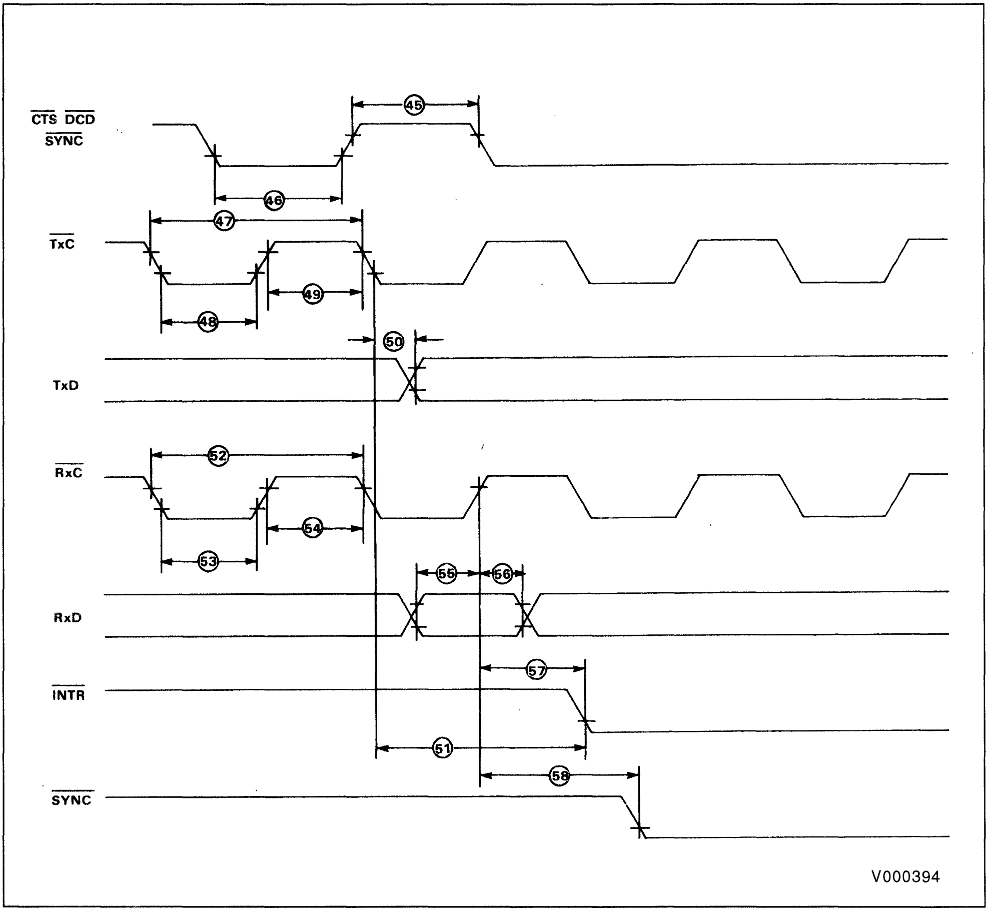
Note : Waveform Measurements for all Inputs and Outputs are Specified at Logic High = 2.0 Volts, Logic Low = 0.8 Volts.

Figure 19 : DMA Interface Timing.



Note : Waveform Measurements for all Inputs and Outputs are Specified at Logic High = 2.0 Volts, Logic Low = 0.8 Volts.

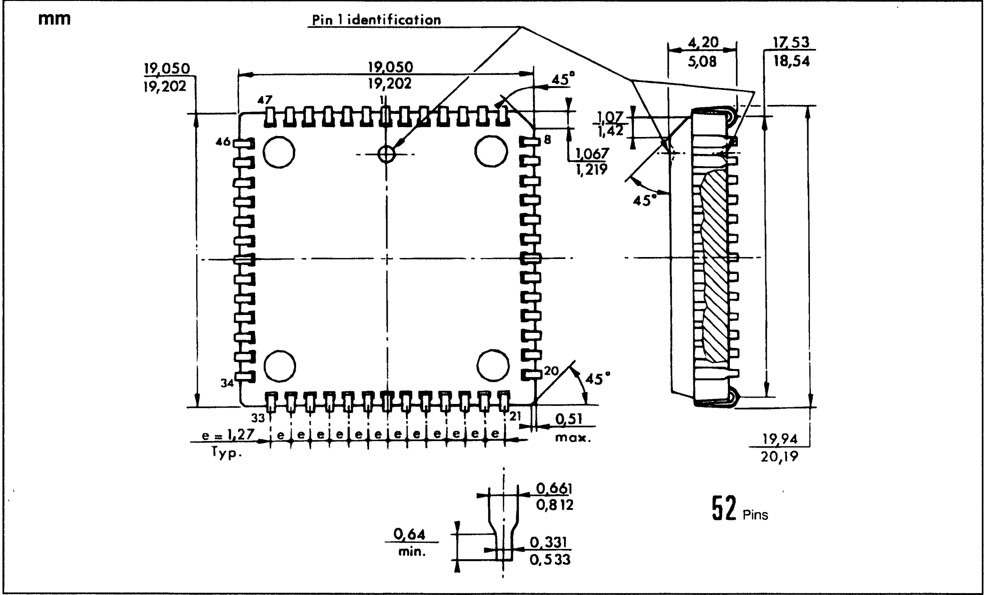
Figure 20 : Serial Interface Timing.



Note : Waveform Measurements for all Inputs and Outputs are Specified at Logic High = 2.0 Volts, Logic Low = 0.8 Volts.

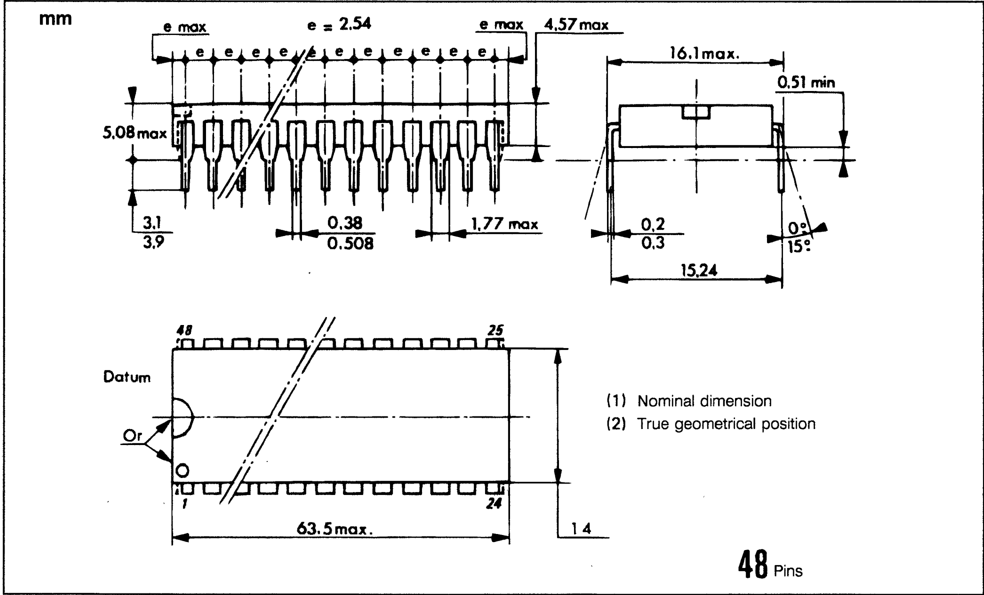
MK68564 52-PIN

Plastic Leader Chip Carrier (Q)



MK68564 48-PIN

Plastic Dual-IN-Line Package



MK68564 ORDERING INFORMATION

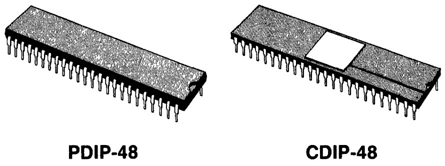
Part No.	Package Type	Max. Clock Frequency	Temperature Range
MK68564N-04	Plastic	4.0 MHz	0° to 70 °C
MK68564N-05	Plastic	5.0 MHz	0° to 70 °C
MK68564Q-04	PLCC	4.0 MHz	0° to 70 °C
MK68564Q-05	PLCC	5.0 MHz	0° to 70 °C

SGS-THOMSON
MICROELECTRONICS

MK68901

MULTI–FUNCTION PERIPHERAL

- 8 INPUT/OUTPUT PINS
  - Individually programmable direction
  - Individual interrupt source capability
    - Programmable edge selection
- 16 SOURCE INTERRUPT CONTROLLER
  - 8 Internal sources
  - 8 External sources
  - Individual source enable
  - Individual source masking
  - Programmable interrupt service modes
    - Polling
    - Vector generation
      - Optional In-service status
  - Daisy chaining capability
- FOUR TIMERS WITH INDIVIDUALLY PROGRAMMABLE PRESCALING
  - Two multimode timers
    - Delay mode
    - Pulse width measurement mode
    - Event counter mode
  - Two delay mode timers
  - Independent clock input
  - Time out output option
- SINGLE CHANNEL USART
  - Full Duplex
  - Asynchronous to 65 kbps
  - Byte synchronous to 1 Mbps
  - Internal/External baud rate generation
  - DMA handshake signals
  - Modem control
  - Loop back mode
- 68000 BUS COMPATIBLE
- 48 PIN DIP OR 52 PIN PLCC



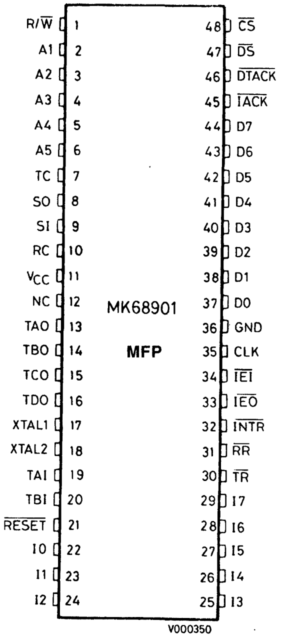
DESCRIPTION

The MK68901 MFP (Multi-Function Peripheral) is a combination of many of the necessary peripheral functions in a microprocessor system.
Included are :

Eight parallel I/O lines
Interrrupt controller for 16 sources
Four timers
Single channel full duplex USART

The use of the MFP in a system can significantly reduce chip count, thereby reducing system cost. The MFP is completely 68000 bus compatible, and 24 directly addressable internal registers provide the

Figure 1 : Pin connections.



necessary control and status interface to the programmer.

The MFP is a derivative of the MK3801 STI, a Z80 family peripheral.

PIN DESCRIPTION

GND : Ground

V_{CC} : +5 volts (± 5%)

$\overline{\text{CS}}$: Chip Select (input, active, low). $\overline{\text{CS}}$ is used to select the MK68901 MFP for accesses to the internal registers. $\overline{\text{CS}}$ and $\overline{\text{IACK}}$ must not be asserted at the same time.

$\overline{\text{DS}}$: Data Strobe (input, active low). $\overline{\text{DS}}$ is used as part of the chip select and interrupt acknowledge functions.

R/$\overline{\text{W}}$: Read/Write (input). R/$\overline{\text{W}}$ is the signal from the bus master indicating whether the current bus cycle is a Read (High) or Write (Low) cycle.

$\overline{\text{DTACK}}$: Data Transfer Acknowledge. (output, active low, tri-stateable) $\overline{\text{DTACK}}$ is used to signal the bus master that data is ready, or that data has been accepted by the MK68901 MFP.

A1-A5 : Address Bus (inputs). The adress bus is used to adress one of the internal registers during a read or write cycle.

D0-D7 : Data Bus (bi-directional, tri-stateable). The data bus is used to receive data from or transmit data to one of the internal registers during a read or write cycle. It is also used to pass a vector during an interrupt acknowledge cycle.

CLK : Clock (input). This input is used to provide the internal timing for the MK68901 MFP.

$\overline{\text{RESET}}$: Device reset. (input, active low). Reset disables the USART receiver and transmitter, stops all timers and forces the timer outputs low, disables all interrupt channels and clears any pending interrupts. The General Purpose Interrupt/I/O lines will be placed in the tri-state input mode. All internal registers (except the timer, USART data registers, and transmit status register) will be cleared.

$\overline{\text{INTR}}$: Interrupt Request (output, active low, open drain). $\overline{\text{INTR}}$ is asserted when the MK68901 MFP is requesting an interrupt. $\overline{\text{INTR}}$ is negated during an interrupt acknowledge cycle or by clearing the pending interrupt(s) through software.

$\overline{\text{IACK}}$: Interrupt Acknowledge (input, active low). $\overline{\text{IACK}}$ is used to signal the MK68901 MFP that the CPU is acknowledging an interrupt. $\overline{\text{CS}}$ and $\overline{\text{IACk}}$ must not be asserted at the same time.

$\overline{\text{IEI}}$: Interrupt Enable In (input, active low). $\overline{\text{IEI}}$ is used to signal the MK68901 MFP that no higher priority device is requesting interrupt service.

$\overline{\text{IEO}}$: Interrupt Enable Out (output, active low). $\overline{\text{IEO}}$ is used to signal lower priority peripherals that neither the MK68901 MFP nor another higher priority peripheral is requesting interrupt service.

10-17 : General Purpose Interrupt I/O lines. These lines may be used as interrupt inputs and/or I/O lines. When used as interrupt inputs, their active edge is programmable. A data direction register is used to define which lines are to be Hi-Z inputs and which lines are to be push-pull TTL compatible outputs.

SO : Serial Output. This is the output of the USART transmitter.

SI : Serial Input. This is the input to the USART receiver.

RC : Receiver Clock. This input controls the serial bit rate of the USART receiver.

TC : Transmitter Clock. This input controls the serial bit rate of the USART transmitter.

$\overline{\text{RR}}$: Receiver Ready. (output, active low) DMA output for receiver, which reflects the status of Buffer Full in port number 15.

$\overline{\text{TR}}$: Transmitter Ready. (output, active low) DMA output for transmitter, which reflects the status of Buffer Empty in port number 16.

TAO,TBO, TCO,TDO: Timer Outputs. Each of the four timers has an output which can produce a square wave. The output will change states each timer cycle ; thus one full period of the timer out signal is equal to two timer cycles. TAO or TBO can be reset (logic "O") by a write to TACR, or TBCR respectively.

XTAL1, XTAL2 : Timer Clock inputs. A crystal can be connected between XTAL1 and XTAL2, or XTAL1 can be driven with a TTL level clock. When driving XTAL1 with a TTL level clock, XTAL2 must be allowed to float.

When using a crystal, external capacitors are required. See figure 33. All chip accesses are independent of the timer clock.

TAI,TBI : Timer A, B inputs. Used when running the timers in the event count or the pulse width measurement mode. The interrupt channels associated with 14 and 13 are used for TAI and TBI, respectively. Thus, when running a timer in the pulse width measurement mode, 14 or 13 can be used for I/O only.

Figure 3 : MK68901 Block Diagram.

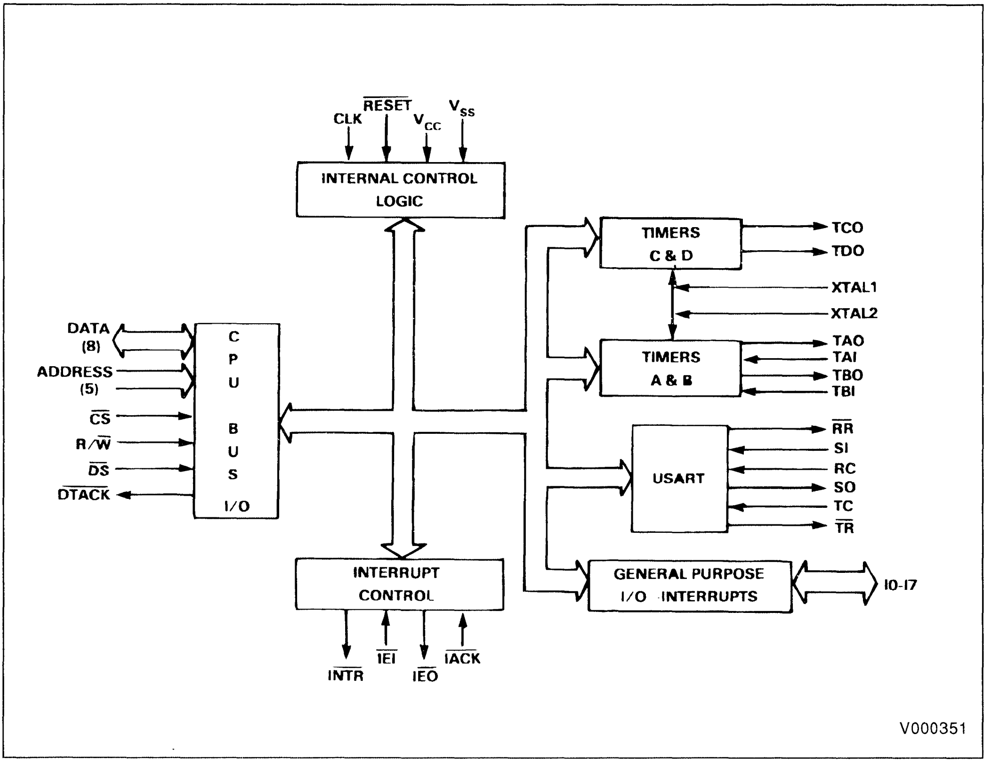


Figure 4 : Register Map.

Address Port N°.	Abbreviation	Register Name
0	GPIP	GENERAL PURPOSE I/O
1	AER	ACTIVE EDGE REGISTER
2	DDR	DATA DIRECTION REGISTER
3	IERA	INTERRUPT ENABLE REGISTER A
4	IERB	INTERRUPT ENABLE REGISTER B
5	IPRA	INTERRUPT PENDING REGISTER A
6	IPRB	INTERRUPT PENDING REGISTER B
7	ISRA	INTERRUPT IN-SERVICE REGISTER A
8	ISRB	INTERRUPT IN-SERVICE REGISTER B
9	IMRA	INTERRUPT MASK REGISTER A
A	IMRB	INTERRUPT MASK REGISTER B
B	VR	VECTOR REGISTER
C	TACR	TIMER A CONTROL REGISTER
D	TBCR	TIMER B CONTROL REGISTER
E	TCDCR	TIMERS C AND D CONTROL REGISTER
F	TADR	TIMER A DATA REGISTER
10	TBDR	TIMER B DATA REGISTER
11	TCDR	TIMER C DATA REGISTER
12	TDDR	TIMER D DATA REGISTER
13	SCR	SYNC CHARACTER REGISTER
14	UCR	USART CONTROL REGISTER
15	RSR	RECEIVER STATUS REGISTER
16	TSR	TRANSMITTER STATUS REGISTER
17	UDR	USART DATA REGISTER

INTERRUPTS

The General Purpose I/O-Interrupt Port (GPIP) provides eight I/O lines that may be operated either as inputs or outputs under software control. In addition, each line may generate an interrupt in either a positive going edge or a negative going edge of the input signal.

The GPIP has three associated registers. One allows the programmer to specify the Active Edge for each bit that will trigger an interrupt. Another register specifies the Data Direction (input or output) associated with each bit. The third register is the actual data I/O register used to input or output data to the port. These three registers are illstrated in figure 5.

The Active Edge Register (AER) allows each of the General Purpose Interrupts to provide an interrupt on either a 1-0 transition or a 0-1 transition. Writing a zero to the appropriate bit of the AER causes the associated input to produce an interrupt on the 1-0 transition. The edge bit is simply one input to an exclusive-or gate, with the other input coming from the input buffer ant the output going to a 1-0 transition detector. Thus, depending upon the state of the input, writing the AER can cause an interrupt-producing transition, which will cause an interrupt on the associated channel, if that channel is enabled. One would then normally configure the AER before enabling interrupts via IERA and IERB.

Note : Changing the edge bit, with the interrupt enabled, may cause an interrupt on that channel.

The Data Direction Register (DDR) is used to define 10-17 as inputs or as outputs on a bit by bit basis. Writing a zero into a bit of the DDR causes the corresponding Interrupt-I/O pin to be a Hi-Z input. Writing a one into a bit of the DDR causes the corresponding pin to be configured as a push-pull output. When data is written into the GPIP, those pins defined as inputs will remain in the Hi-Z state while those pins defined as outputs will assume the state (high or low) of their corresponding bit in the GPIP. When the GPIP is read, the data read will come directly from the corresponding bit of the GPIP register for all pins defined as output, while the data read on all pins defined as inputs will come from the input buffers.

Each individual function in the MK68901 is provided with a unique interrupt vector that is presented to the system during the interrupt acknowledge cycle. The interrupt vector returned during the interrupt acknowledge cycle is shown in figure 6, while the vector register is shown in figure 7.

There are 16 vector addresses generated internally by the MK68901, one for each of the 16 interrupt channels.

The Interrupt Control Registers (figure 8) provide control of interrupt processing for all I/O facilities of the MK68901. These registers allow the programmer to enable or disable any or all of the 16 interrupts, providing masking for any interrupt, and provide access to the pending and in-service status of the interrupt. Optional end-of-interrupt modes are available under software control. All the interrupts are prioritized as shown in figure 9.

Figure 5 : General Purpose I/O Registers.

ACTIVE EDGE REGISTER

PORT 1 (AER)	GPIP 7	GPIP 6	GPIP 5	GPIP 4	GPIP 3	GPIP 2	GPIP 1	GPIP 0	1 RISING 0 FALLING

DATA DIRECTION REGISTER

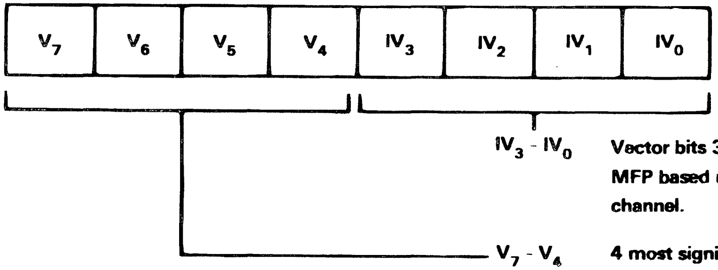
PORT 2 (DDR)	GPIP 7	GPIP 6	GPIP 5	GPIP 4	GPIP 3	GPIP 2	GPIP 1	GPIP 0	1 OUTPUT 0 INPUT

GENERAL PURPOSE I/O DATA REGISTER

PORT 0 (GPIP)	GPIP 7	GPIP 6	GPIP 5	GPIP 4	GPIP 3	GPIP 2	GPIP 1	GPIP 0

V000352

Figure 6 : Interrupt Vector.



IV_3 - IV_0 Vector bits 3-0 supplied by the MFP based upon the interrupting channel.

V_7 - V_4 4 most significant bits. Copied from the vector register.

V000353

Figure 7 : Vector Register.

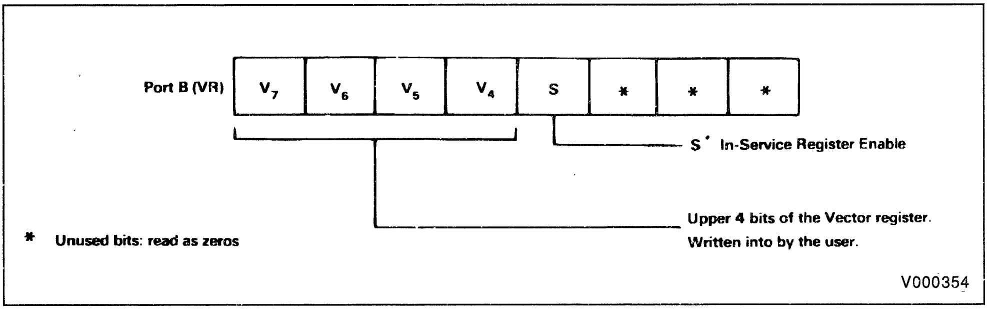

Port B (VR)
V_7
V_6
V_5
V_4
S
*
*
*
S' In-Service Register Enable
Upper 4 bits of the Vector register.
Written into by the user.
* Unused bits: read as zeros
V000354

Figure 8 : Interrupt Control Registers.

INTERRUPT ENABLE REGISTERS

ADDRESS		7	6	5	4	3	2	1	0
PORT 3	A (IERA)	GPIP 7	GPIP 6	TIMER A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TIMER B
PORT 4	B (IERB)	GPIP 5	GPIP 4	TIMER C	TIMER D	GPIP 3	GPIP 2	GPIP 1	GPIP 0

INTERRUPT PENDING REGISTERS

		7	6	5	4	3	2	1	0
PORT 5	A (IPRA)	GPIP 7	GPIP 6	TIMER A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TIMER B
PORT 6	B (IPRB)	GPIP 5	GPIP 4	TIMER C	TIMER D	GPIP 3	GPIP 2	GPIP 1	GPIP 0

WRITING 0 CLEAR
WRITING 1 UNCHANGED

INTERRUPT IN-SERVICE REGISTERS

		7	6	5	4	3	2	1	0
PORT 7	A (ISRA)	GPIP 7	GPIP 6	TIMER A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TIMER B
PORT 8	B (ISRB)	GPIP 5	GPIP 4	TIMER C	TIMER D	GPIP 3	GPIP 2	GPIP 1	GPIP 0

INTERRUPT MASK REGISTERS

		7	6	5	4	3	2	1	0
PORT 9	A (IMRA)	GPIP 7	GPIP 6	TIMER A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	TIMER B
PORT A	B (IMRB)	GPIP 5	GPIP 4	TIMER C	TIMER D	GPIP 3	GPIP 2	GPIP 1	GPIP 0

1 UNMASKED 0 MASKED

V000355

Figure 9 : Interrupt Control Register Definitions

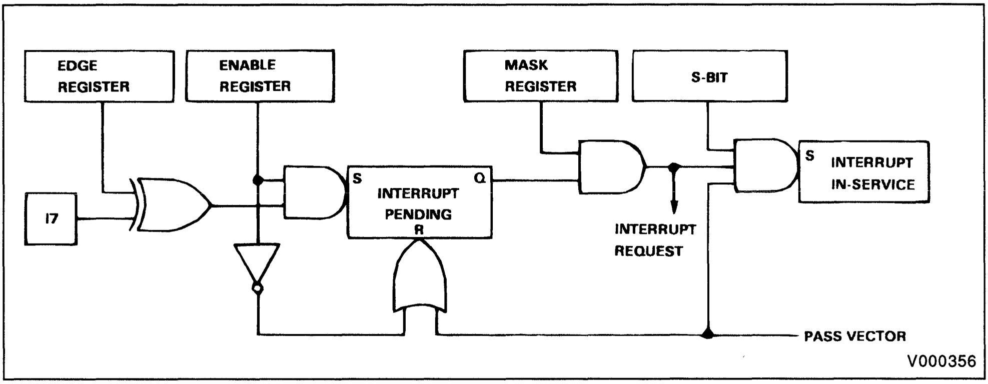
Priority	Channel	Description
HIGHEST	1111	General Purpose Interrupt 7(I7)
	1110	General Purpose Interrupt 6(I6)
	1101	Timer A
	1100	Receive Buffer Full
	1011	Receive Error
	1010	Transmit Buffer Empty
	1001	Transmit Error
	1000	Timer B
	0111	General Purpose Interrupt 5(I5)
	0110	General Purpose Interrupt 4(I4)
	0101	Timer C
	0100	Timer D
	0011	General Purpose Interrupt 3(I3)
	0010	General Purpose Interrupt 2(I2)
	0001	General Purpose Interrupt 1(I1)
LOWEST	0000	General Purpose Interrupt 0(I0)

Interrupts may be either polled or vectored. Each channel may be individual enabled or disabled by writing a one or a zero in the appropriate bit of Interrupt Enable Registers (IERA, IERB - see figure 8 for all registers in this section). When disabled, an interrupt channel is completely inactive. Any internal or external action which would normally produce an interrupt on that channel is ignored and any pending interrupt on that channel will be cleared by disabling that channel. Disabling an interrupt channel has no effect on the corresponding bit in Interrupt In-Service Registers (ISRA, ISRB) ; thus, if the In-service Registers are used and an interrupt is in service on that channel when the channel is disabled, it will remain in service until cleared in the normal manner. IERA and IERB are also readable.

When an interrupt is received on an enabled channel, its corresponding bit in the pending register will be set. When that channel is acknowledged it will pass its vector, and the corresponding bit in the Interrupt Pending Register (IPRA or IRPB) will be cleared. IPRA and IPRB are readable ; thus by polling IPRA and IPRB, it can be determined whether a channel has a pending interrupt. IPRA and IPRB are also writeable and a pending interrupt can be cleared without going through the acknowledge sequence by writing a zero to the appropriate bit. This allows any one bit to be cleared, without altering any other bits, simply by writing all ones except for the bit position to be cleared to IPRA or IPRB. Thus a fully polled interrupt scheme is possible. Note : writing a one to IPRA, IPRB has no effect on the interrupt pending register.

The interrupt mask registers (IMRA and IMRB) may be used to block a channel from making an interrupt request. Writing a zero into the corresponding bit of the mask register will still allow the channel to receive an interrupt and latch it into its pending bit (if that channel is enabled), but will prevent that channel from making an interrupt request. If that channel is causing an interrupt request at the time the corresponding bit in the mask register is cleared, the request will cease. If no other channel is making a request, INTR will go inactive. If the mask bit is reenabled, any pending interrupt is now free to resume its request unless blocked by a higher priority request for service. IMRA and IMRB are also readable . A conceptual circuit of an interrupt channel is shown in figure 10.

Figure 10 : A Conceptual Circuit of an Interrupt Channel.



There are two end-of-interrupt modes : the automatic end-of-interrupt mode and the software end-of-interrupt mode. The mode is selected by writing a one or a zero to the S bit of the Vector Register (VR). If the S bit of the VR is a one, all channels operate in the software end-of-interrupt mode. If the S bit is a zero, all channels operate in the automatic end-of-interrupt mode, and a reset is held on all in-service bits. In the automatic end-of-interrupt mode, the pending bit is cleared when that channel passes its vector. At that point, no further history of that interrupt remains in the MK68901 MFP. In the software end-of-interrupt mode, the in-service bit is set and the pending bit is cleared when the channel passes its vector. With the in-service bit set, no lower priority channel is allowed to request an interrupt or to pass its vector during an acknowledge sequence ; however, a lower priority channel may still receive an interrupt and latch it into the pending bit. A higher priority channel may still request an interrupt and be acknowledged. The in-service bit of a particular channel may be cleared by writing a zero to the corresponding bit in ISRA or ISRB. Typically, this will be done at the conclusion of the interrupt routine just before the return. Thus no lower priority channel will be allowed to request service until the higher priority channel is complete, while channels of still higher priority will be allowed to request service. While the in-service bit is set, a second interrupt on that channel may be received and latched into the pending bit, though no service request will be made in response to the second interrupt until the in-service bit is cleared. ISRA and ISRB may be read at any time. Only a zero may be written into any bit of ISRA and ISRB ; thus the in-service bits may be cleared in software but cannot be set in software. This allows any one bit to be cleared, without altering any other bits, simply by writing all ones except for the bit position to be cleared to ISRA or ISRB, as with IPRA and IPRB.

Figure 11 a : A Conceptual Circuit of the MK68901 MFP Daisy Chaining.

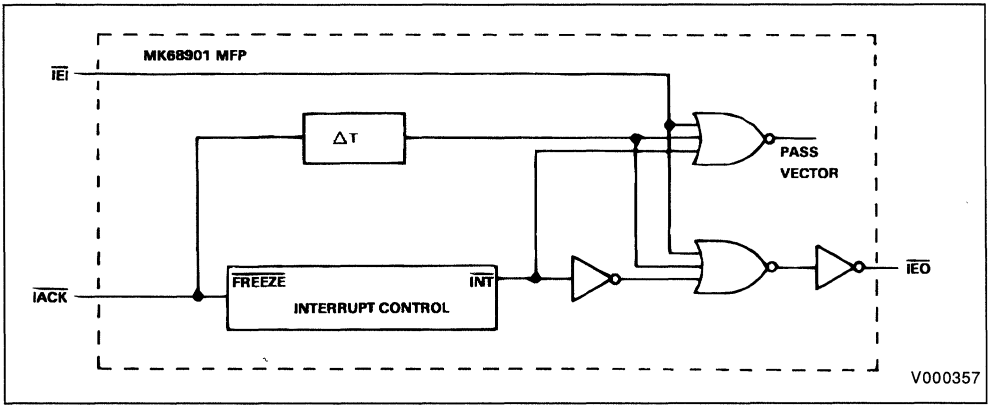
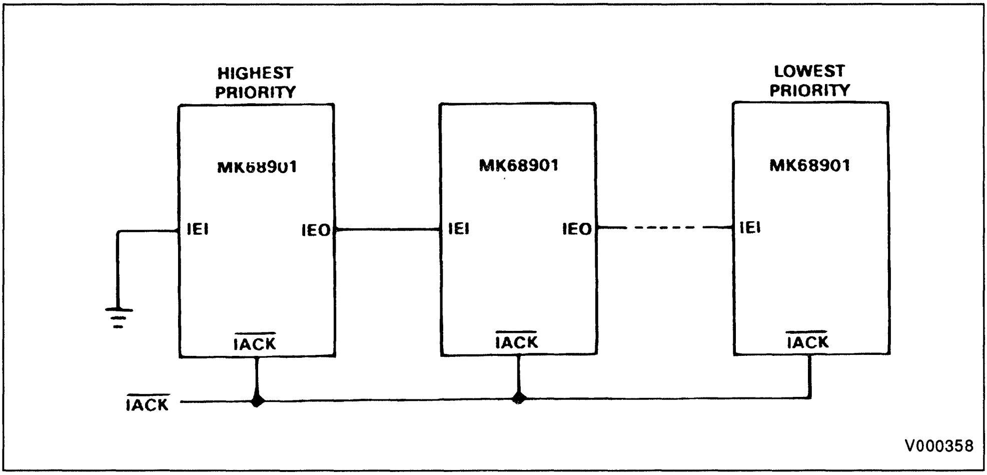


Figure 11 b : Daisy Chaining.



Each interrupt channel responds with a discrete 8-bit vector when acknowledged. The upper four bits of the vector are set by writing the upper four bits of the VR. The four low order bits (bit 3-bit 0) are generated by the interrupting channel.

To acknowledge an interrupt, IACK goes low, the IEI input must go low (or be tied low) and the MK68901 MFP must have an acknowledgeable interrupt pending. The Daisy Chaining capability (figure 11) requires that all parts in a chain have a common IACK. When the common IACK goes low, all parts freeze and prioritize interrupts in parallel. Then priority is passed down the chain, via IEI and IEO, until a part which has a pending interrupt is reached. The part with the pending interrupt, passes a vector, does not propagate IEO, and generates DTACK.

Figure 9 describes the 16 prioritized interrupt channels. As chown, General Purpose Interrupt 7 has the highest priority, while General Purpose Interrupt 0 is assigned the lowest priority. Each of these channels may be reprioritized, in effect, by selectively masking interrupts under software control. The binary numbers under "channel" correspond to the modified bits IV3, IV2, IV1 and IV0, respectively, of the Interrupt Vector for each channel (see figure 6).

Each channel has an enable bit contained in IERA or IERB, a pending latch contained in IPRA or IPRB, a mask bit contained in IMRA or IMRB, and an in-service latch contained in ISRA or ISRB. Additionally, the eight General Purpose Interrupts each have an edge bit contained in the Active Edge Register (AER), a bit to define the line as input or output contained in the Data Direction Register (DDR) and an I/O bit in the General Purpose Interrupt-I/O Port (GPIP).

TIMERS

There are four timers on the MK68901 MFP. Two of the timers (Timer A and Timer B) are full function timers which can perform the basic delay function and can also perform event counting, pulse width measurement, and waveform generation. The other two timers (Timer C and Timer D) are delay timers only. One or both of these timers can be used to supply the baud rate clocks for the USART. All timers are prescaler/counter timers with a common independent clock input (XTAL1, XTAL2). In addition, all timers have a time-out output, function that toggles each time the timer times out.

The four timers are programmed via three Timer Control Registers and four Timer Data Registers. Timers A and B are controlled by the control registers TACR and TBCR, respectively (see figure 12), and by the data registers TADR and TBDR (figure 13). Timers C and D are controlled by the control register TCDCR (see figure 14) and two data registers TCDR and TDDR. Bits in the control registers allow the selection of operational mode, prescale, and control white the data registers are used to read the timer or write into the time constant register. Timer A and B input pins TAI and TBI, are used for the event and pulse width modes for timers A and B.

With the timer stopped, no counting can occur. The timer contents will remain unaltered while the timer is stopped (unless reloaded by writing the Timer Data Register), but any residual count in the prescaler will be lost.

Figure 12 : Timer A and B Control Registers.

Port C (TACR)	*	*	*	TIMER A RESET	AC_3	AC_2	AC_1	AC_0
Port D (TBCR)	*	*	*	TIMER B RESET	BC_3	BC_2	BC_1	BC_0

C_3	C_2	C_1	C_0	
0	0	0	0	Timer Stopped
0	0	0	1	Delay Mode, : 4 Prescale
0	0	1	0	Delay Mode, 10 Prescale
0	0	1	1	Delay Mode, : 16 Prescale
0	1	0	0	Delay Mode, : 50 Prescale
0	1	0	1	Delay Mode, : 64 Prescale
0	1	1	0	Delay Mode, : 100 Prescale
0	1	1	1	Delay Mode, : 200 Prescale
1	0	0	0	Event Count Mode
1	0	0	1	Pulse Width Mode, : 4 Prescale
1	0	1	0	Pulse Width Mode, : 10 Prescale
1	0	1	1	Pulse Width Mode, 16 Prescale
1	1	0	0	Pulse Width Mode, : 50 Prescale
1	1	0	1	Pulse Width Mode, : 64 Prescale
1	1	1	0	Pulse Width Mode, 100 Prescale
1	1	1	1	Pulse Width Mode, : 200 Prescale

V000359

* Unused bits : read as zeros.

In the delay mode, the prescaler is always active. A count pulse will be applied to the main timer unit each time the prescribed number of timer clock cycles has elapsed. Thus, if the prescaler is programmed to divide by ten, a count pulse will be applied to the main counter every ten cycles of the timer clock.

Each time a count pulse is applied to the main counter, it will decrement its contents. The main counter is initially loaded by writing to the Timer Data Register. Each count pulse will cause the current count to decrement. When the timer has decremented down to "01" , the next count pulse will not cause it to decrement to "00". Instead, the next count pulse will cause the timer to be reloaded from the Timer Data Register. Additionally, a "Time out" pulse will be produced. This Time Out pulse is coupled to the timer interrupt channel, and, if that channel is enabled, an interrupt will be produced. The Time Out pulse is also coupled to the timer output pin and will cause the pin to change states. The output will remain in this new state until the next Time Out pulse occurs. Thus the output will complete one full cycle for each two Time Out pulses.

If, for example, the prescaler were programmed to divide by ten, and the Timer Data Register were loaded with 100 (decimal), the main counter would decrement once for every ten cycles of the timer clock. A Time Out pulse will occur (hence an interrupt if that channel is enabled) every 1000 cycles of the timer clock, and the timer output will complete one full cycle every 2000 cycles of the timer clock.

The main counter is an 8-bit binary down counter. It may be read at any time by reading the Timer Data Register. The information read is the information last clocked into the timer read register when the DS pin had last gone high prior to the current read cycle. When written, data is loaded into the Timer Data Register, and the main counter, if the timer is stopped. If the Timer Data Register is written while the timer is running, the new word is not loaded into the timer until it counts through H"01". However, if the timer is written while it is counting through H"01", an indeterminate value will be written into the timer constant register. This may be circumvented by ensuring that the data register is not written when the count is H"01".

If the main counter is loaded with "01", a Time Out Pulse will occur every time the prescaler presents a count pulse to the main counter. If loaded with "00", a Time Out pulse will occur every 256 count pulses.

Figure 13 : Timer Data Registers (A, B, C, and D).

Port F (TADR)	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Port 10 (TBDR)	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Port 11 (TCDR)	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Port 12 (TDDR)	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

V000360

Figure 14 : Timer C and D Register.

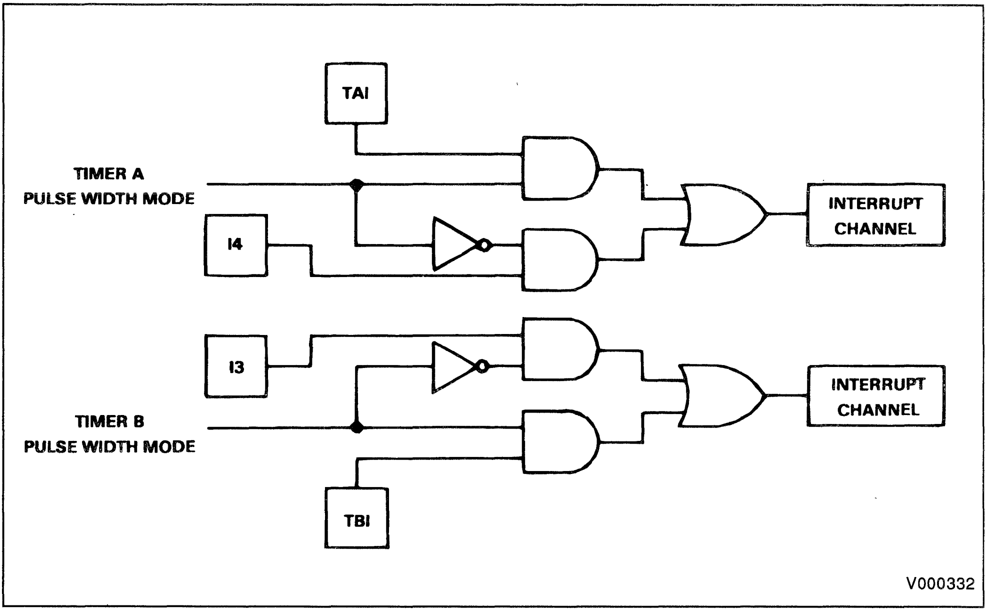
Port E (TCDCR)	*	CC_2	CC_1	CC_0	*	DC_2	DC_1	DC_0

C_2	C_1	C_0	
0	0	0	Timer Stopped
0	0	1	Delay Mode, ÷ 4 Prescale
0	1	0	Delay Mode, ÷ 10 Prescale
0	1	1	Delay Mode, ÷ 16 Prescale
1	0	0	Delay Mode, ÷ 50 Prescale
1	0	1	Delay Mode, ÷ 64 Prescale
1	1	0	Delay Mode, ÷ 100 Prescale
1	1	1	Delay Mode, : 200 Prescale

V000361

* Unused bits : read as zeros.

Figure 15 : A Conceptual Circuit of the MFP Timers in the Pulse Width Measurement Mode.



Changing the prescale value with the timer running can cause the first Time Out pulse to occur at an indeterminate time, (no less than one nor more than 200 timer clock cycles times the number in the time constant register), but subsequent Time Out pulses will then occur at the correct interval.

In addition to the delay mode described above, Timers A and B can also function in the Pulse Width Measurement mode or in the Event Count mode. In either of these two modes, an auxiliary control signal is required. The auxiliary control input for Timer A is TAI, and for Timer B, TBI is used. The interrupt channels associated with 14 and 13 are used for TAI and TBI, respectively, in Pulse Width mode. See Figure 15.

The pulse width measurement mode functions much like the delay mode. However, in this mode, the auxiliary control signal on TAI or TBI acts as an enable to the timer. When the control signal on TAI or TBI is inactive, the timer will be stopped. When it is active, the prescaler and main counter are allowed to run. Thus the width of the active pulse on TAI or TBI is determined by the number of timer counts which occur while the pulse allows the timer to run. The active state of the signal on TAI or TBI is dependent upon the associated Interrupt Channel's edge bit (GPIP 4 for TAI and GPIP 3 for TBI : see Active Edge Register in figure 5). If the edge bit associated with the TAI or TBI input is a one, it will be active high ; thus the timer will be allowed to run when the input is at a high level. If the edge bit is a zero, the TAI or TBI input will be active low. As previously stated, the interrupt channel (13 or 14) associated with the input still functions when the timer is used in the pulse width measurement mode. However, if the timer is programmed for the pulse width measurement mode, the interrupt caused by transitions on the associated TAI or TBI input will occur on the opposite transition.

For example, if the edge bit associated with the TAI input (AER-GPIP 4) is as one, an interrupt would normally be generated on the 0-1 transition of the 14 input signal. If the timer associated with this input (Timer A) is placed in the pulse width measurement mode, the interrupt will occur on the 1-0 transition of the TAI signal instead. Because the edge bit (AER-GPIP 4) is a one, Timer A will be allowed to count while the input is high. When the TAI input makes the high to low transition, Timer A will stop, and it is at this point that the interrupt will occur (assuming that the channel is enabled). This allows the interrupt to signal the CPU that the pulse being measured has terminated ; thus Timer A may now be read to determine the pulse width. (Again note that 13 and 14 may still be used for I/O when the timer is in the pulse width measurement mode). If Timer

A is reprogrammed for another mode, interrupts will again occur on the transition, as normally defined by the edge bit. Note that, like changing the edge bit, placing the timer into or taking it out of the pulse width mode can produce a transition on the signal to the interrupt channel and may cause an interrupt. If measuring consecutive pulses, it is obvious that one must read the contents of the timer and then reinitialize the main counter by writing to the timer data register. If the timer data register is written while the pulse is going to the active state, the write operation may result in an indeterminate value being written into the main counter. If the timer is written after the pulse goes active, the timer counts from the previous contents, and when it counts through H"01", the correct value is written into the timer. The pulse width then includes counts from before the timer was reloaded.

In the event count mode, the prescaler is disabled. Each time the control input on TAI or TBi makes an active transition as defined by the associated Interrupt Channel's edge bit, a count pulse will be generated, and the main counter will decrement. In all other respects, the timer functions as previously described. Altering the edge bit while the timer is in the event count mode can produce a count pulse. The interrupt channel associated with the input (I3 for I4 for TAI) is allowed to function normally. To count transitions reliably, the input must remain in each state (1/O) for a length of time equal to four periods of the timer clock ; thus signals of a frequency up to one fourth of the timer clock can be counted.

The manner in which the timer output pins toggle states has previously been described. All timer outputs will be forced low by a device RESET. The output associated with Timers A and B will toggle on each Time Out pulse regardless of the mode the timers are programmed to. In addition, the outputs from Timers A and B can be forced low at any time by writing a "1" to the reset location in TACR and TBCR, respectively. The output will be forced to the low state during the WRITE operation, and at the conclusion of the operation, the output will again be free to toggle each time a Time Out pulse occurs. This feature will allow waveform generation.

During reset, the Timer Data Registers and the main counters are not reset. Also, if using the reset option on Timers A or B, one must make sure to keep the other bits in the correct state so as not to affect the operation of Timers A and B.

USART

Serial Communication is provided by a full-duplex double-buffered USART, which is capable of either asynchronous or synchronous operation. Variable word length and start/stop bit configurations are available under software control for asynchronous operation. For synchronous operation, a Sync Word is provided to establish synchronization during receive operations. The Sync Word will also be repeatedly transmitted when no other data is available for transmission. Moreover, the MK68901 allows stripping of all Sync Words received in synchronous operation. The handshake control lines RR (Receiver Ready) and TR (Transmitter Ready) allow DMA operation. Separate receive and transmit clocks are available, and separate receive and transmit status and data bytes allow independent operation of the transmit and receive sections.

The USART is provided with three Control/Status Registers and a Data Register. The USART Data Register form is illustrated in figure 16. The programmer may specify operational parameters for the USART via the Control Register, as shown in figure 17. Status of both the Receiver and Transmitter sections is accessed by means of the two Status Registers, as shown in figures 18 and 19. Data written to the Data Register is passed to the transmitter, while reading the Data Register will access data received by the USART.

Figure 16 : USART Data Register.

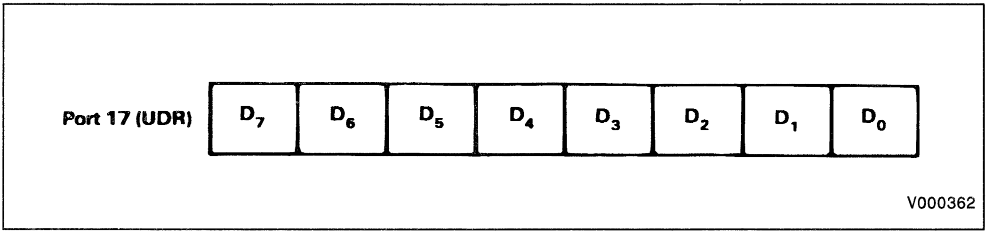


Figure 17 : USART Control Register (UCR).

	UCR$_7$							UCR$_0$
Port 14 (UCR)	1 = ÷ 16 0 = ÷ 1	WL$_1$	WL$_0$	ST$_1$	ST$_0$	PARITY ENABLED ON 1	1 = EVEN 0 = ODD	*

* Unused bits; read as zero

V000363

÷ 16/÷ 1 : When this bit is zero, data will be clocked into and out of the receiver and transmitter at the frequency of their respective clocks. When this bit is loaded with a one, data will be clocked into and out of the receiver and transmitter at one sixteenth the frequency of their respective clocks. Additionally, when placed in the divide by sixteen mode, the receiver data transition resynchronization logic will be enabled.

WL0-WL1 :Word Length Control. These two bits set the length of the data word (exclusive of start bits, stop bits, and parity bits as follows:

WL1	WL0	Word Length
0	0	8 Bits
0	1	7 Bits
1	0	6 Bits
1	1	5 Bits

ST0-ST1 :Start/stop bit control (format control). These two bits set the format as follows :

ST1	ST0	Start Bits	Stop Bits	Format
0	0	0	0	SYNC
0	1	1	1	ASYNC
†1	0	1	$1^1/_2$	ASYNC
1	1	1	2	ASYNC

PARITY : Parity Enabled. When set ("1"), parity will be checked by the receiver, parity will be calculated, and a parity bit will be inserted by the transmitter. When cleared ("0") no parity check will be made and no parity bit will be inserted for transmission.

For a word length of 8 the MFP calculates the parity and appends it when transmitting a sync character. For shorter lengths, the parity must be stored in the Sync Character Register (SCR) along with the sync character.

E/O : Even-Odd. When set ("1"), even parity will be used if parity is enabled. When cleared ("0"), odd parity will be used if parity is enabled.

Note that the synchronous or asynchronous format may be selected independently of a ÷ 1 or ÷ 16 clock. Thus it is possible to clock data synchronously into the device but still use start and stop bits. In this mode, all normal asynchronous format features still apply. Data will be shifted in after a start bit is encountered, and a stop bit will be checked to determine proper framing. If a transmit underrun condition occurs, the output will be placed in a marking state, etc. It is conversely possible to clock data in asynchronously using a synchronous format. There is data transition detection logic built into the receive clock circuitry which will re-synchronize the internal shift clock on each data transition so that, with sufficienty frequent data transitions, start bits are not required. In this mode, all other common synchronous features function normally. This re-synchronization logic is only active in ÷ 16 clock mode.

RECEIVER

The receiver section of the USART is configured by the UCR as previously described. The status of the receiver can be determined by reading and writing to the Receiver Status Register (RSR). The RSR is configured as follows :

Figure 18 : Receiver Status Register (RSR).

	RSR$_7$							RSR$_0$
Port 15 (RSR)	BUFFER FULL	OVERRUN ERROR	PARITY ERROR	FRAME ERROR	FOUND/$\overline{\text{SEARCH}}$ OR BREAK DETECT	MATCH/CHARACTER IN PROGRESS	SYNC STRIP ENABLE	RECEIVER ENABLE

V000364

BF : Buffer Full. This bit is set when the incoming word is transferred to the receive buffer. The bit is cleared when the receive buffer is read by reading the UDR. This bit of the RSR is read only.

OE : Overrun Error. This flag is set if the incoming word is completely received and due to be tranferred to the receive buffer, but the last word in the receive buffer has not yet been read. When this condition occurs, the word in the receive buffer is not overwritten by the new word. Note that the status flags always reflect the status of the data word currently in the receive buffer. As such, the OE flag is not actually set until the good word currently in the buffer has been read. The interrupt associated with this error will also not be generated unti the old word in the receive buffer has been read.

OE flag is cleared by reading the receiver status register, and new data words cannot be shifted to the receive buffer until this is done.

PE : Parity Error. This flag is set if the word received has a parity error. The flag is set when the received word is tranferred from the shift register to the receive buffer if the error condition exists. The flag is cleared when the next word which does not have a parity error is tranferred to the receive buffer.

FE : Frame Error. This flag only applies to the asynchronous format. A frame error is defined as a non-zero data word which is not followed by a stop bit. Like the PE flag, the FE flag is set or cleared when a word is transferred to the receive buffer.

F/S : Found/$\overline{\text{Search}}$. This combination control bit and flag bit is only used with the synchronous format. It can be set or cleared by writing to this bit of the RSR. When this bit is cleared, the receiver is placed in the search mode. In this mode, a bit by bit comparison of the incoming data to the character in the Sync Character Register (SCR) is made. The word length counter is disabled. When a match is found, this bit will be set automatically, and the word length counter will start as sync has not been achieved. An interrupt will be generated on the receive error channel when the match occurs. The word just shifted in will, or necessity, be equal to the sync character, and it will not be transferred to the receive buffer.

B : Break. This flag is used only when the asynchronous format is selected. This flag will be set when an all zero data word, followed by no stop bit, is received. The flag will stay set until both a non-zero bit is received and the RSR has been read at least once since the flag was set. Break indication will not occur if the receive buffer is full.

M/CIP : Match/Character in Progress. If the synchronous format is selected, this flag is the Match flag. It will be set each time the word transferred to the receive buffer matches the sync character. It will be reset each time the word transferred to the receive buffer does not match the sync character. If the asynchronous format is selected, this flag represents Character in Progress. It will be set upon a start bit detect and cleared at the end of the word.

SS : Sync Strip Enable. If this bit is set to a one, data words that match the sync character will not be loaded into the receive buffer, and no buffer full signal will be generated.

RE : Receiver Enable. This control bit is used to enable or disable the receiver. If a zero is written to this bit of the RSR, the receiver will turn off immediately. All flags including the F/S bit will be cleared. If a one is written to this bit, normal receiver operation is enabled. The receive clock has to be running before the receiver is enabled.

There are two interrupt channels associated with the receiver. One channel is used for the normal Buffer Full condition, while the other channel is used whenever an error condition occurs. Only one interrupt is generated per word received, but dedicating two channels allows separate vectors : one for the normal condition, and one for an error condition. If the error channel is disabled, an interrupt will be generated via the Butter Full Channel, whether the word received is normal or in error. Those conditions which produce an interrupt via the error channel are : Overrun, Parity Error, Frame Error, Sync Found, and Break. If a received word has an error associated with it, and the error interrupt channel is enabled, an interrupt will occur on the error channel only.

Each time a word is transferred into the receive buffer, a corresponding set of flags is latched into the RSR. No flags (except CIP) are allowed to change until the data word has been read from the receive buffer. Reading the receive buffer allows a new data word to be transferred to the receive buffer when it is received. Thus one should first read the RSR then read the receive buffer (UDR) to ensure that the flags just read match the data word just read. If done in the reverse order, it is possible that subsequent to reading the data word from the receive buffer, but prior to reading the RSR, a new word may be received and transferred to the receive buffer and, with it, its associated flags latched into the RSR. Thus, when the RSR is read, those flags may actually correspond to a different data word. It is good practice, also to read the RSR prior to a data read as, when an overrun error occurs, the receiver will not assemble new characters until the RSR has been read.

As previously stated, when overrun occurs, the OE flag will not be set and the associated interrupt will not be generated until the receive buffer has been read. If a break occurs, and the receive buffer has not yet been read, only the B flag will be set (OE will not be set). Again, this flag will not be set until the last valid word has been read from the receive buffer. If the break condition ends and another whole data word is received before the receive buffer is read, both the B and OE flags will be set once the receive buffer is read.

If a break occurs while the OE flag is set, the B flag will also be set.

A break generates an interrupt when the condition occurs and again when the condition ends. If the break condition ends before it is acknowledged by reading the RSR, the receiver error interrupt indicating end of break will be generated once the RSR is read.

Anytime the asynchronous format is selected, start bit detection is enabled. New data is not shifted into the shift register until a zero bit is detected. If a ÷ 16 clock is selected, along with the asynchronous format, false start bit detection is also enabled. Any transition has to be stable for 3 positive going edges of the receive clock to be called a valid transition. For a start bit to be good, a valid 0-1 transition must not occur for 8 positive clock transitions after the initial valid 1-0 transition.

After a good start bit has been detected, valid transitions in the data are checked for continously. When a valid transition is detected, the counter is forced to state zero, and no more transition checking is started until state four. At state eight, the "previous state" of the transition checking logic is clocked into the receiver.

As a result of this resynchronization logic, it is possible to run with asynchronous clocks without start and stop bits if there are sufficient valid transitions in the data stream. This logic also makes the unit more tolerant of clock skew for normal asynchronous communications than a device which employs only start bit synchronization.

Figure 19 : Transmitter Status Register (TSR).

	TSR_7							TSR_0
Port 16 (TSR)	BUFFER EMPTY	UNDERRUN ERROR	AUTO TURNAROUND	END OF TRANSMISSION	BREAK	HIGH	LOW	TRANSMITTER ENABLE

V000365

TRANSMITTER

The transmitter section of the USART is configured as to format, word length, etc. by the UCR, as previously described. The status of the transmitter can be determined by reading or writing the Transmitter Status Register (TSR). The TRS is configured as follows :

BE : Buffer Empty. This status bit is set when the word in the transmit buffer is transferred to the output shift register and thus the transmit buffer may be reloaded with the next data word. The flag is cleared when the transmit buffer is reloaded. The transmit buffer is loaded by writing to the UDR.

UE : This bit is set when the last word has been shifted out of the transmit shift register before a new word has been loaded into the transmit buffer. It is not necessary to clear this bit before loading the UDR.

This bit may be cleared by either reading the TSR or by disabling the transmitter. After the setting of the UE bit, one full transmitter clock cycle is required before this bit can be cleared by a read. The timing in some systems may allow a read of the TSR before the required clock cycle has been completed. This would result in the UE bit not being cleared until the following read. To avoid this problem, a dummy read of the TSR should be performed at the end of he UE service routine.

Only one underrun error may be generated between loads of the UDR regardless of the number of transmitter clock cycles between UDR loads.

AT : This bit causes the receiver to be enabled at the end of the transmission of the last word in the transmitter if the transmitter has been disabled.

END : End or Transmission. When the transmitter is turned off with a character still in the output shift register, transmission will continue until that character is shifted out. Once it has cleared the output register, the END bit will be set. If no character is being transmitted when the transmitter is disabled, the transmitter will stop at the next rising edge of the internal shift clock, and END will immediately be set. The END bit is cleared by re-enabling the transmitter.

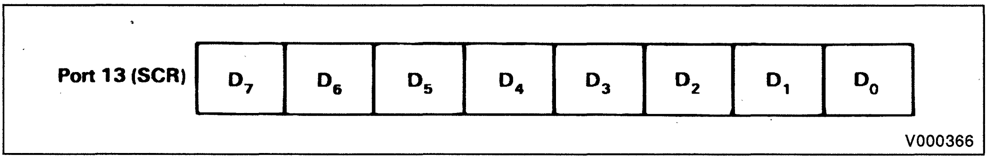
B : Break. This control bit will cause a break to be transmitted. When a "1" is written to the B bit of the TSR, a break will be transmitted upon completion of the character (if any) currently being transmitted. A break will continue to be transmitted until the B bit is cleared by writing a "0" tot his bit of the TSR. At that time, normal transmission will resume. The B bit has no function in the synchronous format. Setting the "B" bit to a one keeps the "BE" bit from being set to a one. So, if there were a word in the buffer at the start of break, it would remain there until the end of break, at which time it would be transmitted (if the transmitter is still enabled). If the buffer were not full at the start of break, it could be written at any time during the break. If the buffer is empty at the end of break, the underrun flag will be set (unless the transmitter is disabled).

The BREAK bit cannot be set until the transmitter has been enabled and the transmitter has had sufficient time (one clock cycle) to perform the internal reset and initialization functions.

H,L : High and Low. These two control bits are used to configure the transmitter output, when the transmitter is disabled, as follows :

H	L	Output State
0	0	Hi-Z
0	1	Low ("0")
1	0	High
1	1	Loop-Connects transmitter output to receiver input, and TC to Receiver Clock (RC and SI are not used ; they are bypassed internally). In loop back mode, transmitter output goes high when disabled.

Figure 20 : SYNC Character Register.



Altering these two bits after Transmitter Enable (XE) is set will alter the output state until END is false. These bits should be set prior to enabling the transmitter. The state of these bits determine the state of the first transmitted character after the transmitter is enabled. If the high impedance mode was selected prior to the transmitter being enabled, the first bit transmitted is indeterminate.

XE : Transmitter Enable. This control bit is used to enable or disable the transmitter. When set, the transmitter is enabled. When cleared, the transmitter will be disabled. If disabled, any word currently in the output register will continue to be transmitted when XE is cleared, the transmitter will turn off at the end of the break character boundary, and no end of break stop bit is transmitted. The transmit clock must be running before the transmitter is enabled A "one" bit always precedes the first word out of the transmitter after the transmitter is enabled. There is a delay between the time the transmitter enable bit is written an when the transmitter reset goes low ; therefore, the H & L bits should be written with the desired state prior to enabling the transmitter.

Like the receiver section, there are two separate interrupt channels associated with the transmitter. The buffer Empty condition causes an interrupt via one channel, while the Underrun and END conditions will cause an interrupt via the second channel. When underrun occurs in the synchronous format, the character in the SCR will be transmitted until a new words is loaded into the transmit buffer. In the asynchronous format, a "Mark" will be continuously transmitted when underrun occurs.

The transmit buffer can be loaded prior to enabling the transmitter. When the transmitter is disabled, any character currently in the process of being transmitted will continue to conclusion, but any character in the transmit buffer will not be transmitted and will remain in the buffer. Thus no buffer empty interrupt will occur nor will the BE flag be ste. If the buffer were already empty, the BE flag would be set and would remain set. When the transmitter is disabled with a character in the output register but with no character in the transmit buffer, an Underrun Error will not occur when the character in progress concludes.

Often it is necessary to send a break for some particular period. To aid in timing a break transmission, a transmission, a transmit error interrupt will be generated at every normal character boundary time during a break transmission. The status register information is unaffected by this error condition interrupt. It should be noted that an underrun error, if present, must be cleared from the TSR, and the interrupt pending register must be cleared of pending transmitter errors at the beginning of the break transmission or no interrupts will be generated at the character boundary time.

It the synchronous format is selected, the sync character should be loaded into the Sync Character Register (SCR) as shouwn in figure 20. This character is compared to the received serial data during a Search, and will be continuously transmitted during an underrun condition.

All flags in the RSR or TSR will continue to function as described whether their associated interrupt channel is disabled or enabled. All interrupt channels are edge triggered and, in many cases, it is the actual output of a flag bit or flag bits which is coupled to the interrupt channel. thus, if a normal interrupt producing condition occurs while the interrupt channel is disabled, no interrupt would be produced even if the channel was subsequently enabled, because a transition did not occur while the interrupt channel was enabled. that particular flag bit would have to occur a second time before another "edge" was produced, causing an interrupt to be generated.

Error conditions in the USART are determined by monitoring the Receive Status Register and the Transmitter Status Register. These error conditions are only valid for each word boundary and are not latched. When executing block tranfers or data, it is necessary to save any errors so that they can be checked at the end of a block. In order to save error conditions during data transfer, the MK68901 MFP interrupt controller may be used by enabling error interrupt for the desired channel (Receive error or Transmit error) and by masking these bits off. Once the tranfer is complete, the Interrupt Pending Register can be polled, to determine the precence of a pending error interrupt, and therefore an error.

Unused bits in the sync character register are zeroed out ; therefore, word length should be set up prior to writing the sync word in some cases. Sync word length is the word length plus one when parity is enabled. The user has to determine the parity of the sync word when the word length is not 8 bits. The MK68901 MFP does not add a parity bit to the sync word if the word length is less than 8 bits. The extra bit in the sync word is transmitted as the parity bit. With a word length of eight, and parity selected, the parity bit for the sync word is computed an added on by the MK68901 MFP.

RR RECEIVER READY

RR is asserted when the Buffer Full bit is set in the RSR unless a parity error or frame error is detected by the receiver.

TR TRANSMITTER READY

TR is asserted when the Buffer Empty bit is set in the TSR unless a break is currently being transmitted.

REGISTER ACCESSES

All register accesses are dependent on CLK as shown in the timing diagrams. To read a register, CS and DS must be asserted, and R/W must by high. The internal read control signal is essentially the combination of CS, DS, and RD/WR. Thus, the read operation will begin when CS and DS go active and will end when either CS or DS goes inactive. The address bus must be stable prior to the start of the operation and must remain stable until the end of the operation. Unless a read operation or interrupt acknowledge cycle is in progress the data bus (D_0-D_7) will remain in the tri-state condition.

To write a register, CS and DS must be asserted and R/W must be low. The address must be stable prior to the start of the operation and must remain stable until the end of the operation. After the MK68901 asserts DTACK, the CPU negates DS,. At this time, the MFP latches the data bus and writes the contents into the appropriate register. Also when DS is negated, the MFP rescinds DTACK.

For an interrupt acknowledge, the operation starts when IACK goes low, and ends when IACK goes high. The data bus is tri-stated when either IACK or DS goes high.

When CS or IACK are asserted the MFP starts an internal cycle. DS is needed to enable the address and data buffers. It is recommended taht CS and IACK be gated by DS so that DS is always present whenever an MFP bus cycle starts.

MK68901 ELECTRICAL SPECIFICATIONS – PRELIMINARY

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
T_A	Temperature under Bias	– 25 to + 100	°C
T_{stg}	Storage Temperature	– 65 to + 150	°C
V_I	Voltage on Any Pin with Respect to Ground	– 0.3 to + 7	V
P_D	Power Dissipation	1.5	W

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect reliability.

D.C. CHARACTERISTICS

T_A = 0°C to 70°C ; V_{CC} = + 5V ± 5% Unless Otherwise Specified

Symbol	Parameter	Test Condition	Min.	Max.	Unit
V_{IH}	Input High Voltage		2.0	V_{CC} + .3	V
V_{IL}	Input Low Voltage		– 0.3	0.8	V
V_{OH}	Output High Voltage (except $\overline{\text{DTACK}}$)	I_{OH} = – 120μA	2.4		V
V_{OL}	Output Low Voltage (except $\overline{\text{DTACK}}$)	I_{OL} = 2.0mA		0.5	V
I_{LL}	Power Supply Current	Outputs Open		180	mA
I_{LI}	Input Leakage Current	V_{IN} = 0 to V_{CC}		± 10	μA
I_{LOH}	Tri-state Output Leakage Current in Float	V_{OUT} = 2.4 to V_{CC}		10	μA
I_{LOL}	Tri-state Output Leakage Current in Float	V_{OUT} = 0.5V		– 10	μA
I_{OH}	$\overline{\text{DTACK}}$ Output Source Current	V_{OUT} = 2.4		– 400	μA
I_{OL}	$\overline{\text{DTACK}}$ Output Sink Current	V_{OUT} = 0.5		5.3	mA

All voltages are referenced to ground.

CAPACITANCE

T_A = 25°C, f = 1MHz unmeasured pins returned to ground.

Symbol	Parameter	Test Condition	Max.	Unit
C_{IN}	Input Capacitance	Unmeasured pins returned to ground	10	pF
C_{OUT}	Tri-state Output Capacitance		10	pF

AC ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0Vdc ± 5%, GND = 0Vdc, T_A = 0°C to 70°C)

Number	Characteristic	Value				Unit	Fig.	Note
		MK68901-4		MK68901-5				
		Min.	Max.	Min.	Max.			
1	$\overline{CS}$, $\overline{DS}$ Width High	50		35		ns	21,22	5
2	R/$\overline{W}$, A1-A5 Valid to Falling $\overline{CS}$ (setup)	0		0		ns	21,22	
3	Data Valid Prior to Falling CLK	280		0		ns	22	
4	$\overline{CS}$, $\overline{IACK}$ Valid to Falling Clock (setup)	50		45		ns	21-24	3
5	CLK Low to $\overline{DTACK}$ Low		220		180	ns	21,22	
6	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ High		60		55	ns	21-24	
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state		100		95	ns	21-24	
8	$\overline{DTACK}$ Low to Data Invalid (hold time)	0		0		ns	22	
9	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to Data Tri-state		50		50	ns	21,23,24	
10	$\overline{CS}$ or $\overline{DS}$ High to R/$\overline{W}$, A1-A5 Invalid (hold time)	0		0		ns	21,22	
11	Data Valid from $\overline{CS}$ Low		310		260	ns	21	3,6
12	Read Data Valid to $\overline{DTACK}$ Low (setup)	50		50		ns	21	
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (hold time)	0		0		ns	21-23	
14	$\overline{IEI}$ Low to Falling $\overline{CLK}$ (setup)	50		50		ns	23,24	
15	$\overline{IEO}$ Valid from Clock Low (delay)		180		180	ns	23	1
16	Data Valid from Clock Low (delay)		300		300	ns	23	
17	$\overline{IEO}$ Invalid from $\overline{IACK}$ High (delay)		150		150	ns	23, 24	
18	$\overline{DTACK}$ Low from Clock High (delay)		180		165	ns	23, 24	
19	$\overline{IEO}$ Valid from $\overline{IEI}$ Low (delay)		100		100	ns	24	1
20	Data Valid from $\overline{IEI}$ Low (delay)		220		220	ns	24	
21	Clock Cycle Time	250	1000	200	1000	ns	21	
22	Clock Width Low	110		90		ns	21	
23	Clock Width High	110		90		ns	21	
24	$\overline{CS}$, $\overline{IACK}$ Inactive to Rising Clock (setup)	100		80		ns	21-23	4,5
25	I/O Minimum Active Pulse Width	100		100		ns	25	
26	$\overline{IACK}$ Width High	2		2		T_{CLK}	23-24	2
27	I/O Data Valid from Rising $\overline{CS}$ or $\overline{DS}$		450		450	ns	26	
28	Receiver Ready Delay from Rising RC		600		600	ns	27	
29	Transmitter Ready Delay from Rising TC		600		600	ns	28	
30	Timer Output Low from Rising Edge of $\overline{CS}$ or $\overline{DS}$ (A & B) (reset T_{OUT})		450		450	ns	29	7
31	T_{OUT} Valid from Internal Timeout		2 t_{CLK} + 300		2 t_{CLK} + 300	ns	29	2
32	Timer Clock Low Time	110		90		ns	29	

AC ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = 5.0Vdc ± 5%, GND = 0Vdc, T_A = 0°C to 70°C)

Number	Characteristic	Value				Unit	Fig.	Note
		MK68901-4		MK68901-5				
		Min.	Max.	Min.	Max.			
33	Timer Clock High Time	110		90		ns	29	
34	Timer Clock Cycle Time	250	1000	200	1000	ns	29	
35	$\overline{\text{RESET}}$ Low Time	2		1.8		µs	30	
36	Delay to Falling $\overline{\text{INTR}}$ from External Interrupt Active Transition		380		380	ns	25	
37	Transmitter Internal Interrupt Delay from Falling Edge of TC		550		550	ns	28	
38	Receiver Buffer Full Interrupt Transition Delay from Rising Edge of RC		800		800	ns	27	
39	Receiver Error Interrupt Transition Delay from Falling Edge of RC		800		800	ns	27	
40	Serial in Set Up Time to Rising Edge of RC (divide by one only)	80		70		ns	27	
41	Data Hold Time from Rising Edge of RC (divide by one only)	350		325		ns	27	
42	Serial Output Data Valid from Falling Edge of TC (÷1)		440		420	ns	28	
43	Transmitter Clock Low Time	500		450		ns	28	
44	Transmitter Clock High Time	500		450		ns	28	
45	Transmitter Clock Cycle Time	1.05	∞	0.95	∞	µs	28	
46	Receiver Clock Low Time	500		450		ns	27	
47	Receiver Clock High Time	500		450		ns	27	
48	Receiver Clock Cycle Time	1.05	∞	0.95	∞	µs	27	
49	$\overline{\text{CS}}$, $\overline{\text{IACK}}$, $\overline{\text{DS}}$ Width Low		80		80	T_{CLK}	29	2
50	Serial Output Data Valid from Falling Edge of TC (÷16)		490		370	ns	28	

Notes :

1. $\overline{\text{IEO}}$ only goes low if no acknowledgeable interrupt is pending. If $\overline{\text{IEO}}$ goes low, $\overline{\text{DTACK}}$ and the data bus remain tri-stated.
2. T_{CLK} refers to the clock applied to the MFP CLK input pin. t_{CLK} refers to the timer clock signal, regardless of whether that signal comes from the XTAL 1/XTAL2 crystal clock inputs or the TAI or TBI timer inputs.
3. If the setup time is not met, $\overline{\text{CS}}$ or IACK will not be recognized until the next falling CLK.
4. If this setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.
5. $\overline{\text{CS}}$ is latched internally, therefore if spec's 1 and 24 are met then $\overline{\text{CS}}$ may be reasserted before the rising clock and still terminate the current bus cycle. The new bus cycle will be delayed by the MK68901 until all appropriate internal operations have completed.
6. Although $\overline{\text{CS}}$ and $\overline{\text{DTACK}}$ are synchronized with the clock, the data out during a read cycle is asynchronous to the clock, relying only on $\overline{\text{CS}}$ for timing.
7. Spec. 30 applies to timer outputs TAO and TBO only.

TIMER A.C. CHARACTERISTICS

Definitions :

Error = Indicated Time Value - Actual Time Value

tpsc = t_{CLK}x Prescale Value

INTERNAL TIMER MODE

Single Interval Error (free running) (note 2)	± 100ns
Cumulative Internal Error	0
Error between Two Timer Reads	± (tpsc + $4t_{CLK}$)
Start Timer to Stop Timer Error	+ ($2t_{CLK}$ + 100ns) to - (tpsc + $6t_{CLK}$ + 100ns)
Start Timer to Read Timer Error	+ 0 to – (tpsc + $6t_{CLK}$ + 400ns)
Start Timer to Interrupt Request Error (note 3)	- $2t_{CLK}$ to – ($4t_{CLK}$ + 800ns)

PULSE WIDTH MEASUREMENT MODE

Measurement Accuracy (note 1)	+ $2t_{CLK}$ to – (tpsc + $4t_{CLK}$)
Minimum Pulse Width	$4t_{CLK}$

EVENT COUNTER MODE

Minimum Active Time of TAI, TBI	$4t_{CLK}$
Minimum Inactive Time of TAI, TBI	$4t_{CLK}$

Notes : 1. Error may be cumulative if repetitively performed.
2. Error with respect to T_{OUT} or INT if note 3 is true.
3. Assuming it is possible for the timer to make an interrupt request immediately.

Figure 21 : Read Cycle.

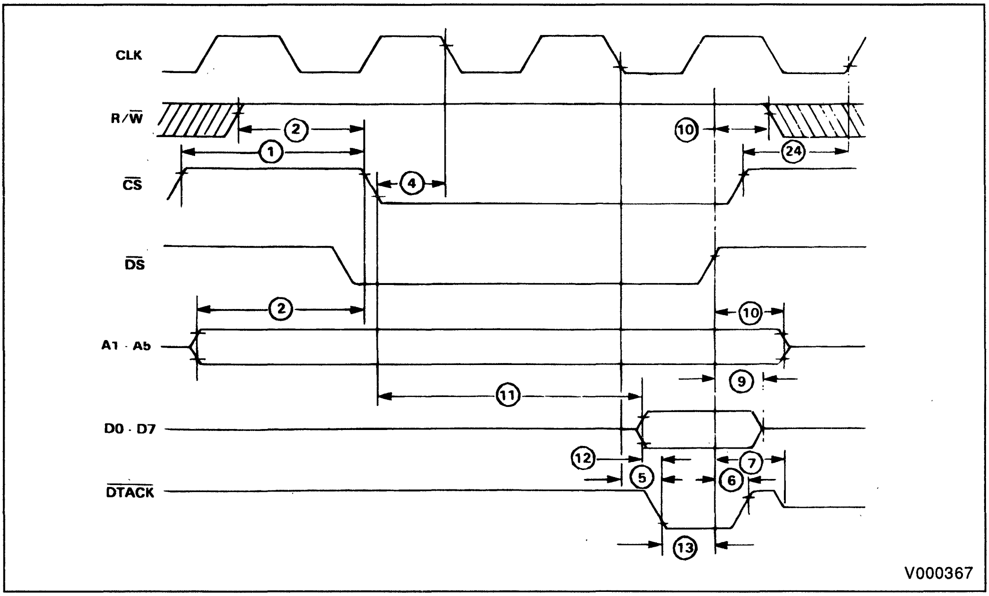
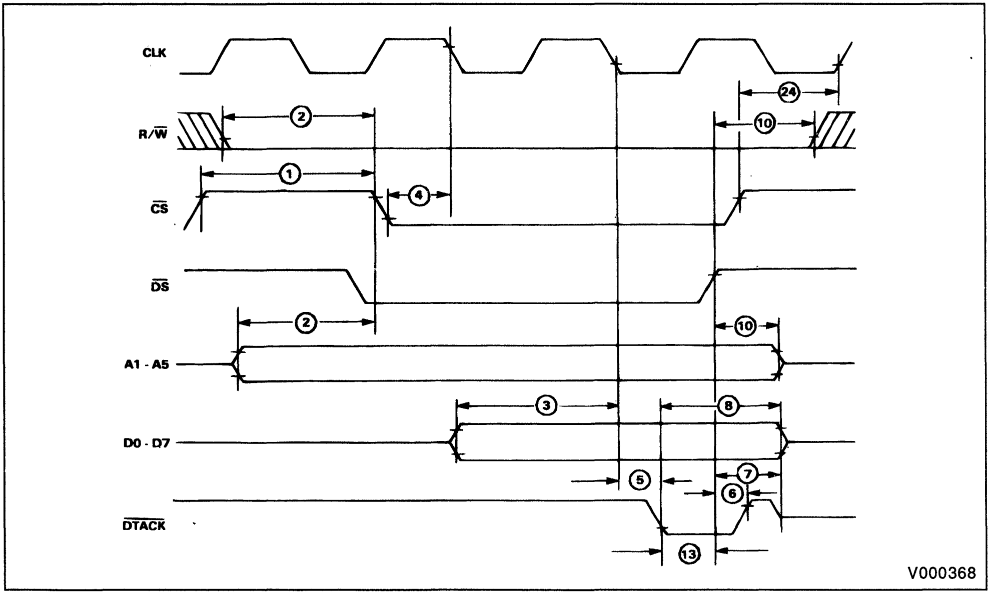


Figure 22 : Write Cycle.



Note : $\overline{\text{CS}}$ and $\overline{\text{IACK}}$ must be a function of $\overline{\text{DS}}$.

Figure 23 : Interrupt Acknowledge ($\overline{IEI}$ low).

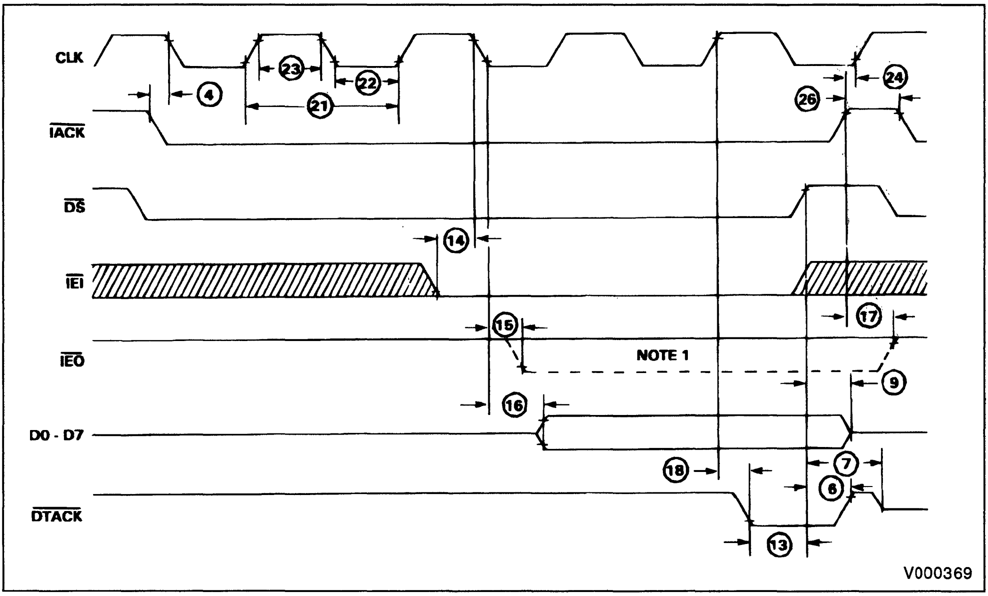
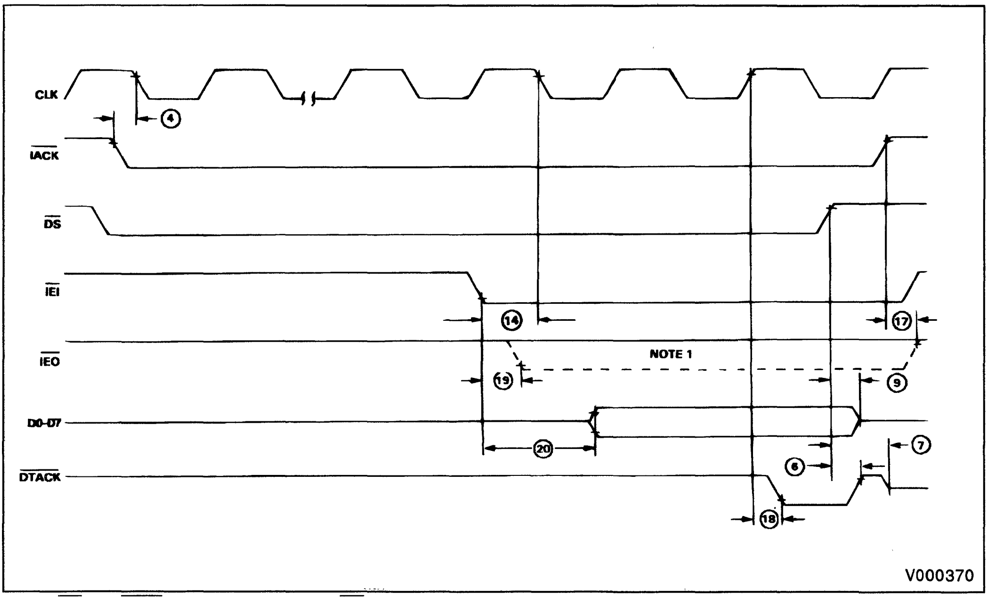
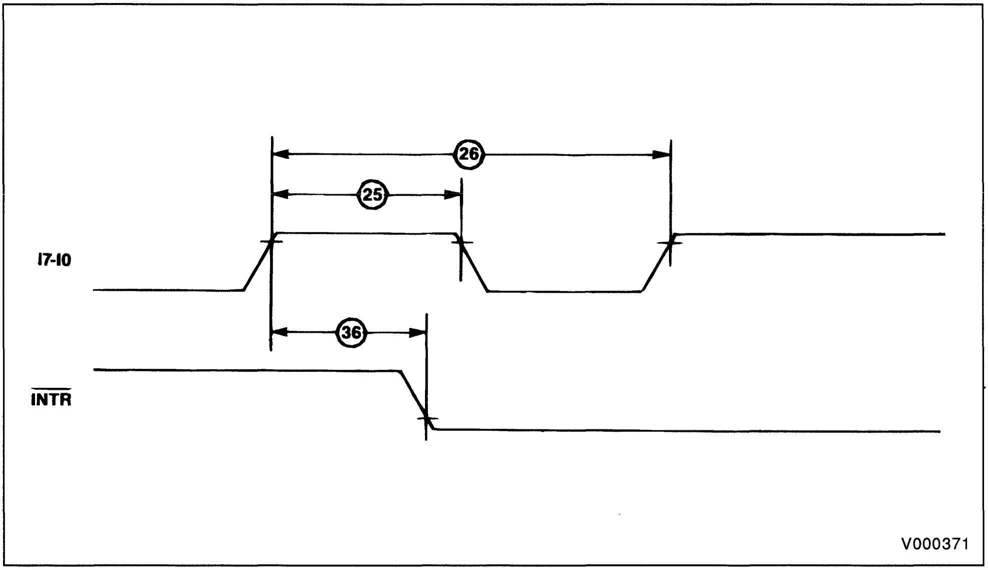


Figure 24 : Interrupt Acknowledge Cycle ($\overline{IEI}$ high).



Note : $\overline{CS}$ and $\overline{IACK}$ must be a function of $\overline{DS}$.

Figure 25 : Interrupt Timing.



Note : Active edge is assumed to be the rising edge.

Figure 26 : Port Timing.

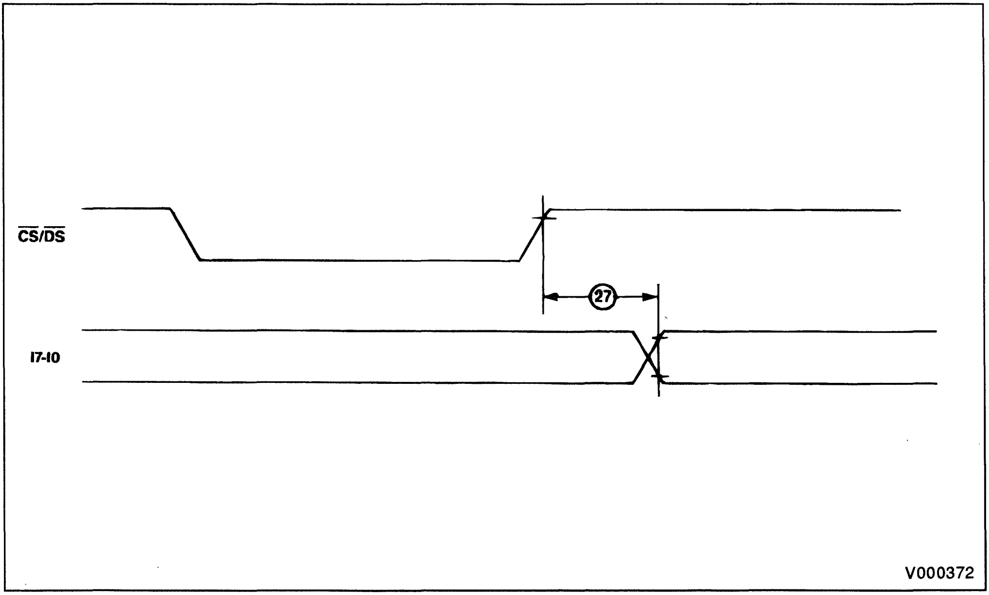


Figure 27 : Receiver Timing.

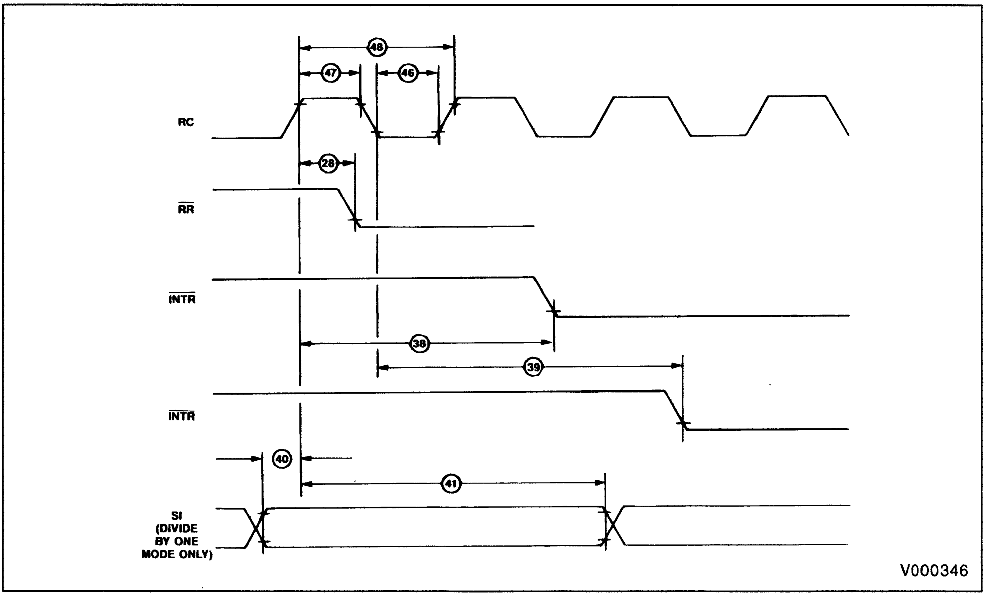


Figure 28 : Transmitter Timing.

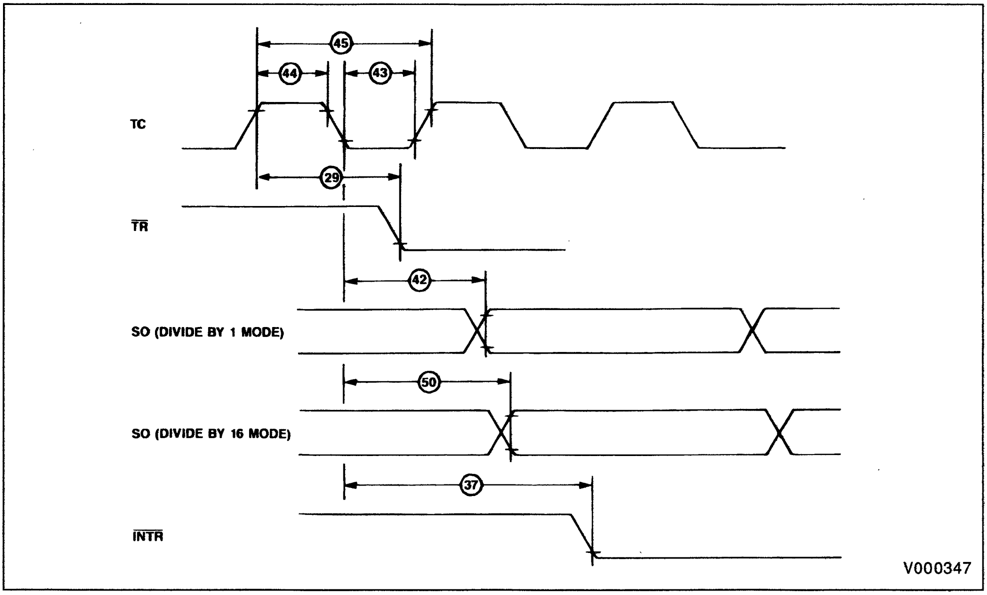


Figure 29 : Timer Timing.

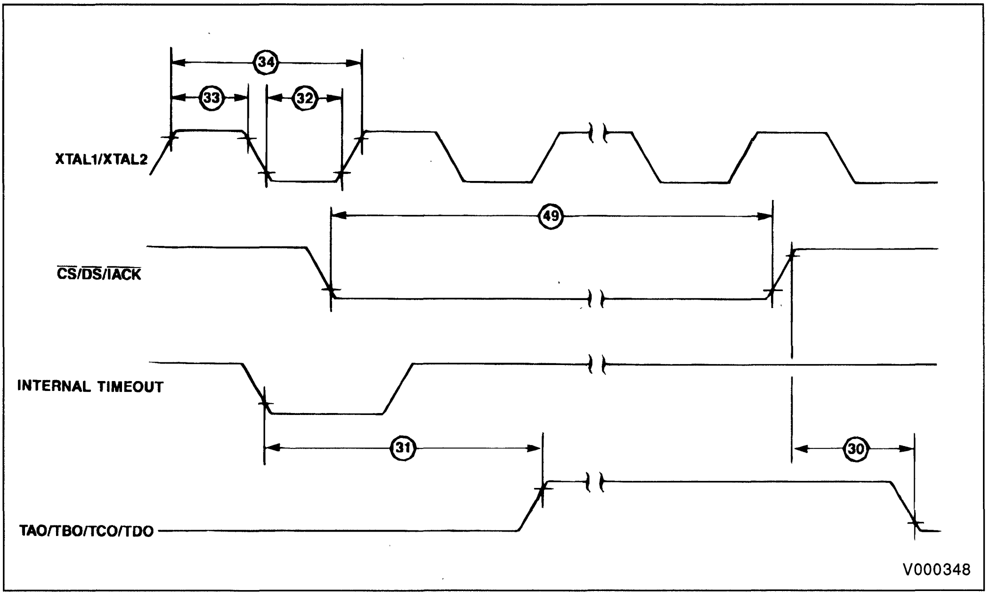


Figure 30 : Reset Timing.

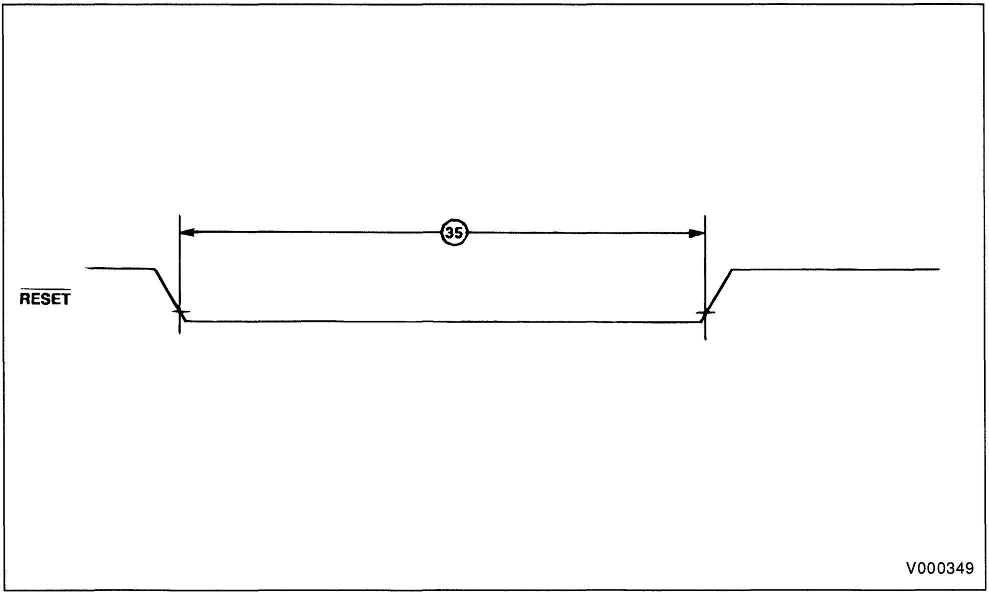
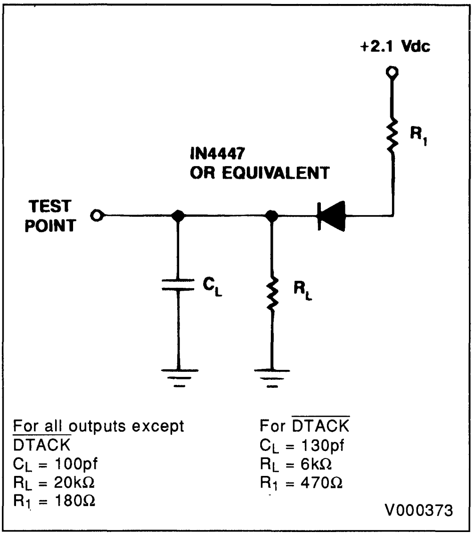


Figure 31 : Typical Output.



For all outputs except $\overline{\text{DTACK}}$
C_L = 100pf
R_L = 20kΩ
R_1 = 180Ω

For $\overline{\text{DTACK}}$
C_L = 130pf
R_L = 6kΩ
R_1 = 470Ω

Figure 32 : INTR Test Load.

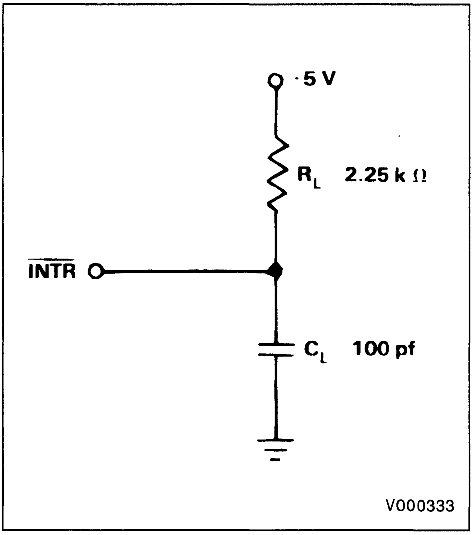
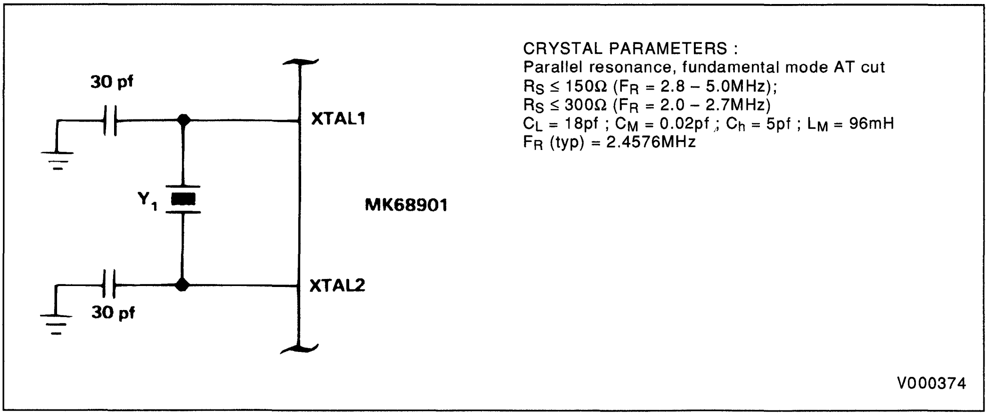


Figure 33 : MK68901 MFP External Oscillator Components.

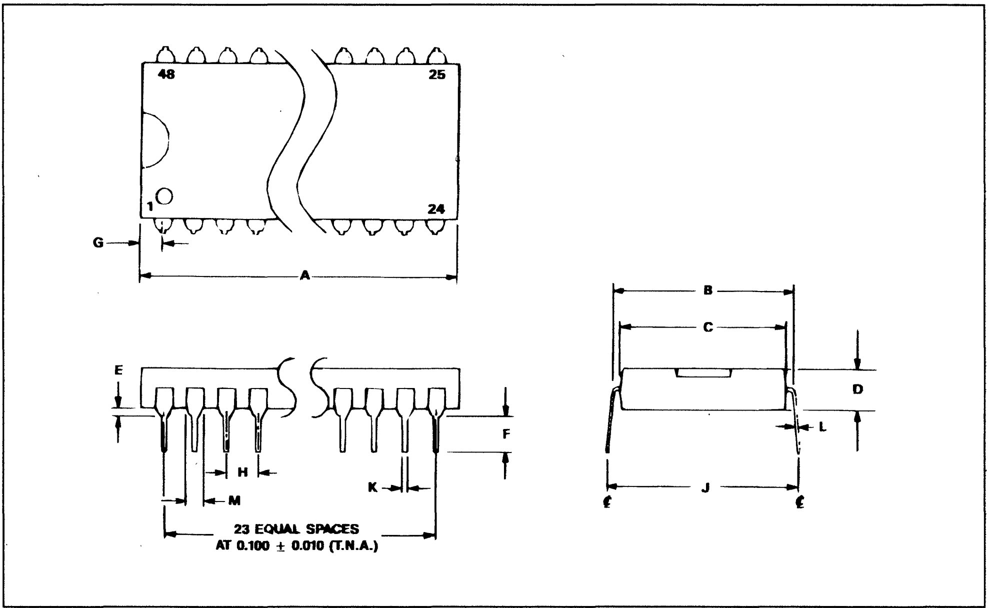


CRYSTAL PARAMETERS :
Parallel resonance, fundamental mode AT cut
$R_S \leq 150\Omega$ (F_R = 2.8 – 5.0MHz);
$R_S \leq 300\Omega$ (F_R = 2.0 – 2.7MHz)
C_L = 18pf ; C_M = 0.02pf ; C_h = 5pf ; L_M = 96mH
F_R (typ) = 2.4576MHz

MK68901 ORDERING INFORMATION

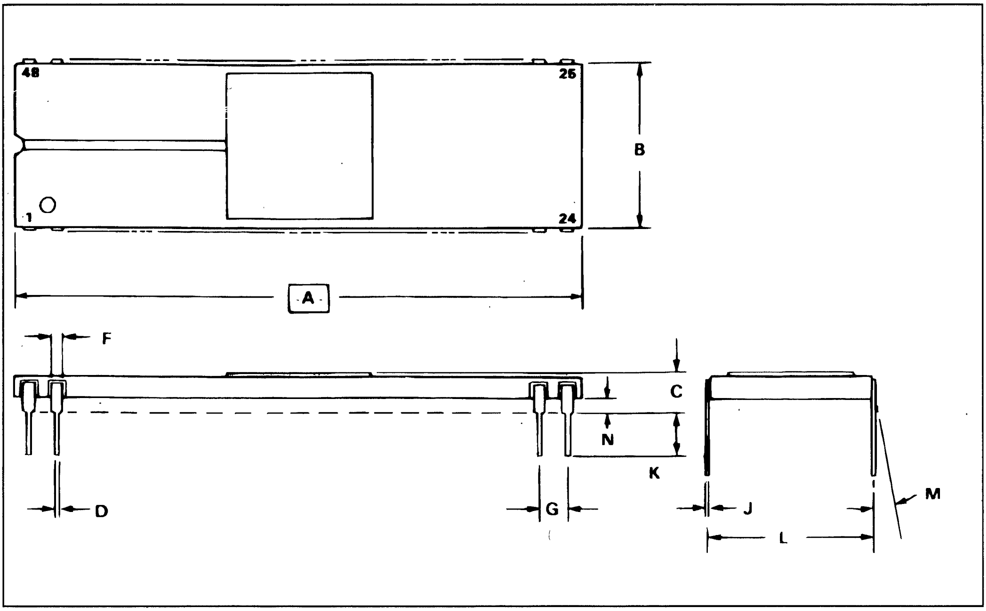
Part Number	Package Type	Max. Clock Frequency	Temperature Range
68901P04	Ceramic DIP	4.0MHz	0° to 70°C
68901P05	Ceramic DIP	5.0MHz	0° to 70°C
68901N04	Plastic DIP	4.0MHz	0° to 70°C
68901N05	Plastic DIP	5.0MHz	0° to 70°C
68901Q04	Plastic PLCC	4.0MHz	0° to 70°C
68901Q05	Plastic PLCC	5.0MHz	0° to 70°C

MK68901 48–PIN PLASTIC DUAL–IN–LINE PACKAGE (N)



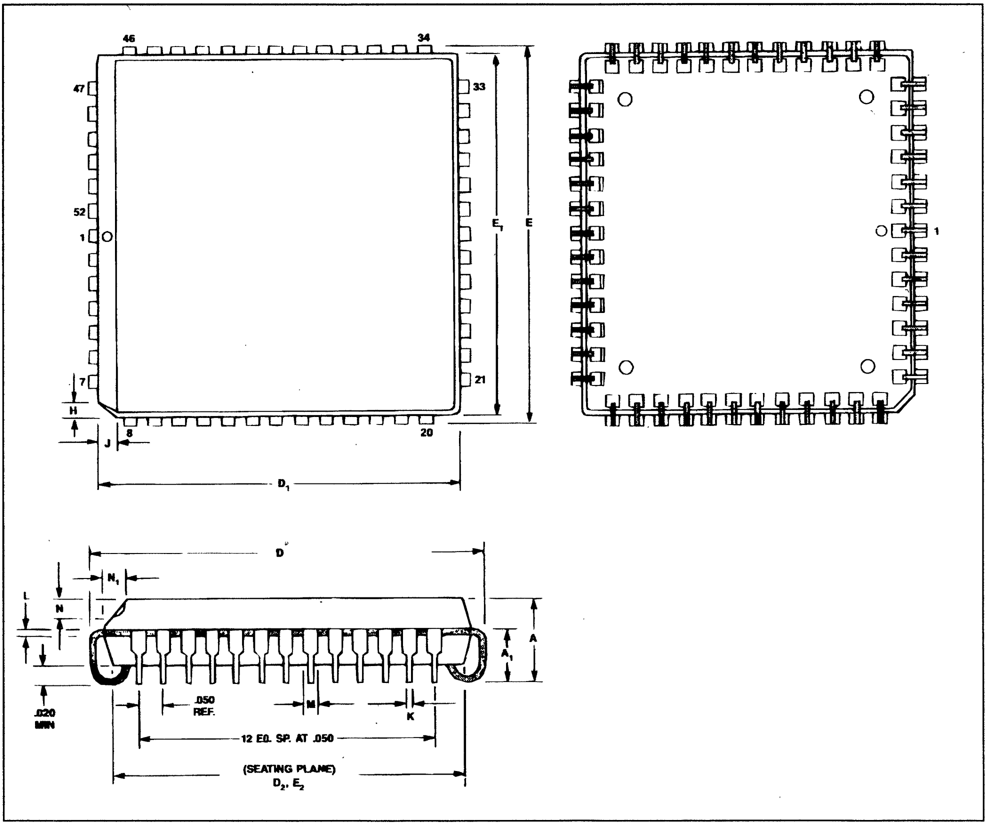
Dim	Millimeters		Inches	
	Min.	Max.	Min.	Max.
A	61.468	62.738	2.420	2.470
B	14.986	16.256	.590	.640
C	13.462	13.97	.530	.550
D	3.556	4064	.140	.160
E	0.381	1.524	.015	.060
F	3048	3.81	.120	.150
G	1.524	2.286	.060	.090
H	1.186	1.794	.090	.110
J	15.24	17.78	.600	.700
K	0.381	0.533	.015	.021
L	0.203	0.305	.008	.012
M	1.143	1.778	.045	.070

MK68901 48–PIN CERAMIC DUAL–IN–LINE PACKAGE (P)



Dim	Inches	
	Min.	Max.
A	2.376	2.424
B	0.576	0.604
C	0.120	0.160
D	0.015	0.021
F	0.030	0.055
G	0.100 BSC	
J	0.008	0.013
K	0.100	0.165
L	0.590	0.616
M	0°	10°
N	0.040	0.060

MK68901 52–PIN PLASTIC LEADED CHIP CARRIER (Q)



Dim	Inches	
	Min.	Max.
A	.165	.180
A_1	.090	.130
D	.785	.795
D_1	.750	.756
D_2	.690	.730
E	.785	.795
E_1	.750	.756
E_2	.690	.730
H	.042	.048
J	.042	.048
K	.013	.024
L	.008	.014
M	.026	.032
N/N_1	.043	.048

MK68901 PIN CONNECTIONS

PLCC	DIP	FUNC.	PLCC	DIP	FUNC.	PLCC	DIP	FUNC.
1	–	NC	19	17	XTAL1	37	33	$\overline{IEO}$
2	1	R/$\overline{W}$	20	18	XTAL2	38	34	IEI
3	2	A1	21	–	NC	39	35	CLK
4	3	A2	22	19	TAI	40	36	GND
5	4	A3	23	20	TBI	41	37	D0
6	5	A4	24	21	$\overline{RESET}$	42	38	D1
7	6	A5	25	22	IO	43	39	D2
8	7	TC	26	23	I1	44	40	D3
9	8	SO	27	24	I2	45	41	D4
10	9	SI	28	25	I3	46	42	D5
11	10	RC	29	26	I4	47	43	D6
12	11	V_{CC}	30	27	I5	48	44	D7
13	–	NC	31	28	I6	49	45	$\overline{IACK}$
14	12	NC	32	29	I7	50	46	$\overline{DTACK}$
15	13	TAO	33	–	NC	51	47	$\overline{DS}$
16	14	TBO	34	30	$\overline{TR}$	52	48	$\overline{CS}$
17	15	TCO	35	31	$\overline{RR}$			
18	16	TDO	36	32	$\overline{INTR}$			

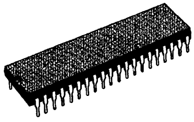
Note : NC – No Connection

SGS-THOMSON
MICROELECTRONICS

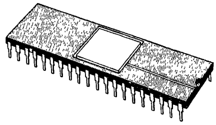
MK85C30

PRELIMINARY CMOS SERIAL COMMUNICATIONS CONTROLLER

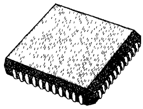
- TWO INDEPENDENT FULL-DUPLEX CHANNELS
- SYNCHRONOUS/ASYNCHRONOUS DATA RATES :
  - Up to 1/4 of the PCLK (i.e., 1Mbit/sec. maximum data rate with 4MHz PCLK. Using external phase-lock loop.
  - Up to 375Kbit/sec. with a 6MHz clock rate. Up to 250Kbit/sec. with a 4MHz clock rate (FM encoding using digital phase-locked loop).
  - Up to 187.5Kbit/sec. with a 6MHz clock rate. Up to 125Kbit/sec. with a 4MHz clock rate (NRZI encoding using digital phase-locked loop).
- ASYNCHRONOUS CAPABILITIES :
  - 5, 6, 7, or 8 bits per character
  - 1, 1-1/2, or 2 stop bits
  - Odd or even parity
  - Times 1, 16, 32, or 64 clock modes
  - Break generation and detection
  - Parity, overrun and framing error detection
- BYTE-ORIENTED SYNCHRONOUS CAPABILITIES :
  - Internal or external character synchronization
  - 1 or 2 sync characters (6 or 8 bits/character) in separate registers
  - Automatic Cyclic redundancy check (CRC) generation/detection
- SDLC/HDLC CAPABILITIES :
  - Abort sequence generation and checking
  - Automatic zero insertion and deletion
  - Automatic flag insertion between messages
  - Address field recognition
  - I-field residue handling
  - CRC generation/detection
  - SDLC loop mode with EOP recognition/loop entry and exit
- RECEIVER DATA REGISTERS QUADRUPLY BUFFERED TRANSMITTER DATA REGISTERED DOUBLE BUFFERED
- NRZ, NRZI, OR FM ENCODING/DECODING
- BAUD-RATE GENERATOR IN EACH CHANNEL
- DIGITAL PHASE-LOCKED LOOP FOR CLOCK RECOVERY
- CRYSTAL OSCILLATOR



PDIP-40



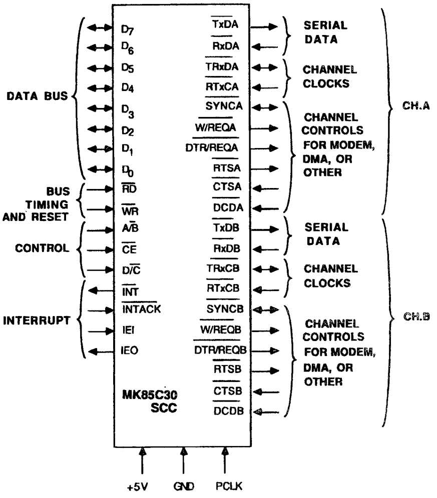
CDIP-40



PLCC44

(Order Codes at the end of the datasheet)

Figure 1 : Logic Functions.



INTRODUCTION

Thank you for your interest in the SCC, one of the most versatile and most popular Serial Data Communications ICs. This document is intended to provide answers to all technical questions about the MK85C30, the CMOS replacement for the NMOS Z8530 Serial Communications Controller. Please read this Preface where we try to anticipate your questions.

- If you are new to serial data communications, you will need additional tutorial information. Of the many introductory texts on this subject, *Technical Aspects of Data Communications* by John E. McNamara, published by Digital Press (DEC) 1982, is one of the best.
- If you have designed with simpler UARTs and USARTs, and HDLC/SDLC devices, the SCC offers you far greater flexibility, but also requires an in-depth study and understanding of the impact and the use of its many powerful features. This manual contains important information.
- If you are familiar with the Z80-SIO, you will feel right at home with the SCC, for it is really a functionally enhanced superset of the Z80-SIO.

Most users read only chapters that are of interest to them. If you are designing the microcomputer hardware structure using the SCC as a peripheral, you will want to read the Initialization Worksheet and Interrupt Routine Sections.

If you are programming a system using the SCC, you will be more interested, on the Initialization Worksheet Section.

Points To Watch Out For :

1. Follow the worksheet for initialization (page). Unexplainable operations may occur if this procedure is not followed.
2. Watch out for Write Recovery time violation (Interfacing Section). Both the CPU clock rate and the SCC clock rate will affect the Write Recovery time.
3. Ensure Mode bits are not changed when writing Commands. (Register Overview page 76). Each Mode bit affects only one function and a Command bit entry requires a rewrite of the entire register ; therefore, care must be taken to insure the integrity of the Mode bits whenever a new command is issued.
4. Data must be valid prior to falling edge of $\overline{\text{WR}}$ or $\overline{\text{DS}}$.
5. If not used, $\overline{\text{INTACK}}$ should be tied high.

GENERAL DESCRIPTION

The CMOS MK85C30 Serial Communications Controller is intended to be a replacement for the NMOS Z8530. The SCC is a dual-channel, multiprotocol data communications peripheral designed for use with 8-bit and 16-bit microprocessors. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including on-chip baud rate generators, digital phase-lock loops, and crystal oscillators, which dramatically reduce the need for external logic.

The SCC handles asynchronous formats, Synchronous byte-oriented protocols such as IBM Bisync, and Synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (telecommunications, cassette, diskette, tape drivers, etc.).

The device can generate and check CRC codes in any Synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

With access to 14 Write registers and 7 Read registers per channel, the user can configure the SCC so that it can handle all asynchronous formats regardless of data size, number of stop bits, or parity requirements. The SCC accommodates all synchronous formats including character, byte, and bit-oriented protocols.

Within each operating mode, the SCC also allows for protocol variations by checking odd or even parity bits, character insertion or deletion, CRC generation and checking/break and abort generation and detection, and many other protocol-dependent features.

The SCC MK85C30 is designed for non-multiplexed buses and is easily interfaced to CPUs such as the 8080, Z80, 6800, 68000 and *Multibus.

GENERAL DESCRIPTION (cont'd)

Figure 2 and Figure 5 show block diagrams of the SCC. Received data enters the receive data pins and follows one of several data paths, depending on the state of the control logic. The contents of the registers and the state of the external control pins establish the internal control logic. Transmitted data follows a similar pattern of control, register, and external pin definition.

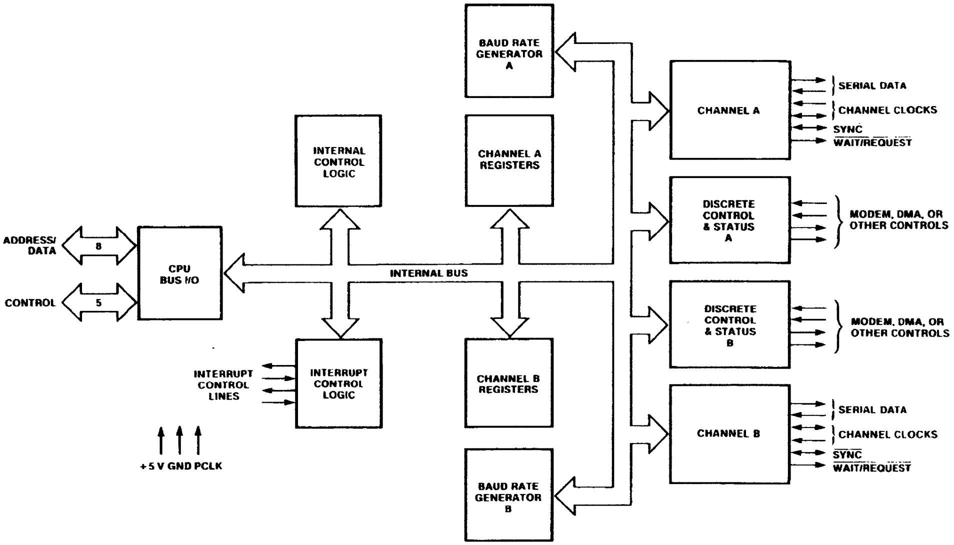
PIN DESCRIPTIONS

The SCC pins are divided into seven functional groups : Address/Data, Bus Timing and Reset, Device Control, Interrupt, Serial Data (both channels), Peripheral Control (both channels), and Clocks (both channels). Figures 3 and 4 show the Pin Configuration in both the proposed packages, Dual in Line and Chip Carrier.

The Address/Data group consists of the bidirectional lines used to transfer data between the CPU and the SCC. The direction of these lines depends on whether the SCC is selected and whether the operation is a Read or a Write.

The Timing and Control groups designate the type of transaction to occur and when this transaction will occur. The Interrupt group provides inputs and outputs to conform to the bus specifications for handling and prioritizing interrupts. The remaining groups are divided into Channel A and Channel B groups for serial data (transmit or receive), peripheral control (such as DMA or modem), and the input and output lines for the receive and transmit clocks.

Figure 2 : SCC Block Diagram.



PIN DESCRIPTION (cont'd)

Here below are described the pin functions of the MK85C30 Serial Communications Controller.

A/$\overline{\mathbf{B}}$. Channel A/Channel B Select (input, Channel A active HIGH). This signal selects the channel in which the Read or Write operation occurs.

$\overline{\mathbf{CE}}$. Chip Enable (input, active LOW). This signal selects the SCC for operation. It must remain active throughout the bus transaction.

D0-D7. Data Lines (bidirectional, 3-state). These I/O lines carry data or control information to and from the SCC.

D/$\overline{\mathbf{C}}$. Data/Control (input, Data active HIGH). This signal defines the type of information transfer performed by the SCC : data or control.

$\overline{\mathbf{RD}}$. Read (input, Active LOW). This signal indicates a Read operation and when the SCC is selected, enables the SCC bus drivers. During the interrupt acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

$\overline{\mathbf{WR}}$. Write (input, active LOW). When the SCC is selected, this signal indicates a Write operation. The coincidence of $\overline{\mathrm{RD}}$ and $\overline{\mathrm{WR}}$ is interpreted as a Reset.

$\overline{\mathbf{CTSA}}$, $\overline{\mathbf{CTSB}}$. Clear to Send (inputs, active LOW). If these pins are programmed as auto enables, a LOW on these inputs enables the respective transmitters. If not programmed as auto enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects transitions on these inputs and can interrupt the CPU on either logic level transitions.

$\overline{\mathbf{DCDA}}$, $\overline{\mathbf{DCDB}}$. Data Carrier Detect (inputs, active LOW). These pins function as receiver enables if they are programmed as auto enable bits ; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accomodate slow rise-time signals. The SCC detects transitions on these pins and can interrupt the CPU on either logic level transitions.

$\overline{\mathbf{DTR/REQA}}$, $\overline{\mathbf{DTR/REQB}}$. Data Carrier Detect (inputs, active LOW). These pins function as receiver enables if they are programmed into the DTR bit. They can also be used as general-purpose outputs (transmit) or as request lines for the DMA controller. The SCC allows full duplex DMA transfers.

IEI. Interrupt Enable In (input, active HIGH). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A HIGH on IEI indicates that no other higher priority

Figure 3 : DIP Pin Connections.

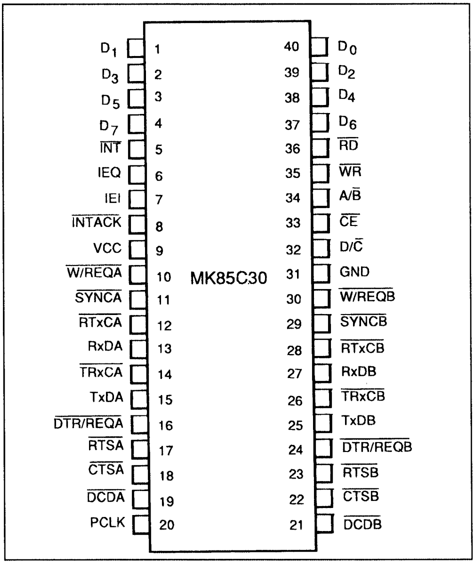
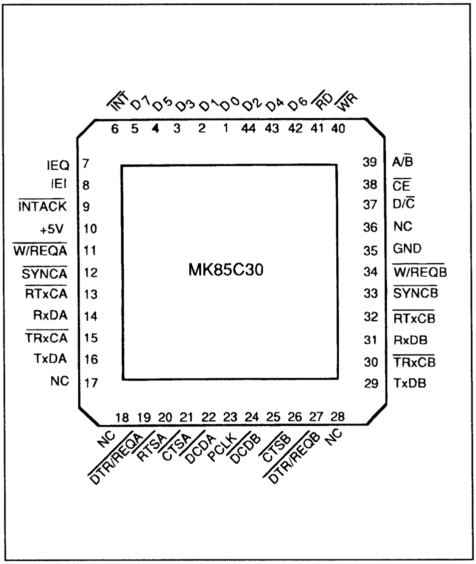


Figure 4 : Chip Carrier Pin Connection.



PIN DESCRIPTION (cont'd)

device has an Interrupt Under Service (IUS) or is requesting an interrupt.

IEO. Interrupt Enable Out (output, active HIGH). IEO is HIGH only if IEI is HIGH and the CPU is not servicing an SCC or SCC interrupt, or the controller is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

$\overline{\text{INTACK}}$. Interrupt Acknowledge (input, active LOW). This signal indicates an active interrupt acknowledge cycle. During this cycle, the interrupt daisy chain settles. When $\overline{\text{RD}}$ or $\overline{\text{DS}}$ becomes active, the SCC places an interrupt vector on the data bus (if IEI is HIGH). $\overline{\text{INTACK}}$ is latched by the rising edge of $\overline{\text{AS}}$ or PCLK.

$\overline{\text{INT}}$. Interrupt Request (output, open-drain, active LOW). This signal is activated when the SCC is requesting an interrupt.

PCLK. Clock (input). This is the master clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock. PCLK is a TTL level signal.

$\overline{\text{RTSA}}$, $\overline{\text{RTSB}}$. Request to Send (outputs, active LOW). When the Request to Send (RTS) bit in Write Register 5 (figure 48) is set, the $\overline{\text{RTS}}$ signal goes LOW. When the RTS bit is reset in the Asynchronous mode and auto enables is on, the signal goes HIGH after the transmitter is empty. In Synchronous mode or in Asynchronous mode with auto enables off, the $\overline{\text{RTS}}$ pins strictly follow the state of the RTS bit. Both pins can be used as general-purpose outputs.

RTxCA, RTxCB. Receive/Transmit Clocks (inputs, active LOW). The functions of these pins are under program control. In each channel, $\overline{\text{RTxC}}$ may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the digital phase-locked loop (refer to Section 4 for bit configurations). This pins can also be programmed for use the respective $\overline{\text{SYNC}}$ pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

RxDA, RxDB. Receive Data (inputs, active HIGH). These input signals receive serial data at standard TTL levels.

$\overline{\text{SYNCA}}$, $\overline{\text{SYNCB}}$. Synchronization (inputs/outputs, active LOW). These pins can act as either inputs, outputs, or as part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$. In this mode, transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 (figure 59), but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, $\overline{\text{SYNC}}$ must be driven LOW two receive clock cycles after the last bit in the sync character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of $\overline{\text{SYNC}}$.

In the Internal Synchronization mode, (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which sync characters are recognized. The sync condition is not latched, so these outputs are active each time a sync character is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

$\overline{\text{TRxCA}}$, $\overline{\text{TRxCB}}$. Transmit/Receive Clocks (inputs or outputs, active LOW). The functions of these pins are under program control. $\overline{\text{TRxC}}$ may supply the receive clock or the transmit clock in the Input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode. (Refer to Section 4 for bit configuration).

TxDA, TxDB. Transmit Data (outputs, active HIGH). This output signal transmits serial data at standard TTL levels.

$\overline{\text{W}}$/$\overline{\text{REQA}}$, $\overline{\text{W}}$/$\overline{\text{REQB}}$. Wait/Request (outputs, open drain when programmed for Wait function, driven HIGH or LOW when programmed for a Request function). These dual-purpose outputs can be programmed as Request (receive) lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait. The SCC allows full duplex DMA transfer.

OVERVIEW

The SCC internal structure provides all the interrupt and control logic necessary to interface with non-multiplexed bus. Interface logic is also provided to monitor modem or peripheral control inputs and outputs. All of the control signals are general purpose and can be applied to various peripheral devices as well as used for modem control.

The center for data activity revolves around the internal read and write registers. The programming of these registers provides the SCC with functional "personality" ; i.e., register values can be assigned before or during program sequencing to determine how the SCC will establish a given communication protocol.

Register Functions

All modes of communication are established by the bit values of the write registers. As data is received or transmitted, read register values may change. These changed values can promote software action or internal hardware action for further register changes.

The register set for each channel includes 14 write registers and seven read registers. Ten write registers are used for control, two for sync character generation, and two for baud rate generation. In addition there are two write registers which are shared by both channels ; one is the interrupt vector register and one is the master interrupt control and reset register. Four read registers indicate status information, two are for baud rate generation, and one for the receive buffer. In addition there are two read registers which are shared by both channels ; one for the interrupt pending bits and one for interrupt vector.

Table 1 lists the assigned functions for each read and write register. The SCC contains only one WR2 (interrupt vector) and one WR9 (master interrupt control). Both registers are accessed and shared by either channel. The Register Description section provides a detailed bit legend and description of each register.

Table 1 : Register Set.

	Read Register Functions
RR0	Transmitt/ Receive buffer status, and External status
RR1	Special Receive Condition status, residue codes, error conditions
RR2	Modified (Channel B only) interrupt vector and Unmodified interrupt vector (Channel A only)
RR3	Interrupt Pending bits (Channel A only)
RR8	Receive buffer
RR10	Miscellaneous XMTR, RCVR status parameters
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External / Status interrupt control information
	Write Register Functions
WR0	Command Register, CRC initialization resets for various modes
WR1	Interrupt conditions, Wait / DMA request control
WR2	Interrupt vector (access through either channel)
WR3	Receive / Control parameters, number of bits per character, RxCRC enable
WR4	Transmit / Receive miscellaneous parameters and codes, clock rate, number of sync characters, stop bits, parity
WR5	Transmit parameters and control, number of Tx bits per character, TxCRC enable
WR6	Sync character (1 st byte) or SDLC flag
WR8	Transmitt buffer
WR9	Master interrupt control and reset (accessed through either channel), reset bits, control interrupt daisy chain
WR10	Miscellaneous transmitter/receiver control bits, NRZI, NRZ, FM encoding, CRC reset
WR11	Clock mode control, source of Rx and Tx clocks
WR12	Lower byte of baud rate generator time constant
WR13	Upper byte of baud rate generator time constant
WR14	Miscellaneous control bits : baud rate generator, Phase-Locked Loop control, auto echo, local loopback
WR14	External/ Status interrupt control information-control external conditions causing interrupts

OVERVIEW (cont'd)

Data Paths

Figure 6 illustrates the data paths involved in the six major areas of the SCC :

- Transmitter
- Receiver
- Baud rate generator
- DPLL
- Clocking options
- Data encoding

All communication modes are established by programming the write registers. As data is received or transmitted, read register values may change, altering the direction of the data path. These changed values can promote software action or internal hardware action for further register changes.

Transmitter. The transmitter has an 8-bit Transmit Data register (WR8) loaded from the internal data bus and a Transmit Shift register loaded from either WR6, WR7, or the Transmit Data register. In byte-oriented modes, WR6 and WR7 can be programmed with sync characters. In Monosync mode, an 8-bit or 6-bit sync character is used (WR6), whereas a 16-bit sync character is used (WR6 and WR7) in Bisync mode. In bit-oriented synchronous modes, the flag contained in WR7 is loaded into the Transmit Shift register at the beginning and end of a message.

If asynchronous data is processed, WR6 and WR7 are not used and the Transmit Shift register is formatted with start and stop bits shifted out to the transmit multiplexer at the selected clock rate. Synchronous data (except SDLC/HDLC) is shifted to the CRC generator as well as to the transmit multiplexer.

SDLC/HDLC data is shifted to the CRC Generator and out through the zero insertion logic (which is disabled while the flags are being sent). A "0" is inserted in all address, control, information, and frame check fields following five continuous "1s" in the data stream. The result of the CRC generator for SDLC data is also routed through the zero insertion logic and then to the transmit multiplexer.

Receiver. The receiver has a three deep 8-bit Data FIFO (paired with an 8-bit Error FIFO), and an 8-bit shift register. This arrangement creates a 3-byte delay time, which allows the CPU time to service an interrupt at the beginning of a block of high-speed data. With each Receive Data FIFO, the error FIFO stores parity and framing errors and other types of status information. The error FIFO is readable in Read Register 1.

Incoming data is routed through one of several paths depending on the mode and character length. In Asynchronous mode, serial data enters the 3-bit delay (figure 5) if the character length of seven or eight bits is selected. If a character length of five or six bits is selected, data enters the receive shift register directly.

In synchronous modes, the data path is determined by the phase of the receive process currently in operation. A synchronous receive operation begins with a hunt phase in which a bit pattern that matches the programmed sync characters (6-bit, 8-bit, or 16-bit is searched).

The incoming data then passes through the Sync register and is compared to a sync character stored in WR6 or WR7 (depending on which mode it is in). The monosync mode matches the sync character programmed in WR7 and the character assembled in the Receive Sync register to establish synchronization.

Synchronization is achieved differently in the Bisync mode. Incoming data is shifted to the Receive Shift register while the next eight bits of the message are assembled in the Receive Sync register. If these two characters match the programmed characters in WR6 and WR7, synchronization is established. Incoming data can then bypass the Receive Sync register and enter the 3-bit delay directly.

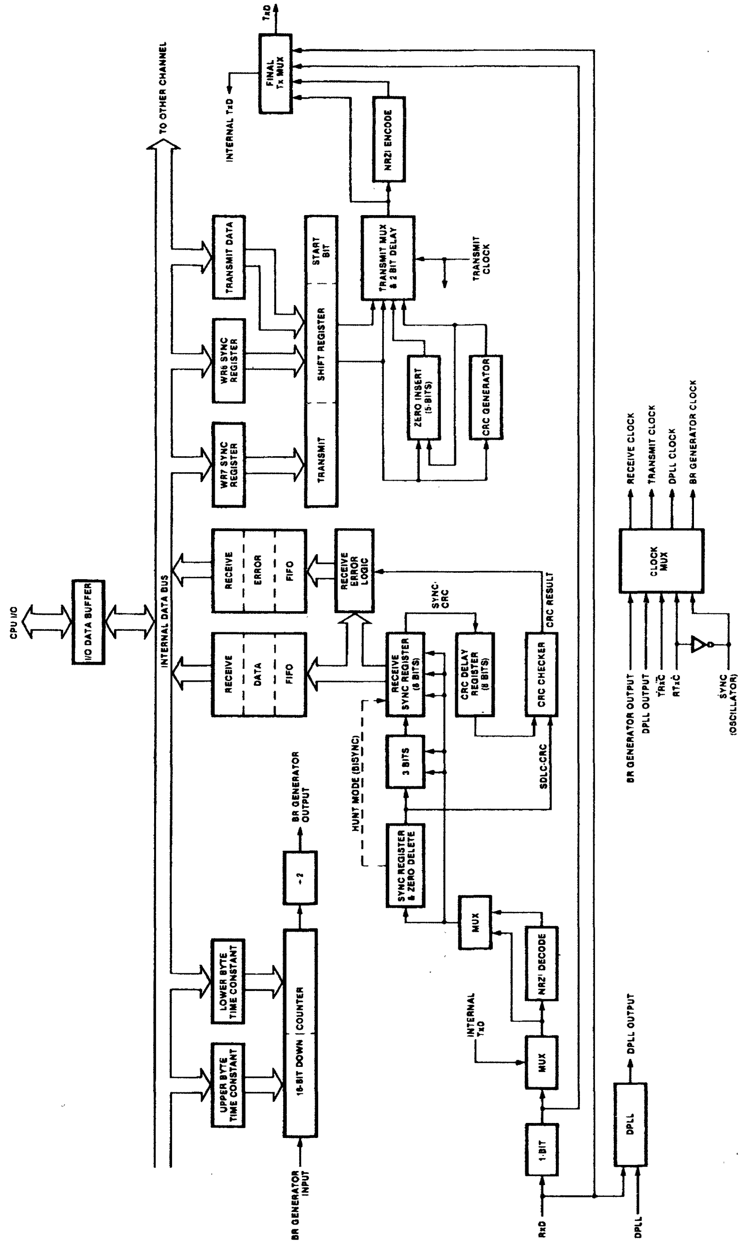
The SDLC mode of operation uses the Receive Sync register to monitor the receive data stream and to perform zero deletion when necessary ; i.e., when five continuous "1s" are received, the sixth bit is inspected and deleted from the data stream if it is "0". The seventh bit is inspected only if the sixth bit equals one. If the seventh bit is "0", a flag sequence has been received and the receiver is synchronized to that flag. If the seventh bit is a "1", an abort or an EOP (End Off Poll) is recognized, depending on the selection of either the normal SDLC mode or SDLC Loop mode.

The same path is taken by incoming data for both SDLC modes. The reformatted data enters the 3-bit delay and is transferred to the Receive Shift register. The SDLC receive operation begins in the hunt phase by attempting to match the assembled character in the Receive Shift Register with the flag pattern in WR7. Then the flag character is recognized, subsequent data is routed through the same path, regardless of character length.

Either the CRC - 16 or CRC - SDLC cyclic redundancy check (CRC) polynomial can be used for both Monosync and Bisync modes, but only the CRC - SDLC polynomial is used for SDLC operation. The data path taken for each mode is also different.

OVERVIEW (cont'd)

Figure 5 : Data Paths.



OVERVIEW (cont'd)

Bisync protocol is a byte-oriented operation that requires the CPU to decide whether or not a data character is to be included in CRC calculation. An 8-bit delay in all synchronous modes except SDLC is allowed for this process. In SDLC mode, all bytes are included in the CRC calculation.

Baud Rate Generator. Each channel in the SCC contains a programmable baud rate generator. Each generator consists of two 8-bit, time-constant registers forming a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output that makes the output a square wave. On start-up, the flip-flop on the output is set High so that it starts in a known state, the value in the time-constant register is again loaded into the counter, and the counter begins counting down. When a count of zero is reached, the output of the baud rate generator toggles, the value in the time-constant register is loaded into the counter, and the process starts over. The time constant can be changed at any time, but the new value does not take effect until the next load of the counter.

No attempt is made to synchronize the loading of a new time constant with the clock used to drive the generator. When the time constant is to be changed, the generator should be stopped by writing to an enable bit in WR14. This ensures the loading of the correct time constant.

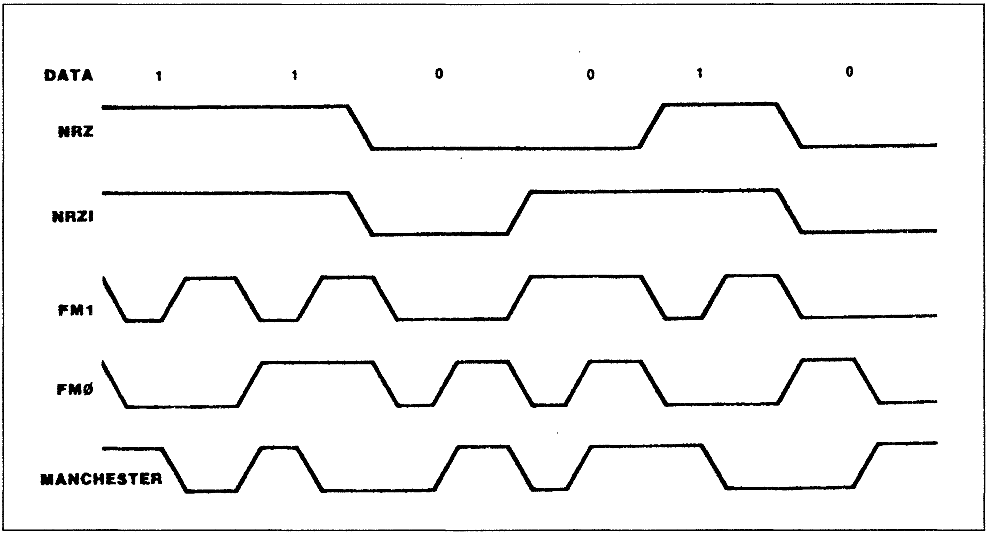
If neither the transmit clock nor the receive clock are programmed to come from the TRXC pin, the output of the baud rate generator may be made available for external use on the TRXC pin.

Digital Phase-locked Loop (DPLL). The SCC contains a digital phase-locked loop that can be used to recover clock information from a data stream with NRZI or FM coding. The DPLL is driven by a clock nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a receive clock for the data. This clock can then be used as the SCC receive clock, the transmit clock, or both.

Clocking Options. The SCC can select several clock sources for internal and external use. Write Register 11 is the Clock Mode Control register for both the receive and transmit clocks. It determines the type of signal on the SYNC and RTxC pins and the direction of the TRxC pin.

Write Register 11 also controls the output of the baud rate generator, the DPLL output, and the selection of either a TT1 or XTAL output for the RTxC pin.

Figure 6 : Data Encoding Methods.



OVERVIEW (cont'd)

Data Encoding. Figure 6 illustrates the four encoding methods used by the SCC. In NRZ encoding, a "1" is represented by a High level and a "0" is represented by a Low level. In NRZI encoding, a "1" is represented by no change in level and a "0" is represented by a change in level. In FM1 (more properly, biphase mark), a transition occurs at the beginning of every bit cell. A "1" is represented by an additional transition at the center of the bit cell and a "0" is represented by the absence of a transition at the center of the bit cell. In FM0 (more properly, biphase space), a transition occurs at the beginning of every bit cell. A "0" is represented by an additional transition at the center of the bit cell and a "1" is represented by the absence of a transition at the center of the bit cell.

In addition to these four methods, the SCC can be used to decode Manchester (biphase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is Low to High, the bit is "0". If the transition is High to Low, the bit is "1".

Data Communications Capabilities

SCC logic handles all asynchronous, byte-oriented synchronous, and bit-oriented synchronous modes of operation. The following section briefly describes asynchronous, synchronous, and SDLC modes of communication.

Asynchronous. Figure 7 represents a typical asynchronous message format using one start bit, seven data bits, one parity bit, and one stop bit. A start bit is a High-to-Low transition detected by an asynchronous receiver and is actually an information bit notifying the receiver of an incoming message.

The start bit also initiates a clock circuit to provide latching pulses during expected data bit intervals. The parity bit is provided for error checking. The parity bit is calculated in both the receiver and the transmitter ; the two results are compared to ensure that the expected and the actual bit values match. The stop bit returns the message unit to the quiescent marking state ; i.e., a constant high state condition lasts until the next High-to-Low start bit indicates an incoming data byte. During reception, the start and stop bits are stripped away and checked for errors, leaving only the working data for CPU interaction. The number of selected bits for each asynchronous function may differ between the transmitter and the receiver.

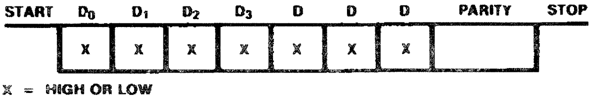
Monosync Mode. Monosync and Bisync modes require clocking information to be transmitted along with the data either by a method of encoding data that contains clocking information, or by a modem that encodes or decodes clock information in the modulation process.

Start and stop bits are not required in synchronous modes. All bits are used to transmit data. This eliminates the "waste" characteristic of asynchronous communication.

Figure 8 shows the character format for synchronous transmission. For example, bits 1-8 might be one character and bits 9-13 part of another character ; or bit 1 might be part of one character, bits 2-9 part of a second character, and bits 10-13 part of a third character. The framing (where each character begins) of each character is accomplished by defining a synchronization character, commonly called a "sync character".

The CPU places the receiver in Hunt mode whenever transmission begins (or whenever a data dropout has occurred and the hardware determines that resynchronization is necessary). In Hunt mode, the receiver shifts a bit into the Receive Shift register and compares the contents of the Receive Shift register and with the sync character (stored in another register), repeating the process until a match occurs. When a match occurs, the receiver begins transferring bytes to the receive FIFO.

Figure 7 : Asynchronous Message Format.



OVERVIEW (cont'd)

Bisynchronous Mode. The Bisync mode of operation (figure 9) is similar to the Monosync mode, except that two sync characters are provided instead of one. Bisync attemps a more structured approach to synchronization through the use of special characters as message "headers" or "trailers". A detailed description of IBM's Bisync can be found in McNamara's Book (see preface).

External Sync Mode. External Sync mode (figure 10) eliminates the use of sync characters in the serial data stream by providing an external sync signal to mark the beginning of a data field ; i.e., an external input pin (Sync) waits for an active state change to indicate the beginning of an information field.

SDLC Mode. Synchronous Data Link Control mode (SDLC) uses synchronization characters similar to Bisync and Monosync modes (such as flags and pad characters), but it is a bit-oriented protocol instead of byte-oriented protocol.

Any data communication link involves at least two stations. The station that is responsible for the data link and issues the commands to control the link is called the "primary station". The other station is a "secondary station". Not all information transfers need to be initiated by a primary station. In SDLC mode, a secondary station can be the initiator.

The basic format for SDLC is a "frame" (figure 11). The information field is not restricted in format or content and can be of any reasonable length (including zero). Its maximum length is that which can be expected to arrive at the receiver error-free most of time. Hence, the determination of maximum length is a function of communication channel error rate.

The two flags that delineate the SDLC frame serve as reference points when positioning the address and control fields, and they initiate the transmission error check. The ending flag indicates to the receiving station that the 16 bits just received constitute the frame check. The ending flag could be followed by another frame, another flag, or an idle. This means that when two frames follow one another, the intervening flag may simultaneously be the ending flag of the first frame and the beginning flag of the next frame. Since the SDLC mode does not use characters of defined length, but rather works on a bit-by-bit basis, the 01111110 (7EH) flag can be recognized at any time.

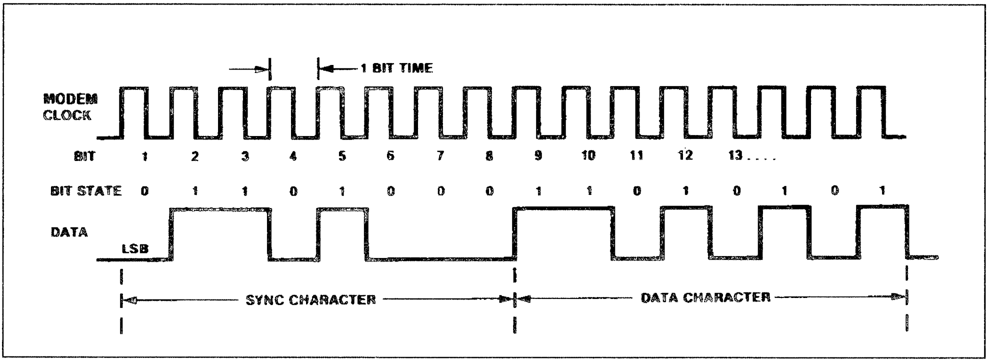
To ensure that the flag is not sent accidentally, SDLC procedures require a binary "0" to be inserted by the transmitter after the transmission of any five continuous "1s". The receiver then removes the "0" following a received succession of five "1s". Inserted and removed "0s" are not included in the CRC calculation.

The address field is 8 bits long and designates the number of secondary station to which the commands or data from the primary station are sent. The control field is eight bits long and is used to initiate all SDLC activities.

The SCC can also serve the High-level synchronous Data Link Communication (HDLC) protocol, which is identical to SDLC except for differences in framing.

SDLC Loop Mode. The SCC supports SDLC Loop mode in addition to normal SDLC. SDLC Loop mode is very similar to normal SDLC but is usually used in application where a point-to-point network is not

Figure 8 : Monosync Data Character Format.


1 BIT TIME
MODEM CLOCK
BIT 1 2 3 4 5 6 7 8 9 10 11 12 13....
BIT STATE 0 1 1 0 1 0 0 0 1 1 0 1 0 1 0 1
DATA
LSB
SYNC CHARACTER
DATA CHARACTER

OVERVIEW (cont'd)

appropriate (for example, Point-Of-Sale terminals). In an SDLC Loop there is a primary station, called the controller, that manages the message traffic flow on the loop, and there are any number of secondary stations.

A secondary station in an SDLC loop is always listening to the messages being sent around the loop, and must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can only place its own message on the loop at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages this bit pattern is unique and thus is easily recognized.

When a secondary station has a message to transmit and it recognizes an EOP on the line, the first thing that it does is to change the last 1 or the EOP to a "0" before transmitting it. This turns the EOP into a Flag sequence. The secondary station now places its message on the loop and terminates its message with an EOP. Any secondary stations further down the loop with messages to transmit can then append its message to the message of the first secondary station by the same process. All secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop, except upon recognizing an EOP.

There are also restrictions as to when and how a secondary station physically becomes part of the loop. A secondary station that has just powered up must monitor the loop, without the one-bit-time delay, until it recognizes an EOP. When an EOP is recognized the one-bit-time delay is switched on. This does not disturb the loop because the line is marking idle between the time that the controller sends the EOP and the time that it receives the EOP back. The secondary station that has gone on-loop cannot place a message on the loop until the next time that an EOP is issued by the controller. A secondary station goes off-loop in a similar manner. When given a command to go off-loop, the secondary station waits until the next EOP to remove the one-bit-time delay.

To operate the SCC in SDLC Loop mode, the SCC must first be programmed just as if normal SDLC were to be used. Loop mode is then selected by writ-

Figure 9 : Bisynchronous Message Format.

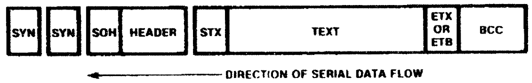


Figure 10 : External Sync Format.

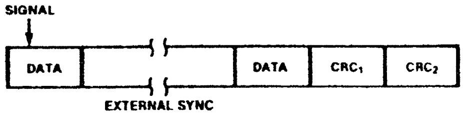
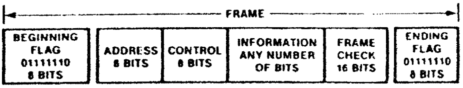


Figure 11 : SDLC Message Format.


FRAME
BEGINNING FLAG 01111110 8 BITS
ADDRESS 8 BITS
CONTROL 8 BITS
INFORMATION ANY NUMBER OF BITS
FRAME CHECK 16 BITS
ENDING FLAG 01111110 8 BITS

OVERVIEW (cont'd)

ing the appropriate control word in WR10. The SCC is now waiting for the EOP so that it can go on loop. While waiting for the EOP, the SCC ties TxD to RxD with only the internal gate delays in the signal path. When the first EOP is recognized by the SCC, the Break/Abort/EOP bit is set in RR0, generating an External/Status interrupt (if so enabled). At the same time, the On-Loop bit in RR10 is set to indicate that the SCC is indeed on-loop, and a one-bit time delay is inserted in the TxD to the RxD patch.

The SCC is now on-loop but cannot transmit a message until a flag and the next EOP are received. The requirement that a flag be received ensures that the SCC cannot erroneously send messages until the controller ends the current polling sequence and starts another one.

A secondary station on the loop is prohibited from transmitting a message during a polling sequence unless it captures the line at the moment the EOP passes by. The SCC does this automatically. If the CPU in the secondary station with SCC needs to transmit a message, the Go-Active-On-Poll bit in WR10 must be set. If this bit is set when the EOP is detected, the SCC changes the EOP to a flag and starts sending another flag. The EOP is reported in the Break/Abort/EOP bit in RR0 and the CPU should write its data bytes to the SCC, just as in normal SDLC frame transmission. When the frame is complete and CRC has been sent, the SCC closes with a flag and reverts to One-Bit-Delay mode. The last zero of the flag, along with the marking line echoed from the RxD pin, form an EOP for secondary stations further down the loop. If the Go-Active-On-Poll bit is not set at the time the EOP passes by, the SCC cannot send a message until a flag (terminating the current polling sequence) and another EOP are received. While the SCC is actually transmitting a message, the loop-sending bit in R10 is set to indicate this.

If SDLC loop is de-selected, the SCC is designed to exit from the loop gracefully. When SDLC Loop mode is de-selected by writing to WR10, the SCC waits until the next polling cycle to remove the on-bit time delay. If a polling cycle is in progress at the time the command is written, the SCC finishes sending any message that it may be transmitting, ends with an EOP, and disconnects TxD from RxD. If no message was in progress, the SCC immediately disconnects TxD from RxD. To ensure proper loop operation after the SCC goes off the loop, and until the external relays take the SCC completely out of the loop, the SCC should be programmed for Mark idle instead of Flag idle. When the SCC goes off the loop, the On-Loop bit is reset.

The SCC allows the user the option of using NRZI in SDLC Loop mode by programming WR20 appropriately. With NRZI encoding, the outputs of secondary stations in the loop may be inverted from their inputs because of messages that they have transmitted. Removing the stations from the loop (removing the one-bit time delay) may cause problems further down the loop because of extraneous transitions on the line. The SCC avoids this problem by making transparent adjustements at the end of each frame it sends in response to an EOP. A response frame from the SCC is terminated by a flag and an EOP. Normally, the flag and the EOP share a zero, but if such sharing would cause the RxD and TxD pins to be of opposite polarity after the EOP, the SCC adds another zero between the flag and the EOP. This causes an extra line transition so that RxD and TxD are identical after the EOP is sent. This extra zero is completely transparent because it only means that the flag and the EOP no longer share a zero. All that a proper loop exit needs, therefore, is the removal of the one-bit time delay.

I/O Capabilities

The SCC can work with three basic forms of I/O operations : polling, interrupts, and block transfer. All three I/O types involve register manipulation during initialization and data transfer.

Polling. During a polling sequence, the status of Read Register 0 is examined in each channel. This register indicates whether or not a receive or transmit data transfer is needed and whether or not any special conditions are present, e.g., errors.

This method of I/O transfer avoids interrupts. All interrupt functions must be disabled in order to operate the device in a polled environment. With no interrupts enabled, this mode of operation must initiate a read cycle of Read Register 0 to detect an incoming character before jumping to a data handler routine.

Interrupts. The SCC provides interrupt capability through the use of pins and a hardware scheme that enhances the maximum speed of serial data. Whenever the interrupt ($\overline{INT}$) pin is active, the SCC is ready to transfer data.

Read and write registers are programmed so that an interrupt vector points to an interrupt service routine. The interrupt vector can also be modified to reflect various status conditions. Therefore, as many as eight different interrupt routines can be referenced.

Transmit interrupts, receive interrupts, and external/status interrupts are the main sources of interrupts. Each interrupt source is enabled under

OVERVIEW (cont'd)

program control, with channel A having a higher priority than channel B and with receive, transmit, and external/status interrupts prioritized respectively within each channel.

Block Transfers. The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{W}/\overline{REQ}$ output in conjunction with the Wait/Request bits in Write Register 1.

The $\overline{W}/\overline{REQ}$ output can be defined by software as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

To a DMA controller, the SCC REQUEST output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the $\overline{WAIT}$ output indicates that the SCC is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle.

INTERFACING THE SCC

This section covers the details of interfacing the MK85C30 to a system. The general timing requirements for the device are described in the hardware information section.

Interfacing the MK85C30

Two control signals, $\overline{RD}$ and $\overline{WR}$, are used by the MK85C30 to time bus transactions. In addition, four other control signals, $\overline{CE}$, D/$\overline{C}$, A/$\overline{B}$ and $\overline{INTACK}$, are used to control the type of bus transaction that will occur. A bus transaction start when the addresses on D/$\overline{C}$ and A/$\overline{B}$ are asserted before $\overline{RD}$ or $\overline{WR}$ fall. The coincidence of $\overline{CE}$ and $\overline{RD}$ or $\overline{CE}$ and $\overline{WR}$ latches the state of D/$\overline{C}$ and A/$\overline{B}$ and starts the internal operation. The $\overline{INTACK}$ signal must have been previously sampled High by a rising edge of PCLK for a read or write cycle to occur. In addition to sampling $\overline{INTACK}$, PCLK is used by the interrupt section to set the IP bits. The MK85C30 generates internal control signals in response to a register access. Since $\overline{RD}$ and $\overline{WR}$ have no phase relationship with PCLK, the circuitry generating these internal control signals provides time for metastable conditions to disappear. This results in a recovery time related to PCLK. This recovery time applies only between transactions involving the MK85C30, and any interventing transactions are ignored. This recovery time is four PCLK cycles, measured from the falling edge of $\overline{RD}$ or $\overline{WR}$ in the case of a read or write of any register.

MK85C30 Read Cycle Timing. The Read cycle timing for the MK85C30 is shown in figure 12. The address on A/$\overline{B}$ and D/$\overline{C}$ is latched by the coincidence of $\overline{RD}$ and $\overline{CE}$ active. $\overline{CE}$ must remain LOW and $\overline{INTACK}$ must remain HIGH throughout the cycle. The MK85C30 bus drivers are enabled while $\overline{CE}$ and $\overline{RD}$ are both LOW. A read with D/$\overline{C}$ HIGH does not disturb the state of the pointers and a read cycle with D/$\overline{C}$ LOW resets the pointers to zero after the internal operation is complete.

MK85C30 Write Cycle Timing. The Write cycle timing for the MK85C30 is shown in figure 13. The address on A/$\overline{B}$ and D/$\overline{C}$, as well as the data on D0-D7, is latched by the coincidence of $\overline{WR}$ and $\overline{CE}$ active. $\overline{CE}$ must remain LOW and $\overline{INTACK}$ must remain HIGH throughout the cycle. A write cycle with D/C HIGH does not disturb the state of pointers and a write cycle with D/$\overline{C}$ LOW resets the pointers to zero after the internal operation is complete.

MK85C30 Interrupt Acknowledge Cycle Timing. The interrupt Acknowledge cycle timing for the MK85C30 is shown in figure 14. The state of $\overline{INTACK}$ is latched by the rising edge of PCLK. While $\overline{INTACK}$ is LOW, the state of A/$\overline{B}$, $\overline{CE}$, D/$\overline{C}$, and $\overline{WR}$ are ignored. Between the time $\overline{INTACK}$ is first sampled LOW and the time $\overline{RD}$ falls, the internal and external IEI/IEO daisy chains settle ; this is A.C. parameter #38 TdIAi(RD). If there is an interrupt pending in the MK85C30, and IEI is HIGH when RD falls, the Interrupt Acknowledge cycle was intended for the MK85C30. This being the case, the MK85C30 sets the appropriate Interrupt Under Service latch, and places an interrupt vector on D0-D7. If the falling edge of $\overline{RD}$ sets an IUS bit in the MK85C30, the $\overline{INT}$ pin goes active in response to the falling edge. Note that there should be only one RD per Acknowledge cycle. Another important fact is that the IP bits in the MK85C30 are updated by PCLK divided by two, and this clock to update IPs is stopped while the pointers point to RR2 and RR3. This prevents data changing during a read, but will delay interrupt requests if the pointers are left pointing at these registers.

MK85C30 Register Access. The registers in the MK85C30 are accessed in a two-step process, using a Register Pointer to perform the addressing. To access a particular register, the pointer bits must be set by writing to WR0 the pointer bits may be written in either channel because only one set exists in

INTERFACING THE SCC (cont'd)

the MK85C30. After the pointer bits are set, the next read or write cycle of the MK85C30 having D/C̅ LOW will access the desired register. At the conclusion of this read or write cycle the pointer bits are reset to "0s", so that the next control write will be to the pointers in WR0. A read or RR8 (the receive data buffer) or a write to WR8 (the transmit data buffer) may either be done in this fashion or by accessing the MK85C30 having D/C̅ pin HIGH. A read or write with D/C̅ HIGH accesses the data registers directly, and independently, of the state of the pointer bits. This allows single-cycle access to the data registers and does not disturb the pointer bits. The fact that the pointer bits are reset to "0", unless explicitly set otherwise, means that WR0 and RR0 may also be accessed in a single cycle. That is, it is not necessary to write the pointer bits with "0" before accessing WR0 or RR0. There are three pointer bits in WR0, and these allow access to the registers with addresses 0 through 7. Note that a command may

Figure 12 : MK85C30 Read Cycle Timing.

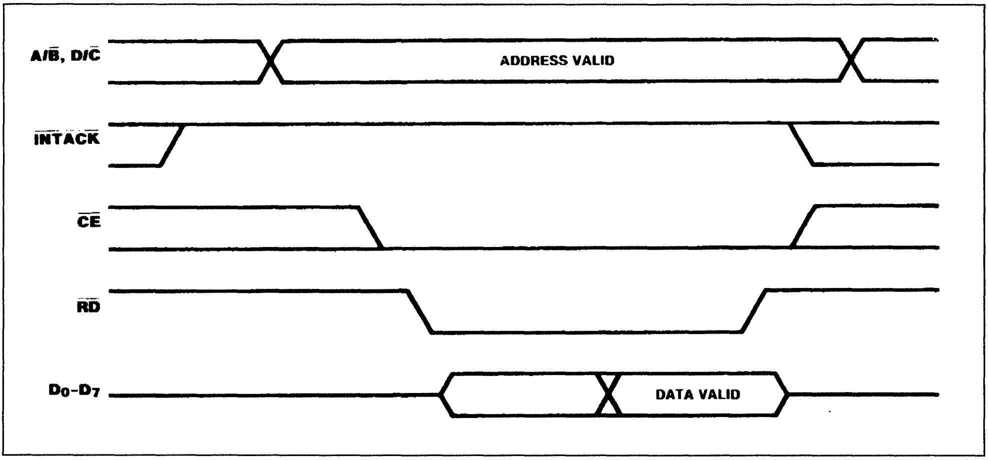
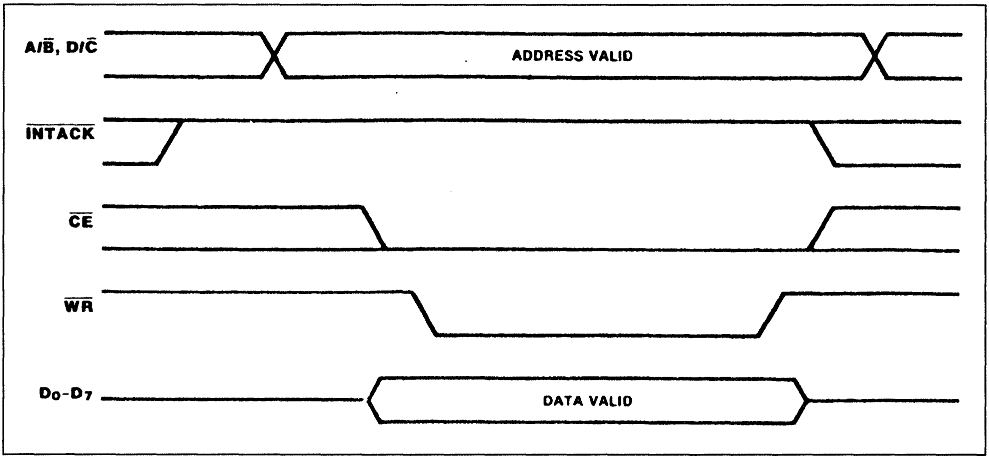
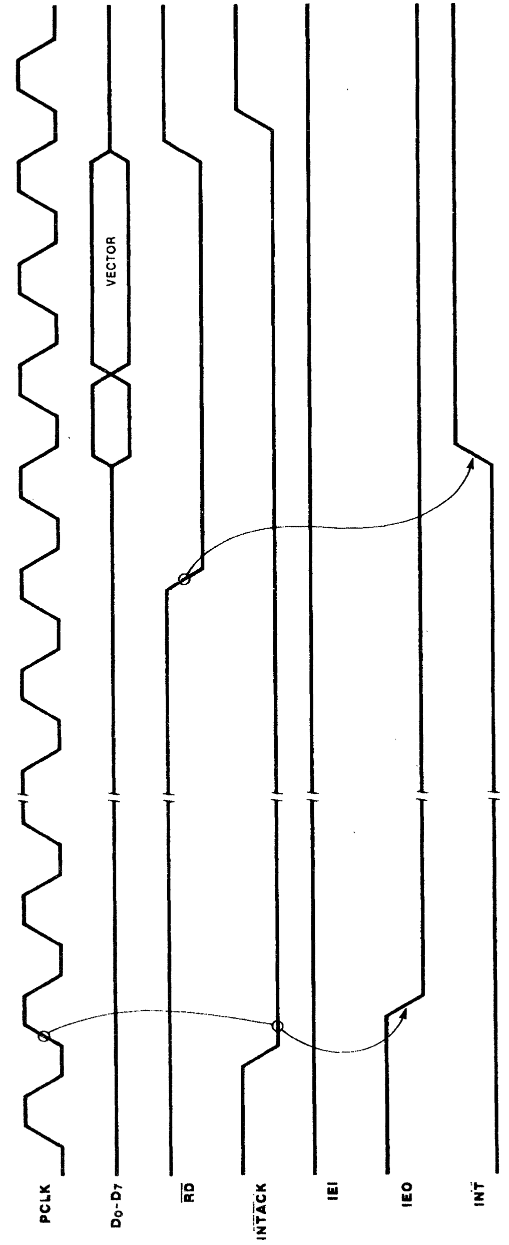


Figure 13 : MK85C30 Write Cycle Timing.



INTERFACING THE SCC (cont'd)

Figure 14 : MK85C30 Interrupt Acknowledge Details.



INTERFACING THE SCC (cont'd)

be written to WR0 at the same time that the pointer bits are written. To access the registers with addresses 8 through 15, a special command must accompany the pointer bits. This precludes concurrentiy issuing a command when pointing to these registers. The register map for the MK85C30 is shown in table 2. If for some reason, the state of the pointer bits is unknown they may be reset to "0" by performing a read cycle with the D/$\overline{C}$ pin held LOW. Once the pointer bits have been set, the desired channel is selected by the state of the A/B pin during the actual read or write of the desired register.

MK85C30 Reset. The MK85C30 may be reset by either hardware or software. Hardware reset occurs when $\overline{RD}$ and $\overline{WR}$ are both LOW, simultaneously, which is normally an illegal condition. As long as both $\overline{RD}$ and $\overline{WR}$ are LOW, the MK85C30 recognizes the reset condition. Once this condition is removed, however, the reset condition is asserted internally for an additional four to five PCLK cycles. During this time any attempt to access the MK85C30 will be ignored. The MK85C30 has three software resets, encoded into command bits in WR9. There are two channel resets, which affect only one channel in the device and some of the bits in the write registers. The third command forces the same result as does a hardware reset. As in the case of a hardware reset, the MK85C30 stretches the reset signal an additional four to five PCLK cycles beyond the ordinary valid access recovery time. The bits in WR9 may be written at the same time as the reset command because these bits are affected only by a hardware reset.

The reset values of the various registers are shown in Figure 15.

Table 2 : MK85C30 Register Map.

A/B	PNT$_2$	PNT$_1$	PNT$_0$	WRITE	READ
0	0	0	0	WR0B	RR0B
0	0	0	1	WR1B	RR1B
0	0	1	0	WR2	RR2B
0	0	1	1	WR3B	RR3B
0	1	0	0	WR4B	(RR0B)
0	1	0	1	WR5B	(RR1B)
0	1	1	0	WR6B	(RR2B)
0	1	1	1	WR7B	(RR3B)
1	0	0	0	WR0A	RR0A
1	0	0	1	WR1A	RR1A
1	0	1	0	WR2	RR2A
1	0	1	1	WR3A	RR3A
1	1	0	0	WR4A	(RR0A)
1	1	0	1	WR5A	(RR1A)
1	1	1	0	WR6A	(RR2A)
1	1	1	1	WR7A	(RR3A)
With the Point High Command					
0	0	0	0	WR8B	RR8B
0	0	0	1	WR9	RR13B
0	0	1	0	WR10B	RR10B
0	0	1	1	WR11B	(RR15B)
0	1	0	0	WR12B	RR12B
0	1	0	1	WR13B	RR13B
0	1	1	0	WR14B	(RR10B)
0	1	1	1	WR15B	RR15B
1	0	0	0	WR8A	RR8A
1	0	0	1	WR9	(RR13A)
1	0	1	0	WR10A	RR10A
1	0	1	1	WR11A	(RR15A)
1	1	0	0	WR12A	RR12A
1	1	0	1	WR13A	RR13A
1	1	1	0	WR14A	RR14A
1	1	1	1	WR15A	RR15A

INTERFACING THE SCC (cont'd)

Figure 15 : MK85C30 Register Reset Values.

HARDWARE RESET								CHANNEL RESET								
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WR0
0	0	.	0	0	.	0	0	0	0	.	0	0	.	0	0	WR1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR2
.	.	.	.	.	.	.	0	.	.	.	.	.	.	.	0	WR3
.	.	.	.	.	1	.	.	.	.	.	.	.	1	.	.	WR4
0	.	.	0	0	0	0	.	0	.	.	0	0	0	0	.	WR5
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR6
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR7
1	1	0	0	0	0	.	.	.	.	0	.	.	.	.	.	WR9
0	0	0	0	0	0	0	0	0	.	.	0	0	0	0	0	WR10
0	0	0	0	1	0	0	0	.	.	.	.	.	.	.	.	WR11
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR12
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR13
.	.	1	0	0	0	0	0	.	.	1	0	0	0	.	.	WR14
1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	WR15
0	1	.	.	.	1	0	0	0	1	.	.	.	1	0	0	RR0
0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	RR1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RR3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RR10

I/O PROGRAMMING CAPABILITIES

Regardless of the version of the SCC, all communication modes can use a choice of polling, interrupt and block transfer. These modes must be selected by the user to select the proper hardware and software required to supply data at the rate required.

Polling

This is the simpliest mode to implement. The software must poll the SCC to determine when data is to be inputted or outputted from the SCC. In this mode, MIE (WR9 bit 3), and Wait/DMA Request Enable (WR1 bit 7) are both reset to 0 to disable any requests. The software must then poll RR0 to determine the status of the receive buffer, transmit buffer and external status.

Interrupt Operations

The SCC, as a microprocessor peripheral, will request an interrupt only when it needs servicing. This allows the CPU to perform other operations while the SCC does not need service. The SCC has an internal priority resolution method to allow the highest priority interrupt to be serviced first.

The SCC is flexible with its interrupt method. The interrupt may be acknowledged with a vector transferred, acknowledged without a vector, or not acknowledged at all.

Interrupt without Acknowledge. In this mode, the Interrupt Acknowledge signal does not have to be generated. This allows a simpler hardware design that does not have to meet the Interrupt acknowledge timing. Soon after the $\overline{\text{INT}}$ goes active, the interrupt controller will jump to the interrupt routine. In the interrupt routine, the code must read RR2 from Channel B to read the vector including status. When the vector is read from Channel B, it always includes the status regardless of the VIS bit (WR9 bit 0). The status given will decode the highest priority interrupt pending at the time it is read. The vector is not latched so that the next read could produce a different vector if another interrupt occurs. The register is disabled from change during the read operation to prevent an error if a higher interrupt occurs exactly during the read operation.

Once the status is read, the interrupt routine must decode the interrupt pending, and clear the condition. Removing the interrupt condition will clear the IP and bring $\overline{\text{INT}}$ inactive, as long as there are no other IP bits set. For example, writing a character to the transmit buffer will clear the transmit buffer empty IP.

When the interrupt IP, decoded from the status, is cleared RR2 can be read again. This allows the interrupt routine to clear all of the IP's within one interrupt request to the CPU.

Interrupt with Acknowledge. After the SCC brings $\overline{\text{INT}}$ active, the CPU must respond by bringing $\overline{\text{INTACK}}$ active. After enough time has elapsed to allow the daisy-chain to settle, the SCC will set the IUS bit for the highest priority IP. If the No Vector bit is not set (WR9 bit 1), the SCC will then place the interrupt vector on the data bus during a read. To speed the interrupt response time, the SCC can also modify 3 bits in the vector to indicate status. To include the status, the VIS bit (WR9 bit 0) must be set. The service routine must then clear the interrupting condition. For example, writing a character to the transmit buffer will clear the transmit buffer empty IP. After the interrupting condition is cleared, the routine can read RR3 to determine if any other IP's are set and clear them. At the end of the interrupt routine, a Reset IUS command (WR0) must be issued to unlock the daisy-chain and allow lower-priority interrupt requestes. This is the only way, short of a software or hardware reset, that an IUS bit may be reset.

If the No Vector bit (WR9 bit 1) is set, the SCC will not place the vector on the data bus. An interrupt controller must then vector the code to the interrupt routine. The interrupt routine must then read RR2 from Channel B to read the status. This is the same as the case of an interrupt without an acknowledge except the IUS is set and the vector will not change until the Reset IUS command in RR0 is issued.

Interrupt Sources. Each channel in the SCC contains 3 sources of interrupt, making a total of 6. These 3 sources of interrupts are the receiver, the transmitter, and External/Status conditions. In addition, there are several conditions that may cause these interrupts.

The receive interrupt request may either be caused by a receive character available or a special condition. The receive character available interrupt is generated when a character is loaded into the FIFO and is ready to be read. The special conditions are receive FIFO overrun, CRC/ framming error. End of frame, and parity. The parity special condition can be included as a special condition or not depending on bit 2 WR1. The special condition status can be read from RR1.

The transmit interrupt request has only one source. It can only be set when the transmit buffer goes from

I/O PROGRAMMING CAPABILITIES (cont'd)

full to empty. Note that this means that the transmit interrupt will not be set until after the first character is written to the SCC.

The External/status Interrupts have several sources which may be individually enabled in WR15. The sources are zero count, DCD, Sync/ Hunt, CTS, transmitter underrun/EOM and Break/Abort.

Each source of interrupt in the SCC has three control/status bits associated with it. There are Interrupt Enable (IE), Interrupt Pending (IP), and Interrupt Under Service (IUS) (see figure 16). The IE bit is written by the processor and serves to control interrupt requests from the SCC. If the IE bit is set for a given source of interrupt, then that source may cause an interrupt request when all of the necessary conditions are met. If the IE bit is reset, no interrupt request will be generated by that source. The IE bits are write-only in the SCC. The IP bit for a given source of interrupt condition in the SCC and is reset directly by the processor, or indirectly by some action that the processor may take. If the corresponding IE bit is not set, the IP bits in the SCC are read-only in RR3A. The IUS bits are completely hidden from the processor's view. An IUS is set during an Interrupt Acknowledge cycle for the highest-priority IP. See table 3 for the interrupt priority. IUS is used to control the operation of the interrupt daisy chain by masking lower-priority interrupts. At the end of an interrupt service routine, the processor must issue a Reset Highest IUS command in WR0 to allow lower-priority interrupts. This is the only way, short of a software or hardware reset, that an IUS bit may be reset.

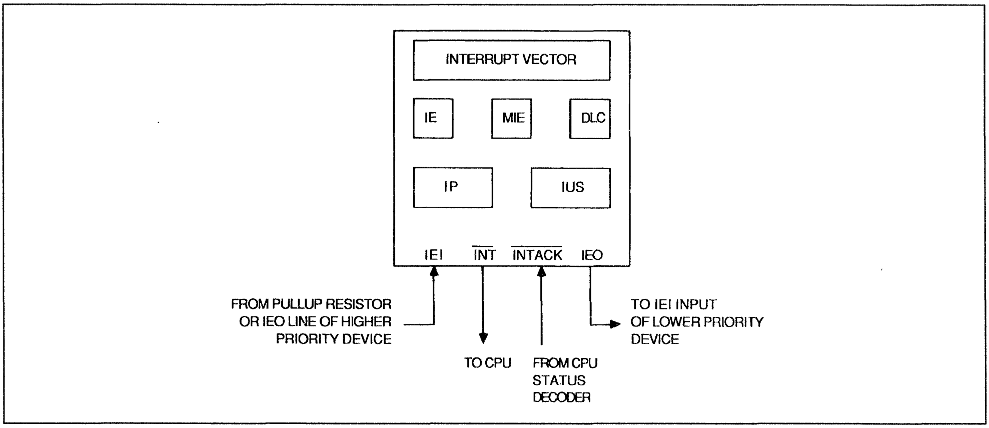
Table 3 : Interrupt Source Priority.

Receiver Channel A Transmit Channel A	High
External / Status Channel A	↓
Receiver Channel B Transmit Channel B	↓
External / Status Channel B	Low

Daisy-chain Priority Resolution. The six sources of interrupt in the SCC are prioritized in a fixed order via a daisy chain ; provision is made, via the IEI and IEO pins, for use of an external daisy chain as well. All Channel A interrupts are higher-priority than any Channel B interrupts, with the receiver, transmitter, and External/Status interrupts prioritized in that order within each channel. The SCC requests an interrupt by pulling the $\overline{\text{INT}}$ pin Low from its open-drain state. This is controlled by the IP bits and the IEI input, among other things. A flowchart of the interrupt sequence for the SCC is shown in figure 17. The internal daisy chain links the six sources of interrupt in a fixed order, chaining the IUS bits for each source. While an IUS is set, all lower-priority interrupt requests are masked off ; during an Interrupt Acknowledge cycle the IP bits are also gated into the daisy chain. This insures that the highest-priority IP will be selected to have its IUS set. The internal daisy chain may be controlled by the MIE bit in WR9. This bit, when reset, has the same effect as pulling the IEI pin Low, thus disabling all interrupt requests.

External Daisy Chain Operations. The SCC generates an interrupt request by pulling $\overline{\text{INT}}$ Low, but only if such interrupt requests are enabled

Figure 16 : Peripheral Interrupt Structure.


INTERRUPT VECTOR
IE
MIE
DLC
IP
IUS
IEI
INT
INTACK
IEO
FROM PULLUP RESISTOR OR IEO LINE OF HIGHER PRIORITY DEVICE
TO CPU
FROM CPU STATUS DECODER
TO IEI INPUT OF LOWER PRIORITY DEVICE

I/O PROGRAMMING CAPABILITIES (cont'd)

(IE is 1, MIE is 1), an IP is set without a higher-priority IUS being set, or no higher-priority IUS being set, or no higher-priority interrupt is being serviced (IEI is High), or no Interrupt Acknowledge transaction is taking place. It is not pulled Low by the SCC at this time, but instead continues to follow IEI until an Interrupt Acknowledge transaction occurs. Some time after $\overline{INT}$ has been pulled Low, the processor initiates an Interrupt Acknowledge transaction. Between the time the SCC recognizes that an Interrupt Acknowledge cycle is in progress and the time during the acknowledge that the processor requests an interrupt vector, the IEI/IEO daisy chain settles. Any peripheral in the daisy chain having an Interrupt Pending (IP is 1) or an Interrupt Under Service (IUS is 1) holds its IEO line Low and all others make IEO follow IEI.

When the processor requests an interrupt vector, only the highest-priority interrupt source with a pending interrupt (IP is 1) has its IEI input High, its IE bit set to "1", and its IUS bit set to "0". This is the interrupt source being acknowledged, and at this point it sets its IUS bit to "1". If its NV bit is "0", the SCC identifies itself by placing the interrupt vector from WR2 on the data bus. If the NV bit is "1", the SCC data bus remains floating, allowing external logic to supply a vector. If the VIS bit in the SCC is "1", the vector also contains status information, encoded as shown in table 4, which further describes the nature of the SCC interrupt. If the VIS bit is "0", the vector held in WR2 is returned without modification. If the SCC is programmed to include status information in the vector, this status may be encoded and placed in either bits 1-3 or in bits 4-6. This operation is selected by programming the Status High/Status Low bit in WR9. At the end of the interrupt service routine, the processor should issue the Reset Highest IUS command to unlock the daisy chain and allow lower-priority interrupt requests. The IP is reset during the interrupt service routine either directly by command, or indirectly, through some action taken by the processor. The external daisy chain may be controlled by the DLC bit in WR9. This bit, when set, forces IEO Low, disabling all lower-priority devices.

Receive Interrupts. The Receive Interrupt mode is controlled by WR1 bits 4 and 3. These select one of the four interrupt modes. The four modes are, Interrupt disabled, Interrupt on first character or special conditions, Interrupt on all characters or special conditions, and Interrupt on special conditions.

Receive Interrupts Disabled. This mode prevents the receiver from requesting an interrupt. It is used in a polled environment where either the status bits in RR0 or the modified vector in RR2 (Channel B) is read. Although the receiver interrupts are disabled, the interrupt logic can still be used to provide status.

Table 4 : Interrupt Vector Modification.

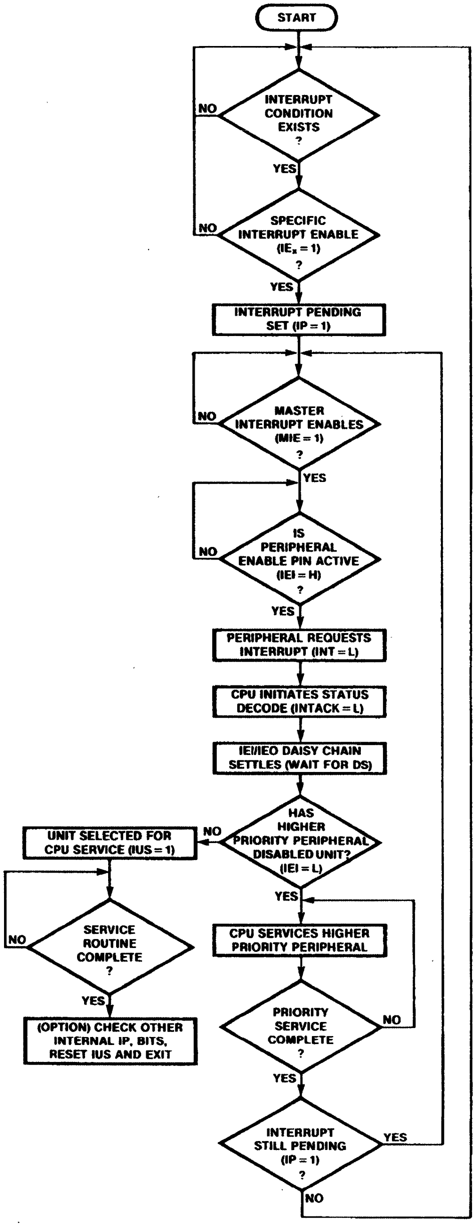
V3	V2	V1	Status High / status Low = 0
V4	V5	V6	Status High / status Low = 1
0	0	0	Ch B Transmit Buffer Empty
0	0	1	Ch B External / Status Change
0	1	0	Ch B Receive Character Avail.
0	1	1	Ch B Special Receive Condition
1	0	0	Ch A Transmit Buffer Empty
1	0	1	Ch A External / status Change
1	1	0	Ch A Receive Character Avail.
1	1	1	Ch A Special Recieve Condition

When these bits indicate that a received character has reached the top of the FIFO, the status in RR1 should be checked and then the data should be read. If status is to be checked, it must be done before the data is read, because the act of reading the data pops both the data and error FIFOs.

Receive Interrupt on First Character or Special Condition. This mode is designed for use with DMA transfers of the receive characters. After this mode is selected, the first character received, or the first character already stored in the FIFO, will set the receiver IP. This IP will be reset when this character is removed from the SCC. No further receive interrupts will occur until the processor issues an Enable Interrupt on Next Receive Character command in WR0 until a special receive condition occurs. The SCC recognizes several special receive conditions. A receive overrun (where a character in the FIFO is written over) is a special receive condition, as is a framing error in Asynchronous mode, or the end-of-frame condition in SDLC mode. In addition, if D2 of WR1 is set, any character with a parity error will generate a special receive condition interrupt. The correct sequence of events when using this mode is to first select the mode and wait for the receive character available interrupt. When the interrupt occurs the processor should read the character and then enable the DMA to transfer the remaining characters. A special receive condition interrupt may occur any time after the first character is received, but is guaranteed to occur after the character having the special condition has been read. The status is not lost in this case, however, because the FIFO is locked by the special condition. In the service routine the processor should read RR1 to obtain the status, and may read the data again if necessary. The FIFO is unlocked by issuing an Error Reset command in WR0. If the special condition was End-

I/O PROGRAMMING CAPABILITIES (cont'd)

Figure 17 : Interrupt Flowchart.


START
INTERRUPT CONDITION EXISTS ?
NO
YES
SPECIFIC INTERRUPT ENABLE (IEₓ = 1) ?
NO
YES
INTERRUPT PENDING SET (IP = 1)
MASTER INTERRUPT ENABLES (MIE = 1) ?
NO
YES
IS PERIPHERAL ENABLE PIN ACTIVE (IEI = H) ?
NO
YES
PERIPHERAL REQUESTS INTERRUPT (INT = L)
CPU INITIATES STATUS DECODE (INTACK = L)
IEI/IEO DAISY CHAIN SETTLES (WAIT FOR DS)
HAS HIGHER PRIORITY PERIPHERAL DISABLED UNIT? (IEI = L)
NO
UNIT SELECTED FOR CPU SERVICE (IUS = 1)
SERVICE ROUTINE COMPLETE ?
NO
YES
(OPTION) CHECK OTHER INTERNAL IP, BITS, RESET IUS AND EXIT
YES
CPU SERVICES HIGHER PRIORITY PERIPHERAL
PRIORITY SERVICE COMPLETE ?
NO
YES
INTERRUPT STILL PENDING (IP = 1) ?
YES
NO

I/O PROGRAMMING CAPABILITIES (cont'd)

of-Frame, the processor should now issue the Enable Interrupt on Next Receive Character command to prepare for the next frame. The first character interrupt and special condition interrupt are distinguished by the status included in the interrupt vector. In all other respects they are identical, including sharing the IP and IUS bits.

Interrupt on all Receive Characters or Special Conditions. This mode is designed for an interrupt-driven system. In this mode the SCC will set the receiver IP on every received character, whether or not it has a special receive condition. This includes characters already in the FIFO when this mode is selected. In this mode of operation the IP is reset when the character is removed from the FIFO, so if the processor requires status for any character, this status must be read before the data is removed from the FIFO. The special receive conditions are identical to those previously mentioned, and as before, the only difference between a "receive character available" interrupt and a "special receive condition" interrupt is the status encoded in the vector. In this mode a special receive condition does not lock the receive data FIFO so that the service routine must read the status in RR1 before reading the data. At moderate to high data rates, where the interrupt overhead is significant, time can usually be saved by checking for another received character before exiting the service routine. This technique eliminates the Interrupt Acknowledge and the processor-state-saving time, but care must be exercised because this receive character must be checked for special receive conditions before it is removed from the SCC.

Receive Interrupt on Special Conditions. This mode is designed for use with DMA transfers of the receive characters. In this mode, only receive characters with special conditions will cause the receiver IP to be set. All other characters are assumed to be transferred via DMA. No special initialization sequence is needed in this mode. Usually the DMA is initialized and enabled, and then this mode is selected in the SCC. A special receive condition interrupt may occur at any time after this mode is selected but the logic guarantees that the interrupt will not occur until after the character with the special condition has been read from the SCC. The special condition locks the FIFO so that the status will be valid when read in the interrupt service routine, and it guarantees that the DMA will not transfer any characters until the special condition has been serviced. In the service routine the processor should read RR1 to obtain the status and unlock the FIFO by issuing an Error Reset command. DMA transfer of the receive characters will then resume.

Transmit Interrupts. Transmit interrupts are controlled by the Transmit Interrupt Enable bit (D1) in WR1. If the interrupt capabilities of the SCC are not required, polling may be used. This is selected by disabling the transmit interrupts and polling the Transmit Buffer Empty bit in RR0. When the Transmit Buffer Empty is set a character may be written to the SCC without fear of writing over previous data. Another way of polling the SCC is to enable the transmit interrupt and then reset the MIE bit in WR9. The processor may then poll the IP bits in RR3A to determine when the transmit buffer is empty. Transmit interrupts should also be disabled in the case of DMA transfer of the transmitted data.

While the transmit interrupts are enabled the SCC will set the transmit IP whenever the transmit buffer becomes empty. This means that the transmit buffer must have been full before the transmit IP can be set. Thus when the transmit interrupts are first enabled, the transmit IP will not be set until after the first character is written to the SCC. In synchronous modes one other condition can cause the transmit IP to be set. This occurs at the end of a transmission after CRC is sent. When the last bit of CRC has cleared the Transmit Shift register and the flag or sync character is loaded into the Transmit Shift register, the SCC will set the transmit IP. Data for the new frame or block to be transmitted may be written at this time. In this particular case the Transmit Buffer Empty bit in RR0 is not set ; only the transmit IP is set. If the transmit Buffer Empty bit is, in fact, set for the transmit interrupt that occurs immediately after CRC transmission, this indicates that data was written while CRC was being sent. This is an indication that the transmitter underflowed, without the CPU being aware of it. The transmit IP is reset either by writing data to the transmit buffer or by issuing the Reset Transmit IP command in WR0. Ordinarily the response to a transmit interrupt is to write more data to the SCC ; however, at the end of a frame or block of data where CRC is to be sent next, the Reset Transmit IP command should be issued in lieu of data.

External/Status Interrupts. There are several sources of External/Status interrupts, each of which may be individually enabled in WR15. The master enable for the External/Status interrupts is located in WR1 (D0). The individual enable bits in WR15 control whether or not latches will be present in the path from the source of interrupt to the status bit in RR0. If an individual enable bit in WR15 is set to "0" the latches are not present in the signal path and the value read in RR0 reflects the current status. An interrupt source whose individual enable in WR15

I/O PROGRAMMING CAPABILITIES (cont'd)

is "0" is not a source of External/Status interrupts even though the External/Status Interrupt Enable bit is set. When an individual enable in WR15 is set to "1", the latch is present in the signal path. The latches for the sources of External/Status interrupts are not independent. Rather, they all close at the same time as a result of a state change by one of the sources of interrupt. Thus, a read of RR0 returns the current status for any bits whose individual enable is "0" and either the current state or the latched state of the remainder of the bits. To guarantee the current status the processor should issue a Reset External/Status Interrupts command in WR0 to open the latches. The External/Status IP is set by the closing of the latches and remains set as long as they are closed. If the master enable for the External/Status interrupts is not set, the IP will never be set, even though the latches may be present in the signal paths and working as described. Because the latches close on the current status but give no indication of change, the processor must maintain a copy of RR0 in memory. When the SCC generates an External/Status interrupt the processor should read RR0 and determine which condition changed state and take appropriate action. The copy of RR0 in memory must then updated and the Reset External/Status Interrupt command issued. Care must be taken in writing the interrupt service routine for the External/Status interrupts because it is possible for more than one status condition to change state at the same time. All of the latch bits in RR0 should be compared to the copy of RR0 in memory. If none have changed and the ZC interrupt is enabled, the Zero Count condition caused the interrupt.

The operation of the individual enable bits in WR15 for each of the six sources of External/Status interrupts is identical, but subtle differences exist in the operation of each source of interrupt. The six sources are Break/Abort, Underrun/EOM, CTS, DCD, Sync/Hunt and Zero Count. The Break/Abort, Underrun/EOM, and Zero Count conditions are internal to the SCC, while Sync/Hunt may be internal or external, and CTS and DCD are purely external signals. In the following discussions each source is assumed to be enabled, so that the latches are present, and the External/Status interrupts are enabled as a whole. Recall that the External/Status IP is set while the latches are closed and that the state of the signal is reflected immediately in RR0 if the latches are not present.

The Break/Abort status is used in asynchronous and SDLC modes but is always "0" in synchronous modes other than SDLC. In asynchronous modes this bit is set when a break sequence (null character plus framing error) is detected in the receive data stream, and remains set as long as "0s" continue to be received. This bit is reset when a "1" is received. A single null character is left in the receive FIFO each time that the break condition is terminated. This character should be read and discarded. In SDLC mode this bit is set by the detection of an abort sequence, which is seven or more contiguous "1s" in the receive data stream. The bit is reset when a "0" is received. A received abort forces the receiver into Hunt, which is also an external/status condition. Though these two bits change state at roughly the same time, one or two External/Status interrupts may be generated as a result. The Break/Abort bit is unique in that both transitions are guaranteed to cause the latches to close, even if another External/Status interrupt is pending at the time these transitions occur. This guarantees that a break or abort will be caught.

The Transmit Underrun/EOM bit is used in synchronous modes to control the transmission of CRC. This bit is reset by issuing the Reset Transmit Underrun/EOM command in WR0. However, this transition does not cause the latches to close ; this occurs only when the bit is set. To inform the processor of this fact, the SCC sets this bit when CRC is loaded into the Transmit Shift register. This bit will also be set if the processor issues the Send Abort command in WR0. The bit is always set in Asynchronous mode.

The CTS bit reports the state of the $\overline{\text{CTS}}$ input, and the DCD bit reports the status of the DCD input. Both bits latch on either input transition. In both cases, after the Reset External/Status Interrupt command is issued, if the latches are closed, they remain closed if there is any odd number of transitions on an input ; they will be open if there is an even number of transitions on the input.

The Zero Count bit is set when the counter in the baud rate generator reaches a count of "0" and is reset when the counter is reloaded. The latches are closed only when this bit is set to "1", and the status in RR0 always reflects the current status. While the Zero Count IE bit in WR15 is reset this bit is forced to "0".

There are a variety of ways in which the Sync/Hunt may be set and reset, depending on the SCC's mode operation. In Asynchronous mode this bit reports the state of the $\overline{\text{SYNC}}$ pin, latching on both input transitions. The same is true of External Sync mode. However, if the crystal oscillator is enabled while in Asynchronous mode this bit will be forced

I/O PROGRAMMING CAPABILITIES (cont'd)

to "0" and the latches will not be closed. Selecting the crystal option in External Sync mode is illegal, but the result will be the same. In Synchronous modes other than SDLC the Sync/Hunt reports the Hunt state of the receiver. Hunt mode is entered when the processor issues the Enter Hunt command in WR3. This forces the receiver to search for a sync character match in the receive data stream. Because both transitions of the Hunt bit close the latches, issuing this command will cause an External/Status interrupt. The SCC resets this bit when character synchronization has been achieved, causing the latches to again be closed. In these synchronous modes the SCC will not reenter the Hunt mode automatically ; only the Enter Hunt command will set this bit. In SDLC mode this bit is also set by the Enter Hunt command, but the receiver will also automatically enter the Hunt mode if an Abort sequence is received. The receiver leaves Hunt upon receipt of a flag sequence. Both transitions of the Hunt bit will cause the latches to be closed. In SDLC mode the receiver will automatically synchronize on Flag characters. The receiver is in Hunt mode when it is enabled, so the Enter Hunt command will probably never be needed.

If careful attention is paid to details, the interrupt service routine for External/Status interrupts is straightforward. To determine which bit or bits changed state, the routine should first read RR0 and compare it to a copy from memory. For each changed bit the appropriate action should be taken and the copy in memory updated. The service routine should close with a Reset External/Status interrupts command to reopen the latches. The copy of RR0 in memory should always have the Zero Count bit set to "0", since this will be the state of the bit after the Reset External/Status interrupts command at the end of the service routine. When the processor issues the Reset Transmit Underrun/EOM latch command in WR0, the Transmit Underrun/EOM bit in the copy of RR0 in memory should be reset because this transition does not cause an interrupt.

Block Transfers

The SCC offers several alternatives for the block transfer of data. The various options are selected by WR1 (bits D7 through D5) and WR14 (bit D2). Each channel in the SCC has two pins which may be used to control the block transfer data. Both pins in each channel may be programmed to act as DMA Request signals, and one pin in each channel may be programmed to act as a Wait signal for the CPU. In either mode, it is advisable to select and enable the mode in two separate accesses of the appropriate register. The first access should select the mode and the second access should enable the function. This procedure prevents glitches on the output pins. Reset forces Wait mode, with W/REQ open-drain.

Wait on Transmit. The Wait function on transmit is selected by setting both D6 and D5 to "0" and then enabling the function by setting D7 of WR1 to "1". In this mode the W/REQ pin carries the WAIT signal, and is open-drain when inactive and Low when active. When the processor attempts to write to the transmit buffer when it is full, the SCC will assert WAIT until the buffer is empty. This allows the use of a block-move instruction to transfer the transmit data. In the MK85C30, WAIT will go active in response to WR going active, but only if the data buffer is being accessed, either directly or via the pointers. The WAIT pin is released in response to the falling edge of PCLK. Details of the timing are shown in figure 18.

Wait on Receive. The Wait function on receive is selected by setting D6 or WR1 to "0", D5 of WR1 to "1", and then enabling the function by settling D7 of WR1 to "1". In this mode the W/REQ pin carries the WAIT signal, and is open-drain when inactive and Low when active. When the processor attempts to read data from the receive FIFO when it is empty, the SCC will assert WAIT until a character has reached the top of the FIFO. This allows the use of a block-move instruction to transfer the receive data. In the MK85C30, WAIT will go active in response to RD going active, but only if the receive data FIFO is being accessed, either directly or via the pointers. The WAIT pin is released in response to the falling edge of PCLK. Details of the timing are shown in figure 19.

DMA Requests. The two DMA request pins W/REQ and DTR/REQ can be programmed to be used as DMA requests. The W/REQ pin can be used as either a transmit or a receive request and the DTR/REQ pin can only be used as a receive request. For full-duplex operation, the W/REQ is, therefore, used for transmit and the DTR/REQ is used for receive. These modes are described below.

DMA Request on Transmit (using W/REQ). The Request on Transmit function is selected by setting D6 of WR to "1", D5 of WR1 to "0", and then enabling the function by setting D7 of WR1 to "1". In this mode the W/REQ pin carries the REQUEST signal, which is active Low. When this mode is selected, but not yet enabled, the W/REQ is driven High. When the enable bit is set, REQUEST goes Low if the transmit buffer is empty at the time, or will remain High until the transmit buffer becomes empty. Note that the REQUEST pin will follow the state of the transmit buffer even though the transmitter is

I/O PROGRAMMING CAPABILITIES (cont'd)

disabled. Thus, if the REQUEST is enabled, the DMA may write data to the SCC before the transmitter is enabled. This will not cause a problem in Asynchronous mode but may cause problems in Synchronous mode because the SCC will send data in preference to flags or sync characters. It may also complicate the CRC initialization, which cannot be done until after the transmitter is enabled. With only one exception, the REQUEST pin directly follows the state of the transmit buffer in this mode. REQUEST goes Low when the transmit buffer empties and remains Low until the transmit buffer is filled. The SCC generates only one falling edge on REQUEST per character requested and the timing for this is shown in figure 20. The one exception occurs in synchronous modes at the end of CRC transmission. At the end of CRC transmission, when the closing flag or sync character is loaded into the Transmit Shift register, REQUEST is pulsed High for one PCLK cycle. The DMA may use this falling edge on REQUEST to write the first character of the next frame or block to the SCC.

In the MK85C30, REQUEST will go High in response to the falling edge of WR, but only when the appropriate transmit buffer in the SCC is accessed. This is shown in figure 21.

DMA Request on Transmit (using DIR/REQ). A second Request on Transmit function is available on the DTR/REQ pin. This mode is selected by setting D2 of WR14 to "1". When this bit is set to "1", REQUEST goes Low if the transmit buffer is empty at the time, or will go High until the transmit buffer becomes empty. While D2 of WR14 is set to "0", the DTR/REQ pin is DTR and follows the inverted state of D7 in WR5. This pin will be High after a channel or hardware reset and in the DTR mode. In the Request mode REQUEST will follow the state of the transmit buffer even though the transmitter is disabled. Thus if REQUEST is enabled before the transmitter is enabled, the DMA may write data to the SCC before the transmitter is enabled. This will not cause a problem in Asynchronous mode, but may cause problems in Synchronous mode because the SCC will send data in preference to flags or sync characters. It may also complicate the CRC initialization, which cannot be done until after the transmitter is enabled. With only one exception, the REQUEST pin directly follows the state of the transmit buffer in this mode. REQUEST goes Low when the transmit buffer empties and remains Low until the transmit buffer is filled. The SCC generates only one falling edge on REQUEST per character requested. The one exception occurs in synchronous modes at the end of CRC transmission. At the end of CRC transmission, when the closing flag or sync character is loaded into the Transmit Shift register, REQUEST is pulsed High for one PCLK cycle. The DMA may use this falling edge on REQUEST to write the first character of the next frame or block to the SCC. The Request signal on DTR/REQ differs from the one on W/REQ in that it does not go immediately High in response to the access which writes to the transmit buffer. This is because the registers in the SCC are not written during the actual access, but are delayed by some number of PCLK cycles. The Request signal on DTR/REQ follows the state of the transmit buffer exactly while the Request signal on W/REQ goes inactive in anticipation of the transmit buffer becoming full. The timing of the Request signal on both pins is shown in figure 21.

DMA Request on Receive. The Request on Receive function is selected by setting D6 and D5 of WR1 to "1" and then enabling the function by setting D7 of WR1 to "1". In this mode the W/REQ pin carries the REQUEST signal, which is active Low. When this mode is selected, but not yet enabled, the W/REQ pin is driven High. When the enable bit is set REQUEST goes Low if the receive buffer contains a character at the time, or will remain High until a character enters the receive buffer. Note that the REQUEST pin will follow the state of the receive buffer even though the receiver is disabled. Thus, if the receiver is disabled and REQUEST is still enabled, the DMA will transfer the previously received data correctly. In this mode the REQUEST pin directly follows the state of the receive buffer with only one exception. REQUEST goes Low when a character enters the receive buffer and remains Low until this character is removed from the receive buffer. The SCC generates only one falling edge on REQUEST per character transfer requested and the timing for this is shown in figure 22. The one exception occurs in the case of a special receive condition in the Receive Interrupt on First Character or Special Condition mode, or the Receive Interrupt on Special Condition Only mode. In the two interrupt modes any receive character with a special receive condition is locked at the top of the FIFO until an Error Reset command is issued. This character in the receive FIFO would ordinarily cause additional DMA Requests after the first time it is read. However, the logic in the SCC guarantees only one falling edge on REQUEST by holding REQUEST High from the time the character with the special receive condition is read, and the FIFO locked, until after the Error Reset command has been issued. Once the FIFO is unlocked by the Error Reset command, REQUEST again follows the state of the receive buffer. In the MK85C30, REQUEST will go High in response to the falling edge of RD, but only when the appropriate receive buffer in the SCC is accessed. This is shown in figure 23.

I/O PROGRAMMING CAPABILITIES (cont'd)

Figure 18 : Wait on Transmit.

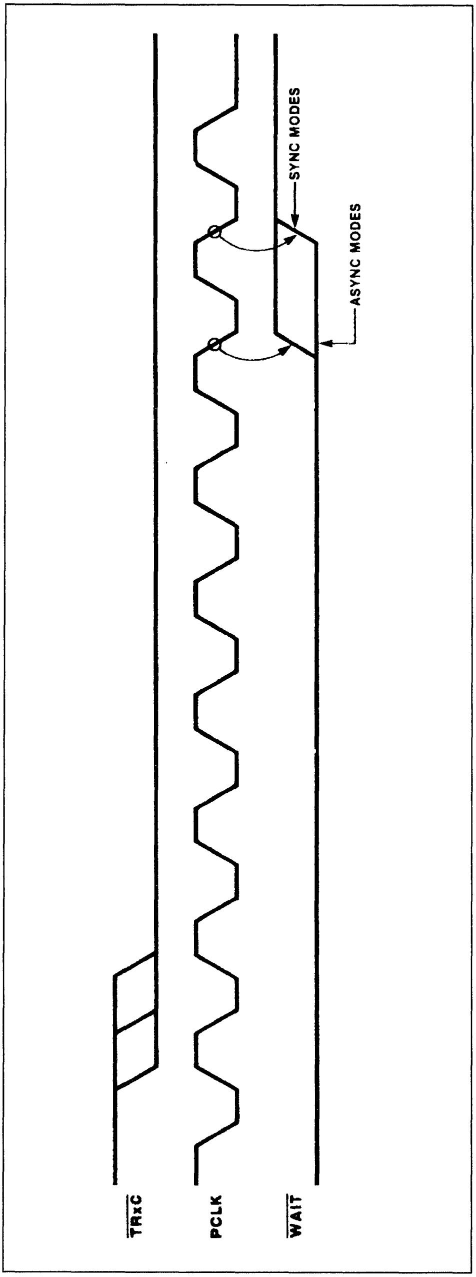
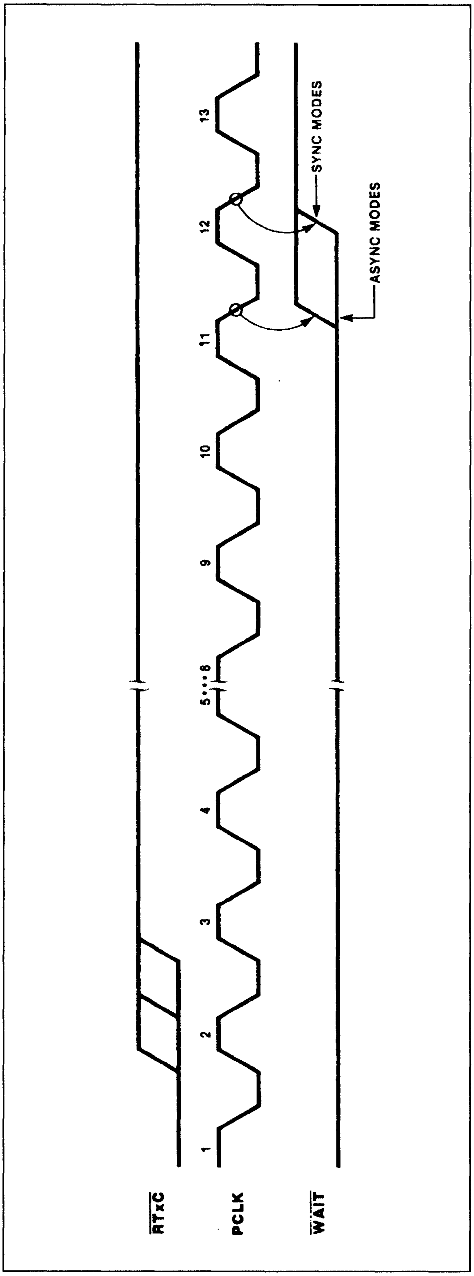


Figure 19 : Wait on Receive.



I/O PROGRAMMING CAPABILITIES (cont'd)

Figure 20 : Transmit Request Assertion.

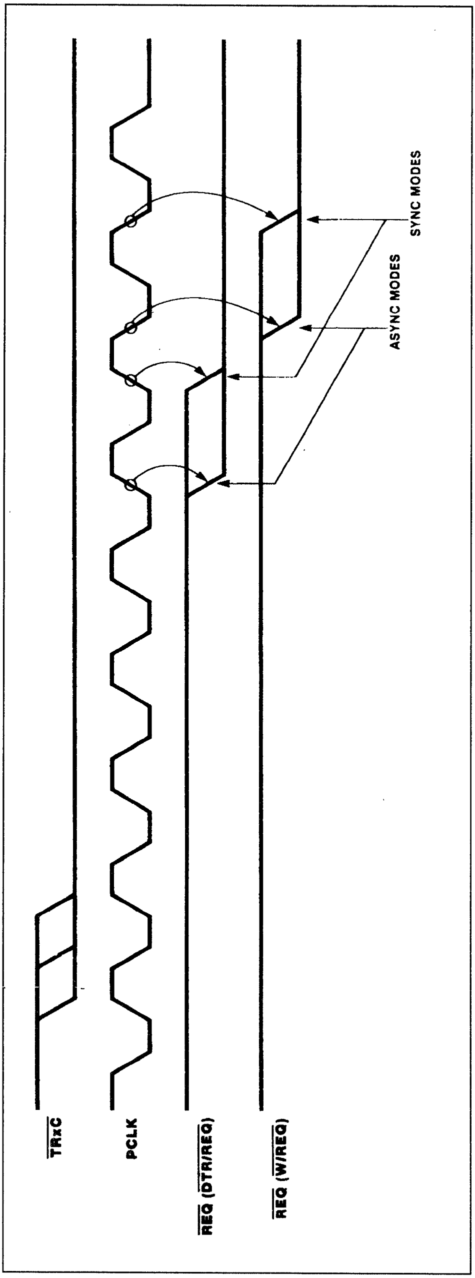
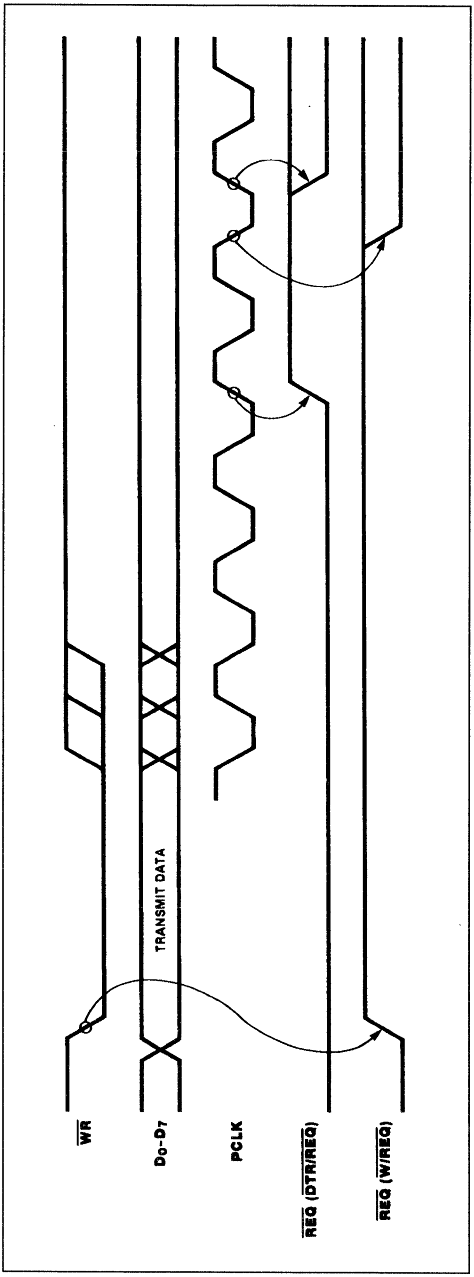

SYNC MODES
ASYNC MODES
$\overline{\text{TRxC}}$
PCLK
$\overline{\text{REQ}}$ (DTR/REQ)
$\overline{\text{REQ}}$ (W/REQ)

Figure 21 : Transmit Request Release.


TRANSMIT DATA
$\overline{\text{WR}}$
D_0-D_7
PCLK
$\overline{\text{REQ}}$ (DTR/REQ)
$\overline{\text{REQ}}$ (W/REQ)

I/O PROGRAMMING CAPABILITIES (cont'd)

Figure 22 : Receive Request Assertion.

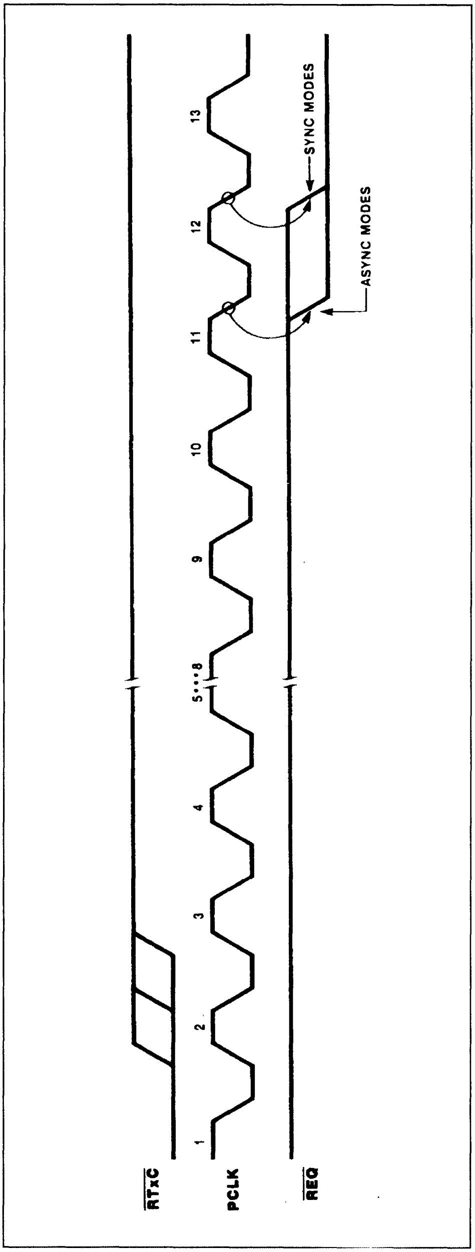
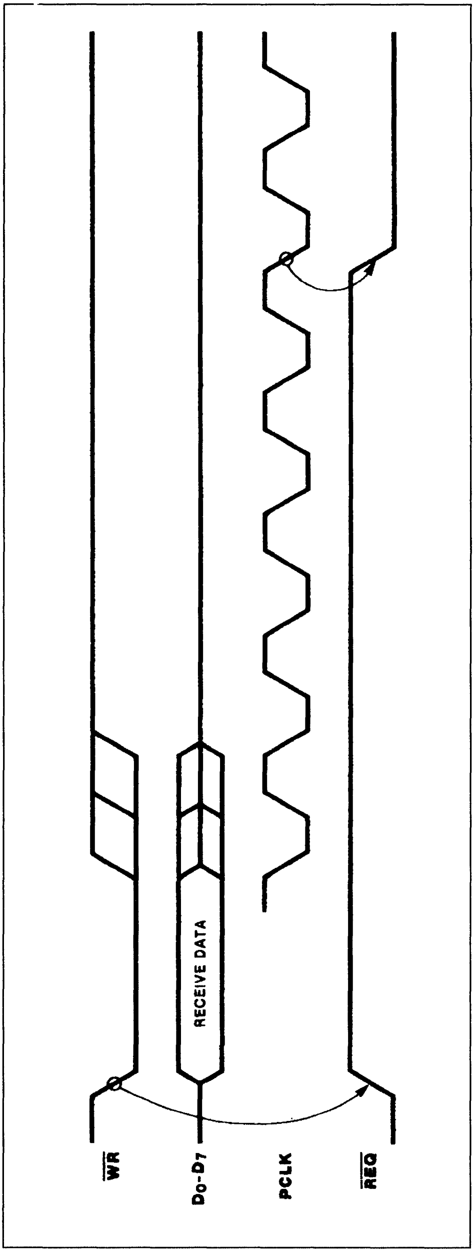


Figure 23 : MK85C30 Receive Request Release.



PROGRAMMING DATA COMMUNICATION MODES

The SCC provides two independent full-duplex channels programmable for use in any common asynchronous or synchronous data communication protocol. These include asynchronous, synchronous byte-oriented protocols, monosync, IBM Bisync, and bit-oriented protocols such as HDLC and SDLC. This chapter is divided into 3 parts : Asynchronous, Synchronous, and SDLC.

Asynchronous Mode

The SCC supports Asynchronous mode with a number of programmable options including the number of bits per character, the number of stop bits, the clock factor, modem interface signals and break detect and generation. Asynchronous mode is selected by programming the desired number of stop bits in D_3 and D_2 of WR4. Programming these two bits with other than "00" places both the receiver and transmitter in Asynchronous mode. In this mode, the SCC ignores the state of bits D_4, D_3, D_2, and D_1 of WR3, bits D_5 and D_4 of WR4, bits D_2 and D_0 of WR5, all of WR6 and WR7 and all of WR10 except D_6 and D_5. Bits that are ignored may be programmed with "1" or "0" or not at all.

Asynchronous Receive. Asynchronous mode is selected by specifying the number of stop bits per character in WR4. This selection applies only the transmitter, however, as the receiver always checks for one stop bit. If after character assembly the receiver finds this stop bit to be a "0", the Framing Error bit in the receive error FIFO is set at the same time that the character is transferred to the receive data FIFO. This error bit accompanies the data to the top of the FIFO, where it generates a special receive condition. The Framing Error bit is not latched, and so must be read in RR1 before the accompanying data is read.

The number of bits per character is controlled by bits D7 and D6 of WR3. Five, six, seven, or eight bits per character may be selected via these two bits. Data is right-justified with the unused bits set to "1s". An additional bit, carrying parity information may be selected by setting bit D0 of WR4 to "1". Note that this also enables parity for the transmitter. The parity sense is selected by bit D1 of WR4. If this bit is set to "1", the received character is checked for even parity, if set to "0", the received character is checked for odd parity. The additional parity bit per character is transferred to the receive data FIFO along with the data if the data plus parity is eight bits or less. The parity Error bit in the receive error FIFO may be programmed to cause a special receive condition interrupt by setting bit D2 of WR1 to "1". This error bit is latched and so will remain active, once set, until an Error Reset command has been issued. If interrupts are not used to transfer data, the Parity Error, Framing Error, and Overrun Error bits in RR1 should be checked before the data is removed from the receive data FIFO.

The break condition is continuous "0s", as opposed to the usual continuous ones during an idle. The SCC recognizes the Break condition upon seeing a null character (all "0s") plus a framing error. Upon recognizing this sequence the Break bit in RR0 will be set and will remain set until a "1" is received. At this point the break condition is no longer present. At the termination of a break the receive data FIFO contains a single null character, which should be read and discarded. The Framing Error bit will not be set for this character, but if odd parity has been selected, the Parity Error bit will be set. Caution should be exercised if the receive data line contains a switch that is not debounced to generate breaks. Switch bounce may cause multiple breaks, recognized by the SCC to be additional characters assembled in the receive data FIFO. It may also cause a receive overrun condition being latched.

The SCC may be programmed to accept a receive clock that is one, sixteen, thirty-two, or sixty-four times the data rate. This is selected by bits D7 and D6 in WR4. The 1X mode is used when bits are synchronized external to the receiver. The 1X mode is the only mode in which a data encoding method other than NRZ may be used. The clock factor is common to the receiver and transmitter.

The SCC provides up to three modem control signals associated with the receiver. The $\overline{\text{SYNC}}$ pin is a general-purpose input whose state is reported in the Sync/Hunt bit in RR0. If the crystal oscillator is enabled, this pin is not available and the Sync/Hunt bit is forced to "0". Otherwise, the SYNC pin may be used to carry the Ring Indicator signal. The $\overline{\text{DTR/REQ}}$ pin carries the inverted state of the DTR bit (D7) in WR5 unless this pin has been programmed to carry a DMA Request signal. The $\overline{\text{DCD}}$ pin is ordinarily a simple input to the $\overline{\text{DCD}}$ bit in RR0. However, if the Auto Enables mode is selected by setting D5 of WR3 to "1", this pin becomes an enable for the receiver. That is, if Auto Enables is on and the $\overline{\text{DCD}}$ pin is HIGH, the receiver is disabled. While the $\overline{\text{DCD}}$ pin is LOW, the receiver is enabled.

The initialization sequence for the receiver in Asynchronous mode is : WR4 first to select the mode, then WR3 and WR5 to select the various options. At this

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

point, the other registers should be initialized as necessary. When all of this is complete the receiver may be enabled by setting bit D_0 or WR3 to "1".

Asynchronous Transmit. Asynchronous mode is selected by specifying the number of stop bits per character in bits D_3 and D_2 of WR4. The three options available are one, one-and-a-half, or two stop bits per character. These two bits only select the number of stop bits for the transmitter, as the receiver always checks for one stop bit.

The number of bits per transmitted character is controlled both by Bits D_6 and D_5 in WR5 and the way the data is formatted within the transmit buffer. The bits in WR5 allow the option of five, six seven, or eight bits per character. When five bits per character is selected the data may be formatted before being written to the transmit buffer to allow transmission of from one to five bits per character.

This formatting is shown in table 5.

Table 5 : Data Format - Five Bits or Less.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	1	1	1	0	0	0	0	One data bit
1	1	1	0	0	0	0	0	Two data bits
1	1	0	0	0	0	0	0	Three data bits
1	0	0	0	0	0	0	0	Four data bits
0	0	0	0	0	0	0	0	Five data bits

In all cases the data must be right-justified, with the unused bits being ignored except in the case of five bits per character. An additional bit, carrying parity information, may be automatically appended to every transmitted character by setting bit D0 of WR4 to "1". This bit is sent in addition to the number of bits specified in WR4 or by the data format. The parity sense is selected by bit D1 of WR4. If this bit is set to "1", the transmitter sends even parity, if set to "0", the parity is odd.

The transmitter may be programmed to send a Break by setting bit D4 of WR5 to "1". The transmitter will send continuous "0s" from the first transmit clock edge after this command is issued, until the first transmit clock edge after this bit is reset. The transmit clock edges referred to here are those that define transmitted bit cell boundaries.

An additional status bit for use in Asynchronous mode is available in bit D_0 or RR1. This bit, called All Sent, is set when the transmitter is completely empty and any previous data or stop bits have reached the TxD pin. The All Sent bit can be used by the processor as an indication that the transmitter may be safely disabled.

The SCC may be programmed to accept a transmit clock that is one, sixteen, thirty-two, or sixty-four time the data rate. This is selected by bits D_7 and D_6 of WR4, in common with the clock factor for the receiver. Note that the chosen clock factor may restrict the number of stop bits that may be transmitted. In particular, when the clock rate and data rate are identical, one-and-a-half stop bits are not allowed. If any length other than one stop bit is desired in the times one mode, only two stop bits may be used.

There are two modem control signals associated with the transmitter provided by the SCC. The $\overline{\text{RTS}}$ pin is a simple output that carries the inverted state of the RTS bit (D_1) in WR5, unless the Auto Enables bit (D_5) is set in WR3. When Auto Enables is set the RTS pin will immediately go LOW when the RTS bit is set. However, when the RTS bit is reset the $\overline{\text{RTS}}$ pin remains LOW until the transmitter is completely empty and the last stop bit has left the TxD pin. Thus the $\overline{\text{RTS}}$ pin may be used to disable external drivers for the transmit data. The $\overline{\text{CTS}}$ pin is ordinarily a simple input to the CTS bit in RR0. However, if Auto Enables mode is selected this pin becomes an enable for the transmitter. That is, if Auto Enables is on and the $\overline{\text{CTS}}$ pin is HIGH, the transmitter is disabled ; the transmitter is enabled while the $\overline{\text{CTS}}$ pin is LOW.

The initialization sequence for the transmitter in Asynchronous mode is : WR4 first to select the mode, then WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is complete, the transmitter may be enabled by setting bit D3 of WR5 to "1". Note that the transmitter and receiver may be initialized at the same time.

Synchronous Mode

In synchronous modes of operation a special bit pattern is used to provide character synchronization. The SCC offers several options to support Synchronous mode including various sync character lengths, the number of bits per data character, parity generation and checking, CRC generation and checking, as well as modem controls and a transmitter to receiver synchronization function. Synchronous mode is selected by programming bits D_3 and D_2 of WR4 with "Os". This selects Synchronous mode, as opposed to Asynchronous mode, but this selection is further modified by bits D_5 and D_7 of WR4 as well as bits D_1 and D_0 of WR10.

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

The sync character or characters are written in WR6 and WR7. In all synchronous modes, except External Sync the state of bits D7 and D6 of WR4 are forced to "0" to select the times one clock mode. In External Sync mode these two bits must be programmed with "0s".

Synchronous Receive. The receiver in the SCC searches for character synchronization only while it is in Hunt mode. In this mode the receiver is idle except that it is searching the incoming data stream for a sync character match. The receiver is in Hunt mode when it is first enabled, and may be placed in Hunt mode by command from the processor. This is accomplished by issuing the Enter Hunt Mode command in WR3. This bit (D_4) is a command ; writing a "0" to it has no effect. The Hunt status of the receiver is reported by the Sync/Hunt bit in RR0. Sync/Hunt is one of the possible sources of external/status interrupts, with both transitions causing an interrupt. This is true even if the Sync/Hunt bit is set as a result of the processor issuing the Enter Hunt Mode command.

An 8-bit sync character is selected by setting bits D_5 and D_4 of WR4, as well as bit D0 of WR10, to "0". With this option the receiver searches the data stream for a match will the eight bits in WR7. The 6-bit sync option requires the same programming except that D0 of WR10 is set to "1" and the sync character is held in the high-order six bits of WR7. The SCC also allows the option of double length sync characters. This is selected by setting bit D_5 of WR4 to "0" and bit D_4 of WR4 to "1". The selection between 12 and 16 bits of sync character is controlled by bit D0 of WR10. A "0" selects 16 bits of sync character, while a "1" in this bit selects a 12-bit sync character. The arrangement of the sync character in WR6 and WR7 is shown in figure 24. For those applications requiring any other sync character length, the SCC makes provision for an external circuit to provide a character synchronization signal on the $\overline{\text{SYNC}}$ pin. This mode is selected by setting bit D_5 and D_4 of WR4 to "1". In this mode the Sync/Hunt bit in RR0 reports the state of the $\overline{\text{SYNC}}$ pin but the receiver must still be placed in Hunt mode when the external logic is searching for a sync character match. When the receiver is in Hunt mode and the $\overline{\text{SYNC}}$ pin is driven LOW, two receive clock cycles after the last bit of the sync character is received, character assembly will begin on the rising edge of the receive clock immediately preceding the activation of $\overline{\text{SYNC}}$. This is shown in figure 25. The receiver leaves Hunt mode when $\overline{\text{SYNC}}$ is driven LOW. In all cases except External Sync mode the $\overline{\text{SYNC}}$ pin is an output that is driven LOW by the SCC to signal that a sync character has been received. The $\overline{\text{SYNC}}$ pin is activated regardless of character boundaries so any external circuitry using it should only respond the $\overline{\text{SYNC}}$ pulse that occurs while the receiver is in Hunt mode. The timing for the $\overline{\text{SYNC}}$ signal is shown in figure 26.

The number of bits per character is controlled by bits D_7 and D_6 of WR3. Five, six, seven, or eight bits per character may be selected via these two bits. The data is right-justified in the receive data buffer. The SCC merely takes a snapshot of the receive data stream at the appropriate times so the "unused" bits in the receive buffer are only the bits following the character in the data stream. An additional bit, carrying parity information, may be selected by setting bit D_0 of WR4 to "1". If this bit is set to "1", the received character is checked for even parity, if set to "0", the received character is checked for odd parity. The additional bit per character is visible in the receive data FIFO if the data plus parity is eight bits or less. The parity bit is not visible when there are eight data bits per character. The Parity Error bit in the receive error FIFO may be programmed to cause a Special Receive Condition interrupt by setting bit D_2 of WR1 to "1". This error bit is latched and so will remain active, once set, until an Error Reset command has been issued. If interrupts are not used to transfer data the Parity Error, CRC Error, and Overrun Error bits in RR1 should be checked before the data is removed from the receive data FIFO. The character length may be changed at any time before the new number of bits has been assembled by the receiver, but, care should be exercised as unexpected results may occur. A representative example, switching from five bits to eight bits and back to five bits is shown in figure 27. It is sometimes desirable to prevent sync characters in the receive data stream from being transferred to the receive data FIFO. This function is available in the SCC by setting the Sync Character Load Inhibit bit (D1) in WR3 to "1". While this bit is set to "1", character about to be loaded into the receive data FIFO is compared with the contents of WR6. If all eight bits match the character, it is not loaded into the receive data FIFO. Because the comparison is across eight bits, this function works correctly only when the number of bits per character is the same as the sync character length. Thus it cannot be used with 12- or 16-bit sync characters. Both leading sync characters and sync characters embedded in the data will be properly removed in the case of an 8-bit sync character, but only the leading sync characters may be properly removed in the case of a 6-bit sync character. Care must be exercised in using this fea-

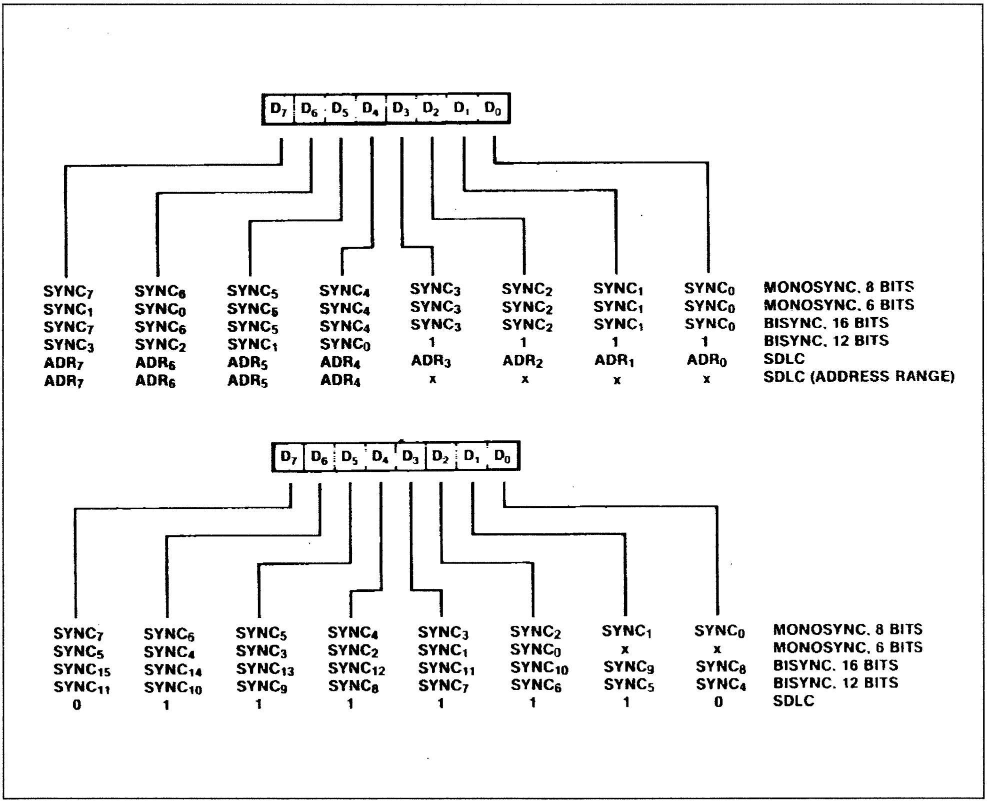
PROGRAMMING DATA COMMUNICATION MODES (cont'd)

ture because sync characters not transferred to the receive data FIFO will automatically be excluded from CRC calculation. This works properly only in the 8-bit case.

Either of two CRC polynomials may be used in synchronous modes, selected by bit D$_2$ in WR5. If this bit is set to "1", the CRC-16 polynomial is used, if this bit is set to "0", the CRC-CCITT polynomial is used. This bit controls the polynomial selection for both the receiver and transmitter. The initial state of the generator and checker is controlled by bit D$_7$ of WR10. When this bit is set to "1", both the generator and checker will have an initial value of all ones, if this bit is set to "0", the initial values will be all "0s". The SCC presets the checker whenever the receiver is in Hunt mode so a CRC reset command is not strictly necessary. However, the CRC checker may be preset by issuing the Reset CRC Checker command in WR0. This command is encoded in bits D$_7$ and D$_6$ of WR0. If CRC is to be used the CRC checker must be enabled by setting bit D0 of WR3 to "1". If sync characters are being stripped from the data stream, this may be done at any time before the first non-sync character is received. If the sync strip feature is not being used, CRC must not be enabled until after the first data character has been transferred to the receive data FIFO. As previously mentioned, 8-bit sync characters stripped from the data stream are automatically excluded from CRC calculation.

Some synchronous protocols require that certain characters be excluded from CRC calculation. This is possible in the SCC because CRC calculation may be enabled and disabled on the fly. To give the processor sufficient time to decide whether or not a particular character should be included in the CRC calculation, the SCC contains an 8-bit time delay between the receive shift register and the CRC

Figure 24 : Sync Character Programming.



D$_7$ D$_6$ D$_5$ D$_4$ D$_3$ D$_2$ D$_1$ D$_0$

SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 8 BITS
SYNC$_1$	SYNC$_0$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 6 BITS
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	BISYNC, 16 BITS
SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	1	1	1	1	BISYNC, 12 BITS
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	ADR$_3$	ADR$_2$	ADR$_1$	ADR$_0$	SDLC
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	x	x	x	x	SDLC (ADDRESS RANGE)

D$_7$ D$_6$ D$_5$ D$_4$ D$_3$ D$_2$ D$_1$ D$_0$

SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 8 BITS
SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	x	x	MONOSYNC, 6 BITS
SYNC$_{15}$	SYNC$_{14}$	SYNC$_{13}$	SYNC$_{12}$	SYNC$_{11}$	SYNC$_{10}$	SYNC$_9$	SYNC$_8$	BISYNC, 16 BITS
SYNC$_{11}$	SYNC$_{10}$	SYNC$_9$	SYNC$_8$	SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	BISYNC, 12 BITS
0	1	1	1	1	1	1	0	SDLC

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

Figure 25 : SYNC as an Output.

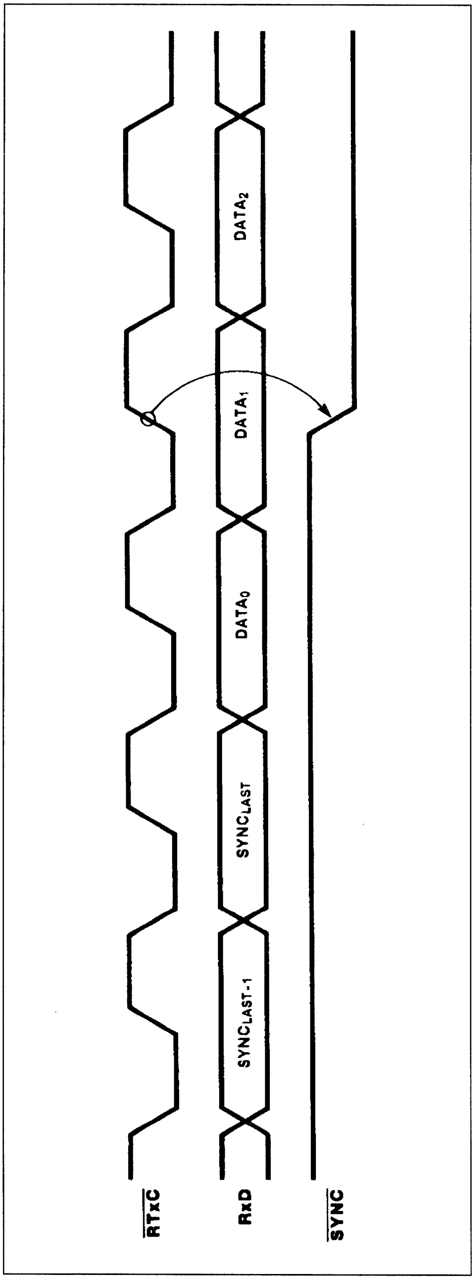
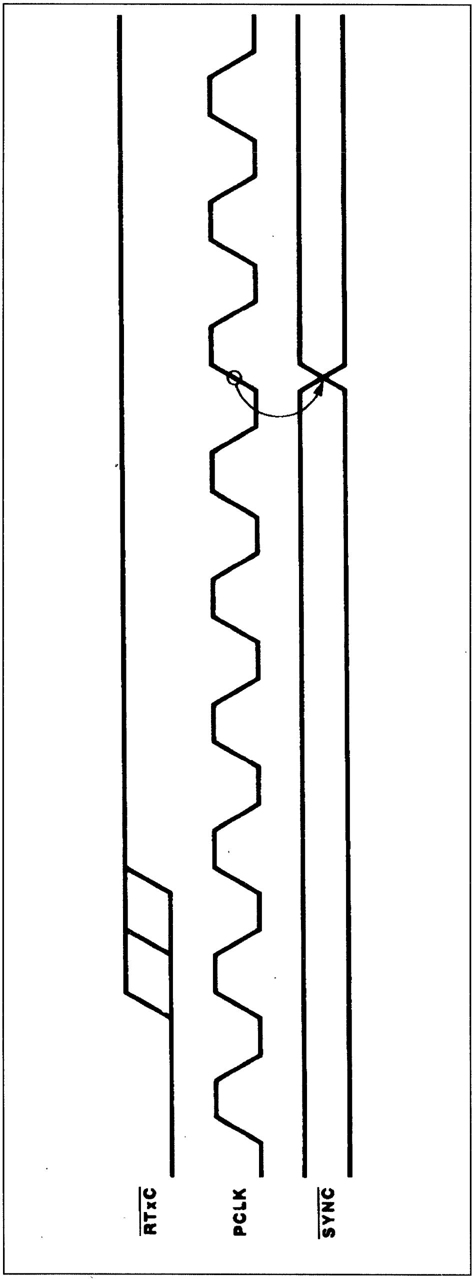


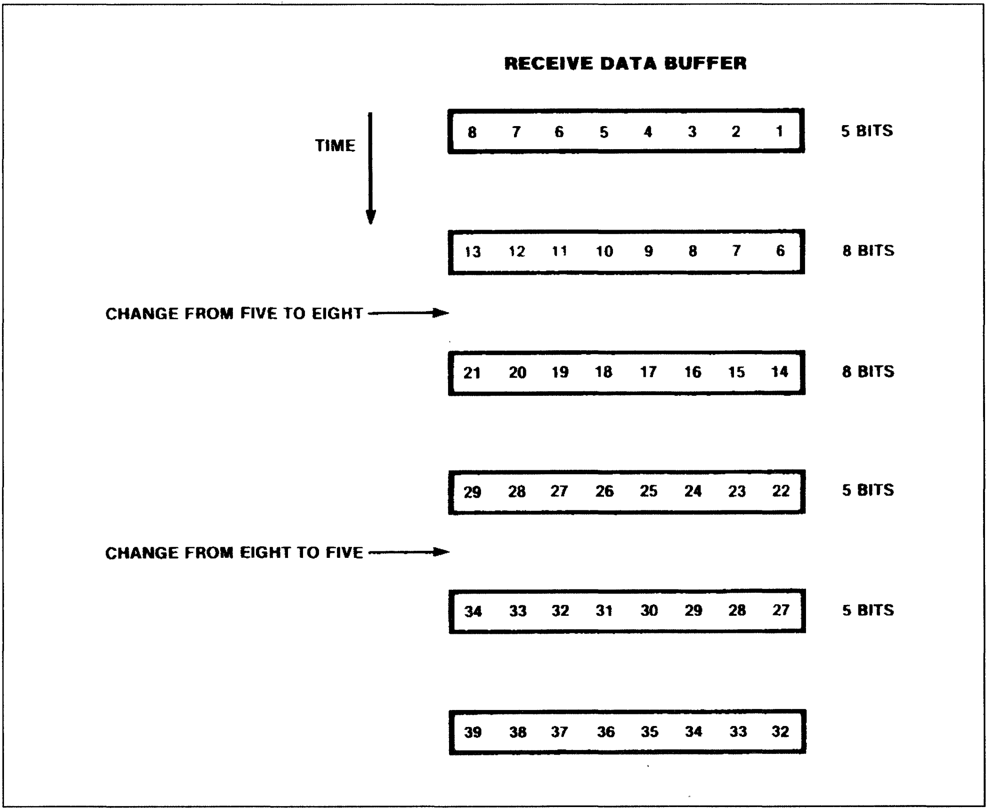
Figure 26 : SYNC as an Input.



PROGRAMMING DATA COMMUNICATION MODES (cont'd)

checker. The logic also guarantees that the calculation will only start or stop on a character boundary by delaying the enable or disable until the next character is loaded into the receive data FIFO. To understand how this works refer to figure 28 and the following explanation. Consider a case where the SCC receives a sequence of eight bytes, called A, B, C, D, E, F, G and H with A receiver first. Now suppose that A is the sync character, that CRC is to be calculated on B, C, E, and F, and that F is the last byte of this message. Before A is received the receiver is in Hunt mode and the CRC is disabled. When A is in the receive shift register it is compared with the contents of WR7. Since A is the sync character, the bit patterns match and receiver leaves Hunt mode, but character A is not transferred to the receive data FIFO. The CRC remains disabled even though somewhere during the next eight-bit-time processor reads B and enables CRC. At the end of an eight-bit-time, B is in the 8-bit delay and C is in the receive shift register. At this point, B is loaded into the receive data FIFO. The CRC remains disabled even though somewhere during the next eight bit times the processor reads B and enables CRC. At the end of the eight-bit-time, B is in the 8-bit delay and C is in the receive shift register. Character C is loaded into the receive data FIFO and at the same time the CRC checker is enabled. During the next eight-bit-time, the processor reads C and leaves the CRC enabled. At the end of these eight-bit-times the SCC has calculated CRC on B, character C is the 8-bit delay and D is in the Receive Shift register. D is then loaded into the receive data buffer and at some point during the next eight-bit-time the processor reads D and disables CRC. At the end of these eight-bit-times CRC has been calculated on C, character D is in the 8-bit delay and E is in the Receive Shift register.

Figure 27 : Changing Character Length.



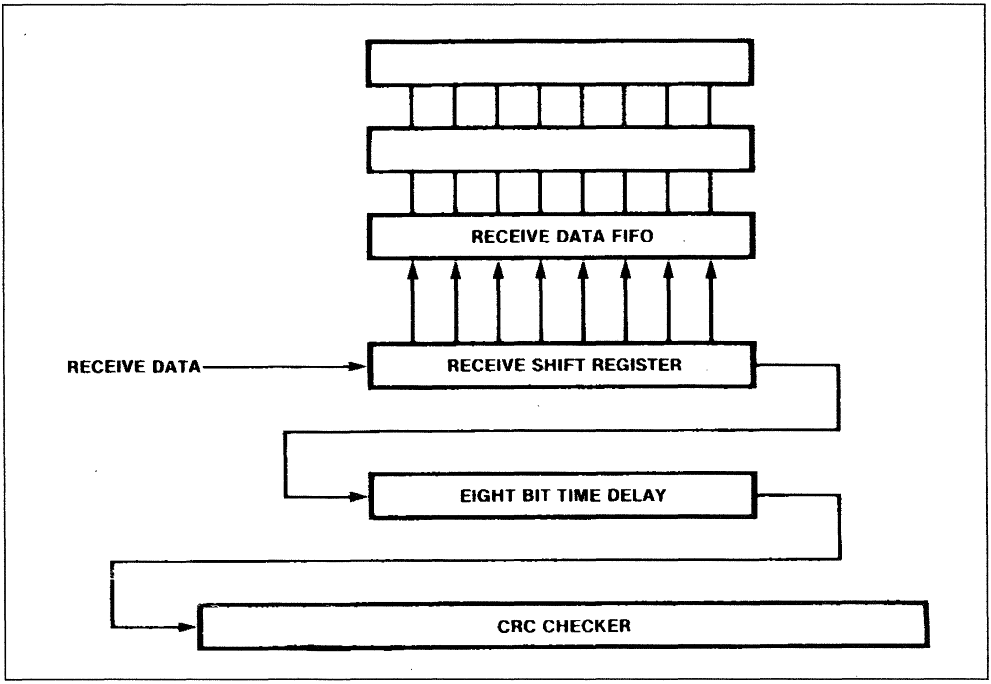
PROGRAMMING DATA COMMUNICATION MODES (cont'd)

Now E is loaded into the receiver data FIFO and, at the same time, the CRC is disabled. During the next eight-bit-times the processor reads E and enables the CRC. During this time E shifts into the 8-bit delay, F enters the Receive Shift register and CRC is not being calculated on D. After these eight-bit-times have elapsed, E is in the 8-bit delay, and F is in the Receive Shift register. Now F is transferred to the receive data FIFO and CRC is enabled. During the next eight-bit-times the processor reads F and leaves the CRC enabled. The processor is usually aware that this is the last character in the message and so prepares to check the result of the CRC computation. However, another sixteen bit-times are required before CRC has been calculated on all of character F. At the end of eight-bit-times F is in the 8-bit delay and G is in the Receive Shift register. At this time G is transferred to the receive data FIFO. Character G must be read and discarded by the processor. Eight bit times later H is transferred to the receive data FIFO also. The result of a CRC calculation is latched in the receive error FIFO at the same time as data is written to the receive data FIFO. Thus the CRC result through character F accompanies character H in the FIFO and will be valid in RR1 until character H is read from the receive data FIFO. The CRC checker may be disabled and reset at any time after character H is transferred to the receive data FIFO. Recall, however, that internally CRC will not be disabled until a character is loaded into the receive data FIFO so the reset command should not be issued until after this occurs. A better alternative is to place the receiver in Hunt mode, which automatically disables and resets the CRC checker.

Up to two modem control signals associated with the receiver are available is synchronous modes. The $\overline{\text{DTR/REQ}}$ pin carries the inverted state of the DTR bit (D7) in WR5 unless this pin has been programmed to carry a DMA Request signal. The $\overline{\text{DCD}}$ pin is ordinarily a simple input to the DCD bit in RR0. However, if the Auto Enables mode is selected by setting D5 of WR3 to "1", this pin becomes an enable for the receiver. That is, if Auto Enables is on and the $\overline{\text{DCD}}$ pin is HIGH the receiver is disabled ; while the $\overline{\text{DCD}}$ pin is LOW the receiver is enabled.

The initialization sequence for the receiver in synchronous modes is WR4 first, to select the mode,

Figure 28 : Receive CRC Data Path.



PROGRAMMING DATA COMMUNICATION MODES (cont'd)

then WR10 to modify it if necessary, WR6 and WR7 to program the sync characters and then WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is complete the receiver is enabled by setting bit D of WR3 to "1".

SYNCHRONOUS TRANSMIT. Once Synchronous mode has been selected, any of three sync character lengths may be selected. An 8-bit sync character is selected by setting bits D_5 and D_4 or WR4, as well as bit D0 of WR10 to "0". With this option selected the transmitter sends the contents of WR6 when it has no data to send. The 6-bit sync option requires the same programming except that bit D0 of WR10 is set to "1" and only the least significant six bits of WR6 and used as a time fill. For a 16-bit sync character, set bit D_4 of WR4 to "1" and D_5 of WR4 and bit D_0 of WR10 to "0". In this mode the transmitter sends the concatenation of WR6 and WR7 as a time fill. Because the receiver requires that sync characters be left-justified in the registers, while the transmitter requires them to be right-justified, only the receiver will work with a 12-bit sync character. While the receiver is in External Sync mode the transmitter sync length may be six or eight bits, as selected by bit D0 of WR10.

The number of bits per transmitted character is controlled by bits D_6 and D_5 of WR5 and the way the data is formatted within the transmit buffer. The bits in WR5 allow the option of five, six, seven, or eight bits per character. When five bits per character is selected the data may be formatted before being written to the transmit buffer to allow transmission of from one to five bits per character. This formatting is shown in table 5. In all cases the data must be right-justified, with the unused bits being ignored except in the case of five bits per character. An additional bit, carrying parity information, may be automatically appended to every transmitted character by setting bit D0 of WR4 to "1". This parity bit is sent in addition to the number of bits specified in WR4 or by the data format. If this bit is set to "1", the transmitter will send even parity, if set to "0", the transmitted parity will be odd.

Either of two CRC polynomials may be used in synchronous modes, selected by bit D_2 in WR5. If this bit is set to "1", the CRC-16 polynomial is used and, if this bit is set to "0", the CRC-CCITT polynomial is used. This bit controls the selection for both the transmitter and receiver. The initial state of the generator and checker is controlled by bit D_7 of WR10. When this bit is set to "1", both the generator and checker will have an initial value of all ones, if this bit is set to "0", the initial values will be all zeros. The SCC does not automatically preset the CRC generator, so this must be done in software. This is accomplished by issuing the Reset Tx CRC Generator command, which is encoded in bits D_7 and D_6 of WR0. For proper results this command must be issued while the transmitter is enabled and sending sync characters. If CRC is to be used, the transmit CRC generator must be enabled by setting bit D0 of WR5 to "1". This bit may also be used to exclude certain characters from the CRC calculation. Sync characters are automatically excluded from the CRC calculation and any characters written as data may also be excluded from the calculation by using bit D_0 of WR5. Internally, the CRC is enabled or disabled for a particular character at the same time as the character is loaded from the transmit buffer to the Transmit Shift register. Thus, to exclude a character from CRC calculation bit, D_0 of WR5 should be set to "0" before the character is written to the transmit buffer. This guarantees that the internal disable will occur when the character moves from the buffer to the shift register. Once the buffer becomes empty, the Tx CRC Enable bit may be written for the next character.

Enabling the CRC generator is not sufficient to control the transmission of CRC. In the SCC this function is controlled by the Tx Underrun/EOM bit, which may be reset by the processor and set by the SCC. When the transmitter underruns (both the transmit buffer and Transmit Shift register are empty) the state of the Tx Underrun/EOM bit determines the action taken by the SCC. If the tx Underrun/EOM bit is set when the underrun occurs, the transmitter will send sync characters, if this bit is reset when the underrun occurs, the transmitter will send the accumulated CRC followed by sync characters. When the CRC is loaded into the transmit Shift register for transmission, the SCC will set the Tx Underrun/EOM bit to indicate this. This transition may be programmed to cause an external/status interrupt, or the Tx Underrun/EOM is available in RR0. The Reset Tx Underrun/EOM Latch command is encoded in bits D_7 and D_6 in WR0. For correct transmission of the CRC at the end of a block of data, this command must be issued after the first character is written to the SCC but before the transmitter underruns after the last character written to the SCC. The command is usually issued immediately after the first character is written to the SCC so that CRC will be sent if an underrun occurs inadvertently during the block of data.

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

In synchronous modes, if the transmitter is disabled during transmission of a character, that character will be sent completely. This applies to both data and sync characters. However, if the transmitter is disabled during the transmission of CRC, the 16-bit transmission will be completed, but the remaining bits will come from the SYNC registers rather than the remainder of the CRC.

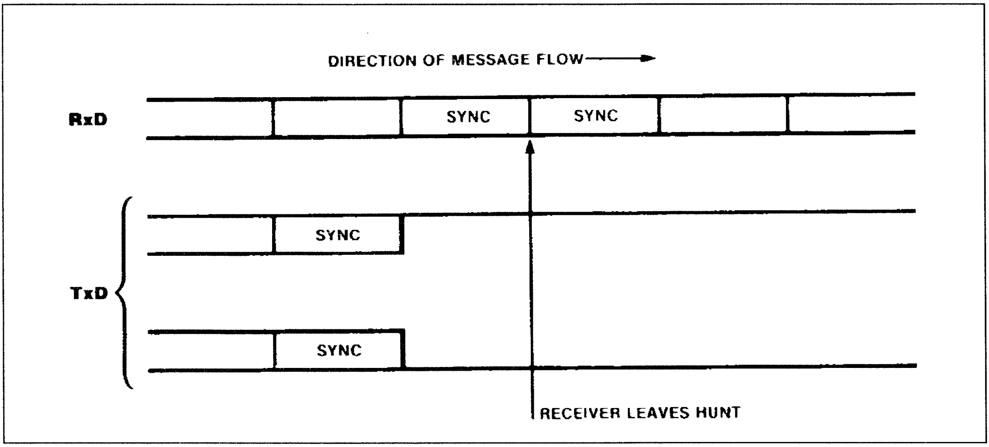
There are two modem control signals associated with the transmitter provided by the SCC. The $\overline{RTS}$ pin is a simple output that carries the inverted state of the RTS bit (D1) in WR5. The $\overline{CTS}$ pin is ordinarily a simple input to the CTS bit in RR0. However, if Auto Enables mode is selected this pin becomes an enable for the transmitter. That is, if Auto Enables is on and the $\overline{CTS}$ pin is HIGH the transmitter is disabled. While the $\overline{CTS}$ pin is LOW, transmitter is enabled.

The initialization sequence for the transmitter in synchronous modes is : WR4 first, to select the mode, then WR10 to modify it if necessary, WR6 and WR7 to program the sync characters, then WR3 and WR5 to program the sync characters, and then WR3 and WR5 to select the various options. At this point, the other registers should be initialized as necessary. When all of this is complete the transmitter may be enabled by setting bit D3 or WR5 to "1". Now that the transmitter is enabled the CRC generator may be initialized by issuing the Reset Tx CRC Generator command in WR0.

Transmitter to Receiver Synchronization. The SCC contains a transmitter-to-receiver synchronization function that may be used to guarantee that the character boundaries for the received and transmitted data are the same. In this mode the receiver is in Hunt and the transmitter is idle, sending either all "1s" or "0s". When the receiver recognizes a sync character, it leaves Hunt mode and one character time later the transmitter is enabled and begins sending sync characters. Beyond this point the receiver and transmitter are again completely independent, except that the character boudaries are now aligned. This is shown in figure 29. There are several restrictions on the use of this feature in the SCC. First, it will only work with 6-bit, 8-bit or 16-bit sync characters, and the data character length for both the receiver and the transmitter must be six bits with a 6-bit sync character or eight bits with an 8-bit or 16-bit sync character. Of course, the receive and transmit clocks must have the same rate as well as the proper phase relation-ship.

A specific sequence of operations must be followed to synchronize the transmitter to the receiver. Both the receiver and transmitter must have been initialized for operation in Synchronous mode sometime in the past, although this initialization need not be redone each time the transmitter is synchronized to the receiver. The transmitter is disabled by setting bit D_3 of WR5 to "0". At this point the transmitter will send continuous "1s". If it is desired that continuous "1s". If it is desired that continuous "0s" be transmitted, the Send Break bit (D_4) in WR5 should be set to "1". The transmitter is now idling but must still be placed in the transmitter to receiver syn-

Figure 29 : Transmitter to Receiver Synchronization.



PROGRAMMING DATA COMMUNICATION MODES (cont'd)

chronization mode. This is accomplished by setting the Loop the processor should set the Go Active on Poll bit (D_4) in WR10. The final step is to force the receiver to search for sync characters. If the receiver is currently disabled the receiver will enter Hunt mode when it is enabled by setting bit D_0 of WR3 to "1". If the receiver is already enabled it may be placed in Hunt mode by setting bit D_4 of WR3 to "1". Once the receiver leaves Hunt mode the transmitter is activate on the following character boundary.

SDLC Mode

SDLC mode is useful in bit-oriented protocols. That is, protocols which use the technique of "0" insertion to achieve data transparency. In SDLC mode, frames of information are opened and closed by a unique bit pattern called a flag. The Flag character has a bit pattern of "01111110" and this sequence is unique because all data between the opening and closing flags is prohibited from having more than five consecutive "1s". The transmitter guarantees this by watching the transmit data stream and inserting a "0" after five consecutive ones, irrespective of character boundaries. In turn, the receiver searches the receive data stream for five consecutive "1s" and deletes the next bit if it is a "0". CRC may be used in SDLC mode but only with the CRC-CCITT polynomial, because the transmitter in the SCC automatically inverts the CRC before transmission, and the receiver - to compensate for this - checks the CRC result for the bit pattern "0001110100001111". This is consistent with bit-oriented protocols such as SDLC, HDLC, and ADCCP. There are two unique bit patterns in SDLC mode besides the flag sequence. They are the Abort and EOP (End of Poll) sequence. An Abort is a sequence of from seven to thirteen consecutive "1s" and is used to signal the premature termination of a frame. The EOP is the bit pattern "11111110", which is used in loop applications as a signal to a secondary station that it may begin transmission.

SDLC mode is selected by setting bit D_5 of WR4 to "1" and bits D_4, D_3, and D_2 of WR4 to "0". In addition, the flag sequence must be written to WR7. Additional control bits for SDLC mode are located in WR10.

SDLC Receive. The receiver in the SCC always searches the receive data stream for flag characters in SDLC mode. Ordinarily, the receiver transfers all received data between flags to the receive data FIFO. However, if the receiver is in Hunt mode no flag is received. The receiver is in Hunt mode when first enabled, or the receiver may be placed in Hunt mode by the processor issuing the Enter Hunt mode command in WR3. This bit (D_4) is a command, and writing a "0" to it has no effect. The Hunt status of the receiver is reported by the Sync/Hunt bit in RR0. Sync/Hunt is one of the possible sources of external/status interrupts, with both transitions causing an interrupt. This is true even if the Sync/Hunt bit is set as a result of the processor issuing the Enter Hunt mode command. The receiver will automatically enter Hunt mode if an abort is received. Because the receiver always searches the receive data stream for flags and automatically enter Hunt Mode when an abort is received, the receiver will always handle frames correctly, and the Enter Hunt Mode command should never to needed. The SCC will drive the $\overline{\text{SYNC}}$ pin LOW to signal that a flag has been recognized. The timing for the $\overline{\text{SYNC}}$ signal is shown in figure 30.

The first byte in an SDLC frame is assumed by the SCC to be the address of the secondary station for which the frame is intended. The SCC provides several options for handling this address. If the Address Search Mode bit (D_2) in WR3 is set to "0", the address recognition logic is disabled and all received frames are transferred to the receive data FIFO. In this mode the software must perform any address recognition. If the Address Search Mode bit is set

Table 6 : Residue Codes.

Residue Code			Bits in Previous Byte				Bits in Second Previous Byte				Bits in Third Previous Byte			
2	1	0	8 B/C	7 B/C	6 B/C	5 B/C	8 B/C	7 B/C	6 B/C	5 B/C	8 B/C	7 B/C	6 B/C	5 B/C
1	0	0	0	0	0	0	3	1	0	0	8	7	5	2
0	1	0	0	0	0	0	4	2	0	0	8	7	6	3
1	1	0	0	0	0	0	5	3	1	0	8	7	6	4
0	0	1	0	0	0	0	6	4	2	0	8	7	6	5
1	0	1	0	0	0	0	7	5	3	1	8	7	6	5
0	1	1	0	0	0	-	8	6	4	-	8	7	6	-
1	1	1	1	0	-	-	8	7	-	-	8	7	-	-
0	0	0	2	-	-	-	8	-	-	-	8	-	-	-

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

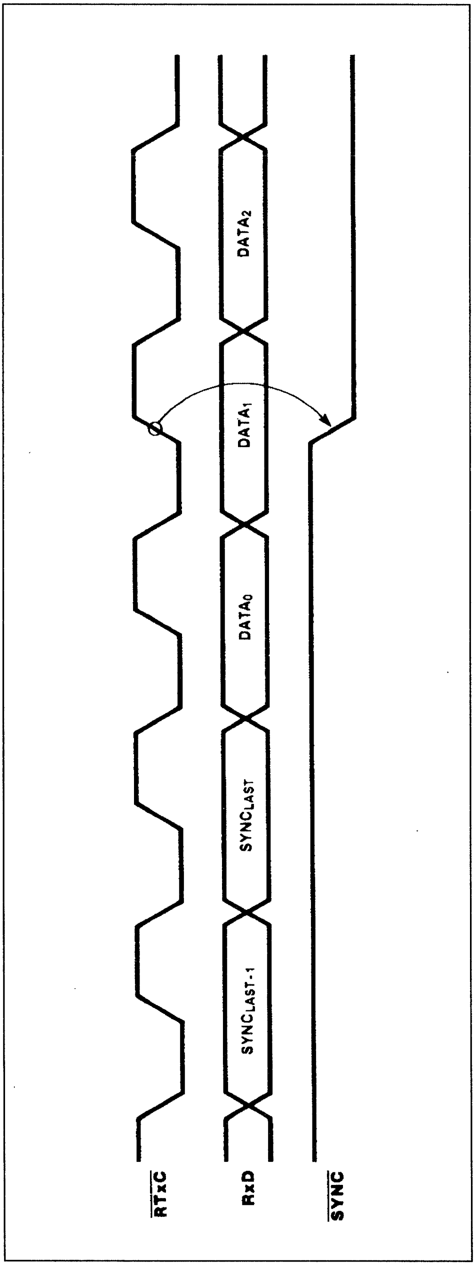
to "1", only those frames whose address matches the address programmed in WR6 or the global address (all "1s") will be transferred to the receive data FIFO. The address comparison will be across all eight bits of WR6 if the Sync Character Load Inhibit bit (D_1) in WR3 is set to "0". The comparison may be modified so that only the four most significant bits of WR6 must match the received address. This mode is selected by setting the Sync Character Load Inhibit bit to "1". In this mode, however, the address field is still eight bits wide. The address field is transferred to the receive data FIFO in the same manner as data. It is not treated differently than data.

The number of bits per character is controlled by bits D_7 and D_6 of WR3. Five, six, seven, or eight bits per character may be selected via these two bits. The data is right-justified in the receive buffer. The SCC merely takes a snapshot of the receive data stream at the appropriate times, so the "unused" bits in the receive buffer are only the bits following the character in the data stream. An additional bit carrying parity information may be selected by setting bit D_6 of WR4 to "1". This also enables parity in the transmitter. The parity sense is selected by bit D_1 of WR4. Parity is not normally used in SDLC mode. The character length may be changed at any time before the new number of bits have been assembled by the receiver. Care should be exercised, however, as unexpected results may occur. A representative example, switching from five bits to eight bits and back to five bits is shown in figure 31.

Most bit-oriented protocols allow an arbitrary number of bits between opening and closing Flags. The SCC allows for this by providing three bits of Residue Code in RR1 that indicate which bits in the last few bytes transferred from the receive data FIFO by the processor are actually valid data bits. The meaning of these three bits with each character length option is shown in table 6. As indicated in the table, these bits allow the processor to determine those bits in the information (and not CRC) field. This allows transparent retransmission of the received frame. The Residue Code bits do not go through a FIFO so they change in RR1 when the last character of the frame is loaded into the receive data FIFO. If there are any characters already in the receive data FIFO the Residue Code will be updated before they are read by the processor. Thus these three bits off RR1 should be ignored by the processor unless the End of Frame bit in RR1 is set.

Only the CRC-CCITT polynomial may be used for CRC calculation in SDLC mode, although the generator and checker may be preset to all "1s" or

Figure 30 : SYNC as an Output.



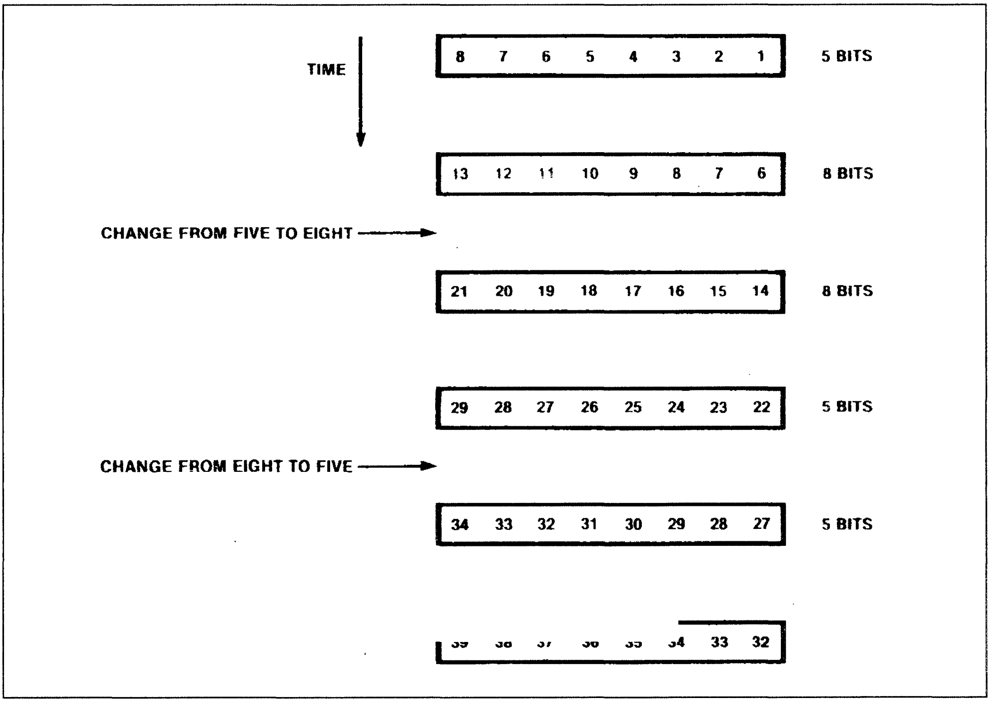
PROGRAMMING DATA COMMUNICATION MODES (cont'd)

all "0s". The CRC-CCITT polynomial is selected by setting D_2 of WR5 to "0", bit D_7 of WR10 controls the preset value. If this bit is set to "1", the generator and checker are preset to "1s", if this bit is reset, the generator and checker are preset to all "0s". The receiver expects the CRC to be inverted before transmission and so checks the CRC result against the value "0001110100001111". The SCC presets the CRC checker whenever the receiver is in Hunt mode or whenever a flag is received so a CRC reset command is not strictly necessary. However, the CRC checker may be preset by issuing the Reset CRC Checker command in WRO. The CRC checker is automatically enabled for all data between the opening and closing flags by the SCC in SDLC mode, and the Rx CRC Enable bit (D_3) in WR3 is ignored. The result of the CRC calculation for the entire frame is valid in RR1 only when accompanied by the end of Frame bit being set in RR1. At all other times the CRC Error bit in RR1 should be ignored by the processor. Care must be exercised so that the processor does not attempt to use the CRC bytes that are transferred as data because not all of the bits are transferred properly. The last two bits of CRC are never transferred to the receive data FIFO and are not recoverable.

A frame is terminated by a closing flag. When the SCC recognizes this flag the contents of the Receive Shift register are transferred to the receive data FIFO, the Residue Code is latched, the CRC Error bit is latched in the status FIFO and the End Of Frame bit is set in the receive status FIFO. The End Of Frame bit, upon reaching the top of the FIFO, will cause a special receive condition. The processor may then read RR1 to determine the result of the CRC calculation as well as the Residue Code. If either the Rx Interrupt on Special Condition Only or the Rx Interrupt on First Character or Special Condition mode are selected, the processor must issue and Error Reset command in WR0 to unlock the receive FIFO.

In addition to searching the data stream for flags, the receiver in the SCC also watches for seven consecutive "1s", which is the abort condition. The presence of seven consecutive "1s" is reported in

Figure 31 : Changing Character Length.



PROGRAMMING DATA COMMUNICATION MODES (cont'd)

the Break/Abort bit in RR0. This is one of the possible external/status interrupts, so transitions of this status may be programmed to cause interrupts. Upon receipt of an abort the receiver is forced into Hunt mode, where it looks for flags. The Hunt status is also a possible external/status condition whose transition may be programmed to cause an interrupt. The transitions of these two bits occur very close together but either one or two external/status interrupts may result. The abort condition is terminated when a "0" is received, either by itself or as the leading "0" of a flag. The receiver does not leave Hunt mode until a flag has been received so two discrete external/status conditions will occur at the end of an abort. An abort received in the middle of a frame terminates the frame reception, but no in an orderly manner, because the character being assembled is lost.

Up to two modem control signals associated with the receiver are available in SDLC mode. The DTR/$\overline{\text{REQ}}$ pin carries the inverted state of the DTR bit (D_7) in WR5 unless this pin has been programmed to carry a DMA Request signal. The DCD pin is ordinarily a simple input to the DCD bit in RR0. However, if the Auto Enables mode is selected by setting bit D5 of WR3 to "1", this pin becomes an enable for the receiver. That is, if Auto Enable is on and the $\overline{\text{DCD}}$ pin is HIGH the receiver is disabled. While the $\overline{\text{DCD}}$ pin is LOW, the receiver is enabled.

The initialization sequence for the receiver in SDLC mode is : WR4 first, to select the mode, then WR10 to modify it if necessary, WR6 to program the address, WR7 to program the flag and the WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is complete the receiver may be enabled by setting bit D_6 of WR3 to "1".

SDLC Transmit. Once SDLC mode has been selected, the flag must be written in WR7, to be used to open and close the transmitted frames. The SCC does not automatically send the address byte ; it merely encapsulates the data supplied by the processor with flags and CRC. Ordinarily, a frame will be terminated by the SCC with CRC and a flag but the SCC may be programmed to send an abort and a flag in place of the CRC. This option allows the SCC to abort a frame transmission in progress if the transmitter is accidentally allowed to underrun. This is controlled by the Abort/$\overline{\text{Flag}}$ On Underrun bit (D_2) in WR10. When this bit is set to "1" the transmitter will send an abort and a flag in place of the CRC when an underrun occurs. The frame will be terminated normally, with CRC and a flag, if this bit is set to "0". The SCC is also able to send an abort by command of the processor. The Send Abort command, issued in WR0, will send eight consecutive "1s" and then the transmitter will idle. Since up to five consecutive "1s" may have been sent prior to the command being issued, a Send Abort will cause a sequence of from eight to thirteen "1s" to be transmitted. The Send Abort command also empties the transmit buffer register. The idle condition for the transmitter is continuous flags, but this is under program control. By setting the Mark/$\overline{\text{Flag}}$ Idle bit (D_3) in WR10 to "1", the transmitter will send continuous "1s" in place of the idle flags. Note that the closing flag will be transmitted correctly even if this mode is selected. The Mark/$\overline{\text{Flag}}$ Idle must be set to "0", allowing a flag to be transmitted, before data is written to the transmit buffer. Care must be exercised in doing this because the continuous "1s" are transmitted eight at a time, and all eight must leave the Transmit Shift register, so that flag may be loaded into it before the first data is written to the transmit buffer. When using the transmitter in SDLC mode, recall that all data passes through the zero inserter, which adds an extra five bit times of delay between the Transmit Shift register and the Transmit Data pin.

The number of bits per transmitted character is controlled by bits D_6 and D_5 of WR5 and the way the data is formatted within the transmit buffer. The bits in WR5 allow the option of five, six, seven, or eight bits per character. When five bits per character is selected, the data may be formatted before being written to the transmit buffer, to allow transmission of one to five bits per character. This formatting is shown in table 3. In all cases the data must be right-justified, with the unused bits being ignored, except in the case of five bits per character.

An additional bit, carrying parity information, may be automatically appended to every transmitted character by setting bit D_6 of WR4 to "1". This bit is sent in addition to the number of bits specified in WR4 or by the data format. The parity sense is selected by bit D_1 of WR4. Parity is not normally used in SDLC mode. The character length may be changed on the fly, but the desired length must be selected before the character is loaded into the transmit shift register from the transmit buffer. The easiest way to ensure this is to write to WR5 to change the character length before writing the data to the transmit buffer.

Only the CRC-CCITT polynomial may be used in SDLC mode. This is selected by setting bit D_2 in WR5 to "0". This bit controls the selection for both the transmitter and receiver. The initial state of the

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

generator and checker is controlled by bit D_7 of WR10. When this bit is set to "1", both the generator, and checker sill have an initial value of all "1s" and, if this bit is set to "0", the initial values will be all "0s". The SCC does not automatically preset the CRC generator so this must be done in software. This is accomplished by issuing the Reset Tx CRC generator command, which is encoded in bits D_7 and D_6 of WR0. For proper results, this command must be issued while the transmitter is enabled and idling. If CRC is to be used the transmit CRC generator must be enabled by setting bit D0 of WR5 to "1". CRC is normally calculated on all characters between opening and closing flags, so this bit is usually set to "1" at initialization and never changed.

Enabling the CRC generator is not sufficient to control the transmission of CRC. In the SCC this function is controlled by the Tx Underrun/EOM bit, which may be reset by the processor and set by the SCC. When the transmitter underruns (both the transmit buffer and transmit shift register are empty) the state of the Tx Underrun EOM bit determines the action taken by the SCC. If the Tx Underrun/EOM bit is set to "1" when the underrun occurs, the transmitter will send flags ; if this bit is reset to "0" when the underrun occurs, the transmitter will send either the accumulated CRC followed by flags, or an abort followed by flags, depending the state of the Abort/$\overline{\text{Flag}}$ on Underrun bit in the WR10. When the CRC or abort is loaded into the Transmit Shift register for transmission, the SCC will set the Tx Underrun/EOM bit to indicate this. This transition may be programmed to cause an external/status interrupt, or the Tx Underrun/EOM bit is available in RR0. The Reset Tx Underrun/EOM Latch command is encoded in bits D_7 and D_6 of WR0.

For correct transmission of the CRC at the end of a frame, this command must be issued after the first character is written to the SCC but before the transmitter underruns after the last character written the SCC. The command is usually issued immediately after the first character is written to the SCC so that the abort or CRC is sent if an underrun occurs inadvertently. The Abort/$\overline{\text{Flag}}$ on Underrun bit (D2) in WR10 is usually set to "1" at the same time as the Tx Underrun/EOM bit is reset so that an abort sill be sent if the transmitter underruns. The bit is then set to "0" near the end of the frame to allow the correct transmission of CRC.

In this paragraph the term "completely sent" means shifted out of the Transmit Shift register, not shifted out of the zero inserter, which is an additional five bit times of delay. In SDLC mode, if the transmitter is disabled during transmission of a character, that character will be "completely sent". This applies to both data and flags. However, if the transmitter is disabled during the transmission of CRC, the 16-bit transmission will be completed but the remaining bits will be from the Flag register rather than the remainder of the CRC.

There are two modem control signals associated with the transmitter provided by the SCC. The $\overline{\text{RTS}}$ pin is a simple output that carries the inverted state of the RTS bit (D_1) in WR5. The $\overline{\text{CTS}}$ pin is ordinarily a simple input to the $\overline{\text{CTS}}$ bit in RR0. However, if Auto Enables mode is selected, this pin becomes an enable for the transmitter. That is, if Auto Enables is on and the $\overline{\text{CTS}}$ pin is HIGH the transmitter is disabled. If the $\overline{\text{CTS}}$ pin is LOW, the transmitter is enabled.

The initialization sequence for the transmitter in SDLC mode is : WR4 first, to select the mode, then WR10 to modify it if necessary, WR7 to program the flag, and then WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is complete, the transmitter may be enabled by setting bit D_3 of WR5 to "1". Now that the transmitter is enabled, the CRC generator may be initialized by issuing the Reset Tx CRC Generator command in WR0.

SDLC Loop Mode. SDLC Loop mode is quite similar to SDLC mode except that two additional control bits are used. They are the Loop Mode bit (D_1) and the Go Active on Poll bit (D_4) in WR10. In addition to these two extra control bits, there are also two status bits in RR10. They are the On Loop bit (D_1) and the Loop Sending bit (D_4). Before Loop mode is selected both the receiver and transmitter must be completely initialized for SDLC operation. Once this is done, Loop mode is selected by setting bit D_1 of WR10 to "1". At this point the SCC connects TxD to RxD with only gate delays in the path. At the same time a flag is loaded into the Transmit Shift register, and is shifted to the end of the zero inserter, ready for transmission. The SCC will remain in this state until the Go Active on Poll bit (D4) in WR10 is set to "1". When this bit is set to "1" the receiver begins looking for a sequence of seven consecutive "1s", indicating either an EOP or an idle line. When the receiver detects this condition the Break/Abort bit in RR0 is set to "1" and a one-bit time delay is inserted in the path from RxD to TxD. The On Loop bit in RR10 is also set to "1" at this time, and the receiver enters the Hunt mode. The SCC cannot transmit on the loop until a flag is received, causing the receiver to leave Hunt mode, and another EOP (bit pattern "11111110") is received.

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

The SCC is now on the loop and capable of transmitting on the loop. As soon as this status is recognized by the processor, the Go Active On Poll bit in WR10 should be set to "0" to prevent the SCC from transmitting on the loop without the consent of the processor.

To transmit a message on the loop, the Go Active On Poll bit in WR10 must be set to "1". Once this is done, the SCC will change the next received EOP into Flag and begin transmitting on the loop. When the EOP is received, the Break/Abort and Hunt bits in RRO will be set to "1", and the Loop Sending bit in RR10 will also be set to "1". Data to be transmitted may be written after the Go Active On Poll bit has been set or after the receiver enter Hunt mode. If the data is written immediately after the Go Active On Poll bit has been set, the SCC will only insert one flag after the EOP is changed into a flag. If the data is not written until after the receiver enters the Hunt mode, flags will be transmitted until the data is written. If only one frame is to be transmitted on the loop in response to an EOP, the processor must set the Go Active on Poll bit to "0" before the last data is written to the transmitter. In this case the transmitter will close the frame with a single flag, and then revert to the one-bit delay. The Loop Sending bit in RR10 is set to "0" when the closing Flag has been sent. If more than one frame is to be transmitted, the Go Active On Poll bit should not be set to "0" until the last frame is being sent. If this bit is not set to "0" before the end of a frame, the transmitter will send Flags until either more data is written to the transmitter, or until the Go Active On Poll bit is set to "0". Note that the state of the Abort/Flag on Underrun and Mark/Flag Idle bits in WR10 are ignored by the SCC in SDLC Loop mode.

To go off the loop in an orderly manner requires actions similar to those taken to go the loop. First, the Go Active On Poll bit must be set to "0" and any transmission in progress completed, if the SCC is currently sending on the loop. Once the SCC is not sending on the loop, an exit from the loop is accomplished by setting the Loop Mode bit in WR10 to "0", and at the same time writing the Abort/Flag on Underrun and Mark/Flag Idle bits with the desired values. The SCC will revert to normal SDLC operation as soon as an EOP is received, or immediately, if the receiver is already in Hunt mode because of the receipt of an EOP.

The initialization sequence for the SCC in SDLC Loop mode is similar to the sequence used in SDLC mode, except that it is somewhat longer. The processor should program WR4 first, to select SDLC mode, and then WR10 to select the CRC preset value and program the Mark/Flag Idle bit. The Loop Mode and Go Active On Poll bits in WR10 should not be set to "1" yet. The flag is written in WR7 and the various options are selected in WR3 and WR5. At this point the other registers should be initialized as necessary, then Loop Mode bit (D_1) in WR10 should be set to "1". When all of this is complete the transmitter may be enabled by setting bit D_3 of WR5 to "1". Now that the transmitter is enabled, the CRC generator may be initialized by issuing the Reset TxCRC Generator command in WR0. The receiver is enabled by setting the Go Active on Poll bit (D_4) in WR10 to "1". The SCC will go on the loop when seven consecutive "1s" are received, and will signal this by setting the On Loop bit in RR10. Note that the seven consecutive "1s" will set the Break/Abort and Hunt bits in RR0 also. Once the SCC is on the loop, the Go Active on Poll bit should be set to "0" until a message is to be transmitted on the loop. To transmit a message on the loop, the Go Active on Poll bit should be set to "1". At this point the processor may either write the first character to the transmit buffer and wait for a transmit buffer empty condition, or wait for the Break/Abort and Hunt bits to be set in RR10 and the Loop Sending bit to be set in RR10 before writing the first data to the transmitter. The Go Active On Poll bit should be set to "0" after the transmission of the frame has begun. To go off of the loop, the processor should set the Go Active On Poll bit in WR10 to "0" and then wait for the Loop Sending bit in RR10 to be set to "0". At this point the Loop Mode bit (D_1) in WR10 is set to "0" to request an orderly exit from the loop. The SCC will exit SDLC Loop mode when seven consecutive "1s" have been received ; at the same time the Break/Abort and Hunt bits in RR0 will be set to "1", and the On Loop bit in RR10 will be set to "0".

SUPPORT CIRCUITRY PROGRAMMING

The SCC incorporates additional circuitry to aid in serial communications. This circuitry includes clocking options, baud rate generator, data encoding, and internal loopback. This section discusses how to program these functions.

Clock Options

The SCC may be programmed to select one of several sources to provide the transmit and receive clocks. In addition, the SCC requires a fundamental, parallel resonant crystal oscillator in each channel, as well as the ability to echo one of several internal clock sources to the outside world. These options are controlled by the bits in WR11.

The crystal oscillator option is controlled by bit D_7 in WR11. When this is set to "0", the crystal oscillator is disabled and all pins function normally. When this bit is set to "1" the crystal oscillator is enabled and a high-gain amplifier is connected between the RTxC pin and the SYNC pin. While the crystal oscillator is enabled, anything that has RTxC selected as its clock source will automatically be connected to the output of the crystal oscillator. While the crystal oscillator is enabled, the SYNC pin is obviously unavailable for other use. In synchronous modes no sync pulse is output, and the External Sync mode cannot be selected. In asynchronous modes the state of the Sync/Hunt bit in RR0 is no longer controlled by the SYNC pin. Instead, the Sync/Hunt bit is forced to "0". The crystal oscillator requires some finite time to stabilize. The oscillator must be allowed to stabilize before it is used as a clock source.

The source of the receive clock is controlled by bits D_6 and D_5 of WR11. The receive clock may be programmed to come from the RTxC pin, the TRxC pin, the output of the baud rate generator, or the transmit output of the DPLL.

The source of the transmit clock is controlled by bits D_4 and D_3 of WR11. The transmit clock may be programmed to come from the RTxC pin, the TRxC pin, the output of the baud rate generator, or the transmit output of the DPLL.

Ordinarily the TRxC pin is an input, but it becomes on output if this pin has not been selected as the source for the transmitter or the receiver, and bit D_2 of WR11 is set to "1". The selection of the signal provided on the TRxC output pin is controlled by bits D_1 and D0 of WR11. The TRxC pin may be programmed to provide the output of the crystal oscillator, the output of the baud rate generator, the receive output of the DPLL or the actual transmit clock. If the output of the crystal oscillator is selected but the crystal oscillator has not been enabled the TRxC pin will be driven HIGH. The option of placing the transmit clock signal on the TRxC pin when it is an output allows access to the transmit output of the DPLL.

Figure 32 shows a simplified schematic diagram of the circuitry used in the clock multiplexing. It shows the inputs to the multiplexer section as well as the various signal inversions that occur in the paths to the outputs. Also shown are the edges used by the receiver, transmitter, baud rate generator and DPLL to sample or send data or otherwise change state. For example, the receiver samples data on the falling edge, but since there is an inversion in the clock path between the RTxC pin and the receiver, a rising edge of the RTxC pin samples the data for the receiver.

Selection of the clocking options may be done anywhere in the initialization sequence, but the final values must be selected before the receiver, transmitter, baud rate generator, or DPLL are enabled to prevent problems from arbitrarily narrow clock signals out of the multiplexers. The same is true of the crystal oscillator, in that the output should be allowed to stabilize before it is used as a clock source.

Baud Rate Generator

Figure 33 shows a block diagram of the baud rate generator. It consists of a 16-bit down-counter, two 8-bit time constant registers and an output divide-by-two. The baud rate generator input comes from the output of a two-input multiplexer, the zero count condition is output to the External/Status Interrupt Section. The baud rate generator may be enabled and disabled by command and is disabled by a hardware reset.

The time constant for the baud rate generator is programmed in WR12 and WR13, with the least-significant byte in WR12. The formulas relating the baud rate to the time constant and vice versa are shown with an example. In these formulas the baud rate generator clock frequency is in Hertz, the desired baud rate in bits/second and the time constant is dimensionless. The example in table 7 assumes a 2.4576MHz clock factor of 16 and shows the time constant for a number of popular baud rates.

SUPPORT CIRCUITRY PROGRAMMING (cont'd)

Time Constant Formulas :

$$\text{Time Constant} = \frac{\text{Clock Frequency}}{2 \cdot (\text{Clock Mode}) \cdot (\text{Baud Rate})} - 2$$

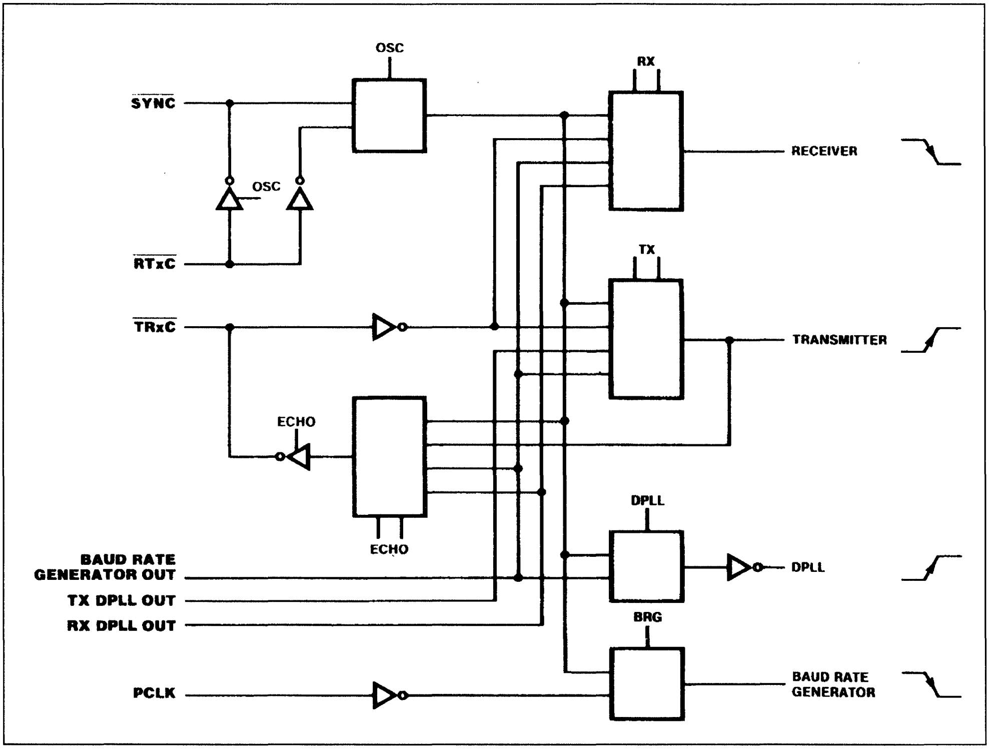
$$\text{Baud Rate} = \frac{\text{Clock Frequency}}{2 \cdot (\text{Clock Mode}) \cdot (\text{Time Const.} + 2)}$$

The clock source for the baud rate generator is selected by bit D_1 of WR14. When this bit is set to "0" the baud rate generator uses the signal on the TxC

Table 7 : Baud Rate Example.

Baud Rate	Divider	
	Decimal	Hex
38400	0	0000H
19200	2	0002H
9600	6	0006H
4800	14	000EH
2400	30	001EH
1200	62	003EH
600	126	007EH
300	254	00FEH
150	510	01FEH
For 2.4576 MHz Clock, X 16 Mode		

Figure 32 : Clock Multiplexer.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

pin as its clock, independent of whether the $\overline{\text{TxC}}$ pin is a simple input or part of the crystal oscillator circuit. When this bit is set to "1" the baud rate generator is clocked by PCLK. To avoid metastable problems in the counter, this bit should be changed only while the baud rate generator is disabled, since arbitrarily narrow pulses can be generated at the output of the multiplexer when it changes status.

The baud rate generator is enabled while bit D_0 of WR14 is set to "1" and is disabled while this bit is

Figure 33 : Baud Rate Generator.

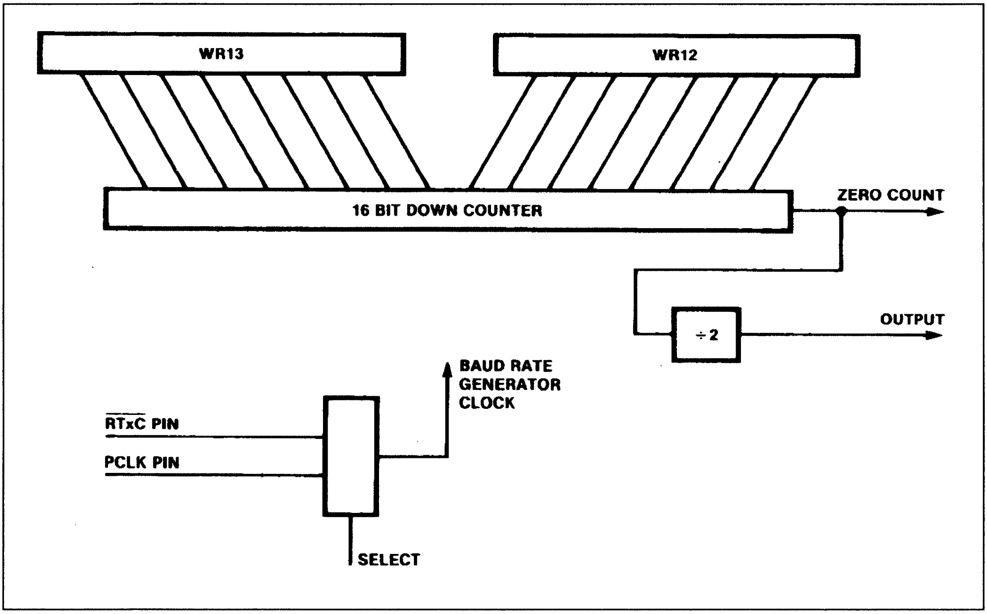
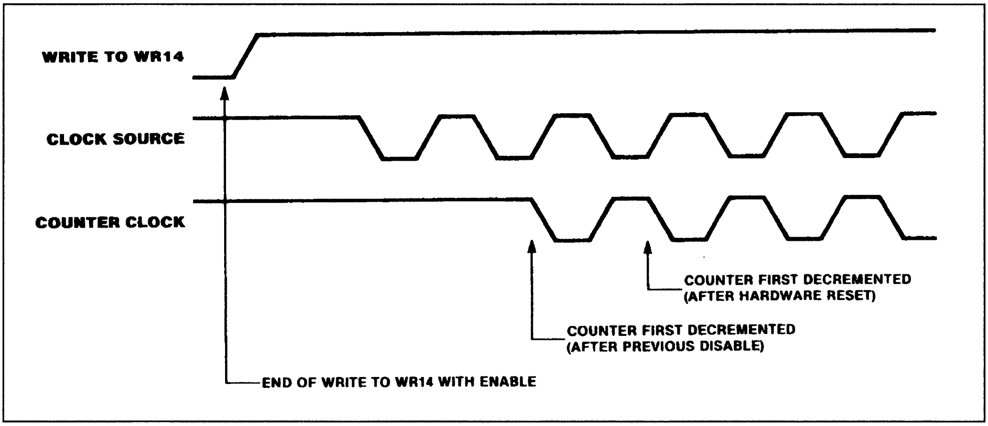


Figure 34 : Baud Rate Generator Start-Up.


WRITE TO WR14
CLOCK SOURCE
COUNTER CLOCK
COUNTER FIRST DECREMENTED (AFTER HARDWARE RESET)
COUNTER FIRST DECREMENTED (AFTER PREVIOUS DISABLE)
END OF WRITE TO WR14 WITH ENABLE

SUPPORT CIRCUITRY PROGRAMMING (cont'd)

set to "0". To prevent metastable problems when the baud rate generator is first enabled, the enable bit is synchronized to the baud rate generator clock. This introduces an additional delay then the baud rate generator is first enabled and this is shown in figure 34. The baud rate generator is disabled immediately when bit D_0 of WR14 is set to "0", because the delay is only necessary on startup. The baud rate generator may be enabled and disabled on the fly, but this delay on startup must be taken into consideration.

Upon reaching a count of "0" the time constant held in WR12 and WR13 is reloaded into the down-counter so that the process of counting down may start over. In addition to reloading the time constant, the output of the baud rate generator toggles, and for the clock cycle with a zero count, the zero count signal goes active to the External/Status Section. This zero count condition from the baud rate generator does not persist, so if it is to be used by the processor, it should be latched in the External/Status latch. While the baud rate generator is disabled the state of the zero count signal is held. This signal is forced active by a hardware reset.

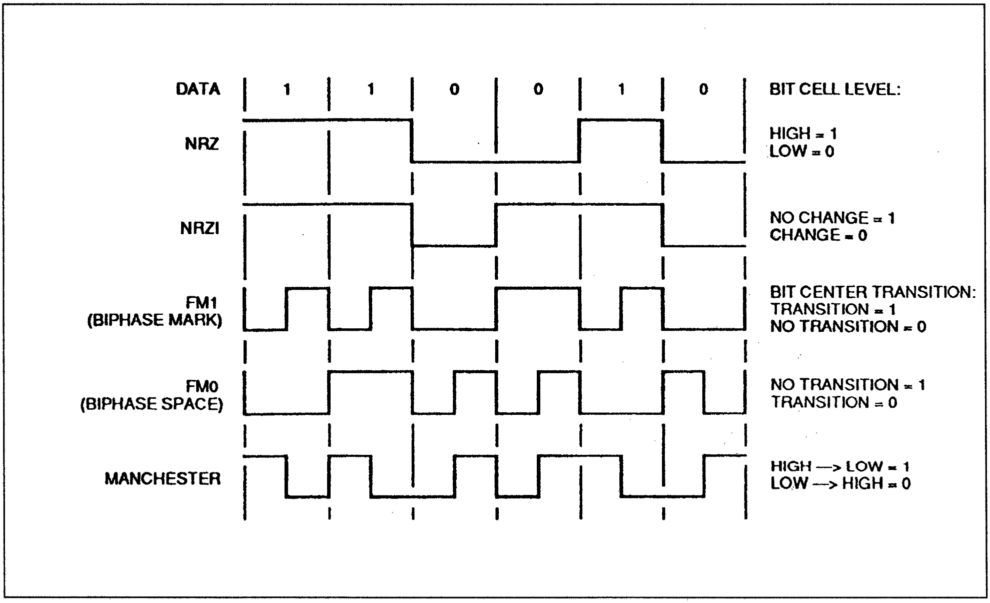
Initializing the baud rate generator is done in four steps. First, the time constant is determined and loaded into WR12 and WR13. Next, the processor must select the clock source for the baud rate generator by writing to bit D_1 of WR14. Finally, the baud rate generator is enabled by setting bit D_0 of WR14 to "1". Note that the first write to WR14 is not necessary after a hardware reset if the clock source is to be the $\overline{\text{RTxC}}$ pin. This is because a hardware reset automatically selects the $\overline{\text{RTxC}}$ pin as the baud rate generator clock source.

Data Encoding

The SCC provides four different data encoding methods, selected by bits D_6 and D_5 in WR10. An example for these four encoding methods is shown in figure 35. Any encoding method may be used in any X1 mode in the SCC, asynchronous or synchronous. The data encoding selected is active even though the transmitter or receiver may be idling or disabled.

In NRZ encoding a "1" is represented by a HIGH level and a "0" is represented by a LOW level. In this encoding method only a minimal amount of clocking in-

Figure 35 : Data Encoding Methods.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

formation is available in the data stream in the form of transitions on bit-cell boundaries. In an arbitrary data pattern this may not be sufficient to generate a clock for the data from the data itself. In the case of SDLC, where the number of consecutive "1s" in the data stream is limited, a minimum number of transitions to generate a clock are guaranteed.

In FM1 encoding, also known as biphase mark, a transition is present on every bit cell boundary, and an addition transition may be present in the middle of the bit cell. In FM1 a "0" is sent as no transition in the center of the bit cell and a "1" is sent as a transition in the center of the bit cell. FM1 encoded data contains sufficient information to recover a clock from the data.

In FM0 encoding, also known as biphase space, a transition is present on every bit cell boundary and an additional transition may be present in the middle on the bit cell. In FM0 a "1" is sent as no transition in the center of the bit cell and a "0" is sent as a transition in the center of the bit cell. FM0 encoded data contains sufficient information to recover a clock from the data.

The data encoding method should be selected in the initialization procedure before the transmitter and receiver are enabled but no other restrictions apply. Note, in figure 35, that in NRZ and NRZI the receiver samples the data only on one edge. However, in FM1 and FM0 the receiver samples the data on both edges. Also, as shown in figure 5-4, the transmitter defines bit cell boundaries by one edge in all cases and uses the other edge in FM1 and FM0 to create the mid-bit transition.

Digital Phase-locked Loop

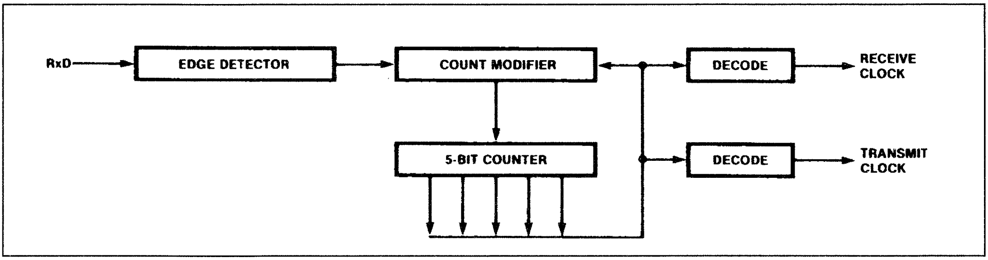
Figure 36 shows a block diagram of the digital phase-locked loop. It consists of a 5-bit counter, an edge detector, and a pair of output decoders. The clock for the DPLL comes from the output of a two-input multiplexer, and the two outputs go to the transmitter and receive clock multiplexers. The DPLL is controlled by the seven commands that are encoded in bits D_7, D_6 and D_5 of WR14.

The clock for the DPLL is selected by two of the commands in WR14. One command selects the output of the baud rate generator as the clock source, and the other command selects the $\overline{\text{RTxC}}$ pin as the clock source, independent of whether the $\overline{\text{RTxC}}$ pin is a simple input or part of the crystal oscillator circuit. To avoid metastable problems in the counter, the clock source selection should be made only while DPLL is disabled, since arbitrarily narrow pulses can be generated at the output of the multiplexer when it changes status.

The DPLL is enabled by issuing the Enter Search Mode command in WR14. This command is also used to reset the DPLL to a known state if it is suspected that synchronization has been lost. When used to enable the DPLL, the Enter Search Mode command unlocks the counter, which is held while the DPLL is disabled and enables the edge detector. If the DPLL is already enabled when this command is issued, the DPLL also enters Search Mode. While in Search mode, the counter is held at a specific count and no outputs are provided. The DPLL remains in this status until an edge is detected in the receive data stream. This first edge is assumed to occur on a bit cell boundary, and the DPLL will begin providing an output to the receiver that will properly sample the data. From this point on the DPLL will adjust its output to remain in phase with the receive data. If the first edge that the DPLL sees does not occur on a bit cell boundary, the DPLL will eventually lock on to the receive data but it will take longer to do so.

The DPLL may be programmed to operate in either of two modes, as selected by command in WR14. In the NRZI mode the DPLL clock must be 32 times the date rate. In this mode the transmit and receive clock outputs of the DPLL are identical, and the clocks are phased so that the receiver samples the data in the middle of the bit cell. In NRZI mode the

Figure 36 : Digital Phase-Locked Loop.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

DPLL does not require a transition in every bit cell, so this mode is useful for recovering the clocking information from NRZ and NRZI data streams. In the FM mode the DPLL clock must be 16 times the data rate. In this mode the transmit clock output of the DPLL lags the receive clock outputs by 90o, to make the transmit and receive bit cell boundaries the same, because the receiver must sample FM data at one-quarters and three-quarters bit time. In FM mode the DPLL requires a transition in every bit cell, and if this transition is not present in two consecutively sampled bit cells, the DPLL will automatically enter the search mode. As in the case of the clock source selection, the mode of operation should only be changed while the DPLL is disabled to prevent unpredictable results.

NRZI Mode Operation

To operate in NRZI mode the DPLL must be supplied with a clock that is 32 times the data rate. The DPLL uses this clock, along with the receive data, to construct receive and transmit clock outputs that are phased to properly receive and transmit data. To do this, the DPLL divides each bit cell into four regions, and makes an adjustment to the count cycle of the 5-bit counter dependent upon in which region a transition on the receive data input occured. This is shown in figure 37. Ordinarily, a bit cell boundary will occur between count 15 and count 16, and the DPLL output will cause the data to be sampled in the middle of the bit cell. The DPLL actually allows the transition marking a bit cell boundary to occur anywhere during the second half of count 15 or the first half of count 16 without making a correction to its count cycle. However, if the transition marking a bit cell boundary occurs between the middle of count 16 and count 31 the DPLL is sampling the data too early in the bit cell. In response to this the DPLL extends its count by one during the next 0 to 31 counting cycle, which effectively moves

Figure 37 : DPLL in NRZI Mode.

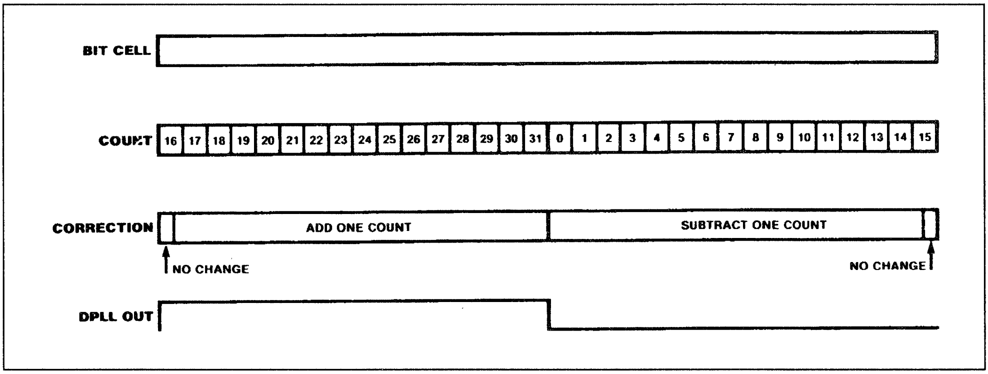
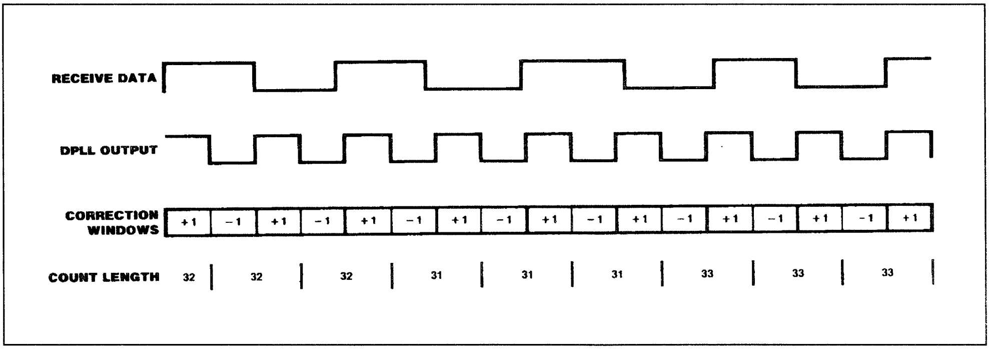


Figure 38 : DPLL Operating Example.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

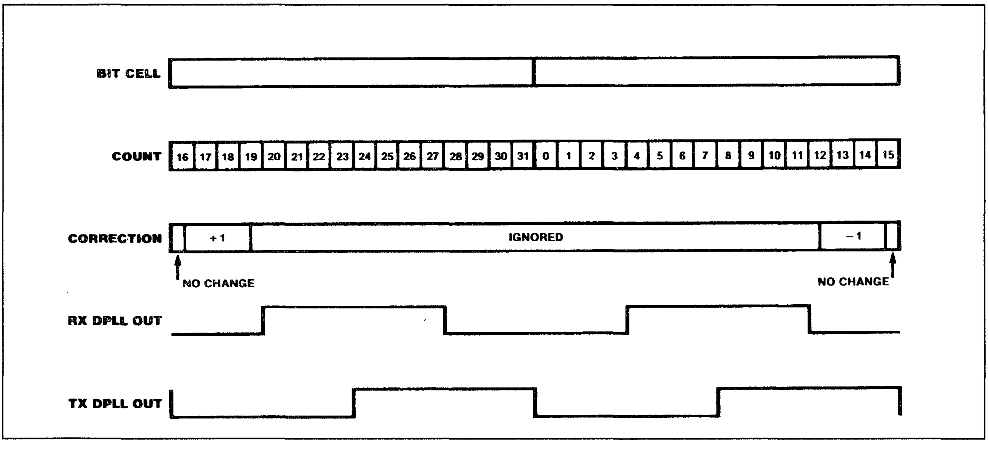
the edge of the clock that samples the receive data closer to the center of the bit cell. In a similar manner, if the transition occurs between count 0 and the middle of count 15, the output of the DPLL is sampling the data too late in the bit cell. To correct this, the DPLL shortens its count by one during the next 0 to 31 counting cycle, which effectively moves the edge of the clock that samples the receive data closer to the center of the bit cell. In NRZI mode, if the DPLL does not see any transition during a counting cycle, no adjustment is made in the following counting cycle. If an adjustment to the counting cycle is necessary the DPLL modifies count five, either deleting it or doubling it. Thus only the LOW time of the DPLL output will be lengthened or shortened. While the DPLL is in search mode, the counter remains at count 16, where the DPLL outputs are both HIGH. The missing clock latches in the DPLL which may be accessed in RR10, are not used in NRZI mode. An example of the DPLL in operation is shown in figure 38.

FM Mode Operation

To operate in FM mode the DPLL must be supplied with a clock that is 16 times the data rate. The DPLL uses this clock, along with the receive data, to construct receive and transmit clock outputs that are phased to receive and transmit data properly. In FM mode that the counter in the DPLL still counts from 0 to 31 but now each cycle corresponds to 2-bit cells. To make adjustments to remain in phase with the receive data, the DPLL divides a pair of bit cells into five regions, making the adjustment to the counter dependent upon which region the transition on the receive data input occured. This is shown in figure 39. Ordinarily a bit cell boundary will occur between count 15 or count 16, and the DPLL receive output will cause the data to be sampled at one-fourth and three-fourths of the way through the bit cell. The DPLL actually allows the transition marking a bit-cell boundary to occur anywhere during the second half of count 15 or the first half of count 16 without making a correction to its count cycle. However, if the transition marking a bit cell boundary occurs between the middle of count 16 and the middle of count 19 the DPLL is sampling the data too early in the bit cell. In response to this the DPLL extends its count by 1 during the next 0 to 31 counting cycle, which effectively moves the receive clock edges closer to where they should be. In FM mode any transitions occurring between the middle of count 19 in one cycle and the middle of count 12 during the next cycle are ignored by the DPLL. This is necessary to guarantee that any data transitions in the bit cells will not cause an adjustment to the counting cycle.

In FM mode the transmit clock and receive clock outputs from the DPLL are not in phase. This is necessary to make the transmit and receive bit cell boundaries coincide, since the receive clock must sample the data one-fourth and three-fourths of the way through the bit cell. As in NRZI mode, if an ad-

Figure 39 : DPLL in FM Mode.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

justment to the counting cycle is necessary, the DPLL modifies count 5, either deleting it or doubling it. If no adjustment is necessary, the count sequence proceeds normally. While the DPLL is in search mode, the counter remains at count 16, where the receive output is LOW and the transmit output is LOW. This fact can be used to provide a transmit clock under software control since the DPLL is in search mode while it it disabled. While the DPLL is disabled the transmit clock output of the DPLL may be toggled by alternately selecting FM and NRZI move in the DPLL. The same is true of the receive clock.

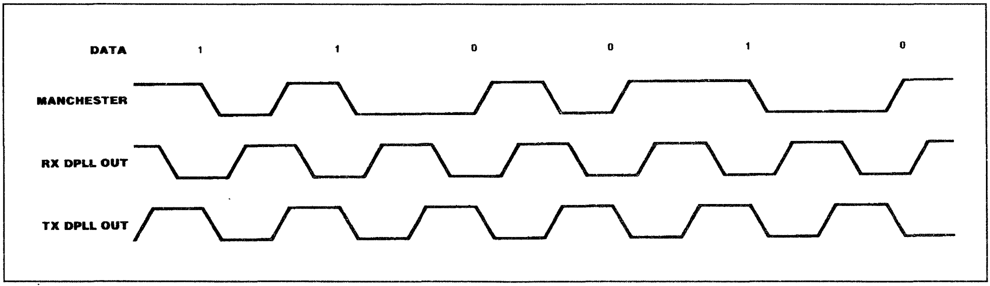
In addition to FM encoded data, the DPLL may also be used to recover the clock from Manchester encoded data, which contains a transition at the center of every bit cell. Here it is the direction of the transition that distinguishes a "1" from a "0". Another way of looking at Manchester encoding is to realize that, during the first half of the bit cell the data is sent, during the second half of the bit cell the complement of the data is sent. This is shown in figure 40, along with the DPLL output if it thinks that the mid-bit transitions are really bit cell boundaries. As is obvious from the figure, if the receiver samples the data on the falling edge of the DPLL receive clock output, the Manchester data will be properly decoded. This occurs if the receiver is programmed to accept NRZ data.

From the above discussion together with an examination of FM0 and FM1 data encoding, it should be obvious that only clock transitions should exist on the receive data pin when the DPLL is programmed to enter search mode. If this is not case the DPLL may attempt to lock on to the data transitions. With FM0 encoding this requires continuous "1s" received when leaving search. In FM1 encoding it is continuous "0s" ; which Manchester encoded data this means alternating "1s" and "0s". With all three of these data encoding methods there will always be at least one transition in every bit cell, and in FM mode the DPLL is designed to expect this transition. In particular, if no transition occurs between the middle of count 12 and the middle of count 19, the DPLL is propably not locked onto the data properly. When the DPLL misses an edge the One Clock Missing bit is RR10 is set to "1" and latched. It will hold this value until a Reset Mission Clock command is issued in WR14 or until the DPLL is disabled or programmed to enter the Search mode. Upon missing this one edge the DPLL takes no other action and does not modify its count during the next counting cycle. However, if the DPLL does not see an edge between the middle of count 12 and the middle of count 19 in two successive 0 to 31 count cycles, a line error condition is assumed. If this occurs, the two Clocks Mission bit in RR10 is set to "1" and latched. At the same time the DPLL enters the Search mode. The DPLL makes the decision to enter Search mode during count 2, where both the receive clock and transmit clock outputs are LOW. This prevents any glitches on the clock outputs when search mode is entered. While in search mode no clock outputs are provided by the DPLL. The Two Clocks Missing bit in RR10 is latched until a Reset Missing Clock command is issued in WR14, or until the DPLL is disabled or programmed to enter the Search mode.

DPLL INITIALIZATION

Initialization of the DPLL may be done at any time during the initialization sequence, but should probably be done after the clock modes have been selected in WR11, and before the receiver and transmitter are enabled. When initializing the DPLL the clock source should be selected first, followed by the selection of the operating mode. At this point the DPLL, may be enabled by issuing the Enter Search Mode command in WR14. Note that a channel or hardware reset disables the DPLL, selects the $\overline{\text{RTxC}}$ pin as the clock source for the DPLL, and places it in the NRZI mode.

Figure 40 : Manchester Clock Recovery.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

Internal Loopback/Auto Echo

The SCC contains two other features useful for diagnostic purposes, controlled by bits in WR14. They are local loopback and auto echo.

Local loopback is selected when bit D4 of WR14 is set to "1". In this mode the output of the transmitter is internally connected to the input of the receiver. At the same time the TxD pin remains connected to the transmitter. In this mode the $\overline{\text{DCD}}$ pin is ignored as a receive enable and the $\overline{\text{CTS}}$ pin is ignored as a transmitter enable even if the Auto Enables mode has been selected. Note that the DPLL input is connected to the RxD pin, not to the input of the receiver. This precludes the use of the DPLL in local loopback.

Auto echo is selected when bit D3 of WR14 is set to "1". In this mode the TxD pin is connected directly to the RxD pin, and the receiver input is connected to the RxD pin. In this mode the $\overline{\text{CTS}}$ pin is ignored as a transmitter enable and the output of the transmitter does not connect to anything. If both the Local Loopback and Auto Echo bits are set to "1", the auto echo mode will be selected, but both the $\overline{\text{CTS}}$ pin and $\overline{\text{DCD}}$ pin will be ignored as auto enables. This, however, should not be considered a normal operating mode, however. Local Loopback is shown schematically in figure 41 and auto echo is shown schematically in figure 42.

Figure 41 : Local Loopback.

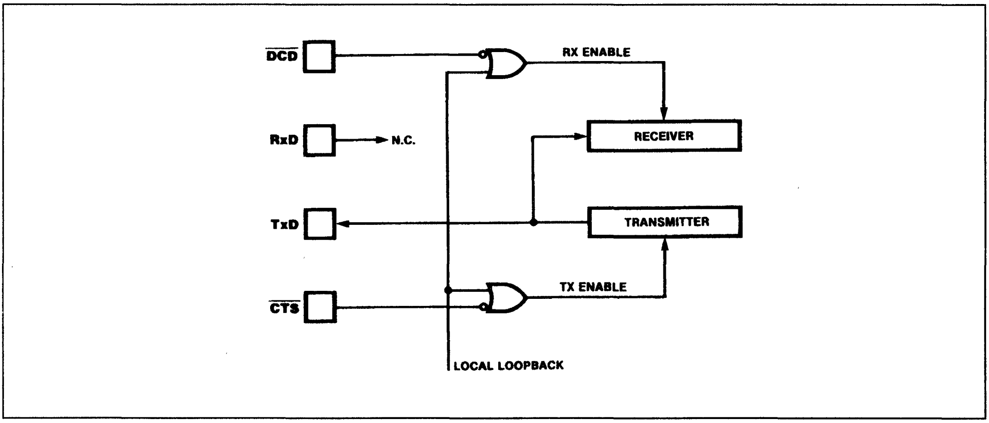
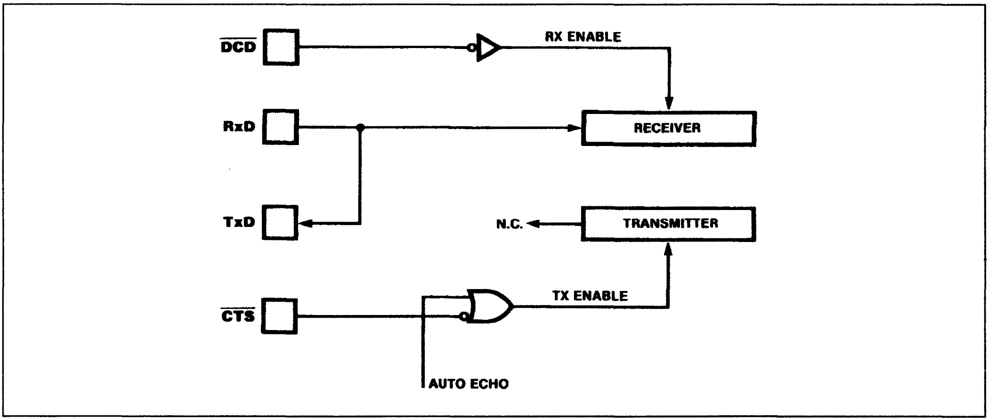


Figure 42 : Auto Echo.



REGISTERS DESCRIPTION

The following section describes the SCC registers. Each register is detailed in terms of bit configuration, the active states (see table 8) of each bit, their definitions, their functions, and their effects upon the internal hardware and external pins.

Table 8 : SCC Registers Description.

Read Register	Description
RR0	Xmit/Receive Buffer Status and Ext Status
RR1	Receive Condition Status/Residue Codes
RR2	Interrupt Vector (modified in BChannel)
RR3	Interrupt Pending (Channel Aonly)
RR8	Receive Buffer
RR10	Loop/Clock Status
RR12	Lower Byte of Time Constant
RR13	Upper Byte of Time Constant
RR15	External Status Interrupt Enable
Write Register	**Description**
WRO	Command Register
WR1	Tx/Rx Interrupt and Data Xfer Mode Definition
WR2	Interrupt Vector
WR3	Receive Parameters and Control
WR4	Tx/Rx Miscellaneous Parameters and Modes
WR5	Transmit Parameter and Controls
WR6	Sync Character or SDLC Address Field
WR7	Sync Character or SDLC Flag
WR8	Transmit Buffer
WR9	Master Interrupt Control
WR10	Misc Transmitter/Receiver Control Bits
WR11	Clock Mode Control
WR12	Lower Byte Baud Rate Generator Time Constant
WR13	Upper Byte of Baud Rate Generator Time Constant
WR14	Miscellaneous Control Bits
WR15	External Status/Interrupt Control

WRITE REGISTERS

The SCC write register set in each channel includes ten control registers (among them is the transmit buffer), two sync character registers, and two baud rate time constant registers. The interrupt control register and the master interrupt control and reset register are shared by both channels.

Write Register 0 (command register). WR0 is the command register and the CRC reset code register. Figure 43 shows the bit configuration for the MK85C30 and includes register select bits in addition to command and reset codes.

Bits D7 and D6 : CRC Reset Codes 0 and 1

Null Code (00). This command has no effect on the SCC and is used when a write to WR0 is necessary for some reason other than a CRC Reset command.

Reset Receive CRC Checker (01). This command is used to initialize the receive CRC circuitry. It is necessary in synchronous modes (except SDLC) if the Enter Hunt Mode command in Write Register 3 is not issued between received messages. Any action that disables the receiver initializes the CRC circuitry. Resetting the Receive CRC Checker command is accomplished automatically in SDLC mode.

Reset Transmit CRC Generator (10). This command initializes the CRC generator. It is usually issued in the initialization routine and after the CRC has been transmitted. A Channel Reset will not initialize the generator and this command should not be issued until after the transmitter has been enabled in the initialization routine.

Reset Transmit Underrun/EOM Latch (11). This command controls the transmission of CRC at the end of transmission (EOM). If this latch has been reset, and a transmit underrun occurs, the SCC automatically appends CRC to the message. In SDLC mode with Abort on Underrun selected, the SCC sends an abort, and Flag on underrun if the TX Underrun/EOM latch as been reset.

At the start of CRC transmission, the Tx Underrun/EOM latch is set. The Reset command can be issued at any time during a message. If the transmitter is disabled, this command will not reset the latch. However, if no External Status interrupt is pending, or if a Reset External Status Int command accompanies this command while the transmitter is disabled, an External/Status interrupt is generated with the Tx Underrun/EOM bit reset in RR0.

Bits D5-D3 : Command Codes

Null Code (000). The Null command has no effect on the SCC.

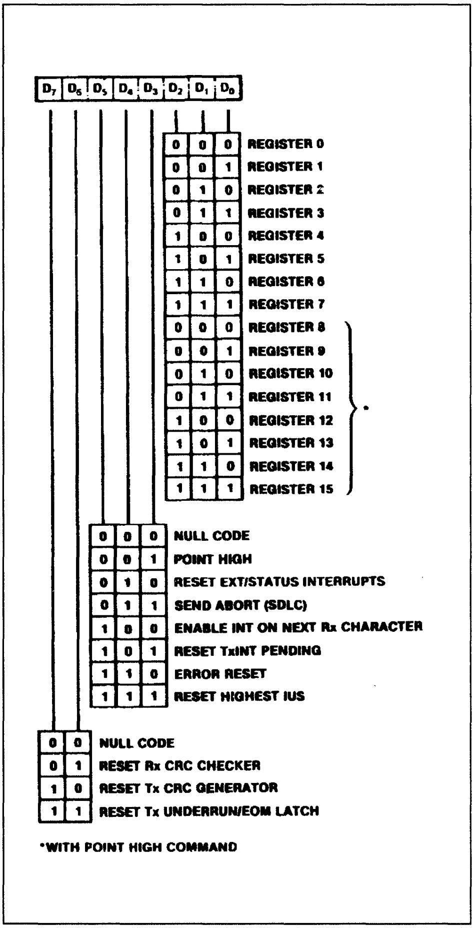
Point High (001). This command effectively adds eight to the Register Pointer (B2-B0) by allowing WR8 through WR15 to be accessed. The Point High command and the Register Pointer bits are written simultaneously.

Reset External/Status Interrupts (010). After an External/Status interrupt (a change on a modem line or a break condition, for example), the status bits in RR0 are latched. This command re-enables the bits and allows interrupts to occur again as a result of a status change. Latching the status bits captures short pulses until the CPU has time to read the change. The SCC contains simple queueing logic associated with most of the external status bits in RR0. If another External/Status condition

REGISTERS DESCRIPTION (cont'd)

changes while a previous condition is still pending (Reset External/Status Interrupts has not yet been issued) and this condition persists until after the command is issued, this second change causes another External/Status interrupt. However, if this second status change does not persist (there are two transitions), another interrupt is not generated. Exceptions to this rule are detailed in the RR0 description.

Figure 43 : Write Register 0 (MK85C30).



Send Abort (011). This command is used in SDLC mode to transmit a sequence of eight to thirteen "1s". This command always empties the transmit buffer and sets Tx Underrun/EOM bit in Read Register 0.

Enable Interrupt on Next Rx Character (100). If the interrupt on the First Received Character mode is selected, this command is used to reactivate that mode after each message is received. The next character to enter the receive FIFO causes a Receive interrupt. Alternatively, the first previously stored character in the FIFO will cause a Receive interrupt.

Reset Tx Interrupt Pending (101). This command is used in cases where there are no more characters to be sent ; e.g., at the end of a message. This command prevents further transmit interrupts until after the next character has been loaded into the transmit buffer or until CRC has been completely sent. This command is necessary to prevent the transmitter from requesting an interrupt when the transmit buffer becomes empty (with Transmit Interrupt Enabled).

Error Reset (110). This command resets the error bits in RR1. If Interrupt on First Rx Character or Interrupt on Special Condition modes are selected and a special condition exists, the data with the special condition is held in the receive FIFO until this command is issued. If either of these modes is selected and this command is issued before the data has been read from the receive FIFO, the data is lost.

Reset Highest IUS (111). This command resets the highest priority Interrupt Under Service (IUS) bit, allowing lower priority conditions to request interrupts. This command allows the use of the internal daisy chain (even in systems without an external daisy chain) and should be the last operation in an interrupt service routine.

Bits 2 through 0 : Register Selection Code

These three bits select Registers 0 through 7. With the Point High command, Registers 8 through 15 are selected.

The following is a summary of the bit descriptions for each write register (WR1-WR15).

Write Register 1 (transmit/receive interrupt and data transfer mode definition). Write Register 1 is the control register for the various SCC interrupt and Wait/Request modes. Figure 44 shows the bit assignments for WR1.

REGISTERS DESCRIPTION (cont'd)

Bit 7 : WAIT/DMA Request Enable

This bit enables the Wait/Request function in conjunction with the Request/Wait Function Select bit (B6). If bit 7 is set to "1", the state of bit 6 determines the activity of the WAIT/REQUEST pin (Wait or Request). If bit 7 is set to "0", the selected function (bit 6) forces the WAIT/REQUEST pin in to the appropriate inactive state (High for Request, floating for Wait).

Bit 6 : WAIT/DMA Request Function

The request function is selected by setting this bit to "1". In the Request mode, the WAIT/REQUEST pin switches from High to Low when the SCC is ready to transfer data. When this bit is "0", the wait function is selected. In the Wait mode, the WAIT/REQUEST pin switches from floating to Low when the CPU attempts to transfer data before the SCC is ready.

Bit 5 : WAIT/DMA Request On Receive Transmit

This bit determines whether the WAIT/REQUEST pin operates in the Transmit mode or the Receive mode. When set to "1", this bit allows the wait/request function to follow the state of the receive buffer ; i.e., depending on the state of bit 6, the WAIT/REQUEST pin is active or inactive in relation to the empty or full state of the receive buffer. Conversely, if this bit is set to "0", the state of the WAIT/REQUEST pin is determined by bit 6 and the state of the transmit buffer. (Note that a transmit request function is available on the DTR/REQUEST pin. This allows full-duplex operation under DMA control for both channels).

The request function may occur only when the SCC is not selected ; e.g., if the internal request becomes active while the SCC is in the middle of a read or write cycle, the external request will not become active until the cycle is complete. An active request output causes a DMA controller to initiate a read or write operation. If the request on Transmit mode is selected in either SDLC or Synchronous Mode, the Request pin is pulsed Low for one PCLK cycle at the end of CRC transmission to allow the immediate transmission of another block of data.

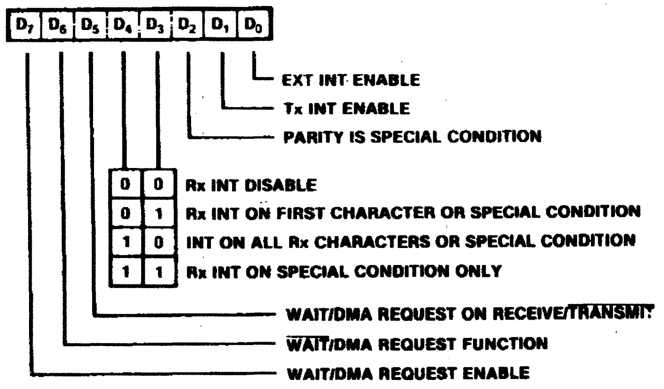
If the Wait On Receive mode, the WAIT pin is active if the CPU attempts to read SCC data that has not yet been received. In the Wait On Transmit mode, the WAIT pin is active if the CPU attempts to write data when the transmit buffer is still full. Both situations can occur frequently when block transfer instructions are used.

Bits 4 and 3 : Receive Interrupt Modes

These two bits specify the various character-available conditions that may cause interrupt requests.

Receive Interrupts Disabled (00). This mode prevents the receiver from requesting an interrupt and is normally used in a polled environment where either the status bits on RR0 or the modified vector in RR2 (Channel B) can be monitored to initiate a service routine. Although the receiver interrupts are disabled, a special condition can still provide a unique vector status in RR2.

Figure 44 : Write Register 1.


D7
D6
D5
D4
D3
D2
D1
D0
EXT INT ENABLE
Tx INT ENABLE
PARITY IS SPECIAL CONDITION
0 0 Rx INT DISABLE
0 1 Rx INT ON FIRST CHARACTER OR SPECIAL CONDITION
1 0 INT ON ALL Rx CHARACTERS OR SPECIAL CONDITION
1 1 Rx INT ON SPECIAL CONDITION ONLY
WAIT/DMA REQUEST ON RECEIVE/TRANSMIT
WAIT/DMA REQUEST FUNCTION
WAIT/DMA REQUEST ENABLE

REGISTERS DESCRIPTION (cont'd)

Receive Interrupt On First Character Or Special Condition (01). The receiver requests an interrupt in this mode on the first available character (or stored FIFO character) or on a special condition. Sync characters to be stripped from the message stream do not cause interrupts.

Special receive conditions are : receiver overrun, framing error, end of frame, or parity error (if selected). If a special receive condition occurs, the data containing the error is stored in the receive FIFO until an Error Reset command is issued by the CPU.

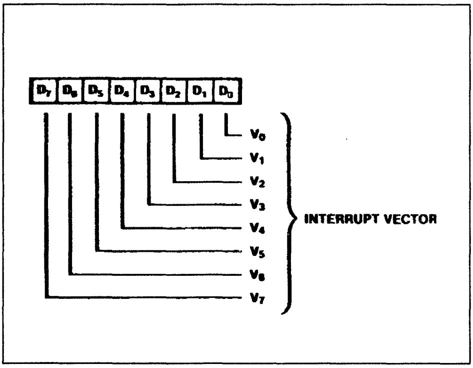
This mode is usually selected when a Block Transfer mode is used. In this interrupt mode, a pending special receive condition remains set until either an Error Reset command, a channel or hardware reset, or until receive interrupts are disabled.

The Receive Interrupt on First Character or Special Condition mode can be re-enabled by the Enable Rx Interrupt on Next Character command in WR0.

Interrupt On All Receive Characters Or Special Condition (10). This mode allows an interrupt for every character received (or character in the receive FIFO) and provides a unique vector when a special condition exists. The receiver Overrun bit and the Parity Error bit in RR1 are two special conditions that are latched. These two bits must be reset by the Error Reset command. Receiver overrun is always a special receive condition, and parity can be programmed to be a special condition.

Data characters with special receive conditions are not held in the receive FIFO in the Interrupt On All Receive Characters or Special Conditions Mode as they are in the other receive interrupt modes.

Figure 45 : Write Register 2.



Receive Interrupt On Special Condition (11). This mode allows the receiver to interrupt only on characters with a special receive condition. When an interrupt occurs, the data containing the error is held in the receive FIFO until an Error Reset command is issued. When using this mode in conjunction with a DMA, the DMA can be initialized and enabled before any characters have been received by the SCC. This eliminates the time critical section of code required in the Receive Interrupt on First Character or Special condition mode ; i.e., all data can be transferred via the DMA so that the CPU need not handle the first received character as a special case.

Bit 2 : Parity Is Special Condition

If this bit is set to "1", any received characters with parity not matching the sense programmed in WR4 give rise to a Special Receive Condition. If parity is disabled (WR4), this bit is ignored. A special condition modifies the status of the interrupt vector stored in WR2. During an interrupt acknowledge cycle, this vector can be placed on the data bus.

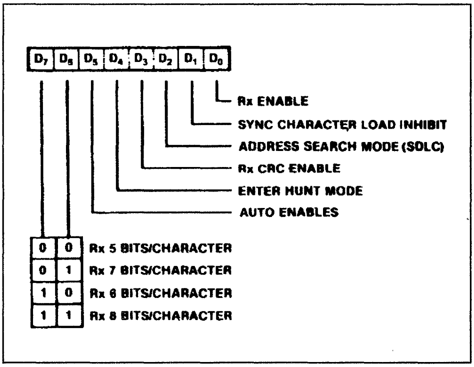
Bit 1 : Transmitter Interrupt Enable

If this bit is set to "1", the transmitter requests an interrupt whenever the transmit buffer becomes empty.

Bit 0 : External/Status Master Interrupt Enable

This bit is the master enable for External/Status interrupts including $\overline{DCD}$, $\overline{CTS}$, $\overline{SYNC}$ pins, break, abort, the beginning of CRC transmission when the Transmit/Underrun/EOM latch is set, or when the counter in the baud rate generator reaches "0". Write Register 15 contains the individual enable bits for each of these sources of External/Status interrupts. This bit is reset by a channel or hardware reset.

Figure 46 : Write Register 3.



REGISTERS DESCRIPTION (cont'd)

Write Register 2 (interrupt vector). WR2 is the interrupt vector register. Only one vector register exists in the SCC, but it can be accessed through either channel. The interrupt vector can be modified by status information. This is controlled by the Vector Includes Status (VIS) and the Status High/Status Low bits in WR9. The bit positions for WR2 are shown in figure 45.

Write Register 3 (receive parameters and control). This register contains the control bits and parameters for the receiver logic as illustrated in figure 46.

Bits 7 AND 6 : Receiver Bits/character

The state of these two bits determines the number of bits to be assembled as a character in the received serial data stream. The number of bits per character can be changed while a character is being assembled but only before the number of bits currently programmed is reached. Unused bits in the Received Data Register (RR8) are set to "1" in asynchronous modes and SDLC modes, the SCC merely transfers an 8-bit section of the serial data stream to the receive FIFO at the appropriate time. Table 9 lists the number of bits per character in the assembled character format.

Table 9 : Receive Bits/Character.

B_7	B_6	
0	0	5 Bits / Character
0	1	7 Bits / Character
1	0	6 Bits / Character
1	1	8 Bits / Character

Bit 5 : Auto Enables

This bit programs the function for both the $\overline{\text{DCD}}$ and $\overline{\text{CTS}}$ pins. $\overline{\text{CTS}}$ becomes the transmitter enable and $\overline{\text{DCD}}$ becomes the receiver enable when this bit is set to "1". However, the Receiver Enable and Transmit Enable bits must be set before the $\overline{\text{DCD}}$ and $\overline{\text{CTS}}$ pins can be used in this manner. When the Auto Enables bit is set to "0", the $\overline{\text{DCD}}$ and $\overline{\text{CTS}}$ pins are merely inputs to the corresponding status bits in Read Register 0. The state of $\overline{\text{DCD}}$ is ignored in the Local Loopback mode. The state of $\overline{\text{CTS}}$ is ignored in both Auto Echo and Local Loopback modes.

Bit 4 : Enter Hunt Mode

This command forces the comparison of sync characters or flags to assembled receive characters for the purpose of synchronization. After reset, the SCC automatically enters the Hunt mode (except asynchronous). Whenever a flag or sync character is matched, the Sync/Hunt bit in Read Register 0 is reset and, if External/Status Interrupt Enable is set, an interrupt sequence is initiated. The SCC automatically enters the Hunt mode when an abort condition is received or when the receiver is disabled.

Bit 3 : Receiver CRC Enable

This bit is used to initiate CRC calculation at the beginning of the last byte transferred from the Receiver Shift register to the receive FIFO. This operation occurs independently of the number of bytes in the receive FIFO. When a particular byte is to be excluded from CRC calculation, this bit should be reset before the next byte is transferred to the receive FIFO. If this feature is used, care must be taken to ensure that eight bits per character is selected in the receiver because of an inherent delay from the Receive Shift register to the CRC checker.

This bit is internally set to "1" in SDLC mode and the SCC calculates CRC on all bits except inserted zeros between the opening and closing character flags. This bit is ignored in asynchronous mode.

Bits 2 : Address Search Mode (SDLC)

Setting this bit in SDLC mode causes messages with addresses not matching the address programmed in WR6 to be rejected. No receiver interrupts can occur in this mode unless there is an address match. The address that the SCC attempts to match can be unique (1 in 256) or multiple (16 in 256), depending on the state of Sync Character Load Inhibit bit. The Address Search mode bit is ignored in all modes except SDLC.

Bit 1 : Sync Character Load Inhibit

If this bit is set to "1" in any synchronous mode except SDLC, the SCC compares the byte in WR6 with the byte about to be stored in the FIFO, and it inhibits this load if the bytes are equal. The SCC does not calculate the CRC on bytes stripped from the Data stream in the manner. If the 6-bit sync option is selected while in Monosync mode, the compare is still across eight bits, so WR6 must be programmed for proper operation.

If the 6-bit sync option is selected with this bit set to "1", all sync characters except the one immediately preceding the data are stripped from the message. If the 6-bit sync option is selected while in the Bisync mode, this bit is ignored.

The address recognition logic of the receiver is modified in SDLC mode if this bit is set to "1", i.e., only the four most significant bits of WR6 must match the receiver address. This procedure allows

REGISTERS DESCRIPTION (cont'd)

the SCC to receive frames from up to 16 separate sources without programming WR6 for each source (if each station address has the four most significant bits in common). The address field in the frame is still eight bits long.

This bit is ignored in SDLC mode if Address Search mode has not been selected.

Bit 0 : Receiver Enable

When this bit is set to "1", receiver operation begins. This bit should be set only after all other receiver parameters are established and the receiver is completely initialized. This bit is reset by a channel or hardware reset command, and it disables the receiver.

Write Register 4 (transmit/receiver miscellaneous parameters and modes). WR4 contains the control bits for both the receiver and the transmitter. These bits should be set in the transmit and receiver initialization routine before issuing the contents of WR1, WR3, WR6, and WR7. Bit positions for WR4 are shown in figure 47.

Bits 7 and 6 : Clock Rate 1 and 0

These bits specify the multiplier between the clock and data rates. In synchronous modes, the 1S mode is forced internally and these bits are ignored unless External Sync mode has been selected.

1X Mode (00). The clock rate and data rate are the same. In External Sync mode, this bit combination specifies that only the $\overline{\text{SYNC}}$ pin can be used to achieve character synchronization.

16X Mode (01). The clock rate is 16 times the data rate. In External Sync mode, this bit combination specifies that only the $\overline{\text{SYNC}}$ pin can be used to achieve character synchronization.

32X Mode (10). The clock rate is 32 times the data rate. In External Sync mode, this bit combination specifies that either the $\overline{\text{SYNC}}$ pin or a match with the character stored in WR7 will signal character synchronization. The sync character can be either six or eight bits long as specified by the 6-bit/8-bit Sync bit in WR10.

64X Mode (11). The clock rate is 64 times the data rate. With this bit combination in External Sync mode, both the receiver and transmitter are placed in SDLC mode. The only variation from normal SDLC operation is that the $\overline{\text{SYNC}}$ pin can be used to start or stop the reception of a frame by forcing the receiver to act as though a flag had been received.

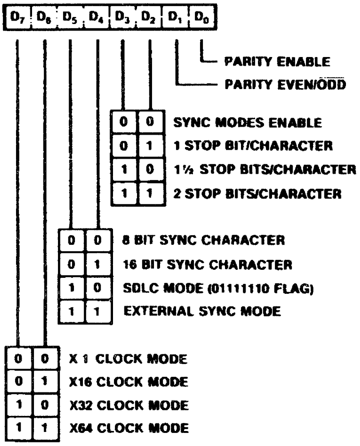
Bits 5 and 4 : SYNC Modes 1 and 0

These two bits select the various options for character synchronization. They are ignored unless synchronous modes are selected in the stop bits field of this register.

Monosync (00). In this mode, the receiver achieves character synchronization by matching the character stored in WR7 with an identical character in the received data stream. The transmitter uses the character stored in WR6 as a time fill. The sync character can be either six or eight bits, depending on the state of the 6-bit/8-bit Sync bit in WR10. If the Sync Character Load Inhibit bit is set, the receiver strips the contents of WR6 from the data stream if received within character boundaries.

Bisync (01). The concatenation of WR7 with WR6 is used for receiver synchronization and as a time fill by the transmitter. The sync character can be 12 or 16 bits in the receiver, depending on the state of the 6-bit/8-bit Sync bit in WR10. The transmitted character is always 16 bits.

Figure 47 : Write Register 4.


D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
PARITY ENABLE
PARITY EVEN/$\overline{\text{ODD}}$
0 0 SYNC MODES ENABLE
0 1 1 STOP BIT/CHARACTER
1 0 1 ½ STOP BITS/CHARACTER
1 1 2 STOP BITS/CHARACTER
0 0 8 BIT SYNC CHARACTER
0 1 16 BIT SYNC CHARACTER
1 0 SDLC MODE (01111110 FLAG)
1 1 EXTERNAL SYNC MODE
0 0 X 1 CLOCK MODE
0 1 X16 CLOCK MODE
1 0 X32 CLOCK MODE
1 1 X64 CLOCK MODE

REGISTERS DESCRIPTION (cont'd)

SDLC Mode (10). In this mode, SDLC is selected and requires a Flag (01111110) to be written to WR7. The receiver address field should be written to WR6. The SDLC CRC polynomial must also be selected (WR5) in SDLC mode.

External Sync Mode (11). In this mode, the SCC expects external logic to signal character synchronization via the $\overline{\text{SYNC}}$ pin. If the crystal oscillator option is selected (in WR11), the internal $\overline{\text{SYNC}}$ signal is forced to "0". In this mode, bits B7-B6 of this register select special version of External Sync mode. In this mode, the transmitter is in Monosync mode using the contents of WR6 as the time fill with the sync character length specified by the 6-bit/8-bit Sync bit in WR10.

Bits 3 and 2 : Stop Bits 1 and 0

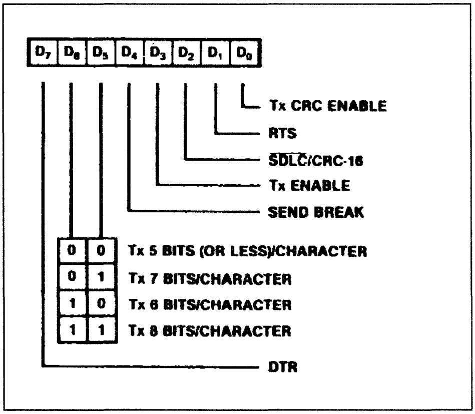
These bits determine the number of stop bits added to each asynchronous character that is transmitted. The receiver always checks for one stop bit in Asynchronous mode. A Special mode specifies that a Synchronous mode is to be selected. B2 is always set to "1" by a channel or hardware reset to ensure that the $\overline{\text{SYNC}}$ pin is in a known state after a reset.

Synchronous Modes Enable (00). This bit combination selects one of the synchronous modes specified by bits B4, B5, B6, and B7 of this register and forces the 1X Clock mode internally.

1 Stop Bit/Character (01). This bit selects Asynchronous mode with one stop bit per character.

1 1/2 Stop Bits/Character (10). These bits select Asynchronous mode with 1-1/2 stop bits per character. This mode can not be used with the 1X clock mode.

Figure 48 : Write Register 5.



2 Stop Bits/Character (11). These bits select Asynchronous mode with two stop bits per transmitted character and check for one received stop bit.

Bit 1 : Parity Even/$\overline{\textit{Odd}}$

This bit determines whether parity is checked as even or odd. A "1" programmed here selects even parity, and a "0" selects odd parity. This bit is ignored if the Parity Enable bit is not set.

Bit 0 : Parity Enable

When this bit is set, an additional bit position beyond those specified in the bits/character control is added to the transmitted data and is expected in the receive data. The Received Parity bit is transferred to the CPU as part of the data unless eight bits per character is selected in the receiver.

Write Register 5 (transmit parameter and controls). WR5 contains control bits that affect the operation of the transmitter. B2 affects both the transmitter and the receiver. Bit positions for WR5 are shown in figure 48.

Bit 7 : Data Terminal Ready

This is the control bit for the $\overline{\text{DTR}}/\overline{\text{REQ}}$ pin while the pin is in the DTR mode (selected in WR14). When set, $\overline{\text{DTR}}$ is Low ; when reset, $\overline{\text{DTR}}$ is High. This bit is ignored when $\overline{\text{DTR}}/\overline{\text{REQ}}$ is programmed to act as a $\overline{\text{REQUEST}}$ pin. This bit is reset by a channel or hardware reset.

Bits 6 and 5 : TXBits/Character 1 and 0

These bits control the number of bits in each byte transferred to the transmit buffer. Bits sent must be right justified with lease significant bits first.

The Five Or Less mode allows transmission of one to five bits per character ; however, the CPU should form at the data character as shown below in table 10. In the Six or Seven Bits/Character modes, unused data bits are ignored.

Table 10 : Tx Bits/Character 1 and 0.

Tx BITS / CHAR 1	Tx BITS / CHAR 0	
0	0	5 or less bits / character
0	1	7 bits / character
1	0	6 bits / character
1	1	8 bits / character

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	1	1	1	0	0	0	0	Sends one data bit
1	1	1	0	0	0	0	0	Sends two data bits
1	1	0	0	0	0	0	0	Sends three data bits
1	0	0	0	0	0	0	0	Sends four data bits
0	0	0	0	0	0	0	0	Sends five data bits

REGISTERS DESCRIPTION (cont'd)

Bit 4 : Send Break

When set, this bit forces the TxD output to send continuous "0s" beginning with the following transmit clock, regardless of any data being transmitted at the time. This bit functions whether or not the transmitter is enabled. When reset, TxD continues to send the contents of the Transmit Shift register, which might be syncs, data or all "1s". If this bit is set while in the X21 mode (Monosync and Loop mode selected) and character synchronization is achieved in the receiver, this bit is automatically reset and the transmitter begins sending syncs or data. This bit can also be reset by a channel or hardware reset.

Bit 3 : Transmit Enable

Data is not transmitted until this bit is set, and the TxD output sends continuous "1s" unless Auto Echo mode or SDLC Loop mode is selected. If this bit is reset after transmission started, the transmission of data or sync characters is completed. If the transmitter is disabled during the transmission of a CRC character, sync or flag characters are sent instead of CRC. This bit is reset by a channel or hardware reset.

Bit 2 : $\overline{SDLC}$/CRC-16

This bit selects the CRC polynomial used by both the transmitter and receiver. When set, the CRC-16 polynomial is used ; when reset, the SDLC polynomial is used. The SDLC/CRC polynomial must be selected when SDLC mode is selected. The CRC generator and checker can be preset to all "0s" or all "1s", depending on the state of the Preset 1/Preset 0 bit in WR10.

Bit 1 : Request to Send

This is the control bit for the RTS pin. When the RTS bit is set, the RST pin goes Low ; when reset, RTS goes High. In the Asynchronous mode with the Auto Enables bit set, RST goes High only after all bits of the character have been sent and the transmit buffer is empty. In synchronous modes of the Asynchronous mode with auto enables off, the pin directly follows the state of this bit. This bit is reset by a channel or hardware reset.

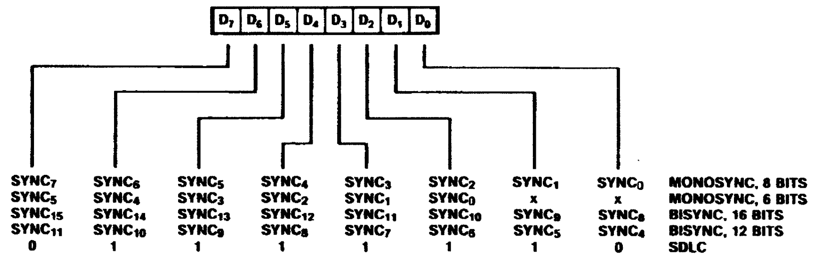
Bit 0 : Transmit CRC Enable

This bit determines whether or not CRC is calculated on a transmit character. If this bit is set at the time the character is loaded from the transmit buffer to the Transmit Shift register, CRC is calculated on that character. CRC is not automatically sent unless this bit is set when the transmit underrun exists.

Write Register 6 (sync characters or SDLC address field). WR6 is programmed to contain the transmit sync character in the Monosync mode, the first byte of a 16-bit sync character in the External Sync mode. WR6 is not used in asynchronous modes. In the SDLC modes, it is programmed to contain the secondary address field used to compare against the address field of the SDLC Frame. In SDLC mode, the SCC does not automatically transmit the station address at the beginning of the response frame. Bit positions for WR6 are shown in figure 49.

Write Register 7 (sync character or SDLC flag). WR7 is programmed to contain the receive sync character in the Monosync mode, a second byte (the last eight bits) of a 16-bit sync character in the Bisync mode, or a Flag character (01111110) in the SDLC modes. WR7 may hold the receive sync character or a flag if one of the special versions of the External Sync mode is selected. WR7 is not used in Asynchronous mode. Bit positions for WR7 are shown in figure 50.

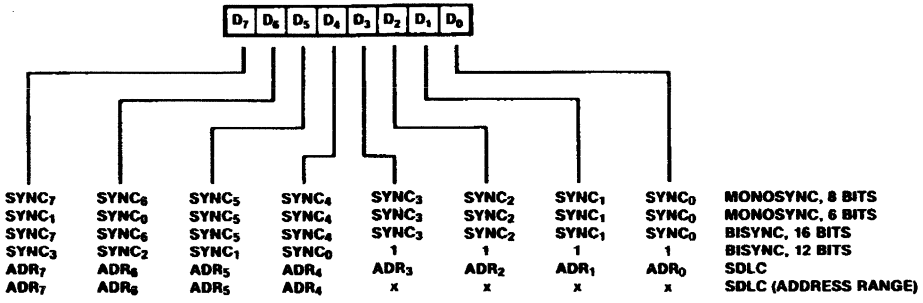
Figure 49 : Write Register 6.



SYNC7	SYNC6	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	MONOSYNC, 8 BITS
SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	x	x	MONOSYNC, 6 BITS
SYNC15	SYNC14	SYNC13	SYNC12	SYNC11	SYNC10	SYNC9	SYNC8	BISYNC, 16 BITS
SYNC11	SYNC10	SYNC9	SYNC8	SYNC7	SYNC6	SYNC5	SYNC4	BISYNC, 12 BITS
0	1	1	1	1	1	1	0	SDLC

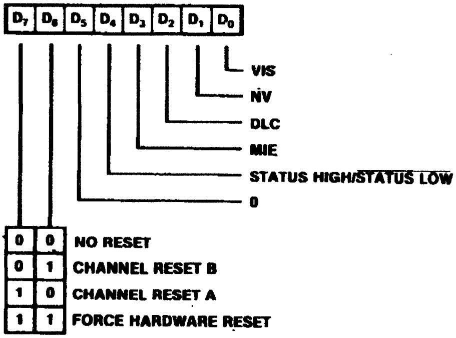
REGISTERS DESCRIPTION (cont'd)

Figure 50 : Write Register 7.



D7	D6	D5	D4	D3	D2	D1	D0	
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 8 BITS
SYNC$_1$	SYNC$_0$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 6 BITS
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	BISYNC, 16 BITS
SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	1	1	1	1	BISYNC, 12 BITS
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	ADR$_3$	ADR$_2$	ADR$_1$	ADR$_0$	SDLC
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	x	x	x	x	SDLC (ADDRESS RANGE)

Figure 51 : Write Register 9 .



D7	D6	
0	0	NO RESET
0	1	CHANNEL RESET B
1	0	CHANNEL RESET A
1	1	FORCE HARDWARE RESET

Write Register 8 (transmit buffer). WR8 is the transmit buffer register.

Write RegisteR 9 (master interrupt control). WR9 is the Master Interrupt Control register and contains the Reset command bits. Only one WR9 exists in the SCC and can be accessed from either channel. The interrupt control bits can be programmed at the same time as the Reset command because these bits are only reset by a hardware reset. Bit positions for WR9 are shown in figure 51.

Bits 7 and 6 : Reset Command Bits

Together, these bits select one of the reset commands for the SCC. Setting either of these bits to "1" disables both the receiver and the transmitter in the corresponding channel, forces TxD for that channel marking, forces the modem control signals High in that channel, resets all IPs and IUSs and disables all interrupts in that channel. Four extra PCLK cycles must be allowed beyond the usual cycle time after any of the active reset commands is issued before any additional commands or controls are written to the channel affected. Four extra PCLK cycles must be allowed beyond the usual cycle time before any additional command or controls are written to the SCC.

REGISTERS DESCRIPTION (cont'd)

No Reset (00). This command has no effect. It is used when a write to WR9 is necessary for some reason other than an SCC Reset command.

Channel Reset B (01). Issuing this command causes a channel reset to be performed on Channel B.

Channel Reset A (10). Issuing this command causes a channel reset to be performed on Channel A.

Force Hardware Reset (11). The effects of this command are identical to those of a hardware reset, except that the Shift Right/Shift Left bit is not changed and the MIE, Status High/Status Low and DLC bits take the programmed values that accompany this command.

Bit 5 : Not Used

Must be "0".

Bit 4 : Status High/$\overline{\text{Status}}$ $\overline{\text{Low}}$

This bit controls which vector bits the SCC will modify to indicate status. When set to "1", the SCC modifies bits V6, V5, and V4 according to table 11. When set to "0", the SCC modifies bits V1, V2, and V3 according to table 11. This bit controls status in both the vector returned during an interrupt acknowledge cycle and the status in RR2B. This bit is reset a hardware reset.

Table 11 : Interrupt Vector Modification.

V3	V2	V1	Status High / Status Low = 0
V4	V5	V6	Status High / Status Low = 1
0	0	0	Ch B Transmit Buffer Empty
0	0	1	Ch B External / Status Change
0	1	0	Ch B Receive Character Avail.
0	1	1	Ch B Special Receive Condition
1	0	0	Ch A Transmit Buffer Empty
1	0	1	Ch A External / Status Change
1	1	0	Ch A Receive Character Avail.
1	1	1	Ch A Special Receive Condition

Bit 3 : Master Interrupt Enable

This bit is set to 1 to globally enable interrupts, and cleared to zero to disable interrupts. Clearing this bit to zero forces the IEO pin to follow the state of the IEI pin unless there is an IUS bit set in the SCC.

No IUS bit can be set after the MIE bit is cleared to zero. This bit is reset by a hardware reset.

Bit 2 : Disable Lower Chain

The Disable Lower Chain bit can be used by the CPU to control the interrupt daisy chain. Setting this bit to "1" forces the IEO pin Low, preventing lower-priority devices on the daisy chain from requesting interrupts. This bit is reset by a hardware reset.

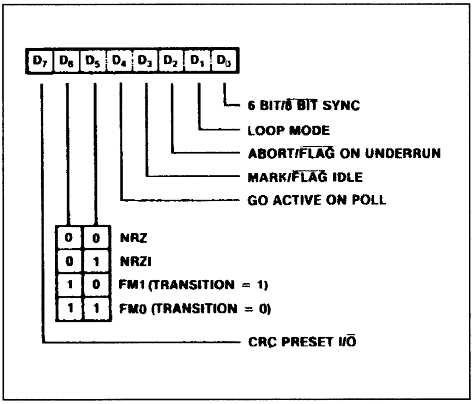
Bit 1 : No Vector

The No Vector bit controls whether or not the SCC will respond to an interrupt acknowledge cycle by placing a vector on the data bus if the SCC is the highest-priority device requesting an interrupt. If this bit is set, no vector is returned ; i.e., AD0-AD7 remain three-stated during an interrupt acknowledge cycle, even if the SCC is the highest-priority device requesting an interrupt.

Bit 0 : Vector Includes Status

The Vector Includes Status Bit controls whether or not then SCC will include status information in the vector it places on the bus in response to an interrupt acknowledge cycle. If this bit is set, the vector returned is variable, with the variable field depending on the highest-priority IP that is set. Table 11 shows the encoding of the status information. This bit is ignored if the No Vector (NV) bit is set.

Figure 52 : Write Register 10.


D7 D6 D5 D4 D3 D2 D1 D0
6 BIT/$\overline{\text{8 BIT}}$ SYNC
LOOP MODE
ABORT/$\overline{\text{FLAG}}$ ON UNDERRUN
MARK/$\overline{\text{FLAG}}$ IDLE
GO ACTIVE ON POLL
0 0 NRZ
0 1 NRZI
1 0 FM1 (TRANSITION = 1)
1 1 FM0 (TRANSITION = 0)
CRC PRESET I/$\overline{\text{O}}$

Write Register 10 (miscellaneous transmitter/receiver control bits). WR10 contains miscellaneous control bits for both the receiver and the transmitter. Bit positions for WR10 are shown in figure 52.

Bit 7 : CRC Preset I/$\overline{\text{O}}$

This bit specifies the initialized condition of the receive CRC checker and the transmit CRC generator. If this bit is set to "1", the CRC generator and checker are preset to "1". If this bit is set to "0", the CRC generator and checker are preset to "0". Either option can be selected with either CRC polynomial.

REGISTERS DESCRIPTION (cont'd)

In SDLC mode, the transmitted CRC is inverted before transmission and the received CRC is checked against the bit pattern "0001110100001111". This bit is reset by a channel or hardware reset. This bit is ignored in Asynchronous mode.

Bit 6 AND 5 : Data Encoding 1 and 2

These bits control the coding method used for both the transmitter and the receiver, as illustrated in table 12. All of the clocking options are available for all coding methods. The DPLL in the SCC is useful for recovering clocking information in NRZI and FM modes. A hardware reset forces NRZ mode. Timing for the various modes in shown in figure 53.

Table 12 : Data Encoding.

Data	Data Encoding	Encoding
0	0	NRZ
0	1	NRZI
1	0	FM1 (transition = 1)
1	1	FM0 (transition = 1)

Bit 4 : Go Active On Poll

When Loop mode is first selected during SDLC operation, the SCC connects RxD to TxD with only gate delays in the path. The SCC does not go on-loop and insert the 1-bit delay between RxD and TxD until this bit has been set and an EOP received. When the SCC is on-loop, the transmitter cannot go active unless this bit is set at the time an EOP is received. The SCC examines this bit whenever the transmitter is active in SDLC Loop mode and is sending a flag. If this bit is set at the time the flag is leaving the Transmit Shift register, another flag or data byte (if the transmit buffer is full) is transmitted. If the Go Active on Poll bit is not set at this time, the transmitter finishes sending the flag and reverts to the 1-Bit Delay mode. Thus, to transmit only one response frame, this bit should be reset after the first data byte is sent to the SCC but before CRC has been transmitted. If the bit is not reset before CRC is transmitted, extra flags are sent, slowing down response time on the loop. If this bit is reset before the first data is written, the SCC completes the transmission of the present flag and reverts to the 1-Bit Delay mode. After gaining control of the loop, the SCC is not able to transmit again until a flag and another EOP have been received. Though not strictly necessary, it is good practice to set this bit only upon receipt of a poll frame to ensure that the SCC does not go on loop without the CPU noticing it.

In synchronous modes other than SDLC with the Loop Mode bit set, this bit must be set before the transmitter can go active in response to a received sync character.

This bit is always ignored in Asynchronous mode and Synchronous modes unless the Loop Mode bit is set. This bit is reset by a channel or hardware reset.

Bit 3 : Mark/$\overline{\text{Flag}}$ Idle

This bit affects only SDLC operation and is used to control the idle line condition. If this bit is set to "0", the transmitter sends flags an idle line. If this bit is set to "1", the transmitter sends continuous "1s" after the closing flag of a frame. The idle line condition is selected byte by byte ; i.e., either a flag or eight "1s" are transmitted. The primary station in an SDLC loop should be programmed for Mark Idle to create the EOP sequence. Mark Idle must be deselected at the beginning of a frame before the first data is written to the SCC, so that an opening flag can be transmitted. This bit is ignored in Loop mode, but the programmed value takes effect upon exiting the Loop mode. This bit is reset by a channel or hardware reset.

Bit 2 : Abort/$\overline{\text{FLag}}$ On Underrun

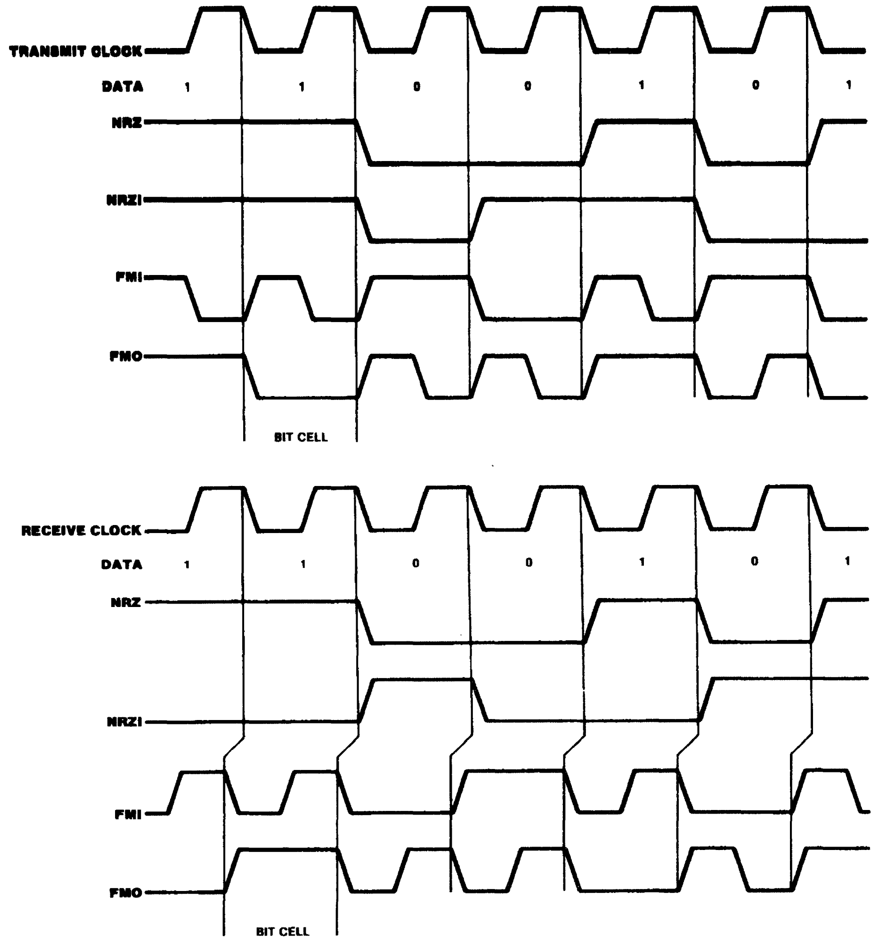
This bit affects only SDLC operation and is used to control how the SCC responds to a transmit underrun condition. If this bit is set to "1" and a transmit underrun occurs, the SCC sends an abort and a flag instead of CRC. If this bit is reset, the SCC sends CRC on a transmit underrun. At the beginning of this 16-bit transmission, the Transmit Underrun/EOM bit is set, causing an External/Status interrupt. The CPU uses this status, along with the byte count from memory or the DMA, to determine whether the frame must be retransmitted. A transmit buffer Empty interrupt occurs at the end of this 16-bit transmission to start the next frame. If both this bit and the Mark/Flag Idle bit are set to "1", all "1s" are transmitted after the transmit underrun. This bit should be set after the first byte of data is sent to the SCC and reset immediately after the last byte of data so that the frame will be terminated properly with CRC and a flag. This bit is ignored in Loop mode, but the programmed value is active upon exiting Loop mode. This bit is reset by a channel or hardware reset.

Bit 1 : Loop Mode

In SDLC mode, the initial set condition of this bit forces the SCC to connect TxD to TxD and to begin searching the incoming data stream so that it can go on loop. All bits pertinent to SDLC mode operation in other registers must be set before this mode

REGISTERS DESCRIPTION (cont'd)

Figure 53 : NRZ(NRZI)FM1(FM0) Timing.



REGISTERS DESCRIPTION (cont'd)

is selected. The transmitter and receiver should not be enabled until after this mode has been selected. As soon as the Go Active Onn Poll bit is set and an EOP is received the SCC goes on loop. If this bit is reset after the SCC is on loop, the SCC waits for the next EOP to go off loop.

In synchronous modes, the SCC uses this bit, along with the Go Active On Poll bit, to synchronize the transmitter to the receiver. The receiver should not be enabled until after this mode is selected. The TxD pin is held marking when this mode is selected unless a break condition is programmed. The receiver waits for a sync character to be received and then enables the transmitter on a character boundary. The break condition, if programmed, is removed. This mode works properly with sync characters of 6, 8, or 16 bits. This bit is ignored in Asynchronous mode and is reset by a channel or hardware reset.

Bit 0 : 6 Bit/8 Bit Sync

This bit is used to select a special case of synchronous modes. If this bit is set to "1" in Monosync mode, the receiver and transmitter sync characters are six bits long instead of the usual eight. If this bit is set to "1" in Bisync mode, the received sync will be 12 bits and the transmitter sync character will remain 16 bits long. This bit is ignored in SDLC and Asynchronous modes but still has effect in the special external sync modes. This bit is reset by a channel or hardware reset.

Write Register 11 (clock mode control). WR11 is the Clock Mode Control register. The bits in this register control the sources of the both the receive and transmit clocks, the type of signal on the SYNC and RTxC pins, and the direction of the TRxC pin. Bit positions for WR11 are shown in figure 54.

Bit 7 : RTxC-XTAL/NO XTAL

This bit controls the type of input signal the SCC expects to see on the RTxC pin. If this bit is set to "0", the SCC expects a TTL-compatible signal as an input to this pin. If this bit is set to "1", the SCC connects a high-gain amplifier between the RTxC and SYNC pins in expectation of a quartz crystal being placed across the pins.

The output of this oscillator is available for use as a clocking source. In this mode of operation, the SYNC pin is unavailable for other use. The SYNC-signal is forced to "0" internally. A hardware reset forces NO XTAL. (At least 20 ms should be allowed after this bit is set to allow the oscillator to stabilize).

Bits 6 AND 5 : Receiver Clock 1 and 0

These bits determine the source of the receive clock as shown in table 13. They do not interfere with any of the modes of operation in the SCC but simply control a multiplexer just before the internal receive clock input. A hardware reset forces the receive clock to come from the TRxC pin.

Table 13 : Receive Clock Source.

Receive Clock 1	Receive Clock 0	
0	0	Receive Clock = RTxC Pin
0	1	Receive Clock = TRxC Pin
1	0	Receive Clock = BR Output
1	1	Receive Clock = DPLL Output

Bits 4 and 3 : Transmit Clock 1 and 0

These bits determine the source of the transmit clock as shown in table 14. They do not interfere with any of the modes of operation of the SCC but simply control a multiplexer just before the internal transmit clock input. The DPLL output that may be used to feed the transmitter in FM modes lags by 90 the output of the DPLL used by the receiver. This makes the received and transmitted bit cells occur simultaneously, neglecting delays. A hardware reset selects the TRxC pin as the source of the transmit clocks.

Table 14 : Transmit Clock Source.

Transmit Clock 1	Transmit Clock 0	
0	0	Transmit Clock = RTxC Pin
0	1	Transmit Clock = TRxC Pin
1	0	Transmit Clock = BR Output
1	1	Transmit Clock = DPLL Output

Bit 2 : TRxC O/I

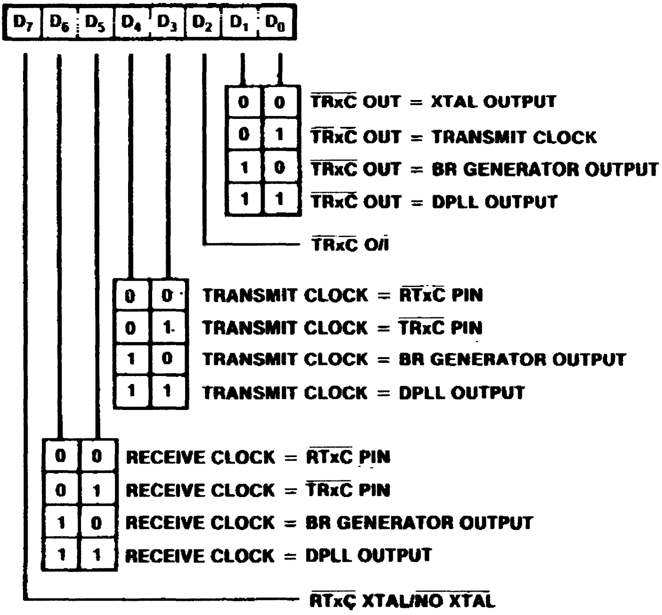
This bit determines the direction of the TRxC pin. If this bit is set to "1", the TRxC pin is an output and carries the signal selected by D1 and D0 of this register. However, if either the receive or the transmit clock is programmed to come from the TRxC pin, TRxC will be an input, regardless of the state of this bit. The TRxC pin is also an input if this bit is set to "0". A hardware reset forces this bit to "0".

Bits 1 and 0 : TRxC Output Source 1 and 0

These bits determine the signal to be echoed out of the SCC via the TRxC pin. No signal is produced if TRxC has been programmed as the source of either

REGISTERS DESCRIPTION (cont'd)

Figure 54 : Write Register 11.


D7 D6 D5 D4 D3 D2 D1 D0
0 0 TRxC OUT = XTAL OUTPUT
0 1 TRxC OUT = TRANSMIT CLOCK
1 0 TRxC OUT = BR GENERATOR OUTPUT
1 1 TRxC OUT = DPLL OUTPUT
TRxC O/I
0 0 TRANSMIT CLOCK = RTxC PIN
0 1 TRANSMIT CLOCK = TRxC PIN
1 0 TRANSMIT CLOCK = BR GENERATOR OUTPUT
1 1 TRANSMIT CLOCK = DPLL OUTPUT
0 0 RECEIVE CLOCK = RTxC PIN
0 1 RECEIVE CLOCK = TRxC PIN
1 0 RECEIVE CLOCK = BR GENERATOR OUTPUT
1 1 RECEIVE CLOCK = DPLL OUTPUT
RTxC XTAL/NO XTAL

the receive or the transmit clock. If TRxC O/I (bit 2) is set to "0", these bits are ignored.

If the XTAL oscillator output is programmed to be echoed, and the Xtal oscillator has not been enabled, the TRxC pin goes High. The DPLL signal that is echoed is the DPLL signal used by the receiver. Hardware reset selects the XTAL oscillator as the output source.

Table 15 : Transmit External Control Selection.

Output Signal	Output Signal	
0	0	TRxC = XTAL Oscillator Output
0	1	TRxC = Transmit Clock
1	0	TRxC = B R Output
1	1	TRxC = DPLL Output (Receive)

Write Register 12 (lower byte of baud rate generator time constant). WR12 contains the lower byte of the time constant for the baud rate generator. The time constant can be changed at any time, but the new value does not take effect until the next time the time constant is loaded into the down counter. No attempt is made to synchronize the loading of the time constant into WR12 and WR13 with the clock driving the down counter. For this reason, it is advisable to disable the baud rate generator while the new time constant is loaded into WR12 and WR13. Ordinarily, this is done anyway to prevent a load of the down counter between the writing of the upper and lower bytes of the time constant.

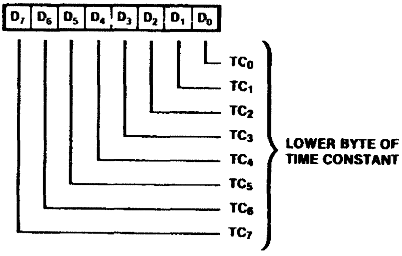
The formula for determining the appropriate time constant for a given baud is shown below with the

REGISTERS DESCRIPTION (cont'd)

desired rate in bits per second and the BR clock period in seconds. This formula is derived because the counter decrements from N down to "0" - plus-one-cycle for reloading the time constant and is then fed to a toggle flip-flop to make the output a square wave. Bit positions for WR12 are shown in figure 55.

Time constant = [1/2 desired rate . BR clock period] - 2

Figure 55 : Write Register 12.



Write Register 13 (upper byte of baud rate generator time constant). WR13 contains the upper byte of the time constant for the baud rate generator. Bit positions for WR13 are shown in figure 56.

Write Register 14 (miscellaneous control bits). WR14 contains some miscellaneous control bits. Bit positions for WR14 are shown in figure 57.

Bits 7 and 5 : Digital Phase-locked Loop Command Bits

These three bits encode the eight commands for the Digital Phase-Locked Loop. A channel or hardware reset disables the DPLL, resets the mission clock latches, sets the source to the RTxC pin and selects NRZI mode. The Enter Search Mode command enables the DPLL after a reset.

Null Command (000). This command has no effect on the DPLL.

Enter Search Mode (001). Issuing this command causes the DPLL to enter the Search mode, where the DPLL searches for a locking edge in the incoming data stream. The action taken by the DPLL upon receipt of this command depends on the operating mode of the DPLL.

Figure 56 : Write Register 13.

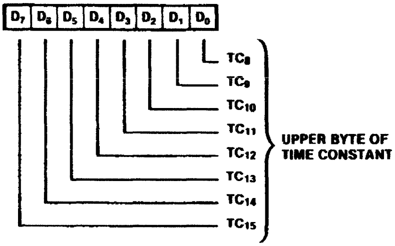
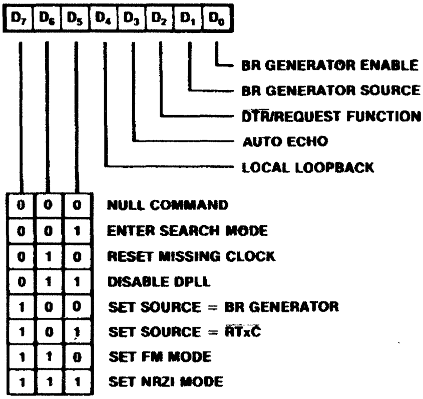


Figure 57 : Write Register 14.



REGISTERS DESCRIPTION (cont'd)

In NRZI mode, the output of the DPLL is High while the DPLL is waiting for an edge in the incoming data stream. After the Search mode is entered, the first edge the DPLL sees is assumed to be a valid data edge, and the DPLL begins the clock recovery operation from that point. The DPLL clock rate must be 32 times the data rate in NRZI mode. Upon leaving the Search mode, the first sampling edge of the DPLL occurs 16 of these 32X clocks after the first data edge and the second sampling edge occurs 48 of these 32X clocks after the first data edge. Beyond this point, the DPLL begins normal operation, adjusting the output to remain in sync with the incoming data.

In FM mode, the output of the DPLL is Low while the DPLL is waiting for an edge in the incoming data stream. The first edge the DPLL detects is assumed to be a valid clock edge. For this to be the case, the line must contain only clock edges ; i.e., with FM1 encoding, the line must be continuous "0s". With FM0 encoding the line must be continuous "1s", whereas Manchester encoding requires alternating "1s" and "0s" on the line. The DPLL clock rate must be 16 times the data rate in FM mode. The DPLL output causes the receiver to sample the data stream in the nominal center of the two halves of the bit cell to decide whether the data was a "1" or a "0". After this command is issued, as in NRZI mode, the DPLL starts sampling immediately after the first edge is detected. (In FM mode, the DPLL examines the clock edge of every other bit cell to decide what correction must be made to remain in sync). If the DPLL does not see an edge during the expected window, the one clock mission bit in RR10 is set. If the DPLL does not see an edge after two successive attempts, the two clocks missing bit in RR10 is set and the DPLL automatically enters the Search mode. This command resets both clock missing latches.

Reset Clock Missing (010). Issuing this command disables the DPLL, resets the clock missing latches in RR10, and forces a continuous Search mode state.

Disable DPLL (001). Issuing this command disables the DPLL, resets the clock missing latches in RR10, and forces a continuous Search mode state.

Set Source = BR Gen (100). Issuing this command forces the clock for the DPLL to come from the output of the baud rate generator.

Set Source = $\overline{\text{RTxC}}$ (101). Issuing the command forces the clock for the DPll to come from the RTxC pin or the crystal oscillator, depending on the state of the XTAL/no XTAL bit in WR11. This mode is selected by a channel or hardware reset.

Set FM Mode (110). This command forces the DPLL to operate in the FM mode and is used to recover the clock from FM or Manchester-encoded data. (Manchester is decoded by placing the receiver in NRZ mode while the DPLL is in FM mode).

Set NRZI Mode (111). Issuing this command forces the DPLL to operate in the NRZI mode. This mode is also selected by a hardware or channel reset.

Bit 4 : Local Loopback

Setting this bit to "1" selects the local Loopback mode of operation. In this mode, the internal transmitted data is routed back to the receiver, as well as to the TxD pin. The $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$ inputs are ignored as enables in Local Loopback mode, even if auto enables is selected. (If so programmed, transitions on these inputs still cause interrupts). This mode works with any Transmit/Receive mode except Loop mode. For meaning-ful results, the frequency of the transmit and receive clocks must be the same. This bit is reset by a channel or hardware reset.

Bit 3 : Auto Enable

Setting this bit to "1" selects the Auto Enable mode of operation. In this mode, the TxD pin is connected to RxD, as in Local Loopback mode, but the receiver still listens to the RxD input. Transmitted data is never seen inside or outside the SCC in this mode, and $\overline{\text{CTS}}$ is ignored as a transmit enable. This bit is reset by a channel or hardware reset.

Bit 2 : DTR/Request Function

This bit selects the function of the $\overline{\text{DTR/REQ}}$ pin follows the state of the DTR bit in WR5. If this bit is set to "1", the $\overline{\text{DTR/REQ}}$ pin follows the state of the DTR bit in WR5. If this bit is set to "1", the $\overline{\text{DTR/REQ}}$ pin goes Low whenever the transmit buffer becomes empty and in any of the synchronous mode when CRC has been sent at the end of a message. The request function on the $\overline{\text{DTR/REQ}}$ pin differs somewhat from the transmit request function available on the $\overline{\text{W/REQ}}$ pin in that $\overline{\text{REQUEST}}$ does not go inactive until the internal operation satisfying the request is complete, which occurs four to five $\overline{\text{PCLK}}$ cycles after the rising edge of $\overline{\text{DS}}$, $\overline{\text{READ}}$ or $\overline{\text{WRITE}}$. If the DMA used is edge-triggered, this difference is unimportant. This bit is reset by a channel or hardware reset.

Bit 1 : Baud Rate Generator Source

This bit selects the source of the clock for the baud rate generator. If this bit is set to "0", the baud rate

REGISTERS DESCRIPTION (cont'd)

generator clock comes from either the $\overline{\text{RTxC}}$ pin or the XTAL oscillator (depending on the state of the XTAL/no XTAL bit). If this bit is set to "1", the clock for the baud rate generator is the SCC's PCLK input. Hardware reset sets this bit to "0", selecting the $\overline{\text{RTxC}}$ pin as the clock source for the baud rate generator.

Bit 0 : Baud Rate Generator Enable

This bit controls the operation of the baud rate generator. The counter in the baud rate generator is enabled for counting when this bit is set to "1", and counting is inhibited when this bit is set to "0". When this bit is set to "1", change in the state of this bit is not reflected by the output of the baud rate generator for two counts of the counter. This allows the command to be synchronized. However, when set to "0", disabling is immediate. This bit is reset by a hardware reset.

Write Register 15 (external/status interrupt control). WR15 is the External/Status Source Control register. If the External/Status interrupts are enabled as a group via WR1, bits in this register control which External/Status conditions can cause an interrupt. Only the External/Status conditions that occur after the controlling bit are sent to "1" will cause an interrupt. This is true even if an External/Status condition is pending at the time the bit is set. Bit positions for WR15 are shown in figure 58.

Bit 7 : Break/Abort IE

If this bit is set to "1", a change in the Break/Abort status of the receiver causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 6 : Tx Underrun/EOM

If this bit is set to "1", a change of state by the Tx Underrun/EOM latch in the transmitter causes an External/Status interrupt. This bit is set to "1" a channel or hardware reset.

Bit 5 : CTS IE

If this bit is set to "1", a change of state on the $\overline{\text{CTS}}$ pin causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 4 : SYNC/Hunt IE

If this bit is set to "1", a change of state on the $\overline{\text{SYNC}}$ pin causes an External/Status interrupt in Asynchronous mode, and a change of state in the Hunt bit in the receiver causes and External/Status interrupt in synchronous modes. This bit is set by a channel or hardware reset.

Bit 3 : DCD IE

If this bit is set to "1", a change of state on the $\overline{\text{DCD}}$ pin causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 2 : Not Used

Must be "0".

Bit 1 : Zero Count IE

If this bit is set to "1", an External/Status interrupt is generated whenever the counter in the baud rate generator reaches "0". This bit is set to "0" by a channel or hardware reset.

Bit 0 : Not Used

Must be "0".

READ REGISTERS

The MK85C30 SCC contains seven read registers in each channel. In addition there are two registers which are shared by both channels. The status of these registers is continually changing and depends on the mode of communication, received and transmitted data, and the manner in which this data is transferred to and from the CPU. The following description details the bit assignments for each register.

Read Register 0 (transmit/receiver buffer status and external status). Read Register 0 contains the status of the receive and transmit buffers. RR0 also contains the status bits for the six sources of External/Status interrupts. The bit configuration is illustrated in figure 59.

Bit 7 : Break/Abort

In the Asynchronous mode, this bit is set when a Break sequence (null character plus framing error) is detected in the receive data stream. This bit is reset when the sequence is terminated, leaving a single null character in the receive FIFO. This character should be read and discarded. In SDLC mode, this bit is set by the detection of an Abort sequence (seven or more "1s"), then reset automatically at the termination of the Abort sequence. In either case, if the Break/Abort IE bit is set, an External/Status interrupt is initiated. Unlike the remainder of the External/Status bits, both transitions are guaranteed to

REGISTERS DESCRIPTION (cont'd)

cause an External/Status interrupt, even if another External/Status interrupt is pending at the time these transitions occur. This procedure is necessary because Abort or Break conditions may not persist.

Bit 6 : TX Underrun/EOM

This bit is set by a channel or hardware reset and when the transmitter is disabled or a Send Abort command is issued. This bit can only be reset by the reset Tx Underrun/EOM Latch command in WR0. When the Transmit Underrun occurs, this bit is set and causes an External/Status interrupt (if the Tx Underrun/EOM IE bit is set).

Only the 0-to-1 transition of this bit causes an interrupt. This bit is always "1" in Asynchronous mode, unless a reset Tx Underrun/EOM Latch command has been erroneously issued. In this case, the Send Abort command can be used to set the bit to one and at the same time cause an External/Status interrupt.

Bit 5 : Clear to Send

If the CTS IE bit in WR15 is set, this bit indicates the state of the $\overline{\text{CTS}}$ pin the last time any of the enabled External/Status bits changed. Any transition on the $\overline{\text{CTS}}$ pin while no interrupt is pending latches the state of the $\overline{\text{CTS}}$ pin and generates an External/Status interrupt. Any odd number of transitions on the $\overline{\text{CTS}}$ pin while another External/Status interrupt is pending also causes an External/Status interrupts condition. If the CTS IE bit is reset, it merely reports the current unlatched state of the $\overline{\text{CTS}}$ pin.

Bit 4 : SYNC/Hunt

The operation of this bit is similar to that of the $\overline{\text{CTS}}$ bit, except that the condition monitored by the bit varies depending on the mode in which the SCC is operating.

When the XTAL oscillator option is selected in asynchronous modes, this bit is forced to "0" (no External/Status interrupt is generated). Selecting the XTAL oscillator in synchronous or SDLC modes had no effect on the operation of this bit.

The XTAL oscillator should not be selected in External Sync mode.

In Asynchronous mode, the operation of this bit is identical to that of the CTS status bit, except that this bit reports the state of the SYNC pin.

In External sync mode the $\overline{\text{SYNC}}$ pin is used by external logic to signal character synchronization. When the Enter Hunt Mode command is issued in External Sync mode, the $\overline{\text{SYNC}}$ pin must be held High by the external sync logic until character synchronization is achieved. A High on the $\overline{\text{SYNC}}$ pin holds the Sync/Hunt bit in the reset condition.

When external synchronization is achieved, $\overline{\text{SYNC}}$ must be driven Low on the second rising edge of the Receive Clock after the last rising edge of the Receive Clock on which the last bit of the receive character was received. Only $\overline{\text{SYNC}}$ is forced Low, it is good practice to keep it Low until the CPU informs the external sync logic that synchronization has been lost or that a new message is about to start. Both transitions on the $\overline{\text{SYNC}}$ pin cause External/Status interrupts if the Sync/Hunt IE bit is set to "1".

The Enter Hunt Mode command should be issued whenever character synchronization is lost. At the same time, the CPU should inform the external logic

Figure 58 : Write Register 15.

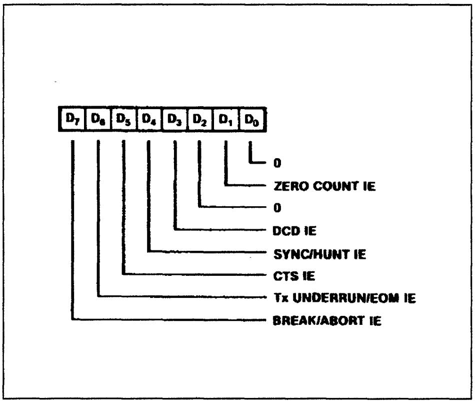
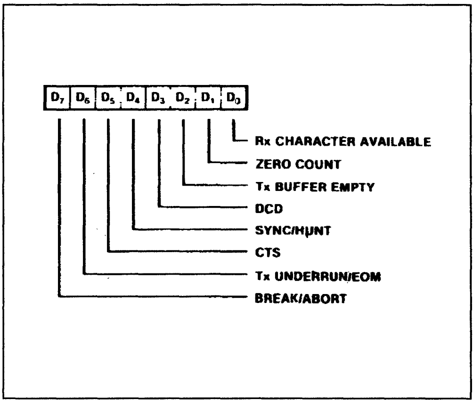


Figure 59 : Read Register 0.



REGISTERS DESCRIPTION (cont'd)

that character synchronization has been lost and that the SCC is waiting for $\overline{\text{SYNC}}$ to become active.

In the Monosync and Bisync Receive modes, the Sync/Hunt status bit is initially set to "1" by the Enter Hunt Mode command. The Sync/Hunt bit is reset when the SCC established character synchronization. Both transitions cause External/Status interrupts if the Sync/Hunt IE bit is set. When the CPU detects the end of message or the loss of character synchronization, the Enter Hunt Mode command should be issued to set the Sync/Hunt bit and cause an External/Status interrupt. In this mode, the $\overline{\text{SYNC}}$ pin is an output, which goes Low every time a sync pattern is detected in the data stream.

In the SDLC modes, the Sync/Hunt bit is initially set by the Enter Hunt Mode command or when the receiver is disabled. It is reset when the opening flag of the first frame is detected by the SCC. An External/Status interrupt is also generated if the Sync/Hunt IE bit is set. Unlike the Monosync and Bisync modes, once the Sync/Hunt bit is reset in SDLC mode, it does not need to be set when the end of the frame is detected. The SCC automatically maintains synchronization. The only way the Sync/Hunt bit can be set again is by the Enter Hunt Mode command or by disabling the receiver.

Bit 3 : Data Carrier Detect

If the DCD bit in WR15 is set, this bit indicates the state of the $\overline{\text{DCD}}$ pin the last time the Enabled External/Status bits changed. Any transition on the $\overline{\text{DCD}}$ pin while no interrupt is pending latches the state of the $\overline{\text{DCD}}$ pin, and generates an External/Status interrupt. Any odd number of transitions on the $\overline{\text{DCD}}$ pin while another External/Status interrupt is pending also causes an External/Status interrupt condition. If the DCD IE is reset, this bit merely reports the current, unlatched state of the $\overline{\text{DCD}}$ pin.

Bit 2 : TX Buffer Empty

This bit is set to "1" when the transmit buffer is empty. It is reset while CRC is sent in a synchronous or SDLC mode and while the transmit buffer is full. The bit is reset when a character is loaded into the transmit buffer. This bit is always in the set condition after a hardware or channel reset.

Bit 1 : Zero Count

If the Zero Count Interrupt Enable bit is set in WR15, this bit is set to one while the counter in the baud rate generator is at the count of zero. If there is no other External/Status interrupt condition pending at the time this bit is set, an External/Status interrupt is generated. However, if there is another External/Status interrupt pending at this time, no interrupt is initiated until interrupt service is complete. If the Zero Count condition does not persit beyond the end of the interrupt service routine, no interrupt will be generated. This bit is not latched High, even thought the other External/Status latches close as a result of the Low-to-High transition on ZC. The interrupt service routine should check the other External/Status conditions for changes. If none changed, ZC was the source. In polled applications, check the IP bit in RR3A for a status change and then proceed as in the interrupt service routine.

Bit 0 : RX Character Available

This bit is set to "1" when at least one character is available in the receive FIFO and is reset when the receive FIFO is complety empty. A channel or hardware reset empties the receive FIFO.

Read Register 1. RR1 contains the Special Receive Condition status bits and the residue codes for the I-field in SDLC mode. Figure 60 shows the bit positions for RR1.

***Bit 7 : End of Frame** (SDLC)*

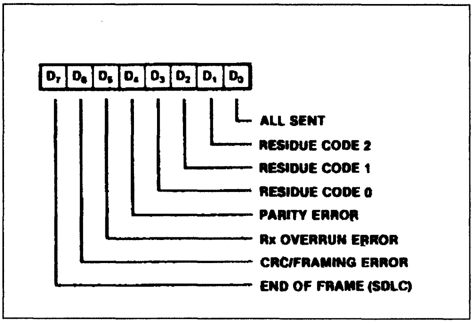
This bit is used only in SDLC mode and indicates that a valid closing flag has been received and that the CRC Error bit and residue codes are valid. This bit can be reset by issuing the Error Reset command. It is also updated by the first character of the following frame. This bit is reset in any mode other than SDLC.

Bit 6 : CRC/Framing Error

If a framing error occurs (in Asynchronous mode), this bit is set (and not latched) for the receive character in which the framing error occurred. Detection of a framing error adds an additional one-half bit to the character time so that the framing error is not interpreted as a new Start bit. In Synchronous and SDLC modes, this bit indicates the result of comparing the CRC checker to the appropriate check value. This bit is reset by issuing an Error Reset command, but the bit is never latched. Therefore, it is always updated when the next character is received. When used for CRC error status in Synchronous or SDLC modes, this bit is usually set since most bit combination, except for a correctly completed message, result in a non-zero CRC.

REGISTERS DESCRIPTION (cont'd)

Figure 60 : Read Register 1.



Bit 5 : Receiver Overrun Error

This bit indicates that the receive FIFO has overflowed. Only the character that has been written over is flagged with this error, and when the character is read, the Error condition is latched until reset by the Error Reset command. The overrun character and all subsequent characters received until the Error Reset command is issued causes a Special Receive Condition vector to be returned.

Bit 4 : Parity Error

When parity is enabled, this bit is set for the characters whose parity does not match the programmed sense (even/odd). This bit is latched so that once an error occurs, it remains set until the Error Reset command is issued. If the parity in Special Condition bit is set, a parity error causes a Special Receive Condition vector to be returned on the character containing the error and on all subsequent characters until the Error Reset command is issued.

Bit 3, 2, and 1 : Residue Codes 2, 1, and 0

In those cases in SDLC mode where the received I-Field is not an integral multiple of the character length, these three bits indicate the length of the I-Field and are meaningful only for the transfer in which the end of frame bit is set. This field is set to "011" by a channel or hardware reset and is forced to this state in Asynchronous mode. These three bits can leave this state only if SDLC is selected and a character is received. The codes signify the following (Reference table 16 when a receive character length is eight bits per character).

I-Field bits are right-justified in all cases. If a receive character length other than eight bits is used for the I-Field, a table similar to table 16 can be constructed for each different character length. Table 17 shows the residue codes for no residue (The I-Field boundary lies on a character boundary).

Table 16 : I-Field Bit Selection (8 Bits only).

Residue	Residue	Residue	I-Field Bits in Last Byte	I-Field Bits in Previous Byte
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

Bit 0 : All Sent

In Asynchronous mode, this bit is set when all characters have completely cleared the transmitter pins. Most modems contain additional delays in the data path, which requires the modem control signals to remain active until after the data has cleared both the transmitter and the modem. This bit is always set in synchronous and SDLC modes.

Table 17 : Residue Bits/Character.

Bits / Char	Residue	Residue	Residue
8	0	1	1
7	0	0	0
6	0	1	0
5	0	0	1

Read Register 2. RR2 contains the interrupt vector written into WR2. When the register is accessed in Channel A, the vector returned is the vector actually stored in WR2. When this register is accessed in Channel B, the vector returned includes status information in bits 1, 2, and 3, or in bits 6, 5, and 4, depending on the state of the Status High/Status Low bit in WR9 and independent of the state of the VIS bit in WR9. The vector is modified according to table 11 shown in the explanation of the VIS bit in WR9. If no interrupts are pending, the status is V3, V2, V1 = 011, or V6, V5, V4 = 110. Figure 61 shows the bit positions for RR2.

Read Register 3. RR3 is the Interrupts Pending register. The status of each of the Interrupt Pending bits in the SCC is reported in this register. This register exists only in Channel A. If this register is accessed in Channel B, all "0s" are returned. The two unused bits are always returned as "0". Figure 62 shows the bit positions for RR3.

REGISTERS DESCRIPTION (cont'd)

Read Register 8. RR8 is the Receive Data register.

Read Register 10. RR10 contains some miscellaneous status bits. Unused bits are always "0". Bit positions for RR10 are shown in figure 63.

Bit 7 : One Clock Missing

While operating in the FM mode, the DPLL sets this bit to "1" when it does not see a clock edge on the incoming lines in the window where it expects one. This bit is latched until reset by a Reset Missing Clock or Enter Search Mode command in WR14. In the NRZI mode of operation and while the DPLL is disabled, this bit is always "0".

Bit 6 : Two Clocks Missing

While operating in the FM mode, the DPLL sets this bit to "1" when it does not see a clock edge in two successive tries. At the same time the DPLL enters the Search mode. This bit is latched until reset by a Reset Missing Clock or Enter Search Mode command in WR10. In the NRZI mode of operation and while the DPLL is disabled, this bit is always "0".

Bit 4 : Loop Sending

This bit is set to "1" in SDLC Loop mode while the transmitter is in control of the Loop, that is, while the SCC is actively transmitting on the loop. This bit is reset at all other times.

This bit can be polled in SDLC mode to determin when the closing flag has been sent.

Bit 1 : On Loop

This bit is set to "1" while the SCC is actually onloop in SDLC Loop mode. This bit is set to "1" in the X.21 mode (Loop mode selected while in monosync) when the transmitter goes active. This bit is "0" at all other times. This bit can also be pulled in SDLC mode to determine when the closing flag has been sent.

Read Register 12. RR12 returns the value stored in WR12, the lower byte of the time constant for the baud rate generator. Figure 64 shows the bit positions for RR12.

Read Register 13. RR13 returns the value stored in WR13, the upper byte of the time constant for the baud rate generator. Figure 65 shows the bit positions for RR13.

Read Register 15. RR15 reflects the value stored in WR15, the External/Status IE bits. The two unused bits are always returned as "0s". Figure 66 shows the bits positions for RR15.

Figure 61 : Write Register 2.

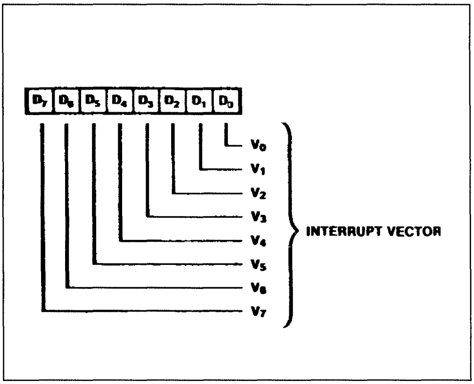
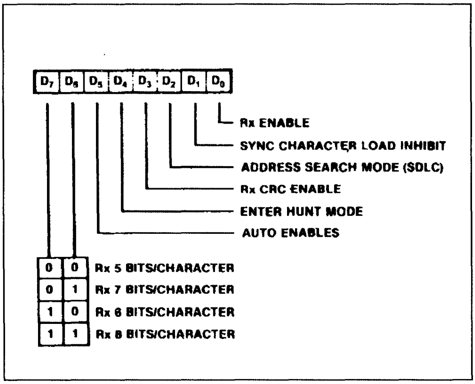


Figure 62 : Write Register 3.



REGISTERS DESCRIPTION (cont'd)

Figure 63 : Write Register 10.

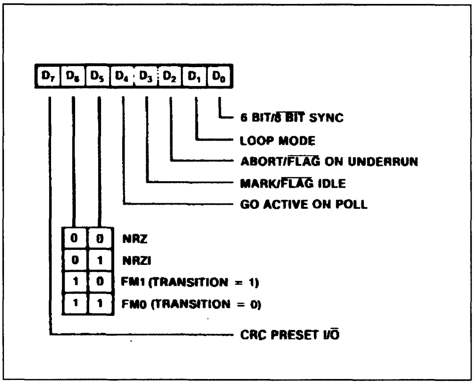


Figure 65 : Write Register 13.

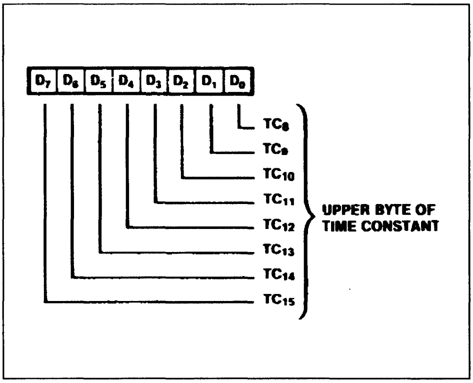


Figure 64 : Write Register 12.

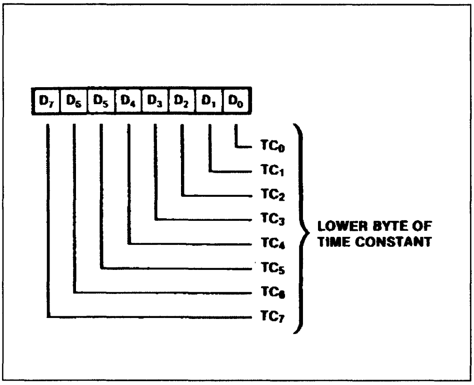
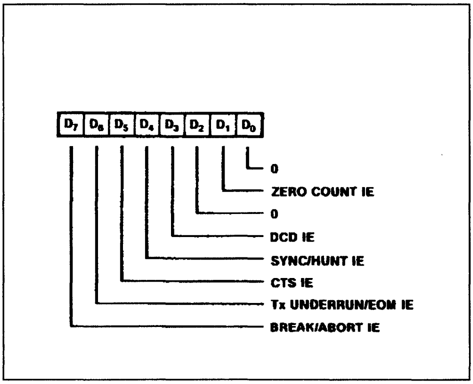


Figure 66 : Write Register 15.



SCC INITIALIZATION WORKSHEET

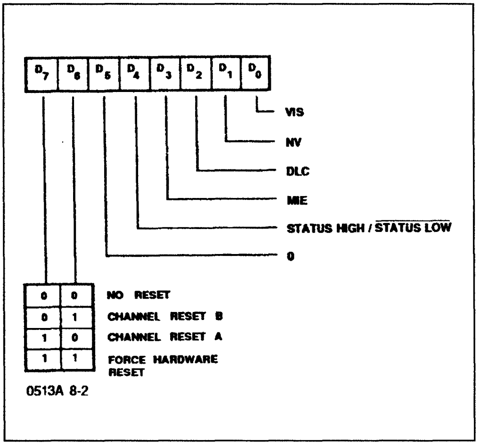
This section describes the software initialization procedure for the Serial Communications Controller (SCC).

Figure 68 provides a worksheet that can be used as an aid when initializing the SCC. Since all SCC operation modes are initialized in a similar manner, the worksheet can be used to tailor the SCC device to the user's individual need. Specific examples are given in the following sections.

Register Overview

Each of the SCC's two channels has its own separate Write registers that are programmed to initialize different operating modes. There are two types of bits in the Write registers : Command bits and Mode bits. An example of a register that contains both types of bits is Write Register 9 (WR9), and is shown in figure 67.

Figure 67 : Write Register 2.



WR9 is the Master Interrupt Control register and contains the Reset command bits. Command bits are denoted by having boxes drawn around them in register diagrams. Bit D5 in this register is not used in this register and must be 0 at all times.

The Command bits, D7 and D6, select one of the reset commands for the SCC. Setting either of these bits to 1 disables both the receiver and the transmitter in the corresponding channel, forces TxD for the channel marking, forces the modem control signals High in that channel, resets all IPs and IUSs, and disables all interrupts in that channel. Functions controlled by the Command bits can only be enabled or disabled, they cannot be toggled.

Bits D4-D0 are Mode bits that can be enabled or disabled either by being set to "1" or reset to "0". Each Mode bit affects only one function. For example, Bit D1 is the No Vector mode bit, it controls whether or not the SCC will respond to an interrupt acknowledge cycle by placing a vector on the data bus. If this bit is set, no vector is returned. In command bits entry, each new command requires a separate rewrite of the entire register. Care must be taken when issuing a command, so that the Mode bits are not changed accidentally.

Initialization Procedure

The SCC initialization procedure is divided into three parts. The first part consists of programming the operation modes (e.g. bits-per-character, parity) and loading the constants (e.g., interrupt vector, time constants). The second part enables the hardware functions (e.g., transmitter, receiver, baud-rate generator). It is important that the operating modes are programmed before the hardware functions are enabled. The third part, if required, consists of enabling the different interrupts.

Table 18 shows the order (from top to bottom) in which the SCC registers are to be programmed. Those registers that need not be programmed are listed as optional in the comments column. The bits in the registers that are marked with an "X" are to be programmed by the user. The bits marked with an "S" are to be set to their previous programmed value. For example, in part 2, Write Register 3, bits D1-D7 are shown with an "S" because they have been programmed in part 1 and must remain set to the same value.

Initialization Table Generation

Figure 68 is a worksheet for the initialization of the SCC. All the bits that must be programmed as either a "0" or a "1" are already filled in ; the remaining bits are left blank and are to be programmed by the user according to the desired mode of operation. The binary value can then be converted to a hexadecimal number and placed in the table, following the Write register notation in the column labeled "HEX". A Program Initialization table is produced when this worksheet is completed.

SCC INITIALIZATION WORKSHEET (cont'd)

Figure 68 : SCC Initialization Worksheet.

Label of SCC Table: ______________________ SCC Base Address ______________________

Description: ____

	REGISTER	HEX		7	6	5	4	3	2	1	0	COMMENTS
MODES	WR9	C	0	1	1	0	0	0	0	0	0	SOFTWARE RESET
	WR0	0		0	0	0	0	0	0			
	WR4											
	WR1			0			0	0		0	0	
	WR2											
	WR3										0	
	WR5							0				
	WR6											
	WR7											
	WR9			0	0	0		0				
	WR10											
	WR11											
	WR12											
	WR13											
	WR14										0	
	WR14										0	
ENABLES	WR14			0	0	0					1	
	WR3										1	
	WR5							1				
	WR0	8	0	1	0	0	0	0	0	1	0	RESET TxCRC
	WR1											
INTERRUPT	WR15											
	WR0	1	0	0	0	0	1	0	0	0	0	RESET Ext/STATUS
	WR0	1	0	0	0	0	1	0	0	0	0	RESET Ext/STATUS
	WR1											
	WR9			0	0	0						

SCC INITIALIZATION WORKSHEET (cont'd)

Table 18 : SCC Initialization Order.

Part 1. Modes and Constants		
WR9	1100000	Hardware Reset
WR0	000000XX	Select Shift Mode (8030 only)
WR4	XXXXXXXX	Tx/Rx Con, Async or Sync Mode
WR1	0XX00X00	Select W/REQ (opt)
WR2	XXXXXXXX	Program Interrupt Vector (opt)
WR3	XXXXXXX0	Select Rx Control
WR5	XXXX0XXX	Selects Tx Control
WR6	XXXXXXXX	Program Sync Character (opt)
WR7	XXXXXXXX	Program Sync Character (opt)
WR9	000X0XXX	Select Interrupt Control
WR10	XXXXXXXX	Miscellaneous Control (opt)
WR11	XXXXXXXX	Clock Control
WR12	XXXXXXXX	Time Constant Lower Byte (opt)
WR13	XXXXXXXX	Time Constant Upper Byte (opt)
WR14	XXXXXXX0	Miscellaneous Control
WR14	XXXSSSSS	Commands (opt)

Part 2. Enables		
WR14	000SSSS1	Baud Rate Enable
WR3	SSSSSSS1	Rx Enable
WR5	SSSS1SSS	Tx Enable
WR0	10000000	Reset Tx CRG (opt)
WR1	XSS00S00	DMA Enable (opt)
Part 3. Interrupt Status		
WR15	XXXXXXXX	Enable External/status
WR0	00010000	Reset External Status
WR0	00010000	Reset External Status Twice
WR1	SSSXXSXX	Enable Rx, Tx and Ext/status
WR9	000SXSSS	Enable Master Interrupt Enable
1 = Set to one		X = User defined
0 = Reset to zero		S = Same as previously prog.

Reset Conditions

Prior the initialization, the SCC should be reset by either hardware or software. A hardware reset can be accomplished by simultaneously grounding. A software reset can be executed by writing a 0H to Write Register 9.

POLLED ASYNCHRONOUS MODE

This section describes the use of the SCC in polled Asynchronous Mode. The device can be set with 5 to 8 bits per character, 1, 1.5, or 2 stop bits, and a wide range of baud rates. In this particular example, 8 bits per character, 2 stop bits and 9600 baud rate are used. An external 2.4576MHz, crystal oscillator is used for baud-rate generation. The SCC can be programmed for local loopback for on-board diagnostics. The user can make use of this feature to test-program the part without additional hardware to simulate an actual transmit and receive environment.

SCC Interface

Figure 69 shows the SCC to CPU interface required for this application. The 8-bit data bus and control lines all come from the user's CPU. The 8530 control lines are $\overline{RD}$, $\overline{WR}$, A/$\overline{B}$, D/$\overline{C}$ and $\overline{CE}$. PCLK comes from the system clock, or an external crystal, up to the maximum rate of the SCC. The IEI and the $\overline{INTACK}$ pins should be pulled up. The baud-rate generator clock is connected to the $\overline{RTxC}$ pin.

SCC Initialization

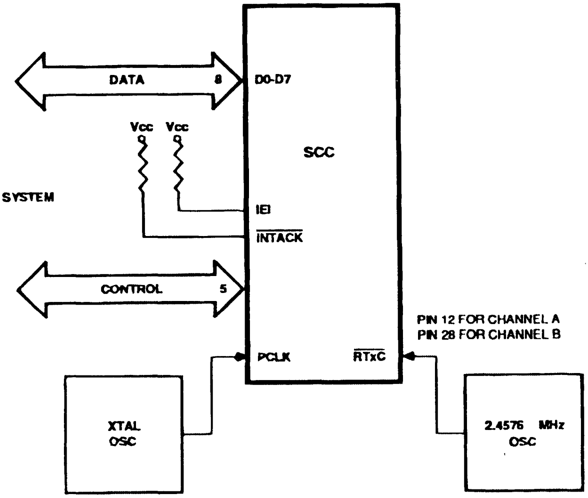
Initialization of the SCC for polled asynchronous communication is divided into two parts ; part one programs the operating modes of the SCC and part two enables them. Care must be taken when writing the software to meet the SCC's Cycle and Reset Recovery times. The Cycle Recovery time, 6 PCLK cycles, applies to the period between any Read or Write cycles affecting the SCC. The Reset Recovery time is the period after a hardware reset caused either by hardware or software ; this recovery time extends the Cycle Recovery time to 11 PCLK cycles. For more details about these recovery times, see the section Interfacing the SCC.

Table 19 : Polled Asynchronous Initialization Procedure.

Register	Value	Comments
WR9	C0H	Force Hardware Reset
WR4	4CH	x16 Clock, 2 Stop Bits, no Parity
WR3	C0H	Rx8 Bits, Rx Disabled
WR5	60H	Tx8 Bits, DTR, RTS, Tx off
WR9	00H	Int. Disabled
WR10	00H	NRZ
WR11	56H	Tx & Rx = BRG out, TRxC = BRG out
WR12	06H	Time Constant = 6
WR13	00H	Time Constant High = 0
WR14	10H	BRG in = RTxC, BRG off, Loopback
Enables		
WR14	11H	BRG Enable
WR3	C1H	Rx Enable
WR5	68H	Tx Enable

POLLED ASYNCHRONOUS MODE (cont'd)

Figure 69 : SCC to CPU Interface in Polled Asynchronous Mode.



SCC Operating Mode Programming

WR9 resets the SCC to a known state by writing a C0 hex. The command, Force Hardware Reset, is identical to a hardware reset.

WR4 selects the asynchronous, x 16 mode, with 2 stop bits and no parity. The x 16 mode means that clock rate is 16 times the data rate.

WR3 selects 8 bits per character and does not enable the receiver. The 8 bits per character allows 8 bits to be assembled from the data stream. The receiver is not enabled at this time because the SCC has not been initialized.

WR5 selects 8 bits per character and does not enable the transmitter. The 8 bits per character allows 8 bits to be sent, as data, with the least significant bit first. The transmitter is not enabled at this time because the SCC has not been initialized.

WR9 selects that there are no interrupts enabled. This inhibits the SCC from requesting an interrupt from the CPU.

WR10 selects NRZ encoding. This NRZ coding is used on the transmitter as well as the receiver.

WR11 selects the $\overline{RTxC}$ pin to TTL clock ; the baud-rate generator is the transmit and receive clocks source, and the TRxC pin as a baud-rate generator output.

WR12 & WR13 and select the baud-rate generator's time constant. The WR13 time constant is determined by the equation :

$$\text{Time Constant} = \frac{\text{Clock Frequency}}{2 \times \text{Baud Rate} \times \text{clock mode}} - 2$$

In this example, the clock frequency is 2.4576MHz, the baud rate is 9600, and the clock mode is 16, the time constant is, therefore, 6 ; expressed as a 16-bit, hexadecimal number, it is 0006H. The time constant LOW (WR12) is, therefore, 06H and the time constant HIGH (WR13) is 00H. The baud rate for this example can be varied, as long as the data rate is less than 1/4 of the PCLK rate.

Table 20 shows the time constants for other common baud rates.

Table 20 : Time Constants for Common Baud Rates.

Baud Rate	Divider	
	Decimal	Hex
38400	0	0000H
19200	2	0002H
9600	6	0006H
4800	14	000EH
2400	30	001EH
1200	62	003EH
600	126	007EH
300	254	00FEH
150	510	01FEH
For 2.4576 MHz Clock, X 16 Mode		

POLLED ASYNCHRONOUS MODE (cont'd)

WR14 selects the baud-rate generator as the RTxC pin, baud-rate generator disabled, and internal loop-back. The baud-rate generator uses the RTxC pin as the clock source and is not enabled at this time because the SCC initialization is not complete.

SCC Operating Mode Enables

WR14 enables the baud-rate generator. Bit 0 (LSB) is changed to a 1 to enable the baud-rate generator ; all other bits must maintain the value selected during initialization.

WR3 enables the receiver. Bit 0 (LSB) is changed to a 1 to enable the receiver, all other bits must maintain the value selected during initialization.

WR5 enables the transmitter. Bit 3 is changed to a 1 to enable the transmitter, all other bits must maintain the value selected during initialization.

Transmit and Receive Routines

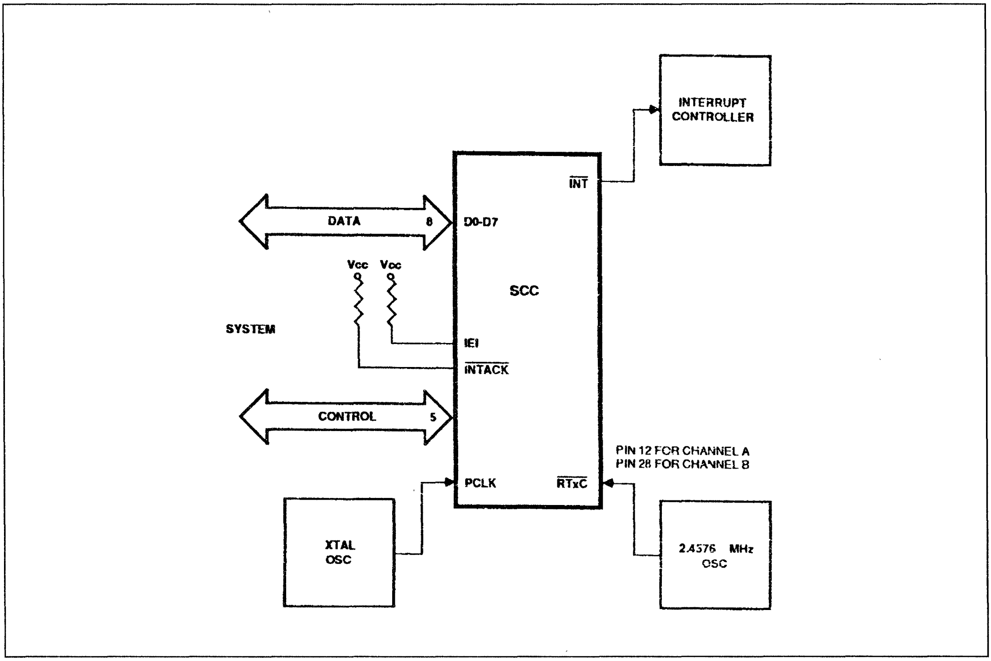
After initialization, and after all enables have been selected, the SCC is ready for communication.

The transmitter buffer and the receive FIFO are empty. The example shown below is coded to transmit and receive characters.

Figure 70 : Transmit and Receive Routine.

```
; Transmit a character
TXCHAR : INPUT    RRO      ;Read RRO
         Test     Bit 2    ;Test transmit
                            buffer empty
         JZ       TXCHAR   ;Loop if not
                            empty
         OUTPUT   CHAR     ;Output character
                            to data port
         RET               Return
; Receive a character
RXCHAR : INPUT    RRO      ;Read RRO
         TEST     BIT 0    ;Test Receive
                            buffer
         JZ       RXCHAR   ;Loop if not full
         INPUT    CHAR     ;Input character
                            from data port
         RET               ;Return
```

Figure 71 : SCC to CPU Interface in Interrupt without INTACK Asynchronous Mode.



INTERRUPT WITHOUT INTACK ASYNCHRONOUS MODE

This section describes the use of the SCC for interrupt-driven Asynchronous Mode. As with the example in the previous chapter, the SCC is set with 8 bits per character, 2 stop bits, at 9600 baud rate. An external 2.4576MHz, crystal oscillator is used for baud-rate generation. Interrupt acknowledge is not generated because of the extra hardware required to produce this signal. In this chapter, the SCC is also programmed for local loopback so that no external loop between the transmit and the receive data lines is needed for on-board diagnostics. This feature allows the user to test-program the part without additional hardware to simulate an actual transmit and receive environment.

SCC Interface

Figure 71 shows the SCC to CPU interface required for this application. The 8-bit data bus and control lines all come from the user's CPU. For the MK85C30, the control lines are $\overline{RD}$, $\overline{WR}$, A/B, D/$\overline{C}$ and $\overline{CE}$. The $\overline{INT}$ signal goes to an interrupt controller which must produce the interrupt vector to the CPU. The PCLK comes from the system clock, or an external crystal oscillator, up to the maximum rate of the SCC. The IEI and the $\overline{INTACK}$ pins should be pulled up. The baud-rate generator clock is connected to the $\overline{RTxC}$ pin.

SCC Initialization

The initialization of the SCC for interrupt-driven asynchronous communication is divided into three parts. Part one programs the operating modes of the SCC, part two and three enable them. Care must be taken when writing the code to meet the SCC's Cycle and Reset Recovery times. The Cycle Recovery time applies to the period between any Read or Write cycles to the SCC, and is 6PCLK cycles. The Reset Recovery time applies to a hardware reset caused either by hardware or software ; this recovery time extends the Cycle Recovery time to 11PCLK cycles. More details about these recovery times can be found in the section Interfacing the SCC.

Table 21.

Register	Value	Comments
WR9	C0H	Force Hardware Reset
WR4	4CH	x16 Clock, 2 Stop Bits, no Parity
WR2	00H	Interrupt Vector 00WR3
	C0H	Rx8 Bits, Rx Disabled
WR5	60H	Tx8 Bits, DTR, RTS, Tx off
WR9	00H	Int. Disabled WR10
	00H	NRZ
WR11	56H	Tx & Rx = BRG out, TRxC = BRG out
WR12	06H	Time Constant = 6
WR13	00H	Time Constant High = 0
WR14	10H	BRG in = RTxC, BRG off, Loopback
Enables		
WR14	11H	BRG Enable
WR3	C1H	Rx Enable
WR5	68H	Tx Enable
Enable Interrupts		
WR1	12H	Rx Int on All Char and Tx Int Enables
WR9	08H	MIE

SCC OPERATING MODES PROGRAMMING

WR9 resets the SCC to a known state by writing a C0 hex. This command, Force Hardware Reset, is identical to a hardware reset.

WR4 selects asynchronous mode, x 16 mode, 2 stop bits and no parity. The x 16 mode means that the clock rate is 16 times the data rate.

WR2 is the interrupt vector of the SCC. Even though a vector is not placed on the bus in this mode the vector including status in read from RR2. By writing 00H to this register the status read will be the only bits set in RR2.

WR3 selects 8 bits per character and does not enable the receiver. The 8 bits per character allows 8 bits to be assembled from the data stream. The receiver is not enabled at this time because the SCC is not completely initialized.

INTERRUPT WITHOUT INTACK ASYNCHRONOUS MODE (cont'd)

WR5 selects 8 bits per character and does not enable the transmitter. The 8 bits per character allows 8 bits to be sent as data with the least significant bit first. The transmitter is not enabled at this time because the SCC is not completely initialized.

WR9 selects that there are no interrupts enabled. This will inhibit the SCC from requesting an interrupt from the CPU.

WR10 selects NRZ encoding. This selects NRZ coding is to be used on the transmitter and the receiver.

WR11 selects the RTxC pin to TTL clock, the transmit and receive clocks source as the baud-rate generator and the TRxC pin as a baud-rate generator output.

WR12 & WR13 select the baud-rate generators time constant. The time constant is determined by the equation :

$$\text{Time Constant} = \frac{\text{Clock Frequency}}{2 \times \text{Baud Rate} \times \text{clock mode}} - 2$$

In this example, the clock frequency is 2.4576MHz, the baud rate is 9600, and the clock mode is 16 ; the time constant is 6. Converting this time constant to a 16-bit hexadecimal number, it becomes 0006H. The time constant LOW (WR12) is 06H and the time constant HIGH (WR13) is 00H. The baud rate for this example can be varied for as long as the data rate is less than 1/4 of the PCLK rate. Table 22 gives the time constants for other common baud rates.

Table 22 : Time Constants for Common Baud Rates.

Baud Rate	Divider	
	Decimal	Hex
38400	0	0000H
19200	2	0002H
9600	6	0006H
4800	14	000EH
2400	30	001EH
1200	62	003EH
600	126	007EH
300	254	00FEH
150	510	01FEH
For 2.4576 MHz Clock, X 16 Mode		

WR14 selects the baud rate source as the $\overline{\text{RTxC}}$ pin, baud rate generator disabled, and internal loopback. The baud-rate generator will use the $\overline{\text{RTxC}}$ pin as the clock source for the baud-rate generator. The baud-rate generator is not enabled at this time because the SCC initialization is not complete.

SCC Operating Mode Enables

WR14 enables the baud-rate generator. Bit 0 (LSB) is changed to a 1 to enable the baud-rate generator ; all other bits must maintain the value selected during initialization.

WR3 enables the receiver. Bit 0 (LSB) is changed to a 1 to enable the receiver ; all other bits must maintain the value selected during initialization.

WR5 enables the transmitter. Bit 3 is changed to a 1 to enable the transmitter ; all other bits must maintain the value selected during initialization.

SCC Operating Mode Interrupts

WR1 enables the Tx and the Rx interrupts. The Rx interrupt is programmed to generate an interrupt an all received characters or special conditions. This provides an interrupt on every character received by the SCC. The external/status interrupts are not enabled in this application.

WR9 sets the master interrupt enable (MIE) bit 3. Seting this bit enables the interrupts pending to generate and interrupt on the INT pin.

Interrupt Routine

When the SCC has been initialized and enabled, it is ready for communication. The transmitter buffer and the receive FIFO are both empty. An interrupt will not be generated until the software writes the first character to the transmit buffer. Once the first character is in the SCC shift register, the first transmit interrupt will occur. The SCC will then keep setting transmit and receive interrupts to the interrupt controller until the end of the message. At the end of the message, a Reset Transmitter Interrupt Pending (WR0) is issued to clear the transmit interrupt. After the last character is read into the SCC, the interrupts will cease until another message is written into the transmitter.

Once an interrupt is received and the interrupt controller vectors to the interrupt routine, RR2 is read from channel B. The value read from RR2 is the vector, including status. This vector shows the status of the highest priority interrupt pending (IP) at the time it is read. Once the highest priority interrupt condition is cleared, RR2 will show the status of the next highest interrupt pending, if one is present. This allows multiple interrupts to be serviced without the overhead of the interrupt acknowledge cycle of the interrupt controller.

The following example shows how the interrupt routine should be coded.

HARDWARE INFORMATION

ABSOLUTE MAXIMUM RATINGS

Parameter	Test Conditions	Unit
Operating Temperature MK85C30X–XX00 MK85C30X–XX10 MK85C30X–XX11	0 to + 70 – 40 to + 85 – 55 to + 125	°C °C °C
Storage Temperature	– 65 to + 150	°C
Voltages on any Pin with Respect to GND	– 0.3 to + 7	V
Total Power Dissipation	700	mW
Oscillator Frequency MK85C30X–08 MK85C30X–10 MK85C30X–12	0 to 8 0 to 10 0 to 12	MHz

Note : Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the reference pin.

Standard conditions are as follows :

- $+4.75V \leq V_{CC} \leq 5.25V$
- GND = 0V
- $0^{\circ}C \leq T_A \leq +70^{\circ}C$

All ac parameters assume a load capacitance of 50pf max.

Figure 73 : Standard Test Load.

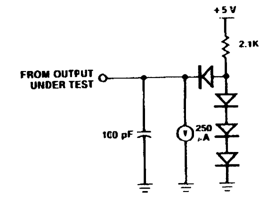
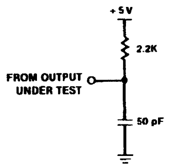


Figure 74 : Open-Drain Test Load.



DC CHARACTERISTICS

Symbol	Parameter	Test Conditions	Min.	Max.	Unit
V_{IH}	Input High Voltage		2.0	V_{CC}+0.3	V
V_{IL}	Input Low Voltage		– 0.3	0.8	V
V_{OH}	Output High Voltage	I_{OH} = – 250 μA	2.4		V
V_{OL}	Output Low Voltage	I_{OL} = + 2.0 mA		0.4	V
I_{IL}	Input Leakage	$0.4 \leq V_{IN} \leq +2.4$ V		± 10.0	μA
I_{OL}	Output Leakage	$0.4 \leq V_{OUT} \leq +2.4$ V		± 10.0	μA
I_{CC}	V_{CC} Supply Current			30	mA

V_{CC} = 5V ± 5% unless otherwise specified, over specified temperature range.

HARDWARE INFORMATION (cont'd)

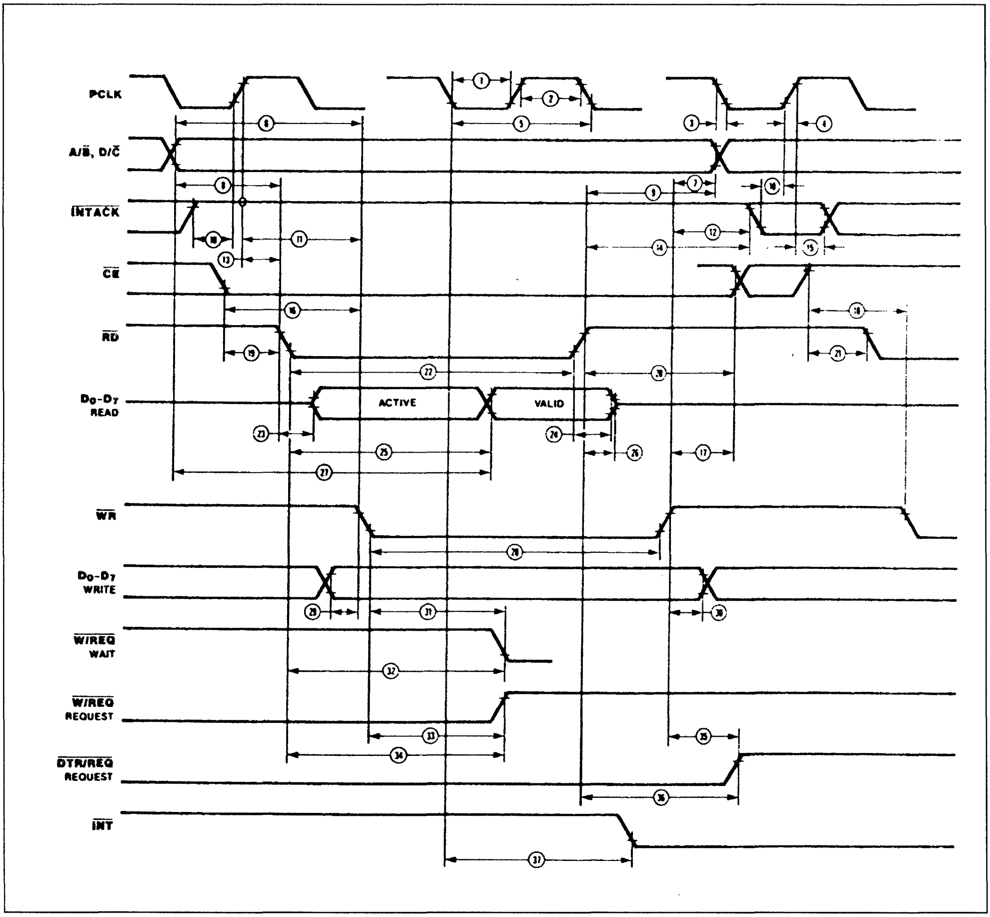
Capacitance

Symbol	Parameter	Test Conditions	Min.	Max.	Unit
C_{IN}	Input Capacitance			10	pf
C_{OUT}	Output Capacitance			15	pf
$C_{I/O}$	Bidirectional Capacitance			20	pf

f = 1MHz, over specified temperature range.
Unmeasured pins returned to ground.

AC Timing Characteristics

Figure 75 : Read and Write Timing.



HARDWARE INFORMATION (cont'd)

Table 23 : AC Timing Characteristics.

N°	Symbol	Parameter	8MHz		10MHz		12MHz		Notes (3)
			Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	
1	TwPC1	PCLK Low Width	50	1000	40	1000	34	1000	
2	TwPCh	PCLK High Width	50	1000	40	1000	34	1000	
3	TfPC	PCLK Fall Time		10		8		8	
4	TrPC	PCLK Rise Time		10		8		8	
5	TcPC	PCLK Cycle Time	125	2000	100	2000	83.3	2000	
6	TsA(WR)	Address to $\overline{\text{WR}}$ ↓ Setup Time	70		60		45		
7	ThA(WR)	Address to $\overline{\text{WR}}$ ↑ Hold Time	0		0		0		
8	TsA(RD)	Address to $\overline{\text{RD}}$ ↓ Setup Time	70		60		45		
9	ThA(RD)	Address to $\overline{\text{RD}}$ ↑ Hold Time	0		0		0		
10	TsIA(PC)	$\overline{\text{INTACK}}$ to PCLK ↑ Setup Time	10		10		10		
11	TsIAi(WR)	$\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ ↓ Setup Time	145		125		100		1
12	ThIA(WR)	$\overline{\text{INTACK}}$ to $\overline{\text{WR}}$ ↑ Hold Time	0		0		0		
13	TsIAi(RD)	$\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↓ Setup Time	145		125		100		1
14	ThIA(RD)	$\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↑ Hold Time	0		0		0		
15	ThIA(PC)	$\overline{\text{INTACK}}$ to PCLK ↑ Hold Time	85		65		40		
16	TsCE1(WR)	$\overline{\text{CE}}$ Low to $\overline{\text{WR}}$ ↓ Setup Time	0		0		0		
17	ThCE(WR)	$\overline{\text{CE}}$ to $\overline{\text{WR}}$ ↑ Hold Time	0		0		0		
18	TsCEh(WR)	$\overline{\text{CE}}$ High to $\overline{\text{WR}}$ ↓ Setup Time	60		50		40		
19	TsCE1(RD)	$\overline{\text{CE}}$ Low to $\overline{\text{RD}}$ ↓ Setup Time	0		0		0		1
20	ThCE(RD)	$\overline{\text{CE}}$ to $\overline{\text{RD}}$ ↑ Hold Time	0		0		0		1
21	TsCEh(RD)	$\overline{\text{CE}}$ High to $\overline{\text{RD}}$ ↓ Setup Time	60		50		40		1
22	TwRD1	$\overline{\text{RD}}$ Low Width	150		120		100		1
23	TdRD(DRA)	$\overline{\text{RD}}$ ↓ to Read Data Active Delay	0		0		0		
24	TdRDr(DR)	$\overline{\text{RD}}$ ↑ to Read Data Not Valid Delay	0		0		0		
25	TdRDf(DR)	$\overline{\text{RD}}$ ↓ to Read Data Valid Delay		140		120		100	
26	TdRD(DRz)	$\overline{\text{RD}}$ ↑ to Read Data Float Delay		40		25		20	2

Notes :
1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a ± 0.5V change in the output with a maximum dc load and minimum ac load.
3. Timings are preliminary and subject to change.

HARDWARE INFORMATION (cont'd)

Figure 76 : Interrupt Acknowledge Timing.

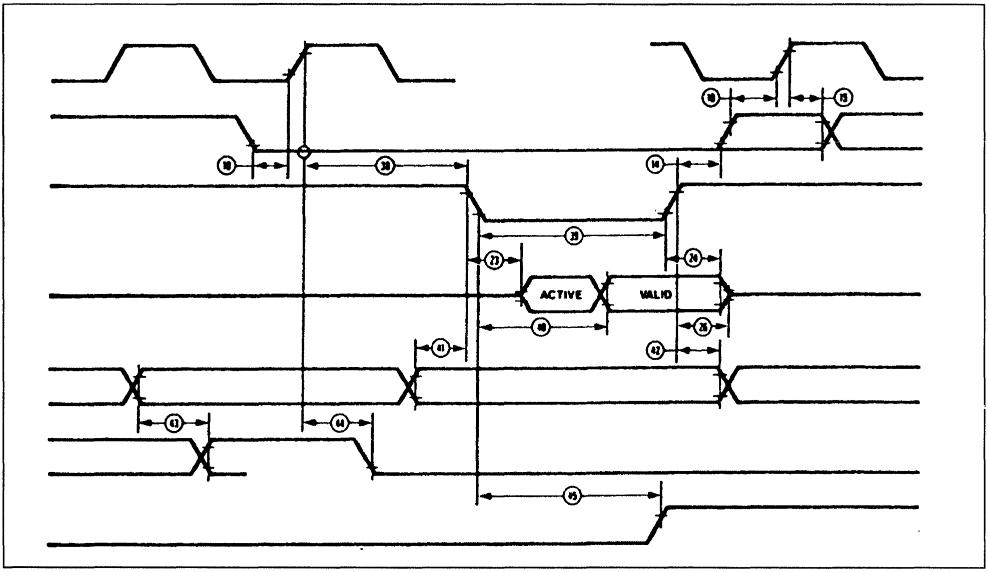


Figure 77 : Reset Timing.

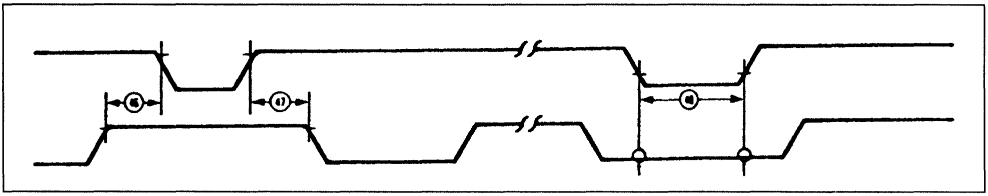
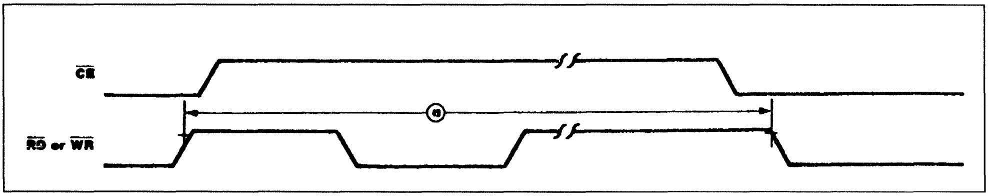


Figure 78 : Cycle Timing.



HARDWARE INFORMATION (cont'd)

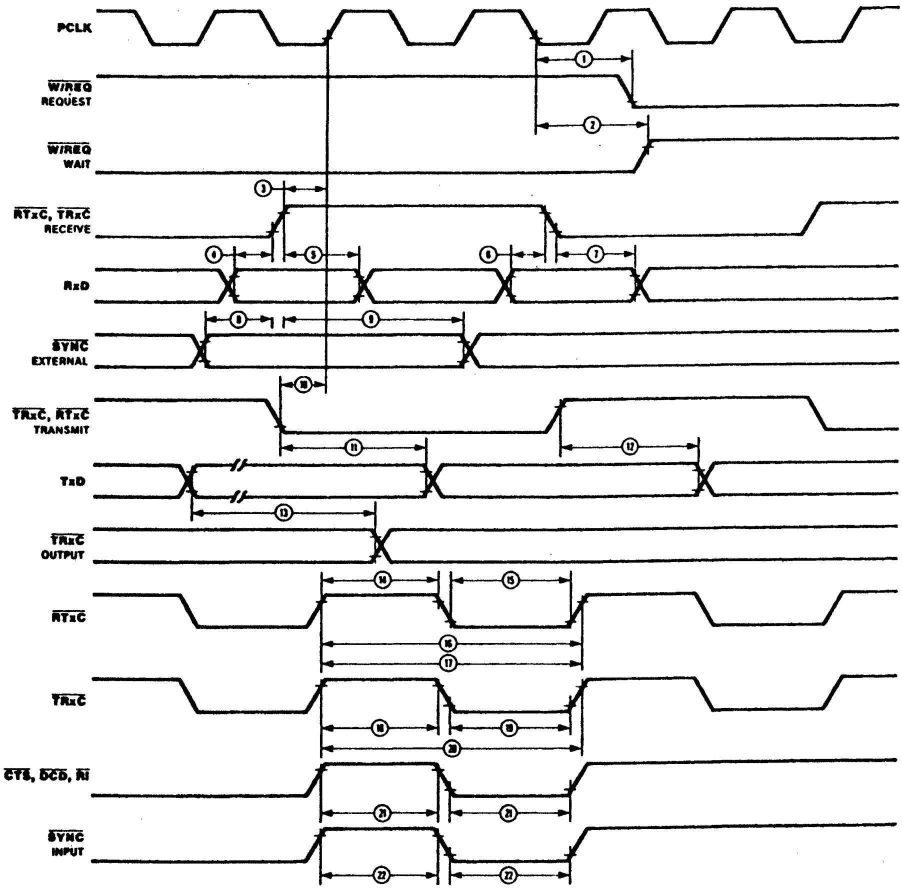
Table 24 : AC Timing Characteristics.

N°	Symbol	Parameter	8MHz		10MHz		12MHz		Notes (4)
			Min. (ns)	**Max.** (ns)	**Min.** (ns)	**Max.** (ns)	**Min.** (ns)	**Max.** (ns)	
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		220		175		150	
28	TwWR1	$\overline{WR}$ Low Width	150		120		100		
29	TsDW(WR)	Write Data to $\overline{WR}$ ↓ Setup Time	10		10		10		
30	ThDW(WR)	Write Data $\overline{WR}$ ↑ Hold Time	0		0		0		
31	TdWR(W)	$\overline{WR}$ ↓ to Wait Valid Delay		170		130		110	2
32	TdRD(W)	$\overline{RD}$ ↓ to Wait Valid Delay		170		130		110	2
33	TdWRf(REQ)	$\overline{WR}$ ↓ to $\overline{W}$ / $\overline{REQ}$ Not Valid Delay		170		130		110	
34	TdRDf(REQ)	$\overline{RD}$ ↓ to $\overline{W}$ / $\overline{REQ}$ Not Valid Delay		170		130		110	
35	TdWRr(REQ)	$\overline{WR}$ ↑ to $\overline{DTR}$ / $\overline{REQ}$ Not Valid Delay		4TcPC		4TcPC		4TcPC	
36	TdRDr(REQ)	$\overline{RD}$ ↑ to $\overline{DTR}$ / $\overline{REQ}$ Not Valid Delay		4TcPC		4TcPC		4TcPC	
37	TdPC(INT)	PCLK ↓ to $\overline{INT}$ Valid Delay		500		500		500	2
38	TdIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ (Acknowledge) Delay	150		120		100		3
39	TwRDA	$\overline{RD}$ (Acknowledge) Width	150		120		100		
40	TdRDA(DR)	$\overline{RD}$ ↓ (Acknowledge) to Read Data Valid Delay		140		100		90	
41	TsIEI(RDA)	IEI to $\overline{RD}$ ↓ (Acknowledge) Setup Time	95		80		70		
42	ThIEI(RDA)	IEI to $\overline{RD}$ ↑ (Acknowledge) Hold Time	0		0		0		
43	TdIEI(IEO)	IEI to IEO Delay Time		95		80		70	
44	TdPC(IEO)	PCLK ↑ to IEO Delay		200		150		130	
45	TdRDA(INT)	$\overline{RD}$ ↓ to $\overline{INT}$ Inactive Delay		450		450		450	2
46	TdRD(WRQ)	$\overline{RD}$ ↑ $\overline{WR}$ ↓ Delay for No Reset	15		10		10		
47	TdWRQ(RD)	$\overline{WR}$ ↑ to $\overline{RD}$ ↓ Delay for No Reset	20		15		15		
48	TwRES	$\overline{WR}$ and $\overline{RD}$ Coincident Low for Reset	150		130		110		
49	Trc	Valid Access Recovery Time	4TcPC		4TcPC		4TcPC		1

Notes :
1. Parameter applies only between transactions involving the SCC.
2. Open-drain output, measured with open-drain test load.
3. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the num of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.
4. Timings are preliminary and subject to change

HARDWARE INFORMATION (cont'd)

Figure 79 : General Timing.



HARDWARE INFORMATION (cont'd)

Table 25 : General Timing Characteristics (cont'd).

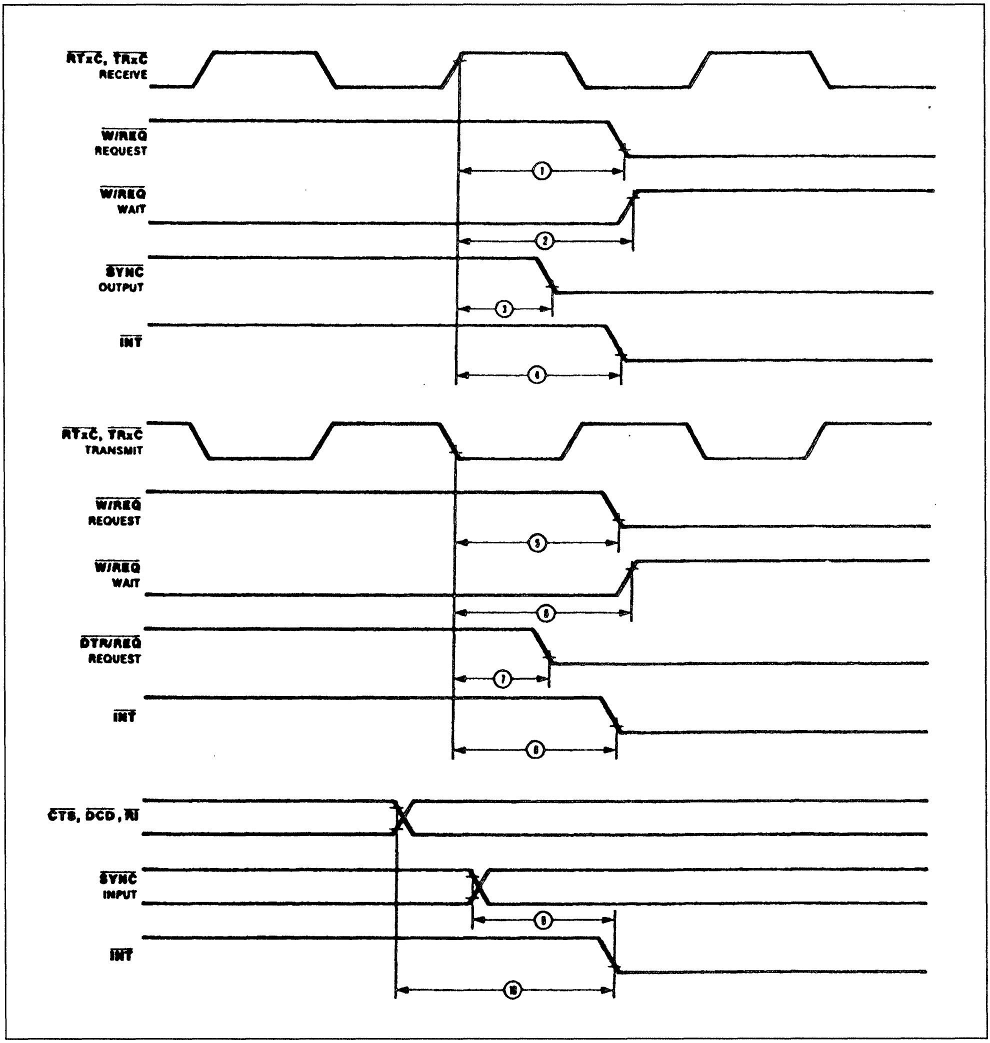
N°	Symbol	Parameter	8MHz		10MHz		12MHz		Notes (8)
			Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	
1	TdPC(REQ)	PCLK ↓ to $\overline{\mathrm{W}}$ / $\overline{\mathrm{REQ}}$ Valid Delay		250		250		250	
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350		350		350	
3	TsRXC(PC)	$\overline{\mathrm{RxC}}$ ↑ to PCLK ↑ Setup Time (PCLK ÷ 4 case only)	80	TwPCL	70	TwPCL	60	TwPCL	1.4
4	TsRXD(RXCr)	RxD to $\overline{\mathrm{RxC}}$ ↑ Setup Time (X1 Mode)	0		0		0		1
5	ThRXD(RXCr)	RxD to $\overline{\mathrm{RxC}}$ ↑ Hold Time (X1 Mode)	150		150		150		1
6	TsRXD(RXCf)	RxD to $\overline{\mathrm{RxC}}$ ↓ Setup Time (X1 Mode)	0		0		0		1.5
7	ThRXD(RXCf)	RxD to $\overline{\mathrm{RxC}}$ ↓ Hold Time (X1 Mode)	150		150		150		1.5
8	TsSY(RXC)	$\overline{\mathrm{SYNC}}$ to $\overline{\mathrm{RxC}}$ ↑ Setup Time	– 200		– 200		– 200		1
9	ThSY(RXC)	$\overline{\mathrm{SYNC}}$ to $\overline{\mathrm{RxC}}$ ↑ Hold Time	3TcPC + 400		3TcPC + 320		3TcPC + 250		1
10	TsTXC(PC)	$\overline{\mathrm{TxC}}$ ↓ to PCLK ↑ Setup Time	0		0		0		2.4
11	TdTXCf(TXD)	$\overline{\mathrm{TxC}}$ ↓ to TxD Delay (X1 Mode)		300		230		200	2
12	TdTXCr(TXD)	$\overline{\mathrm{TxC}}$ ↑ to TxD Delay (X1 Mode)		300		230		200	2.5
13	TdTXD(TRX)	TxD to $\overline{\mathrm{TRxC}}$ Delay (Send Clock Echo)		200		200		200	
14	TwRTXh	$\overline{\mathrm{RTxC}}$ High Width	180		180		150		6
15	TwRTX1	$\overline{\mathrm{RTxC}}$ Low Width	180		180		150		6
16	TcRTX	$\overline{\mathrm{RTxC}}$ Cycle Time	1000		640		500		6.7
17	TcRTXX	Crystal Oscillator Period	125	1000	100	1000	83.3	1000	3
18	TwTRXh	$\overline{\mathrm{TRxC}}$ High Width	150		130		110		6
19	TwTRX1	$\overline{\mathrm{TRxC}}$ Low Width	150		130		110		6
20	TcTRX	$\overline{\mathrm{TRxC}}$ Cycle Time	500		400		333		6.7
21	TwEXT	$\overline{\mathrm{DCD}}$ or $\overline{\mathrm{CTS}}$ Pulse Width	200		200		200		
22	TwSY	$\overline{\mathrm{SYNC}}$ Pulse Width	200		200		200		

Notes :

1. $\overline{\mathrm{RxC}}$ is $\overline{\mathrm{RTxC}}$ or $\overline{\mathrm{TRxC}}$, whichever is supplying the receive clock.
2. $\overline{\mathrm{TxC}}$ is $\overline{\mathrm{TRxC}}$ or $\overline{\mathrm{RTxC}}$, whichever is supplying the transmit clock.
3. Both $\overline{\mathrm{RTxC}}$ and $\overline{\mathrm{SYNC}}$ have 30pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{\mathrm{RxC}}$ and PCLK or $\overline{\mathrm{TxC}}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver ; DPLL and baud rate generat or timing requirements are identical to chip PCLK requirements.
7. The maximum receive on transmit data is 1/4PCLK.
8. Timings are preliminary and subject to change.

HARDWARE INFORMATION (cont'd)

Figure 80 : System Timing.



HARDWARE INFORMATION (cont'd)

Table 26 : System Timing Characteristics (cont'd).

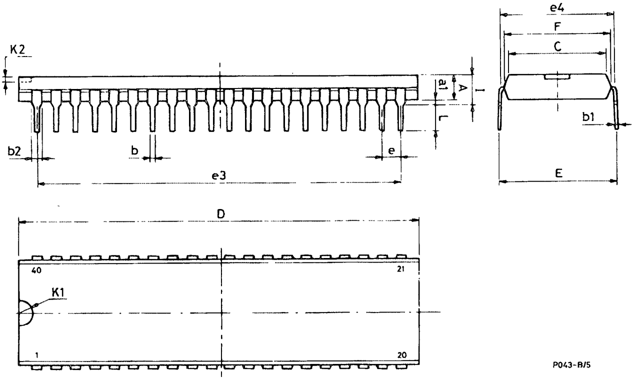
N°	Symbol	Parameter	8MHz		10MHz		12MHz		Notes (4.5)
			Min.	Max.	Min.	Max.	Min.	Max.	
1	TdRXC(REQ)	$\overline{\text{RxC}}$ ↑ to $\overline{\text{W}}$ / $\overline{\text{REQ}}$ Valid Delay	8	12	8	12	8	12	2
2	TdRXC(W)	$\overline{\text{RxC}}$ ↑ to Wait Inactive Delay	8	12	8	12	8	14	1.2
3	TdRXC(SY)	$\overline{\text{RxC}}$ ↑ to $\overline{\text{SYNC}}$ Valid Delay	4	7	4	7	4	7	2
4	TdRXC(INT)	$\overline{\text{RxC}}$ ↑ to $\overline{\text{INT}}$ Valid Delay	10	16	10	16	10	16	1.2
5	TdTXC(REQ)	$\overline{\text{TxC}}$ ↓ to $\overline{\text{W}}$ / $\overline{\text{REQ}}$ Valid Delay	5	8	5	8	5	8	3
6	TdTXC(W)	$\overline{\text{TxC}}$ ↓ to Wait Inactive Delay	5	8	5	8	5	11	1.3
7	TdTXC(DRQ)	$\overline{\text{TxC}}$ ↓ to $\overline{\text{DTR}}$ / $\overline{\text{REQ}}$ Valid Delay	4	7	4	7	4	7	3
8	TdTXC(INT)	$\overline{\text{TxC}}$ ↓ to $\overline{\text{INT}}$ Valid Delay	6	10	6	10	6	10	1.3
9	TdSY(INT)	$\overline{\text{SYNC}}$ Transition to $\overline{\text{INT}}$ Valid Delay	2	6	2	6	2	6	1
10	TdEXT(INT)	$\overline{\text{DCD}}$ or $\overline{\text{CTS}}$ Transition to $\overline{\text{INT}}$ Valid Delay	2	6	2	6	2	6	1

7. The maximum receive on transmit data is 1/4PCLK.
8. Timings are preliminary and subject to change.

Notes :
1. Open-drain output, measured with open-drain test load.
2. $\overline{\text{RxC}}$ is $\overline{\text{RTxC}}$ or $\overline{\text{TRxC}}$, whichever is supplying the receive clock.
3. $\overline{\text{TxC}}$ is $\overline{\text{TRxC}}$ or $\overline{\text{RTxC}}$, whichever is supplying the transmit clock.

PACKAGES MECHANICAL DATA

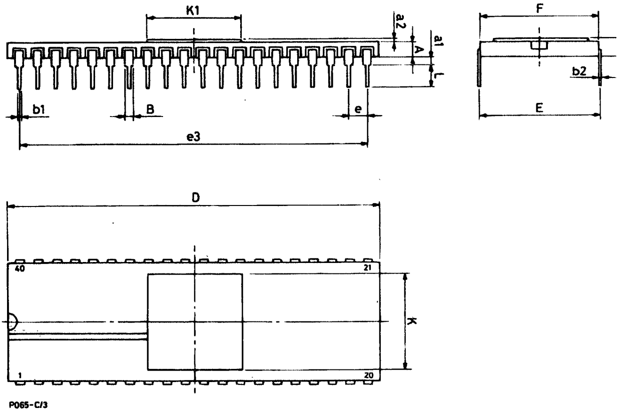
Figure 81 : 40–Pin Dual in Line Plastic (N).



Dim.	mm			Inches		
	Min	Typ	Max	Min	Typ	Max
A						
a1		0.63			.024	
b		0.45			.017	
b1	0.23		0.31	.009		.012
b2		1.27			.050	
C						
D			52.58			2.07
E	15.20		16.68	.598		.656
e		2.54			.100	
e3		48.26			1.90	
e4						
F			14.10			.555
I		4.45			.175	
L		3.30			.129	
K1						
K2						

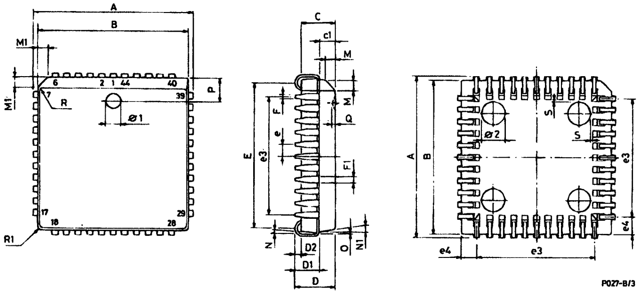
PACKAGE MECHANICAL DATA (cont'd)

Figure 82 : 40–Pin Dual in Line Ceramic (P).



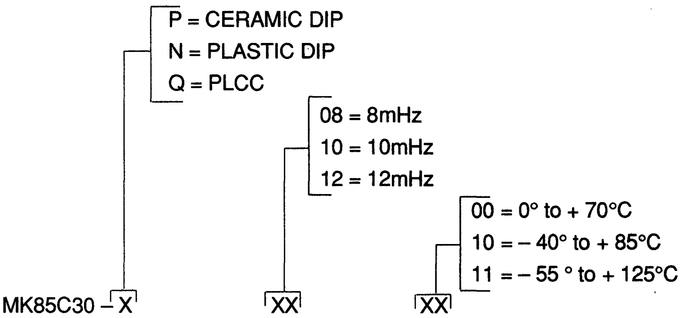
Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A						
a1		1.27			.050	
a2						
B		1.27			.050	
b1		0.45			.017	
b2		0.25			.010	
C		2.33			.091	
D			51.30			2.01
E		15.24			.600	
e		2.54			.100	
e3		48.26			1.90	
F			15.36			.604
L		3.30			.129	
K		12.80			.503	
K1		12.80			.503	

Figure 83 : 44–Lead Plastic Leaded Chip Carrier (Q).



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A	17.40		17.65	0.685		0.694
B	16.51		16.65	0.650		0.655
C	3.65		3.70	0.143		0.145
c1						
D	4.20		4.57	0.165		0.179
d1	2.59		2.74	0.101		0.107
d2		0.68			0.026	
E	14.99		16.00	0.590		0.629
e		1.27			0.050	
e3		12.70			0.500	
e4			1.98			0.077
F		0.46			0.018	
F1		0.71			0.027	
G			0.101			0.004
M		1.16			0.045	
M1		1.14			0.044	
Ø1		1.57				

ORDER CODES



MK85C30 – X XX XX

P = CERAMIC DIP
N = PLASTIC DIP
Q = PLCC

08 = 8mHz
10 = 10mHz
12 = 12mHz

00 = 0° to + 70°C
10 = – 40° to + 85°C
11 = – 55 ° to + 125°C

SGS-THOMSON
MICROELECTRONICS

TS68HC901

HCMOS MULTI FUNCTION PERIPHERAL

PRELIMINARY INFORMATION

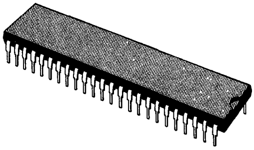
The TS68HC901 multi-function peripheral (CMFP) is a member of the 68000 Family of peripherals and the CMOS version of the MK68901. The CMFP directly interfaces to the 68000 processor via an asynchronous bus structure and can also support both multiplexed and non multiplexed buses. Both vectored, non vectored and polled interrupt schemes are supported, with the CMFP providing unique vector number generation for each of its 16 interrupt sources. Additionally, handshake lines are provided to facilitate DMAC interfacing.

The TS68HC901 performs many of the functions common to most microprocessor-based systems. The resources available to the user include :

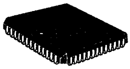
- Eight Individually Programmable I/O Pins with Interrupt Capability
- 16-Source Interrupt Controller with Individual Source Enabling and Masking
- Four Timers, Two of which are Multi-Mode Timers
- Timers may be used as Baud Rate Generators for the Serial Channel
- Single-Channel Full-Duplex Universal Synchronous / Asynchronous Receiver-Transmitter (USART) that Supports Asynchronous and with the Addition of a Polynominal Generator Checker Supports Byte Synchronous Formats.

By incorporating multiple functions within the CMFP, the system designer retains flexibility while minimizing device' count.

The CMOS technology used for the TS68HC901 reduces also the power consumption of the system.

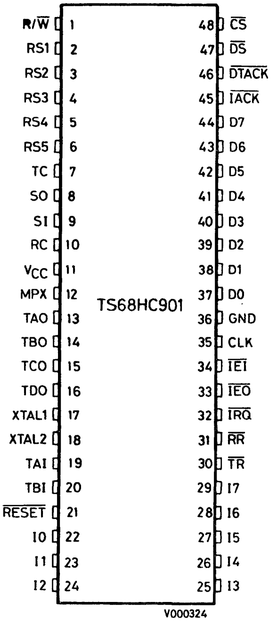


P
(Plastic Package)



FN
(PLCC52)

PIN CONNECTIONS



SECTION 1

INTRODUCTION

The TS68HC901 multi-function peripheral (CMFP) is a member of the 68000 peripherals. The CMFP directly interfaces to the 68000 processor via an asynchronous bus structure. Both vectored and polled interrupt schemes are supported, with the CMFP providing unique vector number generation for each of its 16 interrupt sources. Additionally, handshake lines are provided to facilitate DMAC interfacing. Refer to block diagram of the TS68HC901.

The TS68HC901 performs many of the functions common to most microprocessor-based systems.

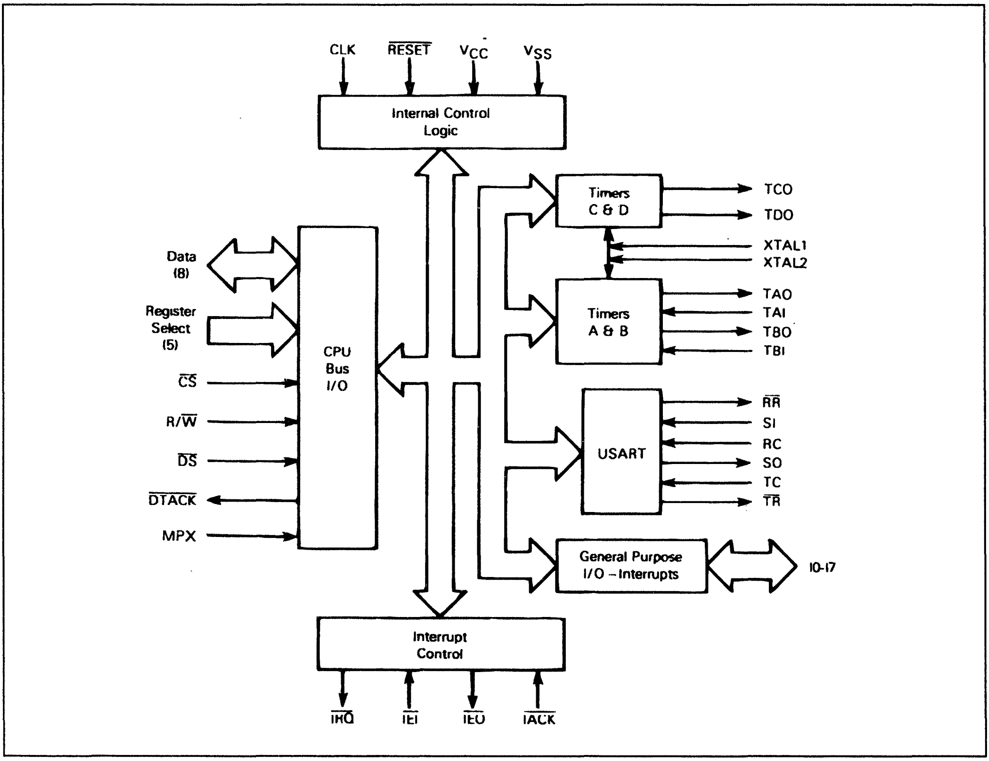
The resources available to the user include :

- Eight Individually Programmable I/O Pins with Interrupt Capability
- 16-Source Interrupt Controller with Individual Source Enabling and Masking
- Four Timers, Two of which are Multi-Mode Timers
- Timers May Be Used as Baud Rate Generators for the Serial Channel
- Single-Channel Full-Duplex Universal Synchronous / Asynchronous Receiver-Transmitter (USART) that Supports Asynchronous and with the Addition of a Polynomial Generator Checker Supports Byte Synchronous Formats

By incorporating multiple functions within the CMFP, the system designer retains flexibility while minimizing device count.

From a programmer's point of view, the versatility of

Figure 1.1 : Block Diagram.



the CMFP may be attributed to its register set. The registers are well organized and allow the CMFP to be easily tailored to a variety of applications. All of the 24 registers are also directly addressable which simplifies programming. The register map is shown in table 1.1.

Table 1.1 : CMFP Register Map.

Address						Abbreviation	Register Name
Hex	Binary						
	RS5	RS4	RS3	RS2	RS1		
01	0	0	0	0	0	GPIP	General Purpose I/O Register
03	0	0	0	0	1	AER	Active Edge Register
05	0	0	0	1	0	DDR	Data Direction Register
07	0	0	0	1	1	IERA	Interrupt Enable Register A
09	0	0	1	0	0	IERB	Interrupt Enable Register B
0B	0	0	1	0	1	IPRA	Interrupt Pending Register A
0D	0	0	1	1	0	IPRB	Interrupt Pending Register B
0F	0	0	1	1	1	ISRA	Interrupt In-service Register A
11	0	1	0	0	0	ISRB	Interrupt In-service Register B
13	0	1	0	0	1	IMRA	Interrupt Mask Register A
15	0	1	0	1	0	IMRB	Interrput Mask Register B
17	0	1	0	1	1	VR	Vector Register
19	0	1	1	0	0	TACR	Timer A Control Register
1B	0	1	1	0	1	TBCR	Timer B Control Register
1D	0	1	1	1	0	TCDCR	Timers C And D Control Register
1F	0	1	1	1	1	TADR	Timer A Data Register
21	1	0	0	0	0	TBDR	Timer B Data Register
23	1	0	0	0	1	TCDR	Timer C Data Register
25	1	0	0	1	0	TDDR	Timer D Data Register
27	1	0	0	1	1	SCR	Synchronous Character Register
29	1	0	1	0	0	UCR	USART Control Register
2B	1	0	1	0	1	RSR	Receiver Status Register
2D	1	0	1	1	0	TSR	Transmitter Status Register
2F	1	0	1	1	1	UDR	USART Data Register

Note : Hex addresses assume that RS1 connects with A1, RS2 connects with A2, etc... and that DS is connected to LDS on the 68000 or DS is connected to DS on the 68008.

SECTION 2

SIGNAL AND BUS OPERATION DESCRIPTION

This section contains a brief description of the input and output signals. A discussion of bus operation during the various operations is also presented.

Note : The terms assertion and negation will be used extensively. This is done to avoid confusion when dealing with a mixture of "active low" and "active high" signals. The term assert or assertion is used to indicate that a signal is active or true, independent of whether that level is represented by a high or low voltage. The term negate or negation is used to indicate that a signal is inactive or false.

2.1. SIGNAL DESCRIPTION

The input and output signals can be functionally organized into the groups shown in figure 2.1. The following paragraphs provide a brief description of the signal and a reference (if applicable) to other sections that contain more detail about its function.

2.1.1. V_{CC} AND GND. These inputs supply power to the CMFP. The V_{CC} is power at + 5 volts and GND is the ground connection.

2.1.2. CLOCK (CLK). The clock input is a single-phase TTL-compatible signal used for internal timing. This input should not be gated off at any time and must conform to minimum and maximum pulse width times. The clock is not necessarily the system clock in frequency nor phase. When the bus is multiplexed (MPX = 1), an address strobe signal is connected to this pin. In the non multiplexed mode (MPX = 0), this input is connected to the system clock when used with a 68000 processor type or to V_{SS} (0 V_{DC}) when used with a 6800 processor type.

2.1.3. ASYNCHRONOUS BUS CONTROL. Asynchronous data transfers are controlled by chip select, data strobe, read/write, and data transfer acknowledge. The low order register select lines, RS1-RS5, select an internal CMFP register for a read or write operation. The reset line initializes the CMFP registers and the internal control signals.

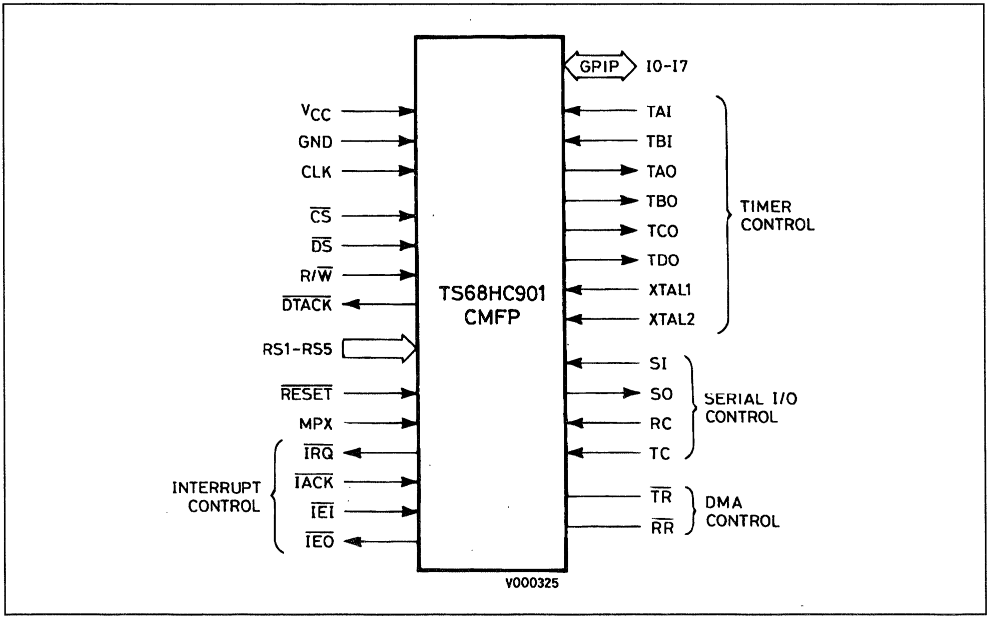
2.1.3.1. Chip Select ($\overline{CS}$).

This input activates the CMFP for internal register access.

2.1.3.2. Data Strobe ($\overline{DS}$).

This input is part of the internal chip select and interrupt acknowledge functions. The CMFP must be

Figure 2-1 : Input and Output Signals.



located on the lower portion of the 16-bit data bus so that the vector number passed to the processor during an interrupt acknowledge cycle will be located in the low byte of the data word. As a result, DS must be connected to the processor's lower data strobe if vectored interrupts are to be used. Note that this forces all registers to be located at odd addresses and latches data on the rising edge for writes. This signal is used as RD with an Intel processor type.

2.1.3.3. Read/Write (R/W).

This input defines a data transfer as a read (high) or a write (low) cycle. This signal is used as WR with an Intel processor type.

2.1.3.4. Data Transfer Acknowledge (DTACK).

This output signals the completion of the operation phase of a bus cycle to the processor. If the bus cycle is a processor read, the CMFP asserts DTACK to indicate that the information on the data bus is valid. If the bus cycle is a processor to the CMFP, DTACK acknowledges the acceptance of the data by the CMFP. DTACK will be asserted only by an CMFP that has CS or IACK (and IEI) asserted. This signal is not used with a 6800 processor type.

2.1.3.5. Register Select Bus (RS1 Through RS5).

The lower five bits if the register select bus select an internal CMFP register during a read or write operation.

2.1.3.6. Data Bus (D0 Through D7).

This bidirectional bus is used to receive data from or transmit data to the CMFP's internal registers during a processor read or write cycle. During an interrupt acknowledge cycle, the data bus is used to pass a vector number to the processor. Since the CMFP is an 8-bit peripheral, the CMFP could be located on either the upper or lower portion of the 16-bit data bus (even or odd address). However, during an interrupt acknowledge cycle, the vector number passed to the processor must be located in the low byte of the data word. As a result, D0-D7 of the CMFP must be connected to the low order eight bits of the processor data bus, placing CMFP registers at odd addresses if vectored interrupts are to be used.

2.1.3.7. Reset (RESET).

This input will initialize the CMFP during power up or in response to a total system reset. Refer to 2.2.3. for further information.

2.1.3.8. MPX.

This input selects the data bus mode :
MPX = 0 : non multiplexed mode
MPX = 1 : multiplexed mode. The register select lines RS1-RS5 and the data bus D0-D7 are multiplexed. An address strobe must be connected to the CLK pin.

2.1.4. INTERRUPT CONTROL. The interrupt request and interrupt acknowledge signals are handshake lines for a vectored interrupt scheme. Interrupt enable in and the interrupt enable out implement a daisy-chained interrupt structure.

2.1.4.1. Interrrupt Request (IRQ).

This output signals the processor that an interrupt is pending from the CMFP. These are 16 interrupt channels that can generate an interrupt request. Clearing the interrupt pending registers (IPRA and IPRB) or clearing the interrupt mask registers (IMRA and IMRB) will cause IRQ to be negated. IRQ will also be negated as the result of an interrupt acknowledge cycle, unless additional interrupts are pending in the CMFP. Refer to **SECTION 3** for further information.

2.1.4.2. Interrupt Acknowledge (IACK).

If both IRQ and IEI are active, the CMFP will begin an interrupt acknowledge cycle when IACK and DS are asserted. The CMFP will supply a unique vector number to the processor which corresponds to the interrupt handler for the particular channel requiring interrupt service. In a daisy-chained interrupt structure, all devices in the chain must have a common IACK. Refer to **2.2.2**. and **3.1.2**. for additional information.

2.1.4.3. Interrupt Enable In (IEI).

This input, together with the IEO signal, provides a daisy-chained interrupt structure for a vectored interrupt scheme. IEI indicates that no higher priority device is requesting interrupt service. So, the highest priority device in the chain should have its IEI pin tied low. During an interrupt acknowledge cycle, an CMFP with a pending interrupt is not allowed to pass a vector number to the processor until its IEI pin is asserted. When the daisy-chain option is not implemented, all CMFPs should have their IEI pin tied low. Refer to **3.2**. for additional information.

2.1.4.4. Interrupt Enable Out (IEO).

This output, together with the IEI signal, provides a daisy-chained interrupt structure for a vectored interrupt scheme. The IEO of a particular CMFP signals lower priority devices that neither the CMFP nor any other higher-priority device is requesting interrupt service. When a daisy-chain is implemented, IEO is tied to the next lower priority device's IEI input. the lowest priority device's IEO is not connected. When the daisy-chain option is not implemented, IEO is not connected. Refer to 3.2 for additional information.

2.1.5. GENERAL PURPOSE I/O INTERRUPT LINES (I0 THROUGH I7). This is an 8-bit pin-programmable I/O port with interrupt capability. The data direction register (DDR) individually defines each line as either a high-impedance input or a TTL-compatible output. As an input, each line can generate an interrupt on the user selected transition of the input signal. Refer to **SECTION 4** for further information.

2.1.6. TIMER CONTROL. These lines provide internal timing and auxiliary timer control inputs required for certain operating modes. Additionally, the timer outputs are included in this group.

2.1.6.1. Timer Clock (XTAL1 AND XTAL2).

This input provides the timing signal for the four timers. A crystal can be connected between the timer clock inputs, XTAL1 and XTAL2, or XTAL2 can be driven with a CMOS-level clock while XTAL1 is grounded. The following crystal parameters are suggested :

a) Parallel resonance, fundamental mode AT-cut
b) Frequency tolerance measured with 18 picofarads load (0.1% accuracy) - drive level 10 microwatts
c) Shunt capacitance equals 7 picofarads maximum
d) Series resistance :
$2.0 < f < 2.7MHz$; $R_S \leq 300\Omega$
$2.8 < f < 4.0MHz$; $R_S \leq 150\Omega$

2.1.6.2. Timer Inputs (TAI AND TBI).

These inputs are control signals for timers A and B in the pulse width measurement mode and event count mode. These signals generate interrupts at the same priority level as the general purpose I/O interrupt lines I4 and I3, respectively. While I4 and I3 do not have interrupt capability when the timers are operated in the pulse width measurement mode or the event count mode, I4 and I3 may still be used for I/O. Refer to **5.1.2** and **5.1.3** for further information.

2.1.6.3. Timer Outputs (TAO, TBO, TCO, AND TDO).

Each timer has an associated output which toggles when its main counter counts through 01 (hexadecimal), regardless of which operational mode is selected. When in the delay mode, the timer output will be a square wave with a period equal to two timer cycles. This output signal may be used to supply the universal synchronous/asynchronous receiver-transmitter (USART) baud rate clocks. Timer outputs TAO and TBO may be cleared at any time by writing a one to the reset location in timer control registers A and B. Also, a device reset forces all timer outputs low. Refer to **5.2.2** for additional information.

2.1.7. SERIAL I/O CONTROL. The full duplex serial channel is implemented by a serial input and output line. The independent receive and transmit sections may be clocked by separate timing signals on the receiver clock input and the transmitter clock input.

2.1.7.1. Serial Input (SI).

This input line is the USART receiver data input. This input is not used in the USART loopback mode. Refer to **6.3.2** for additional information.

2.1.7.2. Serial Output (SO).

This output line is the USART transmitter data output. This output is driven high during a device reset.

2.1.7.3. Receiver Clock (RC).

This input controls the serial bit rate of the receiver. This signal may be supplied by the timer output lines or by any external TTL-level clock which meets the minimum and maximum cycle times. This clock is not used in the USART loopback mode. Refer to 6.3.2 for additional information.

2.1.7.4. Transmitter Clock (TC).

This input controls the serial bit rate of the transmitter. This signal may be supplied by the timer output lines or by an external TTL-level clock which meets the minimum and maximum cycle times.

2.1.8. DMA CONTROL. The USART supports DMA transfers through its receiver ready and transmitter ready status lines.

2.1.8.1. Receiver Ready ($\overline{RR}$).

This output reflects the receiver buffer full status for DMA operations.

2.1.8.2. Transmitter Ready ($\overline{TR}$).

This output reflects the transmitter buffer empty status for DMA operations.

2.1.9. SIGNAL SUMMARY. Table 2.1 is a summary of all the signals discussed in the previous paragraphs.

Table 2.1 : Signal Summary.

Signal Name	Mnemonic	I/O	Active
Power Input	V_{CC}	Input	High
Ground	GND	Input	Low
Clock	CLK	Input	N/A
Chip Select	$\overline{CS}$	Input	Low
Data Strobe	$\overline{DS}$	Input	Low
Read/Write	R/$\overline{W}$	Input	Read-high, Write-low
Data Transfer Acknowledge	$\overline{DTACK}$	Output	Low
Register Select Bus	RS1-RS5	Input	N/A
Data Bus	D0-D7	I/O	N/A
Reset	$\overline{RESET}$	Input	Low
Interrupt Request	$\overline{IRQ}$	Output	Low
Interrupt Acknowledge	$\overline{IACK}$	Input	Low
Interrupt Enable In	$\overline{IEI}$	Input	Low
Interrupt Enable Out	$\overline{IEO}$	Output	Low
General Purpose I/O - Interrupt Lines	I0-I7	I/O	N/A
Timer Clock	XTAL1, XTAL2	Input	High
Timer Inputs	TAI, TBI	Input	N/A
Timer Outputs	TAO, TBO, TCO, TDO	Output	N/A
Serial Input	SI	Input	N/A
Serial Output	SO	Output	N/A
Receiver Clock	RC	Input	N/A
Transmitter Clock	TC	Input	N/A
Receiver Ready	$\overline{RR}$	Output	Low
Transmitter Ready	$\overline{TR}$	Output	Low
MPX	$\overline{MPX}$	Input	N/A

2.2. BUS OPERATION

The following paragraphs explain the control signals and bus operation during data transfer operations and reset.

2.2.1. DATA TRANSFER OPERATIONS. Transfer of data between devices involves the following pins :

Register Select Bus - RS1 through RS5
Data Bus - D0 through D7
Control Signals

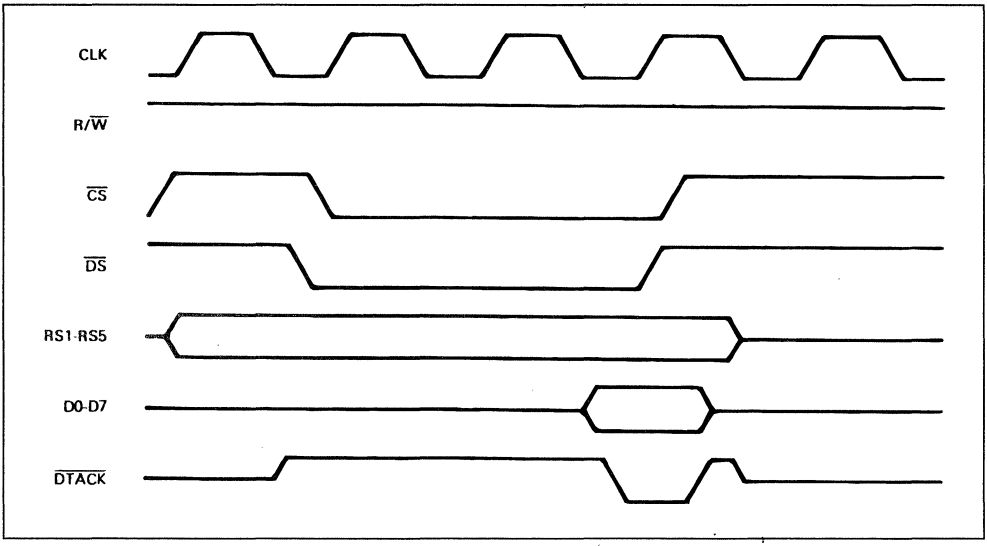
The address and data buses are separate parallel buses used to transfer data using an asynchronous bus structure. In all cycles, the bus master assumes responsibility for deskewing all signals it issues at both the start and end of a cycle. Additionally, the bus master is responsible for deskewing the acknowledge and data signals from the peripheral devices.

2.2.1.1. Read Cycle.

To read an CMFP register, $\overline{CS}$ and $\overline{DS}$ must be asserted, and R/W must be high. The CMFP will place the contents of the register which is selected by the register select bus (RS1 through RS5) on the data bus (D0 through D7) and then assert DTACK. The register addresses are shown in table 1.1.

After the processor has latched the data, $\overline{DS}$ is negated. The negation of either CS or DS will terminate the read operation. The CMFP will drive DTACK high and place it in the high-impedance state. Also, the data bus will be in the high-impedance state. The timing for a read cycle is shown in figure 2.2. Refer to **7.7** for actual timing numbers.

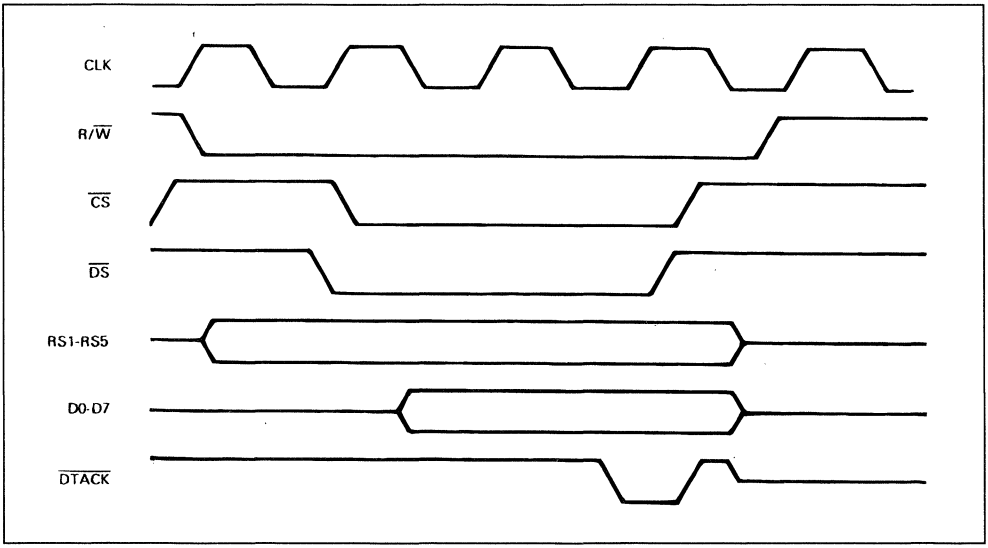
Figure 2.2 : Read Cycle Timing.



2.2.1.2. Write Cycle.

To write a register, CS and DS must be asserted, and R/W must be low. The CMFP will decode the address bus to determine which register is selected (the register map is shown in table 1.1). Then the register will be loaded with the contents of the data bus and DTACK will be asserted. When the processor recognizes DTACK, DS will be negated. The write cycle is terminated when either CS or DS is negated. The CMFP will drive DTACK high and place it in the high-impedance state. The timing for a write cycle is shown in figure 2.3. Refer to 7.7 for actual numbers.

Figure 2.3 : Write Cycle Timing.



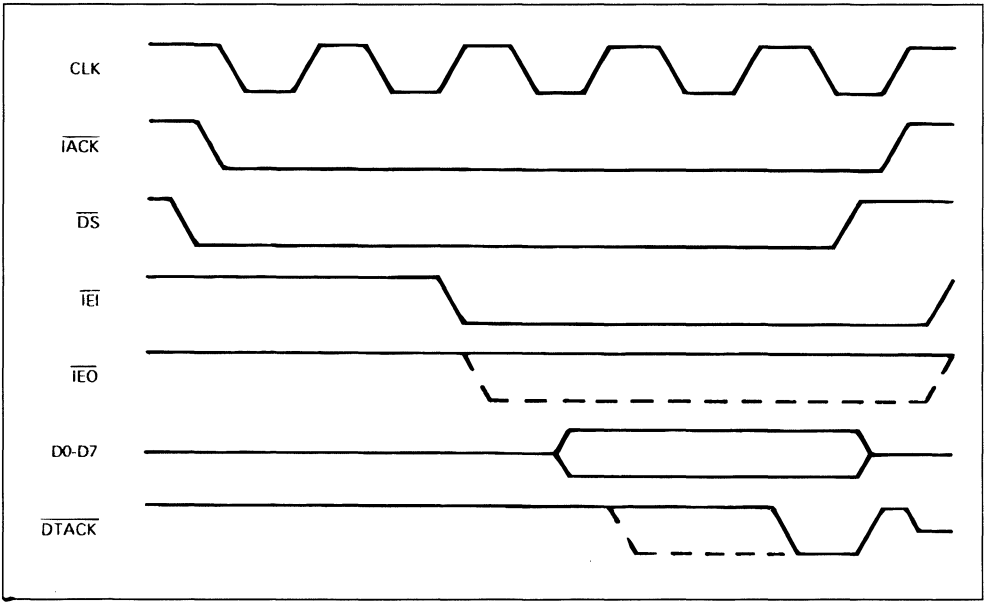
2.2.2. INTERRUPT ACKNOWLEDGE OPERATION. The CMFP has 16 interrupt sources, eight internal sources, and eight external sources. When an interrupt request is pending, the CMFP will assert IRQ. In a vectored interrupt scheme, the processor will acknowledge the interrupt request by performing an interrupt acknowledge cycle. IACK and DS will be asserted. The CMFP responds to the IACK signal by placing a vector number on the lower eight bits of the data bus. This vector number corresponds to the IRQ handler for the particular interrupt requesting service. The format of this vector number is given in figure 3.1.

When the CMFP asserts DTACK to indicate that valid data is on the bus, the processor will latch the data and terminate the bus cycle by negating DS. When either DS or IACK are negated, the CMFP will terminate the interrupt acknowledge operation by driving DTACK high and placing it in the high-impedance state. Also, the data bus will be placed in the high-impedance state. IRQ will be negated as a result of the IACK cycle unless additional interrupts are pending.

The CMFP can be part of a daisy-chain interrupt structure which allows multiple CMFPs to be placed at the same interrupt level by sharing a common IACK signal. A daisy-chain priority scheme is implemented with signals IEI and IEO. IEI indicates that no higher priority device is requesting interrupt service. IEO signals lower priority devices that neither this device nor any higher priority device is requesting service. To daisy-chain CMFPs, the highest priority CMFP has its IEI tied low and successive CMFPs have their IEI connected to the next higher priority device's IEO. Note that when the daisy-chain interrupt structure is not implemented, the IEI of all CMFPs must be tied low. Refer to **3.2** for additional information.

When the processor initiates an interrupt acknowledge cycle by driving IACK and DS, the CMFP whose IEI is low may respond with a vector number if an interrupt is pending. If this device does not have a pending interrupt, IEO is asserted which allows the next lower priority device to respond to the interrupt acknowledge. When an CMFP propagates IEO, it will not drive the data bus nor DTACK during the interrupt acknowledge cycle. The timing for an IACK cycle is shown in figure 2.4. Refer to **7.6** for further information.

Figure 2.4 : IACK Cycle Timing.



2.2.3. RESET OPERATION. The reset operation will initialize the CMFP to a known state. The reset operation requires that the $\overline{\text{RESET}}$ input be asserted for a minimum of two microseconds. During a device reset condition, all internal CMFP registers are cleared except for the timer data registers (TADR, TBDR, TCDR, and TDDR), the USART data register (UDR), the transmitter status register (TSR) and the interrupt vector register. All timers are stopped and the USART receiver and transmitter are disabled. The interrupt channels are also disabled and any pending interrupts are cleared. In addition, the general purpose interrupt I/O lines are placed in the high-impedance input mode and the timer outputs are driven low. External CMFP signals are negated. The interrupt vector register is initialized to a $0F.

2.2.4. NON MULTIPLEXED MODE. In this mode the MPX input must be set to zero, and the TS68HC901 can be used with a 68000 processor type or a 6800 processor type. Refer to figure 7.4, 7.5, 7.8 for the electrical characteristics.

With a 6800 processor type the $\overline{\text{DS}}$ pin is connected to the E signal of the processor, the $\overline{\text{DTACK}}$ signal is not used and the CLK must be zeroed.

2.2.5. MULTIPLEXED MODE. The CMFP can be used either on a MOTOROLA or INTEL bus type. In this case the MPX pin is connected to V_{CC}. The following table gives the signification of the different signals used. A dummy access to the TS68HC901 has to be done before any valid access in order to set up the internal logic of sampling.

Pin	MOTOROLA 6800 Type	MOTOROLA Multiplexed	INTEL
48	$\overline{\text{CS}}$	$\overline{\text{CS}}$	$\overline{\text{CS}}$
47	E	$\overline{\text{DS}}$	$\overline{\text{RD}}$
1	R/$\overline{\text{W}}$	R/$\overline{\text{W}}$	$\overline{\text{WR}}$
35	V_{SS}	AS	ALE

SECTION 3

INTERRUPT STRUCTURE

In a 68000 system, the CMFP will be assigned to one of the seven possible interrupt levels. All interrupt service requests from the CMFP's 16 interrupt channels will be presented at this level. Although, as an interrupt controller, the CMFP will internally prioritize its 16 interrupt sources. Additional interrupt sources may be placed at the same interrupt level by daisy-chaining multiple CMFPs. The CMFPs will be prioritized by their position in the chain.

3.1. INTERRUPT PROCESSING

Each CMFP provides individual interrupt capability for its various functions. When an interrupt is received on one of the external interrupt channels or from one of the eight internal sources, the CMFP will request interrupt service. The 16 interrupt channels are assigned a fixed priority so that multiple pending interrupts are serviced according to their relative importance. Since the CMFP can internally generate 16 vector numbers, the unique vector number which corresponds to the highest priority channel that has a pending interrupt is presented to the processor during an interrupt acknowledge cycle. This unique vector number allows the processor to immediately begin execution of the interrupt handler for the interrupting source, decreasing interrupt latency time.

3.1.1. INTERRUPT CHANNEL PRIORITIZATION. The 16 interrupt channels are prioritized as shown in table 3.1. General purpose interrupt 7 (I7) is the highest priority interrupt channel and I0 is the lowest priority channel. Pending interrupts are presented to the CPU in order of priority unless they have been masked off. By selectively masking interrupts, the channels are in effect re-prioritized.

3.1.2. INTERRUPT VECTOR NUMBER FORMAT. During an interrupt acknowledge cycle, a unique 8-bit vector number is presented to the system which corresponds to the specific interrupt source which is requesting service. The format of the vector is shown in figure 3.1. The most significant four bits of the interrupt vector number are user programmable. These bits are set by writing the upper four bits of the vector register which is shown in figure 3-2. The low order bits are generated internally by the TS68HC901. Note that the binary channel number shown in table 3.1 corresponds to the low order bits of the vector number associated with each channel.

Table 3.1 : Interrupt Channel Prioritization.

Priority	Channel	Description
Highest	1111	General Purpose Interrupt 7 (I7)
	1110	General Purpose Interrupt 6 (I6)
	1101	Timer A
	1100	Receiver Buffer Full
	1011	Receive Error
	1010	Transmitt Buffer Empty
	1001	Transmit Error
	1000	Timer B
	0111	General Purpose Interrupt 5 (I5)
	0110	General Purpose Interrupt 4 (I4)
	0101	Timer C
	0100	Timer D
	0011	General Purpose Interrupt 3 (I3)
	0010	General Purpose Interrupt 2 (I2)
	0001	General Purpose Interrupt 1 (I1)
Lowest	0000	General Purpose Interrupt 0 (I0)

Figure 3.1 : Interrupt Vector Format.

7	6	5	4	3	2	1	0
V7	V6	V5	V4	IV3	IV2	IV1	IV0

V7-V4 The four most significant bits are copied from the vector register

IV3-IV0 These bits are supplied by the CMFP. They are the binary channel number of the highest priority channel that is requesting interrupt service.

Figure 3.2 : Vector Register Format (VR).

	7	6	5	4	3	2	1	0
Address 17 (Hex)	V7	V6	V5	V4	S	*	*	*

* Unused bits are read as zero.

V7-V4 The upper four bits of the vector register are written by the user. These bits become the most significant four bits of the interrupt vector number.
SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

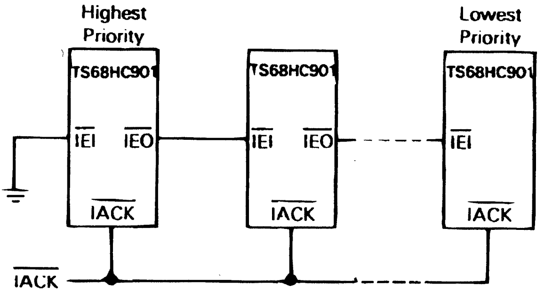
S In-Service Register Enable. When the S bit is zero, the CMFP is in the automatic end-of-interrupt mode and the in-service register bits are forced low. When the S bit is a one, the CMFP is in the software end-of-interrupt mode and the in-Service register bits are enabled. Refer to **3.4.2** and **3.4.3** for additional information.
SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

3.2. DAISY-CHAINING CMFPs

As an interrupt controller, the TS68HC901 CMFP will support eight external interrupt sources in addition to its eight internal interrupt sources. When a system requires more than eight external interrupt sources to be placed at the same interrupt level, sources may be added to the prioritized structure by daisy-chaining CMFPs. Interrupt sources are prioritized internally within each CMFP and the CMFPs are prioritized by their position in the chain. Unique vector numbers are provided for each interrupt source.

The $\overline{\text{IEI}}$ and $\overline{\text{IEO}}$ signals implement the daisy-chained interrupt structure. The $\overline{\text{IEI}}$ of the highest priority CMFP is tied low and the $\overline{\text{IEO}}$ output of this device is tied to the next highest priority CMFP's $\overline{\text{IEI}}$. The $\overline{\text{IEI}}$ and $\overline{\text{IEO}}$ signals are daisy-chained in this manner for all CMFPs in the chain, with the lowest priority CMFP's $\overline{\text{IEO}}$ left unconnected. A diagram of an interrupt daisy-chain is shown in figure 3.3. .

Figure 3.3 : Daisy-Chained Interrupt Structure.



Daisy-chaining requires that all parts in the chain have a common $\overline{\text{IACK}}$. When the common $\overline{\text{IACK}}$ is asserted during an interrupt acknowledge cycle, all parts will prioritize interrupts in parallel. When the $\overline{\text{IEI}}$ signal to a CMFP is asserted, the part may respond to the $\overline{\text{IACK}}$ cycle if it requires interrupt service. Otherwise, the part will assert $\overline{\text{IEO}}$ to the next lower priority device. Thus, priority is passed down the chain via $\overline{\text{IEI}}$ and $\overline{\text{IEO}}$ until a part which has a pending interrupt is reached. The part with the pending interrupt passes a vector number to the processor and does not propagate $\overline{\text{IEO}}$.

3.3. INTERRUPT CONTROL REGISTERS

CMFP interrupt processing is managed by the interrupt enable registers A and B, interrupt pending registers A and B, and interrupt mask registers A and B. These registers allow the programmer to enable or disable individual interrupt channels, mask individual interrupt channels, and access pending interrupt status information. In-service registers A and B allow interrupts to be nested as described in **3.4**. The interrupt control registers are shown in figure 3.4.

3.3.1. INTERRUPT ENABLE REGISTERS.

The interrupt channels are individually enabled or disabled by writing a one or zero, respectively, to the appropriate bit of interrupt enable register A (IERA) or interrupt enable register B (IERB). The processor may read these registers at any time.

When a channel is enabled, interrupts received on the channel will be recognized by the CMFP and $\overline{\text{IRQ}}$ will be asserted to the processor, indicating that interrupt service is required. On the other hand, a disabled channel is completely inactive ; interrupts received on the channel are ignored by the CMFP.

Writing a zero to a bit of interrupt enable register A or B will cause the corresponding bit of interrupt pending register A or B to be cleared. This will terminate all interrupt service requests for the channel and also negate $\overline{\text{IRQ}}$, unless interrupts are pending from other sources. Disabling a channel, however, does not affect the corresponding bit in interrupt in-service registers A or B. So, if the CMFP is in the software end-of-interrupt mode (see **3.4.3**) and an interrupt is in service when a channel is disabled, the in-service status bit for that channel will remain set until cleared by software.

Figure 3.4 : Interrupt Control Registers.

(a) Interrupt Enable Registers (IERA and IERB).

	7	6	5	4	3	2	1	0
Address 07 (Hex)	GPIP7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

	7	6	5	4	3	2	1	0
Address 09 (Hex)	GPIP5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is a zero, the associated interrupt channel is disabled. When a bit is a one, the associated interrupt channel is enabled.

SET	a) MPU writes a one
CLEARED	a) MPU writes a zero b) Reset

Figure 3.4 : Interrupt Control Registers (continued).

(b) Interrupt Pending Registers (IPRA and IPRB).

	7	6	5	4	3	2	1	0
Address 0B (Hex)	GPIP7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

	7	6	5	4	3	2	1	0
Address 0D (Hex)	GPIP5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is a zero, no interrupt is pending on the associated interrupt channel. When a bit is a one, an interrupt is pending on the associated interrupt channel.

SET a) Interrupt is received on an enabled interrupt channel
CLEARED a) Interrupt vector for the associated interrupt channel is passed during an $\overline{\text{IACK}}$ cycle
b) Associated interrupt channel is disabled
c) MPU writes a zero
d) Reset

(c) Interrupt In-Service Registers (ISRA and ISRB).

	7	6	5	4	3	2	1	0
Address 0F (Hex)	GPIP7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

	7	6	5	4	3	2	1	0
Address 11 (Hex)	GPIP5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is a zero, no interrupt processing is in progress for the associated interrupt channel. When a bit is a one, interrupt processing is in progress for the associated interrupt channel.

SET a) Interrupt vector number for the associated interrupt channel is passed during an $\overline{\text{IACK}}$ cycle and the S bit of the vector register is set.
CLEARED a) Interrupt service is completed for the associated interrupt channel
b) The S bit of the vector register is a zero
c) MPU writes a zero
d) Reset

(d) Interrupt Mask Registers (IMRA and IMRB).

	7	6	5	4	3	2	1	0
Address 13 (Hex)	GPIP7	GPIP6	Timer A	RCV Buffer Full	RCV Error	XMIT Buffer Empty	XMIT Error	Timer B

	7	6	5	4	3	2	1	0
Address 15 (Hex)	GPIP5	GPIP4	Timer C	Timer D	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is zero, interrupts are masked for the associated interrupt channel. When a bit is a one, interrupts are not masked for the associated interrupt channel.

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

3.3.2. INTERRUPT PENDING REGISTERS. When an interrupt is received on an enabled channel, the corresponding interrupt pending bit is set in interrupt pending register A or B (IPRA or IPRB). In a vectored interrupt scheme, this bit will be cleared when the processor acknowledges the interrupting channel and the CMFP responds with a vector number. In a polled interrupt system, the interrupt pending registers must be read to determine the interrupting channel and then the Interrupting pending bit is cleared by the interrupt handling routine without performing an interrupt acknowledge sequence.

A single bit of the interrupt pending registers is cleared in software by writing ones to all bit positions except the bit to be cleared. Note that writing ones to IPRA and IPRB has no effect on the contents of the register. A single bit of the interrupt pending registers is also cleared when the corresponding channel is disabled by writing a zero to the appropriate bit of IERA or IERB.

3.3.3. INTERRUPT MASK REGISTERS. Interrupts are masked for a channel by clearing the appropriate bit in interrupt mask register A or B (IMRA or IMRB). Even though an enabled channel is masked, the channel will recognize subsequent interrupts and set its interrupt pending bit. However, the channel is prevented from requesting interrupt service ($\overline{\text{IRQ}}$ to the processor) as long as the mask bit for that channel is cleared.

If a channel is requesting interrupt service at the time that its corresponding bit in IMRA or IMRB is cleared, the request will cease and $\overline{\text{IRQ}}$ will be negated, unless another channel is requesting interrupt service. Later, when the mask bit is set, any pending interrupt on the channel will be processed according to the channel's assigned priority. IMRA and IMRB may be read at any time.

3.4. NESTING CMFP INTERRUPTS

In a 68000 vectored interrupt system, the CMFP is assigned to one of seven possible interrupt levels. When an interrupt is received from the CMFP, an interrupt acknowledge for that level is initiated. Once an interrupt is recognized at a particular level, interrupts at that same level or below are masked by 68000. As long as the processor's interrupt mask is unchanged, the 68000 interrupt structure will prohibit the nesting of interrupts at the same interrupt level. However, additional interrupt requests from the CMFP can be recognized before a previous channel's interrupt service routine is completed by lowering the processor's interrupt mask to the next lower interrupt level within the interrupt handler.

When nesting CMFP interrupts, it may be desirable to permit interrupts on any CMFP channel, regardless of its priority, to preempt or delay interrupt processing of an earlier channel's interrupt service request. Or, it may be desirable to only allow subsequent higher priority channel interrupt requests to supercede previously recognized lower priority interrupt requests. The CMFP interrupt structure provides this flexibility by offering two end-of-interrupt options for vectored interrupt schemes. Note that the end-of-interrupt modes are not active in a polled interrupt scheme.

3.4.1. SELECTING THE END-OF-INTERRUPT MODE. In a vectored interrupt scheme, the CMFP may be programmed to operate in either the automatic end-of-interrupt mode or the software end-of-interrupt mode. The mode is selected by writing the S bit of the vector register (see figure 3.2). When the S bit is programmed to a one, the CMFP is placed in the software end-of-structure mode and when the S bit is a zero, all channels operate in the automatic end-of-interrupt mode.

3.4.2. AUTOMATIC END-OF-INTERRUPT. When an interrupt vector number is passed to the processor during an interrupt acknowledge cycle, the corresponding channel's interrupt pending bit is cleared. In the automatic end-of-interrupt mode, no further history of the interrupt remains in the CMFP. The in-service bits of the interrupt in-service registers (ISRA and ISRB) are forced low. Subsequent interrupts which are received on any CMFP channel will generate an interrupt request to the processor, even if the current interrupt's service routine has not been completed.

3.4.3. SOFTWARE END-OF-INTERRUPT. In the software end-of-interrupt mode, the channel's associated interrupt pending bit is cleared and in addition, the channel's in-service bit of in-service register A or B is set when its vector number is passed to the processor during an $\overline{\text{IACK}}$ cycle. A higher priority channel may subsequently request interrupt service and be acknowledged, but as long as the channel's in-service bit is set, no lower priority channel may request interrupt service nor pass its vector during an interrupt acknowledge sequence.

While only higher priority channels may request interrupt service, any channel can receive an interrupt and set its interrupt pending bit. Even the channel whose in-service bit is set can receive a second interrupt. However, no interrupt service request is made until its in-service bit is cleared.

The in-service bit for a particular channel can be cleared by writing a zero to its corresponding bit in ISRA or ISRB and ones to all other bit positions. Since bits in the in-service registers can only be cleared in software and not set, writing ones to the registers does not alter their contents. ISRA and ISRB may be read at any time.

SECTION 4

GENERAL PURPOSE INPUT/OUTPUT INTERRUPT PORT

The general purpose interrupt input/output (I/O) port (GPIP) provides eight I/O lines (I0 through I7) that may be operated as either inputs or outputs under software control. In addition, these lines may optionally generate an interrupt on either a positive transition or a negative transition of the input signal. The flexibility of the GPIP allows it to be configured as an 8-bit I/O port or for bit I/O. Since interrupts are enabled on a bit-by-bit basis, a subset of the GPIP could be programmed as handshake lines or the port could be connected to as many as eight external interrupt sources, which would be prioritized by the CMFP interrupt controller for interrupt service.

4.1. 6800 INTERRUPT CONTROLLER

The CMFP interrupt controller is particularly useful in a system which has many 6800-type devices. Typically, in a vectored 68000 system, 6800-type peripherals use the autovector which corresponds to their assigned interrupt level since they do not provide a vector number in response to an IACK cycle. The autovector interrupt handler must then poll all 6800-type devices at that interrupt level to determine which device is requesting service. However, by tying the IRQ output from a 6800-type device to the general purpose I/O interrupt port (GPIP) of a CMFP, a unique vector number will be provided to the processor during an interrupt acknowledge cycle. This interrupt structure will significantly reduce interrupt latency for 6800-type devices and other peripheral devices which do not support vector-by-device.

4.2. GPIP CONTROL REGISTERS

The GPIP is programmed via three control registers shown in figure 4.1. These registers control the data direction, provide user access to the port, and specify the active edge for each bit of the GPIP which will produce an interrupt. These registers are described in detail in the following paragraphs.

4.2.1. GPIP DATA REGISTER. The general purpose I/O data register is used to input or output data to the port. When data is written to the GPIP data register, those pins which are defined as inputs will remain in the high-impedance state. Pins which are defined as outputs will assume the state (high or low) of their corresponding bit in the data register. When the GPIP is read, data will be passed directly from the bits of the data register for pins which are defined as outputs. Data from pins defined as inputs will come from the input buffers.

4.2.2. ACTIVE EDGE REGISTER. The active edge register (AER) allows each of the GPIP lines to produce an interrupt on either a one-to-zero or a zero-to-one transition. Writing a zero to the appropriate edge bit of the active edge register causes the associated input to generate an interrupt on the one-to-zero transition. Writing a one to the edge bit will produce an interrupt on the zero-to-one transition of the corresponding GPIP line.

Figure 4.1 : GPIP Control Registers.

(a) GPIP Data Register (GPIP).

	7	6	5	4	3	2	1	0
Address 01 (Hex)	GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0

SET a) MPU writes a one

(b) Active Edge Register (AER)

	7	6	5	4	3	2	1	0
Address 03 (Hex)	GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0

CLEARED a) MPU writes a zero

When a bit is a zero, interrupts will be generated on the falling edge of the associated input signal. When a bit is a one, interrupts will be generated on the rising edge of the associated input signal

SET a) MPU writes a one
CLEARED a) MPU writes a zero

Figure 4.1 : GPIP Control Registers (continued).

(c) Data Direction Register (DDR).

	7	6	5	4	3	2	1	0
Address 05 (Hex)	GPIP7	GPIP6	GPIP5	GPIP4	GPIP3	GPIP2	GPIP1	GPIP0

When a bit is a zero, the associated I/O line is defined to be an input. When a bit is a one, the associated I/O line is defined to be an output

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

Note : The transition detector is an exclusive-OR gate whose inputs are the edge bit and the input buffer. As a result, writing the AER may cause an interrupt-producing transition, depending upon the state of the input. So, the AER should be configured before enabling interrupts via the interrupt enable registers (IERA and IERB). Also, changing the edge bit while interrupts are enabled may cause an interrupt on the corresponding channel.

4.2.3. DATA DIRECTION REGISTER. The data direction register (DDR) allows the programmer to define I0 through I7 as inputs or outputs by writing the corresponding bit. When a bit of the data direction register is written as a zero, the corresponding interrupt I/O pin will be a high-impedance input. Writing a one to any bit of the data direction register will cause the corresponding pin to be configured as a push-pull output.

SECTION 5

TIMERS

The CMFP contains four 8-bit timers which provide many functions typically required in microprocessor systems. The timers can supply the baud rate clocks for the on-chip serial I/O channel, generate periodic interrupts, measure elapsed time, and count signal transitions. In addition, two timers have waveform generation capability.

All timers are prescaler/counter timers with a common independent clock input (XTAL1 or XTAL2) and are not required to be operated from the system clock. Each timer's output signal toggles when the timer's main counter times out. Additionally, timers A and B have auxiliary control signals which are used in two of the operation modes. An interrupt channel is assigned to each timer and when the auxiliary control signals are used, a separate interrupt channel will respond to transitions on these inputs.

5.1. OPERATION MODES

Timers A and B are full function timers which, in addition to the delay mode, operate in the pulse width measurement mode and the event count mode. Timers C and D are delay timers only. A brief discussion of each of the timer modes follows.

5.1.1. DELAY MODE OPERATION. All timers may operate in the delay mode. In this mode, the prescaler is always active. The prescaler specifies the number of timer clock cycles which must elapse before a count pulse is applied to the main counter. A count pulse causes the main counter to decrement by one. When the timer has decremented down to 01 (hexadecimal), the next count pulse will cause the main counter to be reloaded from the timer data register and a time out pulse will be produced. This time out pulse is coupled to the timer's interrupt channel and, if the channel is enabled, an interrupt will occur. The time out pulse also causes the timer output pin to toggle. The output will remain in this new state until the next time out pulse occurs.

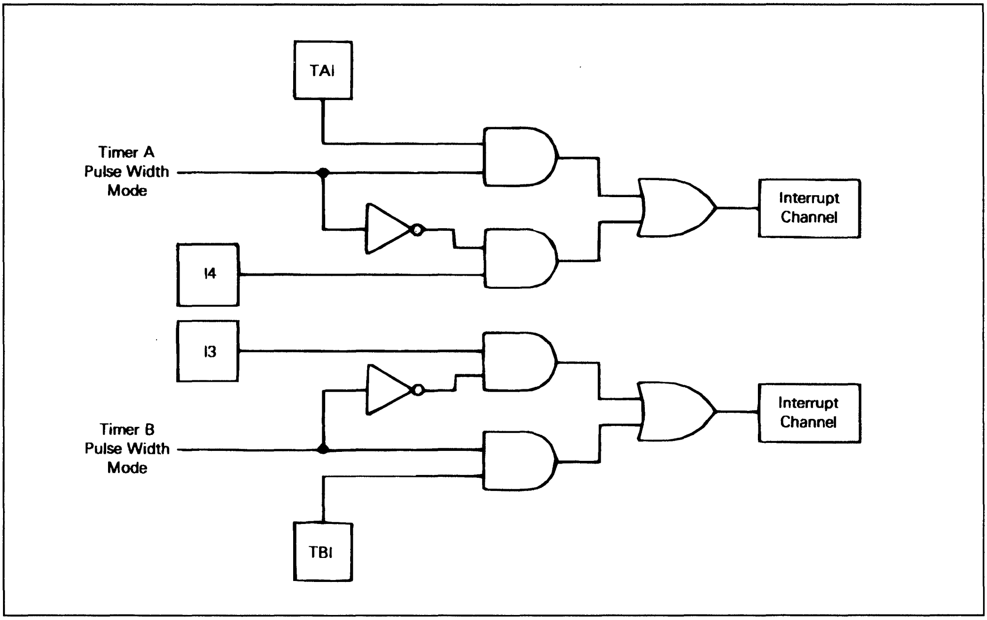
For example, if delay mode with a divide-by-10 prescaler is selected and the timer data register is loaded with 100 (decimal), the main counter will decrement once every 10 timer clock cycles. After 1,000 timer clocks, a time out pulse will be produced. This time out pulse will generate an interrupt if the channel is enabled (IERA, IERB) and in addition, the timer's output line will toggle. The output line will complete one full period every 2,000 cycles of the timer clock.

If the prescaler value is changed while the timer is enabled, the first time out pulse will occur at an indeterminate time no less than one nor more than 200 timer clock cycles. Subsequent time out pulses will then occur at the correct interval.

If the main counter is loaded with 01 (hexadecimal), a time out pulse will occur every time the prescaler presents a count pulse to the main counter. If the main counter is loaded with 00, a time out pulse will occur every 256 count pulses.

5.1.2. PULSE WIDTH MEASUREMENT OPERATION. Besides the delay mode, timers A and B may be programmed to operate in the pulse width measurement mode. In this mode an auxiliary control input is required ; timers A and B auxiliary input lines are TAI and TBI. Also, in the pulse width measurement mode, interrupt channels normally associated with I4 and I3 will respond to transitions on TAI and TBI, respectively. General purpose lines I3 and I4 may still be used for I/O. A conceptual circuit of the timers in the pulse width measurement mode is shown in figure 5.1.

Figure 5.1 : Conceptual Circuit of Timers A and B in Pulse Width Measurement Mode.



The pulse width measurement mode functions similarly to the delay mode, with the auxiliary control signal acting as an enable to the timer. When the control signal is active, the prescaler and main counter are allowed to operate. When the control signal is negated, the timer is stopped. So, the width of the active pulse on TAI or TBI is measured by the number of timer counts which occur while the timer is allowed to operate.

The active state of the auxiliary input line is defined by the associated interrupt channel's edge bit in the active edge register (AER). GPIP4 of the AER is the edge bit associated with TAI and GPIP3 is associated with TBI. When the edge bit is a one, the auxiliary input will be active high, enabling the timer while the input signal is at a high level. If the edge bit is low, the auxiliary input will be active low and the timer will operate while the input signal is at a low level.

The state of the active edge bit also specifies whether a zero-to-one transition or a one-to-zero transition of the auxiliary input pin will produce an interrupt when the interrupt channel is enabled. In normal operation, programming the active edge bit to a one will produce an interrupt on the zero-to-one transition of the associated input signal. Alternately, programming the edge bit to a zero will produce an interrupt on the one-to-zero transition of the input signal. However, in the pulse width measurement mode, the interrupt generated by a transition on TAI or TBI will occur on the opposite transition as that normally defined by the edge bit.

For example, in the pulse width measurement mode, if the edge bit is a one, the timer will be allowed to run while the auxiliary input TAI is high. When TAI transitions from high to low, the timer will stop and, if the interrupt channel is enabled, an interrupt will occur. By having the interrupt occur on the one-to-zero transition instead of the zero-to-one transition, the processor will be interrupted when the pulse being measured has terminated and the width of the pulse is available from the timer. Therefore, the timers act like a divide-by-prescaler that can be programmed by the timer data register and the timer's A and B control register.

After reading the contents of the timer, the main counter must be reinitialized by writing to the timer data register to allow consecutive pulses to be measured. If the timer is written after the auxiliary input signal is active, the timer will count from the previous contents of the timer data register until it counts through 01 (hexadecimal). At that time, the main counter is loaded with the new value from the timer data register, a time out pulse is generated which will toggle the timer output, and an interrupt may be optionally generated on the timer interrupt channel.

Note that the pulse width measured will include counts from before the main counter was reloaded. If the timer data register is written while the pulse is transitioning to the active state, an indeterminate value may be written into the main counter.

Once the timer is reprogrammed for another mode, interrupts will again occur as normally defined by the edge bit. Note that an interrupt may be generated as the result of placing the timer into the pulse width measurement mode or by reprogramming the timer for another mode. Also, an interrupt may be generated by changing the state of the edge bit while in the pulse width measurement mode.

5.1.3. EVENT COUNT MODE OPERATION. In addition to the delay mode and the pulse width measurement mode, timers A and B may be programmed to operate in the event count mode. Like the pulse width measurement mode, the event count mode also requires an auxiliary input signal, TAI or TBI, and the interrupt channels normally associated with I4 and I3 will respond to transitions on TAI and TBI, respectively. General purpose lines I3 and I4 still function normally.

In the event count mode the prescaler is disabled, allowing each active transition on TAI and TBI to produce a count pulse. The count pulse causes the main counter to decrement by one. When the timer counts through 01 (hexadecimal), a time out pulse is generated which will cause the output signal to toggle and may optionally produce an interrupt via the associated timer interrupt channel. The timer's main counter is also reloaded from the timer data register. To count transitions reliably, the input signal may only transition once every four timer clock periods. For this reason, the input signal must have a maximum frequency equal to one-fourth that of the timer clock.

The active edge of the auxiliary input signal is defined by the associated interrupt channel's edge bit. GPIP4 of the AER specifies the active edge for TAI and GPIP3 defines the active edge for TBI. When the edge bit is programmed to a one, a count pulse will be generated on the zero-to-one transition of the auxiliary input signal. When the edge bit is programmed to a zero, a count pulse will be generated on the one-to-zero transition. Also, note that changing the state of the edge bit while the timer is in the event count mode may produce a count pulse.

Besides generating a count pulse, the active transition of the auxiliary input signal will also produce an interrupt on the I3 or I4 interrupt channel, if the interrupt channel is enabled. Typically, in the event count mode, these channels are not enabled since the timer is automatically counting transitions on the input signal. If the interrupt channel were enabled, the number of transitions could be counted in the interrupt routine without requiring the use of the timer.

5.2. TIMER REGISTERS

The four timers are programmed via three control registers and four timer data registers. Control registers TACR and TBCR and timer data registers TADR and TBDR (refer to figure 5-1) are associated with timers A and B respectively. Timers C and D are controlled by the control register TCDCR and the data registers TCDR and TDDR (refer to figure 5.2).

Figure 5.2 : Timer Data Registers.

(a) Timer A Data Register (TADR).

	7	6	5	4	3	2	1	0
Address 1F (Hex)	D7	D6	D5	D4	D3	D2	D1	D0

SET a) MPU writes a one
CLEARED a) MPU writes a zero

(b) Timer B Data Register (TBDR).

	7	6	5	4	3	2	1	0
Address 21 (Hex)	D7	D6	D5	D4	D3	D2	D1	D0

SET a) MPU writes a one
CLEARED a) MPU writes a zero

Figure 5.2 : Timer Data Registers (continued).

(c) Timer C Data Register (TCDR).

	7	6	5	4	3	2	1	0
Address 23 (Hex)	D7	D6	D5	D4	D3	D2	D1	D0

SET	a) MPU writes a one
CLEARED	a) MPU writes a zero

(d) Timer D Data Register (TDDR).

	7	6	5	4	3	2	1	0
Address 25 (Hex)	D7	D6	D5	D4	D3	D2	D1	D0

SET	a) MPU writes a one
CLEARED	a) MPU writes a zero

5.2.1. TIMER DATA REGISTERS. Each timer's main counter is an 8-bit binary down counter. The value of the main counter may be read at any time by reading the timer's data register. The information read is the value of the counter which was captured on the last low-to-high transition of the $\overline{DS}$ pin.

The main counter is initialized by writing to the timer's data register. If the timer is stopped, data is loaded simultaneously into both the timer data register and the main counter. If the timer data register is written while the timer is enabled, the value is not loaded into the timer until the timer counts through 01 (hexadecimal). Writing the timer data register while the timer is counting through 01 (hexadecimal) will cause an indeterminate value to be loaded into the timer's main counter. The four data registers are shown in figure 5.2.

5.2.2. TIMER CONTROL REGISTERS. Bits in the timer control registers select the operation mode, select the prescaler value, and disable the timers. Timer control registers TACR and TBCR also have bits which allow the programmer to reset output lines TAO and TBO. These control registers are shown in figure 5.3.

Figure 5.3 : Timer Control Registers.

(a) Timer A Control Register (TACR).

	7	6	5	4	3	2	1	0
Address 19 (Hex)	*	*	*	Reset TA0	AC3	AC2	AC1	AC0

* Unused bits read as zero

(b) Timer B Control Register (TBCR).

	7	6	5	4	3	2	1	0
Address 1B (Hex)	*	*	*	Reset TB0	BC3	BC2	BC1	BC0

* Unused bits read as zero

Reset TAO/TBO Timer's A and B output lines (TAO and TBO) may be forced low at any time by writing a one to the reset location in TACR and TBCR, respectively. The output will be held low only during the write operation ; at the conclusion of the operation, the output will be allowed to toggle in response to a time-out pulse. When resetting TAO and TBO, the remaining bits in the control register must be written with their previous value to avoid altering the operation mode.

SET a) End of write cycle which clears the bit
CLEARED a) MPU writes a zero
b) Reset

AC3-AC0, BC3-BC0 These bits are decoded to determine the timer operatio$ mode.

AC3 BC3	AC2 BC2	AC1 BC1	AC0 BC0	Operation Mode
0	0	0	0	Timer Stopped *
0	0	0	1	Delay Mode, ÷ 4 Prescaler
0	0	1	0	Delay Mode, ÷ 10 Prescaler
0	0	1	1	Delay Mode, ÷ 16 Prescaler
0	1	0	0	Delay Mode, ÷ 50 Prescaler
0	1	0	1	Delay Mode, ÷ 64 Prescaler
0	1	1	0	Delay Mode, ÷ 100 Prescaler
0	1	1	1	Delay Mode, ÷ 200 Prescaler
1	0	0	0	Event Count Mode
1	0	0	1	Pulse Width Mode, ÷ 4 Prescaler
1	0	1	0	Pulse Width Mode, ÷ 10 Prescaler
1	0	1	1	Pulse Width Mode, ÷ 16 Prescaler
1	1	0	0	Pulse Width Mode, ÷ 50 Prescaler
1	1	0	1	Pulse Width Mode, ÷ 64 Prescaler
1	1	1	0	Pulse Width Mode, ÷ 100 Prescaler
1	1	1	1	Pulse Width Mode, ÷ 200 Prescaler

* Regardless of the operation mode, counting is inhibited when the timer is stopped. The contents of the timer's main counter is not affected, although any residual count in the prescaler is lost.

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

Figure 5.3 : Timer Control Registers (continued).

(c) Timers C and D Control Register (TCDCR).

	7	6	5	4	3	2	1	0
Address 1D (Hex)	*	CC2	CC1	CC0	*	DC2	DC1	DC0

* Unused bits read as zero

CC2-CC0, DC2-DC0 The bits are decoded to determine the timer operation mode.

CC2 DC2	CC1 CC0	DC0 DC1	Operation Mode
0	0	0	Timer Stopped *
0	0	1	Delay Mode, ÷ 4 Prescaler
0	1	0	Delay Mode, ÷ 10 Prescaler
0	1	1	Delay Mode, ÷ 16 Prescaler
1	0	0	Delay Mode, ÷ 50 Prescaler
1	0	1	Delay Mode, ÷ 64 Prescaler
1	1	0	Delay Mode, ÷ 100 Prescaler
1	1	1	Delay Mode, ÷ 200 Prescaler

* When the timer is stopped, counting is inhibited. The contents of the timer's main counter is not affected, although any residual count in the prescaler is lost.

SET	a)	MPU writes a one
CLEARED	a)	MPU writes a zero
	b)	Reset

SECTION 6

UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER-TRANSMITTER

The universal synchronous/asynchronous receiver-transmitter (USART) is a single full-duplex serial channel with a double-buffered receiver and transmitter. There are separate receive and transmit clocks and separate receive and transmit status and data bytes. The receive and transmit sections are also assigned separate interrupt channels. Each section has both a normal condition interrupt channel and an error condition interrupt channel. These channels can be optionally disabled from interrupting the processor and instead, DMA transfers can be performed using the receiver ready and transmitter ready external CMFP signals.

6.1. CHARACTER PROTOCOLS

The CMFP USART supports asynchronous and with the aid of a polynomial generator checker (PGC) supports byte synchronous character formats. These formats are selected independently of the divide-by-one and divide-by-16 clock modes.

When the divide-by-one clock mode is selected, synchronization must be accomplished externally. The receiver will sample the serial data on the rising edge of the receiver clock. In the divide-by-16 clock mode, the data is sampled at mid-bit time to increase transient noise rejection.

Also, when the divide-by-16 clock mode is selected, the USART resynchronization logic is enabled. This logic increases the channel's clock skew tolerance. When a valid transition is detected, an internal counter is reset to state zero. Transition checking is then inhibited until state four. Then at state eight, the previous state of the transition checking logic is clocked into the receive shift register.

6.1.1. ASYNCHRONOUS FORMAT. Variable word length and start/stop bit configurations are available under software control for asynchronous operation. The word length can be five to eight bits and one, one and one-half, or two stop bits can be selected. The user can also select odd, even, or no parity. For character lengths of less than eight bits, the assembled character will consist of the required number of data bits followed by zeros in the unused bit positions and a parity bit, if parity is enabled.

In the asynchronous format, start bit detection is always enabled. New data is not shifted into the receive shift register until a zero bit is received. When the divide-by-16 clock mode is selected, the false start bit logic is also active. Any transition must be stable for three positive receive clock edges to be considered valid. Then a valid zero-to-one transition must not occur for at least eight additional positive clock edges.

6.1.2. SYNCHRONOUS FORMAT. When the synchronous character format is selected, the 8-bit synchronous character loaded into the synchronous character register is compared to received serial data until a match is found. Once synchronization is established, incoming data is clocked into the receiver. The synchronous word will be continuously transmitted during an underrun condition. All synchronous characters can be optionally stripped from the receive buffer. Figure 6.1 shows the synchronous character register.

The synchronous character is typically written after the data word length is selected, since unused bits in the synchronous character register are zeroed out. When parity is enabled, synchronous word length is the data word length plus one. The CMFP will compute and append the parity bit for the synchronous word when a word length of eight is selected. However, if the word length is less than eight, the user must determine the synchronous word parity and write it into synchronous character. The CMFP will then transmit the extra bit in the synchronous word as a parity bit.

Figure 6.1 : Synchronous Character Register (SCR).

	7	6	5	4	3	2	1	0
Address 27 (Hex)	D7	D6	D5	D4	D3	D2	D1	D0

6.1.3. USART CONTROL REGISTER. The USART control register (UCR) selects the clock mode and the character format for the receive and transmit sections. This register is shown in figure 6-2.

6.2. RECEIVER

As data is received on the serial input line (SI), it is clocked into an internal 8-bit shift register until the specified number of data bits have been assembled. This character will then be transferred to the receive buffer, assuming that the last word in the receiver buffer has been read. This transfer produces a buffer full interrupt to the processor.

Reading the receive buffer satisfies the buffer full condition and allows a new data word to be transferred to the receive buffer when it is assembled. The receive buffer is accessed by reading the USART data register (UDR). The UDR is simply an 8-bit data register used when transferring data from the CMFP and the CPU.

Each time a word is transferred to the receive buffer, its status information is latched into the receiver status register (RSR). The RSR is not updated again until the data word in the receive buffer has been read. When a buffer full condition exists, the RSR should always be read before the receive buffer (UDR) to maintain the correct correspondance between data and flags. Otherwise, it is possible that after reading the UDR and prior to reading the RSR, a new word could be received and transferred to the receive buffer. Its associated flags would be latched into the RSR, over-writing the flags for the previous data word. Then when the RSR were read to access the status information for the first data word, the flags for the new word would be retrieved.

Figure 6.2 : USART Control Register (UCR).

	7	6	5	4	3	2	1	0
Address 29 (Hex)	CLK	WL1	WL0	ST1	ST0	PE	E/O	WU

CLK Clock Mode. When this bit is zero, data will be clocked into and out of the receiver and transmitter at the frequency of their respective clocks. When this bit is a one, data will be clocked into and out of the receiver and transmitter at one sixteenth the frequency of their respective clocks. Also, the receiver data transition resynchronization logic will be enabled.

SET - ÷ 16 a) MPU writes a one
CLEARED - ÷ 1 a) MPU writes a zero
b) Reset

WL0, WL1 Word Length. These two bits specify the length of the data word exclusive of start bits, stop bits, and parity.

WL1	WL0	Word Length
0	0	8 Bits
0	1	7 Bits
1	0	6 Bits
1	1	5 Bits

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

ST0, ST1 Start/Stop Bit and Format Control. These two bits select the number of start and stop bits and also specify the character format.

ST1	ST0	Start Bits	Stop Bits	Format
0	0	0	0	Synchronous
0	1	1	1	Asynchronous
1	0	1	1 1/2	Asynchronous*
1	1	1	2	Asynchronous

* Only used with divide-by-16 clock mode

SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

Figure 6.2 : USART Control Register (UCR) (continued).

PE Parity Enable. When this bit is zero, no parity check will be made and no parity bit will be computed for transmission. When this bit is a one, parity will be checked by the receiver and parity will be calculated and inserted during data transmission. Note that parity is not automatically appended to the synchronous character for word lengths of less than eight bits. In this case, the parity should be written into the synchronous character register along with the synchronous word.
SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

E/O Even/Odd Parity. When this bit is zero, odd parity is selected. When this bit is a one, even parity is selected.
SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

WU Bit 0 Reserved. Must be mantained at 0.

6.2.1. RECEIVER INTERRUPT CHANNELS. The USART receive section is assigned two interrupt channels. One indicates the buffer full condition, while the other channel indicates an error condition. Error conditions include overrun, parity error, synchronous found, and break. These interrupting conditions correspond to the BF, OE, PE, and F/S or B bits of the receiver status register. These flags will function as described in 6.2.2. whether the receiver interrupt channels are enabled or disabled.

While only one interrupt is generated per character received, two dedicated interrupt channels allow separate vector numbers to be assigned for normal and abnormal receiver conditions. When a received word has an error associated with it and the error interrupt channel is enabled, an interrupt will be generated on the error channel only. However, if the error channel is disabled, an interrupt for an error condition will be generated on the buffer full interrupt channel along with interrupts produced by the buffer full condition. The receiver status register must always be read to determine which error condition produced the interrupt.

6.2.2. RECEIVER STATUS REGISTER. The receiver status register contains the receive buffer full flag, the synchronous strip enable, the receiver enable, and various status information associated with the data word in the receive buffer. The RSR is latched each time a data word is transferred to the receive buffer. RSR flags cannot change again until the data word has been read. The exception is the character in progress flag which monitors when a new word is being assembled in the asynchronous character format. The receiver status register is shown in figure 6.3.

Figure 6.3 : Receiver Status Register (RSR).

Address 2B	BF	OE	PE	FE	F/S or B	M/CIP	SS	RE

BF Buffer Full. This bit is set when a received word is transferred to the receive buffer. This bit is cleared when the receive buffer is read by accessing the USART data register (UDR). This bit is read only.
SET a) Received word transferred to buffer
CLEARED a) Receive buffer read
b) Reset

OE Overrun Error. An overrun error occurs when a received word is due to be transferred to the receive buffer, but the receive buffer is full. Neither the receive buffer nor the RSR is overwritten. The OE bit is set after the receive buffer full condition is satisfied by reading the UDR. This error condition will generate an interrupt to the processor. The OE bit is cleared by reading the RSR. New data words will not be assembled until the RSR is read.
SET a) Incoming word received and receive buffer full
CLEARED a) Receiver status register read
b) Reset

PE Parity Error. This bit is set when the word transferred to the receive buffer has a parity error. This bit is cleared when the word transferred to the receive buffer does not have a parity error.
SET a) Word in receive buffer has a parity error
CLEARED a) Word in receive buffer does not have a parity error
b) Reset

Figure 6.3 : Receiver Status Register (RSR) (continued).

FE Frame Error. A frame error exists when a non-zero data word is not followed by a stop bit in the asynchronous character format. The FE bit is set when the word transferred to the receive buffer has a frame error. The FE bit is cleared when the word transferred to the receive buffer does not have a frame error.
SET a) Word in receive buffer has a frame error
CLEARED a) Word in receive buffer does not have a frame error
b) Reset

F/S or B Found/Search or Break Detect. In the synchronous character format this bit can be set or cleared in software. When the bit is a zero, the USART receiver is placed in the search mode. The incoming data is compared to the synchronous character register (SCR) and the word length counter is disabled. The F/S bit will automatically be set when a match is found and the word length counter will be enabled. An interrupt will also be produced on the receive error channel.
SET a) Incoming word matches synchronous character
CLEARED a) MPU writes a zero
b) Incoming word does not match synchronous character
c) Reset

In the asynchronous character format, this flag indicates a break condition. A break is detected when an all zero data word with no stop bit is received. The break condition continues until a non-zero data bit is received. The 8-bit is set when the word transferred to the receive buffer is a break indication. A break condition generates an interrupt to the processor. This bit is cleared when a non-zero data bit is received and the break condition has been acknowledged by reading the RSR at least once An end of break interrupt will be generated when the bit is cleared.
SET a) Word in receive buffer is a break
CLEARED a) Break terminates and receiver status register read since beginning of break condition
b) Reset

M or CIP Match/Character in Progress In the synchronous format, this flag indicates that a synchronous character has been received. The M bit is set when the word transferred to the receive buffer matches the synchronous character register. The M bit is cleared when the word transferred to the receive buffer does not match the synchronous character register.
SET a) Word transferred to receive buffer matches the synchronous character
CLEARED a) Word transferred to receive buffer does not match synchronous character
b) Reset

In the asynchronous character format, this flag indicates that a word is being assembled. The CIP bit is set when a start bit is detected. The CIP bit is cleared when the final stop bit has been received.
SET a) Start bit is detected
CLEARED a) End of word detected
b) Reset

SS Synchronous Strip Enable. When this bit is a one, data words that match the synchronous character register will not be loaded into the receive buffer and no buffer full condition will be produced. When this bit is a zero, data words that match the synchronous character register will be transferred to the receive buffer and a buffer full condition will be produced.
SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

RE Receiver Enable. When this bit is a zero, the receiver will be immediately disabled. All flags will be cleared. When this bit is a one, normal receiver operation is enabled. This bit should not be set to a one until the receiver clock is active.
SET a) MPU writes a one
b) Transmitter is disabled in auto-turnaround mode
CLEARED a) MPU writes a zero
b) Reset

6.2.3. SPECIAL RECEIVE CONSIDERATIONS. Certain receive conditions relating to the overrun error flag and the break detect flag require further explanation. Consider the following examples :

1) A break is received while the receive buffer is full. This does not produce an overrun condition. Only the B flag will be set after the receiver buffer is read.

2) A new word is received and the receive buffer is full. A break is received before the receive buffer is read.
Both the B and OE flags will be set when the buffer full condition is satisfied.

6.3. TRANSMITTER

The transmit buffer is loaded by writing to the USART data register (UDR). The data word will be transferred to an internal 8-bit shift register when the last word in the shift register has been transmitted. This will produce a buffer empty condition. If the transmitter completes the transmission of word in the shift register before a new word is written to the transmit buffer, an underrun error will occur. In the asynchronous character format, the transmitter will send a mark until the transmit buffer is written. In the synchronous character format, the transmitter will continuously send the synchronous character.

The transmit buffer can be loaded prior to enabling the transmitter. After the transmitter is enabled, there is a delay before the first bit is output. The serial output line (SO) should be programmed to be high, low, or high impedance when the transmitter is enabled to force the output line to the desired state until the first bit is shifted out. Note that a one bit will always be transmitted prior to the word in the transmit shift register when the transmitter is first enabled.

When the transmitter is disabled, any word currently being transmitted will continue to completion. However, any word in the transmit buffer will not be transmitted and will remain in the buffer. So, no buffer empty condition will occur. If the buffer is empty when the transmitter is disabled, the buffer empty condition will remain, but no underrun condition will be generated when the word in transmission is completed. If no word is being transmitted when the transmitter is disabled, the transmitter will stop at the next rising edge of the internal shift clock.

In the asynchronous character format, the transmitter can be programmed to send a break. The break will be transmitted once the word currently in the shift register has been sent. If the shift register is empty, the break command will be effective immediately. An END interrupt will be generated at every normal character boundary to aid in timing the break transmission. The break will continue until the break command is cleared.

Any character in the transmit buffer at the start of a break will be transmitted when the break is terminated. If the transmit buffer is empty at the start of a break, it may be written at any time during the break. If the buffer is still empty at the end of the break, an underrun condition will exist.

Disabling the transmitter during a break condition causes the transmitter to cease transmission of the break character at the end of the current character. No end of break stop bit will be transmitted. Even if the transmit buffer is empty, no buffer empty condition will occur nor will an underrun condition occur. Also, any word in the transmit buffer will remain.

6.3.1. TRANSMITTER INTERRUPT CHANNELS. The USART transmit section is assigned two interrupt channels. One channel indicates a buffer empty condition and the other channel indicates an underrun or end condition. These interrupting condition correspond to the BE, UE, and END flag bits of the transmitter status register (TSR). The flag bits will function as described in 6.3.2 whether their associated interrupt channel is enabled or disabled.

6.3.2. TRANSMITTER STATUS REGISTER. The transmitter status register contains various transmitter error flags and transmitter control bits for selecting auto-turnaroundand loopback mode. The TSR is shown in figure 6.4.

Figure 6.4 : Transmitter Status Register (TSR).

Address 2D	BE	UE	AT	END	B	H	L	TE

BE Buffer Empty. This bit is set when the word in the transmit buffer is transferred to the transmit shift register. This bit is cleared when the transmit buffer is reloaded by writing to the USART data register (UDR).
SET a) Transmit buffer contents transferred to transmit shift register
CLEARED a) Transmit buffer written

UE Underrun Error. This bit is set when the word in the transmit shift register has been transmitted before a new word is loaded into the transmit buffer. This bit is cleared by reading the TSR or by disabling the transmitter. This bit does not need to be cleared before writing to the UDR.
SET a) Transmit shift register contents transmitted before transmit buffer written
CLEARED a) Transmitter status register read
b) Transmitter disabled

AT Auto-Turnaround. When this bit is set, the receiver will be enabled automatically after the transmitter has been disabled and the last character being transmitted is completed.
SET a) MPU writes a one
CLEARED a) Transmitter disabled

END End of Transmission. When the transmitter is disabled while a character is being transmitted, the END will be set after the character transmission is complete. If no word is being transmitted when the transmitter is disabled, the END bit will be set immediately. The END bit is cleared by reenabling the transmitter.
SET a) Transmitter disabled
CLEARED a) Transmitter enabled

B Break. This bit has no function in the synchronous character format. In the asynchronous character format, when this bit is set to a one, a break will be transmitted upon the completion of the transmission of any word in the transmit shift register. A break consists of an all zero data word with no stop bit. When this bit is cleared by software, the break indication will cease and normal transmission will resume. Note that when B is set, BE cannot be set.
SET a) MPU writes a one
CLEARED a) MPU writes a zero

H, L High and Low. These control bits configure the transmitter output (SO) when the transmitter is disabled. These bits also force the transmitter output after the transmitter is enabled until END is cleared.

H	L	Output State
0	0	High Impedance
0	1	Low
1	0	High
1	1	Loopback Mode

Loopback mode internally connects the transmitter output to the receiver input and the transmitter clock to the receiver clock internally. The receiver clock (RC) and the serial input (SI) are not used. When the transmitter is disabled, SO is forced high.
SET a) MPU writes a one
CLEARED a) MPU writes a zero

TE Transmitter Enable. When this bit is cleared, the transmitter is disabled. The UE bit will be cleared and the END bit will be set. When this bit is set, the transmitter is enabled. The transmitter output will be driven according to the H and L bits until transmission begins. A one bit will be transmitted before the transmission of the word in the transmit shift register is begun.
SET a) MPU writes a one
CLEARED a) MPU writes a zero
b) Reset

6.4. DMA OPERATION

USART error conditions are only valid for each character boundary. When the USART performs block data transfers by using the DMA handshake lines RR (receiver ready) and TR (transmitter ready), errors must be saved and checked at the end of a block. This is accomplished by enabling the error channel for the receiver or transmitter and by masking interrupts for this channel. Once the transfer is complete, interrupt pending register A is read. Any pending receiver or transmitter error indicates an error in the data transfer.

SECTION 7

ELECTRICAL CHARACTERISTICS

This section contains the electrical specifications and associated timing information for the TS68HC901 multi-function peripheral.

7.1. MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_{CC}	Supply Voltage	– 0.3 to + 7.0	V
V_{IN}	Input Voltage	– 0.3 to + 7.0	V
T_A	Operating Temperature Range TS68HC901C TS68HC901V	T_L to T_H 0 to + 70 – 40 to + 85	°C
T_{stg}	Storage Temperature	– 65 to + 150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{CC} or GND).

7.2. THERMAL DATA

θJA	Thermal Resistance Plastic	50	°C/W

7.3. POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from :

$$T_J = T_A + (P_D \bullet \theta_{JA}) \qquad (1)$$

Where :

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to Ambient, °C/W
$P_D = P_{INT} + P_{I/O}$
$P_{INT} = I_{CC} \times V_{CC}$, Watts - Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins - User Determined

For most applications $P_{I/O}$ P_{INT} and can be neglected.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is :

$$P_D = K + (T_J + 273°C) \qquad (2)$$

Solving equations 1 and 2 for K gives :

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} P_D^2 \qquad (3)$$

Where :

K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

7.4. DC ELECTRICAL CHARACTERISTICS
($T_A = T_L$ to T_H $V_{CC} = +5V \pm 5\%$, unless otherwise noted)

Symbol	Parameter	Min.	Max.	Unit
V_{IH}	Input HIgh Voltage Except XTAL1, XTAL2	2.0	V_{DD} + 0.3	V
V_{IH}	Input High Voltage XTAL1, XTAL2	V_{DD} – 1.5	V_{DD} + 0.3	V
V_{IL}	Input Low Voltage	– 0.3	0.8	V
V_{OH}	Output High Voltage, Except $\overline{\text{DTACK}}$ (I_{OH} = - 120 µA)	4.1		V
V_{OL}	Output Low Voltage, Except $\overline{\text{DTACK}}$ (I_{OL} = 2.0 mA)		0.5	V
I_{CC}	Power Supply Current (outputs open)		6	mA
I_{LI}	Input Leakage Current (V_{in} = 0 to V_{CC})		± 10	µA
I_{LOH}	Hi-Z Output Leakage Current in Float (V_{out} = 2.4 to V_{CC})		10	µA
I_{LOL}	Hi-Z Output Leakage Current in Float (V_{out} = 0.5 V)		– 10	µA
I_{OH}	$\overline{\text{DTACK}}$ Output Source Current (V_{out} = 2.4 V)		– 400	µA
I_{OL}	$\overline{\text{DTACK}}$ Output Sink Current (V_{out} = 0.5 V)		5.3	mA
P_D	Power Dissipation		32	mW

7.5. CAPACITANCE (T_A = 25°C, f = 1MHz, unmeasured pins returned to ground)

Symbol	Parameter	Min.	Max.	Unit
C_{IN}	Input Capacitance		10	pF
C_{OUT}	Hi-Z Output Capacitance		10	pF

Figure 7.1 : $\overline{\text{IRQ}}$ Test Load.

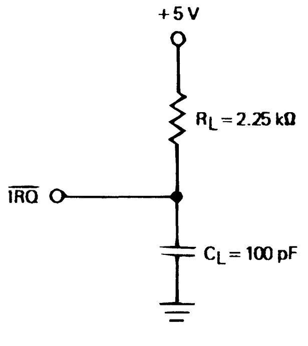
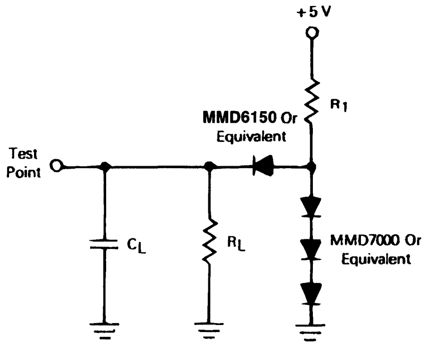


Figure 7.2 : Typical Test Load.



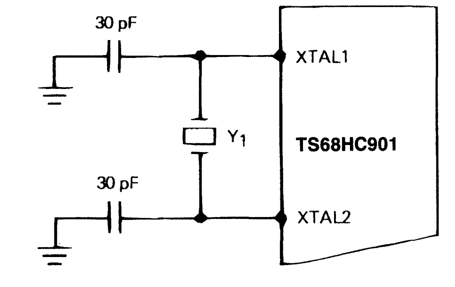
for all outputs except $\overline{\text{DTACK}}$
C_L = 100pF
R_L = 20kΩ
R_1 = 1.90kΩ

for $\overline{\text{DTACK}}$
C_L = 130pF
R_L = 6kΩ
R_1 = 740Ω

7.6. CLOCK TIMING

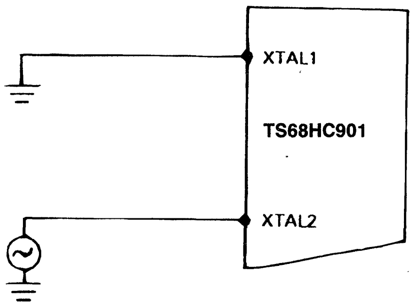
Symbol	Parameter	4 MHz		5 MHz		8 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
f	Frequency of Operation	1.0	4.0	1.0	5.0	1.0	8.0	MHz
t_{cyc}	Cycle Time	250	1000	200	1000	125	1000	ns
t_{CL}, t_{CH}	Clock Pulse Width	110	480	90	480	55	480	ns
t_{Cr}, t_{Cf}	Rise and Fall Times	–	15	–	10	–	10	ns

Figure 7.3 : CMFP External Oscillator Components.



Crystal Parameters
Parallel resonance fundamental mode AT cut
$R_S \le 150\Omega$ (f = 2.8 - 4.0MHz)
$R_S \le 300\Omega$ (f = 2.0 - 2.7MHz)
C_L = 18pF, C_M = 0.02pF, C_R = 5pF, L_M = 96MHz
f (typical) = 2.4576MHz

Figure 7.3.1 : CMFP External Clock Connection.



Other possible configuration :
XTAL1 driven with a CMOS clock and XTAL2 not connected.

7.7. AC ELECTRICAL CHARACTERISTICS T_{amb} = 0°C to 70°C, V_{CC} = 5.0 V_{DC} ± 5%, V_{SS} = 0 V_{DC} (unless otherwise specified). See figures 7-4 through 7-10.

N°	Parameter	4 MHz		5 MHz		8 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
1	$\overline{\text{CS}}$, $\overline{\text{DS}}$ Width High	50		35		25		
2	R/$\overline{\text{W}}$, A1-A5 Valid to Falling $\overline{\text{CS}}$ (setup)	30		25		20		ns
3	Data Valid Prior to Rising $\overline{\text{DS}}$	280		150		100		ns
4[(3)]	$\overline{\text{CS}}$, $\overline{\text{IACK}}$ Valid to Falling Clock (setup)	50		50		50		ns
4a [(4)]	Falling Clock to Next $\overline{\text{CS}}$ Low	100		80		50		ns
5	CLK Low to $\overline{\text{DTACK}}$ Low		220		180		90	ns
6	$\overline{\text{CS}}$, $\overline{\text{DS}}$ or $\overline{\text{IACK}}$ High to $\overline{\text{DTACK}}$ High		60		55		50	ns
7	$\overline{\text{CS}}$, $\overline{\text{DS}}$ or $\overline{\text{IACK}}$ High to $\overline{\text{DTACK}}$ Tri-state		100		100		100	ns
8	$\overline{\text{DTACK}}$ Low to Data Invalid (hold time)	0		0		0		ns
9	$\overline{\text{CS}}$, $\overline{\text{DS}}$ or $\overline{\text{IACK}}$ High to Data Tri-state		50		50		50	ns
10	$\overline{\text{CS}}$ or $\overline{\text{DS}}$ High to R/$\overline{\text{W}}$, A1-A5 Invalid (hold time)	0		0		0		ns

7.7. AC ELECTRICAL CHARACTERISTICS (continued)

N°.	Parameter	4 MHz		5 MHz		8 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
11[3,5]	Data Valid from $\overline{\text{CS}}$ Low		310		260		200	ns
12	Read Data Valid to $\overline{\text{DTACK}}$ Low (setup time)	50		50		20		ns
13	$\overline{\text{DTACK}}$ Low to $\overline{\text{DS}}$, $\overline{\text{CS}}$ or $\overline{\text{IACK}}$ High (hold time)	0		0		30		ns
14	$\overline{\text{IEI}}$ Low to Falling $\overline{\text{CLK}}$ (setup)	50		50		50		ns
15[1]	$\overline{\text{IEO}}$ Valid from Clock Low (delay)		180		180		120	ns
16	Data Valid from Clock Low (delay)		300		300		180	ns
17	$\overline{\text{IEO}}$ Invalid from $\overline{\text{IACK}}$ High (delay)		150		150		100	ns
18	$\overline{\text{DTACK}}$ Low from Clock High (delay)		180		165		100	ns
19[1]	$\overline{\text{IEO}}$ Valid from $\overline{\text{IEI}}$ Low (delay)		100		100		100	ns
20	Data Valid from $\overline{\text{IEI}}$ Low (delay)		220		220		195	ns
21	Clock Cycle Time	250	1000	200	1000	125	1000	ns
22	Clock Width Low	110		90		55		ns
23	Clock Width High	110		90		55		ns
24[4]	$\overline{\text{DS}}$ Inactive to Rising Clock (setup)	100		80		50		ns
25	I/O Minimun Active Pulse Width	100		100		100		ns
26	$\overline{\text{IACK}}$ Width High/Minimun Delay between two Pulses	2		2		2		CLK
27	I/O Data Valid from Rising $\overline{\text{CS}}$ or $\overline{\text{DS}}$		450		450		350	ns
28	Receiver Ready Delay from Falling RC		600		600		200	ns
29	Transmitter Ready Delay from Falling TC		600		600		200	ns
30[6]	Timer Ouput Low from Rising Edge of $\overline{\text{CS}}$ or $\overline{\text{DS}}$ (A & B) (reset T_{OUT})		450		450		200	ns
31[2]	T_{OUT} Valid from Internal Timeout		2 t_{CLK} + 300		2 t_{CLK} + 300		2 t_{CLK} + 300	ns
32	Timer Clock Low Time	110		90		55		ns
33	Timer Clock High Time	110		90		55		ns
34	Timer Clock Cycle Time	250	1000	200	1000	125	1000	ns
35	$\overline{\text{RESET}}$ Low Time	2		1.8		1.5		µs
36	Delay to Falling $\overline{\text{INTR}}$ from External Interrupt Active Transition		380		380		250	ns
37	Transmitter Internal Interrupt Delay from Falling Edge of TC		550		550		350	ns
39	Receiver Buffer Full Interrupt Transition Delay from Rising Edge of RC		800		800		400	ns
38	Receiver Error Interrupt Transition Delay from Falling Edge of RC		800		800		400	ns

Notes :

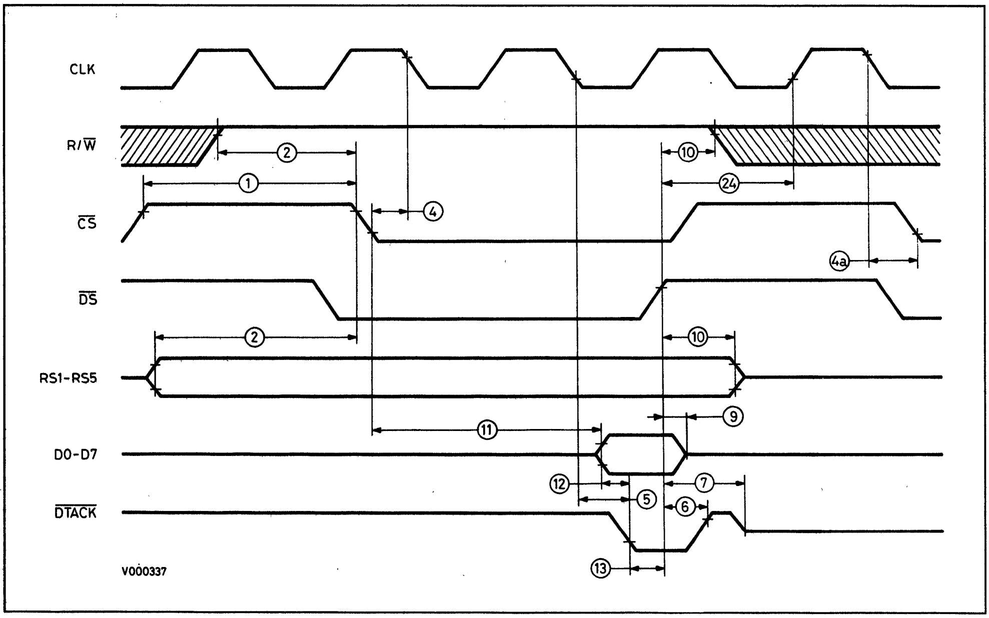
1. $\overline{\text{IEO}}$ only goes low if no acknowledgeable interrupt is pending. If $\overline{\text{IEO}}$ goes low, $\overline{\text{DTACK}}$ and the data bus remain tri-stated.
2. T_{CLK} refers to the clock applied to the CMFP CLK input pin. t_{CLK} refers to the timer clock signal regardless of whether that signal comes from the XTAL1/XTAL2 crystal clock inputs or the TAI or TBI timer inputs.
3. If the setup time is not met, $\overline{\text{CS}}$ or $\overline{\text{IACK}}$ will not be recognized until the next falling CLK.
4. If the setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.
5. Althrough $\overline{\text{CS}}$ and $\overline{\text{DTACK}}$ are synchronized with the clock, the data out during a read cycle is asynchronous to the clock, relying only on $\overline{\text{CS}}$ for timing.
6. Spec. 30 applies to timer outputs TAO and TBO only.

7.7. AC ELECTRICAL CHARACTERISTICS (continued)

N°.	Parameter	4 MHz		5 MHz		8 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
40	Serial in Set Up Time to Rising Edge of RC (divide by one only)	80		70		50		ns
41	Data Hold Time from Rising Edge of RC (divide by one only)	350		325		100		ns
42	Serial Output Data Valid from Falling Edge of TC (÷ 1)		440		420		200	ns
43	Transmitter Clock Low Time	500		450		250		ns
44	Transmitter Clock High Time	500		450		250		ns
45	Transmitter Clock Cycle Time	1.05		0.95		0.55		µs
46	Receiver Clock Low Time	500		450		250		ns
47	Receiver Clock High Time	500		450		250		ns
48	Receiver Clock Cycle Time	1.05		0.95		0.55		µs
49[2]	$\overline{CS}$, $\overline{IACK}$, $\overline{DS}$ Width Low		80		80		80	T_{CLK}
50	Serial Output Data Valid from Falling Edge of TC (÷ 16)		490		370		250	ns

Note : 2. T_{CLK} refers to the clock applied to the CMFP CLK input pin. t_{CLK} refers to the timer clock signal regardless of whether that signal comes from the XTAL1/XTAL2 crystal clock inputs or the TAI or TBI timer inputs.

Figure 7.4 : Read Cycle Timing.



7.7.1. AC ELECTRICAL CHARACTERISTICS - READ CYCLES
(V_{CC} = 5.0 V_{DC} ± 5%, V_{SS} = 0 V_{DC}, T_A = T_L to T_H unless otherwise noted)

N°.	Parameter	4 MHZ		5 MHZ		8 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
1	$\overline{CS}$, $\overline{DS}$ Width High	50		35		25		
2	R/$\overline{W}$, A1-A5 Valid to Falling $\overline{CS}$ (setup)	30		25		20		ns
4(3)	$\overline{CS}$, $\overline{IACK}$ Valid to Falling Clock (setup)	50		50		50		ns
4a(4)	Falling Clock to Next $\overline{CS}$ Low	100		80		50		ns
5	CLK Low to $\overline{DTACK}$ Low		220		180		90	ns
6	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ High		60		55		50	ns
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state		100		100		100	ns
9	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to Data Tri-state		50		50		50	ns
10	$\overline{CS}$ or $\overline{DS}$ High to R/$\overline{W}$, A1-A5 Invalid (hold time)	0		0		0		ns
11(3,5)	Data Valid from $\overline{CS}$ Low		310		260		200	ns
12	Read Data Valid to $\overline{DTACK}$ Low (setup time)	50		50		20		ns
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (hold time)	0		0		0		ns
24(4)	$\overline{DS}$ Inactive to Rising Clock (setup)	100		80		50		ns

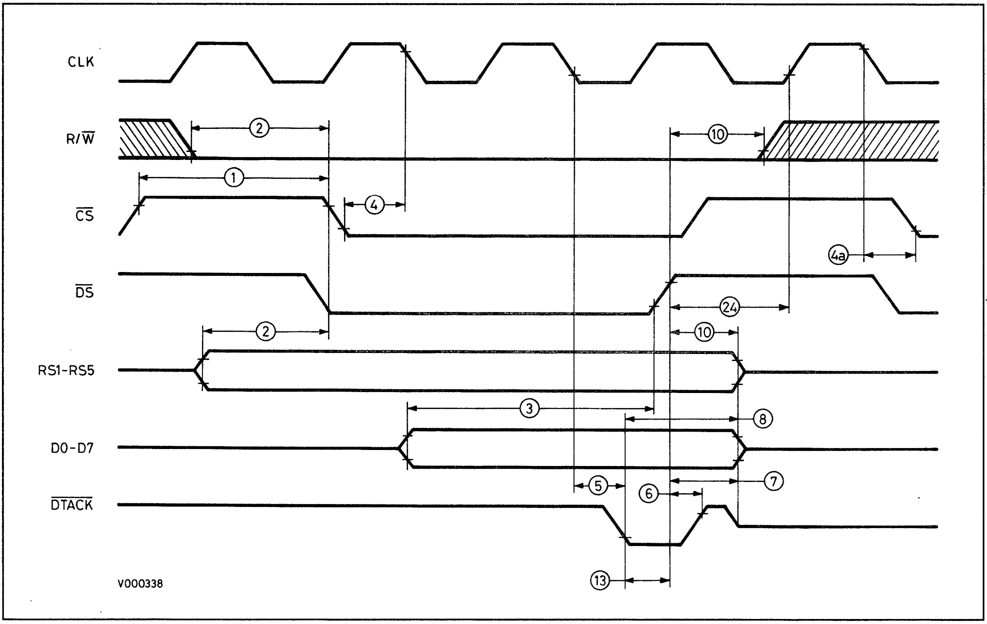
Notes : 3. If the setup time is not met, $\overline{CS}$ or $\overline{IACK}$ will not be recognized until the next falling CLK.
4. If the setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.
5. Althrough $\overline{CS}$ and $\overline{DTACK}$ are synchronized with the clock, the data out during a read cycle is asynchronous to the clock, relying only on CS for timing.

7.7.2. AC ELECTRICAL CHARACTERISTICS - WRITE CYCLES
(V_{CC} = 5.0 V_{DC} ± 5%, V_{SS} = 0 V_{DC}, T_A = T_L to T_H unless otherwise noted)

N°.	Parameter	4 MHZ		5 MHZ		8 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
1	$\overline{CS}$, $\overline{DS}$ Width High	50		35		25		
2	R/$\overline{W}$, A1-A5 Valid to Falling $\overline{CS}$ (setup)	30		25		20		ns
3	Data Valid Prior to Rising $\overline{DS}$	280		150		100		ns
4(3)	$\overline{CS}$, $\overline{IACK}$ Valid to Falling Clock (setup)	50		50		50		ns
4a(4)	Falling Clock to Next $\overline{CS}$ Low	100		80		50		ns
5	CLK Low to $\overline{DTACK}$ Low		220		180		90	ns
6	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ High		60		55		50	ns
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state		100		100		100	ns
8	$\overline{DTACK}$ Low to Data Invalid (hold time)	0		0		0		ns
10	$\overline{CS}$ or $\overline{DS}$ High to R/$\overline{W}$, A1-A5 Invalid (hold time)	0		0		0		ns
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (hold time)	0		0		0		ns
24(4)	$\overline{DS}$ Inactive to Rising Clock (setup)	100		80		50		n

Notes : 3. If the setup time is not met, $\overline{CS}$ or $\overline{IACK}$ will not be recognized until the next falling CLK.
4. If the setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.

Figure 7.5 : Write Cycle Timing.



7.7.3. AC ELECTRICAL CHARACTERISTICS - INTERRUPT ACKNOWLEDGE CYCLES
(V_{CC} = 5.0 V_{DC} ± 5%, V_{SS} = 0 V_{DC}, T_A = T_L to T_H - C unless otherwise noted)
See Figures 7.6 and 7.7.

N°.	Parameter	4 MHz		5 MHz		8 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
4(3)	$\overline{CS}$, $\overline{IACK}$ Valid to Falling Clock (setup)	50		50		50		ns
5	$\overline{CLK}$ Low to $\overline{DTACK}$ Low		220		180		90	ns
6	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ High		60		55		50	ns
7	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to $\overline{DTACK}$ Tri-state		100		100		100	ns
9	$\overline{CS}$, $\overline{DS}$ or $\overline{IACK}$ High to Data Tri-state		50		50		50	ns
13	$\overline{DTACK}$ Low to $\overline{DS}$, $\overline{CS}$ or $\overline{IACK}$ High (hold time)	0		0		0		ns
14	$\overline{IEI}$ Low to Falling CLK (setup)	50		50		50		ns
15(1)	$\overline{IEO}$ Valid from Clock Low (delay)		180		180		120	ns
16	Data Valid from Clock Low (delay)		300		300		180	ns
17	$\overline{IEO}$ Invalid from IACK High (delay)		150		150		100	ns
18	$\overline{DTACK}$ Low from Clock High (delay)		180		165		100	ns
19(1)	$\overline{IEO}$ Valid from $\overline{IEI}$ Low (delay)		100		100		100	ns
20	Data Valid from $\overline{IEI}$ Low (delay)		220		200		195	ns
21	Clock Cycle Time	250	1000	200	1000	125	1000	ns
22	Clock Width Low	110		90		55		ns
23	Clock Width High	110		90		55		ns
24(4)	$\overline{DS}$, Inactive to Rising Clock (setup)	100		80		50		ns
25	I/O Minimum Active Pulse Width	100		100		100		ns
26	$\overline{IACK}$ Width High/minimun Delay between two Pulses	2		2		2		CLK

Notes : 1. $\overline{IEO}$ only goes low if no acknowledgeable interrupt is pending. If $\overline{IEO}$ goes low, $\overline{DTACK}$ and the data bus remain tri-stated.
3. If the setup time is not met, $\overline{CS}$ or $\overline{IACK}$ will not be recognized until the next falling CLK.
4. If the setup time is met (for consecutive cycles), the minimum hold-off time of one clock cycle will be obtained. If not met, the hold-off will be two clock cycles.

Figure 7.6 : Interrupt Acknowledge Cycle ($\overline{IEI}$ Low).

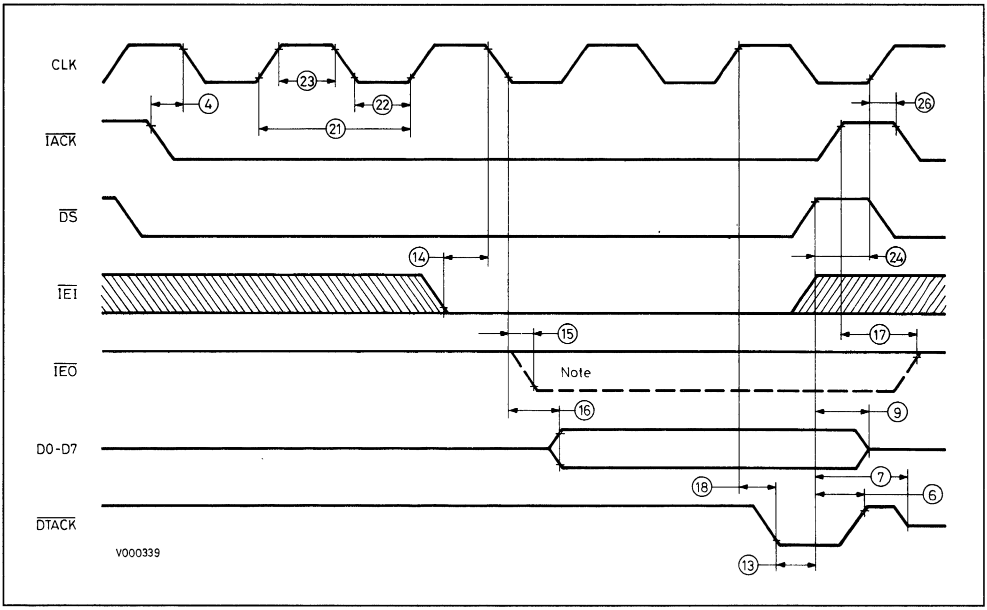
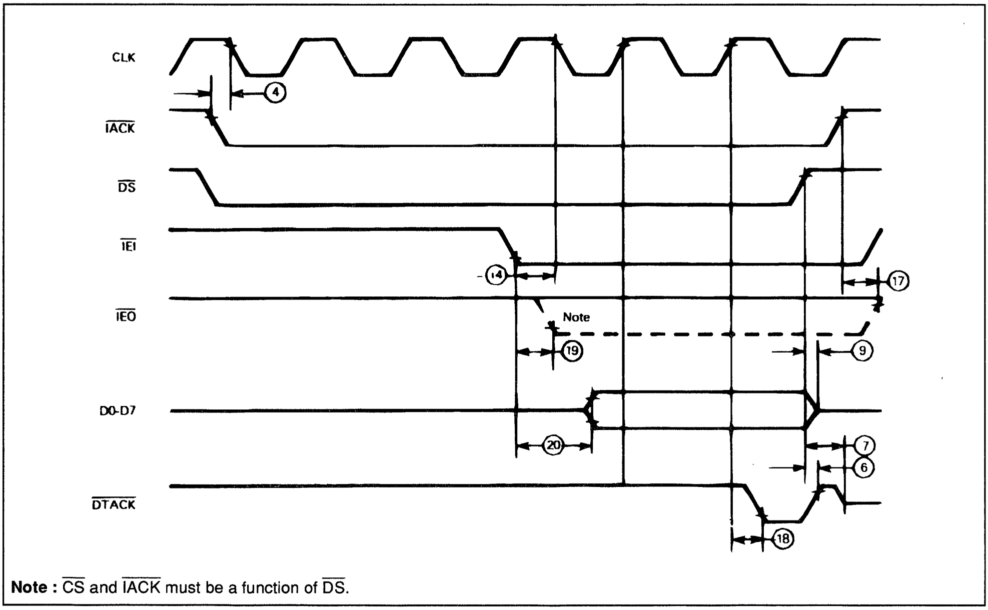


Figure 7.7 : Interrupt Acknowledge Cycle ($\overline{IEI}$ High)

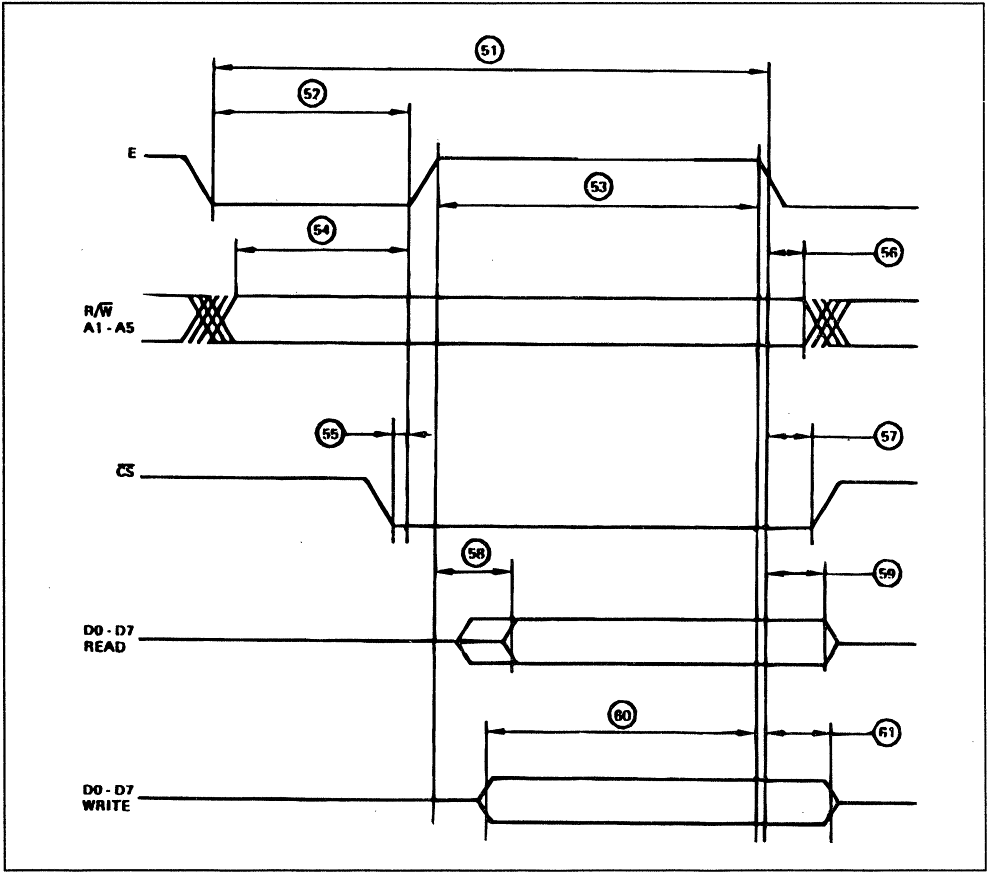


Note : $\overline{CS}$ and $\overline{IACK}$ must be a function of $\overline{DS}$.

7.7.4. AC ELECTRICAL CHARACTERISTICS - 6800 INTERFACE TIMING (V_{CC} = 5.0 V_{DC} ± 5%, V_{SS} = 0 V_{DC}, T_A = 0°C to 70°C unless otherwise noted). See figure 7.8.

N°.	Parameter	Min.	Max.	Unit
51	Cycle Time	1000		ns
52	Pulse Width, E High	430		ns
53	Pulse Width, E Low	450		ns
54	Address, R/$\overline{W}$ Setup Time Before E	80		ns
55	$\overline{CS}$ Setup Time Before E	80		ns
56	Address Hold Time	10		ns
57	$\overline{CS}$ Hold Time	10		ns
58	Output Data Delay Time (read)		250	ns
59	Data Hold Time (read)	0	100	ns
60	Input Data Setup Time (write)	280		ns
61	Data Hold Time (write)	20		ns

Figure 7.8 : 6800 Interfacing Timing.



7.7.5. AC ELECTRICAL CHARACTERISTICS - MULTIPLEXED BUS TIMING (V_{CC} = 5.0 V_{DC} ± 5%, V_{SS} = 0 V_{DC}, T_A = 0°C to 70°C unless otherwise noted). See figures 7.9, 7.10.

N°.	Parameter	Min.	Max.	Unit
62	Cycle Time	800		ns
63	Pulse Width $\overline{DS}$ Low or RD/$\overline{WR}$ High	350		ns
64	Pulse Width $\overline{DS}$ High or RD/$\overline{WR}$ Low	340		ns
65	Pulse Width AS/ALE High	100		ns
66	Delay AS Fall to $\overline{DS}$ Rise or ALE Fall to RD/WR Fall	30		ns
67	Delay $\overline{DS}$ or RD/$\overline{WR}$ Rise to AS/ALE Rise	30		ns
68	R/$\overline{W}$ Setup Time to $\overline{DS}$	100		ns
69	R/$\overline{W}$ Hold Time to $\overline{DS}$	10		ns
70	Address Setup Time to AS/ALE	20		ns
71	Address Hold Time to AS/ALE	20		ns
72	Data Setup Time to $\overline{DS}$ or $\overline{WR}$ (write)	280		ns
73	Delay Data to $\overline{DS}$ or RD (read)		250	ns
74	Data Hold Time to $\overline{DS}$ or WR (write)	20		ns
75	Data Hold Time to $\overline{DS}$ or RD (read)	0	100	ns
76	CE Setup TIme to AS/ALE Fall	20		ns
77	CE Hold Time to $\overline{DS}$, RD or $\overline{WR}$	20		ns

Figure 7.9 : Multiplexed Bus Timing Motorola Type.

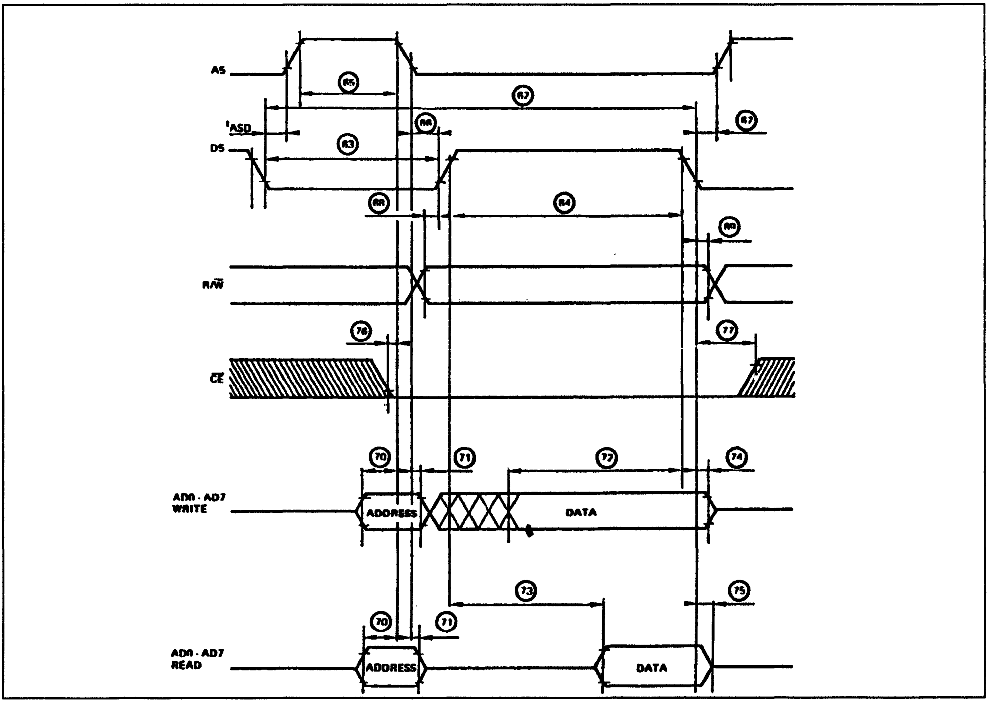


Figure 7.10 : Multiplexed Bus Timing - Intel Type.

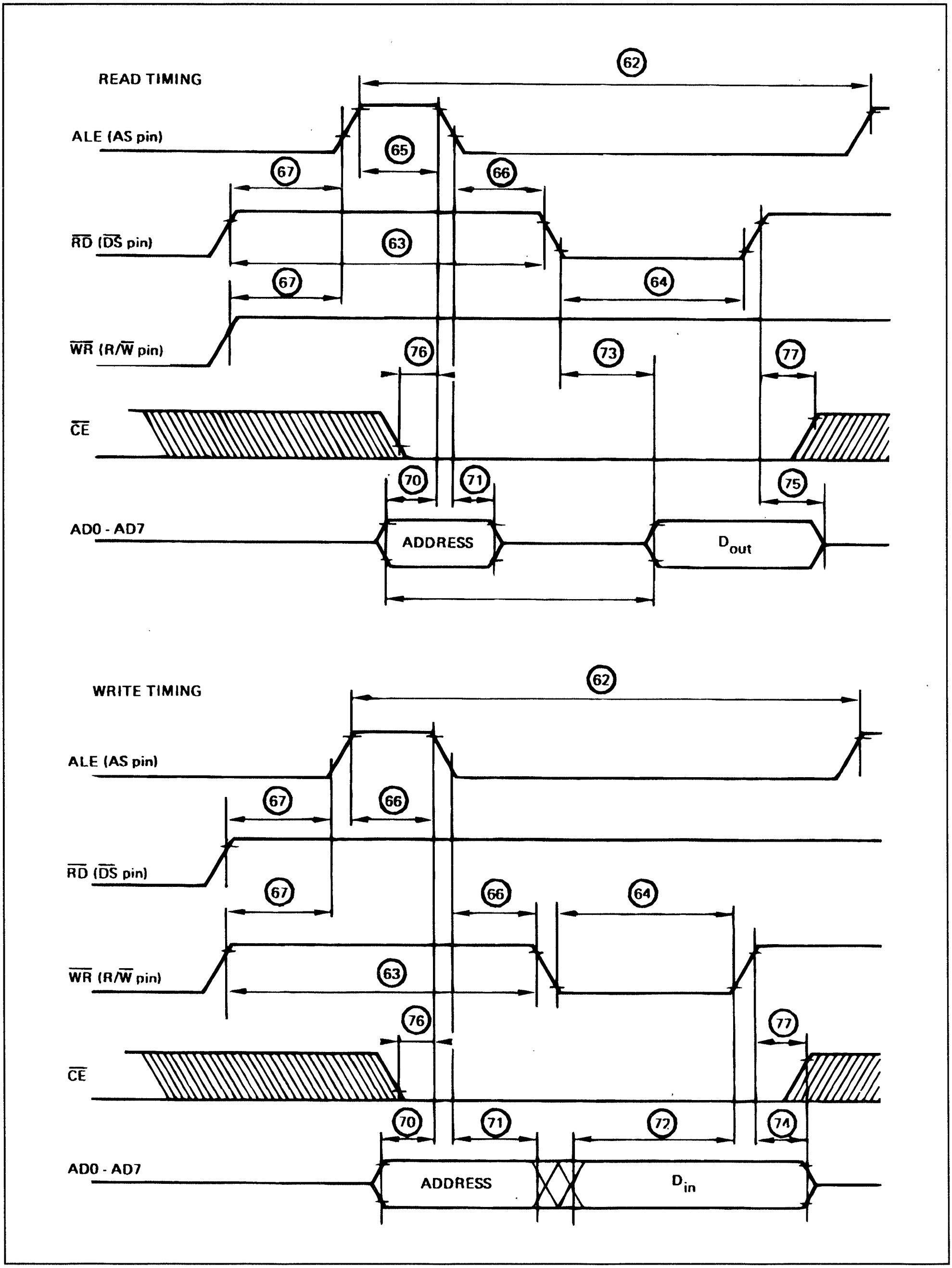
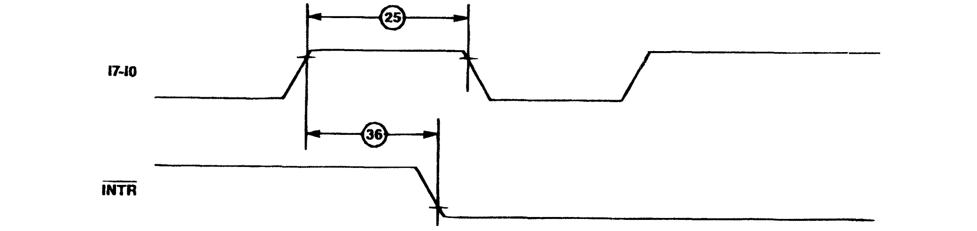


Figure 7.11 : Interrupt Timing.



Note : Active edge is assumed to be the rising edge.

Figure 7.12 : Port Timing.

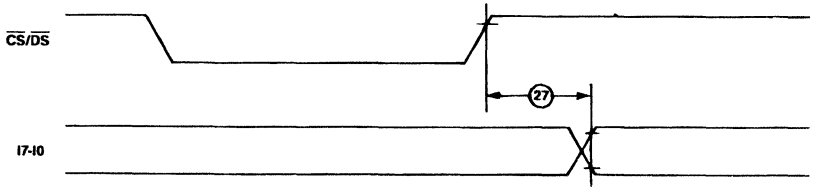


Figure 7.13 : Receiver Timing.

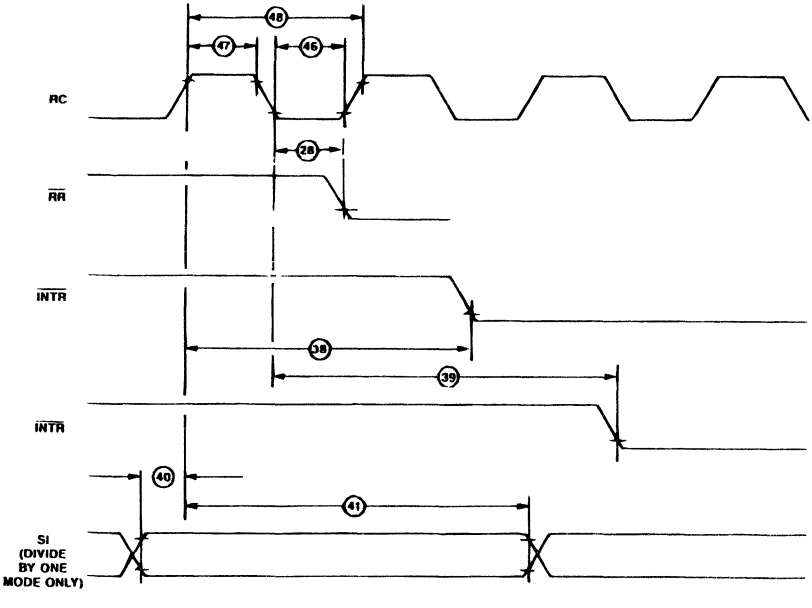


Figure 7.14 : Transmitter Timing.

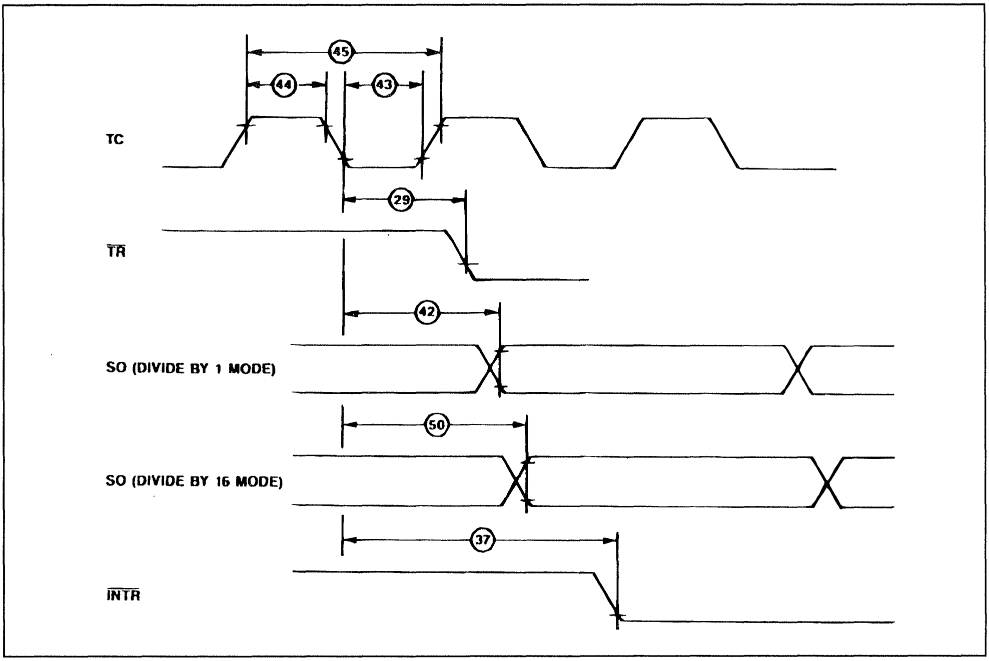


Figure 7.15 : Timer Timing.

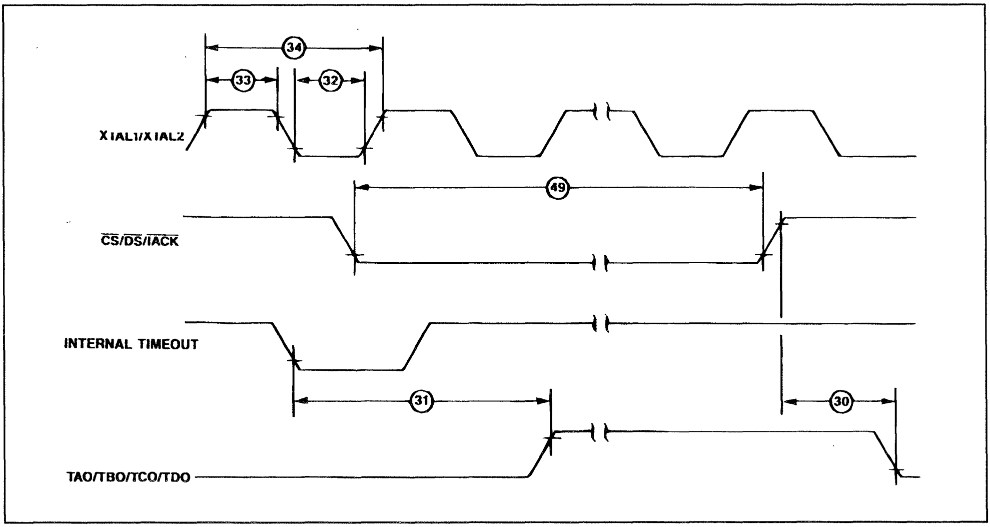
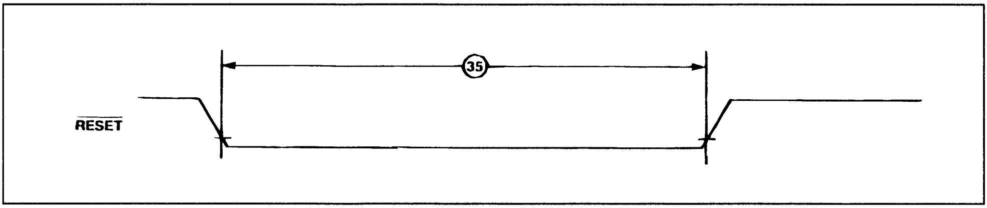


Figure 7.16 : Reset Timing.



7.8. TIMER AC CHARACTERISTICS

DEFINITION

Parameter
Error = Indicated Time Value - Actual Time Value
$t_{psc} = t_{CLK}$ x Prescale Value

INTERNAL TIMER MODE

Parameter	Value
Single Interval Error (free running) (see note 2)	± 100ns
Cumulative Internal Error	0
Error between Two Timer Reads	$\pm (t_{psc} - 4\ t_{CLK})$
Start Timer to Stop Timer Error	$2\ t_{CLK}$ + 100ns to $-(t_{psc} + 6\ t_{CLK}$ + 100ns)
Start Timer to Read Timer Error	0 to $-(t_{psc} + 6\ t_{CLK}$ + 400ns)
Start Timer to Interrupt Request Error (see note 3)	$-2\ t_{CLK}$ to $-(4\ t_{CLK}$ + 800 ns)

PULSE WIDTH MEASUREMENT MODE

Parameter	Value
Measurement Accuracy (see note 1)	$2\ t_{CLK}$ to $-(t_{psc} + 4\ t_{CLK})$
Minimum Pulse Width	$4\ t_{CLK}$

EVENT COUNTER MODE

Parameter	Value
Minimum Active Time of TAI and TBI	$4\ t_{CLK}$
Minimum Inactive Time of TAI and TBI	$4\ t_{CLK}$

Notes : 1. Error may be cumulative if repetitively performed.
2. Error with respect to t_{out} or IRQ if note 3 is true.
3. Assuming it is possible for the timer to make an interrupt request immediately.

7.9. FREQUENCY RANGE SUMMARY

The following table shows the maximum operating frequency of the TS68HC901 internal peripherals, according to the type used.

Type	8 MHz	5 MHz	4 MHz	Unit
Timer	8	5	4	MHz
USART	2	1.1	1	MHz
68000 Interface	8	5	4	MHz
6800 Interface	1	1	1	MHz
Multiplexed Interface	1.25	1.25	1.25	MHz

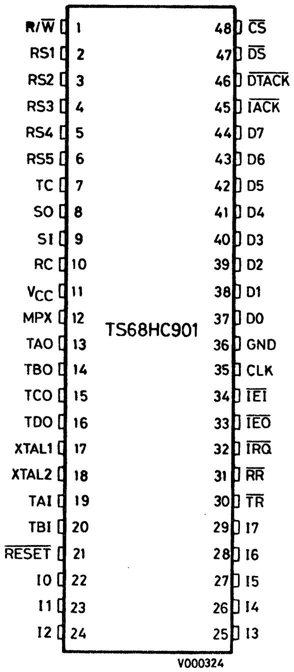
SECTION 8

MECHANICAL DATA AND ORDERING INFORMATION

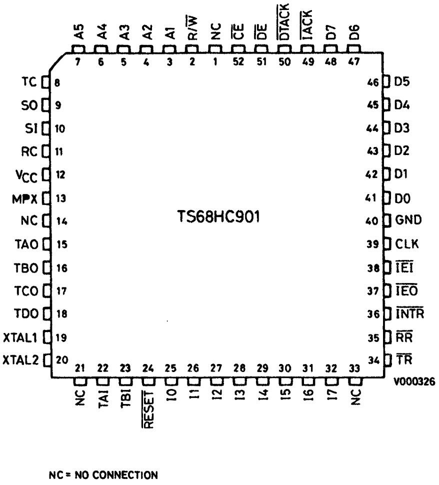
This section contains the pin assignments, package dimensions, and ordering information for the TS68HC901.

8.1. PIN ASSIGNMENTS

48-Pin Dual in-line.



52-Pin Quad Pack (PLCC).

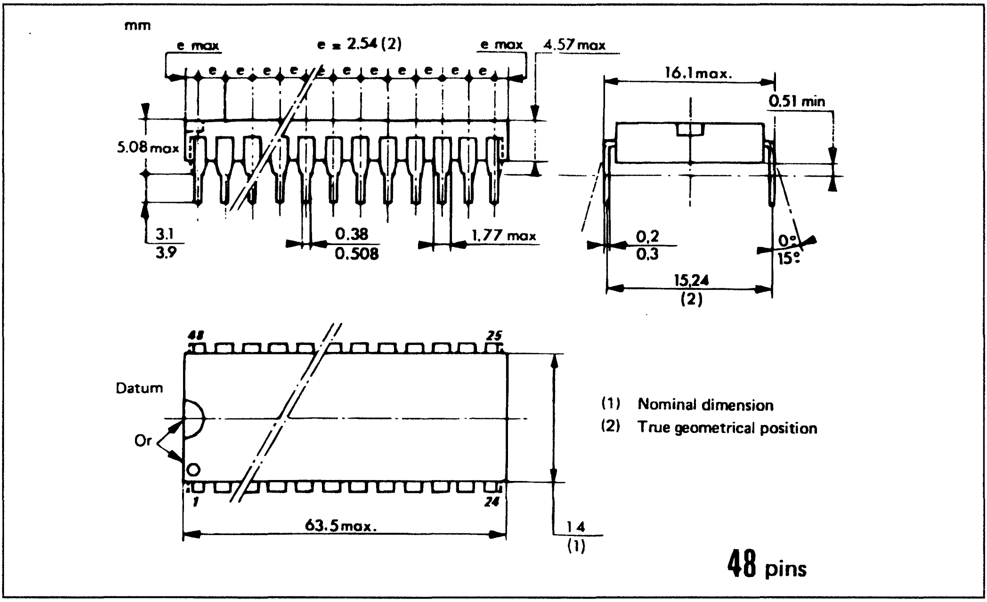


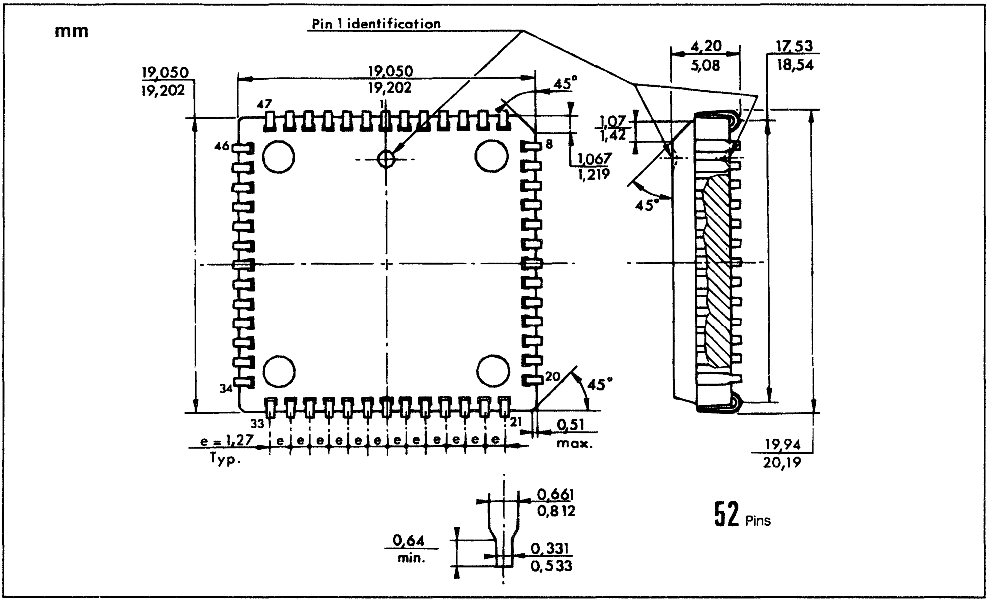
ORDERING INFORMATION

STANDARD VERSIONS

Part Number	Frequency (MHz)	Temperature Range	Package Type
TS68HC901CP4 TS68HC901CP5 TS68HC901CP8	4.0 5.0 8.0	0 °C to + 70 °C 0 °C to + 70 °C 0 °C to + 70 °C	Plastic DIL P Suffix
TS68HC901FN4 TS68HC901FN5 TS68HC901FN8	4.0 5.0 8.0	0 °C to + 70 °C 0 °C to + 70 °C 0 °C to + 70 °C	PLCC FN Suffix

8.2. PACKAGE MECHANICAL DATA





SGS-THOMSON
MICROELECTRONICS

Z8530

SERIAL COMMUNICATIONS CONTROLLER

Thank you for your interest in the SCC, one of the most versatile and most popular Serial Data Communications ICs. This document is intended to provide answers to all technical questions about the Z8530 Serial Communications Controller. Please ready this Preface where we try to anticipate your questions.

- If you are new to serial data communications, you will need additional tutorial information. Of the many introductory texts on this subject, *Technical Aspects of Data Communications* by John E. McNamara, published by Digital Press (DEC) 1982, is one of the best.
- If you have designed with simpler UARTs and USARTs, and HDLC/SDLC devices, the SCC offers you far greater flexibility, but also requires an in-depth study and understanding of the impact and the use of its many powerful features. This manual contains important information.
- If you are familiar with the Z80-SIO, you will feel right at home with the SCC, for it is really a functionally enhanced superset of the Z80-SIO.

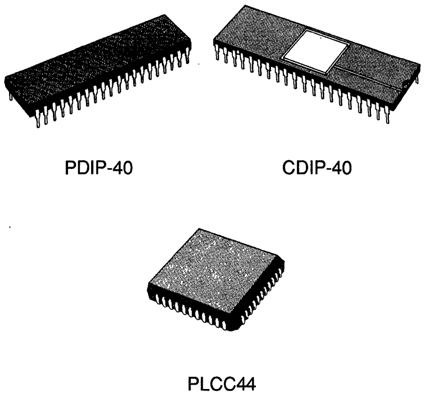
Most users read only chapters that are of interest to them. If you are designing the microcomputer hardware structure using the SCC as a peripheral, you will want to read the Initialisation Worksheet and Interrupt Routine Sections.

If you are programming a system using the SCC, you will be more interested, on the Initialization Worksheet Section.

Points To Watch Out For :

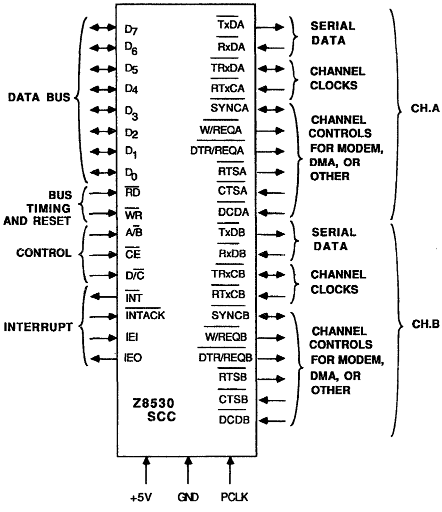
1. Follow the worksheet for initialization (page). Unexplainable operations may occur if this procedure is not followed.
2. Watch out for Write Recovery time violation (Interfacing Section). Both the CPU clock rate and the SCC clock rate will affect the Write Recovery time.
3. Ensure Mode bits are not changed when writing Commands. (Register Overview page 75). Each Mode bit affects only one function and a Command bit entry requires a rewrite of the entire register ; therefore, care must be taken to insure the integrity of the Mode bits whenever a new command is issued.
4. Data must be valid prior to falling edge of $\overline{WR}$ or $\overline{DS}$.
5. If not used, $\overline{INTACK}$ should be tied high.

Temperature Range



(Ordering Information at the end of the datasheet)

Figure 1 : Logic Functions.



CAPABILITIES

- Two independent full-duplex channels.
- Synchronous/Isosynchronous data rates :
  - Up to 1/4 of the PCLK (i.e., 1 Mbit/sec. maximum data rate with 4 MHz PCLK. Using external phase-lock loop.
  - Up to 375 Kbit/sec. with a 6 MHz clock rate. Up to 250 Kbit/sec. with a 4 MHz clock rate (FM encoding using digital phase-locked loop).
  - Up to 187.5 Kbit/sec. with a 6 MHz clock rate Up to 125 Kbit/sec. with a 4 MHz clock rate (NRZI encoding using digital phase-locked loop).
- Asynchronous capabilities :
  - 5, 6, 7, or 8 bits per character
  - 1, 1-1/2, or 2 stop bits
  - Odd or even parity
  - Times 1, 16, 32, or 64 clock modes
  - Break generation and detection
  - Parity, overrun and framing error detection.
- Byte-oriented synchronous capabilities :
  - Internal or external character synchronization
  - 1 or 2 sync characters (6 or 8 bits/character) in separate registers
  - Automatic Cyclic redundancy check (CRC) generation/detection.
- SDLC/HDLC capabilities :
  - Abort sequence generation and checking
  - Automatic zero insertion and deletion
  - Automatic flag insertion between messages
  - Address field recognition
  - I-field residue handling
  - CRC generation/detection
  - SDLC loop mode with EOP recognition/loop entry and exit.
- Receiver data registers quadruply buffered. Transmitter data registered double buffered.
- NRZ, NRZI, or FM encoding/decoding.
- Baud-rate generator in each channel.
- Digital phase-locked loop for clock recovery.
- Crystal oscillator.

GENERAL DESCRIPTION

The SCC Serial Communications Controller is a dual-channel, multiprotocol data communications peripheral designed for use with 8-bit and 16-bit microprocessors. The SCC functions as a serial-to-parallel, parallel-to-serial converter/controller. The SCC can be software-configured to satisfy a wide variety of serial communications applications. The device contains a variety of new, sophisticated internal functions including on-chip baud rate generators, digital phase-lock loops, and crystal oscillators, which dramatically reduce the need for external logic.

The SCC handles asynchronous formats, Synchronous byte-oriented protocols such as IBM Bisync, and Synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (telecommunications, cassette, diskette, tape drivers, etc.).

The device can generate and check CRC codes in any Synchronous mode and can be programmed to check data integrity in various modes. The SCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

With access to 14 Write registers and 7 Read registers per channel, the user can configure the SCC so that it can handle all asynchronous formats regardless of data size, number of stop bits, or parity requirements. The SCC accommodates all synchronous formats including character, byte, and bit-oriented protocols.

Within each operating mode, the SCC also allows for protocol variations by checking odd or even parity bits, character insertion or deletion, CRC generation and checking/break and abort generation and detection, and many other protocol-dependent features.

The SCC Z8530 is designed for non-multiplexed buses and is easily interfaced to CPUs such as the 8080, Z80, 6800, 68000 and *Multibus.

GENERAL DESCRIPTION (cont'd)

Figure 2 and Figure 5 show block diagrams of the SCC. Received data enters the receive data pins and follows one of several data paths, depending on the state of the control logic. The contents of the registers and the state of the external control pins establish the internal control logic. Transmitted data follows a similar pattern of control, register, and external pin definition.

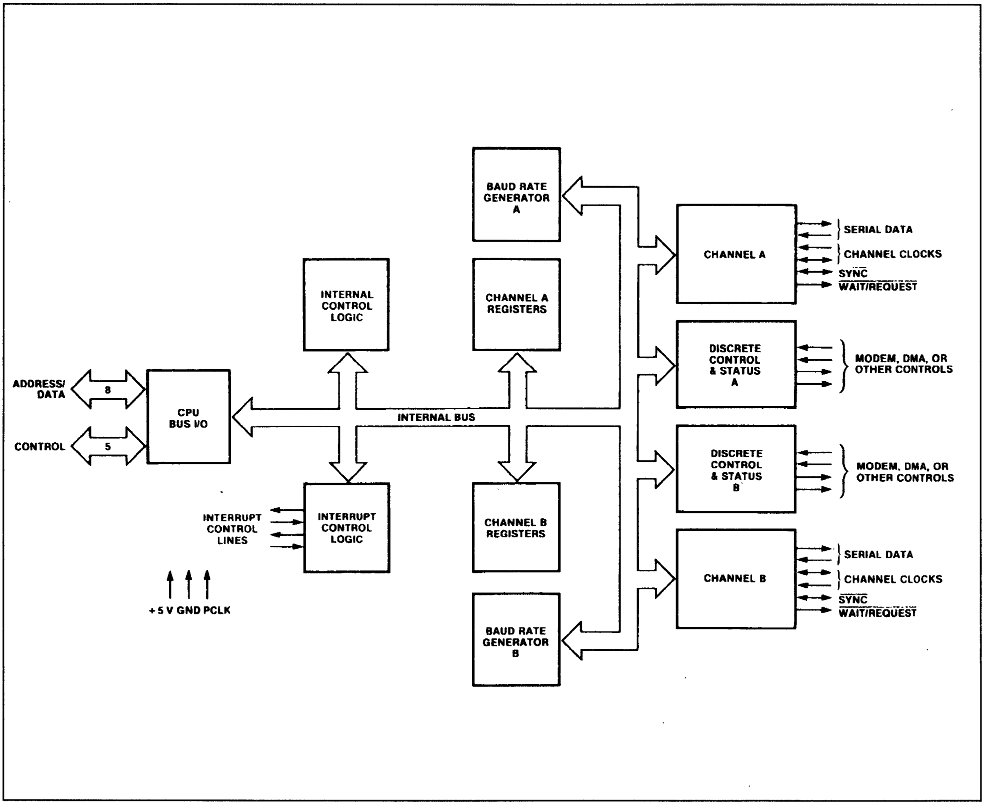
PIN DESCRIPTIONS

The SCC pins are divided into seven functional groups : Address/Data, Bus Timing and Reset, Device Control, Interrupt, Serial Data (both channels), Peripheral Control (both channels), and Clocks (both channels). Figures 3 and 4 show the Pin Configuration in both the proposed packages, Dual in Line and Chip Carrier.

The Address/Data group consists of the bidirectional lines used to transfer data between the CPU and the SCC. The direction of these lines depends on whether the SCC is selected and whether the operation is a Read or a Write.

The Timing and Control groups designate the type of transaction to occur and when this transaction will occur. The Interrupt group provides inputs and outputs to conform to the bus specifications for handling and prioritizing interrupts. The remaining groups are divided into Channel A and Channel B groups for serial data (transmit or receive), periphe-

Figure 2 : SCC Block Diagram.



PIN DESCRIPTION (cont'd)

ral control (such as DMA or modem), and the input and output lines for the receive and transmit clocks.

Here below are described the pin functions of the Z8530 Serial Communications Controller.

A/$\overline{\text{B}}$. Channel A/Channel B Select (input, Channel A active HIGH). This signal selects the channel in which the Read or Write operation occurs.

$\overline{\text{CE}}$. Chip Enable (input, active LOW). This signal selects the SCC for operation. It must remain active throughout the bus transaction.

D0-D7. Data Lines (bidirectional, 3-state). These I/O lines carry data or control information to and from the SCC.

D/$\overline{\text{C}}$. Data/Control (input, Data active HIGH). This signal defines the type of information transfer performed by the SCC : data or control.

$\overline{\text{RD}}$. Read (input, Active LOW). This signal indicates a Read operation and when the SCC is selected, enables the SCC bus drivers. During the interrupt acknowledge cycle, this signal gates the interrupt vector onto the bus if the SCC is the highest priority device requesting an interrupt.

$\overline{\text{WR}}$. Write (input, active LOW). When the SCC is selected, this signal indicates a Write operation. The coincidence of $\overline{\text{RD}}$ and $\overline{\text{WR}}$ is interpreted as a Reset.

$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$. Clear to Send (inputs, active LOW). If these pins are programmed as auto enables, a LOW on these inputs enables the respective transmitters. If not programmed as auto enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC detects transitions on these inputs and can interrupt the CPU on either logic level transitions.

$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$. Data Carrier Detect (inputs, active LOW). These pins function as receiver enables if they are programmed as auto enable bits ; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accomodate slow rise-time signals. The SCC detects transitions on these pins and can interrupt the CPU on either logic level transitions.

$\overline{\text{DTR}}$/ $\overline{\text{REQA}}$, $\overline{\text{DTR}}$/ $\overline{\text{REQB}}$. Data Carrier Detect (inputs, active LOW). These pins function as receiver enables if they are programmed into the DTR bit. They can also be used as general-purpose outputs (transmit) or as request lines for the DMA controller. The SCC allows full duplex DMA transfers.

IEI. Interrupt Enable In (input, active HIGH). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt-driven device. A HIGH on IEI indicates that no other higher priority

Figure 3 : DIP Pin Connections.

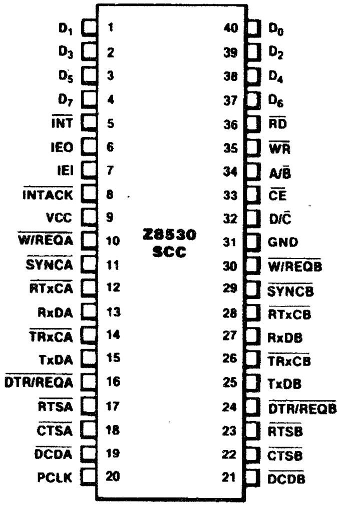
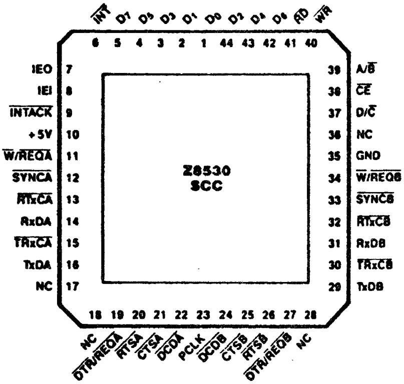


Figure 4 : Chip Carrier Pin Connection.



PIN DESCRIPTION (cont'd)

device has an Interrupt Under Service (IUS) or is requesting an interrupt.

IEO. Interrupt Enable Out (output, active HIGH). IEO is HIGH only if IEI is HIGH and the CPU is not servicing an SCC or SCC interrupt, or the controller is not requesting an interrupt (interrupt acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

INTACK. Interrupt Acknowledge (input, active LOW). This signal indicates an active interrupt acknowledge cycle. During this cycle, the interrupt daisy chain settles. When RD or DS becomes active, the SCC places an interrupt vector on the data bus (if IEI is HIGH). INTACK is latched by the rising edge of AS or PCLK.

INT. Interrupt Request (output, open-drain, active LOW). This signal is activated when the SCC is requesting an interrupt.

PCLK. Clock (input). This is the master clock used to synchronize internal signals. PCLK is not required to have any phase relationship with the master system clock. PCLK is a TTL level signal.

RTSA, RTSB. Request to Send (outputs, active LOW). When the Request to Send (RTS) bit in Write Register 5 (Figure 48) is set, the RTS signal goes LOW. When the RTS bit is reset in the Asynchronous mode and auto enables is on, the signal goes HIGH after the transmitter is empty. In Synchronous mode or in Asynchronous mode with auto enables off, the RTS pins strictly follow the state of the RTS bit. Both pins can be used as general-purpose outputs.

RTxCA, RTxCB. Receive/ Transmit Clocks (inputs, active LOW). The functions of these pins are under program control. In each channel, RTxC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the digital phase-locked loop (refer to Section 4 for bit configurations). This pins can also be programmed for use the respective SYNC pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

RxDA, RxDB. Receive Data (inputs, active HIGH). These input signals receive serial data at standard TTL levels.

SYNCA, SYNCB. Synchronization (inputs/outputs, active LOW). These pins can act as either inputs, outputs, or as part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to CTS and DCD. In this mode, transitions on these lines affect the state of the Sync/ Hunt status bits in Read Register 0 (Figure 59), but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, SYNC must be driven LOW two receive clock cycles after the last bit in the sync character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of SYNC.

In the Internal Synchronization mode, (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which sync characters are recognized. The sync condition is not latched, so these outputs are active each time a sync character is recognized (regardless of character boundaries). In SDLC mode, these pins act as outputs and are valid on receipt of a flag.

TRxCA, TRxCB. Transmit/ Receive Clocks (inputs or outputs, active LOW). The functions of these pins are under program control. TRxC may supply the receive clock or the transmit clock in the Input mode or supply the output of the digital phase-locked loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode. (Refer to Section 4 for bit configuration).

TxDA, TxDB. Transmit Data (outputs, active HIGH). This output signal transmits serial data at standard TTL levels.

W/ REQA, W/ REQB. Wait/ Request (outputs, open drain when programmed for Wait function, driven HIGH or LOW when programmed for a Request function). These dual-purpose outputs can be programmed as Request (receive) lines for a DMA controller or as Wait lines to synchronize the CPU to the SCC data rate. The reset state is Wait. The SCC allows full duplex DMA transfer.

OVERVIEW

The SCC internal structure provides all the interrupt and control logic necessary to interface with non-multiplexed bus. Interface logic is also provided to monitor modem or peripheral control inputs and outputs. All of the control signals are general purpose and can be applied to various peripheral devices as well as used for modem control.

The center for data activity revolves around the internal read and write registers. The programming of these registers provides the SCC with functional "personality" ; i.e., register values can be assigned before or during program sequencing to determine how the SCC will establish a given communication protocol.

Register Functions

All modes of communication are established by the bit values of the write registers. As data is received or transmitted, read register values may change. These changed values can promote software action or internal hardware action for further register changes.

The register set for each channel includes 14 write registers and seven read registers. Ten write registers are used for control, two for sync character generation, and two for baud rate generation. In addition there are two write registers which are shared by both channels ; one is the interrupt vector register and one is the master interrupt control and reset register. Four read registers indicate status information, two are for baud rate generation, and one for the receive buffer. In addition there are two read registers which are shared by both channels ; one for the interrupt pending bits and one for interrupt vector.

Table 1 lists the assigned functions for each read and write register. The SCC contains only one WR2 (interrupt vector) and one WR9 (master interrupt control). Both registers are accessed and shared by either channel. Chapter 7 provides a detailed bit legend and description of each register.

Table 1 : Register Set.

	Read Register Functions
RR0	Transmitt/ Receive buffer status, and External status
RR1	Special Receive Condition status, residue codes, error conditions
RR2	Modified (Channel B only) interrupt vector and Unmodified interrupt vector (Channel A only)
RR3	Interrupt Pending bits (Channel A only)
RR8	Receive buffer
RR10	Miscellaneous XMTR, RCVR status parameters
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External / Status interrupt control information
	Write Register Functions
WR0	Command Register, CRC initialization resets for various modes
WR1	Interrupt conditions, Wait / DMA request control
WR2	Interrupt vector (access through either channel)
WR3	Receive / Control parameters, number of bits per character, RxCRC enable
WR4	Transmit / Receive miscellaneous parameters and codes, clock rate, number of sync characters, stop bits, parity
WR5	Transmit parameters and control, number of Tx bits per character, TxCRC enable
WR6	Sync character (1 st byte) or SDLC flag
WR8	Transmitt buffer
WR9	Master interrupt control and reset (accessed through either channel), reset bits, control interrupt daisy chain
WR10	Miscellaneous transmitter/receiver control bits, NRZI, NRZ, FM encoding, CRC reset
WR11	Clock mode control, source of Rx and Tx clocks
WR12	Lower byte of baud rate generator time constant
WR13	Upper byte of baud rate generator time constant
WR14	Miscellaneous control bits : baud rate generator, Phase-Locked Loop control, auto echo, local loopback
WR14	External/ Status interrupt control information-control external conditions causing interrupts

OVERVIEW (cont'd)

Data Paths

Figure 6 illustrates the data paths involved in the six major areas of the SCC :

- Transmitter
- Receiver
- Baud rate generator
- DPLL
- Clocking options
- Data encoding

All communication modes are established by programming the write registers. As data is received or transmitted, read register values may change, altering the direction of the data path. These changed values can promote software action or internal hardware action for further register changes.

Transmitter. The transmitter has an 8-bit Transmit Data register (WR8) loaded from the internal data bus and a Transmit Shift register loaded from either WR6, WR7, or the Transmit Data register. In byte-oriented modes, WR6 and WR7 can be programmed with sync characters. In Monosync mode, an 8-bit or 6-bit sync character is used (WR6), whereas a 16-bit sync character is used (WR6 and WR7) in Bisync mode. In bit-oriented synchronous modes, the flag contained in WR7 is loaded into the Transmit Shift register at the beginning and end of a message.

If asynchronous data is processed, WR6 and WR7 are not used and the Transmit Shift register is formatted with start and stop bits shifted out to the transmit multiplexer at the selected clock rate. Synchronous data (except SDLC/HDLC) is shifted to the CRC generator as well as to the transmit multiplexer.

SDLC/HDLC data is shifted to the CRC Generator and out through the zero insertion logic (which is disabled while the flags are being sent). A "0" is inserted in all address, control, information, and frame check fields following five contiguous "1s" in the data stream. The result of the CRC generator for SDLC data is also routed through the zero insertion logic and then to the transmit multiplexer.

Receiver. The receiver has a three deep 8-bit Data FIFO (paired with an 8-bit Error FIFO), and an 8-bit shift register. This arrangement creates a 3-byte delay time, which allows the CPU time to service an interrupt at the beginning of a block of high-speed data. With each Receive Data FIFO, the error FIFO stores parity and framing errors and other types of status information. The error FIFO is readable in Read Register 1.

Incoming data is routed through one of several paths depending on the mode and character length. In Asynchronous mode, serial data enters the 3-bit delay (Figure 5) if the character length of seven or eight bits is selected. If a character length of five or six bits is selected, data enters the receive shift register directly.

In synchronous modes, the data path is determined by the phase of the receive process currently in operation. A synchronous receive operation begins with a hunt phase in which a bit pattern that matches the programmed sync characters (6-bit, 8-bit, or 16-bit is searched).

The incoming data then passes through the Sync register and is compared to a sync character stored in WR6 or WR7 (depending on which mode it is in). The monosync mode matches the sync character programmed in WR7 and the character assembled in the Receive Sync register to establish synchronization.

Synchronization is achieved differently in the Bisync mode. Incoming data is shifted to the Receive Shift register while the next eight bits of the message are assembled in the Receive Sync register. If these two characters match the programmed characters in WR6 and WR7, synchronization is established. Incoming data can then bypass the Receive Sync register and enter the 3-bit delay directly.

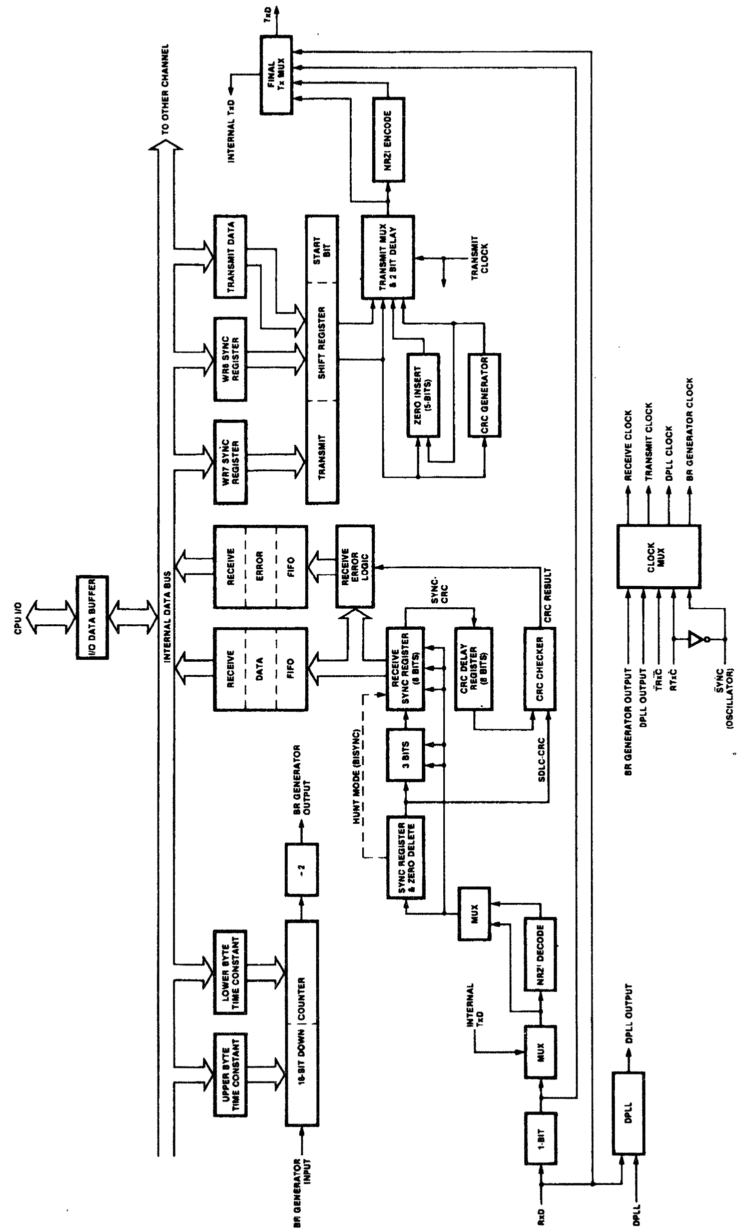
The SDLC mode of operation uses the Receive Sync register to monitor the receive data stream and to perform zero deletion when necessary ; i.e., when five continuous "1s" are received, the sixth bit is inspected and deleted from the data stream if it is "0". The seventh bit is inspected only if the sixth bit equals one. If the seventh bit is "0", a flag sequence has been received and the receiver is synchronized to that flag. If the seventh bit is a "1", an abort or an EOP (End Off Poll) is recognized, depending on the selection of either the normal SDLC mode or SDLC Loop mode.

The same path is taken by incoming data for both SDLC modes. The reformatted data enters the 3-bit delay and is transferred to the Receive Shift register. The SDLC receive operation begins in the hunt phase by attempting to match the assembled character in the Receive Shift Register with the flag pattern in WR7. Then the flag character is recognized, subsequent data is routed through the same path, regardless of character length.

Either the CRC - 16 or CRC - SDLC cyclic redundancy check (CRC) polynomial can be used for both Monosync and Bisync modes, but only the CRC - SDLC polynomial is used for SDLC operation. The data path taken for each mode is also different.

OVERVIEW (cont'd)

Figure 5 : Data Paths.



SGS-THOMSON MICROELECTRONICS

OVERVIEW (cont'd)

Bisync protocol is a byte-oriented operation that requires the CPU to decide whether or not a data character is to be included in CRC calculation. An 8-bit delay in all synchronous modes except SDLC is allowed for this process. In SDLC mode, all bytes are included in the CRC calculation.

Baud Rate Generator. Each channel in the SCC contains a programmable baud rate gene-rator. Each generator consists of two 8-bit, time-constant registers forming a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output that makes the output a square wave. On start-up, the flip-flop on the output is set High so that it starts in a known state, the value in the time-constant register is again loaded into the counter, and the counter begins counting down. When a count of zero is reached, the output of the baud rate generator toggles, the value in the time-constant register is loaded into the counter, and the process starts over. The time constant can be changed at any time, but the new value does not take effect until the next load of the counter.

No attempt is made to synchronize the loading of a new time constant with the clock used to drive the generator. When the time constant is to be changed, the generator should be stopped by writing to an enable bit in WR14. This ensures the loading of the correct time constant.

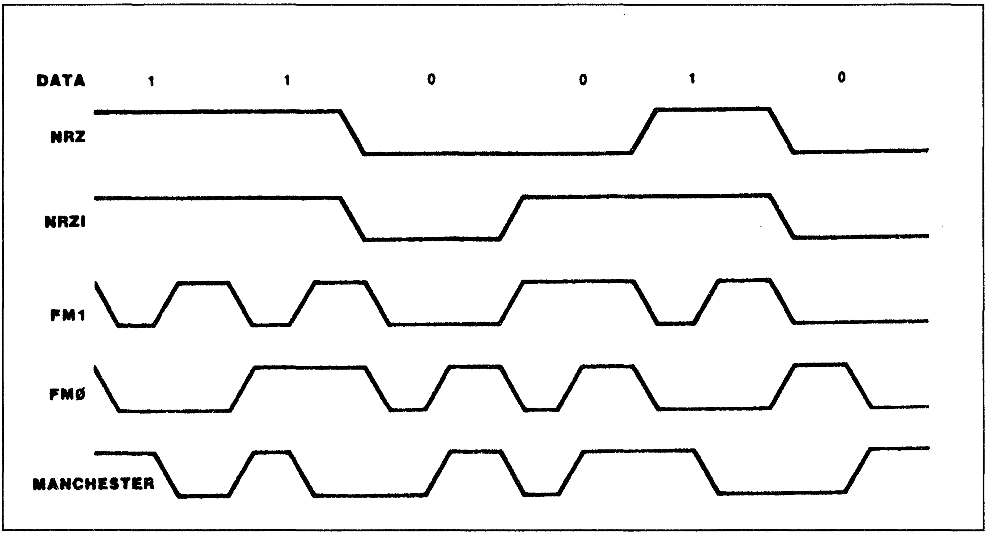
If neither the transmit clock nor the receive clock are programmed to come from the $\overline{\text{TRXC}}$ pin, the output of the baud rate generator may be made available for external use on the $\overline{\text{TRXC}}$ pin.

Digital Phase-locked Loop (DPLL). The SCC contains a digital phase-locked loop that can be used to recover clock information from a data stream with NRZI or FM coding. The DPLL is driven by a clock nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a receive clock for the data. This clock can then be used as the SCC receive clock, the transmit clock, or both.

Clocking Options. The SCC can select several clock sources for internal and external use. Write Register 11 is the Clock Mode Control register for both the receive and transmit clocks. It determines the type of signal on the $\overline{\text{SYNC}}$ and $\overline{\text{RTxC}}$ pins and the direction of the $\overline{\text{TRxC}}$ pin.

Write Register 11 also controls the output of the baud rate generator, the DPLL output, and the selection of either a TT1 or XTAL output for the $\overline{\text{RTxC}}$ pin.

Figure 6 : Data Encoding Methods.



OVERVIEW (cont'd)

Data Encoding. Figure 6 illustrates the four encoding methods used by the SCC. In NRZ encoding, a "1" is represented by a High level and a "0" is represented by a Low level. In NRZI encoding, a "1" is represented by no change in level and a "0" is represented by a change in level. In FM1 (more properly, biphase mark), a transition occurs at the beginning of every bit cell. A "1" is represented by an additional transition at the center of the bit cell and a "0" is represented by the absence of a tran-sition at the center of the bit cell. In FM0 (more properly, biphase space), a transition occurs at the beginning of every bit cell. A "0" is represented by an additional transition at the center of the bit cell and a "1" is represented by the absence of a transition at the center of the bit cell.

In addition to these four methods, the SCC can be used to decode Manchester (biphase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. Manchester encoding always produces a transition at the center of the bit cell. If the transition is Low to High, the bit is "0". If the transition is High to Low, the bit is "1".

Data Communications Capabilities

SCC logic handles all asynchronous, byte-oriented synchronous, and bit-oriented synchronous modes of operation. The following section briefly describes asynchronous, synchronous, and SDLC modes of communication.

Asynchronous. Figure 7 represents a typical asynchronous message format using one start bit, seven data bits, one parity bit, and one stop bit. A start bit is a High-to-Low transition detected by an asynchronous receiver and is actually an information bit notifying the receiver of an incoming message.

The start bit also initiates a clock circuit to provide latching pulses during expected data bit intervals. The parity bit is provided for error checking. The parity bit is calculated in both the receiver and the transmitter ; the two results are compared to ensure that the expected and the actual bit values match. The stop bit returns the message unit to the quiescent marking state ; i.e., a constant high state condition lasts until the next High-to-Low start bit indicates an incoming data byte. During reception, the start and stop bits are stripped away and checked for errors, leaving only the working data for CPU interaction. The number of selected bits for each asynchronous function may differ between the transmitter and the receiver.

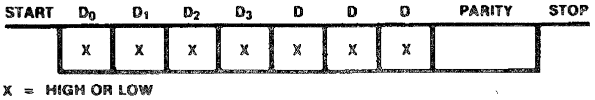
Monosync Mode. Monosync and Bisync modes require clocking information to be transmitted along with the data either by a method of encoding data that contains clocking information, or by a modem that encodes or decodes clock information in the modulation process.

Start and stop bits are not required in synchronous modes. All bits are used to transmit data. This eliminates the "waste" characteristic of asynchronous communication.

Figure 8 shows the character format for synchronous transmission. For example, bits 1-8 might be one character and bits 9-13 part of another character ; or bit 1 might be part of one character, bits 2-9 part of a second character, and bits 10-13 part of a third character. The framing (where each character begins) of each character is accomplished by defining a synchronization character, commonly called a "sync character".

The CPU places the receiver in Hunt mode whenever transmission begins (or whenever a data dropout has occurred and the hardware determines that resynchronization is necessary). In Hunt mode, the receiver shifts a bit into the Receive Shift register and compares the contents of the Receive Shift register and with the sync character (stored in another register), repeating the process until a match occurs. When a match occurs, the receiver begins transferring bytes to the receive FIFO.

Figure 7 : Asynchronous Message Format.



OVERVIEW (cont'd)

Bisynchronous Mode. The Bisync mode of operation (Figure 9) is similar to the Monosync mode, except that two sync characters are provided instead of one. Bisync attemps a more structured approach to synchronization through the use of special characters as message "headers" or "trailers". A detailed description of IBM's Bisync can be found in McNamara's Book (See Preface).

External Sync Mode. External Sync mode (Figure 10) eliminates the use of sync characters in the serial data stream by providing an external sync signal to mark the beginning of a data field ; i.e., an external input pin (Sync) waits for an active state change to indicate the beginning of an information field.

SDLC Mode. Synchronous Data Link Control mode (SDLC) uses synchronization characters similar to Bisync and Monosync modes (such as flags and pad characters), but it is a bit-oriented protocol instead of byte-oriented protocol.

Any data communication link involves at least two stations. The station that is responsible for the data link and issues the commands to control the link is called the "primary station". The other station is a "secondary station". Not all information transfers need to be initiated by a primary station. In SDLC mode, a secondary station can be the initiator.

The basic format for SDLC is a "frame" (Figure 11).The information field is not restricted in format or content and can be of any reasonable length (including zero). Its maximum length is that which can be expected to arrive at the receiver error-free most of time. Hence, the determination of maximum length is a function of communication channel error rate.

The two flags that delineate the SDLC frame serve as reference points when positioning the address and control fields, and they initiate the transmission error check. The ending flag indicates to the receiving station that the 16 bits just received constitute the frame check. The ending flag could be followed by another frame, another flag, or an idle. This means that when two frames fallow one another, the intervening flag may simultaneously be the ending flag of the first frame and the begin-ning flag of the next frame. Since the SDLC mode does not use characters of defined length, but rather works on a bit-by-bit basis, the 01111110 (7EH) flag can be recognized at any time.

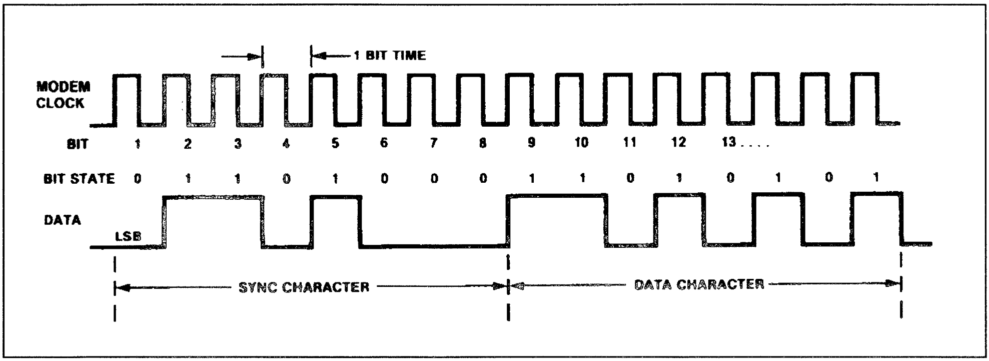
To ensure that the flag is not sent accidentally, SDLC procedures require a binary "0" to be inserted by the transmitter after the transmission of any five contiguous "1s". The receiver then removes the "0" following a received succession of five "1s". Inserted and removed "0s" are not included in the CRC calculation.

The address field is 8 bits long and designates the number of secondary station to which the commands or data from the primary station are sent.The control field is eight bits long and is used to initiate all SDLC activities.

The SCC can also serve the High-level synchronous Data Link Communication (HDLC) protocol, which is identical to SDLC except for differences in framing.

SDLC Loop Mode. The SCC supports SDLC Loop mode in addition to normal SDLC. SDLC Loop mode is very similar to normal SDLC but is usually used in application where a point-to-point network is not ap-

Figure 8 : Monosync Data Character Format.



OVERVIEW (cont'd)

propriate (for example, Point-Of-Sale terminals). In an SDLC Loop there is a primary station, called the controller, that manages the message traffic flow on the loop, and there are any number of secondary stations.

A secondary station in an SDLC loop is always listening to the messages being sent around the loop, and must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay. The secondary station can only place its own message on the loop at specific times. The controller signals that secondary stations may transmit messages by sending a special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern 11111110. Because of zero insertion during messages this bit pattern is unique and thus is easily recognized.

When a secondary station has a message to transmit and it recognizes an EOP on the line , the first thing that it does is to change the last 1 or the EOP to a "0" before transmitting it. This turns the EOP into a Flag sequence. The secondary station now places its message on the loop and terminates its message with an EOP. Any secondary stations further down the loop with messages to transmit can then append its message to the message of the first secondary station by the same process. All secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop, except upon recognizing an EOP.

There are also restrictions as to when and how a secondary station physically becomes part of the loop. A secondary station that has just powered up must monitor the loop, without the one-bit-time delay, until it recognizes an EOP. When an EOP is recognized the one-bit-time delay is switched on. This does not disturb the loop because the line is marking idle between the time that the controller sends the EOP and the time that it receives the EOP back. The secondary station that has gone on-loop cannot place a message on the loop until the next time that an EOP is issued by the controller. A secondary station goes off-loop in a similar manner. When given a command to go off-loop, the secondary station waits until the next EOP to remove the one-bit-time delay.

To operate the SCC in SDLC Loop mode, the SCC must first be programmed just as if normal SDLC were to be used. Loop mode is then selected by writing the appropriate control word in WR10. The SCC

Figure 9 : Bisynchronous Message Format.

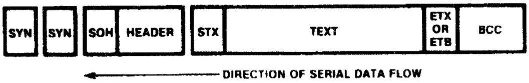


Figure 10 : External Sync Format.

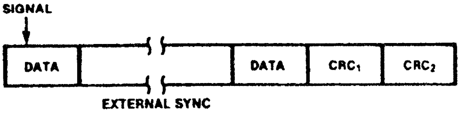
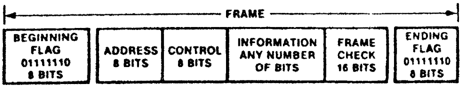


Figure 11 : SDLC Message Format.


FRAME
BEGINNING FLAG 01111110 8 BITS
ADDRESS 8 BITS
CONTROL 8 BITS
INFORMATION ANY NUMBER OF BITS
FRAME CHECK 16 BITS
ENDING FLAG 01111110 8 BITS

OVERVIEW (cont'd)

is now waiting for the EOP so that it can go on loop. While waiting for the EOP, the SCC ties TxD to RxD with only the internal gate delays in the signal path. When the first EOP is recognized by the SCC, the Break/Abort/EOP bit is set in RR0, generating an External/Status interrupt (if so enabled). At the same time, the On-Loop bit in RR10 is set to indicate that the SCC is indeed on-loop, and a one-bit time delay is inserted in the TxD to the RxD patch.

The SCC is now on-loop but cannot transmit a message until a flag and the next EOP are received. The requirement that a flag be received ensures that the SCC cannot erroneously send messages until the controller ends the current polling sequence and starts another one.

A secondary station on the loop is prohibited from transmitting a message during a polling sequence unless it captures the line at the moment the EOP passes by. The SCC does this automatically. If the CPU in the secondary station with SCC needs to transmit a message, the Go-Active-On-Poll bit in WR10 must be set. If this bit is set when the EOP is detected, the SCC changes the EOP to a flag and starts sending another flag. The EOP is reported in the Break/Abort/EOP bit in RR0 and the CPU should write its data bytes to the SCC, just as in normal SDLC frame transmission. When the frame is complete and CRC has been sent, the SCC closes with a flag and reverts to One-Bit-Delay mode. The last zero of the flag, along with the marking line echoed from the RxD pin, form an EOP for secondary stations further down the loop. If the Go-Active-On-Poll bit is not set at the time the EOP passes by, the SCC cannot send a message until a flag (terminating the current polling sequence) and another EOP are received. While the SCC is actually transmitting a message, the loop-sending bit in R10 is set to indicate this.

If SDLC loop is de-selected, the SCC is designed to exit from the loop gracefully . When SDLC Loop mode is de-selected by writing to WR10, the SCC waits until the next polling cycle to remove the on-bit time delay. If a polling cycle is in progress at the time the command is written, the SCC finishes sending any message that it may be transmitting, ends with an EOP, and disconnects TxD from RxD. If no message was in progress, the SCC immediately disconnects TxD from RxD. To ensure proper loop operation after the SCC goes off the loop, and until the external relays take the SCC completely out of the loop, the SCC should be programmed for Mark idle instead of Flag idle. When the SCC goes off the loop, the On-Loop bit is reset.

The SCC allows the user the option of using NRZI in SDLC Loop mode by programming WR20 appropriately. With NRZI encoding, the outputs of secondary stations in the loop may be inverted from their inputs because of messages that they have transmitted. Removing the stations from the loop (removing the one-bit time delay) may cause problems further down the loop because of extraneous transitions on the line. The SCC avoids this problem by making transparent adjustements at the end of each frame it sends in response to an EOP. A response frame from the SCC is terminated by a flag and an EOP. Normally, the flag and the EOP share a zero, but if such sharing would cause the RxD and TxD pins to be of opposite polarity after the EOP, the SCC adds another zero between the flag and the EOP. This causes an extra line transition so that RxD and TxD are identical after the EOP is sent. This extra zero is completely transparent because it only means that the flag and the EOP no longer share a zero. All that a proper loop exit needs, therefore, is the removal of the one-bit time delay.

I/O Capabilities.

The SCC can work with three basic forms of I/O operations : polling, interrupts, and block transfer. All three I/O types involve register manipulation during initialization and data transfer.

Polling. During a polling sequence, the status of Read Register 0 is examined in each channel. This register indicates whether or not a receive or transmit data transfer is needed and whether or not any special conditions are present, e.g., errors.

This method of I/O transfer avoids interrupts. All interrupt functions must be disabled in order to operate the device in a polled environment. With no interrupts enabled, this mode of operation must initiate a read cycle of Read Register 0 to detect an incoming character before jumping to a data handler routine.

Interrupts. The SCC provides interrupt capability through the use of pins and a hardware scheme that enhances the maximum speed of serial data. Whenever the interrupt ($\overline{INT}$) pin is active, the SCC is ready to transfer data.
Read and write registers are programmed so that an interrupt vector points to an interrupt service routine. The interrupt vector can also be modified to reflect various status conditions. Therefore, as many as eight different interrupt routines can be referenced.

Transmit interrupts, receive interrupts, and external/status interrupts are the main sources of interrupts. Each interrupt source is enabled under pro-

OVERVIEW (cont'd)

gram control, with channel A having a higher priority than channel B and with receive, transmit, and external/status interrupts prioritized respectively within each channel.

Block Transfers. The SCC provides a Block Transfer mode to accommodate CPU block transfer functions and DMA controllers. The Block Transfer mode uses the $\overline{W/REQ}$ output in conjunction with the Wait/Request bits in Write Register 1.

The $\overline{W/REQ}$ output can be defined by software as a $\overline{WAIT}$ line in the CPU Block Transfer mode or as a $\overline{REQUEST}$ line in the DMA Block Transfer mode.

To a DMA controller, the SCC REQUEST output indicates that the SCC is ready to transfer data to or from memory. To the CPU, the $\overline{WAIT}$ output indicates that the SCC is not ready to transfer data, thereby requesting the CPU to extend the I/O cycle.

INTERFACING THE SCC

This section covers the details of interfacing the Z8530 to a system. The general timing requirements for the device are described in the hardware information section.

Interfacing the Z8530

Two control signals, $\overline{RD}$ and $\overline{WR}$, are used by the Z8530 to time bus transactions. In addition, four other control signals, $\overline{CE}$, D/ $\overline{C}$, A/ $\overline{B}$ and $\overline{INTACK}$, are used to control the type of bus transaction that will occur. A bus transaction start when the addresses on D/ $\overline{C}$ and A/ $\overline{B}$ are asserted before $\overline{RD}$ or $\overline{WR}$ fall. The coincidence of $\overline{CE}$ and $\overline{RD}$ or $\overline{CE}$ and $\overline{WR}$ latches the state of D/ $\overline{C}$ and A/ $\overline{B}$ and starts the internal operation. The $\overline{INTACK}$ signal must have been previously sampled High by a rising edge of PCLK for a read or write cycle to occur. In addition to sampling $\overline{INTACK}$, PCLK is used by the interrupt section to set the IP bits. The Z8530 generates internal control signals in response to a register access. Since $\overline{RD}$ and $\overline{WR}$ have no phase relationship with PCLK, the circuitry generating these internal control signals provides time for metastable conditions to disappear. This results in a recovery time related to PCLK. This recovery time applies only between transactions involving the Z8530, and any interventing transactions are ignored. This recovery time is four PCLK cycles, measured from the falling edge of $\overline{RD}$ or $\overline{WR}$ in the case of a read or write of any register.

Z8530 Read Cycle Timing. The Read cycle timing for the Z8530 is shown in Figure 12. The address on A / $\overline{B}$ and D/ $\overline{C}$ is latched by the coincidence of $\overline{RD}$ and $\overline{CE}$ active. $\overline{CE}$ must remain LOW and $\overline{INTACK}$ must remain HIGH throughout the cycle. The Z8530 bus drivers are enabled while $\overline{CE}$ and $\overline{RD}$ are both LOW. A read with D/ $\overline{C}$ HIGH does not disturb the state of the pointers and a read cycle with D/ $\overline{C}$ LOW resets the pointers to zero after the internal operation is complete.

Z8530 Write Cycle Timing. The Write cycle timing for the Z8530 is shown in Figure 13. The address on A / $\overline{B}$ and D/ $\overline{C}$, as well as the data on D0-D7, is latched by the coincidence of $\overline{WR}$ and $\overline{CE}$ active. $\overline{CE}$ must remain LOW and $\overline{INTACK}$ must remain HIGH throughout the cycle. A write cycle with D/ $\overline{C}$ HIGH does not disturb the state of pointers and a write cycle with D/ $\overline{C}$ LOW resets the pointers to zero after the internal operation is complete.

Z8530 Interrupt Acknowledge Cycle Timing. The interrupt Acknowledge cycle timing for the Z8530 is shown in Figure 14. The state of $\overline{INTACK}$ is latched by the rising edge of PCLK. While $\overline{INTACK}$ is LOW, the state of A/ $\overline{B}$, $\overline{CE}$, D/ $\overline{C}$, and $\overline{WR}$ are ignored. Between the time $\overline{INTACK}$ is first sampled LOW and the time $\overline{RD}$ falls, the internal and external IEI/ IEO daisy chains settle ; this is A.C. parameter #38 TdIAi(RD). If there is an interrupt pending in the Z8530, and IEI is HIGH when RD falls, the Interrupt Acknowledge cycle was intended for the Z8530. This being the case, the Z8530 sets the appropriate Interrupt Under Service latch, and places an interrupt vector on D0-D7. If the falling edge of $\overline{RD}$ sets an IUS bit in the Z8530, the $\overline{INT}$ pin goes active in response to the falling edge. Note that there should be only one RD per Acknowledge cycle. Another important fact is that the IP bits in the Z8530 are updated by PCLK divided by two, and this clock to update IPs is stopped while the pointers point to RR2 and RR3. This prevents data changing during a read, but will delay interrupt requests if the pointers are left pointing at these registers.

Z8530 Register Access. The registers in the Z8530 are accessed in a two-step process, using a Register Pointer to perform the addressing. To access a particular register, the pointer bits must be set by writing to WR0 the pointer bits may be written in either channel because only one set exists in the Z8530. After the pointer bits are set, the next read

INTERFACING THE SCC (cont'd)

or write cycle of the Z8530 having D/$\overline{C}$ LOW will access the desired register. At the conclusion of this read or write cycle the pointer bits are reset to "0s", so that the next control write will be to the pointers in WR0. A read or RR8 (the receive data buffer) or a write to WR8 (the transmit data buffer) may either be done in this fashion or by accessing the Z8530 having D/$\overline{C}$ pin HIGH. A read or write with D/$\overline{C}$ HIGH accesses the data registers directly, and independently, of the state of the pointer bits. This allows single-cycle access tò the data registers and does not disturb the pointer bits. The fact that the pointer bits are reset to "0", unless explicitly set otherwise, means that WR0 and RR0 may also be accessed in a single cycle. That is, it is not necessary to write the pointer bits with "0" before accessing WR0 or RR0. There are three pointer bits in WR0, and these allow access to the registers with addresses 0 through 7. Note that a command may be written to WR0 at the same time that the pointer

Figure 12 : Z8530 Read Cycle Timing.

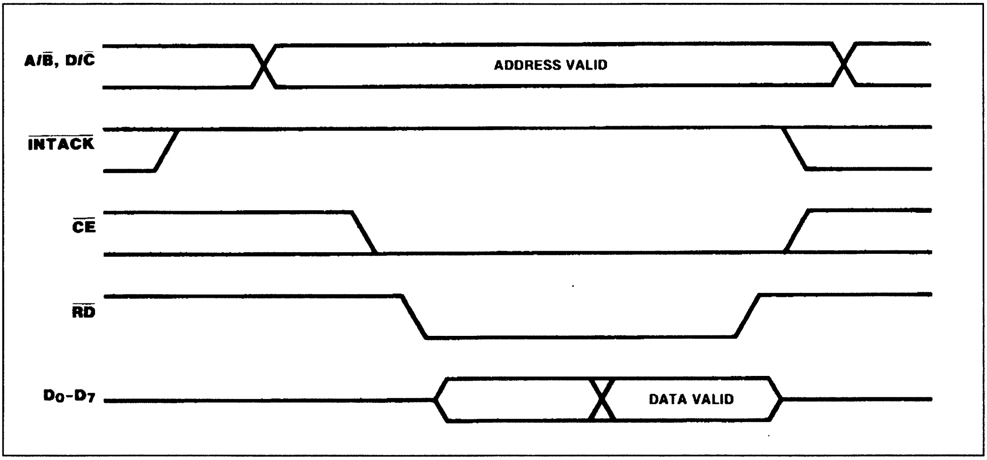
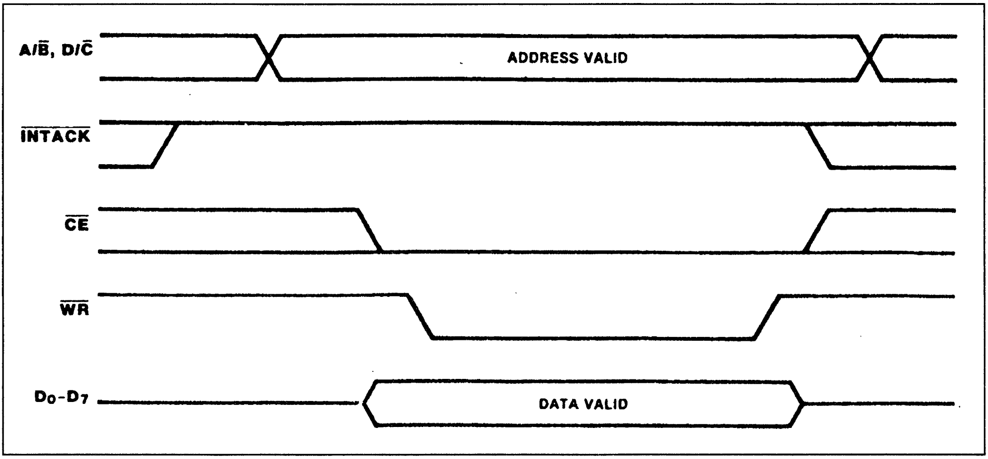
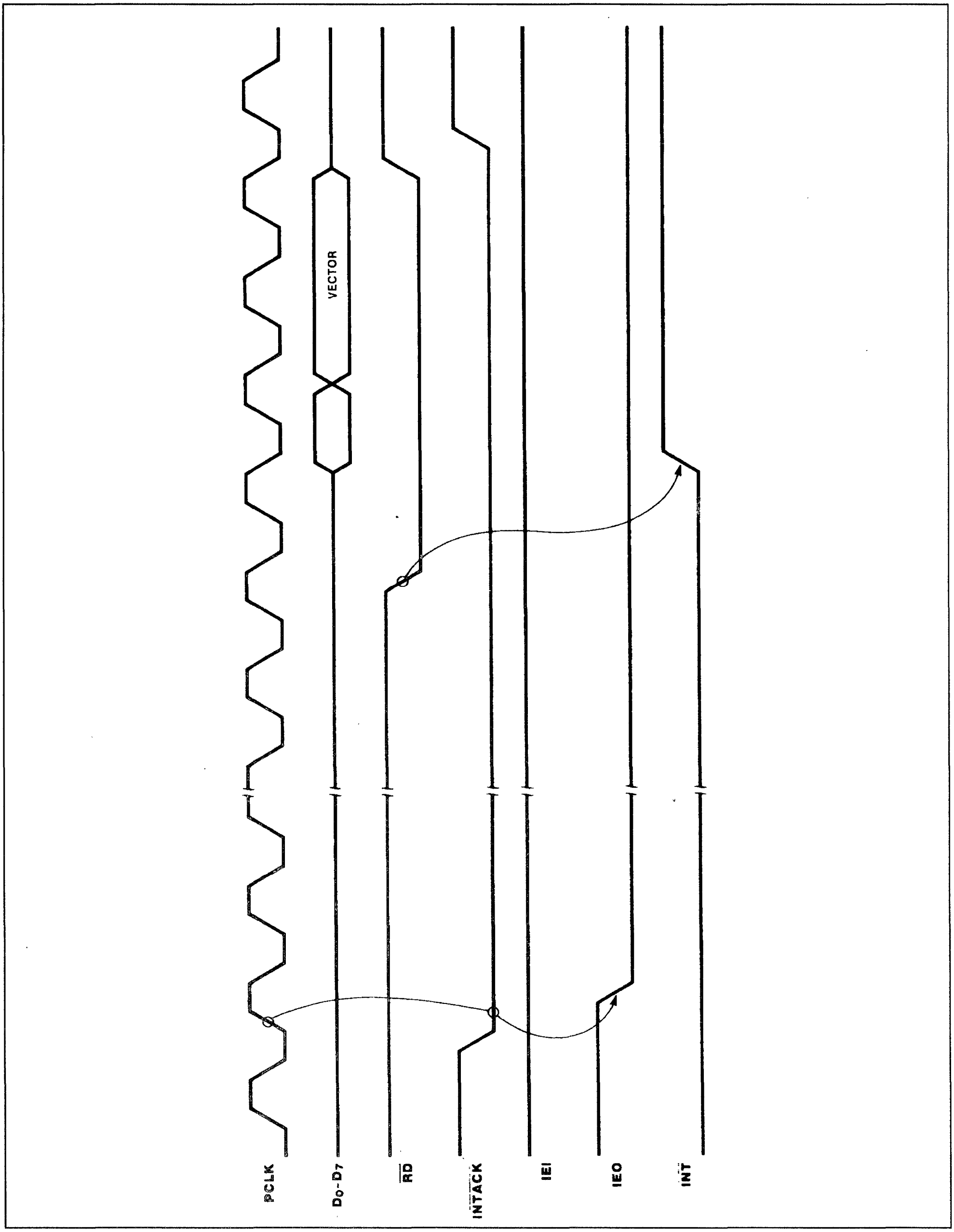


Figure 13 : Z8530 Write Cycle Timing.



INTERFACING THE SCC (cont'd)

Figure 14 : Z8530 Interrupt Acknowledge Details.



INTERFACING THE SCC (cont'd)

bits are written. To access the registers with addresses 8 through 15, a special command must accompany the pointer bits. This precludes concurrently issuing a command when pointing to these registers. The register map for the AmZ8530 is shown in Table 2. If for some reason, the state of the pointer bits is unknown they may be reset to "0" by performing a read cycle with the D/ $\overline{C}$ pin held LOW. Once the pointer bits have been set, the desired channel is selected by the state of the A/ $\overline{B}$ pin during the actual read or write of the desired register.

Z8530 Reset. The Z8530 may be reset by either hardware or software. Hardware reset occurs when $\overline{RD}$ and $\overline{WR}$ are both LOW, simultaneously, which is normally an illegal condition. As long as both $\overline{RD}$ and $\overline{WR}$ are LOW, the Z8530 recognizes the reset condition. Once this condition is removed, however, the reset condition is asserted internally for an additional four to five PCLK cycles. During this time any attempt to access the Z8530 will be ignored. The Z8530 has three software resets, encoded into command bits in WR9. There are two channel resets, which affect only one channel in the device and some of the bits in the write registers. The third command forces the same result as does a hardware reset. As in the case of a hardware reset, the Z8530 stretches the reset signal an additional four to five PCLK cycles beyond the ordinary valid access recovery time. The bits in WR9 may be written at the same time as the reset command because these bits are affected only by a hardware reset.

The reset values of the various registers are shown in Figure 15.

Table 2 : Z8530 Register Map.

A/B	PNT$_2$	PNT$_1$	PNT$_0$	WRITE	READ
0	0	0	0	WR0B	RR0B
0	0	0	1	WR1B	RR1B
0	0	1	0	WR2	RR2B
0	0	1	1	WR3B	RR3B
0	1	0	0	WR4B	(RR0B)
0	1	0	1	WR5B	(RR1B)
0	1	1	0	WR6B	(RR2B)
0	1	1	1	WR7B	(RR3B)
1	0	0	0	WR0A	RR0A
1	0	0	1	WR1A	RR1A
1	0	1	0	WR2	RR2A
1	0	1	1	WR3A	RR3A
1	1	0	0	WR4A	(RR0A)
1	1	0	1	WR5A	(RR1A)
1	1	1	0	WR6A	(RR2A)
1	1	1	1	WR7A	(RR3A)
With the Point High Command					
0	0	0	0	WR8B	RR8B
0	0	0	1	WR9	RR13B
0	0	1	0	WR10B	RR10B
0	0	1	1	WR11B	(RR15B)
0	1	0	0	WR12B	RR12B
0	1	0	1	WR13B	RR13B
0	1	1	0	WR14B	(RR10B)
0	1	1	1	WR15B	RR15B
1	0	0	0	WR8A	RR8A
1	0	0	1	WR9	(RR13A)
1	0	1	0	WR10A	RR10A
1	0	1	1	WR11A	(RR15A)
1	1	0	0	WR12A	RR12A
1	1	0	1	WR13A	RR13A
1	1	1	0	WR14A	RR14A
1	1	1	1	WR15A	RR15A

INTERFACING THE SCC (cont'd)

Figure 15 : Z8530 Register Reset Values.

HARDWARE RESET								CHANNEL RESET								
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WR0
0	0	.	0	0	.	0	0	0	0	.	0	0	.	0	0	WR1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR2
.	.	.	.	.	.	.	0	.	.	.	.	.	.	.	0	WR3
.	.	.	.	.	1	.	.	.	.	.	.	.	1	.	.	WR4
0	.	.	0	0	0	0	.	0	.	.	0	0	0	0	.	WR5
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR6
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR7
1	1	0	0	0	0	.	.	.	.	0	.	.	.	.	.	WR9
0	0	0	0	0	0	0	0	0	.	.	0	0	0	0	0	WR10
0	0	0	0	1	0	0	0	.	.	.	.	.	.	.	.	WR11
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR12
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	WR13
.	.	1	0	0	0	0	0	.	.	1	0	0	0	.	.	WR14
1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	WR15
0	1	.	.	.	1	0	0	0	1	.	.	.	1	0	0	RR0
0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	RR1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RR3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RR10

I/O PROGRAMMING CAPABILITIES

Regardless of the version of the SCC, all communication modes can use a choice of polling, interrupt and block transfer. These modes must be selected by the user to select the proper hardware and software required to supply data at the rate required.

Polling

This is the simpliest mode to implement. The software must poll the SCC to determine when data is to be inputted or outputted from the SCC.In this mode, MIE (WR9 bit 3), and Wait/ DMA Request Enable (WR1 bit 7) are both reset to 0 to disable any requests. The software must then poll RR0 to determine the status of the receive buffer, transmit buffer and external status.

Interrupt Operations

The SCC, as a microprocessor peripheral, will request an interrupt only when it needs servicing. This allows the CPU to perform other operations while the SCC does not need service. The SCC has an internal priority resolution method to allow the highest priority interrupt to be serviced first.

The SCC is flexible with its interrupt method. The interrupt may be acknowledged with a vector transferred, acknowledged without a vector, or not acknowledged at all.

Interrupt Without Acknowledge. In this mode, the Interrupt Acknowledge signal does not have to be generated. This allows a simpler hardware design that does not have to meet the Interrupt acknowledge timing. Soon after the INT goes active, the interrupt controller will jump to the interrupt routine. In the interrupt routine, the code must read RR2 from Channel B to read the vector including status. When the vector is read from Channel B, it always includes the status regardless of the VIS bit (WR9 bit 0). The status given will decode the highest priority interrupt pending at the time it is read. The vector is not latched so that the next read could produce a different vector if another interrupt occurs. The register is disabled from change during the read operation to prevent an error if a higher interrupt occurs exactly during the read operation.

Once the status is read, the interrupt routine must decode the interrupt pending, and clear the condition. Removing the interrupt condition will clear the IP and bring INT inactive, as long as there are no other IP bits set. For example, writing a character to the transmit buffer will clear the transmit buffer empty IP.

When the interrupt IP, decoded from the status, is cleared RR2 can be read again. This allows the interrupt routine to clear all of the IP's within one interrupt request to the CPU.

Interrupt With Acknowledge. After the SCC brings INT active, the CPU must respond by bringing INTACK active. After enough time has elapsed to allow the daisy-chain to settle, the SCC will set the IUS bit for the highest priority IP. If the No Vector bit is not set (WR9 bit 1), the SCC will then place the interrupt vector on the data bus during a read. To speed the interrupt response time, the SCC can also modify 3 bits in the vector to indicate status. To include the status, the VIS bit (WR9 bit 0) must be set. The service routine must then clear the interrupting condition. For example, writing a character to the transmit buffer will clear the transmit buffer empty IP. After the interrupting condition is cleared, the routine can read RR3 to determine if any other IP's are set and clear them. At the end of the interrupt routine, a Reset IUS command (WR0) must be issued to unlock the daisy-chain and allow lower-priority interrupt requestes. This is the only way, short of a software or hardware reset, that an IUS bit may be reset.

If the No Vector bit (WR9 bit 1) is set, the SCC will not place the vector on the data bus. An interrupt controller must then vector the code to the interrupt routine. The interrupt routine must then read RR2 from Channel B to read the status. This is the same as the case of an interrupt without an acknowledge except the IUS is set and the vector will not change until the Reset IUS command in RR0 is issued.

Interrupt Sources. Each channel in the SCC contains 3 sources of interrupt, making a total of 6. These 3 sources of interrupts are the receiver, the transmitter, and External/ Status conditions. In addition, there are several conditions that may cause these interrupts.

The receive interrupt request may either be caused by a receive character available or a special condition. The receive character available interrupt is generated when a character is loaded into the FIFO and is ready to be read. The special conditions are receive FIFO overrun, CRC/ framming error. End of frame, and parity. The parity special condition can be included as a special condition or not depending on bit 2 WR1. The special condition status can be read from RR1.

The transmit interrupt request has only one source. It can only be set when the transmit buffer goes from

I/O PROGRAMMING CAPABILITIES (cont'd)

full to empty. Note that this means that the transmit interrupt will not be set until after the first character is written to the SCC.

The External/ status Interrupts have several sources which may be individually enabled in WR15. The sources are zero count, DCD, Sync/ Hunt, CTS, transmitter underrun/ EOM and Break/ Abort.

Each source of interrupt in the SCC has three control/status bits associated with it. There are Interrupt Enable (IE), Interrupt Pending (IP), and Interrupt Under Service (IUS) (see Figure 16). The IE bit is written by the processor and serves to control interrupt requests from the SCC. If the IE bit is set for a given source of interrupt, then that source may cause an interrupt request when all of the necessary conditions are met. If the IE bit is reset, no interrupt request will be generated by that source. The IE bits are write-only in the SCC. The IP bit for a given source of interrupt condition in the SCC and is reset directly by the processor, or indirectly by some action that the processor may take. If the corresponding IE bit is not set, the IP bits in the SCC are read-only in RR3A. The IUS bits are completely hidden from the processor's view. An IUS is set during an Interrupt Acknowledge cycle for the highest-priority IP. See Table 3 for the interrupt priority. IUS is used to control the operation of the interrupt daisy chain by masking lower-priority interrupts. At the end of an interrupt service routine, the processor must issue a Reset Highest IUS command in WR0 to allow lower-priority interrupts. This is the only way, short of a software or hardware reset, that an IUS bit may be reset.

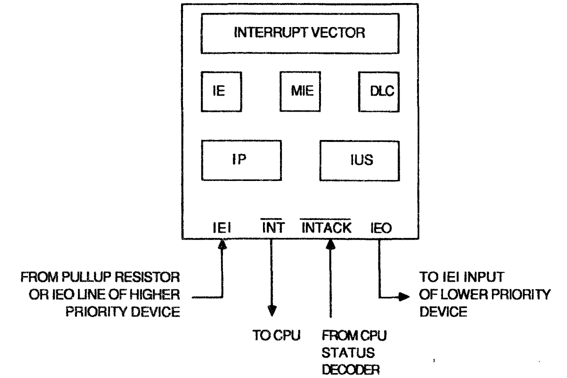
Table 3 : Interrupt Source Priority.

Receiver Channel A Transmit Channel A	High
External / Status Channel A	↓
Receiver Channel B Transmit Channel B	↓
External / Status Channel B	Low

Daisy-chain Priority Resolution. The six sources of interrupt in the SCC are prioritized in a fixed order via a daisy chain ; provision is made, via the IEI and IEO pins, for use of an external daisy chain as well. All Channel A interrupts are higher-priority than any Channel B interrupts, with the receiver, transmitter, and External/ Status interrupts prioritized in that order within each channel. The SCC requests an interrupt by pulling the $\overline{\text{INT}}$ pin Low from its open-drain state. This is controlled by the IP bits and the IEI input, among other things. A flowchart of the interrupt sequence for the SCC is shown in Figure 17. The internal daisy chain links the six sources of interrupt in a fixed order, chaining the IUS bits for each source. While an IUS is set, all lower-priority interrupt requests are masked off ; during an Interrupt Acknowledge cycle the IP bits are also gated into the daisy chain. This insures that the highest-priority IP will be selected to have its IUS set. The internal daisy chain may be controlled by the MIE bit in WR9. This bit, when reset, has the same effect as pulling the IEI pin Low, thus disabling all interrupt requests.

External Daisy Chain Operations. The SCC generates an interrupt request by pulling $\overline{\text{INT}}$ Low, but

Figure 16 : Peripheral Interrupt Structure.


INTERRUPT VECTOR
IE
MIE
DLC
IP
IUS
IEI
$\overline{\text{INT}}$
$\overline{\text{INTACK}}$
IEO
FROM PULLUP RESISTOR OR IEO LINE OF HIGHER PRIORITY DEVICE
TO CPU
FROM CPU STATUS DECODER
TO IEI INPUT OF LOWER PRIORITY DEVICE

I/O PROGRAMMING CAPABILITIES (cont'd)

only if such interrupt requests are enabled (IE is 1, MIE is 1), an IP is set without a higher-priority IUS being set, or no higher-priority IUS being set, or no higher-priority interrupt is being serviced (IEI is High), or no Interrupt Acknowledge transaction is taking place. It is not pulled Low by the SCC at this time, but instead continues to follow IEI until an Interrupt Acknowledge transaction occurs. Some time after $\overline{\text{INT}}$ has been pulled Low, the processor initiates an Interrupt Acknowledge transaction. Between the time the SCC recognizes that an Interrupt Acknowledge cycle is in progress and the time during the acknowledge that the processor requests an interrupt vector, the IEI/IEO daisy chain settles. Any peripheral in the daisy chain having an Interrupt Pending (IP is 1) or an Interrupt Under Service (IUS is 1) holds its IEO line Low and all others make IEO follow IEI.

When the processor requests an interrupt vector, only the highest-priority interrupt source with a pending interrupt (IP is 1) has its IEI input High, its IE bit set to "1", and its IUS bit set to "0". This is the interrupt source being acknowledged, and at this point it sets its IUS bit to "1". If its NV bit is "0", the SCC identifies itself by placing the interrupt vector from WR2 on the data bus. If the NV bit is "1", the SCC data bus remains floating, allowing external logic to supply a vector. If the VIS bit in the SCC is "1", the vector also contains status information, encoded as shown in Table 4, which further describes the nature of the SCC interrupt. If the VIS bit is "0", the vector held in WR2 is returned without modification. If the SCC is programmed to include status information in the vector, this status may be encoded and placed in either bits 1-3 or in bits 4-6. This operation is selected by programming the Status High/Status Low bit in WR9. At the end of the interrupt service routine, the processor should issue the Reset Highest IUS command to unlock the daisy chain and allow lower-priority interrupt requests. The IP is reset during the interrupt service routine either directly by command, or indirectly, through some action taken by the processor. The external daisy chain may be controlled by the DLC bit in WR9. This bit, when set, forces IEO Low, disabling all lower-priority devices.

Receive Interrupts. The Receive Interrupt mode is controlled by WR1 bits 4 and 3. These select one of the four interrupt modes. The four modes are, Interrupt disabled, Interrupt on first character or special conditions, Interrupt on all characters or special conditions, and Interrupt on special conditions.

Receive Interrupts Disabled. This mode prevents the receiver from requesting an interrupt. It is used in a polled environment where either the status bits in RR0 or the modified vector in RR2 (Channel B) is read. Although the receiver interrupts are disabled, the interrupt logic can still be used to provide status.

Table 4 : Interrupt Vector Modification.

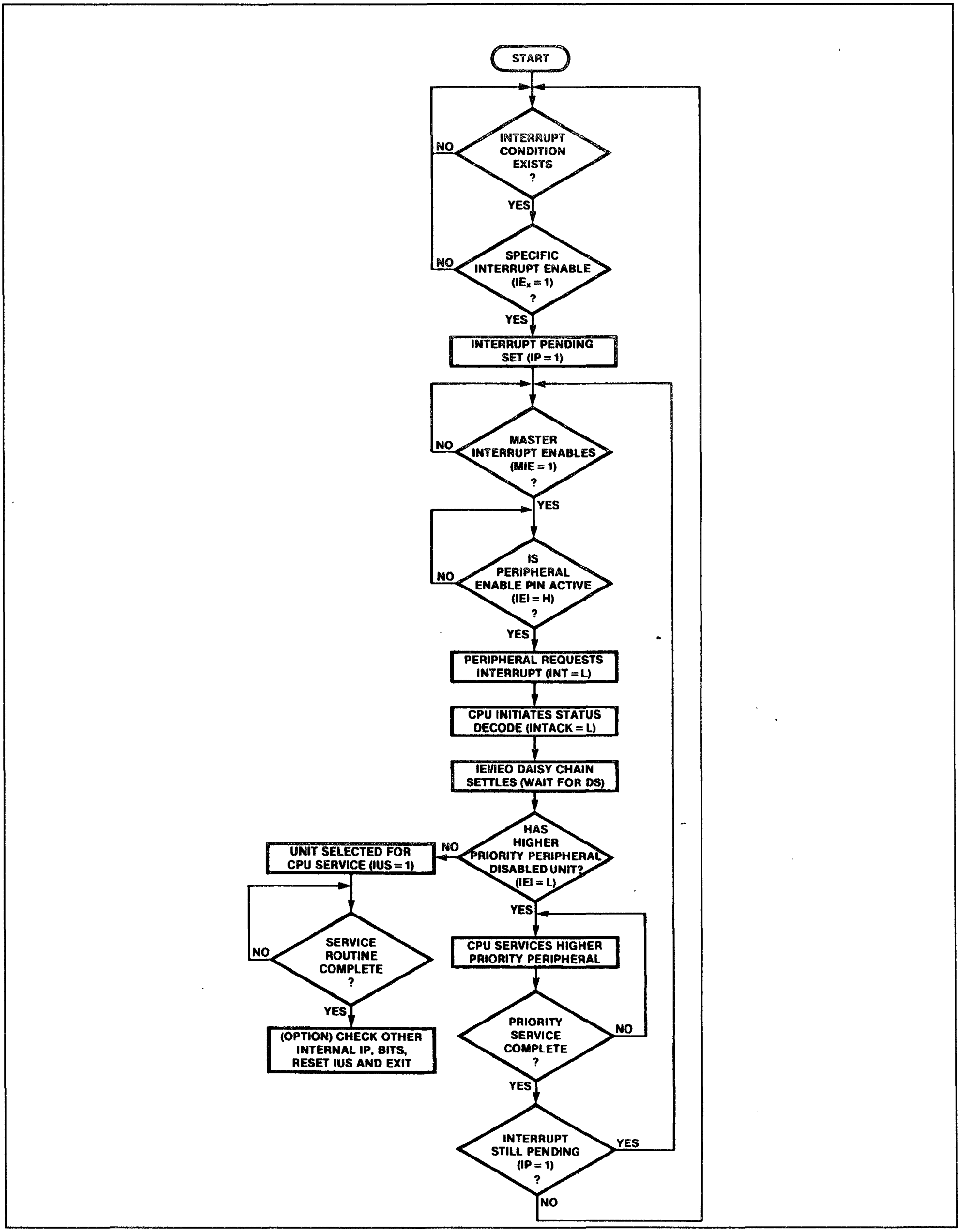
V3	V2	V1	Status High / status Low = 0
V4	V5	V6	Status High / status Low = 1
0	0	0	Ch B Transmit Buffer Empty
0	0	1	Ch B External / Status Change
0	1	0	Ch B Receive Character Avail.
0	1	1	Ch B Special Receive Condition
1	0	0	Ch A Transmit Buffer Empty
1	0	1	Ch A External / status Change
1	1	0	Ch A Receive Character Avail.
1	1	1	Ch A Special Recieve Condition

When these bits indicate that a received character has reached the top of the FIFO, the status in RR1 should be checked and then the data should be read. If status is to be checked, it must be done before the data is read, because the act of reading the data pops both the data and error FIFOs.

Receive Interrupt On First Character Or Special Condition. This mode is designed for use with DMA transfers of the receive characters. After this mode is selected, the first character received, or the first character already stored in the FIFO, will set the receiver IP. This IP will be reset when this character is removed from the SCC. No further receive interrupts will occur until the processor issues an Enable Interrupt on Next Receive Character command in WR0 until a special receive condition occurs. The SCC recognizes several special receive conditions. A receive overrun (where a character in the FIFO is written over) is a special receive condition, as is a framing error in Asynchronous mode, or the end-of-frame condition in SDLC mode. In addition, if D2 of WR1 is set, any character with a parity error will generate a special receive condition interrupt. The correct sequence of events when using this mode is to first select the mode and wait for the receive character available interrupt. When the interrupt occurs the processor should read the character and then enable the DMA to transfer the remaining characters. A special receive condition interrupt may occur any time after the first character is received, but is guaranteed to occur after the character having the special condition has been read. The status is not lost in this case, however, because the FIFO is locked by the special condition. In the service routine the processor should read RR1 to obtain the status, and may read the data again if necessary. The FIFO is unlocked by issuing an Error Reset command in WR0. If the special condition was End-

I/O PROGRAMMING CAPABILITIES (cont'd)

Figure 17 : Interrupt Flowchart.


START
INTERRUPT CONDITION EXISTS ?
NO
YES
SPECIFIC INTERRUPT ENABLE (IEx = 1) ?
NO
YES
INTERRUPT PENDING SET (IP = 1)
MASTER INTERRUPT ENABLES (MIE = 1) ?
NO
YES
IS PERIPHERAL ENABLE PIN ACTIVE (IEI = H) ?
NO
YES
PERIPHERAL REQUESTS INTERRUPT (INT = L)
CPU INITIATES STATUS DECODE (INTACK = L)
IEI/IEO DAISY CHAIN SETTLES (WAIT FOR DS)
HAS HIGHER PRIORITY PERIPHERAL DISABLED UNIT? (IEI = L)
NO
YES
UNIT SELECTED FOR CPU SERVICE (IUS = 1)
SERVICE ROUTINE COMPLETE ?
NO
YES
(OPTION) CHECK OTHER INTERNAL IP, BITS, RESET IUS AND EXIT
CPU SERVICES HIGHER PRIORITY PERIPHERAL
PRIORITY SERVICE COMPLETE ?
NO
YES
INTERRUPT STILL PENDING (IP = 1) ?
YES
NO

I/O PROGRAMMING CAPABILITIES (cont'd)

of-Frame, the processor should now issue the Enable Interrupt on Next Receive Character command to prepare for the next frame. The first character interrupt and special condition interrupt are distinguished by the status included in the interrupt vector. In all other respects they are identical, including sharing the IP and IUS bits.

Interrupt On All Receive Characters Or Special Conditions. This mode is designed for an interrupt-driven system. In this mode the SCC will set the receiver IP on every received character, whether or not it has a special receive condition. This includes characters already in the FIFO when this mode is selected. In this mode of operation the IP is reset when the character is removed from the FIFO, so if the processor requires status for any character, this status must be read before the data is removed from the FIFO. The special receive conditions are identical to those previously mentioned, and as before, the only difference between a "receive character available" interrupt and a "special receive condition" interrupt is the status encoded in the vector. In this mode a special receive condition does not lock the receive data FIFO so that the service routine must read the status in RR1 before reading the data. At moderate to high data rates, where the interrupt overhead is significant, time can usually be saved by checking for another received character before exiting the service routine. This technique eliminates the Interrupt Acknowledge and the processor-state-saving time, but care must be exercised because this receive character must be checked for special receive conditions before it is removed from the SCC.

Receive Interrupt On Special Conditions. This mode is designed for use with DMA transfers of the receive characters. In this mode, only receive characters with special conditions will cause the receiver IP to be set. All other characters are assumed to be transferred via DMA. No special initialization sequence is needed in this mode. Usually the DMA is initialized and enabled, and then this mode is selected in the SCC. A special receive condition interrupt may occur at any time after this mode is selected but the logic guarantees that the interrupt will not occur until after the character with the special condition has been read from the SCC . The special condition locks the FIFO so that the status will be valid when read in the interrupt service routine, and it guarantees that the DMA will not transfer any characters until the special condition has been serviced. In the service routine the processor should read RR1 to obtain the status and unlock the FIFO by issuing an Error Reset command. DMA transfer of the receive characters will then resume.

Transmit Interrupts. Transmit interrupts are controlled by the Transmit Interrupt Enable bit (D1) in WR1. If the interrupt capabilities of the SCC are not required, polling may be used. This is selected by disabling the transmit interrupts and polling the Transmit Buffer Empty bit in RR0. When the Transmit Buffer Empty is set a character may be written to the SCC without fear of writing over previous data. Another way of polling the SCC is to enable the transmit interrupt and then reset the MIE bit in WR9. The processor may then poll the IP bits in RR3A to determine when the transmit buffer is empty. Transmit interrupts should also be disabled in the case of DMA transfer of the transmitted data.

While the transmit interrupts are enabled the SCC will set the transmit IP whenever the transmit buffer becomes empty. This means that the transmit buffer must have been full before the transmit IP can be set. Thus when the transmit interrupts are first enabled, the transmit IP will not be set until after the first character is written to the SCC. In synchronous modes one other condition can cause the transmit IP to be set. This occurs at the end of a transmission after CRC is sent. When the last bit of CRC has cleared the Transmit Shift register and the flag or sync character is loaded into the Transmit Shift register, the SCC will set the transmit IP. Data for the new frame or block to be transmitted may be written at this time. In this particular case the Transmit Buffer Empty bit in RR0 is not set ; only the transmit IP is set. If the transmit Buffer Empty bit is, in fact, set for the transmit interrupt that occurs immediately after CRC transmission, this indicates that data was written while CRC was being sent. This is an indication that the transmitter underflowed, without the CPU being aware of it. The transmit IP is reset either by writing data to the transmit buffer or by issuing the Reset Transmit IP command in WR0. Ordinarily the response to a transmit interrupt is to write more data to the SCC ; however, at the end of a frame or block of data where CRC is to be sent next, the Reset Transmit IP command should be issued in lieu of data.

External/Status Interrupts. There are several sources of External/Status interrupts, each of which may be individually enabled in WR15. The master enable for the External/Status interrupts is located in WR1 (D0). The individual enable bits in WR15 control whether or not latches will be present in the path from the source of interrupt to the status bit in RR0. If an individual enable bit in WR15 is set to "0" the latches are not present in the signal path and the value read in RR0 reflects the current status. An interrupt source whose individual enable in WR15 is "0" is not a source of External/Status interrupts

I/O PROGRAMMING CAPABILITIES (cont'd)

even though the External/Status Interrupt Enable bit is set. When an individual enable in WR15 is set to "1", the latch is present in the signal path. The latches for the sources of External/Status interrupts are not independent. Rather, they all close at the same time as a result of a state change by one of the sources of interrupt. Thus, a read of RR0 returns the current status for any bits whose individual enable is "0" and either the current state or the latched state of the remainder of the bits. To guarantee the current status the processor should issue a Reset External/Status Interrupts command in WR0 to open the latches. The External/Status IP is set by the closing of the latches and remains set as long as they are closed. If the master enable for the External/Status interrupts is not set, the IP will never be set, even though the latches may be present in the signal paths and working as described. Because the latches close on the current status but give no indication of change, the processor must maintain a copy of RR0 in memory. When the SCC generates an External/Status interrupt the processor should read RR0 and determine which condition changed state and take appropriate action. The copy of RR0 in memory must then updated and the Reset External/Status Interrupt command issued. Care must be taken in writing the interrupt service routine for the External/Status interrupts because it is possible for more than one status condition to change state at the same time. All of the latch bits in RR0 should be compared to the copy of RR0 in memory. If none have changed and the ZC interrupt is enabled, the Zero Count condition caused the interrupt.

The operation of the individual enable bits in WR15 for each of the six sources of External/Status interrupts is identical, but subtle differences exist in the operation of each source of interrupt. The six sources are Break/Abort, Underrun/EOM, CTS, DCD, Sync/Hunt and Zero Count. The Break/Abort, Underrun/EOM, and Zero Count conditions are internal to the SCC, while Sync/Hunt may be internal or external, and CTS and DCD are purely external signals. In the following discussions each source is assumed to be enabled, so that the latches are present, and the External/Status interrupts are enabled as a whole. Recall that the External/Status IP is set while the latches are closed and that the state of the signal is reflected immediately in RR0 if the latches are not present.

The Break/Abort status is used in asynchronous and SDLC modes but is always "0" in synchronous modes other than SDLC. In asynchronous modes this bit is set when a break sequence (null character plus framing error) is detected in the receive data stream, and remains set as long as "0s" continue to be received. This bit is reset when a "1" is received. A single null character is left in the receive FIFO each time that the break condition is terminated. This character should be read and discarded. In SDLC mode this bit is set by the detection of an abort sequence, which is seven or more contiguous "1s" in the receive data stream. The bit is reset when a "0" is received. A received abort forces the receiver into Hunt, which is also an external/status condition. Though these two bits change state at roughly the same time, one or two External/ Status interrupts may be generated as a result. The Break/Abort bit is unique in that both transitions are guaranteed to cause the latches to close, even if another External/ Status interrupt is pending at the time these transitions occur. This guarantees that a break or abort will be caught.

The Transmit Underrun/ EOM bit is used in synchronous modes to control the transmission of CRC. This bit is reset by issuing the Reset Transmit Underrun/ EOM command in WR0. However, this transition does not cause the latches to close ; this occurs only when the bit is set. To inform the processor of this fact, the SCC sets this bit when CRC is loaded into the Transmit Shift register. This bit will also be set if the processor issues the Send Abort command in WR0. The bit is always set in Asynchronous mode.

The CTS bit reports the state of the $\overline{\text{CTS}}$ input, and the DCD bit reports the status of the DCD input. Both bits latch on either input transition. In both cases, after the Reset External/ Status Interrupt command is issued, if the latches are closed, they remain closed if there is any odd number of transitions on an input ; they will be open if there is an even number of transi-tions on the input.

The Zero Count bit is set when the counter in the baud rate generator reaches a count of "0" and is reset when the counter is reloaded. The latches are closed only when this bit is set to "1", and the status in RR0 always reflects the current status.While the Zero Count IE bit in WR15 is reset this bit is forced to "0" .

There are a variety of ways in which the Sync/ Hunt may be set and reset, depending on the SCC's mode operation. In Asynchronous mode this bit reports the state of the $\overline{\text{SYNC}}$ pin, latching on both input transitions. The same is true of External Sync mode. However, if the crystal oscillator is enabled while in Asynchronous mode this bit will be forced to "0" and the latches will not be closed. Selecting the crystal option in External Sync mode is illegal, but the result will be the same. In Synchronous modes other than SDLC the Sync/Hunt reports the

I/O PROGRAMMING CAPABILITIES (cont'd)

Hunt state of the receiver. Hunt mode is entered when the processor issues the Enter Hunt command in WR3. This forces the receiver to search for a sync character match in the receive data stream. Because both transitions of the Hunt bit close the latches, issuing this command will cause an External/Status interrupt. The SCC resets this bit when character synchronization has been achieved, causing the latches to again be closed. In these synchronous modes the SCC will not reenter the Hunt mode automatically ; only the Enter Hunt command will set this bit. In SDLC mode this bit is also set by the Enter Hunt command, but the receiver will also automatically enter the Hunt mode if an Abort sequence is received. The receiver leaves Hunt upon receipt of a flag sequence. Both transitions of the Hunt bit will cause the latches to be closed. In SDLC mode the receiver will automatically synchronize on Flag characters. The receiver is in Hunt mode when it is enabled, so the Enter Hunt command will probably never be needed.

If careful attention is paid to details, the interrupt service routine for External/Status interrupts is straightforward. To determine which bit or bits changed state, the routine should first read RR0 and compare it to a copy from memory. For each changed bit the appropriate action should be taken and the copy in memory updated. The service routine should close with a Reset External/Status interrupts command to reopen the latches. The copy of RR0 in memory should always have the Zero Count bit set to "0", since this will be the state of the bit after the Reset External/Status interrupts command at the end of the service routine. When the processor issues the Reset Transmit Underrun/EOM latch command in WR0, the Transmit Underrun/EOM bit in the copy of RR0 in memory should be reset because this transition does not cause an interrupt.

Block Transfers

The SCC offers several alternatives for the block transfer of data. The various options are selected by WR1 (bits D7 through D5) and WR14 (bit D2). Each channel in the SCC has two pins which may be used to control the block transfer data. Both pins in each channel may be programmed to act as DMA Request signals, and one pin in each channel may be programmed to act as a Wait signal for the CPU. In either mode, it is advisable to select and enable the mode in two separate accesses of the appropriate register. The first access should select the mode and the second access should enable the function. This procedure prevents glitches on the output pins. Reset forces Wait mode, with W/ REQ open-drain.

Wait On Transmit. The Wait function on transmit is selected by setting both D6 and D5 to "0" and then enabling the function by setting D7 of WR1 to "1". In this mode the W/ REQ pin carries the WAIT signal, and is open-drain when inactive and Low when active. When the processor attempts to write to the transmit buffer when it is full, the SCC will assert WAIT until the buffer is empty. This allows the use of a block-move instruction to transfer the transmit data. In the Z8530, WAIT will go active in res-ponse to WR going active, but only if the data buf-fer is being accessed, either directly or via the pointers. The WAIT pin is released in response to the falling edge of PCLK. Details of the timing are shown in Figure 18.

Wait On Receive. The Wait function on receive is selected by setting D6 or WR1 to "0", D5 of WR1 to "1", and then enabling the function by set-tling D7 of WR1 to "1". In this mode the W/ REQ pin carries the WAIT signal, and is open-drain when inactive and Low when active. When the processor attempts to read data from the receive FIFO when it is empty, the SCC will assert WAIT until a character has reached the top of the FIFO. This allows the use of a block-move instruction to transfer the receive data. In the Z8530, WAIT will go active in res-ponse to RD going active, but only if the receive data FIFO is being accessed, either directly or via the pointers. The WAIT pin is released in response to the falling edge of PCLK. Details of the timing are shown in Figure 19.

DMA Requests. The two DMA request pins W/ REQ and DTR/ REQ can be programmed to be used as DMA requests. The W/ REQ pin can be used as either a transmit or a receive request and the DTR/ REQ pin can only be used as a receive request. For full-duplex operation, the W/ REQ is, therefore, used for transmit and the DTR/ REQ is used for receive. These modes are described below.

DMA Request On Transmit (using W/REQ). The Request on Transmit function is selected by setting D6 of WR to "1", D5 of WR1 to "0", and then enabling the function by setting D7 of WR1 to "1". In this mode the W/ REQ pin carries the REQUEST signal, which is active Low. When this mode is selected, but not yet enabled, the W/ REQ is driven High. When the enable bit is set, REQUEST goes Low if the transmit buffer is empty at the time, or will remain High until the transmit buffer becomes empty. Note that the REQUEST pin will follow the state of the transmit buffer even though the transmitter is disabled. Thus, if the REQUEST is enabled, the DMA may write data to the SCC before the transmitter is enabled. This will not cause a problem in Asynchro-

I/O PROGRAMMING CAPABILITIES (cont'd)

nous mode but may cause problems in Synchronous mode because the SCC will send data in preference to flags or sync characters. It may also complicate the CRC initialization, which cannot be done until after the transmitter is enabled. With only one exception, the REQUEST pin directly follows the state of the transmit buffer in this mode. REQUEST goes Low when the transmit buffer empties and remains Low until the transmit buffer is filled. The SCC generates only one falling edge on REQUEST per character requested and the timing for this is shown in Figure 20. The one exception occurs in synchronous modes at the end of CRC transmission. At the end of CRC transmission, when the closing flag or sync character is loaded into the Transmit Shift register, REQUEST is pulsed High for one PCLK cycle. The DMA may use this falling edge on REQUEST to write the first character of the next frame or block to the SCC.

In the Z8530, REQUEST will go High in response to the falling edge of WR, but only when the appropriate transmit buffer in the SCC is accessed. This is shown in Figure 21.

DMA Request On Transmit (using DIR/ REQ). A second Request on Transmit function is available on the DTR/ REQ pin. This mode is selected by setting D2 of WR14 to "1". When this bit is set to "1", REQUEST goes Low if the transmit buf-fer is empty at the time, or will go High until the transmit buffer becomes empty. While D2 of WR14 is set to "0", the DTR/ REQ pin is DTR and follows the inverted state of D7 in WR5. This pin will be High after a channel or hardware reset and in the DTR mode. In the Request mode REQUEST will follow the state of the transmit buffer even though the transmitter is disabled. Thus if REQUEST is enabled before the transmitter is enabled, the DMA may write data to the SCC before the transmitter is enabled. This will not cause a problem in Asynchronous mode, but may cause problems in Synchronous mode because the SCC will send data in preference to flags or sync characters. It may also complicate the CRC initialization, which cannot be done until after the transmitter is enabled. With only one exception, the REQUEST pin directly follows the state of the transmit buffer in this mode. REQUEST goes Low when the transmit buffer empties and remains Low until the transmit buffer is filled. The SCC generates only one falling edge on REQUEST per character requested. The one exception occurs in synchronous modes at the end of CRC transmission. At the end of CRC transmission, when the closing flag or sync character is loaded into the Transmit Shift register, REQUEST is pulsed High for one PCLK cycle. The DMA may use this falling edge on REQUEST to write the first character of the next frame or block to the SCC. The Request signal on DTR/ REQ differs from the one on W/ REQ in that it does not go immediately High in response to the access which writes to the transmit buffer. This is because the registers in the SCC are not written during the actual access, but are delayed by some number of PCLK cycles. The Request signal on DTR/ REQ follows the state of the transmit buffer exactly while the Request signal on W/ REQ goes inactive in anticipation of the transmit buffer becoming full. The timing of the Request signal on both pins is shown in Figure 21.

DMA Request On Receive. The Request on Receive function is selected by setting D6 and D5 of WR1 to "1" and then enabling the function by setting D7 of WR1 to "1". In this mode the W/ REQ pin carries the REQUEST signal, which is active Low. When this mode is selected, but not yet enabled, the W/ REQ pin is driven High. When the enable bit is set REQUEST goes Low if the receive buffer contains a character at the time, or will remain High until a character enters the receive buffer. Note that the REQUEST pin will follow the state of the receive buffer even though the receiver is disabled. Thus, if the receiver is disabled and REQUEST is still enabled, the DMA will transfer the previously received data correctly. In this mode the REQUEST pin directly follows the state of the receive buffer with only one exception. REQUEST goes Low when a character enters the receive buffer and remains Low until this character is removed from the receive buf-fer. The SCC generates only one falling edge on REQUEST per character transfer requested and the timing for this is shown in Figure 22. The one exception occurs in the case of a special receive condition in the Receive Interrupt on First Character or Special Condition mode, or the Receive Interrupt on Special Condition Only mode. In the two interrupt modes any receive character with a special receive condition is locked at the top of the FIFO until an Error Reset command is issued. This character in the receive FIFO would ordinarily cause additional DMA Requests after the first time it is read. However, the logic in the SCC guarantees only one falling edge on REQUEST by holding REQUEST High from the time the character with the special receive condition is read, and the FIFO locked, until after the Error Reset command has been issued. Once the FIFO is unlocked by the Error Reset command, REQUEST again follows the state of the receive buffer. In the Z8530, REQUEST will go High in response to the falling edge of RD, but only when the appropriate receive buffer in the SCC is accessed. This is shown in Figure 23.

I/O PROGRAMMING CAPABILITIES (cont'd)

Figure 18 : Wait on Transmit.

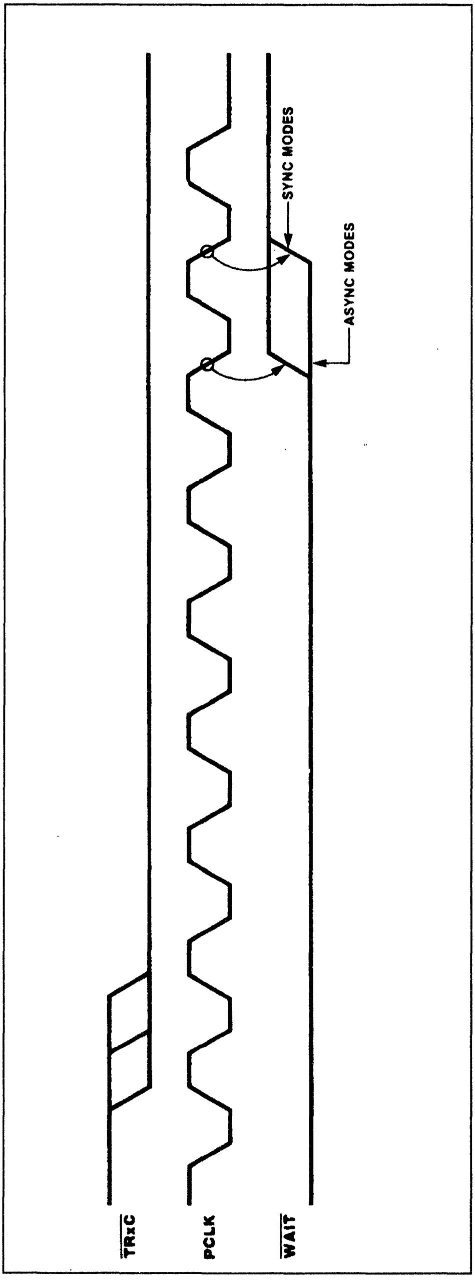
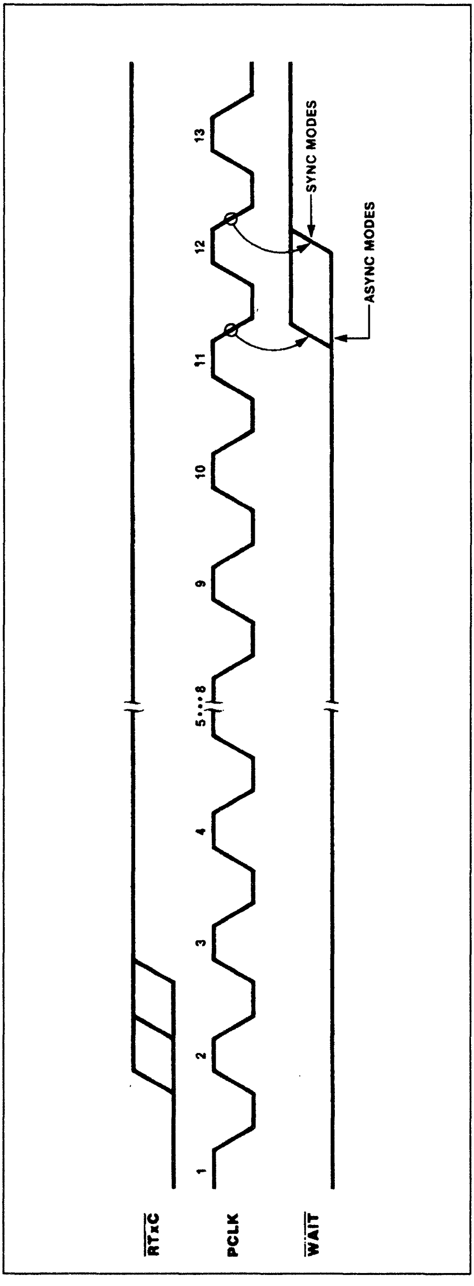


Figure 19 : Wait on Receive.



I/O PROGRAMMING CAPABILITIES (cont'd)

Figure 20 : Transmit Request Assertion.

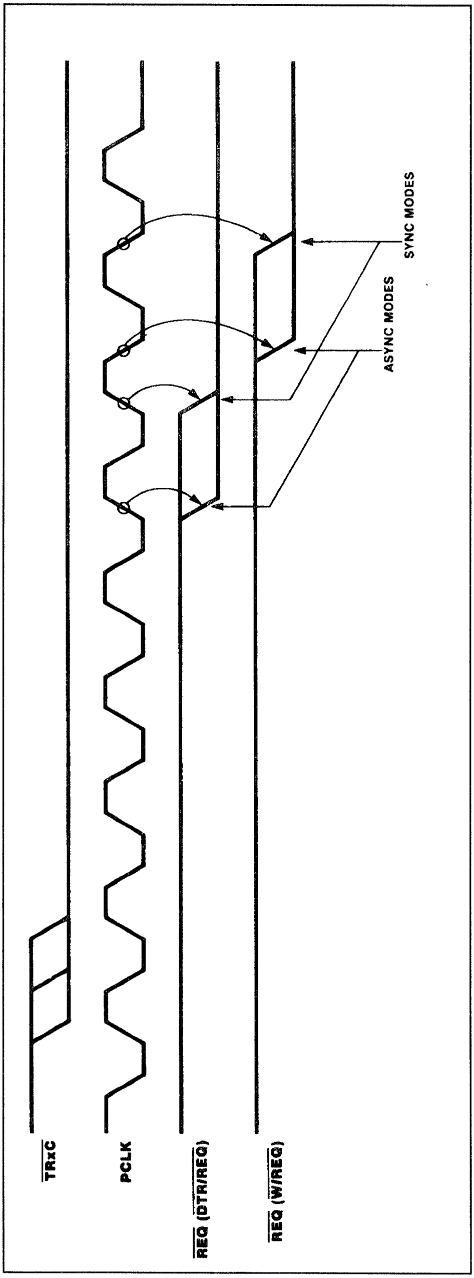
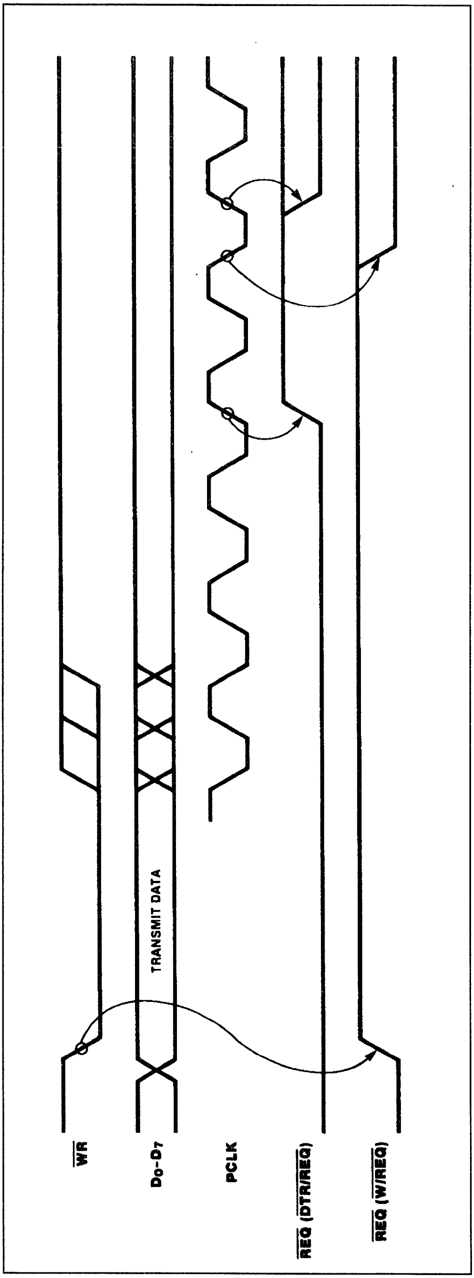


Figure 21 : Transmit Request Release.



I/O PROGRAMMING CAPABILITIES (cont'd)

Figure 22 : Receive Request Assertion.

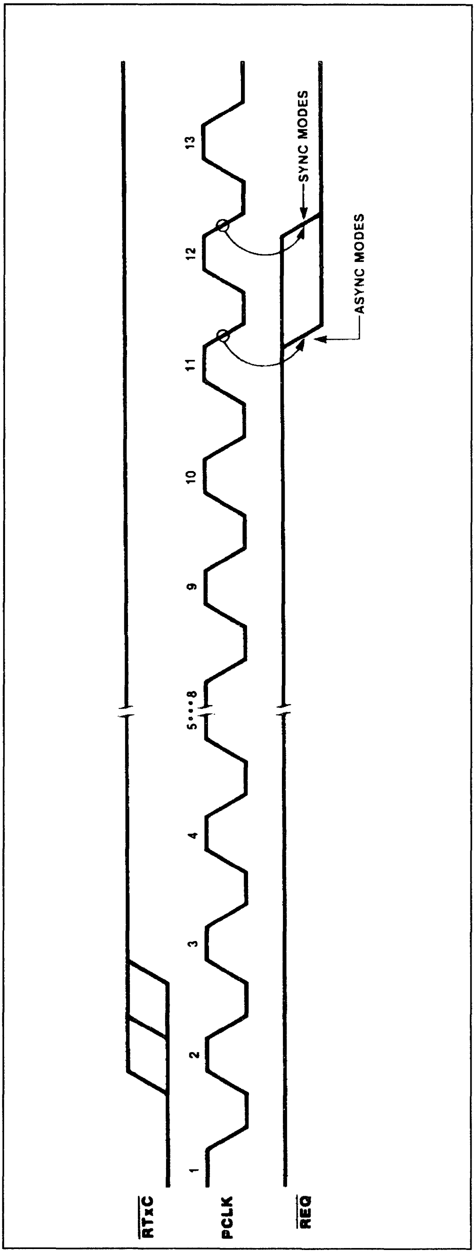
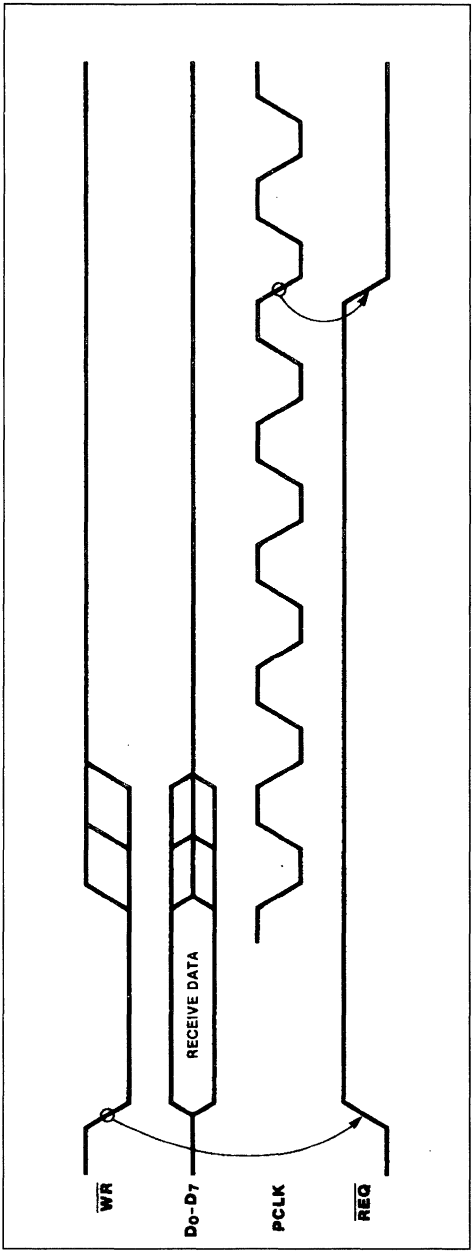


Figure 23 : Z8530 Receive Request Release.



PROGRAMMING DATA COMMUNICATION MODES

The SCC provides two independent full-duplex channels programmable for use in any common asynchronous or synchronous data communication protocol. These include asynchronous, synchronous byte-oriented protocols, monosync, IBM Bisync, and bit-oriented protocols such as HDLC and SDLC. This chapter is divided into 3 parts : Asynchronous, Synchronous, and SDLC.

Asynchronous Mode

The SCC supports Asynchronous mode with a number of programmable options including the number of bits per character, the number of stop bits, the clock factor, modem interface signals and break detect and generation. Asynchronous mode is selected by programming the desired number of stop bits in D_3 and D_2 of WR4. Programming these two bits with other than "00" places both the receiver and transmitter in Asynchronous mode. In this mode, the SCC ignores the state of bits D_4, D_3, D_2, and D_1 of WR3, bits D_5 and D_4 of WR4, bits D_2 and D_0 of WR5, all of WR6 and WR7 and all of WR10 except D_6 and D_5. Bits that are ignored may be programmed with "1" or "0" or not at all.

Asynchronous Receive. Asynchronous mode is selected by specifying the number of stop bits per character in WR4. This selection applies only the transmitter, however, as the receiver always checks for one stop bit. If after character assembly the receiver finds this stop bit to be a "0", the Framing Error bit in the receive error FIFO is set at the same time that the character is transferred to the receive data FIFO. This error bit accompanies the data to the top of the FIFO, where it generates a special receive condition. The Framing Error bit is not latched, and so must be read in RR1 before the accompanying data is read.

The number of bits per character is controlled by bits D7 and D6 of WR3. Five, six, seven, or eight bits per character may be selected via these two bits. Data is right-justifed with the unused bits set to "1s". An additional bit, carrying parity information may be selected by setting bit D0 of WR4 to "1". Note that this also enables parity for the transmitter. The parity sense is selected by bit D1 of WR4. If this bit is set to "1", the received character is checked for even parity, if set to "0", the received character is checked for odd parity. The additional parity bit per character is transferred to the receive data FIFO along with the data if the data plus parity is eight bits or less. The parity Error bit in the receive error FIFO may be programmed to cause a special receive condition interrupt by setting bit D2 of WR1 to "1". This error bit is latched and so will remain active, once set, until an Error Reset command has been issued. If interrupts are not used to transfer data, the Parity Error, Framing Error, and Overrun Error bits in RR1 should be checked before the data is removed from the receive data FIFO.

The break condition is continuous "0s", as opposed to the usual continuous ones during an idle. The SCC recognizes the Break condition upon seeing a null character (all "0s") plus a framing error. Upon recognizing this sequence the Break bit in RR0 will be set and will remain set until a "1" is received. At this point the break condition is no longer present. At the termination of a break the receive data FIFO contains a single null character, which should be read and discarded. The Framing Error bit will not be set for this character, but if odd parity has been selected, the Parity Error bit will be set. Caution should be exercised if the receive data line contains a switch that is not debounced to generate breaks. Switch bounce may cause multiple breaks, recognized by the SCC to be additional characters assembled in the receive data FIFO. It may also cause a receive overrun condition being latched.

The SCC may be programmed to accept a receive clock that is one, sixteen, thirty-two, or sixty-four times the data rate. This is selected by bits D7 and D6 in WR4. The 1X mode is used when bits are synchronized external to the receiver. The 1X mode is the only mode in which a data encoding method other than NRZ may be used. The clock factor is common to the receiver and transmitter.

The SCC provides up to three modem control signals associated with the receiver. The $\overline{\text{SYNC}}$ pin is a general-purpose input whose state is reported in the Sync/Hunt bit in RR0. If the crystal oscillator is enabled, this pin is not available and the Sync/Hunt bit is forced to "0". Otherwise, the SYNC pin may be used to carry the Ring Indicator signal. The $\overline{\text{DTR}}$/ $\overline{\text{REQ}}$ pin carries the inverted state of the DTR bit (D7) in WR5 unless this pin has been programmed to carry a DMA Request signal. The $\overline{\text{DCD}}$ pin is ordinarily a simple input to the $\overline{\text{DCD}}$ bit in RR0. However, if the Auto Enables mode is selected by setting D5 of WR3 to "1", this pin becomes an enable for the receiver. That is, if Auto Enables is on and the $\overline{\text{DCD}}$ pin is HIGH, the receiver is disabled. While the $\overline{\text{DCD}}$ pin is LOW, the receiver is enabled.

The initialization sequence for the receiver in Asynchronous mode is : WR4 first to select the mode,then WR3 and WR5 to select the various options. At this point, the other registers should be ini-

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

tialized as necessary. When all of this is complete the receiver may be enabled by setting bit D_0 or WR3 to "1".

Asynchronous Transmit. Asynchronous mode is selected by specifying the number of stop bits per character in bits D_3 and D_2 of WR4. The three options available are one, one-and-a-half, or two stop bits per character. These two bits only select the number of stop bits for the transmitter, as the receiver always checks for one stop bit.

The number of bits per transmitted character is controlled both by Bits D_6 and D_5 in WR5 and the way the data is formatted within the transmit buffer. The bits in WR5 allow the option of five, six seven, or eight bits per character. When five bits per character is selected the data may be formatted before being written to the transmit buffer to allow transmission of from one to five bits per character.

This formatting is shown in Table 5.

Table 5 : Data Format - Five Bits or Less.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
1	1	1	1	0	0	0	0	One data bit
1	1	1	0	0	0	0	0	Two data bits
1	1	0	0	0	0	0	0	Three data bits
1	0	0	0	0	0	0	0	Four data bits
0	0	0	0	0	0	0	0	Five data bits

In all cases the data must be right-justified, with the unused bits being ignored except in the case of five bits per character. An additional bit, carrying parity information, may be automatically appended to every transmitted character by setting bit D0 of WR4 to "1". This bit is sent in addition to the number of bits specified in WR4 or by the data format. The parity sense is selected by bit D1 of WR4. If this bit is set to "1", the transmitter sends even parity, if set to "0", the parity is odd.

The transmitter may be programmed to send a Break by setting bit D4 of WR5 to "1". The transmitter will send continuous "0s" from the first transmit clock edge after this command is issued, until the first transmit clock edge after this bit is reset. The transmit clock edges referred to here are those that define transmitted bit cell boundaries.

An additional status bit for use in Asynchronous mode is available in bit D_0 or RR1. This bit, called All Sent, is set when the transmitter is completely empty and any previous data or stop bits have reached the TxD pin. The All Sent bit can be used by the processor as an indication that the transmitter may be safely disabled.

The SCC may be programmed to accept a transmit clock that is one, sixteen, thirty-two, or sixty-four time the data rate. This is selected by bits D_7 and D_6 of WR4, in common with the clock factor for the receiver. Note that the chosen clock factor may restrict the number of stop bits that may be transmitted. In particular, when the clock rate and data rate are identical, one-and-a-half stop bits are not allowed. If any length other than one stop bit is desired in the times one mode, only two stop bits may be used.

There are two modem control signals associated with the transmitter provided by the SCC. The $\overline{RTS}$ pin is a simple output that carries the inverted state of the RTS bit (D_1) in WR5, unless the Auto Enables bit (D_5) is set in WR3. When Auto Enables is set the RTS pin will immediately go LOW when the RTS bit is set. However, when the RTS bit is reset the $\overline{RTS}$ pin remains LOW until the transmitter is completely empty and the last stop bit has left the TxD pin. Thus the $\overline{RTS}$ pin may be used to disable external drivers for the transmit data. The $\overline{CTS}$ pin is ordinarily a simple input to the CTS bit in RR0. However, if Auto Enables mode is selected this pin becomes an enable for the transmitter. That is, if Auto Enables is on and the $\overline{CTS}$ pin is HIGH, the transmitter is disabled ; the transmitter is enabled while the $\overline{CTS}$ pin is LOW.

The initialization sequence for the transmitter in Asynchronous mode is : WR4 first to select the mode, then WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is complete, the transmitter may be enabled by setting bit D3 of WR5 to "1". Note that the transmitter and receiver may be initialized at the same time.

Synchronous Mode

In synchronous modes of operation a special bit pattern is used to provide character synchronization. The SCC offers several options to support Synchronous mode including various sync character lengths, the number of bits per data character, parity generation and checking, CRC generation and checking, as well as modem controls and a transmitter to receiver synchronization function. Synchronous mode is selected by programming bits D_3 and D_2 of WR4 with "Os". This selects Synchronous mode, as opposed to Asynchronous mode, but this selection is further modified by bits D_5 andD_7 of WR4 as well as bits D_1 and D_0 of WR10.

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

The sync character or characters are written in WR6 and WR7. In all synchronous modes, except External Sync the state of bits D_7 and D_6 of WR4 are forced to "0" to select the times one clock mode. In External Sync mode these two bits must be programmed with "0s".

Synchronous Receive. The receiver in the SCC searches for character synchronization only while it is in Hunt mode. In this mode the receiver is idle except that it is searching the incoming data stream for a sync character match. The receiver is in Hunt mode when it is first enabled, and may be placed in Hunt mode by command from the processor. This is accomplished by issuing the Enter Hunt Mode command in WR3. This bit (D_4) is a command ; writing a "0" to it has no effect. The Hunt status of the receiver is reported by the Sync/ Hunt bit in RR0. Sync/Hunt is one of the possible sources of external/status interrupts, with both transitions causing an interrupt. This is true even if the Sync/ Hunt bit is set as a result of the processor issuing the Enter Hunt Mode command.

An 8-bit sync character is selected by setting bits D_5 and D_4 of WR4, as well as bit D0 of WR10, to "0". With this option the receiver searches the data stream for a match will the eight bits in WR7. The 6-bit sync option requires the same programming except that D0 of WR10 is set to "1" and the sync character is held in the high-order six bits of WR7. The SCC also allows the option of double length sync characters. This is selected by setting bit D_5 of WR4 to "0" and bit D_4 of WR4 to "1". The selection between 12 and 16 bits of sync character is controlled by bit D0 of WR10. A "0" selects 16 bits of sync character, while a "1" in this bit selects a 12-bit sync character. The arrangement of the sync character in WR6 and WR7 is shown in Figure 24. For those applications requiring any other sync character length, the SCC makes provision for an external circuit to provide a character synchronization signal on the $\overline{\text{SYNC}}$ pin. This mode is selected by setting bit D_5 and D_4 of WR4 to "1". In this mode the Sync/Hunt bit in RR0 reports the state of the $\overline{\text{SYNC}}$ pin but the receiver must still be placed in Hunt mode when the external logic is searching for a sync character match. When the receiver is in Hunt mode and the $\overline{\text{SYNC}}$ pin is driven LOW, two receive clock cycles after the last bit of the sync character is received, character assembly will begin on the rising edge of the receive clock immediately preceding the activation of $\overline{\text{SYNC}}$. This is shown in Figure 25. The receiver leaves Hunt mode when $\overline{\text{SYNC}}$ is driven LOW. In all cases except External Sync mode the $\overline{\text{SYNC}}$ pin is an output that is driven LOW by the SCC to signal that a sync character has been received. The $\overline{\text{SYNC}}$ pin is activated regardless of character boundaries so any external circuitry using it should only respond the $\overline{\text{SYNC}}$ pulse that occurs while the receiver is in Hunt mode. The timing for the $\overline{\text{SYNC}}$ signal is shown in Figure 26.

The number of bits per character is controlled by bits D_7 and D_6 of WR3. Five, six, seven, or eight bits per character may be selected via these two bits. The data is right-justified in the receive data buffer. The SCC merely takes a snapshot of the receive data stream at the appropriate times so the "unused" bits in the receive buffer are only the bits following the character in the data stream. An additional bit, carrying parity information, may be selected by setting bit D_0 of WR4 to "1". If this bit is set to "1", the received character is checked for even parity, if set to "0", the received character is checked for odd parity. The additional bit per character is visible in the receive data FIFO if the data plus parity is eight bits or less. The parity bit is not visible when there are eight data bits per character. The Parity Error bit in the receive error FIFO may be programmed to cause a Special Receive Condition interrupt by setting bit D_2 of WR1 to "1". This error bit is latched and so will remain active, once set, until an Error Reset command has been issued. If interrupts are not used to transfer data the Parity Error, CRC Error, and Overrun Error bits in RR1 should be checked before the data is removed from the receive data FIFO. The character lenght may be changed at any time before the new number of bits has been assembled by the receiver, but, care should be exercised as unexpected results may occur. A representative example, switching from five bits to eight bits and back to five bits is shown in Figure 27. It is sometimes desirable to prevent sync characters in the receive data stream from being transferred to the receive data FIFO. This function is available in the SCC by setting the Sync Character Load Inhibit bit (D1) in WR3 to "1". While this bit is set to "1", character about to be loaded into the receive data FIFO is compared with the contents of WR6. If all eight bits match the character, it is not loaded into the receive data FIFO. Because the comparison is across eight bits, this function works correctly only when the number of bits per character is the same as the sync character length. Thus it cannot be used with 12- or 16-bit sync characters. Both leading sync characters and sync characters embedded in the data will be properly removed in the case of an 8-bit sync character, but only the leading sync characters may be properly removed in the case of a 6-bit sync character. Care must be exercised in using this feature be-

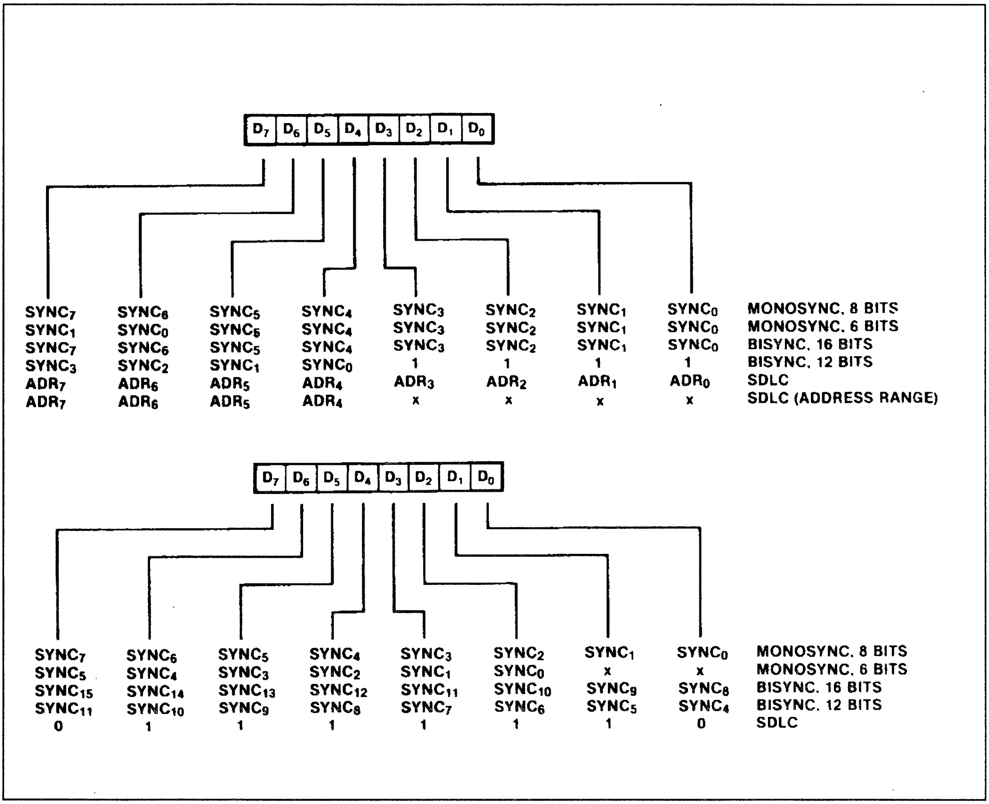
PROGRAMMING DATA COMMUNICATION MODES (cont'd)

cause sync characters not transferred to the receive data FIFO will automatically be excluded from CRC calculation. This works properly only in the 8-bit case.

Either of two CRC polynomials may be used in synchronous modes, selected by bit D_2 in WR5. If this bit is set to "1", the CRC-16 polynomial is used, if this bit is set to "0", the CRC-CCITT polynomial is used. This bit controls the polynomial selection for both the receiver and transmitter. The initial state of the generator and checker is controlled by bit D_7 of WR10. When this bit is set to "1", both the generator and checker will have an initial value of all ones, if this bit is set to "0", the initial values will be all "0s". The SCC presets the checker whenever the receiver is in Hunt mode so a CRC reset command is not strictly necessary. However, the CRC checker may be preset by issuing the Reset CRC Checker command in WR0. This command is encoded in bits D_7 and D_6 of WR0. If CRC is to be used the CRC checker must be enabled by setting bit D0 of WR3 to "1". If sync characters are being stripped from the data stream, this may be done at any time before the first non-sync character is received. If the sync strip feature is not being used, CRC must not be enabled until after the first data character has been transferred to the receive data FIFO. As previously mentioned, 8-bit sync characters stripped from the data stream are automatically excluded from CRC calculation.

Some synchronous protocols require that certain characters be excluded from CRC calculation. This is possible in the SCC because CRC calculation may be enabled and disabled on the fly. To give the processor sufficient time to decide whether or not a particular character should be included in the CRC calculation, the SCC contains an 8-bit time delay between the receive shift register and the CRC

Figure 24 : Sync Character Programming.



D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 8 BITS
SYNC$_1$	SYNC$_0$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 6 BITS
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	BISYNC, 16 BITS
SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	1	1	1	1	BISYNC, 12 BITS
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	ADR$_3$	ADR$_2$	ADR$_1$	ADR$_0$	SDLC
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	x	x	x	x	SDLC (ADDRESS RANGE)

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 8 BITS
SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	x	x	MONOSYNC, 6 BITS
SYNC$_{15}$	SYNC$_{14}$	SYNC$_{13}$	SYNC$_{12}$	SYNC$_{11}$	SYNC$_{10}$	SYNC$_9$	SYNC$_8$	BISYNC, 16 BITS
SYNC$_{11}$	SYNC$_{10}$	SYNC$_9$	SYNC$_8$	SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	BISYNC, 12 BITS
0	1	1	1	1	1	1	0	SDLC

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

Figure 25 : SYNC as an Output.

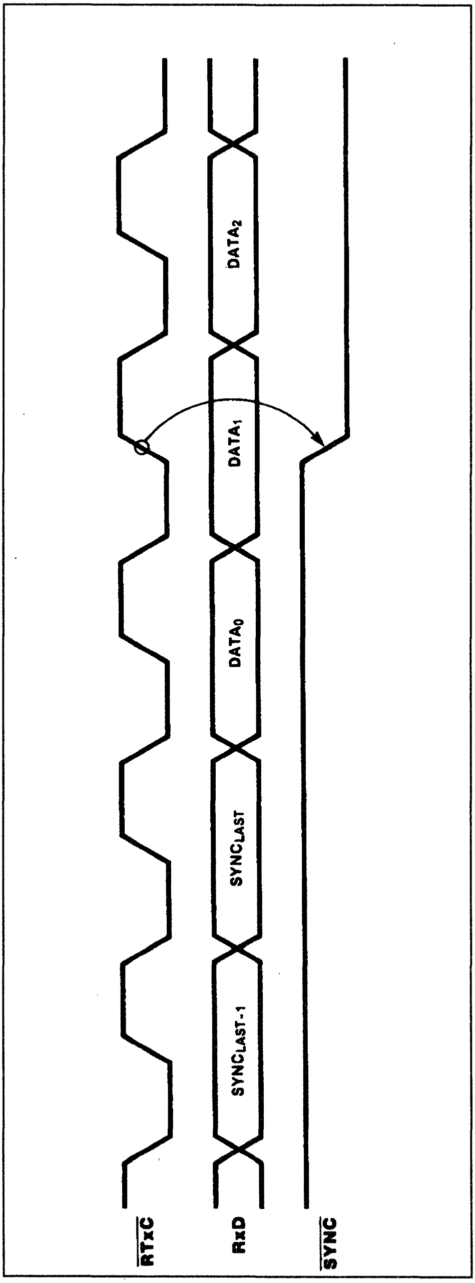
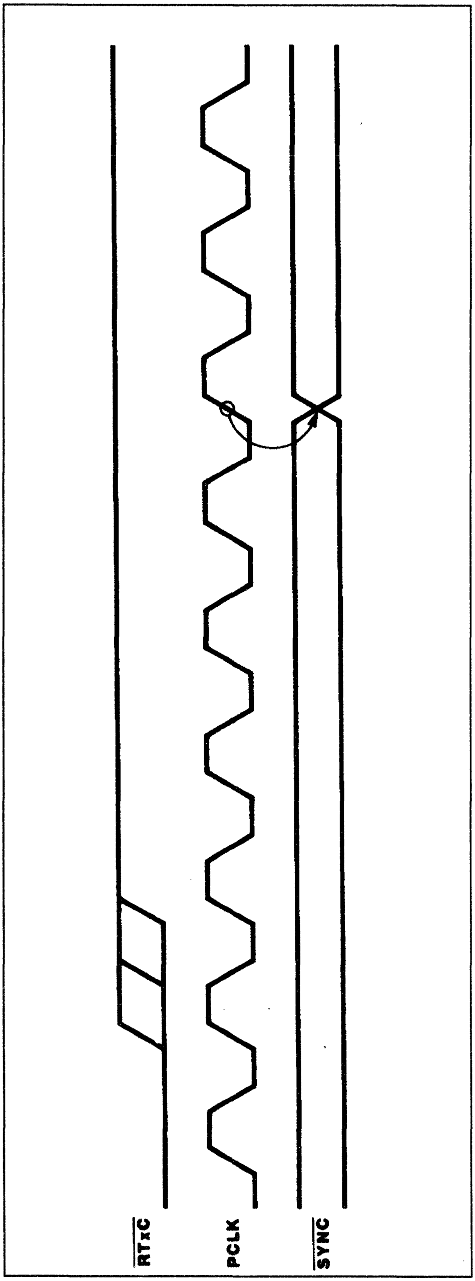


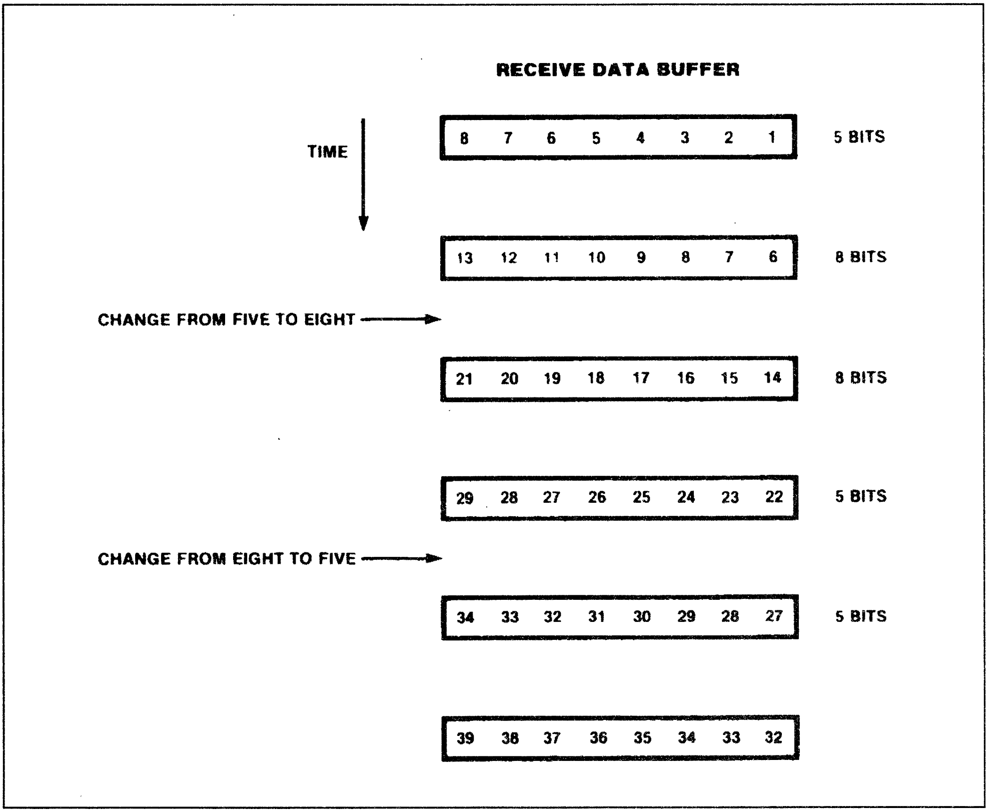
Figure 26 : SYNC as an Input.



PROGRAMMING DATA COMMUNICATION MODES (cont'd)

checker. The logic also guarantees that the calculation will only start or stop on a character boundary by delaying the enable or disable until the next character is loaded into the receive data FIFO. To understand how this works refer to Figure 28 and the following explanation. Consider a case where the SCC receives a sequence of eight bytes, called A, B, C, D, E, F, G and H with A receiver first. Now suppose that A is the sync character, that CRC is to be calculated on B, C, E, and F, and that F is the last byte of this message. Before A is received the receiver is in Hunt mode and the CRC is disabled. When A is in the receive shift register it is compared with the contents of WR7. Since A is the sync character, the bit patterns match and receiver leaves Hunt mode, but character A is not transferred to the receive data FIFO. The CRC remains disabled even though somewhere during the next eight-bit-time processor reads B and enables CRC. At the end of an eight-bit-time, B is in the 8-bit delay and C is in the receive shift register. At this point, B is loaded into the receive data FIFO. The CRC remains disabled even though somewhere during the next eight bit times the processor reads B and enables CRC. At the end of the eight-bit-time, B is in the 8-bit delay and C is in the receive shift register. Character C is loaded into the receive data FIFO and at the same time the CRC checker is enabled. During the next eight-bit-time, the processor reads C and leaves the CRC enabled. At the end of these eight-bit-times the SCC has calculated CRC on B, character C is the 8-bit delay and D is in the Receive Shift register. D is then loaded into the receive data buffer and at some point during the next eight-bit-time the processor reads D and disables CRC. At the end of these eight-bit-times CRC has been calculated on C, character D is in the 8-bit delay and E is in the Receive Shift register.

Figure 27 : Changing Character Length.



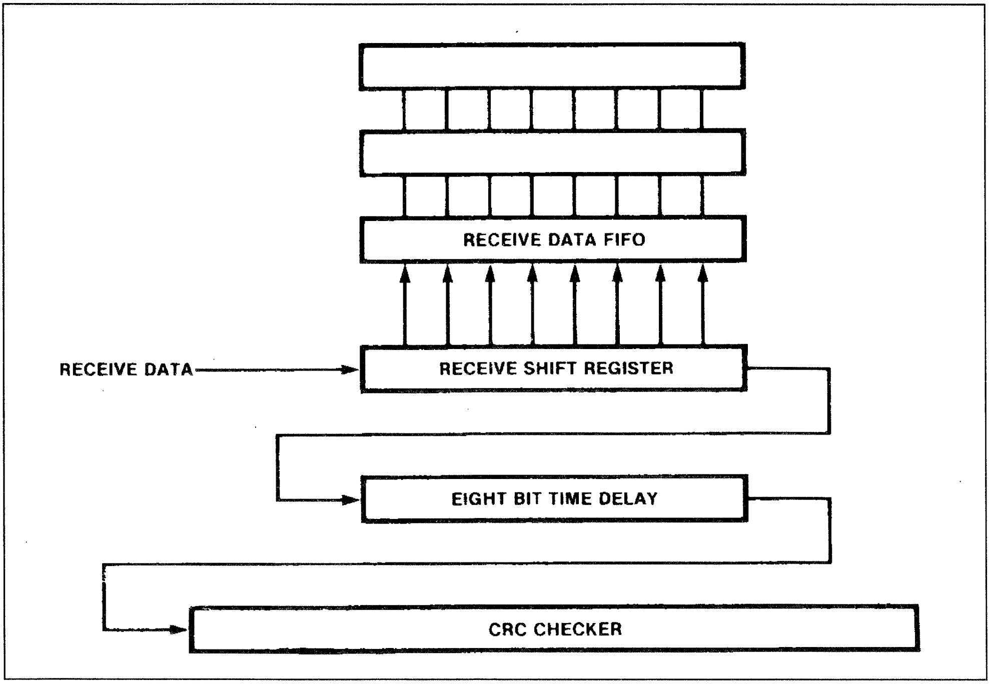
PROGRAMMING DATA COMMUNICATION MODES (cont'd)

Now E is loaded into the receiver data FIFO and, at the same time, the CRC is disabled. During the next eight-bit-times the processor reads E and enables the CRC. During this time E shifts into the 8-bit delay, F enters the Receive Shift register and CRC is not being calculated on D. After these eight-bit-times have elapsed, E is in the 8-bit delay, and F is in the Receive Shift register. Now F is transferred to the receive data FIFO and CRC is enabled. During the next eight-bit-times the processor reads F and leaves the CRC enabled. The processor is usually aware that this is the last character in the message and so prepares to check the result of the CRC computation. However, another sixteen bit-times are required before CRC has been calculated on all of character F. At the end of eight-bit-times F is in the 8-bit delay and G is in the Receive Shift register. At this time G is transferred to the receive data FIFO. Character G must be read and discarded by the processor. Eight bit times later H is transferred to the receive data FIFO also. The result of a CRC calculation is latched in the receive error FIFO at the same time as data is written to the receive data FIFO. Thus the CRC result through character F accompanies character H in the FIFO and will be valid in RR1 until character H is read from the receive data FIFO. The CRC checker may be disabled and reset at any time after character H is transferred to the receive data FIFO. Recall, however, that internally CRC will not be disabed until a character is loaded into the receive data FIFO so the reset command should not be issued until after this occurs. A better alternative is to place the receiver in Hunt mode, which automatically disables and resets the CRC checker.

Up to two modem control signals associated with the receiver are available is synchronous modes. The $\overline{\text{DTR}}/\overline{\text{REQ}}$ pin carries the inverted state of the DTR bit (D7) in WR5 unless this pin has been programmed to carry a DMA Request signal. The $\overline{\text{DCD}}$ pin is ordinarily a simple input to the DCD bit in RR0. However, if the Auto Enables mode is selected by setting D5 of WR3 to "1", this pin becomes an enable for the receiver. That is, if Auto Enables is on and the $\overline{\text{DCD}}$ pin is HIGH the receiver is disabled ; while the $\overline{\text{DCD}}$ pin is LOW the receiver is enabled.

The initialization sequence for the receiver in synchronous modes is WR4 first, to select the mode,

Figure 28 : Receive CRC Data Path.



PROGRAMMING DATA COMMUNICATION MODES (cont'd)

then WR10 to modify it if necessary, WR6 and WR7 to program the sync characters and then WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is complete the receiver is enabled by setting bit D of WR3 to "1".

Synchronous Transmit. Once Synchronous mode has been selected, any of three sync character lengths may be selected. An 8-bit sync character is selected by setting bits D_5 and D_4 or WR4, as well as bit D0 of WR10 to "0". With this option selected the transmitter sends the contents of WR6 when it has no data to send. The 6-bit sync option requires the same programming except that bit D0 of WR10 is set to "1" and only the least significant six bits of WR6 and used as a time fill. For a 16-bit sync character, set bit D_4 of WR4 to "1" and D_5 of WR4 and bit D_0 of WR10 to "0". In this mode the transmitter sends the concatenation of WR6 and WR7 as a time fill. Because the receiver requires that sync characters be left-justified in the registers, while the transmitter requires them to be right-justified, only the receiver will work with a 12-bit sync character. While the receiver is in External Sync mode the transmitter sync length may be six or eight bits, as selected by bit D0 of WR10.

The number of bits per transmitted character is controlled by bits D_6 and D_5 of WR5 and the way the data is formatted within the transmit buffer. The bits in WR5 allow the option of five, six, seven, or eight bits per character. When five bits per character is selected the data may be formatted before being written to the transmit buffer to allow transmisssion of from one to five bits per character. This formatting is shown in Table 5. In all cases the data must be right-justified, with the unused bits being ignored except in the case of five bits per character. An additional bit, carrying parity information, may be automatically appended to every transmitted character by setting bit D_0 of WR4 to "1". This parity bit is sent in addition to the number of bits specified in WR4 or by the data format. If this bit is set to "1", the transmitter will send even parity, if set to "0", the transmitted parity will be odd.

Either of two CRC polynomials may be used in synchronous modes, selected by bit D_2 in WR5. If this bit is set to "1", the CRC-16 polynomial is used and, if this bit is set to "0", the CRC-CCITT polynomial is used. This bit controls the selection for both the transmitter and receiver. The initial state of the generator and checker is controlled by bit D_7 of WR10. When this bit is set to "1", both the generator and checker will have an initial value of all ones, if this bit is set to "0", the initial values will be all zeros. The SCC does not automatically preset the CRC generator, so this must be done in software. This is accomplished by issuing the Reset Tx CRC Generator command, which is encoded in bits D_7 and D_6 of WR0. For proper results this command must be issued while the transmitter is enabled and sending sync characters. If CRC is to be used, the transmit CRC generator must be enabled by setting bit D0 of WR5 to "1". This bit may also be used to exclude certain characters from the CRC calculation. Sync characters are automatically excluded from the CRC calculation and any characters written as data may also be excluded from the calculation by using bit D_0 of WR5. Internally, the CRC is enabled or disabled for a particular character at the same time as the character is loaded from the transmit buffer to the Transmit Shift register. Thus, to exclude a character from CRC calculation bit, D_0 of WR5 should be set to "0" before the character is written to the transmit buffer. This guarantees that the internal disable will occur when the character moves from the buffer to the shift register. Once the buffer becomes empty, the Tx CRC Enable bit may be written for the next character.

Enabling the CRC generator is not sufficient to control the transmission of CRC. In the SCC this function is controlled by the Tx Underrun/EOM bit, which may be reset by the processor and set by the SCC. When the transmitter underruns (both the transmit buffer and Transmit Shift register are empty) the state of the Tx Underrun/EOM bit determines the action taken by the SCC. If the tx Underrun/EOM bit is set when the underrun occurs, the transmitter will send sync characters, if this bit is reset when the underrun occurs, the transmitter will send the accumulated CRC followed by sync characters. When the CRC is loaded into the transmit Shift register for transmission, the SCC will set the Tx Underrun/EOM bit to indicate this. This transition may be programmed to cause an external/status interrupt, or the Tx Underrun/EOM is available in RR0. The Reset Tx Underrun/EOM Latch command is encoded in bits D_7 and D_6 in WR0. For correct transmission of the CRC at the end of a block of data, this command must be issued after the first character is written to the SCC but before the transmitter underruns after the last character written to the SCC. The command is usually issued immediately after the first character is written to the SCC so that CRC will be sent if an underrun occurs inadvertently during the block of data.

In synchronous modes, if the transmitter is disabled during transmission of a character, that character

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

will be sent completely. This applies to both data and sync characters. However, if the transmitter is disabled during the transmission of CRC, the 16-bit transmission will be completed, but the remaining bits will come from the SYNC registers rather than the remainder of the CRC.

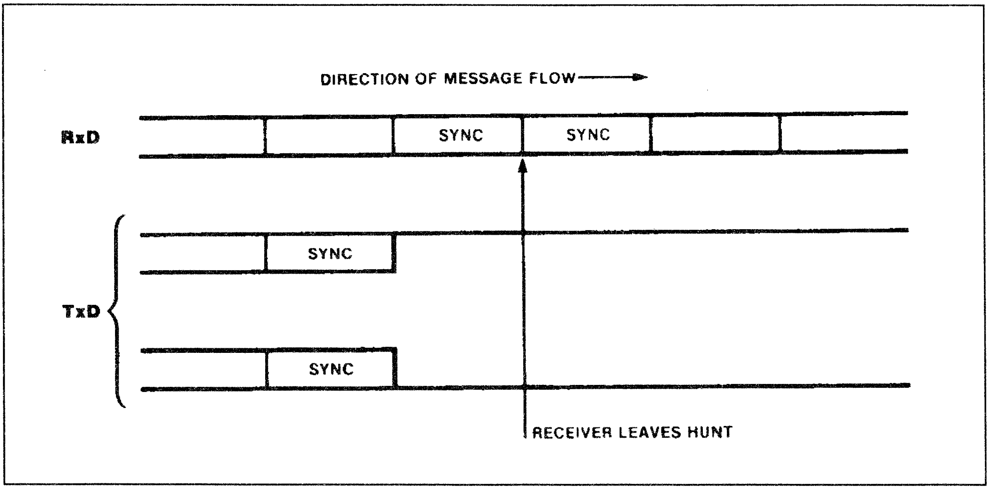
There are two modem control signals associated with the transmitter provided by the SCC. The $\overline{RTS}$ pin is a simple output that carries the inverted state of the RTS bit (D_1) in WR5. The $\overline{CTS}$ pin is ordinarily a simple input to the CTS bit in RR0. However, if Auto Enables mode is selected this pin becomes an enable for the transmitter. That is, if Auto Enables is on and the $\overline{CTS}$ pin is HIGH the transmitter is disabled. While the $\overline{CTS}$ pin is LOW, transmitter is enabled.

The initialization sequence for the transmitter in synchronous modes is : WR4 first, to select the mode, then WR10 to modify it if necessary, WR6 and WR7 to program the sync characters, then WR3 and WR5 to program the sync characters, and then WR3 and WR5 to select the various options. At this point, the other registers should be initialized as necessary. When all of this is complete the transmitter may be enabled by setting bit D_3 or WR5 to "1". Now that the transmitter is enabled the CRC generator may be initialized by issuing the Reset Tx CRC Generator command in WR0.

Transmitter To Receiver Synchronization. The SCC contains a transmitter-to-receiver synchronization function that may be used to guarantee that the character boundaries for the received and transmitted data are the same. In this mode the receiver is in Hunt and the transmitter is idle, sending either all "1s" or "0s". When the receiver recognizes a sync character, it leaves Hunt mode and one character time later the transmitter is enabled and begins sending sync characters. Beyond this point the receiver and transmitter are again completely independent, except that the character boudaries are now aligned. This is shown in Figure 29. There are several restrictions on the use of this feature in the SCC. First, it will only work with 6-bit, 8-bit or 16-bit sync characters, and the data character length for both the receiver and the transmitter must be six bits with a 6-bit sync character or eight bits with an 8-bit or 16-bit sync character. Of course, the receive and transmit clocks must have the same rate as well as the proper phase relation-ship.

A specific sequence of operations must be followed to synchronize the transmitter to the receiver. Both the receiver and transmitter must have been initialized for operation in Synchronous mode sometime in the past, although this initialization need not be redone each time the transmitter is synchronized to the receiver. The transmitter is disabled by setting bit D_3 of WR5 to "0". At this point the transmitter will send continuous "1s". If it is desired that continuous "1s". If it is desired that continuous "0s" be transmitted, the Send Break bit (D_4) in WR5 should be set to "1". The transmitter is now idling but must still be placed in the transmitter to receiver synchronization mode. This is accomplished by setting the Loop

Figure 29 : Transmitter to Receiver Synchronization.



PROGRAMMING DATA COMMUNICATION MODES (cont'd)

the processor should set the Go Active on Poll bit (D_4) in WR10. The final step is to force the receiver to search for sync characters. If the receiver is currently disabled the receiver will enter Hunt mode when it is enabled by setting bit D_0 of WR3 to "1". If the receiver is already enabled it may be placed in Hunt mode by setting bit D_4 of WR3 to "1". Once the receiver leaves Hunt mode the transmitter is activate on the following character boundary.

SDLC Mode

SDLC mode is useful in bit-oriented protocols. That is, protocols which use the technique of "0" insertion to achieve data transparency. In SDLC mode, frames of information are opened and closed by a unique bit pattern called a flag. The Flag character has a bit pattern of "01111110" and this sequence is unique because all data between the opening and closing flags is prohibited from having more than five consecutive "1s". The transmitter guarantees this by watching the transmit data stream and inserting a "0" after five consecutive ones, irrespective of character boundaries. In turn, the receiver searches the receive data stream for five consecutive "1s" and deletes the next bit if it is a "0". CRC may be used in SDLC mode but only with the CRC-CCITT polynomial, because the transmitter in the SCC automatically inverts the CRC before transmission, and the receiver - to compensate for this - checks the CRC result for the bit pattern "0001110100001111". This is consistent with bit-oriented protocols such as SDLC, HDLC, and ADCCP. There are two unique bit patterns in SDLC mode besides the flag sequence. They are the Abort and EOP (End of Poll) sequence. An Abort is a sequence of from seven to thirteen consecutive "1s" and is used to signal the premature termination of a frame. The EOP is the bit pattern "11111110", which is used in loop applications as a signal to a secondary station that it may begin transmission.

SDLC mode is selected by setting bit D_5 of WR4 to "1" and bits D_4, D_3, and D_2 of WR4 to "0". In addition, the flag sequence must be written to WR7. Additional control bits for SDLC mode are located in WR10.

SDLC Receive. The receiver in the SCC always searches the receive data stream for flag characters in SDLC mode. Ordinarily, the receiver transfers all received data between flags to the receive data FIFO. However, if the receiver is in Hunt mode no flag is received. The receiver is in Hunt mode when first enabled, or the receiver may be placed in Hunt mode by the processor issuing the Enter Hunt mode command in WR3. This bit (D_4) is a command, and writing a "0" to it has no effect. The Hunt status of the receiver is reported by the Sync/Hunt bit in RR0. Sync/Hunt is one of the possible sources of external/status interrupts, with both transitions causing an interrupt. This is true even if the Sync/Hunt bit is set as a result of the processor issuing the Enter Hunt mode command. The receiver will automatically enter Hunt mode if an abort is received. Because the receiver always searches the receive data stream for flags and automatically enter Hunt Mode when an abort is received, the receiver will always handle frames correctly, and the Enter Hunt Mode command should never to needed. The SCC will drive the $\overline{\text{SYNC}}$ pin LOW to signal that a flag has been recognized. The timing for the $\overline{\text{SYNC}}$ signal is shown in Figure 30.

The first byte in an SDLC frame is assumed by the SCC to be the address of the secondary station for which the frame is intended. The SCC provides several options for handling this address. If the Address Search Mode bit (D_2) in WR3 is set to "0", the address recognition logic is disabled and all received frames are transferred to the receive data FIFO. In this mode the software must perform any address recognition. If the Address Search Mode bit is

Table 6 : Residue Codes.

Residue Code			Bits in Previous Byte				Bits in Second Previous Byte				Bits in Third Previous Byte			
2	1	0	8 B/C	7 B/C	6 B/C	5 B/C	8 B/C	7 B/C	6 B/C	5 B/C	8 B/C	7 B/C	6 B/C	5 B/C
1	0	0	0	0	0	0	3	1	0	0	8	7	5	2
0	1	0	0	0	0	0	4	2	0	0	8	7	6	3
1	1	0	0	0	0	0	5	3	1	0	8	7	6	4
0	0	1	0	0	0	0	6	4	2	0	8	7	6	5
1	0	1	0	0	0	0	7	5	3	1	8	7	6	5
0	1	1	0	0	0	-	8	6	4	-	8	7	6	-
1	1	1	1	0	-	-	8	7	-	-	8	7	-	-
0	0	0	2	-	-	-	8	-	-	-	8	-	-	-

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

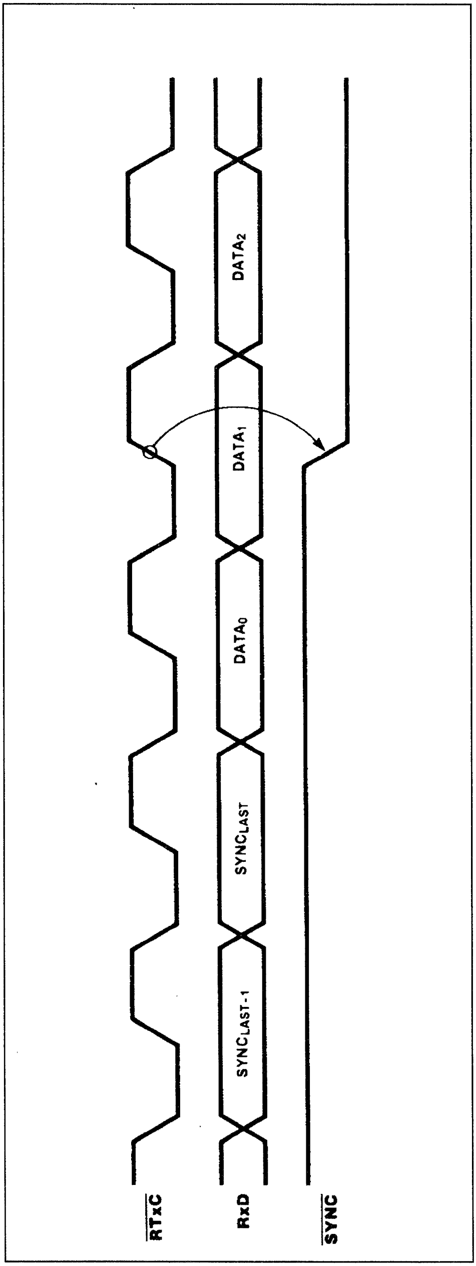
set to "1", only those frames whose address matches the address programmed in WR6 or the global address (all "1s") will be transferred to the receive data FIFO. The address comparison will be across all eight bits of WR6 if the Sync Character Load Inhibit bit (D_1) in WR3 is set to "0". The comparison may be modified so that only the four most significant bits of WR6 must match the received address. This mode is selected by setting the Sync Character Load Inhibit bit to "1". In this mode, however, the address field is still eight bits wide. The address field is transferred to the receive data FIFO in the same manner as data. It is not treated differently than data.

The number of bits per character is controlled by bits D_7 and D_6 of WR3. Five, six, seven, or eight bits per character may be selected via these two bits. The data is right-justified in the receive buffer. The SCC merely takes a snapshot of the receive data stream at the appropriate times, so the "unused" bits in the receive buffer are only the bits following the character in the data stream. An additional bit carrying parity information may be selected by setting bit D_6 of WR4 to "1". This also enables parity in the transmitter. The parity sense is selected by bit D_1 of WR4. Parity is not normally used in SDLC mode. The character length may be changed at any time before the new number of bits have been assembled by the receiver. Care should be exercised, however, as unexpected results may occur. A representative example, switching from five bits to eight bits and back to five bits is shown in Figure 31.

Most bit-oriented protocols allow an arbitrary number of bits between opening and closing Flags. The SCC allows for this by providing three bits of Residue Code in RR1 that indicate which bits in the last few bytes transferred from the receive data FIFO by the processor are actually valid data bits. The meaning of these three bits with each character length option is shown in Table 6. As indicated in the table, these bits allow the processor to determine those bits in the information (and not CRC) field. This allows transparent retransmission of the received frame. The Residue Code bits do not go through a FIFO so they change in RR1 when the last character of the frame is loaded into the receive data FIFO. If there are any characters already in the receive data FIFO the Residue Code will be updated before they are read by the processor. Thus these three bits off RR1 should be ignored by the processor unless the End of Frame bit in RR1 is set.

Only the CRC-CCITT polynomial may be used for CRC calculation in SDLC mode, although the gene-

Figure 30 : SYNC as an Output.



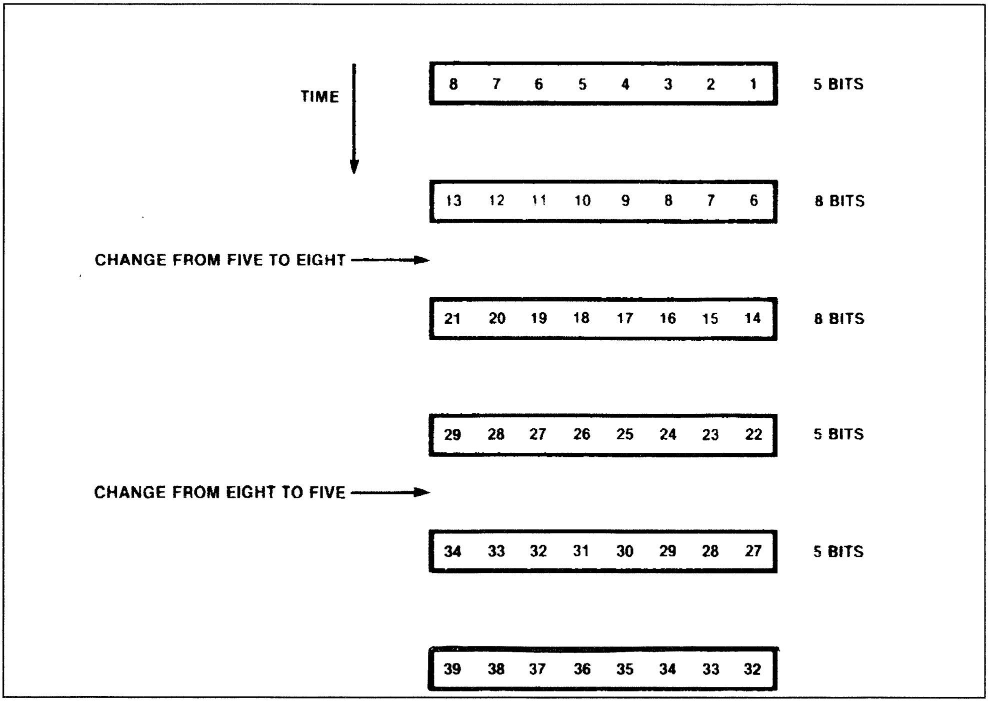
PROGRAMMING DATA COMMUNICATION MODES (cont'd)

rator and checker may be preset to all "1s" or all "0s". The CRC-CCITT polynomial is selected by setting D_2 of WR5 to "0", bit D_7 of WR10 controls the preset value. If this bit is set to "1", the generator and checker are preset to "1s", if this bit is reset, the generator and checker are preset to all "0s". The receiver expects the CRC to be inverted before transmission and so checks the CRC result against the value "0001110100001111". The SCC presets the CRC checker whenever the receiver is in Hunt mode or whenever a flag is received so a CRC reset command is not strictly necessary. However, the CRC checker may be preset by issuing the Reset CRC Checker command in WRO. The CRC checker is automatically enabled for all data between the opening and closing flags by the SCC in SDLC mode, and the Rx CRC Enable bit (D_3) in WR3 is ignored. The result of the CRC calculation for the entire frame is valid in RR1 only when accompagnied by the end of Frame bit being set in RR1. At all other times the CRC Error bit in RR1 should be ignored by the processor. Care must be exercised so that the processor does not attempt to use the CRC bytes that are transferred as data because not all of the bits are transferred properly. The last two bits of CRC are never transferred to the receive data FIFO and are not recoverable.

A frame is terminated by a closing flag. When the SCC recognizes this flag the contents of the Receive Shift register are transferred to the receive data FIFO, the Residue Code is latched, the CRC Error bit is latched in the status FIFO and the End Of Frame bit is set in the receive status FIFO. The End Of Frame bit, upon reaching the top of the FIFO, will cause a special receive condition. The processor may then read RR1 to determine the result of the CRC calculation as well as the Residue Code. If either the Rx Interrupt on Special Condition Only or the Rx Interrupt on First Character or Special Condition mode are selected, the processor must issue and Error Reset command in WR0 to unlock the receive FIFO.

In addition to searching the data stream for flags, the receiver in the SCC also watches for seven consecutive "1s", which is the abort condition. The

Figure 31 : Changing Character Length.



PROGRAMMING DATA COMMUNICATION MODES (cont'd)

presence of seven consecutive "1s" is reported in the Break/Abort bit in RR0. This is one of the possible external/status interrupts, so transitions of this status may be programmed to cause interrupts. Upon receipt of an abort the receiver is forced into Hunt mode, where it looks for flags. The Hunt status is also a possible external/status condition whose transition may be programmed to cause an interrupt. The transitions of these two bits occur very close together but either one or two external/status interrupts may result. The abort condition is terminated when a "0" is received, either by itself or as the leading "0" of a flag. The receiver does not leave Hunt mode until a flag has been received so two discrete external/status conditions will occur at the end of an abort. An abort received in the middle of a frame terminates the frame reception, but no in an orderly manner, because the character being assembled is lost.

Up to two modem control signals associated with the receiver are available in SDLC mode. The DTR/ $\overline{\text{REQ}}$ pin carries the inverted state of the DTR bit (D_7) in WR5 unless this pin has been programmed to carry a DMA Request signal. The DCD pin is ordinarily a simple input to the DCD bit in RR0. However, if the Auto Enables mode is selected by setting bit D5 of WR3 to "1", this pin becomes an enable for the receiver. That is, if Auto Enable is on and the $\overline{\text{DCD}}$ pin is HIGH the receiver is disabled. While the $\overline{\text{DCD}}$ pin is LOW, the receiver is enabled.

The initialization sequence for the receiver in SDLC mode is : WR4 first, to select the mode, then WR10 to modify it if necessary, WR6 to program the address, WR7 to program the flag and the WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is complete the receiver may be enabled by setting bit D_6 of WR3 to "1".

SDLC Transmit. Once SDLC mode has been selected, the flag must be written in WR7, to be used to open and close the transmitted frames. The SCC does not automatically send the address byte ; it merely encapsulates the data supplied by the processor with flags and CRC. Ordinarily, a frame will be terminated by the SCC with CRC and a flag but the SCC may be programmed to send an abort and a flag in place of the CRC. This option allows the SCC to abort a frame transmission in progress if the transmitter is accidentally allowed to underrun. This is controlled by the Abort/ $\overline{\text{Flag}}$ On Underrun bit (D_2) in WR10. When this bit is set to "1" the transmitter will send an abort and a flag in place of the CRC when an underrun occurs. The frame will be terminated normally, with CRC and a flag, if this bit is set to "0". The SCC is also able to send an abort by command of the processor. The Send Abort command, issued in WR0, will send eight consecutive "1s" and then the transmitter will idle. Since up to five consecutive "1s" may have been sent prior to the command being issued, a Send Abort will cause a sequence of from eight to thirteen "1s" to be transmitted. The Send Abort command also empties the transmit buffer register. The idle condition for the transmitter is continuous flags, but this is under program control. By setting the Mark/ $\overline{\text{Flag}}$ Idle bit (D_3) in WR10 to "1", the transmitter will send continuous "1s" in place of the idle flags. Note that the closing flag will be transmitted correctly even if this mode is selected. The Mark/ $\overline{\text{Flag}}$ Idle must be set to "0", allowing a flag to be transmitted, before data is written to the transmit buffer. Care must be exercised in doing this because the continuous "1s" are transmitted eight at a time, and all eight must leave the Transmit Shift register, so that flag may be loaded into it before the first data is written to the transmit buffer. When using the transmitter in SDLC mode, recall that all data passes through the zero inserter, which adds an extra five bit times of delay between the Transmit Shift register and the Transmit Data pin.

The number of bits per transmitted character is controlled by bits D_6 and D_5 of WR5 and the way the data is formatted within the transmit buffer. The bits in WR5 allow the option of five, six, seven, or eight bits per character. When five bits per character is selected, the data may be formatted before being written to the transmit buffer, to allow transmission of one to five bits per character. This formatting is shown in Table 4-1. In all cases the data must be right-justified, with the unused bits being ignored, except in the case of five bits per character.

An additional bit, carrying parity information, may be automatically appended to every transmitted character by setting bit D_6 of WR4 to "1". This bit is sent in addition to the number of bits specified in WR4 or by the data format. The parity sense is selected by bit D_1 of WR4. Parity is not normally used in SDLC mode. The character length may be changed on the fly, but the desired length must be selected before the character is loaded into the transmit shift register from the transmit buffer. The easiest way to ensure this is to write to WR5 to change the character length before writing the data to the transmit buffer.

Only the CRC-CCITT polynomial may be used in SDLC mode. This is selected by setting bit D_2 in WR5 to "0". This bit controls the selection for both

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

the transmitter and receiver. The initial state of the generator and checker is controlled by bit D_7 of WR10. When this bit is set to "1", both the generator, and checker sill have an initial value of all "1s" and, if this bit is set to "0", the initial values will be all "0s". The SCC does not automatically preset the CRC generator so this must be done in software. This is accomplished by issuing the Reset Tx CRC generator command, which is encoded in bits D_7 and D_6 of WR0. For proper results, this command must be issued while the transmitter is enabled and idling. If CRC is to be used the transmit CRC generator must be enabled by setting bit D0 of WR5 to "1". CRC is normally calculated on all characters between opening and closing flags, so this bit is usually set to "1" at initialization and never changed.

Enabling the CRC generator is not sufficient to control the transmission of CRC. In the SCC this function is controlled by the Tx Underrun/ EOM bit, which may be reset by the processor and set by the SCC. When the transmitter underruns (both the transmit buffer and transmit shift register are empty) the state of the Tx Underrun EOM bit determines the action taken by the SCC. If the Tx Underrun/EOM bit is set to "1" when the underrun occurs, the transmitter will send flags ; if this bit is reset to "0" when the underrun occurs, the transmitter will send either the accumulated CRC followed by flags, or an abort followed by flags, depending the state of the Abort/ $\overline{\text{Flag}}$ on Underrun bit in the WR10. When the CRC or abort is loaded into the Transmit Shift register for transmission, the SCC will set the Tx Underrun/EOM bit to indicate this. This transition may be programmed to cause an external/status interrupt, or the Tx Underrun/ EOM bit is available in RR0. The Reset Tx Underrun/EOM Latch command is encoded in bits D_7 and D_6 of WR0.

For correct transmission of the CRC at the end of a frame, this command must be issued after the first character is written to the SCC but before the transmitter underruns after the last character written the SCC. The command is usually issued immediately after the first character is written to the SCC so that the abort or CRC is sent if an underrun occurs inadvertently. The Abort/ $\overline{\text{Flag}}$ on Underrun bit (D2) in WR10 is usually set to "1" at the same time as the Tx Underrun/EOM bit is reset so that an abort sill be sent if the transmitter underruns. The bit is then set to "0" near the end of the frame to allow the correct transmission of CRC.

In this paragraph the term "completely sent" means shifted out of the Transmit Shift register, not shifted out of the zero inserter, which is an additional five bit times of delay. In SDLC mode, if the transmitter is disabled during transmission of a character, that character will be "completely sent". This applies to both data and flags. However, if the transmitter is disabled during the transmission of CRC, the 16-bit transmission will be completed but the remaining bits will be from the Flag register rather than the remainder of the CRC.

There are two modem control signals associated with the transmitter provided by the SCC. The $\overline{\text{RTS}}$ pin is a simple output that carries the inverted state of the RTS bit (D_1) in WR5. The $\overline{\text{CTS}}$ pin is ordinarily a simple input to the $\overline{\text{CTS}}$ bit in RR0. However, if Auto Enables mode is selected, this pin becomes an enable for the transmitter. That is, if Auto Enables is on and the $\overline{\text{CTS}}$ pin is HIGH the transmitter is disabled. If the $\overline{\text{CTS}}$ pin is LOW, the transmitter is enabled.

The initialization sequence for the transmitter in SDLC mode is : WR4 first, to select the mode, then WR10 to modify it if necessary, WR7 to program the flag, and then WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is complete, the transmitter may be enabled by setting bit D_3 of WR5 to "1". Now that the transmittter is enabled, the CRC generator may be initialized by issuing the Reset Tx CRC Generator command in WR0.

SDLC Loop Mode. SDLC Loop mode is quite similar to SDLC mode except that two additional control bits are used. They are the Loop Mode bit (D_1) and the Go Active on Poll bit (D_4) in WR10. In addition to these two extra control bits, there are also two status bits in RR10. They are the On Loop bit (D_1) and the Loop Sending bit (D_4). Before Loop mode is selected both the receiver and transmitter must be completely initialized for SDLC operation. Once this is done, Loop mode is selected by setting bit D_1 of WR10 to "1". At this point the SCC connects TxD to RxD with only gate delays in the path. At the same time a flag is loaded into the Transmit Shift register, and is shifted to the end of the zero inserter, ready for transmission. The SCC will remain in this state until the Go Active on Poll bit (D4) in WR10 is set to "1". When this bit is set to "1" the receiver begins looking for a sequence of seven consecutive "1s", indicating either an EOP or an idle line. When the receiver detects this condition the Break/Abort bit in RR0 is set to "1" and a one-bit time delay is inserted in the path from RxD to TxD. The On Loop bit in RR10 is also set to "1" at this time, and the receiver enters the Hunt mode. The SCC cannot transmit on the loop until a flag is received, causing the receiver

PROGRAMMING DATA COMMUNICATION MODES (cont'd)

to leave Hunt mode, and another EOP (bit pattern " 11111110") is received. The SCC is now on the loop and capable of transmitting on the loop. As soon as this status is recognized by the processor, the Go Active On Poll bit in WR10 should be set to "0" to prevent the SCC from transmitting on the loop without the consent of the processor.

To transmit a message on the loop, the Go Active On Poll bit in WR10 must be set to "1". Once this is done, the SCC will change the next received EOP into Flag and begin transmitting on the loop. When the EOP is received, the Break/Abort and Hunt bits in RRO will be set to "1", and the Loop Sending bit in RR10 will also be set to "1". Data to be transmitted may be written after the Go Active On Poll bit has been set or after the receiver enter Hunt mode. If the data is written immediately after the Go Active On Poll bit has been set, the SCC will only insert one flag after the EOP is changed into a flag. If the data is not written until after the receiver enters the Hunt mode, flags will be transmitted until the data is written. If only one frame is to be transmitted on the loop in response to an EOP, the processor must set the Go Active on Poll bit to "0" before the last data is written to the transmitter. In this case the transmitter will close the frame with a single flag, and then revert to the one-bit delay. The Loop Sending bit in RR10 is set to "0" when the closing Flag has been sent. If more than one frame is to be transmitted, the Go Active On Poll bit should not be set to "0" until the last frame is being sent. If this bit is not set to "0" before the end of a frame, the transmitter will send Flags until either more data is written to the transmitter, or until the Go Active On Poll bit is set to "0". Note that the state of the Abort/ $\overline{\text{Flag}}$ on Underrun and Mark/ $\overline{\text{Flag}}$ Idle bits in WR10 are ignored by the SCC in SDLC Loop mode.

To go off the loop in an orderly manner requires actions similar to those taken to go the loop. First, the Go Active On Poll bit must be set to "0" and any transmission in progress completed, if the SCC is currently sending on the loop. Once the SCC is not sending on the loop, an exit from the loop is accomplished by setting the Loop Mode bit in WR10 to "0", and at the same time writing the Abort/ Flag on Underrun and Mark/$\overline{\text{Flag}}$ Idle bits with the desired values. The SCC will revert to normal SDLC operation as soon as an EOP is received, or immediately, if the receiver is already in Hunt mode because of the receipt of an EOP.

The initialization sequence for the SCC in SDLC Loop mode is similar to the sequence used in SDLC mode, except that it is somewhat longer. The processor should program WR4 first, to select SDLC mode, and then WR10 to select the CRC preset value and program the Mark/Flag Idle bit. The Loop Mode and Go Active On Poll bits in WR10 should not be set to"1" yet. The flag is written in WR7 and the various options are selected in WR3 and WR5. At this point the other registers should be initialized as necessary, then Loop Mode bit (D_1) in WR10 should be set to "1". When all of this is complete the transmitter may be enabled by setting bit D_3 of WR5 to "1". Now that the transmitter is enabled, the CRC generator may be initialized by issuing the Reset TxCRC Generator command in WR0. The receiver is enabled by setting the Go Active on Poll bit (D_4) in WR10 to "1". The SCC will go on the loop when seven consecutive "1s" are received, and will signal this by setting the On Loop bit in RR10. Note that the seven consecutive "1s" will set the Break/Abort and Hunt bits in RR0 also. Once the SCC is on the loop, the Go Active on Poll bit should be set to "0" until a message is to be transmitted on the loop. To transmit a message on the loop, the Go Active on Poll bit should be set to "1". At this point the processor may either write the first character to the transmit buffer and wait for a transmit buffer empty condition, or wait for the Break/Abort and Hunt bits to be set in RR10 and the Loop Sending bit to be set in RR10 before writing the first data to the transmitter. The Go Active On Poll bit should be set to "0" after the transmission of the frame has begun. To go off of the loop, the processor should set the Go Active On Poll bit in WR10 to "0" and then wait for the Loop Sending bit in RR10 to be set to "0". At this point the Loop Mode bit (D_1) in WR10 is set to "0" to request an orderly exit from the loop. The SCC will exit SDLC Loop mode when seven consecutive "1s" have been received ; at the same time the Break/Abort and Hunt bits in RR0 will be set to "1", and the On Loop bit in RR10 will be set to "0".

SUPPORT CIRCUITRY PROGRAMMING

The SCC incorporates additional circuitry to aid in serial communications. This circuitry includes clocking options, baud rate generator, data encoding, and internal loopback. This section discusses how to program these functions.

Clock Options

The SCC may be programmed to select one of several sources to provide the transmit and receive clocks. In addition, the SCC requires a fundamental, parallel resonant crystal oscillator in each channel, as well as the ability to echo one of several internal clock sources to the outside world. These options are controlled by the bits in WR11.

The crystal oscillator option is controlled by bit D_7 in WR11. When this is set to "0", the crystal oscillator is disabled and all pins function normally. When this bit is set to "1" the crystal oscillator is enabled and a high-gain amplifier is connected between the RTxC pin and the $\overline{\text{SYNC}}$ pin. While the crystal oscillator is enabled, anything that has $\overline{\text{RTxC}}$ selected as its clock source will automatically be connected to the output of the crystal oscillator. While the crystal oscillator is enabled, the $\overline{\text{SYNC}}$ pin is obviously unavailable for other use. In synchronous modes no sync pulse is output, and the External Sync mode cannot be selected. In asynchronous modes the state of the Sync/ Hunt bit in RR0 is no longer controlled by the $\overline{\text{SYNC}}$ pin. Instead, the Sync/ Hunt bit is forced to "0". The crystal oscillator requires some finite time to stabilize. The oscillator must be allowed to stabilize before it is used as a clock source.

The source of the receive clock is controlled by bits D_6 and D_5 of WR11. The receive clock may be programmed to come from the $\overline{\text{RTxC}}$ pin , the $\overline{\text{TRxC}}$ pin, the output of the baud rate generator, or the transmit output of the DPLL.

The source of the transmit clock is controlled by bits D_4 and D_3 of WR11. The transmit clock may be programmed to come from the $\overline{\text{RTxC}}$ pin, the $\overline{\text{TRxC}}$ pin, the output of the baud rate generator, or the transmit output of the DPLL.

Ordinarily the TRxC pin is an input, but it becomes on output if this pin has not been selected as the source for the transmitter or the receiver, and bit D_2 of WR11 is set to "1". The selection of the signal provided on the $\overline{\text{TRxC}}$ ouput pin is controlled by bits D_1 and D0 of WR11. The $\overline{\text{TRxC}}$ pin may be programmed to provide the output of the crystal oscillator, the output of the baud rate generator, the receive output of the DPLL or the actual transmit clock. If the output of the crystal oscillator is selected but the crystal oscillator has not been enabled the $\overline{\text{TRxC}}$ pin will be driven HIGH. The option of placing the transmit colck signal on the $\overline{\text{TRxC}}$ pin when it is an output allows access to the transmit output of the DPLL.

Figure 32 shows a simplified schematic diagram of the circuitry used in the clock multiplexing. It shows the inputs to the multiplexer section as well as the various signal inversions that occur in the paths to the outputs. Also shown are the edges used by the receiver, transmitter, baud rate generator and DPLL to sample or send data or otherwise change state. For example, the receiver samples data on the falling edge, but since there is an inversion in the clock path between the $\overline{\text{RTxC}}$ pin and the receiver, a rising edge of the $\overline{\text{RTxC}}$ pin samples the data for the receiver.

Selection of the clocking options may be done anywhere in the initialization sequence, but the final values must be selected before the receiver, transmitter, baud rate generator, or DPLL are enabled to prevent problems from arbitrarily narrow clock signals out of the multiplexers. The same is true of the crystal oscillator, in that the output should be allowed to stabilize before it is used as a clock source.

Baud Rate Generator

Figure 33 shows a block diagram of the baud rate generator. It consists of a 16-bit down-counter, two 8-bit time constant registers and an output divide-by-two. The baud rate generator input comes from the output of a two-input multiplexer, the zero count condition is output to the External/ Status Interrupt Section. The baud rate generator may be enabled and disabled by command and is disabled by a hardware reset.

The time constant for the baud rate generator is programmed in WR12 and WR13, with the least-significant byte in WR12. The formulas relating the baud rate to the time constant and vice versa are shown with an example. In these formulas the baud rate generator clock frequency is in Hertz, the desired baud rate in bits/ second and the time constant is dimensionless. The example in Table 7 assumes a 2.4576 MHz clock factor of 16 and shows the time constant for a number of popular baud rates.

SUPPORT CIRCUITRY PROGRAMMING (cont'd)

Time Constant Formulas :

$$\text{Time Constant} = \frac{\text{Clock Frequency}}{2 \bullet (\text{Clock Mode}) \cdot (\text{Baud Rate})} - 2$$

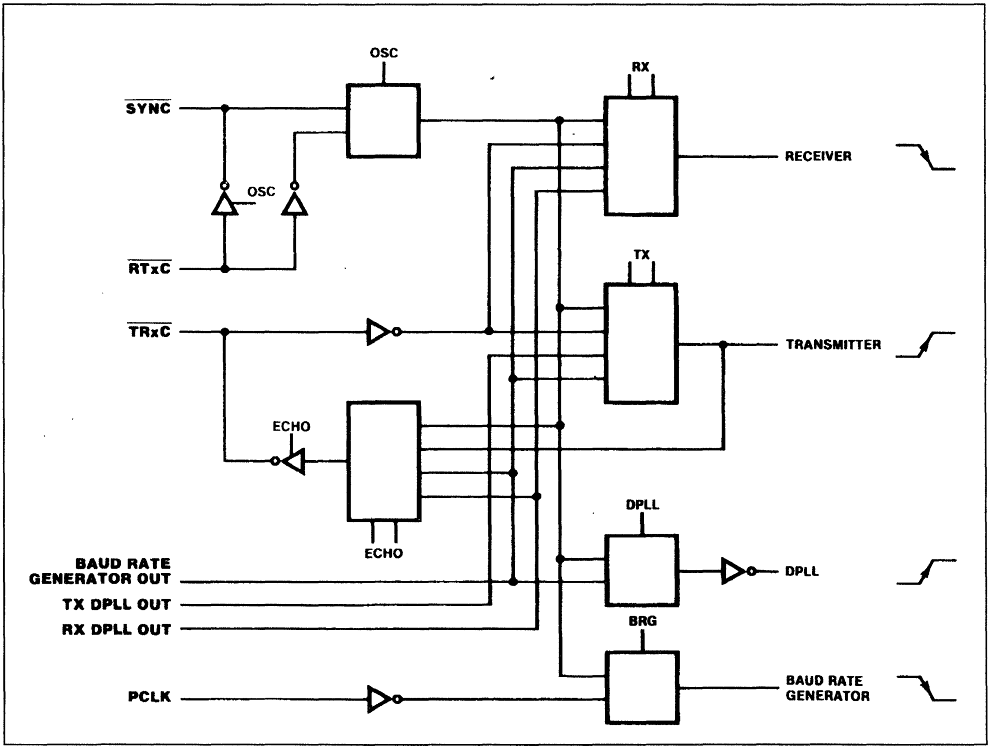
$$\text{Baud Rate} = \frac{\text{Clock Frequency}}{2 \bullet (\text{Clock Mode}) \cdot (\text{Time Const.} + 2)}$$

The clock source for the baud rate generator is selected by bit D_1 of WR14. When this bit is set to "0" the baud rate generator uses the signal on the TxC

Table 7 : Baud Rate Example.

Baud Rate	Divider	
	Decimal	Hex
38400	0	0000H
19200	2	0002H
9600	6	0006H
4800	14	000EH
2400	30	001EH
1200	62	003EH
600	126	007EH
300	254	00FEH
150	510	01FEH
For 2.4576 MHz Clock, X 16 Mode		

Figure 32 : Clock Multiplexer.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

pin as its clock, independent of whether the $\overline{TxC}$ pin is a simple input or part of the crystal oscillator circuit. When this bit is set to "1" the baud rate generator is clocked by PCLK. To avoid metastable problems in the counter, this bit should be changed only while the baud rate generator is disabled, since arbitrarily narrow pulses can be generated at the output of the multiplexer when it changes status. The baud rate generator is enabled while bit D_0 of WR14 is set to "1" and is disabled while this bit is

Figure 33 : Baud Rate Generator.

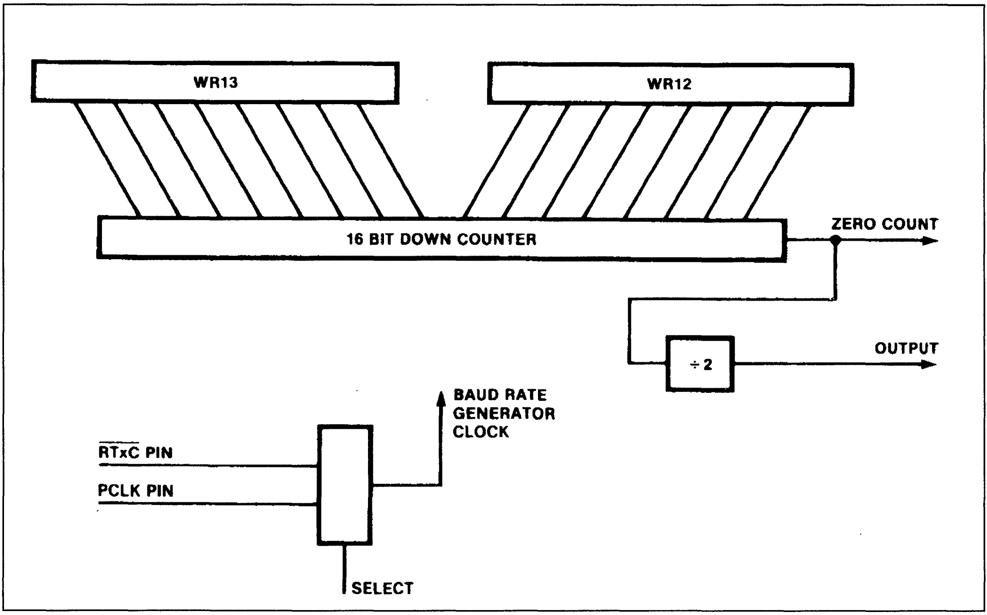
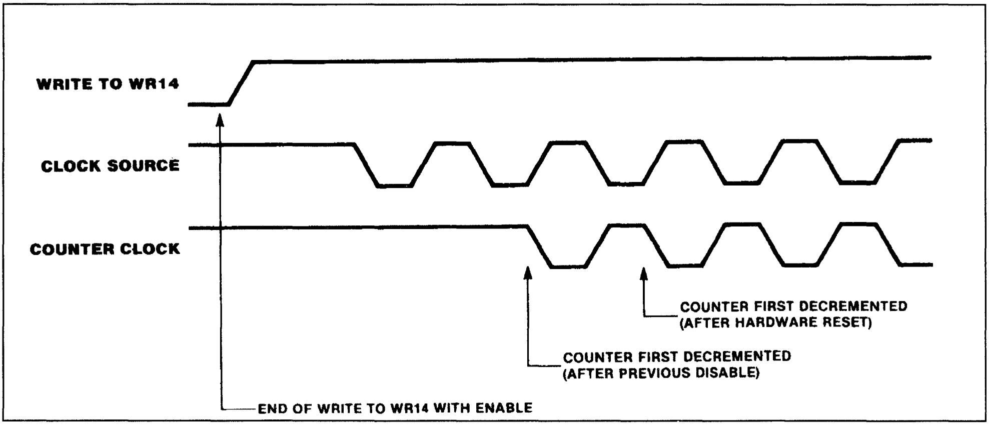


Figure 34 : Baud Rate Generator Start-Up.


WRITE TO WR14
CLOCK SOURCE
COUNTER CLOCK
COUNTER FIRST DECREMENTED (AFTER HARDWARE RESET)
COUNTER FIRST DECREMENTED (AFTER PREVIOUS DISABLE)
END OF WRITE TO WR14 WITH ENABLE

SUPPORT CIRCUITRY PROGRAMMING (cont'd)

set to "0". To prevent metastable problems when the baud rate generator is first enabled, the enable bit is synchronized to the baud rate ge-nerator clock. This introduces an additional delay then the baud rate generator is first enabled and this is shown in Figure 5-3. The baud rate generator is disabled immediately when bit D_0 of WR14 is set to "0", because the delay is only necessary on startup. The baud rate generator may be enabled and disabled on the fly, but this delay on startup must be taken into consideration.

Upon reaching a count of "0" the time constant held in WR12 and WR13 is reloaded into the downcounter so that the process of counting down may start over. In addition to reloading the time constant, the output of the baud rate generator toggles, and for the clock cycle with a zero count, the zero count signal goes active to the External/Status Section. This zero count condition from the baud rate generator does not persist, so if it is to be used by the processor, it should be latched in the External/Status latch. While the baud rate generator is disabled the state of the zero count signal is held. This signal is forced active by a hardware reset.

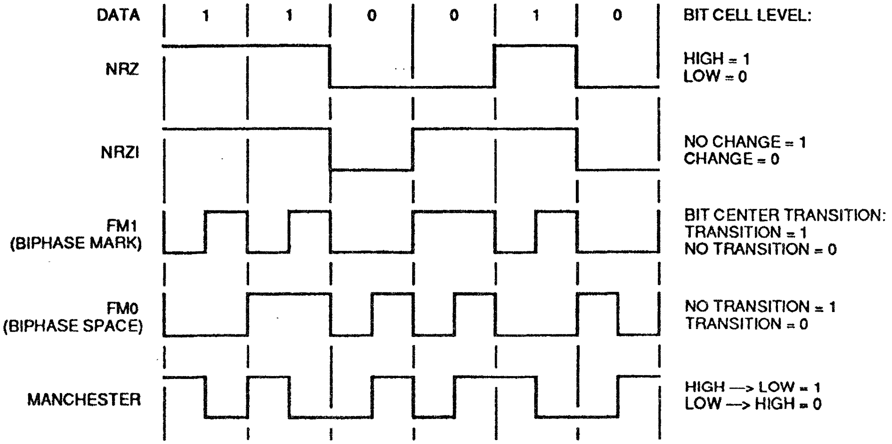
Initializing the baud rate generator is done in four steps. First, the time constant is determined and loaded into WR12 and WR13. Next, the processor must select the clock source for the baud rate generator by writing to bit D_1 of WR14. Finally, the baud rate generator is enabled by setting bit D_0 of WR14 to "1". Note that the first write to WR14 is not necessary after a hardware reset if the clock source is to be the $\overline{\text{RTxC}}$ pin. This is because a hardware reset automatically selects the $\overline{\text{RTxC}}$ pin as the baud rate generator clock source.

Data Encoding

The SCC provides four different data encoding methods, selected by bits D_6 and D_5 in WR10. An example for these four encoding methods is shown in Figure 35. Any encoding method may be used in any X1 mode in the SCC, asynchronous or synchronous. The data encoding selected is active even though the transmitter or receiver may be idling or disabled.

In NRZ encoding a "1" is represented by a HIGH level and a "0" is represented by a LOW level. In this encoding method only a minimal amount of clocking

Figure 35 : Data Encoding Methods.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

information is available in the data stream in the form of transitions on bit-cell boundaries. In an arbitrary data pattern this may not be sufficient to generate a clock for the data from the data itself. In the case of SDLC, where the number of consecutive "1s" in the data stream is limited, a minimum number of transitions to generate a clock are guaranteed.

In FM1 encoding, also known as biphase mark, a transition is present on every bit cell boundary, and an addition transition may be present in the middle of the bit cell. In FM1 a "0" is sent as no transition in the center of the bit cell and a "1" is sent as a transition in the center of the bit cell. FM1 encoded data contains sufficient information to recover a clock from the data.

In FM0 encoding, also known as biphase space, a transition is present on every bit cell boundary and an additional transition may be present in the middle on the bit cell. In FM0 a "1" is sent as no transition in the center of the bit cell and a "0" is sent as a transition in the center of the bit cell. FM0 encoded data contains sufficient information to recover a clock from the data.

The data encoding method should be selected in the initialization procedure before the transmitter and receiver are enabled but no other restrictions apply. Note, in Figure 35, that in NRZ and NRZI the receiver samples the data only on one edge. However, in FM1 and FM0 the receiver samples the data on both edges. Also, as shown in Figure 5-4, the transmitter defines bit cell boundaries by one edge in all cases and uses the other edge in FM1 and FM0 to create the mid-bit transition.

Digital Phase-locked Loop

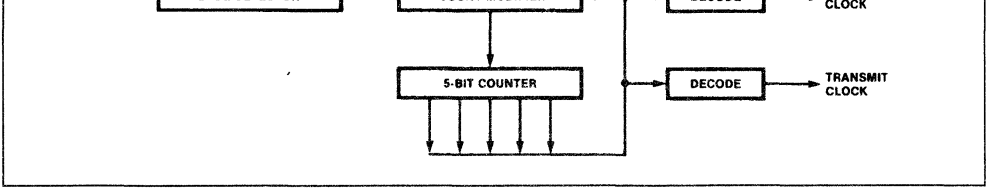
Figure 36 shows a block diagram of the digital phase-locked loop. It consists of a 5-bit counter, an edge detector, and a pair of output decoders. The clock for the DPLL comes from the output of a two-input multiplexer, and the two outputs go to the transmitter and receive clock multiplexers. The DPLL is controlled by the seven commands that are encoded in bits D_7, D_6 and D_5 of WR14.

The clock for the DPLL is selected by two of the commands in WR14. One command selects the output of the baud rate generator as the clock source, and the other command selects the RTxC pin as the clock source, independent of whether the RTxC pin is a simple input or part of the crystal oscillator circuit. To avoid metastable problems in the counter, the clock source selection should be made only while DPLL is disabled, since arbitrarily narrow pulses can be generated at the output of the multiplexer when it changes status.

The DPLL is enabled by issuing the Enter Search Mode command in WR14. This command is also used to reset the DPLL to a known state if it is suspected that synchronization has been lost. When used to enable the DPLL, the Enter Search Mode command unlocks the counter, which is held while the DPLL is disabled and enables the edge detector. If the DPLL is already enabled when this command is issued, the DPLL also enters Search Mode. While in Search mode, the counter is held at a specific count and no outputs are provided. The DPLL remains in this status until an edge is detected in the receive data stream. This first edge is assumed to occur on a bit cell boundary, and the DPLL will begin providing an output to the receiver that will properly sample the data. From this point on the DPLL will adjust its output to remain in phase with the receive data. If the first edge that the DPLL sees does not occur on a bit cell boundary, the DPLL will eventually lock on to the receive data but it will take longer to do so.

The DPLL may be programmed to operate in either of two modes, as selected by command in WR14. In the NRZI mode the DPLL clock must be 32 times the date rate. In this mode the transmit and receive clock outputs of the DPLL are indentical, and the clocks are phased so that the receiver samples the data in the middle of the bit cell. In NRZI mode the DPLL does not require a transition in every bit cell,

Figure 36 : Digital Phase-Locked Loop.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

so this mode is useful for recovering the clocking information from NRZ and NRZI data streams. In the FM mode the DPLL clock must be 16 times the data rate. In this mode the transmit clock output of the DPLL lags the receive clock outputs by 90o, to make the transmit and receive bit cell boundaries the same, because the receiver must sample FM data at one-quarters and three-quarters bit time. In FM mode the DPLL requires a transition in every bit cell, and if this transition is not present in two consecutively sampled bit cells, the DPLL will automatically enter the search mode. As in the case of the clock source selection, the mode of operation should only be changed while the DPLL is disabled to prevent unpredictable results.

NRZI Mode Operation

To operate in NRZI mode the DPLL must be supplied with a clock that is 32 times the data rate. The DPLL uses this clock, along with the receive data, to construct receive and transmit clock outputs that are phased to properly receive and transmit data. To do this, the DPLL divides each bit cell into four regions, and makes an adjustment to the count cycle of the 5-bit counter dependent upon in which region a transition on the receive data input occured. This is shown in Figure 37. Ordinarily, a bit cell boundary will occur between count 15 and count 16, and the DPLL output will cause the data to be sampled in the middle of the bit cell. The DPLL actually allows the transition marking a bit cell boundary to occur anywhere during the second half of count 15 or the first half of count 16 without making a correction to its count cycle. However, if the transition marking a bit cell boundary occurs between the middle of count 16 and count 31 the DPLL is sampling the data too early in the bit cell. In response to this the DPLL extends its count by one during the next 0 to 31 counting cycle, which effectively moves the edge of the clock that samples the receive data closer to

Figure 37 : DPLL in NRZI Mode.

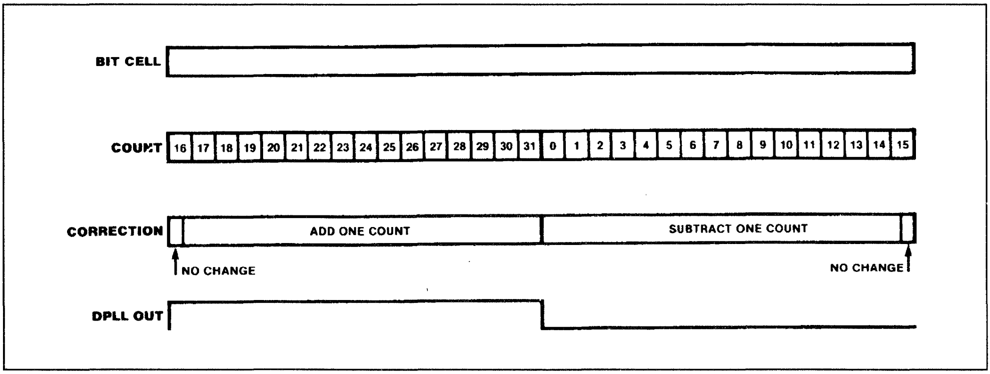
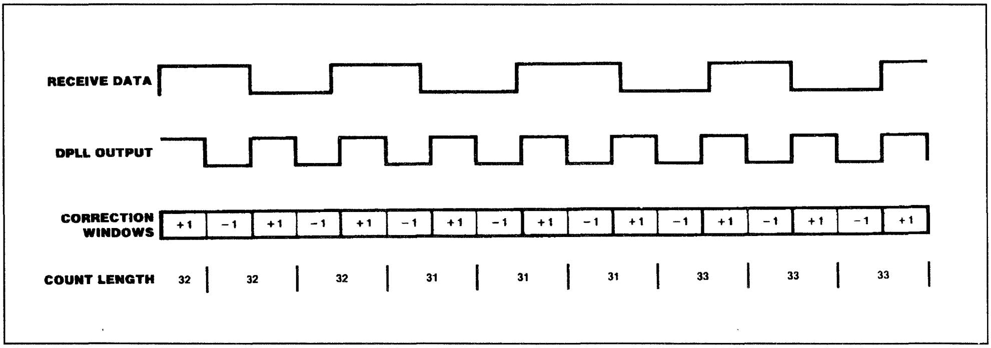


Figure 38 : DPLL Operating Example.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

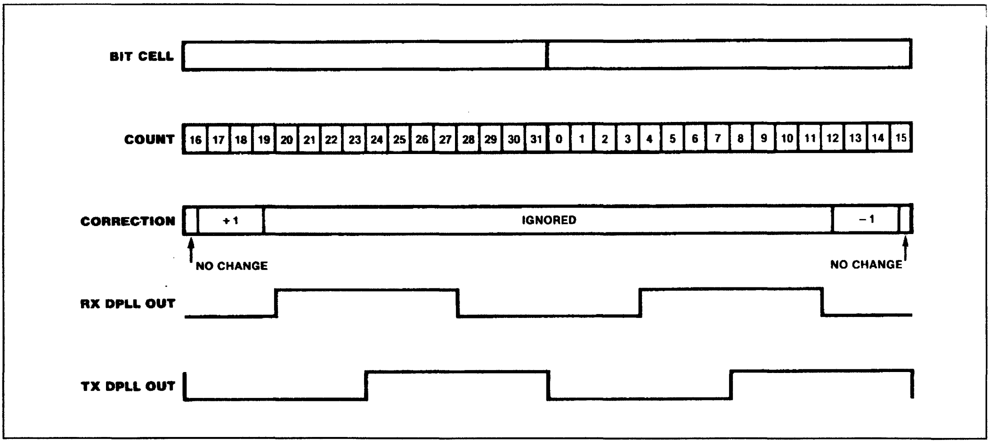
the center of the bit cell. In a similar manner, if the transition occurs between count 0 and the middle of count 15, the output of the DPLL is sampling the data too late in the bit cell. To correct this, the DPLL shortens its count by one during the next 0 to 31 counting cycle, which effectively moves the edge of the clock that samples the receive data closer to the center of the bit cell. In NRZI mode, if the DPLL does not see any transition during a counting cycle, no adjustment is made in the following counting cycle. If an adjustment to the counting cycle is necessary the DPLL modifies count five, either deleting it or doubling it. Thus only the LOW time of the DPLL output will be lengthened or shortened. While the DPLL is in search mode, the counter remains at count 16, where the DPLL outputs are both HIGH. The missing clock latches in the DPLL which may be accessed in RR10, are not used in NRZI mode. An example of the DPLL in operation is shown in Figure 38.

FM Mode Operation

To operate in FM mode the DPLL must be supplied with a clock that is 16 times the data rate. The DPLL uses this clock, along with the receive data, to construct receive and transmit clock outputs that are phased to receive and transmit data properly. In FM mode that the counter in the DPLL still counts from 0 to 31 but now each cycle corresponds to 2-bit cells. To make adjustments to remain in phase with the receive data, the DPLL divides a pair of bit cells into five regions, making the adjustment to the counter dependent upon which region the transition on the receive data input occured. This is shown in Figure 39. Ordinarily a bit cell boundary will occur between count 15 or count 16, and the DPLL receive output will cause the data to be sampled at one-fourth and three-fourths of the way through the bit cell. The DPLL actually allows the transition marking a bit-cell boundary to occur anywhere during the second half of count 15 or the first half of count 16 without making a correction to its count cycle. However, if the transition marking a bit cell boundary occurs between the middle of count 16 and the middle of count 19 the DPLL is sampling the data too early in the bit cell.In response to this the DPLL extends its count by 1 during the next 0 to 31 counting cycle, which effectively moves the receive clock edges closer to where they should be. In FM mode any transitions occurring between the middle of count 19 in one cycle and the middle of count 12 during the next cycle are ignored by the DPLL. This is necessary to guarantee that any data transitions in the bit cells will not cause an adjustment to the counting cycle.

In FM mode the transmit clock and receive clock outputs from the DPLL are not in phase. This is necessary to make the transmit and receive bit cell boundaries coincide, since the receive clock must sample the data one-fourth and three-fourths of the way through the bit cell. As in NRZI mode, if an adjustment to the counting cycle is necessary, the DPLL

Figure 39 : DPLL in FM Mode.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

modifies count 5, either deleting it or doubling it. If no adjustment is necessary, the count sequence proceeds normally. While the DPLL is in search mode, the counter remains at count 16, where the receive output is LOW and the transmit output is LOW. This fact can be used to provide a transmit clock under software control since the DPLL is in search mode while it it disabled. While the DPLL is disabled the transmit clock output of the DPLL may be toggled by alternately selecting FM and NRZI move in the DPLL. The same is true of the receive clock.

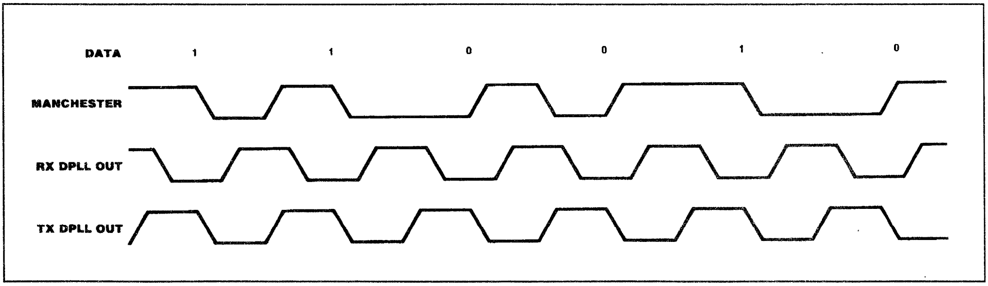
In addition to FM encoded data, the DPLL may also be used to recover the clock from Manchester encoded data, which contains a transition at the center of every bit cell. Here it is the direction of the transition that distinguishes a "1" from a "0". Another way of looking at Manchester encoding is to realize that, during the first half of the bit cell the data is sent, during the second half of the bit cell the complement of the data is sent. This is shown in Figure 40, along with the DPLL output if it thinks that the mid-bit transitions are really bit cell boundaries. As is obvious from the figure, if the receiver samples the data on the falling edge of the DPLL receive clock output, the Manschester data will be properly decoded. This occurs if the receiver is programmed to accept NRZ data.

From the above discussion together with an examination of FM0 and FM1 data encoding, it should be obvious that only clock transitions should exist on the receive data pin when the DPLL is programmed to enter search mode. If this is not case the DPLL may attempt to lock on to the data transitions. With FM0 encoding this requires continuous "1s" received when leaving search. In FM1 encoding it is continuous "0s" ; which Manchester encoded data this means alternating "1s" and "0s". With all three of these data encoding methods there will always be at least one transition in every bit cell, and in FM mode the DPLL is designed to expect this transition. In particular, if no transition occurs between the middle of count 12 and the middle of count 19, the DPLL is propably not locked onto the data properly. When the DPLL misses an edge the One Clock Missing bit is RR10 is set to "1" and latched. It will hold this value until a Reset Mission Clock command is issued in WR14 or until the DPLL is disabled or programmed to enter the Search mode. Upon missing this one edge the DPLL takes no other action and does not modify its count during the next counting cycle. However, if the DPLL does not see an edge between the middle of count 12 and the middle of count 19 in two successive 0 to 31 count cycles, a line error condition is assumed. If this occurs, the two Clocks Mission bit in RR10 is set to "1" and latched. At the same time the DPLL enters the Search mode. The DPLL makes the decision to enter Search mode during count 2, where both the receive clock and transmit clock outputs are LOW. This prevents any glitches on the clock outputs when search mode is entered. While in search mode no clock outputs are provided by the DPLL. The Two Clocks Missing bit in RR10 is latched until a Reset Missing Clock command is issued in WR14, or until the DPLL is disabled or programmed to enter the Search mode.

DPLL Initialization

Initialization of the DPLL may be done at any time during the initialization sequence, but should probably be done after the clock modes have been selected in WR11, and before the receiver and transmitter are enabled. When initializing the DPLL the clock source should be selected first, followed by the selection of the operating mode. At this point the DPLL, may be enabled by issuing the Enter Search Mode command in WR14. Note that a channel or hardware reset disables the DPLL, selects the $\overline{\text{RTxC}}$ pin as the clock source for the DPLL, and places it in the NRZI mode.

Figure 40 : Manchester Clock Recovery.



SUPPORT CIRCUITRY PROGRAMMING (cont'd)

Internal Loopback/Auto Echo

The SCC contains two other features useful for diagnostic purposes, controlled by bits in WR14. They are local loopback and auto echo.

Local loopback is selected when bit D4 of WR14 is set to "1". In this mode the output of the transmitter is internally connected to the input of the receiver. At the same time the TxD pin remains connected to the transmitter. In this mode the $\overline{\text{DCD}}$ pin is ignored as a receive enable and the $\overline{\text{CTS}}$ pin is ignored as a transmitter enable even if the Auot Enables mode has been selected. Note that the DPLL input is connected to the RxD pin, not to the input of the receiver. This precludes the use of the DPLL in local loopback.

Auto echo is selected when bit D3 of WR14 is set to "1". In this mode the TxD pin is connected directly to the RxD pin, and the receiver input is connected to the RxD pin. In this mode the $\overline{\text{CTS}}$ pin is ignored as a transmitter enable and the output of the transmitter does not connect to anything. If both the Local Loopback and Auto Echo bits are set to "1", the auto echo mode will be selected, but both the $\overline{\text{CTS}}$ pin and $\overline{\text{DCD}}$ pin will be ignored as auto enables. This, however, should not be considered a normal operating mode, however. Local Loopback is shown schematically in Figure 41 and auto echo is shown schematically in Figure 42.

Figure 41 : Local Loopback.

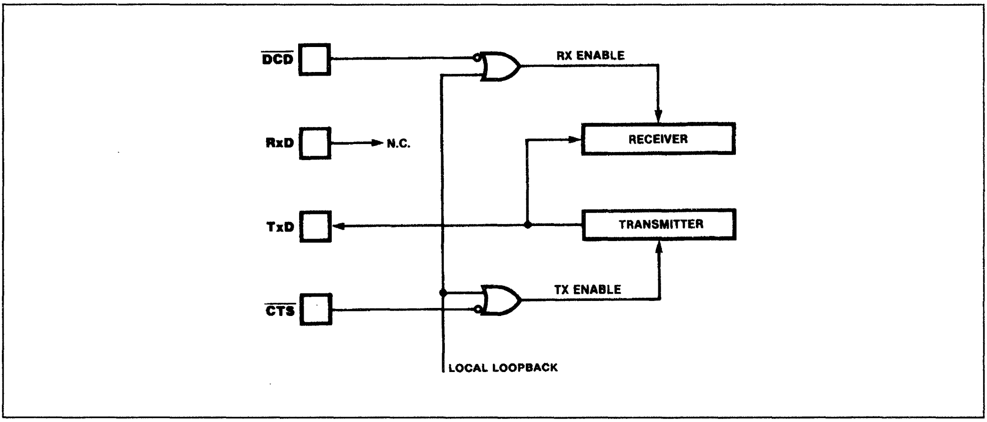
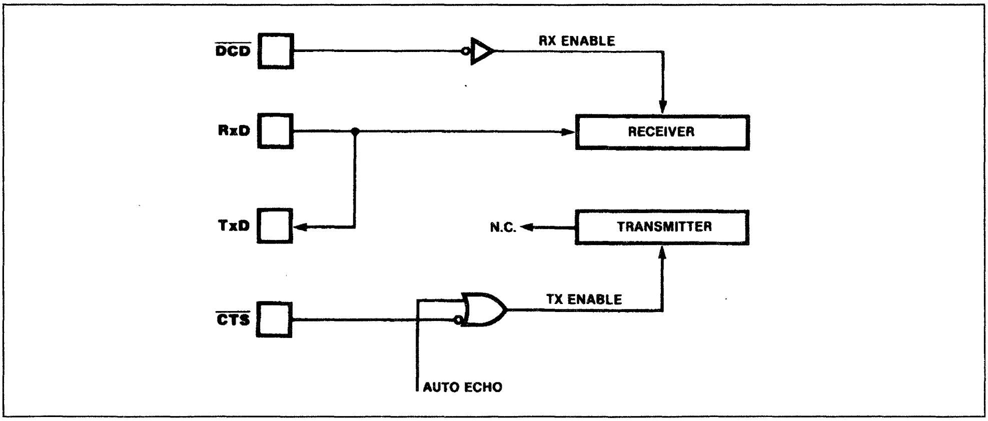


Figure 42 : Auto Echo.



REGISTERS DESCRIPTION

The following section describes the SCC registers. Each register is detailed in terms of bit configuration, the active states (See Table 8) of each bit, their definitions, their functions, and their effects upon the internal hardware and external pins.

Table 8 : SCC Registers Description.

Read Register	Description
RR0	Xmit/Receive Buffer Status and Ext Status
RR1	Receive Condition Status/Residue Codes
RR2	Interrupt Vector (modified in BChannel)
RR3	Interrupt Pending (Channel Aonly)
RR8	Receive Buffer
RR10	Loop/Clock Status
RR12	Lower Byte of Time Constant
RR13	Upper Byte of Time Constant
RR15	External Status Interrupt Enable
Write Register	Description
WRO	Command Register
WR1	Tx/Rx Interrupt and Data Xfer Mode Definition
WR2	Interrupt Vector
WR3	Receive Parameters and Control
WR4	Tx/Rx Miscellaneous Parameters and Modes
WR5	Transmit Parameter and Controls
WR6	Sync Character or SDLC Address Field
WR7	Sync Character or SDLC Flag
WR8	Transmit Buffer
WR9	Master Interrupt Control
WR10	Misc Transmitter/Receiver Control Bits
WR11	Clock Mode Control
WR12	Lower Byte Baud Rate Generator Time Constant
WR13	Upper Byte of Baud Rate Generator Time Constant
WR14	Miscellaneous Control Bits
WR15	External Status/Interrupt Control

WRITE REGISTERS

The SCC write register set in each channel includes ten control registers (among them is the transmit buffer), two sync character registers, and two baud rate time constant registers. The interrupt control register and the master interrupt control and reset register are shared by both channels.

Write Register 0 (command register). WR0 is the command register and the CRC reset code register. Figure 43 shows the bit configuration for the Z8530 and includes register select bits in addition to command and reset codes.

Bits D7 and D6 : CRC Reset Codes 0 And 1

Null Code (00). This command has no effect on the SCC and is used when a write to WR0 is necessary for some reason other than a CRC Reset command.

Reset Receive CRC Checker (01). This command is used to initialize the receive CRC circuitry. It is necessary in synchronous modes (except SDLC) if the Enter Hunt Mode command in Write Register 3 is not issued between received messages. Any action that disables the receiver initializes the CRC circuitry. Resetting the Receive CRC Checker command is accomplished automatically in SDLC mode.

Reset Transmit CRC Generator (10). This command initializes the CRC generator. It is usually issued in the initialization routine and after the CRC has been transmitted. A Channel Reset will not initialize the generator and this command should not be issued until after the transmitter has been enabled in the initialization routine.

Reset Transmit Underrun/EOM Latch (11). This command controls the transmission of CRC at the end of transmission (EOM). If this latch has been reset, and a transmit underrun occurs, the SCC automatically appends CRC to the mes-sage. In SDLC mode with Abort on Underrun se-lected, the SCC sends an abort, and Flag on under-run if the TX Underrun/EOM latch as been reset.

At the start of CRC transmission, the Tx Underrun/EOM latch is set. The Reset command can be issued at any time during a message. If the transmitter is disabled, this command will not reset the latch. However, if no External Status interrupt is pending, or if a Reset External Status Int command accompanies this command while the transmitter is disabled, an External/Status interrupt is generated with the Tx Underrun/ EOM bit reset in RR0.

Bits D5-D3 : Command Codes

Null Code (000). The Null command has no effect on the SCC.

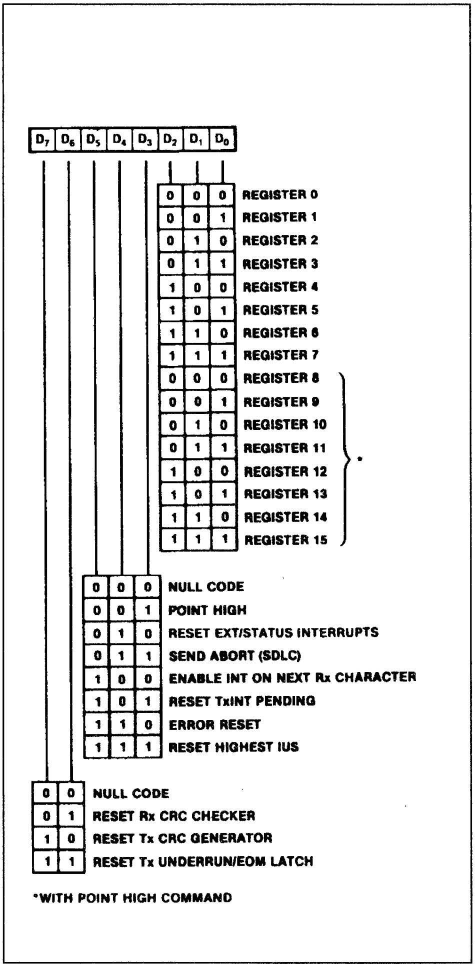
Point High (001). This command effectively adds eight to the Register Pointer (B2-B0) by allowing WR8 through WR15 to be accessed. The Point High command and the Register Pointer bits are written simultaneously.

Reset External/Status Interrupts (010). After an External/Status interrupt (a change on a modem line or a break condition, for example), the status bits in RR0 are latched. This command re-enables the bits and allows interrupts to occur again as a result of a status change. Latching the status bits captures short pulses until the CPU has time to read the change. The SCC contains simple queueing logic associated with most of the external status bits

REGISTERS DESCRIPTION (cont'd)

in RR0. If another External/Status condition changes while a previous condition is still pending (Reset External/Status Interrupts has not yet been issued) and this condition persists until after the command is issued, this second change causes another External/Status interrupt. However, if this second status change does not persist (there are two transitions), another interrupt is not generated. Exceptions to this rule are detailed in the RR0 description.

Figure 43 : Write Register 0 (Z8530).



D2	D1	D0	
0	0	0	REGISTER 0
0	0	1	REGISTER 1
0	1	0	REGISTER 2
0	1	1	REGISTER 3
1	0	0	REGISTER 4
1	0	1	REGISTER 5
1	1	0	REGISTER 6
1	1	1	REGISTER 7
0	0	0	REGISTER 8 *
0	0	1	REGISTER 9 *
0	1	0	REGISTER 10 *
0	1	1	REGISTER 11 *
1	0	0	REGISTER 12 *
1	0	1	REGISTER 13 *
1	1	0	REGISTER 14 *
1	1	1	REGISTER 15 *

D5	D4	D3	
0	0	0	NULL CODE
0	0	1	POINT HIGH
0	1	0	RESET EXT/STATUS INTERRUPTS
0	1	1	SEND ABORT (SDLC)
1	0	0	ENABLE INT ON NEXT Rx CHARACTER
1	0	1	RESET TxINT PENDING
1	1	0	ERROR RESET
1	1	1	RESET HIGHEST IUS

D7	D6	
0	0	NULL CODE
0	1	RESET Rx CRC CHECKER
1	0	RESET Tx CRC GENERATOR
1	1	RESET Tx UNDERRUN/EOM LATCH

*WITH POINT HIGH COMMAND

Send Abort (011). This command is used in SDLC mode to transmit a sequence of eight to thirteen "1s". This command always empties the transmit buffer and sets Tx Underrun/EOM bit in Read Register 0.

Enable Interrupt On Next Rx Character (100). If the interrupt on the First Received Character mode is selected, this command is used to reactivate that mode after each message is received. The next character to enter the receive FIFO causes a Receive interrupt. Alternatively,, the first previously stored character in the FIFO will cause a Receive interrupt.

Reset Tx Interrupt Pending (101). This command is used in cases where there are no more characters to be sent ; e.g., at the end of a message. This command prevents further transmit interrupts until after the next character has been loaded into the transmit buffer or until CRC has been completely sent. This command is necessary to prevent the transmitter from requesting an interrupt when the transmit buffer becomes empty (with Transmit Interrupt Enabled).

Error Reset (110). This command resets the error bits in RR1. If Interrupt on First Rx Character or Interrupt on Special Condition modes are selected and a special condition exists, the data with the special condition is held in the receive FIFO until this command is issued. If either of these modes is selected and this command is issued before the data has been read from the receive FIFO, the data is lost.

Reset Highest IUS (111). This command resets the highest priority Interrupt Under Service (IUS) bit, allowing lower priority conditions to request interrupts. This command allows the use of the internal daisy chain (even in systems without an extenal daisy chain) and should be the last operation in an interrupt service routine.

Bits 2 through 0 : Register Selection Code

These three bits select Registers 0 through 7. With the Point High command, Registers 8 through 15 are selected.

The following is a summary of the bit descriptions for each write register (WR1-WR15)

Write Register 1 (transmit/receive interrupt and data transfer mode definition). Write Register 1 is the control register for the various SCC interrupt and Wait/Request modes. Figure 44 shows the bit assignments for WR1.

REGISTERS DESCRIPTION (cont'd)

Bit 7 : WAIT/DMA Request Enable

This bit enables the Wait/Request function in conjunction with the Request/Wait Function Select bit (B6). If bit 7 is set to "1", the state of bit 6 determines the activity of the WAIT/REQUEST pin (Wait or Request). If bit 7 is set to "0", the selected function (bit 6) forces the WAIT/REQUEST pin in to the appropriate inactive state (High for Request, floating for Wait).

Bit 6 : WAIT/DMA Request Function

The request function is selected by setting this bit to "1". In the Request mode, the WAIT/REQUEST pin switches from High to Low when the SCC is ready to transfer data. When this bit is "0", the wait function is selected. In the Wait mode, the WAIT/REQUEST pin switches from floating to Low when the CPU attempts to transfer data before the SCC is ready.

Bit 5 : WAIT/DMA Request On Receive Transmit

This bit determines whether the WAIT/REQUEST pin operates in the Transmit mode or the Receive mode. When set to "1", this bit allows the wait/request function to follow the state of the receive buffer ; i.e., depending on the state of bit 6, the WAIT/REQUEST pin is active or inactive in relation to the empty or full state of the receive buffer. Conversely, if this bit is set to "0", the state of the WAIT/REQUEST pin is determined by bit 6 and the state of the transmit buffer. (Note that a transmit request function is available on the DTR/REQUEST pin. This allows full-duplex operation under DMA control for both channels.)

The request function may occur only when the SCC is not selected ; e.g., if the internal request becomes active while the SCC is in the middle of a read or write cycle, the external request will not become active until the cycle is complete. An active request output causes a DMA controller to initiate a read or write operation. If the request on Transmit mode is selected in either SDLC or Synchronous Mode, the Request pin is pulsed Low for one PCLK cycle at the end of CRC transmission to allow the immediate transmission of another block of data.

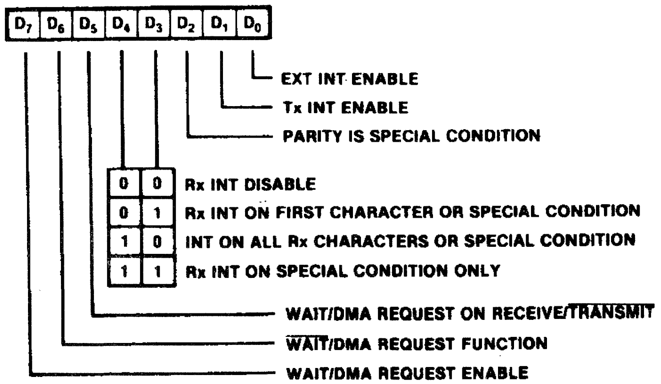
If the Wait On Receive mode, the WAIT pin is active if the CPU attempts to read SCC data that has not yet been received. In the Wait On Transmit mode, the WAIT pin is active if the CPU attempts to write data when the transmit buffer is still full. Both situations can occur frequently when block transfer instructions are used.

Bits 4 and 3 : Receive Interrupt Modes

These two bits specify the various character-available conditions that may cause interrupt requests.

Receive Interrupts Disabled (00). This mode prevents the receiver from requesting an interrupt and is normally used in a polled environment where either the status bits on RR0 or the modified vector in RR2 (Channel B) can be monitored to initiate a service routine. Although the receiver interrupts are disabled, a special condition can still provide a unique vector status in RR2.

Figure 44 : Write Register 1.


D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0
EXT INT ENABLE
Tx INT ENABLE
PARITY IS SPECIAL CONDITION
0 0 Rx INT DISABLE
0 1 Rx INT ON FIRST CHARACTER OR SPECIAL CONDITION
1 0 INT ON ALL Rx CHARACTERS OR SPECIAL CONDITION
1 1 Rx INT ON SPECIAL CONDITION ONLY
WAIT/DMA REQUEST ON RECEIVE/TRANSMIT
WAIT/DMA REQUEST FUNCTION
WAIT/DMA REQUEST ENABLE

REGISTERS DESCRIPTION (cont'd)

Receive Interrupt On First Character Or Special Condition (01). The receiver requests an interrupt in this mode on the first available character (or stored FIFO character) or on a special condition. Sync characters to be stripped from the message stream do not cause interrupts.

Special receive conditions are : receiver overrun, framing error, end of frame, or parity error (if selected). If a special receive condition occurs, the data containing the error is stored in the receive FIFO until an Error Reset command is issued by the CPU.

This mode is usually selected when a Block Transfer mode is used. In this interrupt mode, a pending special receive condition remains set until either an Error Reset command, a channel or hardware reset, or until receive interrupts are disabled.

The Receive Interrupt on First Character or Special Condition mode can be re-enabled by the Enable Rx Interrupt on Next Character command in WR0.

Interrupt On All Receive Characters Or Special Condition (10). This mode allows an interrupt for every character received (or character in the receive FIFO) and provides a unique vector when a special condition exists. The receiver Overrun bit and the Parity Error bit in RR1 are two special conditions that are latched. These two bits must be reset by the Error Reset command. Receiver overrun is always a special receive condi-tion, and parity can be programmed to be a special condition.

Data characters with special receive conditions are not held in the receive FIFO in the Interrupt On All Receive Characters or Special Conditions Mode as they are in the other receive interrupt modes.

Receive Interrupt On Special Condition (11). This mode allows the receiver to interrupt only on characters with a special receive condition. When an interrupt occurs, the data containing the error is held in the receive FIFO until an Error Reset command is issued. When using this mode in conjunction with a DMA, the DMA can be initialized and enabled before any characters have been received by the SCC. This eliminates the time critical section of code required in the Receive Interrupt on First Character or Special condition mode ; i.e., all data can be transferred via the DMA so that the CPU need not handle the first received character as a special case.

Bit 2 : Parity Is Special Condition

If this bit is set to "1", any received characters with parity not matching the sense programmed in WR4 give rise to a Special Receive Condition. If parity is disabled (WR4), this bit is ignored. A special condition modifies the status of the interrupt vector stored in WR2. During an interrupt acknowledge cycle, this vector can be placed on the data bus.

Bit 1 : Transmitter Interrupt Enable

If this bit is set to "1", the transmitter requests an interrupt whenever the transmit buffer becomes empty.

Bit 0 : External/Status Master Interrupt Enable

This bit is the master enable for External/Status interrupts including $\overline{\text{DCD}}$, $\overline{\text{CTS}}$, $\overline{\text{SYNC}}$ pins, break, abort, the beginning of CRC transmission when the Transmit/Underrun/EOM latch is set, or when the counter in the baud rate generator reaches "0". Write Register 15 contains the individual enable bits for each of these sources of External/Status interrupts. This bit is reset by a channel or hardware reset.

Figure 45 : Write Register 2.

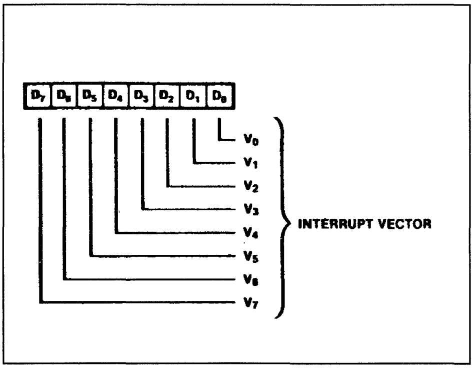
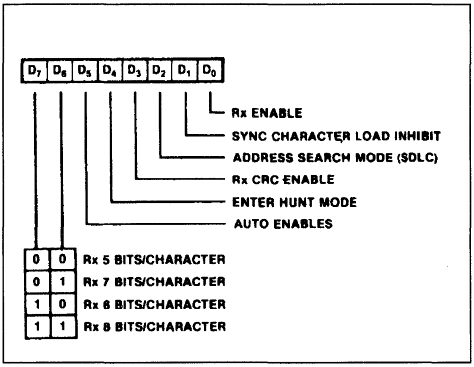


Figure 46 : Write Register 3.



REGISTERS DESCRIPTION (cont'd)

Write Register 2 (interrupt vector). WR2 is the interrupt vector register. Only one vector register exists in the SCC, but it can be accessed through either channel. The interrupt vector can be modified by status information. This is controlled by the Vector Includes Status (VIS) and the Status High/Status Low bits in WR9. The bit positions for WR2 are shown in Figure 45.

Write Register 3 (receive parameters and control). This register contains the control bits and parameters for the receiver logic as illustrated in Figure 46.

Bits 7 and 6 : Receiver Bits/Character

The state of these two bits determines the number of bits to be assembled as a character in the received serial data stream. The number of bits per character can be changed while a character is being assembled but only before the number of bits currently programmed is reached. Unused bits in the Received Data Register (RR8) are set to "1" in asynchronous modes and SDLC modes, the SCC merely transfers an 8-bit section of the serial data stream to the receive FIFO at the appropriate time. Table 9 lists the number of bits per character in the assembled character format.

Table 9 : Receive Bits/Character

B_7	B_6	
0	0	5 Bits / Character
0	1	7 Bits / Character
1	0	6 Bits / Character
1	1	8 Bits / Character

Bit 5 : Auto Enables

This bit programs the function for both the $\overline{\text{DCD}}$ and $\overline{\text{CTS}}$ pins. $\overline{\text{CTS}}$ becomes the transmitter enable and $\overline{\text{DCD}}$ becomes the receiver enable when this bit is set to "1". However, the Receiver Enable and Transmit Enable bits must be set before the $\overline{\text{DCD}}$ and $\overline{\text{CTS}}$ pins can be used in this manner. When the Auto Enables bit is set to "0", the $\overline{\text{DCD}}$ and $\overline{\text{CTS}}$ pins are merely inputs to the corresponding status bits in Read Register 0. The state of $\overline{\text{DCD}}$ is ignored in the Local Loopback mode. The state of $\overline{\text{CTS}}$ is ignored in both Auto Echo and Local Loopback modes.

Bit 4 : Enter Hunt Mode

This command forces the comparison of sync characters or flags to assembled receive characters for the purpose of synchronization. After reset, the SCC automatically enters the Hunt mode (except asynchronous). Whenever a flag or sync character is matched, the Sync/Hunt bit in Read Register 0 is reset and, if External/Status Interrupt Enable is set, an interrupt sequence is initiated. The SCC automatically enters the Hunt mode when an abort condition is received or when the receiver is disabled.

Bit 3 : Receiver CRC Enable

This bit is used to initiate CRC calculation at the beginning of the last byte transferred from the Receiver Shift register to the receive FIFO. This operation occurs independently of the number of bytes in the receive FIFO. When a particular byte is to be excluded from CRC calculation, this bit should be reset before the next byte is transferred to the receive FIFO. If this feature is used, care must be taken to ensure that eight bits per character is selected in the receiver because of an inherent delay from the Receive Shift register to the CRC checker.

This bit is internally set to "1" in SDLC mode and the SCC calculates CRC on all bits except inserted zeros between the opening and closing character flags. This bit is ignored in asynchronous mode.

Bits 2 : Adress Search Mode (SDLC)

Setting this bit in SDLC mode causes messages with addresses not matching the address programmed in WR6 to be rejected. No receiver interrupts can occur in this mode unless there is an address match. The address that the SCC attempts to match can be unique (1 in 256) or multiple (16 in 256), depending on the state of Sync Character Load Inhibit bit. The Address Search mode bit is ignored in all modes except SDLC.

Bit 1 : SYNC Character Load Inhibit

If this bit is set to "1" in any synchronous mode except SDLC, the SCC compares the byte in WR6 with the byte about to be stored in the FIFO, and it inhibits this load if the bytes are equal. The SCC does not calculate the CRC on bytes stripped from the Data stream in the manner. If the 6-bit sync option is selected while in Monosync mode, the compare is still across eight bits, so WR6 must be programmed for proper operation.

If the 6-bit sync option is selected with this bit set to "1", all sync characters except the one immediately preceding the data are stripped from the message. If the 6-bit sync option is selected while in the Bisync mode, this bit is ignored.

The address recognition logic of the receiver is modified in SDLC mode if this bit is set to "1", i.e., only the four most significant bits of WR6 must match the receiver address. This procedure allows the

REGISTERS DESCRIPTION (cont'd)

SCC to receive frames from up to 16 separate sources without programming WR6 for each source (if each station address has the four most significant bits in common). The address field in the frame is still eight bits long.

This bit is ignored in SDLC mode if Address Search mode has not been selected.

Bit 0 : Receiver Enable

When this bit is set to "1", receiver operation begins. This bit should be set only after all other receiver parameters are established and the receiver is completely initialized. This bit is reset by a channel or hardware reset command, and it disables the receiver.

Write Register 4 (transmit/receiver miscellaneous parameters and modes). WR4 contains the control bits for both the receiver and the transmitter. These bits should be set in the transmit and receiver initialization routine before issuing the contents of WR1, WR3, WR6, and WR7. Bit positions for WR4 are shown in Figure 47.

Bits 7 and 6 : Clock Rate 1 And 0

These bits specify the multiplier between the clock and data rates. In synchronous modes, the 1S mode is forced internally and these bits are ignored unless External Sync mode has been selected.

1X Mode (00). The clock rate and data rate are the same. In External Sync mode, this bit combination specifies that only the SYNC pin can be used to achieve character synchronization.

16X Mode (01). The clock rate is 16 times the data rate. In External Sync mode, this bit combin-ation specifies that only the SYNC pin can be used to achieve character synchronization.

32X Mode (10). The clock rate is 32 times the data rate. In External Sync mode, this bit combination specifies that either the SYNC pin or a match with the character stored in WR7 will signal character synchronization. The sync character can be either six or eight bits long as specified by the 6-bit/8-bit Sync bit in WR10.

64X Mode (11). The clock rate is 64 times the data rate. With this bit combination in External Sync mode, both the receiver and transmitter are placed in SDLC mode. The only variation from normal SDLC operation is that the SYNC pin can be used to start or stop the reception of a frame by forcing the receiver to act as though a flag had been received.

Bits 5 and 4 : SYNC Modes 1 And 0

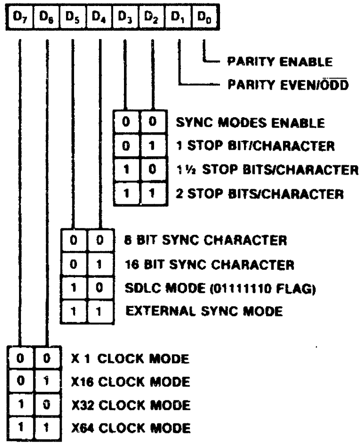
These two bits select the various options for character synchronization. They are ignored unless synchronous modes are selected in the stop bits field of this register.

Monosync (00). In this mode, the receiver achieves character synchronization by matching the character stored in WR7 with an identical character in the received data stream. The transmitter uses the character stored in WR6 as a time fill. The sync character can be either six or eight bits, depending on the state of the 6-bit/8-bit Sync bit in WR10. If the Sync Character Load Inhibit bit is set, the receiver strips the contents of WR6 from the data stream if received within character boundaries.

Bisync (01). The concatenation of WR7 with WR6 is used for receiver synchronization and as a time fill by the transmitter. The sync character can be 12 or 16 bits in the receiver, depending on the state of the 6-bit/8-bit Sync bit in WR10. The transmitted character is always 16 bits.

SDLC Mode (10). In this mode, SDLC is selected

Figure 47 : Write Register 4.



REGISTERS DESCRIPTION (cont'd)

and requires a Flag (01111110) to be written to WR7. The receiver address field should be written to WR6. The SDLC CRC polynomial must also be selected (WR5) in SDLC mode.

External Sync Mode (11). In this mode, the SCC expects external logic to signal character synchronization via the $\overline{\text{SYNC}}$ pin. If the crystal oscillator option is selected (in WR11), the internal $\overline{\text{SYNC}}$ signal is forced to "0". In this mode, bits B7-B6 of this register select special version of External Sync mode. In this mode, the transmitter is in Monosync mode using the contents of WR6 as the time fill with the sync character length specified by the 6-bit/8-bit Sync bit in WR10.

Bits 3 and 2 : Stop Bits 1 and 0

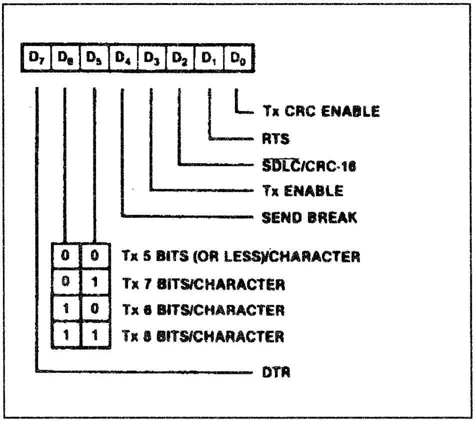
These bits determine the number of stop bits added to each asynchronous character that is transmitted. The receiver always checks for one stop bit in Asynchronous mode. A Special mode specifies that a Synchronous mode is to be selected. B2 is always set to "1" by a channel or hardware reset to ensure that the $\overline{\text{SYNC}}$ pin is in a known state after a reset.

Synchronous Modes Enable (00). This bit combination selects one of the synchronous modes specified by bits B4, B5, B6, and B7 of this register and forces the 1X Clock mode internally.

1 Stop Bit/Character (01). This bit selects Asynchronous mode with one stop bit per character.

1 1/2 Stop Bits/Character (10). These bits select Asynchronous mode with 1-1/2 stop bits per character. This mode can not be used with the 1X clock mode.

Figure 48 : Write Register 5.



2 Stop Bits/Character (11). These bits select Asynchronous mode with two stop bits per transmitted character and check for one received stop bit.

Bit 1 : Parity Even/$\overline{\text{Odd}}$

This bit determines whether parity is checked as even or odd. A "1" programmed here selects even parity, and a "0" selects odd parity. This bit is ignored if the Parity Enable bit is not set.

Bit 0 : Parity Enable

When this bit is set, an additional bit position beyond those specified in the bits/character control is added to the transmitted data and is expected in the receive data. The Received Parity bit is transferred to the CPU as part of the data unless eight bits per character is selected in the receiver.

Write Register 5 (transmit parameter and controls). WR5 contains control bits that affect the operation of the transmitter. B2 affects both the transmitter and the receiver. Bit positions for WR5 are shown in Figure 48.

Bit 7 : Data Terminal Ready

This is the control bit for the $\overline{\text{DTR}}/\overline{\text{REQ}}$ pin while the pin is in the DTR mode (selected in WR14). When set, $\overline{\text{DTR}}$ is Low ; when reset, $\overline{\text{DTR}}$ is High. This bit is ignored when $\overline{\text{DTR}}/\overline{\text{REQ}}$ is programmed to act as a $\overline{\text{REQUEST}}$ pin. This bit is reset by a channel or hardware reset.

Bits 6 and 5 : TXBits/Character 1 and 0

These bits control the number of bits in each byte transferred to the transmit buffer. Bits sent must be right justified with lease significant bits first.

The Five Or Less mode allows transmission of one to five bits per character ; however, the CPU should form at the data character as shown below in Table 10. In the Six or Seven Bits/Character modes, unused data bits are ignored.

Table 10 : Tx Bits/Character 1 and 0.

<table>
<tr><td colspan="4">Tx BITS / CHAR 1</td><td colspan="4">Tx BITS / CHAR 0</td><td></td></tr>
<tr><td colspan="4">0</td><td colspan="4">0</td><td>5 or less bits / character</td></tr>
<tr><td colspan="4">0</td><td colspan="4">1</td><td>7 bits / character</td></tr>
<tr><td colspan="4">1</td><td colspan="4">0</td><td>6 bits / character</td></tr>
<tr><td colspan="4">1</td><td colspan="4">1</td><td>8 bits / character</td></tr>
<tr><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Sends one data bit</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Sends two data bits</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Sends three data bits</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Sends four data bits</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Sends five data bits</td></tr>
</table>

REGISTERS DESCRIPTION (cont'd)

Bit 4 : Send Break

When set, this bit forces the TxD output to send continuous "0s" beginning with the following transmit clock, regardless of any data being transmitted at the time. This bit functions whether or not the transmitter is enabled. When reset, TxD continues to send the contents of the Transmit Shift register, which might be syncs, data or all "1s". If this bit is set while in the X21 mode (Monosync and Loop mode selected) and character synchronization is achieved in the receiver, this bit is automatically reset and the transmitter begins sending syncs or data. This bit can also be reset by a channel or hardware reset.

Bit 3 : Transmit Enable

Data is not transmitted until this bit is set, and the TxD output sends continuous "1s" unless Auto Echo mode or SDLC Loop mode is selected. If this bit is reset after transmission started, the transmission of data or sync characters is completed. If the transmitter is disabled during the transmission of a CRC character, sync or flag characters are sent instead of CRC. This bit is reset by a channel or hardware reset.

Bit 2 : $\overline{\text{SDLC}}$/CRC-16

This bit selects the CRC polynomial used by both the transmitter and receiver. When set, the CRC-16 polynomial is used ; when reset, the SDLC polynomial is used. The SDLC/CRC polynomial must be selected when SDLC mode is selected. The CRC generator and checker can be preset to all "0s" or all "1s", depending on the state of the Preset 1/ Preset 0 bit in WR10.

Bit 1 : Request To Send

This is the control bit for the RTS pin. When the RTS bit is set, the RST pin goes Low ; when reset, RTS goes High. In the Asynchronous mode with the Auto Enables bit set, RST goes High only after all bits of the character have been sent and the transmit buffer is empty. In synchronous modes of the Asynchronous mode with auto enables off, the pin directly follows the state of this bit. This bit is reset by a channel or hardware reset.

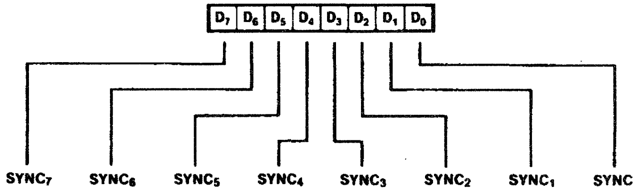
Bit 0 : Transmit CRC Enable

This bit determines whether or not CRC is calculated on a transmit character. If this bit is set at the time the character is loaded from the transmit buffer to the Transmit Shift register, CRC is calculated on that character. CRC is not automatically sent unless this bit is set when the transmit underrun exists.

Write Register 6 (sync characters or SDLC address field). WR6 is programmed to contain the transmit sync character in the Monosync mode, the first byte of a 16-bit sync character in the External Sync mode. WR6 is not used in asynchronous modes. In the SDLC modes, it is programmed to contain the secondary address field used to compare against the address field of the SDLC Frame. In SDLC mode, the SCC does not automatically transmit the station address at the beginning of the response frame. Bit positions for WR6 are shown in Figure 49.

Write Register 7 (sync character or SDLC flag). WR7 is programmed to contain the receive sync character in the Monosync mode, a second byte (the last eight bits) of a 16-bit sync character in the Bisync mode, or a Flag character (01111110) in the SDLC modes. WR7 may hold the receive sync character or a flag if one of the special versions of the External Sync mode is selected. WR7 is not used in Asynchronous mode. Bit positions for WR7 are shown in Figure 50.

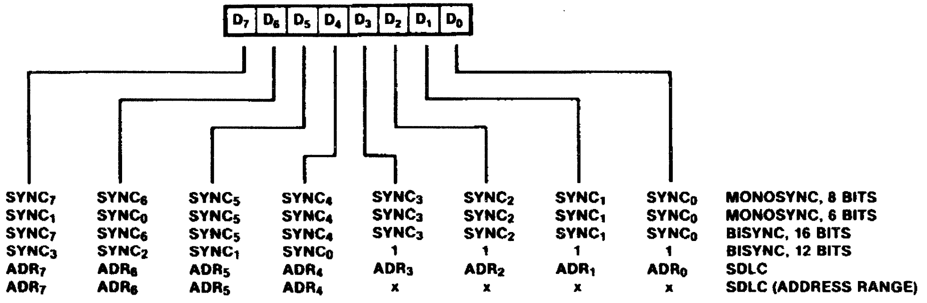
Figure 49 : Write Register 6.



SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 8 BITS
SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	x	x	MONOSYNC, 6 BITS
SYNC$_{15}$	SYNC$_{14}$	SYNC$_{13}$	SYNC$_{12}$	SYNC$_{11}$	SYNC$_{10}$	SYNC$_9$	SYNC$_8$	BISYNC, 16 BITS
SYNC$_{11}$	SYNC$_{10}$	SYNC$_9$	SYNC$_8$	SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	BISYNC, 12 BITS
0	1	1	1	1	1	1	0	SDLC

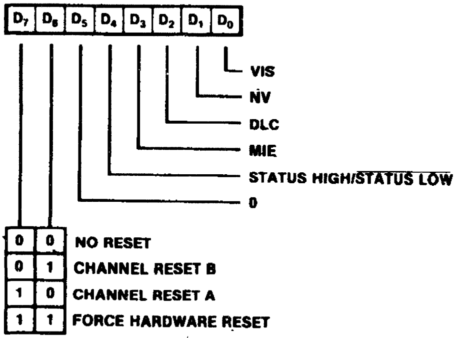
REGISTERS DESCRIPTION (cont'd)

Figure 50 : Write Register 7.



D$_7$	D$_6$	D$_5$	D$_4$	D$_3$	D$_2$	D$_1$	D$_0$	
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 8 BITS
SYNC$_1$	SYNC$_0$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	MONOSYNC, 6 BITS
SYNC$_7$	SYNC$_6$	SYNC$_5$	SYNC$_4$	SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	BISYNC, 16 BITS
SYNC$_3$	SYNC$_2$	SYNC$_1$	SYNC$_0$	1	1	1	1	BISYNC, 12 BITS
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	ADR$_3$	ADR$_2$	ADR$_1$	ADR$_0$	SDLC
ADR$_7$	ADR$_6$	ADR$_5$	ADR$_4$	x	x	x	x	SDLC (ADDRESS RANGE)

Figure 51 : Write Register 9 .



D$_7$	D$_6$	
0	0	NO RESET
0	1	CHANNEL RESET B
1	0	CHANNEL RESET A
1	1	FORCE HARDWARE RESET

Write Register 8 (transmit buffer). WR8 is the transmit bufffer register.

Write Register 9 (master interrupt control). WR9 is the Master Interrupt Control register and contains the Reset command bits. Only one WR9 exists in the SCC and can be accessed from either channel. The interrupt control bits can be programmed at the same time as the Reset command because these bits are only reset by a hardware reset. Bit positions for WR9 are shown in Figure 51.

Bits 7 and 6 : Reset Command Bits

Together, these bits select one of the reset commands for the SCC. Setting either of these bits to "1" disables both the receiver and the transmitter in the corresponding channel, forces TxD for that channel marking, forces the modem control signals High in that channel, resets all IPs and IUSs and disables all interrupts in that channel. Four extra PCLK cycles must be allowed beyond the usual cycle time after any of the active reset commands is issued before any additional commands or controls are written to the channel affected. Four extra PCLK cycles must be allowed beyond the usual cycle time before any additional command or controls are written to the SCC.

REGISTERS DESCRIPTION (cont'd)

No Reset (00). This command has no effect. It is used when a write to WR9 is necessary for some reason other than an SCC Reset command.

Channel Reset B (01). Issuing this command causes a channel reset to be performed on Channel B.

Channel Reset A (10). Issuing this command causes a channel reset to be performed on Channel A.

Force Hardware Reset (11). The effects of this command are identical to those of a hardware reset, except that the Shift Right/Shift Left bit is not changed and the MIE, Status High/Status Low and DLC bits take the programmed values that accompany this command.

Bit 5 : Not Used

Must be "0".

Bit 4 : Status High/Status Low

This bit controls which vector bits the SCC will modify to indicate status. When set to"1", the SCC modifies bits V6, V5, and V4 according to Table 11. When set to "0", the SCC modifies bits V1, V2, and V3 according to Table 11. This bit controls status in both the vector returned during an interrupt acknowledge cycle and the status in RR2B. This bit is reset a hardware reset.

Table 11 : Interrupt Vector Modification.

V3	V2	V1	Status High / Status Low = 0
V4	V5	V6	Status High / Status Low = 1
0	0	0	Ch B Transmit Buffer Empty
0	0	1	Ch B External / Status Change
0	1	0	Ch B Receive Character Avail.
0	1	1	Ch B Special Receive Condition
1	0	0	Ch A Transmit Buffer Empty
1	0	1	Ch A External / Status Change
1	1	0	Ch A Receive Character Avail.
1	1	1	Ch A Special Receive Condition

Bit 3 : Master Interrupt Enable

This bit is set to 1 to globally enable interrupts, and cleared to zero to disable interrupts. Clearing this bit to zero forces the IEO pin to follow the state of the IEI pin unless there is an IUS bit set in the SCC.

No IUS bit can be set after the MIE bit is cleared to zero. This bit is reset by a hardware reset.

Bit 2 : Disable Lower Chain

The Disable Lower Chain bit can be used by the CPU to control the interrupt daisy chain. Setting this bit to "1" forces the IEO pin Low, preventing lower-priority devices on the daisy chain from requesting interrupts. This bit is reset by a hardware reset.

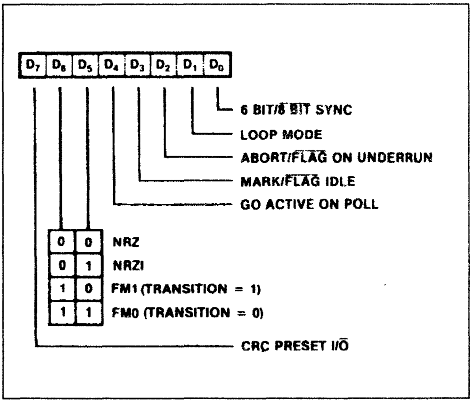
Bit 1 : No Vector

The No Vector bit controls whether or not the SCC will respond to an interrupt acknowledge cycle by placing a vector on the data bus if the SCC is the highest-priority device requesting an interrupt. If this bit is set, no vector is returned ; i.e., AD0-AD7 remain three-stated during an interrupt acknowledge cycle, even if the SCC is the highest-priority device requesting an interrupt.

Bit 0 : Vector Includes Status

The Vector Includes Status Bit controls whether or not then SCC will include status information in the vector it places on the bus in response to an interrupt acknowledge cycle. If this bit is set, the vector returned is variable, with the variable field depending on the highest-priority IP that is set. Table 11 shows the encoding of the status information. This bit is ignored if the No Vector (NV) bit is set.

Figure 52 : Write Register 10.



Write Register 10 (miscellaneous transmitter/receiver control bits). WR10 contains miscellaneous control bits for both the receiver and the transmitter. Bit positions for WR10 are shown in Figure 52.

Bit 7 : CRC Preset I/O

This bit specifies the initialized condition of the receive CRC checker and the transmit CRC generator. If this bit is set to "1", the CRC generator and checker are preset to "1". If this bit is set to "0", the CRC generator and checker are preset to "0".Either option can be selected with either CRC polynomial.

REGISTERS DESCRIPTION (cont'd)

In SDLC mode, the transmitted CRC is inverted before transmission and the received CRC is checked against the bit pattern "0001110100001111". This bit is reset by a channel or hardware reset. This bit is ignored in Asynchronous mode.

Bit 6 and 5 : Data Encoding 1 and 2

These bits control the coding method used for both the transmitter and the receiver, as illustrated in Table 12. All of the clocking options are available for all coding methods. The DPLL in the SCC is useful for recovering clocking information in NRZI and FM modes. A hardware reset forces NRZ mode. Timing for the various modes in shown in Figure 53.

Table 12 : Data Encoding

Data	Data Encoding	Encoding
0	0	NRZ
0	1	NRZI
1	0	FM1 (transition = 1)
1	1	FM0 (transition = 1)

Bit 4 : Go Active On Poll

When Loop mode is first selected during SDLC operation, the SCC connects RxD to TxD with only gate delays in the path. The SCC does not go on-loop and insert the 1-bit delay between RxD and TxD until this bit has been set and an EOP received. When the SCC is on-loop, the transmitter cannot go active unless this bit is set at the time an EOP is received. The SCC examines this bit whenever the transmitter is active in SDLC Loop mode and is sending a flag. If this bit is set at the time the flag is leaving the Transmit Shift register, another flag or data byte (if the transmit buffer is full) is transmitted. If the Go Active on Poll bit is not set at this time, the transmitter finishes sending the flag and reverts to the 1-Bit Delay mode. Thus, to transmit only one response frame, this bit should be reset after the first data byte is sent to the SCC but before CRC has been transmitted. If the bit is not reset before CRC is transmitted, extra flags are sent, slowing down response time on the loop. If this bit is reset before the first data is written, the SCC completes the transmission of the present flag and reverts to the 1-Bit Delay mode. After gaining control of the loop, the SCC is not able to transmit again until a flag and another EOP have been received. Though not strictly necessary, it is good practice to set this bit only upon receipt of a poll frame to ensure that the SCC does not go on loop without the CPU noticing it.

In synchronous modes other than SDLC with the Loop Mode bit set, this bit must be set before the transmitter can go active in response to a received sync character.

This bit is always ignored in Asynchronous mode and Synchronous modes unless the Loop Mode bit is set. This bit is reset by a channel or hardware reset.

Bit 3 : Mark/ $\overline{\text{Flag}}$ Idle

This bit affects only SDLC operation and is used to control the idle line condition. If this bit is set to "0", the transmitter sends flags an idle line. If this bit is set to "1", the transmitter sends continuous "1s" after the closing flag of a frame. The idle line condition is selected byte by byte ; i.e., either a flag or eight "1s" are transmitted. The primary station in an SDLC loop should be programmed for Mark Idle to create the EOP sequence . Mark Idle must be deselected at the beginning of a frame before the first data is written to the SCC, so that an opening flag can be transmitted. This bit is ignored in Loop mode, but the programmed value takes effect upon exiting the Loop mode. This bit is reset by a channel or hardware reset.

Bit 2 : Abort/$\overline{\text{Flag}}$ On Underrun

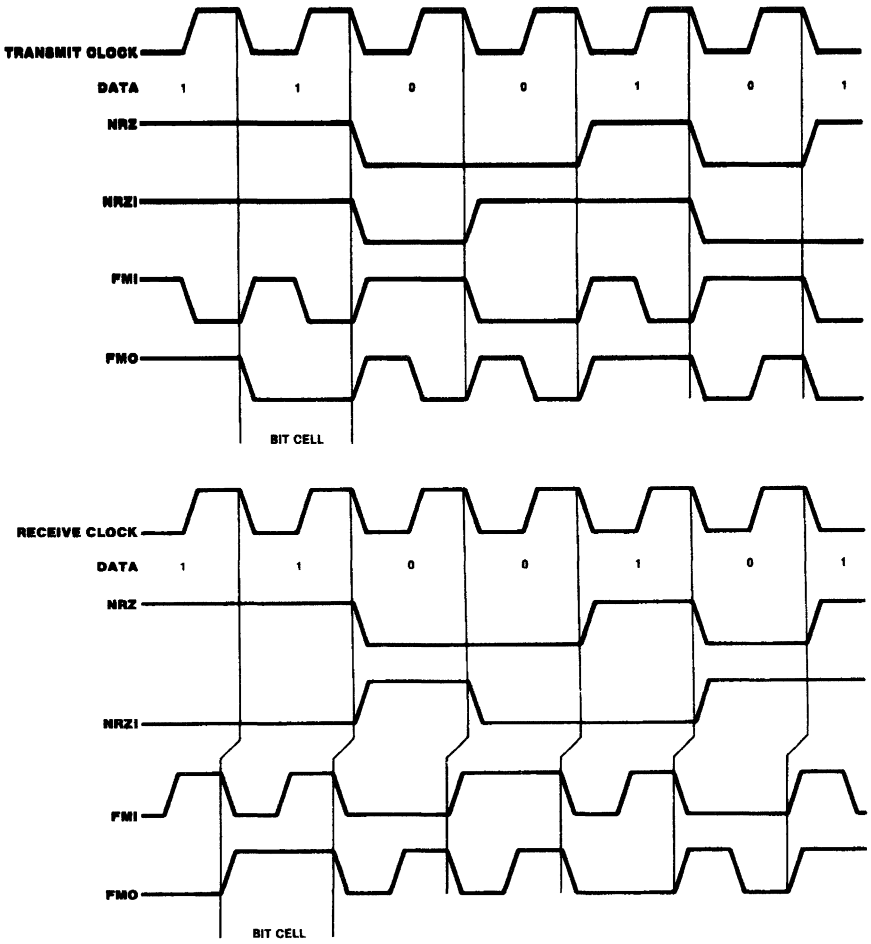
This bit affects only SDLC operation and is used to control how the SCC responds to a transmit underrun condition. If this bit is set to "1" and a transmit underrun occurs, the SCC sends an abort and a flag instead of CRC. If this bit is reset, the SCC sends CRC on a transmit underrun. At the beginning of this 16-bit transmission, the Transmit Underrun/EOM bit is set, causing an External/Status interrupt. The CPU uses this status, along with the byte count from memory or the DMA, to determine whether the frame must be retransmitted. A transmit buffer Empty interrupt occurs at the end of this 16-bit transmission to start the next frame. If both this bit and the Mark/Flag Idle bit are set to "1", all "1s" are transmitted after the transmit underrun. This bit should be set after the first byte of data is sent to the SCC and reset immediately after the last byte of data so that the frame will be terminated properly with CRC and a flag. This bit is ignored in Loop mode, but the programmed value is active upon exiting Loop mode. This bit is reset by a channel or hardware reset.

Bit 1 : Loop Mode

In SDLC mode, the initial set condition of this bit forces the SCC to connect TxD to TxD and to begin searching the incoming data stream so that it can go on loop. All bits pertinent to SDLC mode operation in other registers must be set before this mode

REGISTERS DESCRIPTION (cont'd)

Figure 53 : NRZ(NRZI)FM1(FM0) Timing.



REGISTERS DESCRIPTION (cont'd)

is selected. The transmitter and receiver should not be enabled until after this mode has been selected. As soon as the Go Active Onn Poll bit is set and an EOP is received the SCC goes on loop. If this bit is reset after the SCC is on loop, the SCC waits for the next EOP to go off loop.

In synchronous modes, the SCC uses this bit, along with the Go Active On Poll bit, to synchronize the transmitter to the receiver. The receiver should not be enabled until after this mode is selected. The TxD pin is held marking when this mode is selected unless a break condition is programmed. The receiver waits for a sync character to be received and then enables the transmitter on a character boundary. The break condition, if programmed, is removed. This mode works properly with sync characters of 6,8, or 16 bits. This bit is ignored in Asynchronous mode and is reset by a channel or hardware reset.

Bit 0 : 6 Bit/8 Bit SYNC

This bit is used to select a special case of synchronous modes. If this bit is set to "1" in Monosync mode, the receiver and transmitter sync characters are six bits long instead of the usual eight. If this bit is set to "1" in Bisync mode, the received sync will be 12 bits and the transmitter sync character will remain 16 bits long. This bit is ignored in SDLC and Asynchronous modes but still has effect in the special external sync modes. This bit is reset by a channel or hardware reset.

Write Register 11 (clock mode control). WR11 is the Clock Mode Control register. The bits in this register control the sources of the both the receive and transmit clocks, the type of signal on the SYNC and RTxC pins, and the direction of the TRxC pin. Bit positions for WR11 are shown in Figure 54.

Bit 7 : RTxC–XTAL/NO XTAL

This bit controls the type of input signal the SCC expects to see on the RTxC pin. If this bit is set to "0", the SCC expects a TTL-compatible signal as an input to this pin. If this bit is set to "1", the SCC connects a high-gain amplifier between the RTxC and SYNC pins in expectation of a quartz crystal being placed across the pins.

The output of this oscillator is available for use as a clocking source. In this mode of operation, the SYNC pin is unavailable for other use. The SYNC-signal is forced to "0" internally. A hardware reset forces NO XTAL. (At least 20 ms should be allowed after this bit is set to allow the oscillator to stabilize.)

Bits 6 and 5 : Receiver Clock 1 And 0

These bits determine the source of the receive clock as shown in Table 13. They do not interfere with any of the modes of operation in the SCC but simply control a multiplexer just before the internal receive clock input. A hardware reset forces the receive clock to come from the TRxC pin.

Table 13 : Receive Clock Source.

Receive Clock 1	Receive Clock 0	
0	0	Receive Clock = RTxC Pin
0	1	Receive Clock = TRxC Pin
1	0	Receive Clock = BR Output
1	1	Receive Clock = DPLL Output

Bits 4 and 3 : Transmit Clock 1 and 0

These bits determine the source of the transmit clock as shown in Table 14. They do not interfere with any of the modes of operation of the SCC but simply control a multiplexer just before the internal transmit clock input. The DPLL output that may be used to feed the transmitter in FM modes lags by 90 the output of the DPLL used by the receiver. This makes the received and transmitted bit cells occur simultaneously, neglecting delays. A hardware reset selects the TRxC pin as the source of the transmit clocks.

Table 14 : Transmit Clock Source.

Transmit Clock 1	Transmit Clock 0	
0	0	Transmit Clock = RTxC Pin
0	1	Transmit Clock = TRxC Pin
1	0	Transmit Clock = BR Output
1	1	Transmit Clock = DPLL Output

Bit 2 : TRxC O/I

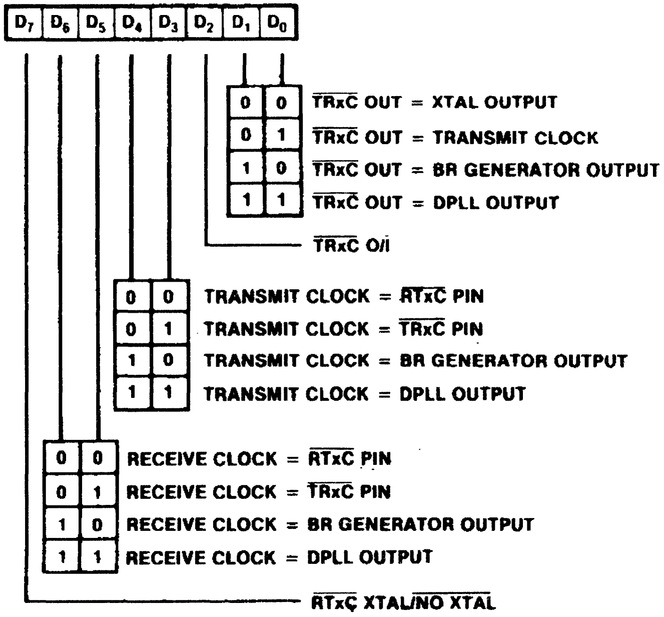
This bit determines the direction of the TRxC pin. If this bit is set to "1", the TRxC pin is an output and carries the signal selected by D1 and D0 of this register. However, if either the receive or the transmit clock is programmed to come from the TRxC pin, TRxC will be an input, regardless of the state of this bit. The TRxC pin is also an input if this bit is set to "0". A hardware reset forces this bit to "0".

Bits 1 and 0 : TRxC Output Source 1 And 0

These bits determine the signal to be echoed out of the SCC via the TRxC pin. No signal is produced if TRxC has been programmed as the source of either

REGISTERS DESCRIPTION (cont'd)

Figure 54 : Write Register 11.


D7 D6 D5 D4 D3 D2 D1 D0
0 0 TRxC OUT = XTAL OUTPUT
0 1 TRxC OUT = TRANSMIT CLOCK
1 0 TRxC OUT = BR GENERATOR OUTPUT
1 1 TRxC OUT = DPLL OUTPUT
TRxC O/I
0 0 TRANSMIT CLOCK = RTxC PIN
0 1 TRANSMIT CLOCK = TRxC PIN
1 0 TRANSMIT CLOCK = BR GENERATOR OUTPUT
1 1 TRANSMIT CLOCK = DPLL OUTPUT
0 0 RECEIVE CLOCK = RTxC PIN
0 1 RECEIVE CLOCK = TRxC PIN
1 0 RECEIVE CLOCK = BR GENERATOR OUTPUT
1 1 RECEIVE CLOCK = DPLL OUTPUT
RTxC XTAL/NO XTAL

the receive or the transmit clock. If TRxC O/I (bit 2) is set to "0", these bits are ignored.

If the XTAL oscillator output is programmed to be echoed, and the Xtal oscillator has not been enabled, the TRxC pin goes High. The DPLL signal that is echoed is the DPLL signal used by the receiver. Hardware reset selects the XTAL oscillator as the output source.

Table 15 : Transmit External Control Selection.

Output Signal	Output Signal	
0	0	TRxC = XTAL Oscillator Output
0	1	TRxC = Transmit Clock
1	0	TRxC = B R Output
1	1	TRxC = DPLL Output (Receive)

Write Register 12 (lower byte of baud rate generator time constant). WR12 contains the lower byte of the time constant for the baud rate generator. The time constant can be changed at any time, but the new value does not take effect until the next time the time constant is loaded into the down counter. No attempt is made to synchronize the loading of the time constant into WR12 and WR13 with the clock driving the down counter. For this reason, it is advisable to disable the baud rate generator while the new time constant is loaded into WR12 and WR13. Ordinarily, this is done anyway to prevent a load of the down counter between the writing of the upper and lower bytes of the time constant.

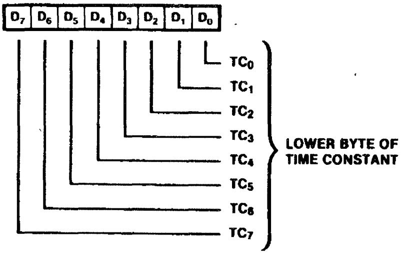
The formula for determining the appropriate time constant for a given baud is shown below with the desired rate in bits per second and the BR clock pe-

REGISTERS DESCRIPTION (cont'd)

riod in seconds. This formula is derived because the counter decrements from N down to "0"- plus-one-cycle for reloading the time constant and is then fed to a toggle flip-flop to make the output a square wave. Bit positions for WR12 are shown in Figure 55.

Time constant = [1/2 desired rate • BR clock period] - 2

Figure 55 : Write Register 12.



Write Register 13 (upper byte of baud rate generator time constant). WR13 contains the upper byte of the time constant for the baud rate generator. Bit positions for WR13 are shown in Figure 56.

Write Register 14 (miscellaneous control bits). WR14 contains some miscellaneous control bits. Bit positions for WR14 are shown in Figure 57.

Bits 7 and 5 : Digital Phase-Locked Loop Command Bits

These three bits encode the eight commands for the Digital Phase-Locked Loop. A channel or hardware reset disables the DPLL, resets the mission clock latches, sets the source to the RTxC pin and selects NRZI mode. The Enter Search Mode command enables the DPLL after a reset.

Null Command (000). This command has no effect on the DPLL.

Enter Search Mode (001). Issuing this command causes the DPLL to enter the Search mode, where the DPLL searches for a locking edge in the incoming data stream. The action taken by the DPLL upon receipt of this command depends on the operating mode of the DPLL.

Figure 56 : Write Register 13.

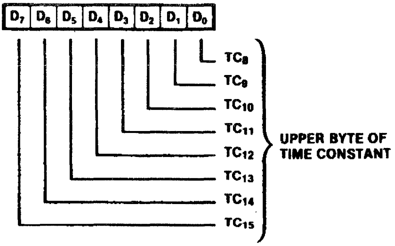
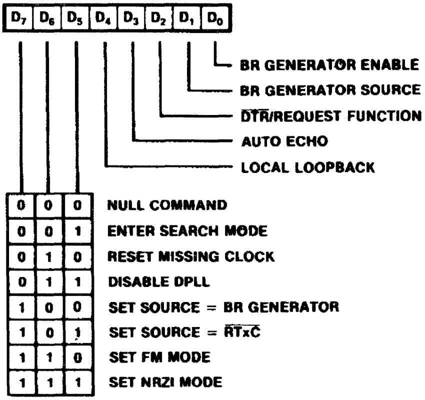


Figure 57 : Write Register 14.



REGISTERS DESCRIPTION (cont'd)

In NRZI mode, the output of the DPLL is High while the DPLL is waiting for an edge in the incoming data stream. After the Search mode is entered, the first edge the DPLL sees is assumed to be a valid data edge, and the DPLL begins the clock recovery operation from that point. The DPLL clock rate must be 32 times the data rate in NRZI mode. Upon leaving the Search mode, the first sampling edge of the DPLL occurs 16 of these 32X clocks after the first data edge and the second sampling edge occurs 48 of these 32X clocks after the first data edge. Beyond this point, the DPLL begins normal operation, adjusting the output to remain in sync with the incoming data.

In FM mode, the output of the DPLL is Low while the DPLL is waiting for an edge in the incoming data stream. The first edge the DPLL detects is assumed to be a valid clock edge. For this to be the case, the line must contain only clock edges ; i.e., with FM1 encoding, the line must be continuous "0s". With FM0 encoding the line must be continuous "1s", whereas Manchester encoding requires alternating "1s" and "0s" on the line. The DPLL clock rate must be 16 times the data rate in FM mode. The DPLL output causes the receiver to sample the data stream in the nominal center of the two halves of the bit cell to decide whether the data was a "1" or a "0". After this command is issued, as in NRZI mode, the DPLL starts sampling immediately after the first edge is detected. (In FM mode, the DPLL examines the clock edge of every other bit cell to decide what correction must be made to remain in sync). If the DPLL does not see an edge during the expected window, the one clock mission bit in RR10 is set. If the DPLL does not see an edge after two successive attempts, the two clocks missing bit in RR10 is set and the DPLL automatically enters the Search mode. This command resets both clock missing latches.

Reset Clock Missing (010). Issuing this command disables the DPLL, resets the clock missing latches in RR10, and forces a continuous Search mode state.

Disable DPLL (001). Issuing this command disables the DPLL, resets the clock missing latches in RR10, and forces a continuous Search mode state.

Set Source = BR Gen (100). Issuing this command forces the clock for the DPLL to come from the output of the baud rate generator.

Set Source = RTxC (101). Issuing the command forces the clock for the DPll to come from the RTxC pin or the crystal oscillator, de-pending on the state of the XTAL/no XTAL bit in WR11. This mode is selected by a channel or hardware reset.

Set FM Mode (110). This command forces the DPLL to operate in the FM mode and is used to recover the clock from FM or Manchester-encoded data. (Manchester is decoded by placing the receiver in NRZ mode while the DPLL is in FM mode.)

Set NRZI Mode (111). Issuing this command forces the DPLL to operate in the NRZI mode. This mode is also selected by a hardware or channel reset.

Bit 4 : Local Loopback

Setting this bit to "1" selects the local Loopback mode of operation. In this mode, the internal transmitted data is routed back to the receiver, as well as to the TxD pin. The CTS ans DCD inputs are ignored as enables in Local Loopback mode, even if auto enables is selected. (If so programmed, transitions on these inputs still cause interrupts.) This mode works with any Transmit/Receive mode ex-cept Loop mode. For meaning-ful results, the frequency of the transmit and receive clocks must be the same. This bit is reset by a channel or hardware reset.

Bit 3 : Auto Enable

Setting this bit to "1" selects the Auto Enable mode of operation. In this mode, the TxD pin is connected to RxD, as in Local Loopback mode, but the receiver still listens to the RxD input. Transmitted data is never seen inside or outside the SCC in this mode, and CTS is ignored as a transmit enable. This bit is reset by a channel or hardware reset.

Bit 2 : DTR/Request Function

This bit selects the function of the DTR/REQ pin follows the state of the DTR bit in WR5. If this bit is set to "1", the DTR/REQ pin follows the state of the DTR bit in WR5. If this bit is set to "1", the DTR/REQ pin goes Low whenever the transmit buffer becomes empty and in any of the synchronous mode when CRC has been sent at the end of a message. The request function on the DTR/REQ pin differs somewhat from the transmit request function available on the W/REQ pin in that REQUEST does not go inactive until the internal operation satisfying the request is complete, which occurs four to five PCLK cycles after the rising edge of DS, READ or WRITE. If the DMA used is edge-triggered, this difference is unimportant. This bit is reset by a channel or hardware reset.

Bit 1 : Baud Rate Generator Source

This bit selects the source of the clock for the baud rate generator. If this bit is set to "0", the baud rate

REGISTERS DESCRIPTION (cont'd)

generator clock comes from either the $\overline{\text{RTxC}}$ pin or the XTAL oscillator (depending on the state of the XTAL/no XTAL bit). If this bit is set to "1", the clock for the baud rate generator is the SCC's PCLK input. Hardware reset sets this bit to "0", selecting the $\overline{\text{RTxC}}$ pin as the clock source for the baud rate generator.

Bit 0 : Baud Rate Generator Enable

This bit controls the operation of the baud rate generator. The counter in the baud rate generator is enabled for counting when this bit is set to "1", and counting is inhibited when this bit is set to "0". When this bit is set to "1", change in the state of this bit is not reflected by the output of the baud rate generator for two counts of the counter. This allows the command to be synchronized. However, when set to "0", disabling is immediate. This bit is reset by a hardware reset.

Write Register 15 (external/status interrupt control). WR15 is the External/Status Source Control register. If the External/Status interrupts are enabled as a group via WR1, bits in this register control which External/Status conditions can cause an interrupt. Only the External/Status conditions that occur after the controlling bit are sent to "1" will cause an interrupt. This is true even if an External/Status condition is pending at the time the bit is set. Bit positions for WR15 are shown in Figure 58.

Bit 7 : Break/Abort IE

If this bit is set to "1", a change in the Break/Abort status of the receiver causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 6 : Tx Underrun/EOM

If this bit is set to "1", a change of state by the Tx Underrun/EOM latch in the transmitter causes an External/Status interrupt. This bit is set to "1" a channel or hardware reset.

Bit 5 : CTS IE

If this bit is set to "1", a change of state on the $\overline{\text{CTS}}$ pin causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 4 : SYNC/Hunt IE

If this bit is set to "1", a change of state on the $\overline{\text{SYNC}}$ pin causes an External/Status interrupt in Asynchronous mode, and a change of state in the Hunt bit in the receiver causes and External/Status interrupt in synchronous modes. This bit is set by a channel or hardware reset.

Bit 3 : DCD IE

If this bit is set to "1", a change of state on the $\overline{\text{DCD}}$ pin causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 2 : Not Used

Must be "0".

Bit 1 : Zero Count IE

If this bit is set to "1", an External/Status interrupt is generated whenever the counter in the baud rate generator reaches "0". This bit is set to "0" by a channel or hardware reset.

Bit 0 : Not Used

Must be "0".

READ REGISTERS

The Z8530 SCC contains seven read registers in each channel. In addition there are two registers which are shared by both channels. The status of these registers is continually changing and depends on the mode of communication, received and transmitted data, and the manner in which this data is transferred to and from the CPU. The following description details the bit assignments for each register.

Read Register 0 (transmit/ receiver buffer status and external status). Read Register 0 contains the status of the receive and transmit buffers. RR0 also contains the status bits for the six sources of External/Status interrupts. The bit configuration is illustrated in Figure 59.

Bit 7 : Break/Abort

In the Asynchronous mode, this bit is set when a Break sequence (null character plus framing error) is detected in the receive data stream. This bit is reset when the sequence is terminated, leaving a single null character in the receive FIFO. This character should be read and discarded. In SDLC mode, this bit is set by the detection of an Abort sequence (seven or more "1s"), then reset automatically at the termination of the Abort sequence. In either case, if the Break/Abort IE bit is set, an External/Status interrupt is initiated. Unlike the remainder of the External/Status bits, both transitions are guaranteed to

REGISTERS DESCRIPTION (cont'd)

cause an External/Status interrupt, even if another External/Status interrupt is pending at the time these transitions occur. This procedure is necessary because Abort or Break conditions may not persist.

Bit 6 : TX Underrun/EOM

This bit is set by a channel or hardware reset and when the transmitter is disabled or a Send Abort command is issued. This bit can only be reset by the reset Tx Underrrun/EOM Latch command in WR0. When the Transmit Underrun occurs, this bit is set and causes an External/Status interrupt (if the Tx Underrun/EOM IE bit is set).

Only the 0-to-1 transition of this bit causes an interrupt. This bit is always "1" in Asynchronous mode, unless a reset Tx Underrun/EOM Latch command has been erroneously issued. In this case, the Send Abort command can be used to set the bit to one and at the same time cause an External/Status interrupt.

Bit 5 : Clear to Send

If the CTS IE bit in WR15 is set, this bit indicates the state of the $\overline{CTS}$ pin the last time any of the enabled External/Status bits changed. Any transition on the $\overline{CTS}$ pin while no interrupt is pending latches the state of the $\overline{CTS}$ pin and generates an External/Status interrupt. Any odd number of transitions on the $\overline{CTS}$ pin while another External/Status interrupt is pending also causes an External/Status interrupts condition. If the CTS IE bit is reset, it merely reports the current unlatched state of the $\overline{CTS}$ pin.

Bit 4 : SYNC/Hunt

The operation of this bit is similar to that of the $\overline{CTS}$ bit, except that the condition monitored by the bit varies depending on the mode in which the SCC is operating.

When the XTAL oscillator option is selected in asynchronous modes, this bit is forced to "0" (no External/Status interrupt is generated). Selecting the XTAL oscillator in synchronous or SDLC modes had no effect on the operation of this bit.

The XTAL oscillator should not be selected in External Sync mode.

In Asynchronous mode, the operation of this bit is identical to that of the CTS status bit, except that this bit reports the state of the SYNC pin.

In External sync mode the $\overline{SYNC}$ pin is used by external logic to signal character synchronization. When the Enter Hunt Mode command is issued in External Sync mode, the $\overline{SYNC}$ pin must be held High by the external sync logic until character synchronization is achieved. A High on the $\overline{SYNC}$ pin holds the Sync/Hunt bit in the reset condition.

When external synchronization is achieved, $\overline{SYNC}$ must be driven Low on the second rising edge of the Receive Clock after the last rising edge of the Receive Clock on which the last bit of the receive character was received. Only $\overline{SYNC}$ is forced Low, it is good practice to keep it Low until the CPU informs the external sync logic that synchronization has been lost or that a new message is about to start. Both transitions on the $\overline{SYNC}$ pin cause External/Status interrupts if the Sync/Hunt IE bit is set to "1".

The Enter Hunt Mode command should be issued

Figure 58 : Write Register 15.

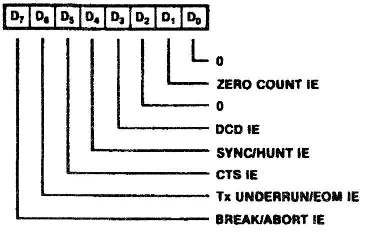
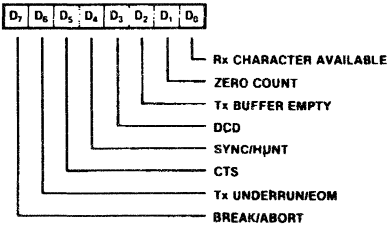


Figure 59 : Read Register 0.



REGISTERS DESCRIPTION (cont'd)

whenever character synchronization is lost. At the same time, the CPU should inform the external logic that character synchronization has been lost and that the SCC is waiting for SYNC to become active.

In the Monosync and Bisync Receive modes, the Sync/Hunt status bit is initially set to "1" by the Enter Hunt Mode command. The Sync/Hunt bit is reset when the SCC established character synchronization. Both transitions cause External/Status interrupts if the Sync/Hunt IE bit is set. When the CPU detects the end of message or the loss of character synchronization, the Enter Hunt Mode command should be issued to set the Sync/Hunt bit and cause an External/Status interrupt. In this mode, the SYNC pin is an output, which goes Low every time a sync pattern is detected in the data stream.

In the SDLC modes, the Sync/Hunt bit is initially set by the Enter Hunt Mode command or when the receiver is disabled. It is reset when the opening flag of the first frame is detected by the SCC. An External/Status interrupt is also generated if the Sync/Hunt IE bit is set. Unlike the Monosync and Bisync modes, once the Sync/Hunt bit is reset in SDLC mode, it does not need to be set when the end of the frame is detected. The SCC automatically maintains synchronization. The only way the Sync/Hunt bit can be set again is by the Enter Hunt Mode command or by disabling the receiver.

Bit 3 : Data Carrier Detect

If the DCD bit in WR15 is set, this bit indicates the state of the DCD pin the last time the Enabled External/Status bits changed. Any transition on the DCD pin while no interrupt is pending latches the state of the DCD pin, and generates an External/Status interrupt. Any odd number of transitions on the DCD pin while another External/Status interrupt is pending also causes an External/Status interrupt condition. If the DCD IE is reset, this bit merely reports the current, unlatched state of the DCD pin.

Bit 2 : TX Buffer Empty

This bit is set to "1" when the transmit buffer is empty. It is reset while CRC is sent in a synchronous or SDLC mode and while the transmit buffer is full. The bit is reset when a character is loaded into the transmit buffer. This bit is always in the set condition after a hardware or channel reset.

Bit 1 : Zero Count

If the Zero Count Interrupt Enable bit is set in WR15, this bit is set to one while the counter in the baud rate generator is at the count of zero. If there is no other External / Status interrupt condition pending at the time this bit is set, an External/Status interrupt is generated. However, if there is another External/Status interrupt pending at this time, no interrupt is initiated until interrupt service is complete. If the Zero Count condition does not persit beyond the end of the interrupt service routine, no interrupt will be generated. This bit is not latched High, even thought the other External/Status latches close as a result of the Low-to-High transition on ZC. The interrupt service routine should check the other External / Status conditions for changes. If none changed, ZC was the source. In polled applications, check the IP bit in RR3A for a status change and then proceed as in the interrupt service routine.

Bit 0 : RX Character Available

This bit is set to "1" when at least one character is available in the receive FIFO and is reset when the receive FIFO is complety empty. A channel or hardware reset empties the receive FIFO.

Read Register 1: RR1 contains the Special Receive Condition status bits and the residue codes for the I-field in SDLC mode. Figure 60 shows the bit positions for RR1.

Bit 7 : End of Frame (SDLC)

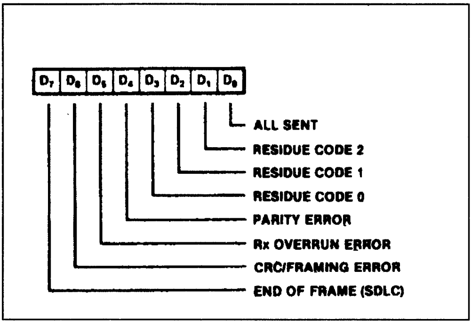
This bit is used only in SDLC mode and indicates that a valid closing flag has been received and that the CRC Error bit and residue codes are valid. This bit can be reset by issuing the Error Reset command. It is also updated by the first character of the following frame. This bit is reset in any mode other than SDLC.

Bit 6 : CRC/Framing Error

If a framing error occurs (in Asynchronous mode), this bit is set (and not latched) for the receive character in which the framing error occurred. Detection of a framing error adds an additional one-half bit to the character time so that the framing error is not interpreted as a new Start bit. In Synchronous and SDLC modes, this bit indicates the result of comparing the CRC checker to the apppropriate check value. This bit is reset by issuing an Error Reset command, but the bit is never latched. Therefore, it is always updated when the next character is received. When used for CRC error status in Synchronous or SDLC modes, this bit is usually set since most bit combination, except for a correctly completed message, result in a non-zero CRC.

REGISTERS DESCRIPTION (cont'd)

Figure 60 : Read Register 1.



Bit 5 : Receiver Overrun Error

This bit indicates that the receive FIFO has overflowed. Only the character that has been written over is flagged with this error, and when the character is read, the Error condition is latched until reset by the Error Reset command. The overrun character and all subsequent characters received until the Error Reset command is issued causes a Special Receive Condition vector to be returned.

Bit 4 : Parity Error

When parity is enabled, this bit is set for the characters whose parity does not match the programmed sense (even/odd). This bit is latched so that once an error occurs, it remains set until the Error Reset command is issued. If the parity in Special Condition bit is set, a parity error causes a Special Receive Condition vector to be returned on the character containing the error and on all subsequent characters until the Error Reset command is issued.

Bit 3,2, and 1 : Residue Codes 2,1, And 0

In those cases in SDLC mode where the received I-Field is not an integral multiple of the character length, these three bits indicate the length of the I-Field and are meaningful only for the transfer in which the end of frame bit is set. This field is set to "011" by a channel or hardware reset and is forced to this state in Asynchronous mode. These three bits can leave this state only if SDLC is selected and a character is received. The codes signify the following (Reference Table 16 when a receive character length is eight bits per character).

I-Field bits are right-justified in all cases. If a receive character length other than eight bits is used for the I-Field, a table similar to Table 16 can be constructed for each different character length. Table 17 shows the residue codes for no residue (The I-Field boundary lies on a character boundary).

Table 16 : I-Field Bit Selection (8 Bits only).

Residue	Residue	Residue	I-Field Bits in Last Byte	I-Field Bits in Previous Byte
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

Bit 0 : All Sent

In Asynchronous mode, this bit is set when all characters have completely cleared the transmitter pins. Most modems contain additional delays in the data path, which requires the modem control signals to remain active until after the data has cleared both the transmitter and the modem. This bit is always set in synchronous and SDLC modes.

Table 17 : Residue Bits/Character.

Bits / Char	Residue	Residue	Residue
8	0	1	1
7	0	0	0
6	0	1	0
5	0	0	1

Read Register 2. RR2 contains the interrupt vector written into WR2. When the register is accessed in Channel A, the vector returned is the vector actually stored in WR2. When this register is accessed in Channel B, the vector returned includes status information in bits 1, 2, and 3, or in bits 6, 5, and 4, depending on the state of the Status High/Status Low bit in WR9 and independent of the state of the VIS bit in WR9. The vector is modified according to Table 11 shown in the explanation of the VIS bit in WR9. If no interrupts are pending, the status is V3, V2, V1 = 011, or V6, V5, V4 = 110. Figure 61 shows the bit positions for RR2.

Read Register 3. RR3 is the Interrupts Pending register. The status of each of the Interrupt Pending bits in the SCC is reported in this register. This register exists only in Channel A. If this register is accessed in Channel B, all "0s" are returned. The two unused bits are always returned as "0". Figure 62 shows the bit positions for RR3.

REGISTERS DESCRIPTION (cont'd)

Read Register 8. RR8 is the Receive Data register.

Read Register 10. RR10 contains some miscellaneous status bits. Unused bits are always "0". Bit positions for RR10 are shown in Figure 63.

Bit 7 : One Clock Missing

While operating in the FM mode, the DPLL sets this bit to "1" when it does not see a clock edge on the incoming lines in the window where it expects one. This bit is latched until reset by a Reset Missing Clock or Enter Search Mode command in WR14. In the NRZI mode of operation and while the DPLL is disabled, this bit is always "0".

Bit 6 : Two Clocks Missing

While operating in the FM mode, the DPLL sets this bit to "1" when it does not see a clock edge in two successive tries. At the same time the DPLL enters the Search mode. This bit is latched until reset by a Reset Missing Clock or Enter Search Mode command in WR10. In the NRZI mode of operation and while the DPLL is disabled, this bit is always "0".

Bit 4 : Loop Sending

This bit is set to "1" in SDLC Loop mode while the transmitter is in control of the Loop, that is, while the SCC is actively transmitting on the loop. This bit is reset at all other times.

This bit can be polled in SDLC mode to determin when the closing flag has been sent.

Bit 1 : On Loop

This bit is set to "1" while the SCC is actually onloop in SDLC Loop mode. This bit is set to "1" in the X.21 mode (Loop mode selected while in monosync) when the transmitter goes active. This bit is "0" at all other times. This bit can also be pulled in SDLC mode to determine when the closing flag hass been sent.

Read Register 12. RR12 returns the value stored in WR12, the lower byte of the time constant for the baud rate generator. Figure 64 shows the bit positions for RR12

Read Register 13. RR13 returns the value stored in WR13, the upper byte of the time constant for the baud rate generator. Figure 65 shows the bit positions for RR13.

Read Register 15. RR15 reflects the value stored in WR15, the External/Status IE bits. The two unused bits are always returned as "0s". Figure 66 shows the bits positions for RR15.

Figure 61 : Write Register 2.

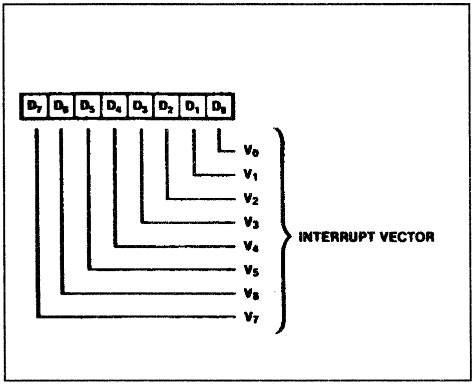
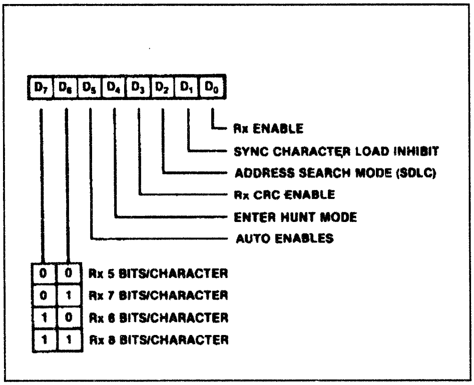


Figure 62 : Write Register 3.



REGISTERS DESCRIPTION (cont'd)

Figure 63 : Write Register 10.

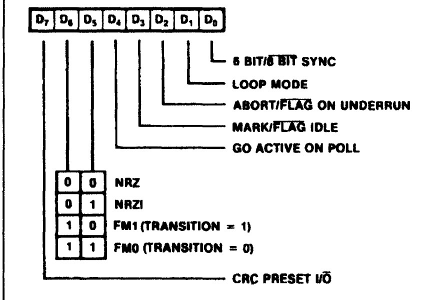


Figure 65 : Write Register 13.

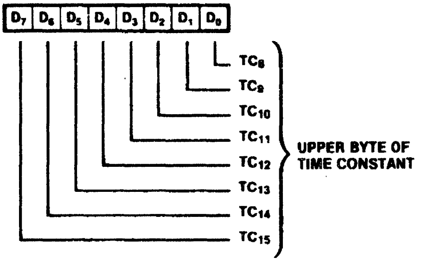


Figure 64 : Write Register 12.

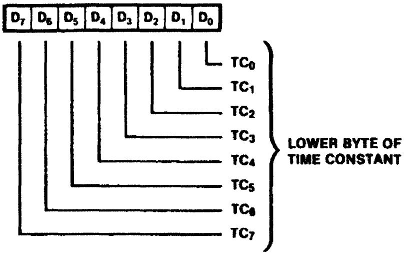
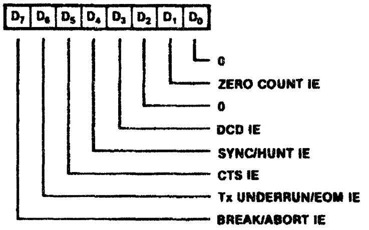


Figure 66 : Write Register 15.



SCC INITIALIZATION WORKSHEET

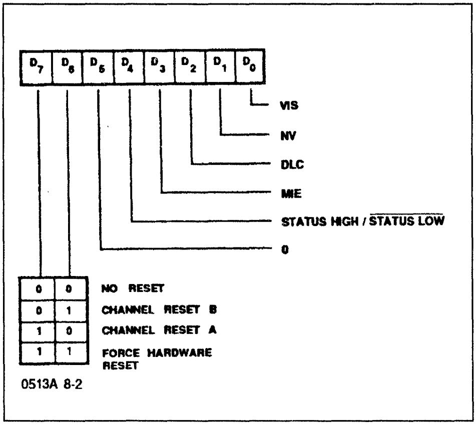
This section describes the software initialization procedure for the Serial Communications Controller (SCC).

Figure 68 provides a worksheet that can be used as an aid when initializing the SCC. Since all SCC operation modes are initialized in a similar manner, the worksheet can be used to tailor the SCC device to the user's individual need. Specific examples are given in the following sections.

Register Overview

Each of the SCC's two channels has its own separate Write registers that are programmed to initialize different operating modes. There are two types of bits in the Write registers : Command bits and Mode bits. An example of a register that contains both types of bits is Write Register 9 (WR9), and is shown in figure 67.

Figure 67 : Write Register 2.



WR9 is the Master Interrupt Control register and contains the Reset command bits. Command bits are denoted by having boxes drawn around them in register diagrams. Bit D5 in this register is not used in this register and must be 0 at all times.

The Command bits, D7 and D6, select one of the reset commands for the SCC. Setting either of these bits to 1 disables both the receiver and the transmitter in the corresponding channel, forces TxD for the channel marking, forces the modem control signals High in that channel, resets all IPs and IUSs, and disables all interrupts in that channel. Functions controlled by the Command bits can only be enabled or disabled, they cannot be toggled.

Bits D4-D0 are Mode bits that can be enabled or disabled either by being set to "1" or reset to "0". Each Mode bit affects only one function. For example, Bit D1 is the No Vector mode bit, it controls whether or not the SCC will respond to an interrupt acknowledge cycle by placing a vector on the data bus. If this bit is set, no vector is returned. In command bits entry, each new command requires a separate rewrite of the entire register. Care must be taken when issuing a command, so that the Mode bits are not changed accidentally.

Initialization Procedure

The SCC initialization procedure is divided into three parts. The first part consists of programming the operation modes (e.g. bits-per-character, parity) and loading the constants (e.g., interrupt vector, time constants). The second part enables the hardware functions (e.g., transmitter, receiver, baud-rate generator). It is important that the operating modes are programmed before the hardware functions are enabled. The third part, if required, consists of enabling the different interrupts.

Table 18 shows the order (from top to bottom) in which the SCC registers are to be programmed. Those registers that need not be programmed are listed as optional in the comments column. The bits in the registers that are marked with an "X" are to be programmed by the user. The bits marked with an "S" are to be set to their previous programmed value. For example, in part 2, Write Register 3, bits D1-D7 are shown with an "S" because they have been programmed in part 1 and must remain set to the same value.

Initialization Table Generation

Figure 68 is a worksheet for the initialization of the SCC. All the bits that must be programmed as either a "0" or a "1" are already filled in ; the remaining bits are left blank and are to be programmed by the user according to the desired mode of operation. The binary value can then be converted to a hexadecimal number and placed in the table, following the Write register notation in the column labeled "HEX". A Program Initialization Table is produced when this worksheet is completed.

SCC INITIALIZATION WORKSHEET (cont'd)

Figure 68 : SCC Initialization Worksheet.

Label of SCC Table: ______________________ SCC Base Address ______________________

Description: ____

	REGISTER	HEX		BINARY 7	6	5	4	3	2	1	0	COMMENTS
MODES	WR9	C	0	1	1	0	0	0	0	0	0	SOFTWARE RESET
	WR0	0		0	0	0	0	0	0			
	WR4											
	WR1			0			0	0		0	0	
	WR2											
	WR3										0	
	WR5							0				
	WR6											
	WR7											
	WR9			0	0	0		0				
	WR10											
	WR11											
	WR12											
	WR13											
	WR14										0	
	WR14										0	
ENABLES	WR14			0	0	0					1	
	WR3										1	
	WR5							1				
	WR0	8	0	1	0	0	0	0	0	1	0	RESET TxCRC
	WR1											
INTERRUPT	WR15											
	WR0	1	0	0	0	0	1	0	0	0	0	RESET Ext/STATUS
	WR0	1	0	0	0	0	1	0	0	0	0	RESET Ext/STATUS
	WR1											
	WR9			0	0	0						

SCC INITIALIZATION WORKSHEET (cont'd)

Table 18 : SCC Initialization Order.

Part 1. Modes and Constants		
WR9	1100000	Hardware Reset
WR0	000000XX	Select Shift Mode (8030 only)
WR4	XXXXXXXX	Tx/Rx Con, Async or Sync Mode
WR1	0XX00X00	Select W/REQ (opt)
WR2	XXXXXXXX	Program Interrupt Vector (opt)
WR3	XXXXXXX0	Select Rx Control
WR5	XXXX0XXX	Selects Tx Control
WR6	XXXXXXXX	Program Sync Character (opt)
WR7	XXXXXXXX	Program Sync Character (opt)
WR9	000X0XXX	Select Interrupt Control
WR10	XXXXXXXX	Miscellaneous Control (opt)
WR11	XXXXXXXX	Clock Control
WR12	XXXXXXXX	Time Constant Lower Byte (opt)
WR13	XXXXXXXX	Time Constant Upper Byte (opt)
WR14	XXXXXXX0	Miscellaneous Control
WR14	XXXSSSSS	Commands (opt)

Part 2. Enables		
WR14	000SSSS1	Baud Rate Enable
WR3	SSSSSSS1	Rx Enable
WR5	SSSS1SSS	Tx Enable
WR0	10000000	Reset Tx CRG (opt)
WR1	XSS00S00	DMA Enable (opt)
Part 3. Interrupt Status		
WR15	XXXXXXXX	Enable External/status
WR0	00010000	Reset External Status
WR0	00010000	Reset External Status Twice
WR1	SSSXXSXX	Enable Rx, Tx and Ext/status
WR9	000SXSSS	Enable Master Interrupt Enable
1 = Set to one 0 = Reset to zero		X = User defined S = Same as previously prog.

Reset Conditions

Prior the initialization, the SCC should be reset by either hardware or software. A hardware reset can be accomplished by simultaneously grounding. A software reset can be executed by writing a 0H to Write Register 9.

POLLED ASYNCHRONOUS MODE

This section describes the use of the SCC in polled Asynchronous Mode. The device can be set with 5 to 8 bits per character, 1, 1.5, or 2 stop bits, and a wide range of baud rates. In this particular example, 8 bits per character, 2 stop bits and 9600 baud rate are used. An external 2.4576MHz, crystal oscillator is used for baud-rate generation. The SCC can be programmed for local loopback for on-board diagnostics. The user can make use of this feature to test-program the part without additional hardware to simiulate an actual transmit and receive environment.

SCC Interface

Figure 69 shows the SCC to CPU interface required for this application. The 8-bit data bus and control lines all come from the user's CPU. The 8530 control lines are $\overline{\text{RD}}$, $\overline{\text{WR}}$, A/$\overline{\text{B}}$, D/$\overline{\text{C}}$ and $\overline{\text{CE}}$. PCLK comes from the system clock, or an external crystal, up to the maximum rate of the SCC (ex. 6MHz for the Z8530A). The IEI and the $\overline{\text{INTACK}}$ pins should be pulled up. The baud-rate generator clock is connected to the $\overline{\text{RTxC}}$ pin.

SCC Initialization

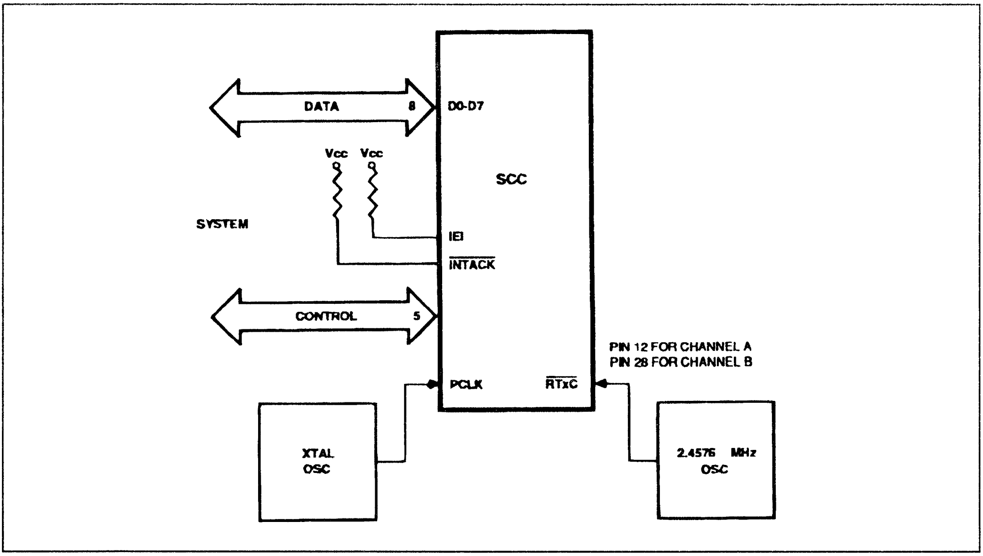
Initialization of the SCC for polled asynchronous communication is divided into two parts ; part one programs the operating modes of the SCC and part two enables them. Care must be taken when writing the software to meet the SCC's Cycle and Reset Recovery times. The Cycle Recovery time, 6 PCLK cycles, applies to the period between any Read or Write cycles affecting the SCC. The Reset Recovery time is the period after a hardware reset caused either by hardware or software ; this recovery time extends the Cycle Recovery time to 11 PCLK cycles. For more details about these recovery times, see the section Interfacing the SCC.

Table 19 : Polled Asynchronous Initialization Procedure.

Register	Value	Comments
WR9	C0H	Force Hardware Reset
WR4	4CH	x16 Clock, 2 Stop Bits, no Parity
WR3	C0H	Rx8 Bits, Rx Disabled
WR5	60H	Tx8 Bits, DTR, RTS, Tx off
WR9	00H	Int. Disabled
WR10	00H	NRZ
WR11	56H	Tx & Rx = BRG out, TRxC = BRG out
WR12	06H	Time Constant = 6
WR13	00H	Time Constant High = 0
WR14	10H	BRG in = RTxC, BRG off, Loopback
Enables		
WR14	11H	BRG Enable
WR3	C1H	Rx Enable
WR5	68H	Tx Enable

POLLED ASYNCHRONOUS MODE (cont'd)

Figure 69.



SCC Operating Mode Programming

WR9 resets the SCC to a known state by writing a C0 hex. The command, Force Hardware Reset, is identical to a hardware reset.

WR4 selects the asynchronous, x 16 mode, with 2 stop bits and no parity. The x 16 mode means that clock rate is 16 times the data rate.

WR3 selects 8 bits per character and does not enable the receiver. The 8 bits per character allows 8 bits to be assembled from the data stream. The receiver is not enabled at this time because the SCC has not been initialized.

WR5 selects 8 bits per character and does not enable the transmitter. The 8 bits per character allows 8 bits to be sent, as data, with the least significant bit first. The transmitter is not enabled at this time because the SCC has not been initialized.

WR9 selects that there are no interrupts enabled. This inhibits the SCC from requesting an interrupt from the CPU.

WR10 selects NRZ encoding. This NRZ coding is used on the transmitter as well as the receiver.

WR11 selects the $\overline{\text{RTxC}}$ pin to TTL clock ; the baud-rate generator is the transmit and receive clocks source, and the TRxC pin as a baud-rate generator output.

WR12 & WR13 and select the baud-rate generator's time constant. The WR13 time constant is determined by the equation :

$$\text{Time Constant} = \frac{\text{Clock Frequency}}{2 \text{ x Baud Rate x clock mode}} - 2$$

In this example, the clock frequency is 2.4576MHz, the baud rate is 9600, and the clock mode is 16, the time constant is, therefore, 6 ; expressed as a 16-bit, hexadecimal number, it is 0006H. The time constant LOW (WR12) is, therefore, 06H and the time constant HIGH (WR13) is 00H. The baud rate for this example can be varied, as long as the data rate is less than 1/4 of the PCLK rate.

Table 20 shows the time constants for other common baud rates.

Table 20 : Time Constants for Common Baud Rates.

Baud Rate	Divider	
	Decimal	Hex
38400	0	0000H
19200	2	0002H
9600	6	0006H
4800	14	000EH
2400	30	001EH
1200	62	003EH
600	126	007EH
300	254	00FEH
150	510	01FEH
For 2.4576 MHz Clock, X 16 Mode		

POLLED ASYNCHRONOUS MODE (cont'd)

WR14 selects the baud-rate generator as the $\overline{\text{RTxC}}$ pin, baud-rate generator disabled, and internal loopback. The baud-rate generator uses the $\overline{\text{RTxC}}$ pin as the clock source and is not enabled at this time because the SCC initialization is not complete.

SCC Operating Mode Enables

WR14 enables the baud-rate generator. Bit 0 (LSB) is changed to a 1 to enable the baud-rate generator; all other bits must maintain the value selected during initialization.

WR3 enables the receiver. Bit 0 (LSB) is changed to a 1 to enable the receiver, all other bits must maintain the value selected during initialization.

WR5 enables the transmitter. Bit 3 is changed to a 1 to enable the transmitter, all other bits must maintain the value selected during initialization.

Transmit and Receive Routines

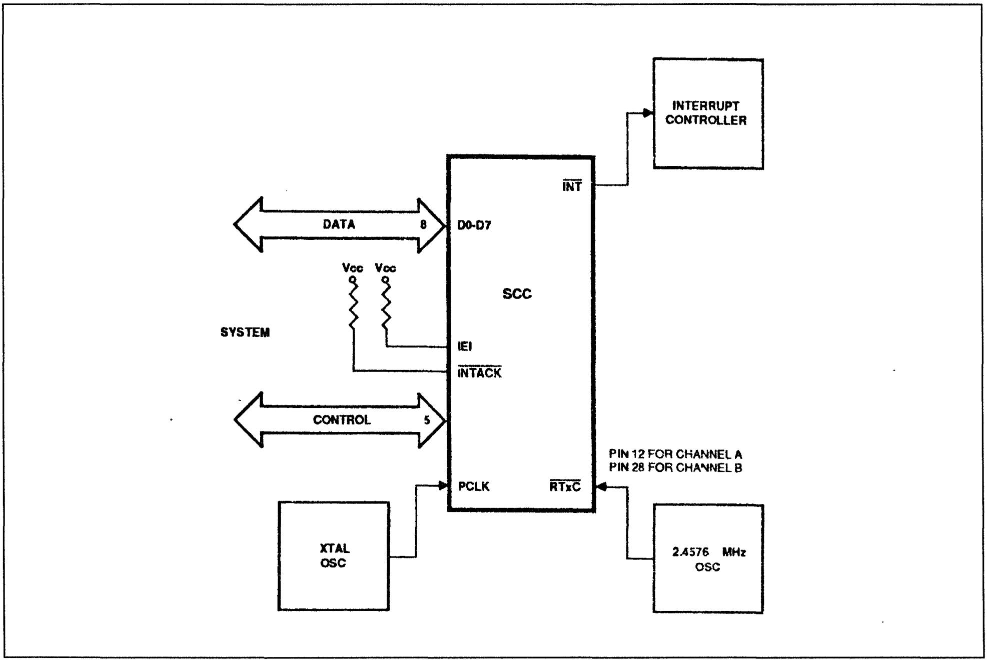
After initialization, and after all enables have been selected, the SCC is ready for communication.

The transmitter buffer and the receive FIFO are empty. The example shown below is coded to transmit and receive characters.

Figure 70 : Transmit and Receive Routine.

```
; Transmit a character
TXCHAR : INPUT   RRO      ;Read RRO
         Test    Bit 2    ;Test transmit
                          buffer empty
         JZ      TXCHAR   ;Loop if not
                          empty
         OUTPUT  CHAR     ;Output chara
                          cter to data port
         RET              Return
; Receive a character
RXCHAR : INPUT   RRO      ;Read RRO
         TEST    BIT 0    ;Test Receive
                          buffer
         JZ      RXCHAR   ;Loop if not
                           full
         INPUT   CHAR     ;Input charac
                          ter from data
                           port
         RET              ; Return
```

Figure 71.



INTERRUPT WITHOUT INTACK ASYNCHRONOUS MODE

This section describes the use of the SCC for interrupt-driven Asynchronous Mode. As with the example in the previous chapter, the SCC is set with 8 bits per character, 2 stop bits, at 9600 baud rate. An external 2.4576MHz, crystal oscillator is used for baud-rate generation. Interrupt acknowledge is not generated because of the extra hardware required to produce this signal. In this chapter, the SCC is also programmed for local loopback so that no external loop between the transmit and the receive data lines is needed for on-board diagnostics. This feature allows the user to test-program the part without additional hardware to simiulate an actual transmit and receive environment.

SCC Interface

Figure 71 shows the SCC to CPU interface required for this application. The 8-bit data bus and control lines all come from the user's CPU. For the 8530, the control lines are $\overline{RD}$, $\overline{WR}$, A/$\overline{B}$, D/$\overline{C}$ and $\overline{CE}$. The $\overline{INT}$ signal goes to an interrupt controller which must produce the interrupt vector to the CPU. The PCLK comes from the system clock, or an external crystal oscillator, up to the maximum rate of the SCC (ex. 6MHz for the Z8530A). The IEI and the $\overline{INTACK}$ pins should be pulled up. The baud-rate generator clock is connected to the $\overline{RTxC}$ pin.

SCC Initialization

The initialization of the SCC for interrupt-driven asynchronous communication is divided into three parts. Part one programs the operating modes of the SCC, part two and three enable them. Care must be taken when writing the code to meet the SCC's Cycle and Reset Recovery times. The Cycle Recovery time applies to the period between any Read or Write cycles to the SCC, and is 6PCLK cycles. The Reset Recovery time applies to a hardware reset caused either by hardware or software ; this recovery time extends the Cycle Recovery time to 11PCLK cycles. More details about these recovery times can be found in the section Interfacing the SCC.

Table 21.

Register	Value	Comments
WR9	C0H	Force Hardware Reset
WR4	4CH	x16 Clock, 2 Stop Bits, no Parity
WR2	00H	Interrupt Vector 00WR3
	C0H	Rx8 Bits, Rx Disabled
WR5	60H	Tx8 Bits, DTR, RTS, Tx off
WR9	00H	Int. Disabled WR10
	00H	NRZ
WR11	56H	Tx & Rx = BRG out, TRxC = BRG out
WR12	06H	Time Constant = 6
WR13	00H	Time Constant High = 0
WR14	10H	BRG in = RTxC, BRG off, Loopback
Enables		
WR14	11H	BRG Enable
WR3	C1H	Rx Enable
WR5	68H	Tx Enable
Enable Interrupts		
WR1	12H	Rx Int on All Char and Tx Int Enables
WR9	08H	MIE

SCC Operating Modes Programming

WR9 resets the SCC to a known state by writing a C0 hex. This command, Force Hardware Reset, is identical to a hardware reset.

WR4 selects asynchronous mode, x16 mode, 2 stop bits and no parity. The x16 mode means that the clock rate is 16 times the data rate.

WR2 is the interrupt vector of the SCC. Even though a vector is not placed on the bus in this mode the vector including status in read from RR2. By writing 00H to this register the status read will be the only bits set in RR2.

WR3 selects 8 bits per character and does not enable the receiver. The 8 bits per character allows 8 bits to be assembled from the data stream. The receiver is not enabled at this time because the SCC is not completely initialized.

WR5 selects 8 bits per character and does not

INTERRUPT WITHOUT INTACK ASYNCHRONOUS MODE (cont'd)

enable the transmitter. The 8 bits per character allows 8 bits to be sent as data with the least significant bit first. The transmitter is not enabled at this time because the SCC is not completely initialized.

WR9 selects that there are no interrupts enabled. This will inhibit the SCC from requesting an interrupt from the CPU.

WR10 selects NRZ encoding. This selects NRZ coding is to be used on the transmitter and the receiver.

WR11 selects the RTxC pin to TTL clock, the transmit and receive clocks source as the baud-rate generator and the TRxC pin as a baud-rate generator output.

WR12 & WR13 select the baud-rate generators time constant. The time constant is determined by the equation :

$$\text{Time Constant} = \frac{\text{Clock Frequency}}{2 \times \text{Baud Rate} \times \text{clock mode}} - 2$$

In this example, the clock frequency is 2.4576MHz, the baud rate is 9600, and the clock mode is 16 ; the time constant is 6. Converting this time constant to a 16-bit hexadecimal number, it becomes 0006H. The time constant LOW (WR12) is 06H and the time constant HIGH (WR13) is 00H. The baud rate for this example can be varied for as long as the data rate is less than 1/4 of the PCLK rate. Table 22 gives the time constants for other common baud rates.

Table 22 : Time Constants for Common Baud Rates.

Baud Rate	Divider	
	Decimal	Hex
38400	0	0000H
19200	2	0002H
9600	6	0006H
4800	14	000EH
2400	30	001EH
1200	62	003EH
600	126	007EH
300	254	00FEH
150	510	01FEH
For 2.4576 MHz Clock, X 16 Mode		

WR14 selects the baud rate source as the RTxC pin, baud rate generator disabled, and internal loopback. The baud-rate generator will use the RTxC pin as the clock source for the baud-rate generator. The baud-rate generator is not enabled at this time because the SCC initialization is not complete.

SCC Operating Mode Enables

WR14 enables the baud-rate generator. Bit 0 (LSB) is changed to a 1 to enable the baud-rate generator ; all other bits must maintain the value selected during initialization.

WR3 enables the receiver. Bit 0 (LSB) is changed to a 1 to enable the receiver ; all other bits must maintain the value selected during initialization.

WR5 enables the transmitter. Bit 3 is changed to a 1 to enable the transmitter ; all other bits must maintain the value selected during initialization.

SCC Operating Mode Interrupts

WR1 enables the Tx and the Rx interrupts. The Rx interrupt is programmed to generate an interrupt an all received characters or special conditions. This provides an interrupt on every character received by the SCC. The external/status interrupts are not enabled in this application.

WR9 sets the master interrupt enable (MIE) bit 3. Seting this bit enables the interrupts pending to generate and interupt on the INT pin.

Interrupt Routine

When the SCC has been initialized and enabled, it is ready for communication. The transmitter buffer and the receive FIFO are both empty. An interrupt will not be generated until the software writes the first character to the transmit buffer. Once the first character is in the SCC shift register, the first transmit interrupt will occur. The SCC will then keep setting transmit and receive interrupts to the interrupt controller until the end of the message. At the end of the message, a Reset Transmitter Interrupt Pending (WR0) is issued to clear the transmit interrupt. After the last character is read into the SCC, the interrupts will cease until another message is written into the transmitter.

Once an interrupt is received and the interrupt controller vectors to the interrupt routine, RR2 is read from channel B. The value read from RR2 is the vector, including status. This vector shows the status of the highest priority interrupt pending (IP) at the time it is read. Once the highest priority interrupt condition is cleared, RR2 will show the status of the next highest interrupt pending, if one is present. This allows multiple interrupts to be serviced without the overhead of the interrupt acknowledge cycle of the interrupt controller.

The following example shows how the interrupt routine should be coded.

HARDWARE INFORMATION

Absolute Maximum Ratings

Parameter	Test Conditions	Unit
Operating Temperature Z8530 B1/D1/C1 B6/D6/C6	0 to + 70 – 40 to + 85	°C °C
Storage Temperature	– 65 to + 150	°C
Voltages on any Pin with Respect to GND	– 0.3 to + 7	V
Total Power Dissipation	700	mW
Oscillator Frequency Z8530 Z8530A Z8530B	0 to 4 0 to 6 0 to 8	MHz

Note : Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of these specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Test Conditions

The characteristics below apply for the following standard test conditions, unless otherwise noted. All voltages are referenced to GND. Positive current flows into the reference pin.

Standard conditions are as follows :

- $+4.75\ V \leq V_{CC} \leq +5.25\ V$
- GND = 0 V
- $0\ °C \leq T_A \leq +70\ °C$

All ac parameters assume a load capacitance of 50 pf max.

Figure 73 : Standard Test Load.

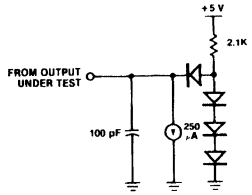
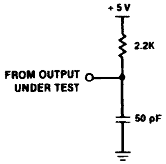


Figure 74 : Open-Drain Test Load.



DC Characteristics

Symbol	Parameter	Test Conditions	Min.	Max.	Unit
V_{IH}	Input High Voltage		2.0	V_{CC}+0.3	V
V_{IL}	Input Low Voltage		– 0.3	0.8	V
V_{OH}	Output High Voltage	I_{OH} = – 250 µA	2.4		V
V_{OL}	Output Low Voltage	I_{OL} = + 2.0 mA		0.4	V
I_{IL}	Input Leakage	$0.4 \leq V_{IN} \leq +2.4$ V		± 10.0	µA
I_{OL}	Output Leakage	$0.4 \leq V_{OUT} \leq +2.4$ V		± 10.0	µA
I_{CC}	V_{CC} Supply Current		250		mA

V_{CC} = 5 V ± 5 % unless otherwise specified, over specified temperature range.

HARDWARE INFORMATION (cont'd)

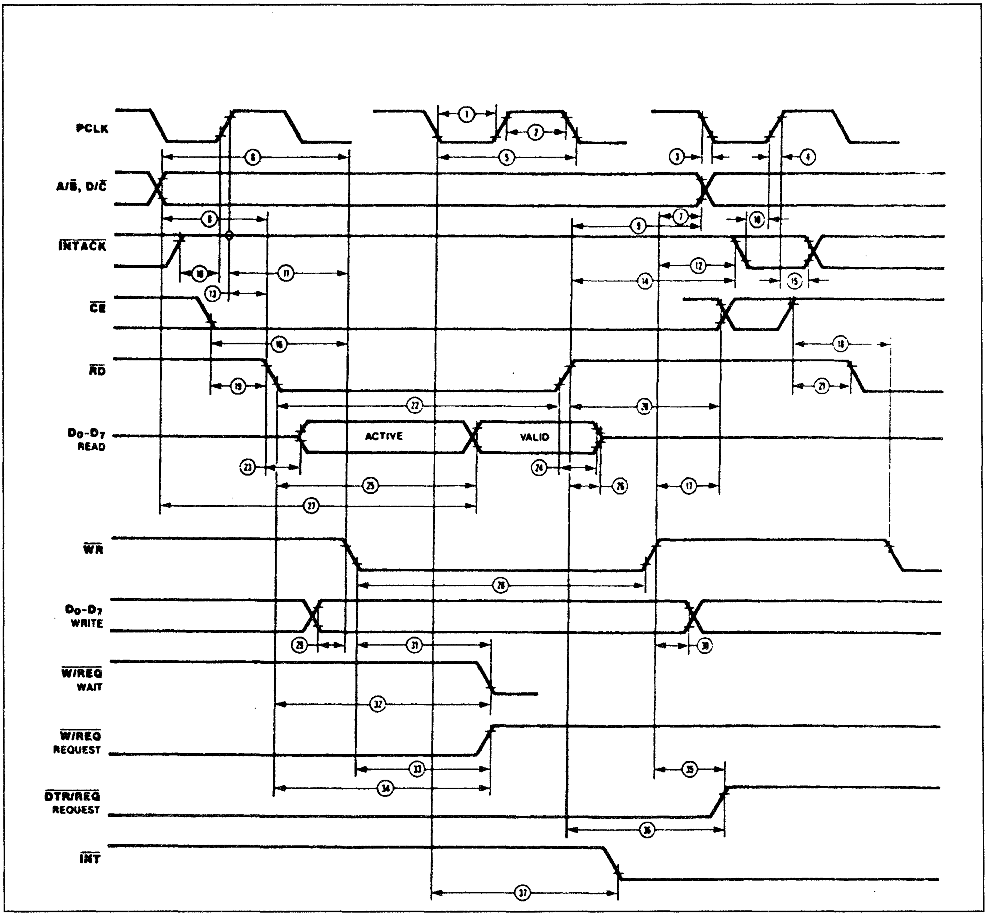
Capacitance

Symbol	Parameter	Test Conditions	Min.	Max.	Unit
C_{IN}	Input Capacitance			10	pf
C_{OUT}	Output Capacitance			15	pf
$C_{I/O}$	Bidirectional Capacitance			20	pf

f = 1 MHz, over specified temperature range.
Unmeasured pins returned to ground.

AC Timing Characteristics

Figure 75 : Read and Write Timing.



HARDWARE INFORMATION (cont'd)

Table 23 : AC Timing Characteristics.

N°	Symbol	Parameter	Z8530		Z8530A		Z8530B		Notes*
			Min. (ns)	**Max.** (ns)	**Min.** (ns)	**Max.** (ns)	**Min.** (ns)	**Max.** (ns)	
1	TwPC1	PCLK Low Width	105	2000	70	1000	50	1000	
2	TwPCh	PCLK High Width	105	2000	70	1000	50	1000	
3	TfPC	PCLK Fall Time		20		10		10	
4	TrPC	PCLK Rise Time		20		15		10	
5	TcPC	PCLK Cycle Time	250	4000	165	2000	125	2000	
6	TsA(WR)	Address to $\overline{WR}$ ↓ Setup Time	80		80		70		
7	ThA(WR)	Address to $\overline{WR}$ ↑ Hold Time	0		0		0		
8	TsA(RD)	Address to $\overline{RD}$ ↓ Setup Time	80		80		70		
9	ThA(RD)	Address to $\overline{RD}$ ↑ Hold Time	0		0		0		
10	TsIA(PC)	$\overline{INTACK}$ to PCLK ↑ Setup Time	10		10		10		
11	TsIAi(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↓ Setup Time	200		160		145		1
12	ThIA(WR)	$\overline{INTACK}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		
13	TsIAi(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↓ Setup Time	200		160		145		1
14	ThIA(RD)	$\overline{INTACK}$ to $\overline{RD}$ ↑ Hold Time	0		0		0		
15	ThIA(PC)	$\overline{INTACK}$ to PCLK ↑ Hold Time	100		100		85		
16	TsCE1(WR)	$\overline{CE}$ Low to $\overline{WR}$ ↓ Setup Time	0		0		0		
17	ThCE(WR)	$\overline{CE}$ to $\overline{WR}$ ↑ Hold Time	0		0		0		
18	TsCEh(WR)	$\overline{CE}$ High to $\overline{WR}$ ↓ Setup Time	100		70		60		
19	TsCE1(RD)	$\overline{CE}$ Low to $\overline{RD}$ ↓ Setup Time	0		0		0		1
20	ThCE(RD)	$\overline{CE}$ to $\overline{RD}$ ↑ Hold Time	0		0		0		1
21	TsCEh(RD)	$\overline{CE}$ High to $\overline{RD}$ ↓ Setup Time	100		70		60		1
22	TwRD1	$\overline{RD}$ Low Width	240		200		150		1
23	TdRD(DRA)	$\overline{RD}$ ↓ to Read Data Active Delay	0		0		0		
24	TdRDr(DR)	$\overline{RD}$ ↑ to Read Data Not Valid Delay	0		0		0		
25	TdRDf(DR)	$\overline{RD}$ ↓ to Read Data Valid Delay		250		180		140	
26	TdRD(DRz)	$\overline{RD}$ ↑ to Read Data Float Delay		70		45		40	2

Notes : 1. Parameter does not apply to Interrupt Acknowledge transactions.
2. Float delay is defined as the time required for a ± 0.5 V change in the output with a maximum dc load and minimum ac load.
* Timings are preliminary and subject to change.

HARDWARE INFORMATION (cont'd)

Figure 76 : Interrupt Acknowledge Timing.

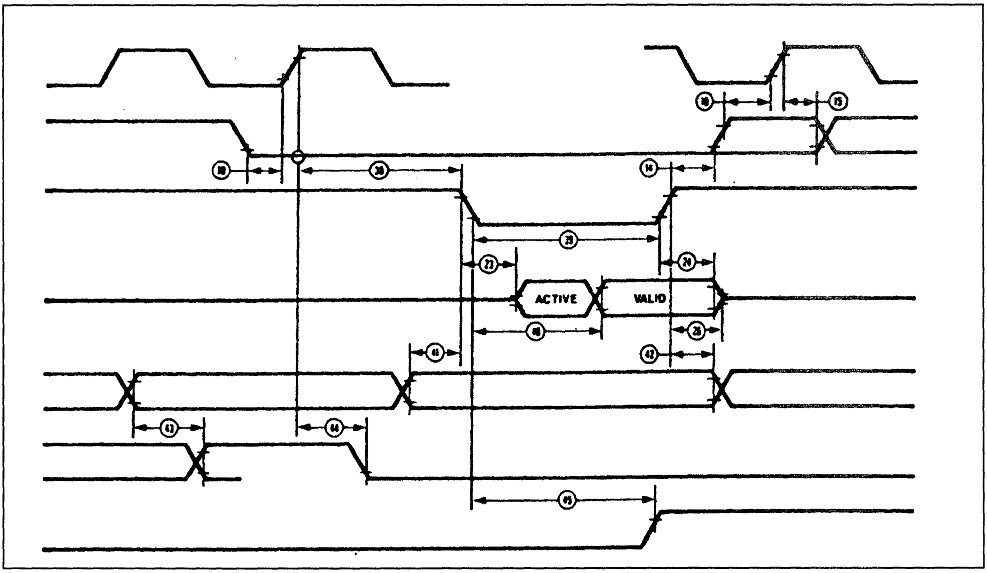


Figure 77 : Reset Timing.

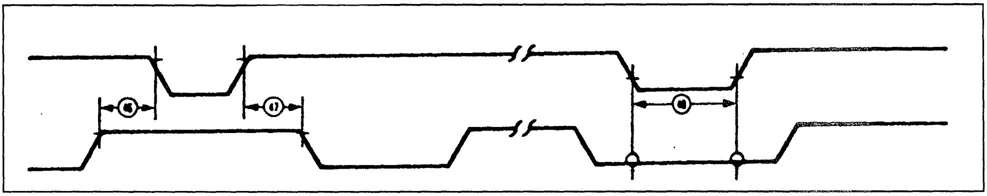
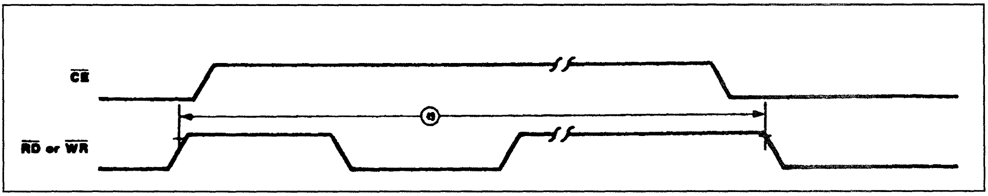


Figure 78 : Cycle Timing.



HARDWARE INFORMATION (cont'd)

Table 24 : AC Timing Characteristics.

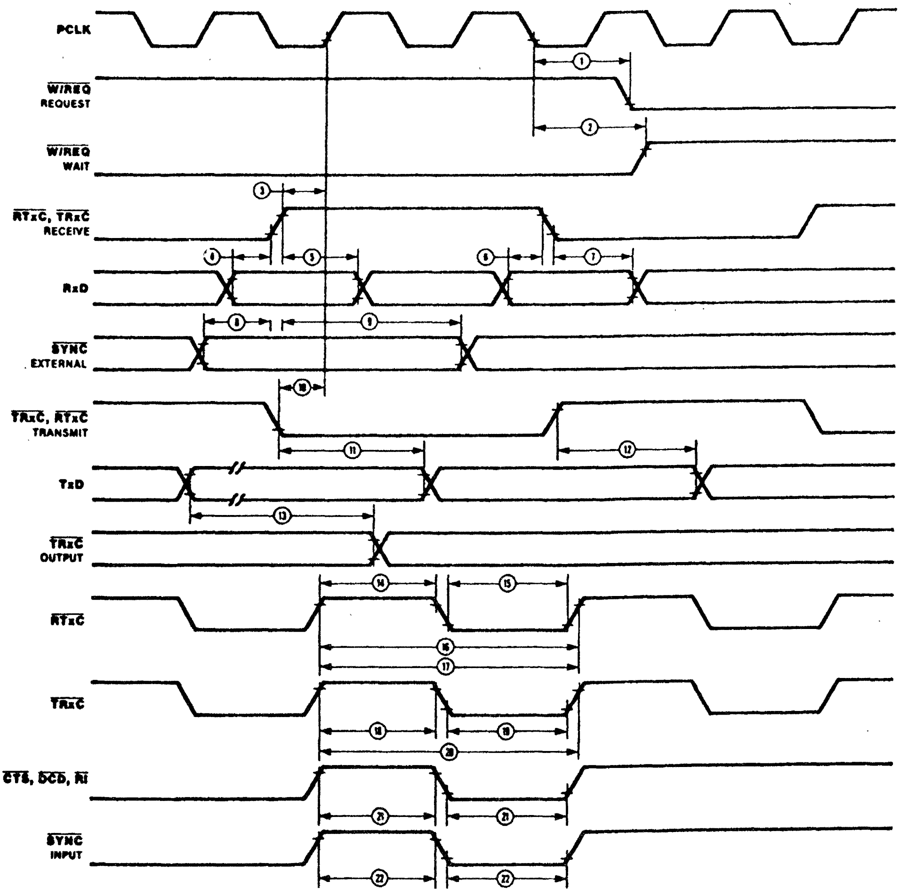
N°	Symbol	Parameter	Z8530		Z8530A		Z8530B		Notes*
			Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	
27	TdA(DR)	Address Required Valid to Read Data Valid Delay		300		280		220	
28	TwWR1	$\overline{\text{WR}}$ Low Width	240		200		150		
29	TsDW(WR)	Write Data to $\overline{\text{WR}}$ ↓ Setup Time	10		10		10		
30	ThDW(WR)	Write Data $\overline{\text{WR}}$ ↑ Hold Time	0		0		0		
31	TdWR(W)	$\overline{\text{WR}}$ ↓ to Wait Valid Delay		240		200		170	4
32	TdRD(W)	$\overline{\text{RD}}$ ↓ to Wait Valid Delay		240		200		170	4
33	TdWRf(REQ)	$\overline{\text{WR}}$ ↓ to $\overline{\text{W}}$ / $\overline{\text{REQ}}$ Not Valid Delay		240		200		170	
34	TdRDf(REQ)	$\overline{\text{RD}}$ ↓ to $\overline{\text{W}}$ / $\overline{\text{REQ}}$ Not Valid Delay		240		200		170	
35	TdWRr(REQ)	$\overline{\text{WR}}$ ↑ to $\overline{\text{DTR}}$ / $\overline{\text{REQ}}$ Not Valid Delay		4TcPC		4TcPC		4TcPC	
36	TdRDr(REQ)	$\overline{\text{RD}}$ ↑ to $\overline{\text{DTR}}$ / $\overline{\text{REQ}}$ Not Valid Delay		4TcPC		4TcPC		4TcPC	
37	TdPC(INT)	PCLK ↓ to $\overline{\text{INT}}$ Valid Delay		500		500		500	4
38	TdIAi(RD)	$\overline{\text{INTACK}}$ to $\overline{\text{RD}}$ ↓ (Acknowledge) Delay	250		200		150		5
39	TwRDA	$\overline{\text{RD}}$ (Acknowledge) Width	250		200		150		
40	TdRDA(DR)	$\overline{\text{RD}}$ ↓ (Acknowledge) to Read Data Valid Delay		250		180		140	
41	TsIEI(RDA)	IEI to $\overline{\text{RD}}$ ↓ (Acknowledge) Setup Time	120		100		95		
42	ThIEI(RDA)	IEI to $\overline{\text{RD}}$ ↑ (Acknowledge) Hold Time	0		0		0		
43	TdIEI(IEO)	IEI to IEO Delay Time		120		100		95	
44	TdPC(IEO)	PCLK ↑ to IEO Delay		250		250		200	
45	TdRDA(INT)	$\overline{\text{RD}}$ ↓ to $\overline{\text{INT}}$ Inactive Delay		500		500		450	4
46	TdRD(WRQ)	$\overline{\text{RD}}$ ↑ $\overline{\text{WR}}$ ↓ Delay for No Reset	30		15		15		
47	TdWRQ(RD)	$\overline{\text{WR}}$ ↑ to $\overline{\text{RD}}$ ↓ Delay for No Reset	30		30		20		
48	TwRES	$\overline{\text{WR}}$ and $\overline{\text{RD}}$ Coincident Low for Reset	250		200		150		
49	Trc	Valid Access Recovery Time	4TcPC		4TcPC		4TcPC		3

Notes :
3. Parameter applies only between transactions involving the SCC.
4. Open-drain output, measured with open-drain test load.
5. Parameter is system dependent. For any SCC in the daisy chain, TdIAi(RD) must be greater than the num of TdPC(IEO) for the highest priority device in the daisy chain, TsIEI(RDA) for the SCC, and TdIEIf(IEO) for each device separating them in the daisy chain.

* Timings are preliminary and subject to change.

HARDWARE INFORMATION (cont'd)

Figure 79 : General Timing.


PCLK
$\overline{W/REQ}$ REQUEST
$\overline{W/REQ}$ WAIT
$\overline{RTxC}$, $\overline{TRxC}$ RECEIVE
RxD
$\overline{SYNC}$ EXTERNAL
$\overline{TRxC}$, $\overline{RTxC}$ TRANSMIT
TxD
$\overline{TRxC}$ OUTPUT
$\overline{RTxC}$
$\overline{TRxC}$
$\overline{CTS}$, $\overline{DCD}$, $\overline{RI}$
$\overline{SYNC}$ INPUT
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
21
22
22

HARDWARE INFORMATION (cont'd)

Table 25 : General Timing Characteristics (cont'd).

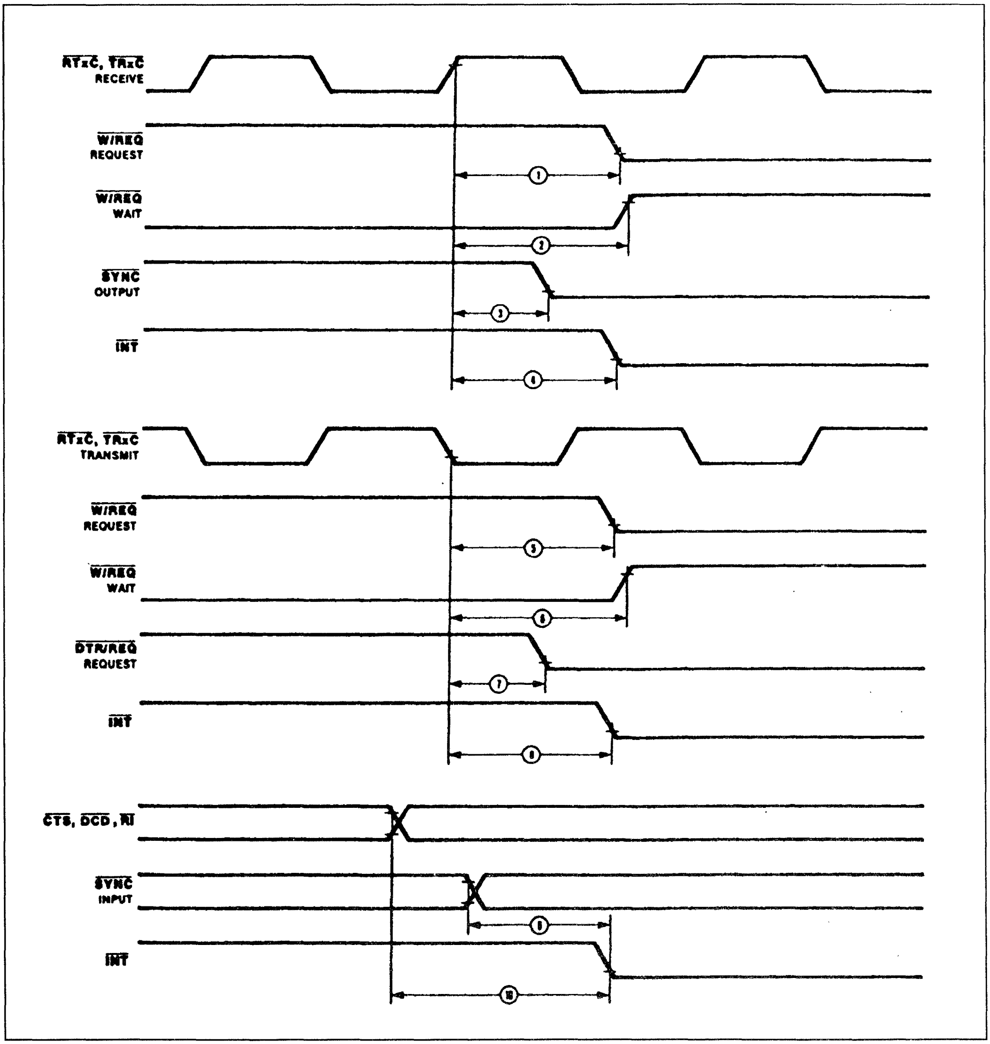
N°	Symbol	Parameter	Z8530		Z8530A		Z8530B		Notes*
			Min. (ns)	**Max.** (ns)	**Min.** (ns)	**Max.** (ns)	**Min.** (ns)	**Max.** (ns)	
1	TdPC(REQ)	PCLK ↓ to $\overline{\text{W}}$ / $\overline{\text{REQ}}$ Valid Delay		250		250		250	
2	TdPC(W)	PCLK ↓ to Wait Inactive Delay		350		350		350	
3	TsRXC(PC)	$\overline{\text{RxC}}$ ↑ to PCLK ↑ Setup Time (PCLK ÷ 4 case only)	80	TwPCL	70	TwPCL	60	TwPCL	1.4
4	TsRXD(RXCr)	RxD to $\overline{\text{RxC}}$ ↑ Setup Time (X1 Mode)	0		0		0		1
5	ThRXD(RXCr)	RxD to $\overline{\text{RxC}}$ ↑ Hold Time (X1 Mode)	150		150		150		1
6	TsRXD(RXCf)	RxD to $\overline{\text{RxC}}$ ↓ Setup Time (X1 Mode)	0		0		0		1.5
7	ThRXD(RXCf)	RxD to $\overline{\text{RxC}}$ ↓ Hold Time (X1 Mode)	150		150		150		1.5
8	TsSY(RXC)	$\overline{\text{SYNC}}$ to $\overline{\text{RxC}}$ ↑ Setup Time	– 200		– 200		– 200		1
9	ThSY(RXC)	$\overline{\text{SYNC}}$ to $\overline{\text{RxC}}$ ↑ Hold Time	3TcPC + 400		3TcPC + 320		3TcPC + 250		1
10	TsTXC(PC)	$\overline{\text{TxC}}$ ↓ to PCLK ↑ Setup Time	0		0		0		2.4
11	TdTXCf(TXD)	$\overline{\text{TxC}}$ ↓ to TxD Delay (X1 Mode)		300		230		200	2
12	TdTXCr(TXD)	$\overline{\text{TxC}}$ ↑ to TxD Delay (X1 Mode)		300		230		200	2.5
13	TdTXD(TRX)	TxD to $\overline{\text{TRxC}}$ Delay (Send Clock Echo)		200		200		200	
14	TwRTXh	$\overline{\text{RTxC}}$ High Width	180		180		150		6
15	TwRTX1	$\overline{\text{RTxC}}$ Low Width	180		180		150		6
16	TcRTX	$\overline{\text{RTxC}}$ Cycle Time	1000		640		500		6.7
17	TcRTXX	Crystal Oscillator Period	250	1000	165	1000	125	1000	3
18	TwTRXh	$\overline{\text{TRxC}}$ High Width	180		180		150		6
19	TwTRX1	$\overline{\text{TRxC}}$ Low Width	180		180		150		6
20	TcTRX	$\overline{\text{TRxC}}$ Cycle Time	1000		640		500		6.7
21	TwEXT	$\overline{\text{DCD}}$ or $\overline{\text{CTS}}$ Pulse Width	200		200		200		
22	TwSY	$\overline{\text{SYNC}}$ Pulse Width	200		200		200		

Notes :
1. $\overline{\text{RxC}}$ is $\overline{\text{RTxC}}$ or $\overline{\text{TRxC}}$, whichever is supplying the receive clock.
2. $\overline{\text{TxC}}$ is $\overline{\text{TRxC}}$ or $\overline{\text{RTxC}}$, whichever is supplying the transmit clock.
3. Both $\overline{\text{RTxC}}$ and $\overline{\text{SYNC}}$ have 30 pF capacitors to ground connected to them.
4. Parameter applies only if the data rate is one-fourth the PCLK rate. In all other cases, no phase relationship between $\overline{\text{RxC}}$ and PCLK or $\overline{\text{TxC}}$ and PCLK is required.
5. Parameter applies only to FM encoding/decoding.
6. Parameter applies only for transmitter and receiver ; DPLL and baud rate generator timing requirements are identical to chip PCLK requirements.
7. The maximum receive on transmit data is 1/4PLCK.

* Timings are preliminary and subject to change.

HARDWARE INFORMATION (cont'd)

Figure 80 : System Timing.



HARDWARE INFORMATION (cont'd)

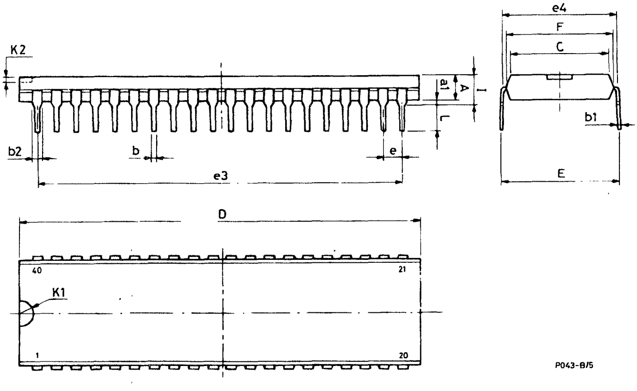
Table 26 : System Timing Characteristics (cont'd).

N°	Symbol	Parameter	Z8530		Z8530A		Z8530B		Notes* †
			Min.	Max.	Min.	Max.	Min.	Max.	
1	TdRXC(REQ)	$\overline{\mathrm{RxC}}$ ↑ to $\overline{\mathrm{W}}$ / $\overline{\mathrm{REQ}}$ Valid Delay	8	12	8	12	8	12	2
2	TdRXC(W)	$\overline{\mathrm{RxC}}$ ↑ to Wait Inactive Delay	8	12	8	12	8	14	1.2
3	TdRXC(SY)	$\overline{\mathrm{RxC}}$ ↑ to $\overline{\mathrm{SYNC}}$ Valid Delay	4	7	4	7	4	7	2
4	TdRXC(INT)	$\overline{\mathrm{RxC}}$ ↑ to $\overline{\mathrm{INT}}$ Valid Delay	10	16	10	16	10	16	1.2
5	TdTXC(REQ)	$\overline{\mathrm{TxC}}$ ↓ to $\overline{\mathrm{W}}$ / $\overline{\mathrm{REQ}}$ Valid Delay	5	8	5	8	5	8	3
6	TdTXC(W)	$\overline{\mathrm{TxC}}$ ↓ to Wait Inactive Delay	5	8	5	8	5	11	1.3
7	TdTXC(DRQ)	$\overline{\mathrm{TxC}}$ ↓ to $\overline{\mathrm{DTR}}$ / $\overline{\mathrm{REQ}}$ Valid Delay	4	7	4	7	4	7	3
8	TdTXC(INT)	$\overline{\mathrm{TxC}}$ ↓ to $\overline{\mathrm{INT}}$ Valid Delay	6	10	6	10	6	10	1.3
9	TdSY(INT)	$\overline{\mathrm{SYNC}}$ Transition to $\overline{\mathrm{INT}}$ Valid Delay	2	6	2	6	2	6	1
10	TdEXT(INT)	$\overline{\mathrm{DCD}}$ or $\overline{\mathrm{CTS}}$ Transition to $\overline{\mathrm{INT}}$ Valid Delay	2	6	2	6	2	6	1

Notes : 1. Open-drain output, measured with open-drain test load.
2. $\overline{\mathrm{RxC}}$ is $\overline{\mathrm{RTxC}}$ or $\overline{\mathrm{TRxC}}$, whichever is supplying the receive clock.
3. TxC is TRxC or RTxC, whichever is supplying the transmit clock.
* Timings are preliminary and subject to change.
↑ Units equal to TcPC.

PACKAGES MECHANICAL DATA

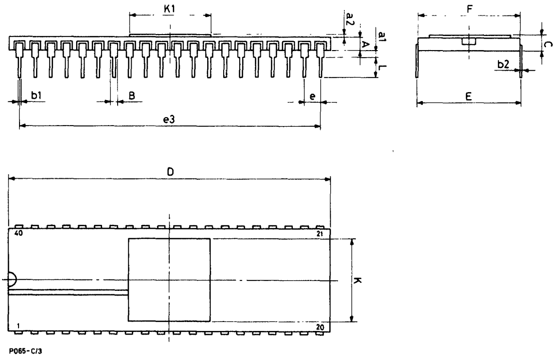
Figure 81: Z8530 40-Pin Dual in Line Plastic



Dim.	mm			Inches		
	Min	Typ	Max	Min	Typ	Max
A						
a1		0.63			.024	
b		0.45			.017	
b1	0.23		0.31	.009		.012
b2		1.27			.050	
C						
D			52.58			2.07
E	15.20		16.68	.598		.656
e		2.54			.100	
e3		48.26			1.90	
e4						
F			14.10			.555
I		4.45			.175	
L		3.30			.129	
K1						
K2						

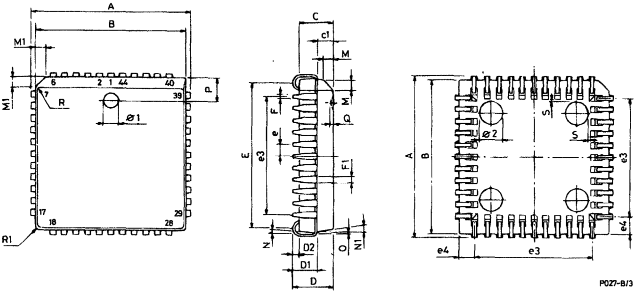
PACKAGE MECHANICAL DATA (cont'd)

Figure 82: Z8530 40-Pin Dual in Line Ceramic Multilayer.



Dim.	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A						
a1		1.27			.050	
a2						
B		1.27			.050	
b1		0.45			.017	
b2		0.25			.010	
C		2.33			.091	
D			51.30			2.01
E		15.24			.600	
e		2.54			.100	
e3		48.26			1.90	
F			15.36			.604
L		3.30			.129	
K		12.80			.503	
K1		12.80			.503	

Figure 83 : Z8530 44–Lead Plastic Leaded Chip Carrier.



Dim.	mm			Inches		
	Min	Typ	Max	Min	Typ	Max
A	17.40		17.65	0.685		0.694
B	16.51		16.65	0.650		0.655
C	3.65		3.70	0.143		0.145
c1						
D	4.20		4.57	0.165		0.179
d1	2.59		2.74	0.101		0.107
d2		0.68			0.026	
E	14.99		16.00	0.590		0.629
e		1.27			0.050	
e3		12.70			0.500	
e4			1.98			0.077
F		0.46			0.018	
F1		0.71			0.027	
G			0.101			0.004
M		1.16			0.045	
M1		1.14			0.044	
Ø1		1.57				

ORDERING INFORMATION

Sales Type	Frequency	Supply Voltage	Temp. Range	Package
Z8530B1V	4 MHz	5 V ± 5 %	0 to + 70 °C	PDIP-40
Z8530B6V	4 MHz	5 V ± 5 %	– 40 to + 85 °C	PDIP-40
Z8530D1N	4 MHz	5 V ± 5 %	0 to + 70 °C	CDIP-40
Z8530D6N	4 MHz	5 V ± 5 %	– 40 to + 85 °C	CDIP-40
Z8530D2N	4 MHz	5 V ± 5 %	– 55 to + 125 °C	CDIP-40
Z8530C1V	4 MHz	5 V ± 5 %	0 to + 70 °C	PLCC44
Z8530C6V	4 MHz	5 V ± 5 %	– 40 to + 85 °C	PLCC44
Z8530AB1V	6 MHz	5 V ± 5 %	0 to + 70 °C	PDIP-40
Z8530AB6V	6 MHz	5 V ± 5 %	– 40 to + 85 °C	PDIP-40
Z8530AD1N	6 MHz	5 V ± 5 %	0 to + 70 °C	CDIP-40
Z8530AD6N	6 MHz	5 V ± 5 %	– 40 to + 85 °C	CDIP-40
Z8530AD2N	6 MHz	5 V ± 5 %	– 55 to + 125 °C	CDIP-40
Z8530AC1V	6 MHz	5 V ± 5 %	0 to + 70 °C	PLCC44
Z8530AC6V	6 MHz	5 V ± 5 %	– 40 to + 85 °C	PLCC44
Z8530BB1V	8 MHz	5 V ± 5 %	0 to + 70 °C	PDIP-40
Z8530BB6V	8 MHz	5 V ± 5 %	– 40 to + 85 °C	PDIP-40
Z8530BD1N	8 MHz	5 V ± 5 %	0 to + 70 °C	CDIP-40
Z8530BD6N	8 MHz	5 V ± 5 %	– 40 to + 85 °C	CDIP-40
Z8530BC1V	8 MHz	5 V ± 5 %	0 to + 70 °C	PLCC44
Z8530BC6V	8 MHz	5 V ± 5 %	– 40 to + 85 °C	PLCC44

Note : PDIP = Plastic DIP ; CDIP = Ceramic Multilayer DIP ; PLCC = Plastic Leaded Chip Carrier.

SGS-THOMSON
MICROELECTRONICS

APPLICATION NOTE

INTERFACING Z8500 UNIVERSAL PERIPHERALS TO THE TS68000

INTRODUCTION

This Application Note discusses interfacing the SGS-THOMSON Z8500 family of peripherals to the TS68000 microprocessor. The Z8500 peripheral family includes the Z8536 Counter/Timer and Parallel I/O Unit (CIO), The Z8038 FIFO Input/Output Interface Unit (FIO), and the Z8530 Serial Communications Controller (SCC). This document discusses the Z8500/68000 interfaces and presents hardware examples, and verification techniques. One of the three hardware examples given in this paper shows how to implement the Z8500/68000 interface using a single-chip programmable logic array (PAL).

This Application Note about interfacing supplements the following documents, which discuss the individual components of the interface.

- Z8038 FIO/FIFO Datasheet
- Z8530 SCC Datasheet
- Z8531 A-SCC Datasheet
- Z8536 CIO Datasheet
- TS68000 Datasheet

This Application Note is divided into four sections. The first section gives a general description of the Z8500 family and discusses pin functions, interrupt structures, and the programming of operating modes. The second section discusses the Z8500 interface itself. It shows how the different Z8500 control signals are generated from the 68000 signals and summarizes the critical timings for the three types of bus cycles. The third section shows three examples of implementing the TS68000 to Z8500 peripheral interface. The fourth section suggests methods of verifying the interface design by checking the three different types of bus cycle : Read, Write, and Interrupt Acknowledge.

GENERAL Z8500 FAMILY DESCRIPTION

The Z8500 family is made up of programmable peripherals that can interface easily to the bus of any nonmultiplexed CPU microprocessor, such as the TS68000. The three members of this family, the CIO, SCC, and FIO, can solve many design problems. The peripherals' operating modes can be programmed simply by writing to their internal registers.

PROGRAMMING THE OPERATING MODES

The CPU can access two types of register : Control and Data. Depending on the peripheral, registers are selected with either the A_0, A_1, $A/\overline{B}$, or $D/\overline{C}$ function pins.

Peripheral operating modes are initialized by programming internal registers. Since these registers are not directly addressable by the CPU, a two-step procedure using the Control register is required : first, the address of the internal register is written to the Control register, then the data is written to the Control register. A state machine determines whether an address or data is being written to the Control register. Reading an internal register follows a similar two-step procedure : first, the address is written then the data is read.

The Data registers that are most frequently accessed, for example, the SCC's transmit and receive buffer, can be addressed directly by the CPU with a single read or write operation. This reduces overhead in data transfers between the peripheral and CPU.

GENERATING Z8500 CONTROL SIGNALS

This section shows how to generate the Z8500 control signals. To simplify the discussion, the section is divided into two parts. The first part takes each individual Z8500 signal and shows how it is generated from the TS68000 signals. The second part discusses the Z8500 timing that must be met when generating the control signals.

Z8500 SIGNAL GENERATION

The right-hard side of table 1 lists the Z8500 signals that must be generated. Each of these signals is discussed in a separate paragraph.

A_0,A_1, $A/\overline{B}$, $D/\overline{C}$. These pins are used to select the peripheral's Control and Data registers that program the different operating modes. They can be connected to the TS68000 A_1 and A_2 Address bus lines.

$\overline{CE}$. Each peripheral has an active Low Chip Enable that can be derived by ANDing the selected address decode and the 68000's Address Strobe ($\overline{AS}$). The active Low $\overline{AS}$ guarantees that the TS68000 addresses are valid.

D_0-D_7. The Z8500 Data bus can be directly connected to the lowest byte (D_0-D_7) of the 68000 Data bus.

IEI, IEO. The peripherals use these pins to decide the interrupt priority. The highest priority device should have its IEI tied High. Its IEO should be connected to the IEI pin of the next highest priority device. This pattern continues with the next highest priority peripheral, until the peripherals are all connected, as shown in Figure 1.

$\overline{\textbf{INT}}$. The interrupt request pins for each peripheral in the daisy chain can be wire-ORed and connected to the 68000's ILP $_n$ pins. The 68000 has seven interrupt levels that can be encoded into the $\overline{ILP}_0$, $\overline{ILP}_1$, and $\overline{ILP}_2$ pins. Multiple 68000 interrupt levels can be implemented by using a multiplexer like the 74LS148.

$\overline{\textbf{INTACK}}$. The $\overline{INTACK}$ pin signals the peripheral that an Interrupt Acknowledge cycle is occurring. The following equation describes how $\overline{INTACK}$ is generated :

$$\overline{INTACK} = \overline{(FC_0) . (FC_1) . (FC_2) . (\overline{A_S})}$$

The 68000 FC_0-FC_2 are status pins that indicate an Interrupt Acknowledge when they are all High. They should be ANDed with inverted $\overline{AS}$ to guarantee their validity. The $\overline{INTACK}$ signals must be synchronized with PCLK to guarantee set-up and hold times. This can be accomplished by changing the state of $\overline{INTACK}$ on the falling edge of PCLK. If the $\overline{INTACK}$ pin is not used, it must be tied High.

PCLK. The SCC and CIO require a clock for internal synchronization. The clock can be generated by dividing down the 68000 CLK.

$\overline{\textbf{RD}}$. The Read strobe goes active Low under three conditions : hardware reset, normal Read cycle, and an Interrupt Acknowledge cycle. The following equation describes how RD is generated :

$$\overline{RD} = [\overline{(R/\overline{W}) . (AS) + RESET)}]$$

The Read strobe timing must meet both the Read timing and Interrupt Acknowledge timing discussed in the following section. In addition to enabling the Data bus drivers, the falling edge of $\overline{RD}$ sets the Interrupt Under Service (IUS) bits during an Interrupt Acknowledge cycle.

$\overline{\textbf{WR}}$. This signal strobes data into the peripheral. A data to-write setup time requires that data be valid before $\overline{WR}$ goes active Low. The $\overline{WR}$ equation for generating the $\overline{WR}$ strobe is made up of two components : an active reset and a normal Write cycle, as shown in the allowing equation :

$$\overline{WR} = [\overline{\overline{(R/\overline{W})} . (AS) + RESET}]$$

Forcing $\overline{RD}$ and $\overline{WR}$ simultaneously Low resets the peripherals.

Table 1 : Z8500 and TS68000 Pin Functions.

TS68000 Signals		Z8500 Signals	
Mnemonic	**Function**	**Mnemonic**	**Function**
A_1-A_{23}	Address Bus	A_0, A_1, $A/\overline{B}$, $D/\overline{C}$*	Register Select
$\overline{AS}$	Address Strobe	$\overline{CE}$	Chip Enable
CLK	68000 Clock (8MHz)	D_0-D_7	Data Bus
D_0-D_{15}	Data Bus	IEI, IEO	Interrupt Daisy Chain Control
$\overline{DTACK}$	Data Transfer Acknowledge	$\overline{INT}$	Interrupt Request
FC_0-FC_2	Processor Status	$\overline{INTACK}$	Interrupt Acknowledge
$\overline{ILP}_0$-$\overline{ILP}_2$	Interrupt Request	PCLK	Peripheral Clock
$R/\overline{W}$	Read/write	$\overline{RD}$	Read Strobe
$\overline{VMA}$	Valid Memory Address	$\overline{WR}$	Write Strobe
$\overline{VPA}$	Valid Peripheral Address		

Note : * The register select pins on each peripheral have different names.

Z8500 TIMING CYCLES

This section discusses the timing parameters that must be met when generating the control signals. The Z8500 family uses the control signals to communicate with the CPU via three types of bus cycle : Read, Write, and Interrupt Acknowledge. The discussion that follows pertains to the 4MHz peripherals, but the 6MHz devices have similar timing considerations.

Although the peripherals have a standard CPU interface, some of their particular timing requirements vary. The worst-case parameters are shown below ; the timing can be optimized if only one or two of the Z8500 family devices are used.

READ CYCLE

The Read cycle transfers data from the peripheral to the CPU. It begins by selecting the peripheral and appropriate register (Data or Control). The data is gated onto the bus with the RD line. A setup time of 80ns from the time the register select input (A/B, C/D, A_0, A_1) are stable to the falling edge of RD guarantees that the proper register is accessed. The access time specification is usually measured from the falling edge of RD to valid data and varies between peripherals. The SCC specifies an additional register select to valid data time. The Read cycle timing is shown in figure 2.

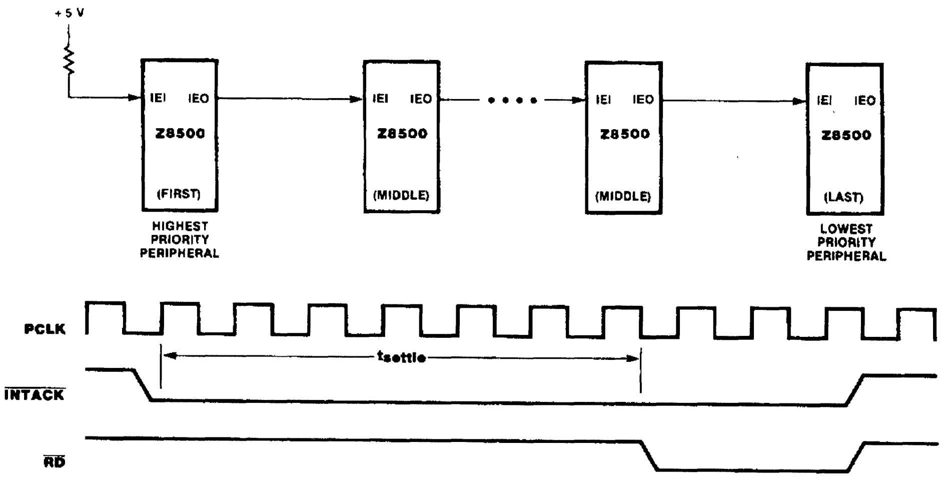
WRITE CYCLE

The Write cycle transfers data from the CPU to the peripheral. It begins by selecting the peripheral and addressing the desired register. A setup time of 80ns from register select stable to the falling edge of WR is required. The data must be valid prior to the falling edge of WR. The WR pulse width is specified at 400ns. Write cycle timing is shown in figure 2.

INTERRUPT ACKNOWLEDGE CYCLE

The Z8500 peripheral interrupt structure offers the designer many options. In the simplest case, the Z8500 peripherals can be polled with interrupts disabled. If using interrupts, the timing shown in figure 2 should be observed. (Detailed discussions of the interrupt processing can be found in the relevant datasheet).

Figure 1 : Peripheral Interrupt Daisy Chain.



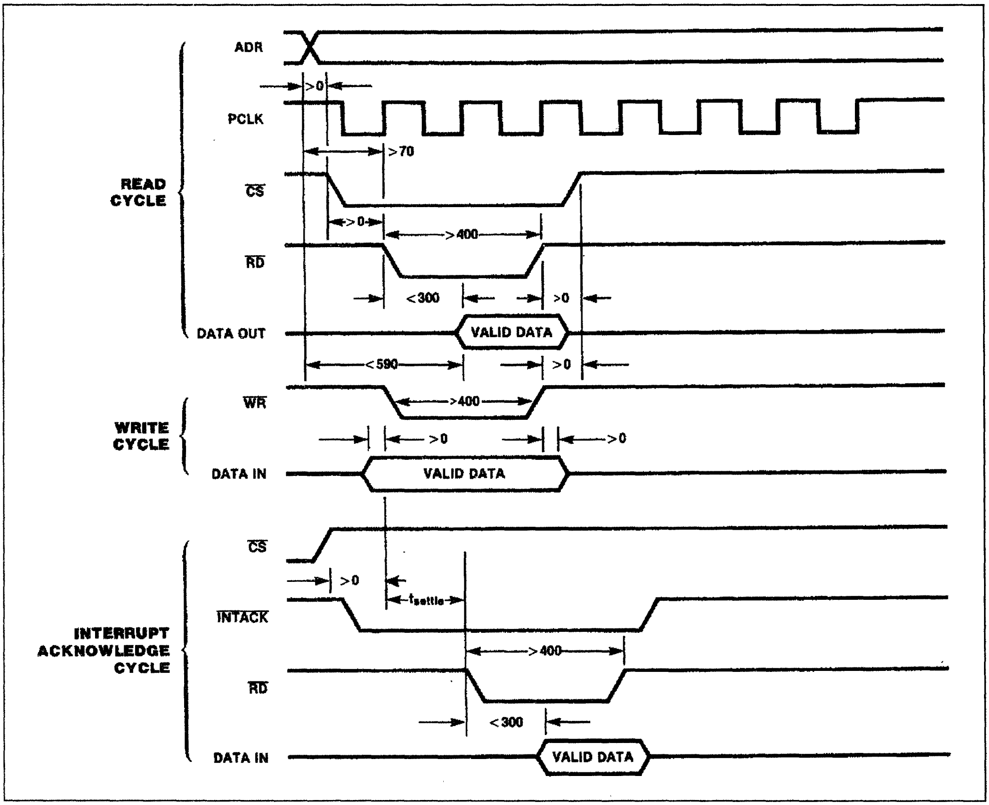
	t_{settle} (NS)		
PERIPHERAL (4MHz)	FIRST	MIDDLE	LAST
CIO	350	150	100
FIO	350	150	100
SCC	250	120	120

An interrupt sequence begins with an $\overline{INT}$ going active because of an interrupt condition. The CPU acknowledges the interrupt with an INTACK signal.

A daisy-chain settle time (dependent upon the number of devices in the chain) ensures that the interrupts are prioritized. The falling edge of $\overline{RD}$ causes the IUS bit to be set and enables a vector to go out on the bus.

The table given figure 1 can be used to calculate the amount of settling time required by a daisy chain. Even if there is only one peripheral in the chain, a minimum settling time is still required because of the internal daisy chain. The first column specifies the amount of settling time for only one peripheral. If there are two peripherals, the time is computed by adding together the times shown in the first and the last columns. For each additional peripheral in the chain, the time specified in the middle column is added.

Figure 2 : Z8500 Interface Timing (4MHz).



RECOVERY TIME

The read/write recovery time specifies a minimum amount of time between Read or Write cycles to the same peripheral. The recovery time differs among peripherals and is summarized in figure 3. In most cases, this parameter is met because of the time required for instruction fetches. The recovery time specification does not have to be met if $\overline{\text{CE}}$ is deselected when Read or Write occurs.

68000 INTERFACE EXAMPLES

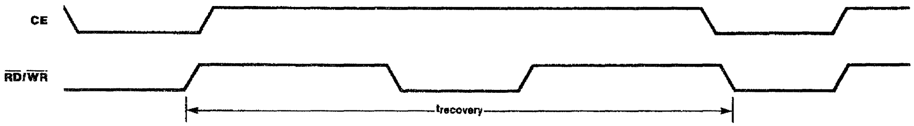
This section shows three examples, presented in increasing order of complexity, for interfacing the 4MHz Z8500 peripherals to an 8MHz TS68000. Faster CPUs or peripherals can be used by modifying some of the timing. These examples suggest possible ways of implementing the interface but may require some modifications to operate properly. They were chosen because they give the user a variety of interface design ideas. The first example uses a minimum amount of TTL logic to implement the interface because the Valid Peripheral Address (VPA) cycle meets the Z8500 timing requirements. In this mode the 68000 accepts only nonvectored interrupts. The second example uses the Data Transfer Acknowledge (DTACK) pin. This interface allows faster operation and makes use of the Z8500's 8-bit vectored interrupts. The third example also uses a DTACK cycle and is similar to the second, except the external logic is integrated into a single chip, the PAL20X10 programmable array logic.

EXAMPLE 1 : A TTL INTERFACE USING A VPA CYCLE

The 68000 has a special input pin, Valid Peripheral Address (VPA), that can be activated by the Z8500 chip select logic at the beginning of the cycle to indicate to the 68000 that a peripheral is being accessed. This generates a special Read/Write cycle that meets the peripheral timing requirements. This cycle allows the Z8500 control signals to be generated easily. The 68000 responds to interrupts using an autovector and the Z8500 can be programmed not to return a vector.

Figure 4 shows how the hardware can be implemented. PCLK is generated by dividing down the 68000 CLK. $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{INTACK}}$ are simply ANDed 68000 singnals. The worst-case daisy-chain settle time is 450ns. Connecting $\overline{\text{INT}}$ to $\overline{\text{IPL}}_0$ generates a level 1 interrupt. The internal registers are accessed by A_0, A_1, D/$\overline{\text{C}}$ and A/$\overline{\text{B}}$, which can be the 68000 lowest order addresses. The timing is shown in figure 5.

Figure 3 : Recovery Time.



Peripheral (4MHz)	Recovery Time
CIO	Greater than 3 PCLK cycles or 1000ns
FIO	Greater than 1000ns
SCC	Greater than 6 PCLK cycles + 200ns

Note : the diagram shows that the recovery time is measured between consecutive reads and writes only if the peripheral is selected.

Figure 4 : Interface Using the $\overline{\text{VPA}}$ Cycle.

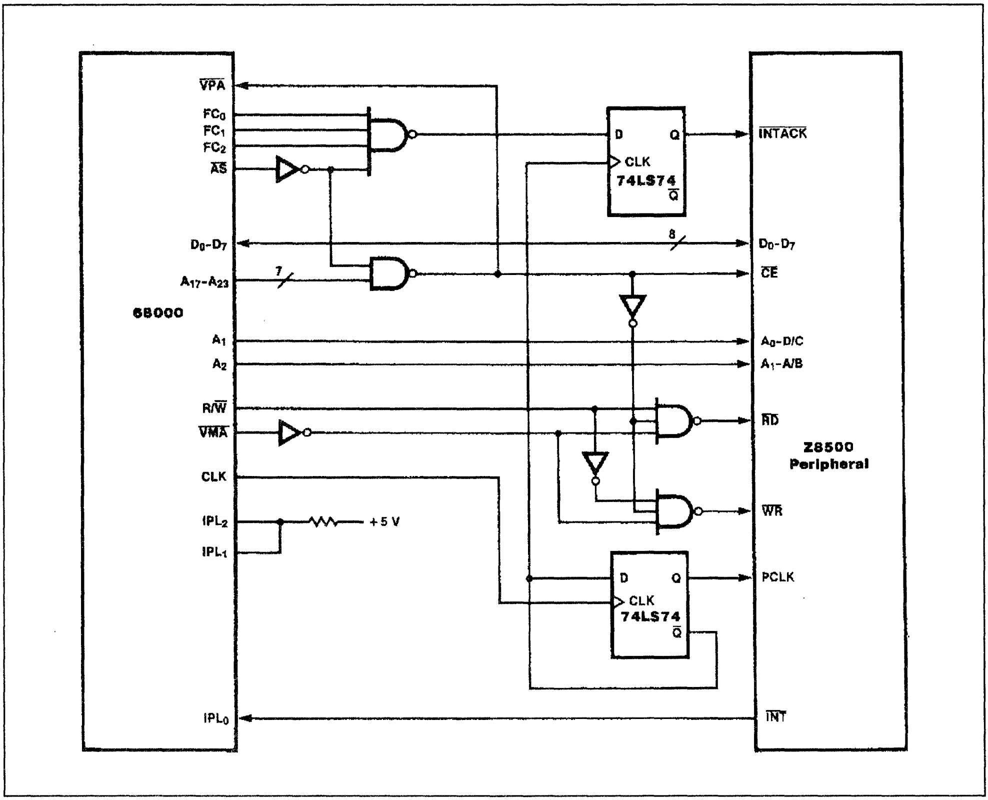
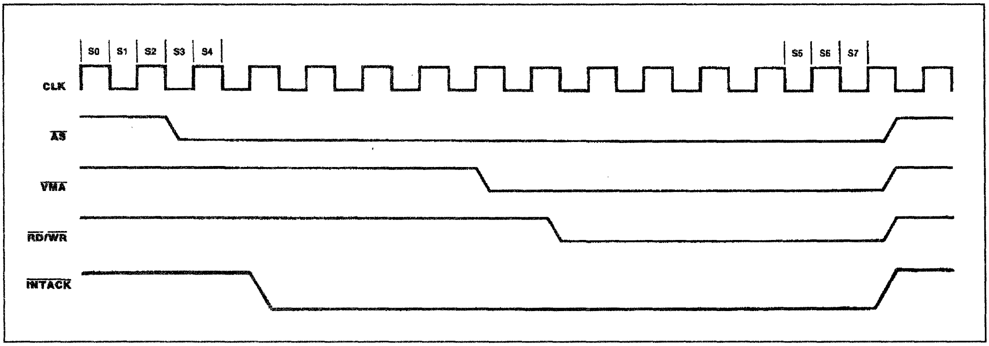


Figure 5 : $\overline{\text{VPA}}$ Cycle Timing.



FUNCTIONAL DESCRIPTION

$\overline{\text{VPA}}$ is pulled Low at the beginning of the cycle and the CPU automatically inserts Wait states until E is synchronized.

VPA = [(AS) . (CE)]

RD = [(CE) . (VMA) . $(\text{R}/\overline{\text{W}})$]

WR = [(CE) . (VMA) . $(\overline{\text{R}/\overline{\text{W}}})$]

INTACK = [(FC0) . (FC1) . (FC2) . (AS)]

EXAMPLE 2 : A TTL INTERFACE USING DTACK CYCLES

Using at the 68000 Data Transfer Acknowledge (DTACK) cycle is a second way of interfacing to the Z8500 peripherals. The 68000 inserts Wait states until the $\overline{\text{DTACK}}$ input is strobed Low to complete the transfer. In addition to generating the control signals, the interface logic must also generate DTACK.

The timing shown in figure 6 can be generated by the hardware shown in figure 7. The 8-bit Shift register (74LS164) is used to generate the proper timing. At the beginning of each cycle, Q_A (figure 7) is set High for one PCLK cycle and then reset. This pulse is shifted through the Q_A-Q_H outputs and is used to generate $\overline{\text{RD}}$, $\overline{\text{WR}}$, and DTACK signals. Some of the extra Wait states can be eliminated by tapping the Shift register sooner (e.g., Q_C).

EXAMPLE 3 : SINGLE-CHIP PAL INTERFACE

This example illustrates how to interface the 4MHz Z8500 peripherals to the 8MHz 68000 using a PAL20X10 device to generate all the required control signals. The PAL reduces the required interface logic to a single chip, thus minimizing board space. This interface offers flexibility because the internal logic can be reprogrammed without changing the pin functions. The PAL uses 68000 signals to generate Read, Write, and Interrupt Acknowledge cycles. In addition to generating the Z8500 control signals, the PAL also generates a DTACK (Data Transfer Acknowledge) to inform the 68000 of a completed data transfer cycle. This allows the 68000 to use the peripheral's vectored interrupts.

Figure 6 : Timing for $\overline{\text{DTACK}}$ Interface.

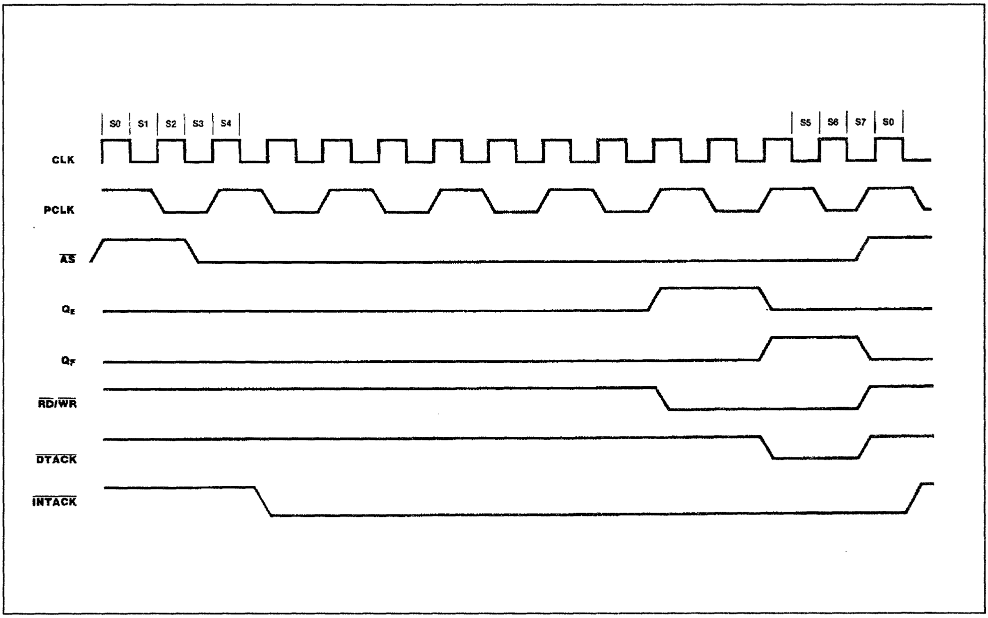
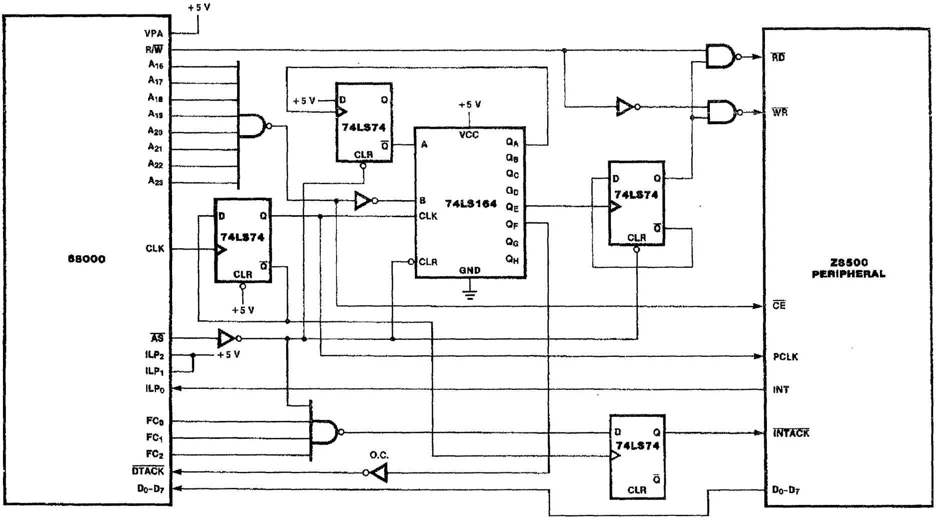

S0
S1
S2
S3
S4
S5
S6
S7
S0
CLK
PCLK
$\overline{\text{AS}}$
Q_E
Q_F
$\overline{\text{RD/WR}}$
$\overline{\text{DTACK}}$
$\overline{\text{INTACK}}$

Figure 7 : Hardware Diagram for DTACK Interface.



FUNCTIONAL DESCRIPTION

Figure 8 shows the PAL's pin functions. The PAL generates five control signals, of which four (WR, RD, C_0, and INTACK) go to the Z8500 and one (DTACK) goes to the 68000. The remaining signals are used internally to generate these outputs. Timing diagrams for the Read, Write, and Interrupt Acknowledge cycles are shown in Figure 9.

The PAL uses a 4-bit downcounter to generate the proper placement of the control signals where C_0 is the least-significant bit and C_3 is the most-significant bit. All of the PAL is clocked with the rising edge of the 68000's CLK. The counter toggles between counts 14 and 15 and starts counting down when AS goes active. The counter goes back to toggling when AS goes inactive. CYC goes active Low at the same time the counter starts counting down. The equations in Table 2 can be entered into a development board to program the PAL.

Figure 8 : PAL Pinout.

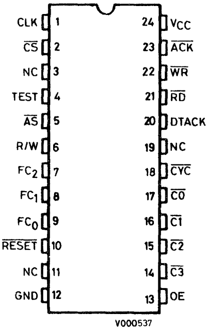


Figure 9 : PAL Interface Timing.

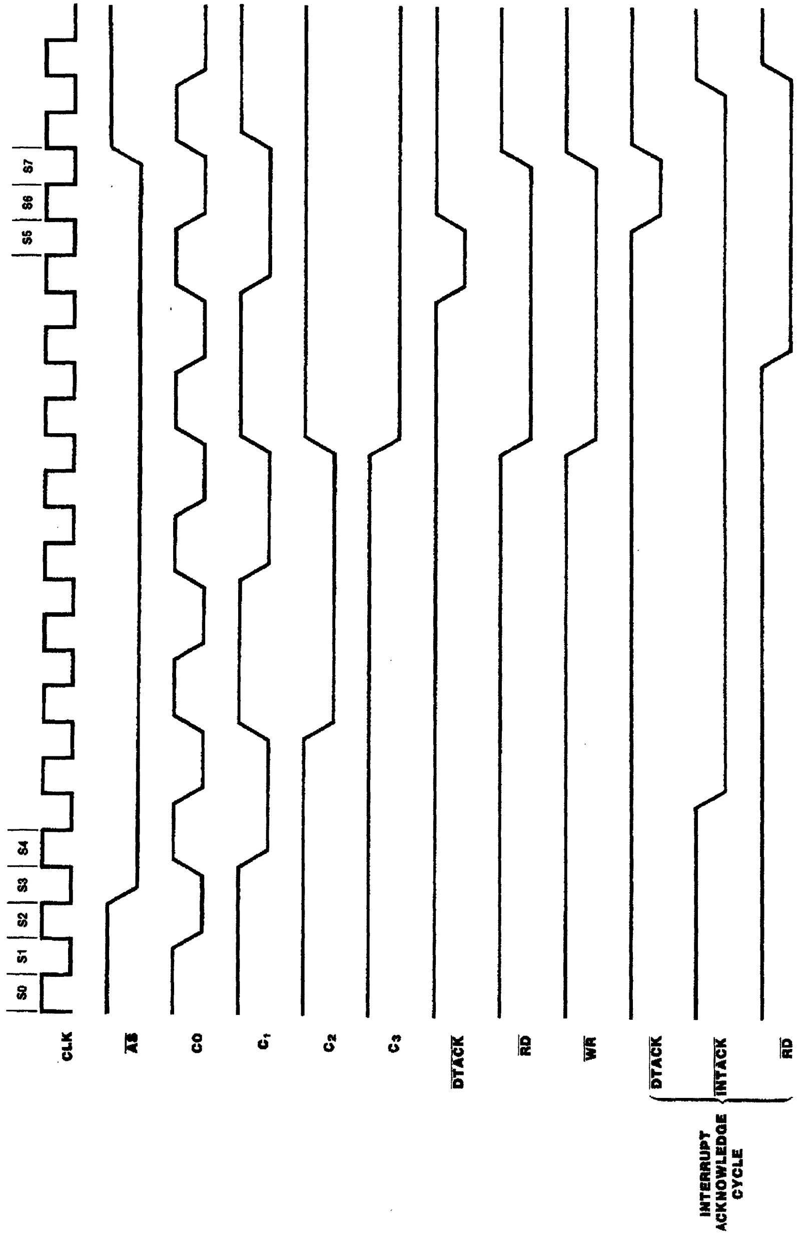
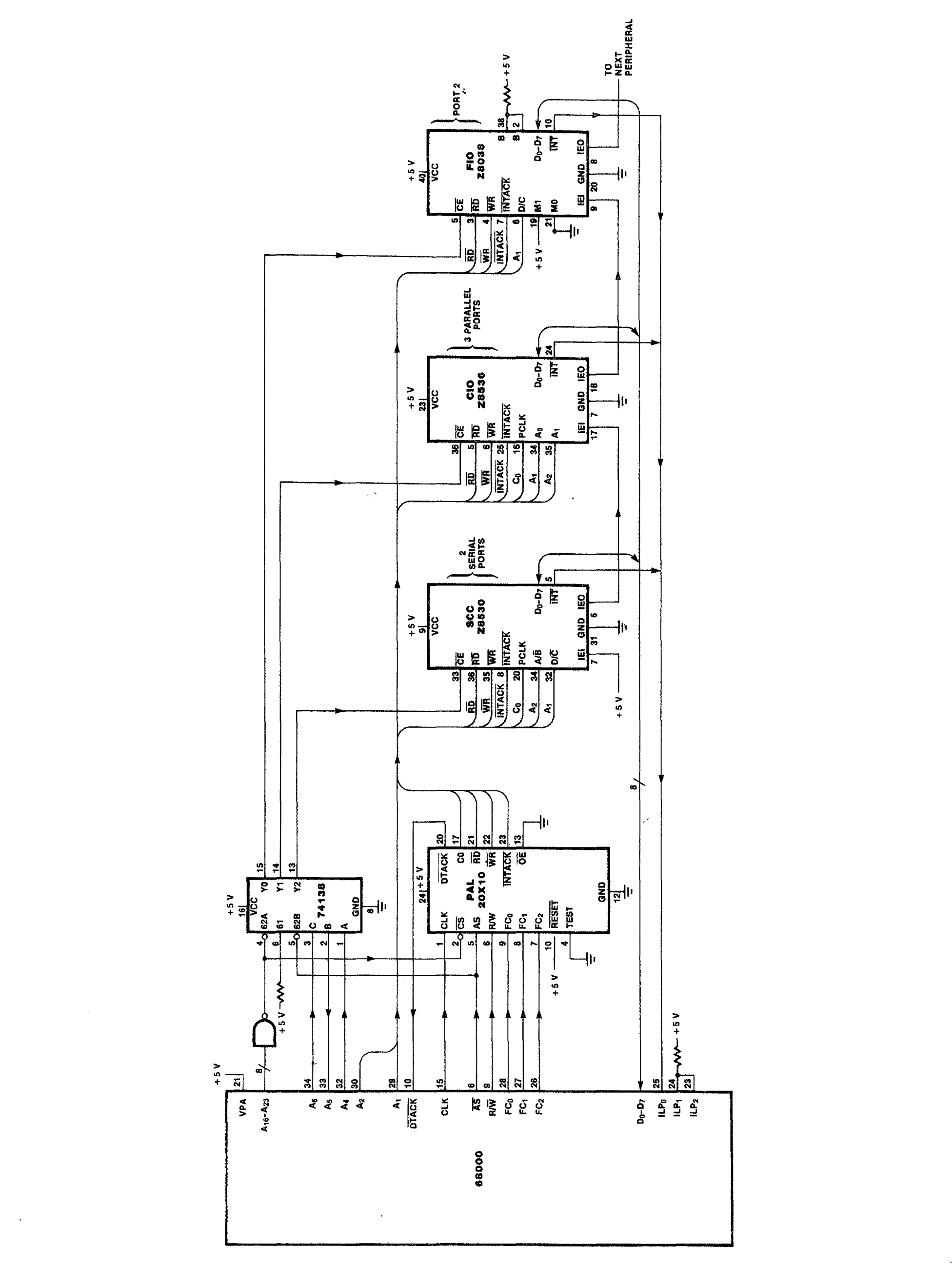


Table 2 : PAL Equations.

```
PAL20X10                                                    PAL DESIGN SPECIFICATION
P7089
MC68000 TO SGS-THOMSON PERIPHERAL INTERFACE
MMI, SUNNYVALE, CA
CLK / CS NC TEST / AS RW
FC2 FC1 FC0 / RESET NC GND
/ OE / C3 / C2 / C1 / C0 / CYC
NC / DTK / RD / WR / ACK VCC


CD      :=      /C0 */TEST                                  ; COUNT/HOLD (LSB)


C1      :=      /RESET*AS*C1                                ; HOLD
        :+:     /RESET*AS*C0                                ; DECREMENT


C2      :=      /RESET*AS*C2                                ; HOLD
        :+:     /RESET*AS*C0*C1                             ; DECREMENT


C3      :=      /RESET*AS*C3                                ; HOLD
        :+:     /RESET*AS*C0*C1*C2                          ; DECREMENT


DTK     :=      /RESET*/ACK*CYC*C3*/C2*/C1*C0*CS            ; DTACK FOR RD/WR CYCLE
         +      /RESET*ACK*CYC*C3*/C2*C1*/C0                ; DTACK FOR INTERRUPT
                                                            ; OPERATION


CYC     :=      /RESET*AS*/CYC*C0                           ; NEW CYCLE STARTED
         +      /RESET*AS*CYC                               ; PROCESSING OF CYCLE
        :+:     /RESET*CYC*DTK                              ; END OF CYCLE


RD      :=      /RESET*CYC*/ACK*RW*C3*/C2*CS                ; NORMAL READ OPERATION
         +      /RESET*CYC*/ACK*RW*/C3*C2*C1*C0*CS          ; NORMAL READ OPERATION
        :+:     /RESET*CYC*ACK*RW*C3                        ; READ DURING OPERATION


WR      :=      /RESET*CYC*/ACK*/RW*C3*/C2*CS               ; WRITE
         +      /RESET*CYC*/ACK*/RW*/C3*C2*C1*C0*CS         ; WRITE
        :+:     RESET


ACK     :=      /RESET*FC0*FC1*FC2*AS*CYC*/CD               ; INTERRUPT ACKNOWLEDGE
         +      /RESET*FC0*FC1*FC2*CYC                      ; INTERRUPT ACKNOWLEDGE
```

Figure 10 : PAL Hardware Diagram.



HARDWARE DIAGRAM

The hardware diagram of the PAL interface is shown in figure 10. The 68000 signals CLK, CS, AS, R/W, FC_0, FC_1 and FC_2 are used to generate the Z8500 control signals. The control signals are synchronous with the rising edge of the 68000's CLK. TEST and OE must be grounded. CS is used to enable DTACK, RD, and WR as shown in the equations. The Z8500 INT is connected to ILP $_0$, which generates a 68000 level 1 interrupt. The peripherals are memory-mapped into the highest 64K byte block of memory, where A_{17} - A_{23} equals "FF_H". Addresses A_4 - A_6 are used to select the peripheral ; A_1 - A_3 select the internal registers. Table 3 shows the peripheral's memory map.

Table 3 : Peripheral Memory Map.

Peripheral	Register	Hex Address
SCC (Z8530)	Channel B Control Channel B Data Channel A Control Channel B Data	FF0020 FF0022 FF0024 FF0026
CIO (Z8536)	Port C's Data Register Port B's Data Register Port A's Data Register Control Register	FF0010 FF0012 FF0014 FF0016
FIO (Z8038)	Data Register Control Register	FF0000 FF0002

INTERFACE VERIFICATION TECHNIQUES

This section suggests possible ways of verifying the Read, Write, and Interrupt Acknowledge cycles.

READ CYCLE VERIFICATION

The Read cycle should be checked first because it is the simplest operation. The Z8500 should be hardware reset by simultaneously pulling RD and WR Low. When the peripheral is in the reset state, the Control register containing the reset bit can be read without writing the pointer. Reading back the FIO and CIO Control register should yield a 01_H.

The SCC's Read cycle can be verified by reading the bits in RR0. Bits D_2 and D_6 are set to 1 and bits D_0, D_1, and D_7 are 0. Bits D_3 - D_5 reflect the input pins DCD, SYNC, and CTS, respectively.

WRITE CYCLE VERIFICATION

This Write cycle can be checked by writing to a register and reading back the results. Both the CIO and FIO must have their reset bits cleared by writing 00_H to their Control registers and reading back the result. The SCC can be checked by writing and reading to an arbitrary read/write register, for example, the Time Constant register (WR12 or WR13).

INTERRUPT ACKNOWLEDGE CYCLE VERIFICATION

Verifying an Interrupt Acknowledge (INTACK) cycle consists of several steps. First, the peripheral makes an Interrupt Request (INT) to the CPU. When the processor is ready to service the interrupt, it initiate an Interrupt Acknowledge (INTACK) cycle. This peripheral then puts an 8-bit vector on the bus, and the 68000 uses that vector to get to the correct service routine. This test checks the simplest case.

First, load the Interrupt Vector register with a vector, disable the Vector Includes status (VIS), and enable interrupts (IE = 1, MIE = 1, IEI = 1). Disabling VIS guarantees that only one vector is put on the bus. The address of the service routine corresponding to the 8-bit vector number must be loaded into the 68000's vector table.

Initiating an interrupt sequence in the FIO and CIO can be accomplished by setting one of the interrupt pending (IP) bits and seeing if the 68000 jumps to the service routine (setting a breakpoint at the beginning of the service routine is an easy way to check if this has happened).

Initiating an interrupt sequence in the SCC is not quite as simple because the IP bits are not as accessible to the user. An interrupt can be generated indirectly via the CTS pin by enabling the following : CTS IE (WR15 20), EXT INT EN (WR1 01), and MIE (WR9 08). Any transition on the CTS pin can initiate the interrupt sequence. The interrupt can be re-enabled by RESET EXT/STATUS INT (WR0 10) and RESET HIGHEST IUS (WR0 38).

CONCLUSION

SGS-THOMSON Z8500 family of nonmultiplexed Address/Data bus peripherals can interface easily with the SGS-THOMSON TS68000 and provide all the support required in a high-performance microprocessor system. The many features offered by the SCC, FIO and CIO solve many system design problems by making interfacing to the external world easy. These intelligent peripherals also greatly enhance the system performance by relieving the CPU of many burdensome overhead tasks. Additionally, the powerful interrupt structure allows the TS68000 to use vectors and reduce interrupt response time.

NOTES

NOTES

NOTES

NOTES

NOTES

SALES OFFICES

EUROPE

DENMARK

2730 HERLEV
Herlev Torv, 4
Tel. (45-42) 94.85.33
Telex: 35411
Telefax: (45-42) 948694

FINLAND

LOHJA SF-08150
Karjalankatu, 2
Tel. 12.155.11
Telefax. 12.155.66

FRANCE

94253 GENTILLY Cedex
7 - avenue Gallieni - BP. 93
Tel.: (33-1) 47.40.75.75
Telex: 632570 STMHQ
Telefax: (33-1) 47.40.79.10

67000 STRASBOURG
20, Place des Halles
Tel. (33) 88.75.50.66
Telex. 870001F
Telefax: (33) 88.22.29.32

GERMANY

6000 FRANKFURT
Gutleutstrasse 322
Tel. (49-69) 237492
Telex: 176997 689
Telefax: (49-69) 231957
Teletex: 6997689=STVBP

8011 GRASBRUNN
Bretonischer Ring 4
Neukeferloh Technopark
Tel.: (49-89) 46006-0
Telex: 528211
Telefax: (49-89) 4605454
Teletex: 897107=STDISTR

3000 HANNOVER 1
Eckenerstrasse 5
Tel. (49-511) 634191
Telex 175118418
Teletex: 5118418 csfbeh
Telefax: (49-511) 633552

8500 NÜRNBERG 20
Erlenstegenstrasse, 72
Tel.: (49-911) 59893-0
Telex: 626243
Telefax: (49-911) 5980701

5200 SIEGBURG
Frankfurter Str. 22a
Tel. (49-2241) 660 84-86
Telex: 889510
Telefax: (49-2241) 67584

7000 STUTTGART
Oberer Kirchhaldenweg 135
Tel. (49-711) 692041
Telex: 721718
Telefax: (49-711) 691408

ITALY

20090 ASSAGO (MI)
V.le Milanofiori - Strada 4 - Palazzo A/4/A
Tel. (39-2) 89213.1 (10 linee)
Telex: 330131 - 330141 SGSAGR
Telefax: (39-2) 8250449

40033 CASALECCHIO DI RENO (BO)
Via R. Fucini, 12
Tel. (39-51) 591914
Telex: 512442
Telefax: (39-51) 591305

00161 ROMA
Via A. Torlonia, 15
Tel. (39-6) 8443341
Telex: 620653 SGSATE I
Telefax: (39-6) 8444474

NETHERLANDS

5652 AR EINDHOVEN
Meerenakkerweg 1
Tel.: (31-40) 550015
Telex: 51186
Telefax: (31-40) 528835

SPAIN

08021 BARCELONA
Calle Platon, 6 4th Floor, 5th Door
Tel. (34-3) 4143300-4143361
Telefax: (34-3) 2021461

28027 MADRID
Calle Albacete, 5
Tel. (34-1) 4051615
Telex: 27060 TCCEE
Telefax: (34-1) 4031134

SWEDEN

S-16421 KISTA
Borgarfjordsgatan, 13 - Box 1094
Tel.: (46-8) 7939220
Telex: 12078 THSWS
Telefax: (46-8) 7504950

SWITZERLAND

1218 GRAND-SACONNEX (GENEVA)
Chemin Francois-Lehmann, 18/A
Tel. (41-22) 7986462
Telex: 415493 STM CH
Telefax: (41-22) 7984869

UNITED KINGDOM and EIRE

MARLOW, BUCKS
Planar House, Parkway
Globe Park
Tel.: (44-628) 890800
Telex: 847458
Telefax: (44-628) 890391

SALES OFFICES

AMERICAS

BRAZIL

05413 SÃO PAULO
R. Henrique Schaumann 286-CJ33
Tel. (55-11) 883-5455
Telex: (391)11-37988 "UMBR BR"
Telefax : 11-551-128-22367

CANADA

BRAMPTON, ONTARIO
341 Main St. North
Tel. (416) 455-0505
Telefax: 416-455-2606

U.S.A.

NORTH & SOUTH AMERICAN
MARKETING HEADQUARTERS
1000 East Bell Road
Phoenix, AZ 85022
(1)-(602) 867-6100

SALES COVERAGE BY STATE

ALABAMA
Huntsville - (205) 533-5995

ARIZONA
Phoenix - (602) 867-6340

CALIFORNIA
Santa Ana - (714) 957-6018
San Jose - (408) 452-8585

COLORADO
Boulder (303) 449-9000

ILLINOIS
Schaumburg - (708) 517-1890

INDIANA
Kokomo - (317) 459-4700

MASSACHUSETTS
Lincoln - (617) 259-0300

MICHIGAN
Livonia - (313) 462-4030

NEW JERSEY
Voorhees - (609) 772-6222

NEW YORK
Poughkeepsie - (914) 454-8813

NORTH CAROLINA
Raleigh - (919) 787-6555

TEXAS
Carrollton - (214) 466-8844

FOR RF AND MICROWAVE POWER TRANSISTORS CONTACT
THE FOLLOWING REGIONAL OFFICE IN THE U.S.A.

PENNSYLVANIA
Montgomeryville - (215) 362-8500

ASIA / PACIFIC

AUSTRALIA

NSW 2027 EDGECLIFF
Suite 211, Edgecliff centre
203-233, New South Head Road
Tel. (61-2) 327.39.22
Telex: 071 126911 TCAUS
Telefax: (61-2) 327.61.76

CHINA

BEIJING
Liason Office
Beijing No. 5 Semiconductor
Device Factory
14 Wu Lu Tong Road
Da Sheng Man Wai
Tel. (861) 2024378
Telex 222722 STM CH

HONG KONG

WANCHAI
22nd Floor - Hopewell centre
183 Queen's Road East
Tel. (852-5) 8615788
Telex: 60955 ESGIES HX
Telefax: (852-5) 8656589

INDIA

NEW DELHI 110001
LiasonOffice
62, Upper Ground Floor
World Trade Centre
Barakhamba Lane
Tel. 3715191
Telex: 031-66816 STMI IN
Telefax: 3715192

MALAYSIA

PULAU PINANG 10400
4th Floor - Suite 4-03
Bangunan FOP-123D Jalan Anson
Tel. (04) 379735
Telefax (04) 379816

KOREA

SEOUL 121
8th floor Shinwon Building
823-14, Yuksam-Dong
Kang-Nam-Gu
Tel. (82-2) 553-0399
Telex: SGSKOR K29998
Telefax: (82-2) 552-1051

SINGAPORE

SINGAPORE 2056
28 Ang Mo Kio - Industrial Park 2
Tel. (65) 4821411
Telex: RS 55201 ESGIES
Telefax: (65) 4820240

TAIWAN

TAIPEI
12th Floor
571, Tun Hua South Road
Tel. (886-2) 755-4111
Telex: 10310 ESGIE TW
Telefax: (886-2) 755-4008

JAPAN

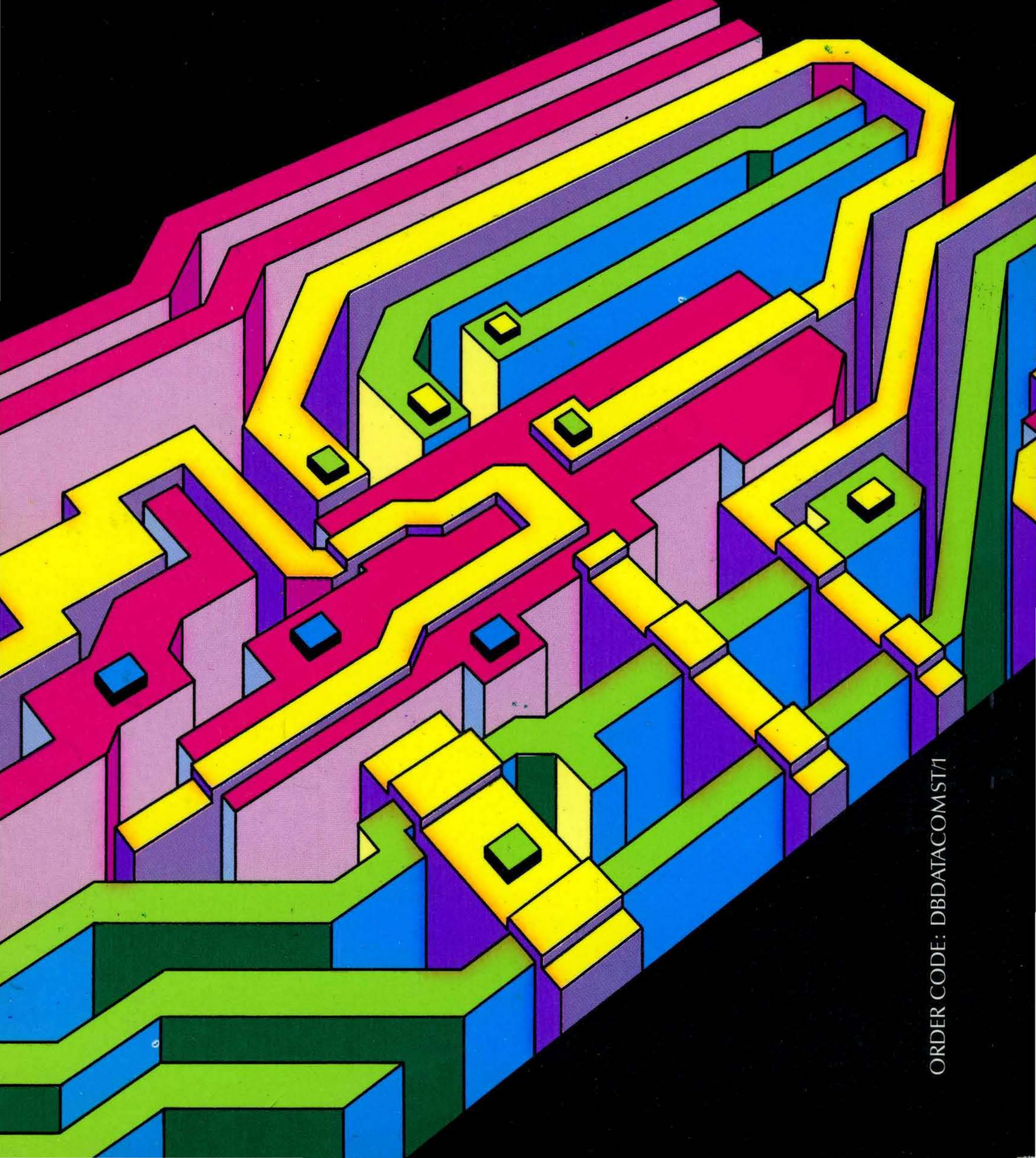
TOKYO 108
Nisseki - Takanawa Bld. 4F
2-18-10 Takanawa
Minato-Ku
Tel. (81-3) 3280-4121
Telefax: (81-3) 3280-4131

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsability for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may results from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all informations previously supplied.

Cover design by Keit & Koppel, Segrate, Italy
Typesetting and layout on Desk Top Publishing
by AZIMUT, Hénin Bt., France

Printed by Garzanti, Cernusco S/N, Italy

SGS-THOMSON Microelectronics GROUP OF COMPANIES
Australia - Brazil - China - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands -
Singapore - Spain - Sweden - Switzerland - Taiwan - United Kingdom - U.S.A.



ORDER CODE: DBDATACOMST/1