GC30-3004-0

Systems

IBM 3705 Communications Controller Principles of Operation

IBM

Preface

This reference publication contains the hardware operation and programming requirements of the IBM 3705 Communications Controller.

It is intended for any user of the 3705 who is attempting to write or modify a 3705 control program. The prerequisites for this manual are: (1) an understanding of basic teleprocessing operations, (2) a thorough knowledge of System/360 and System/370 channel operation, and (3) the publication *Introduction to the IBM 3705 Communications Controller*, GA27-3051. As a corequisite the reader is directed to the *IBM 3705 Communications Controller Assembler Language* manual, GC30-3003.

Descriptions of specific terminal devices and line interfaces used with the 3705 appear in separate publications. Other IBM publications concerning the 3705 Communications Controller are identified and described in the *IBM System/360 Bibliography*, GA22-6822.

This manual is divided into ten chapters and four appendixes.

Chapter 1: Introduction
Provides a general description of the 3705 and its functional units.

Chapter 2: System Structure
Describes the registers, interrupt scheme, and levels of the control program.

Chapter 3: Storage and Line Addressing
Describes the basic storage addressing procedure and the format for addressing the individual communication lines.

Chapter 4: Instruction Set
Describes each of the 3705 machine instructions with their format and condition codes.

Chapters 5 through 9:

Provide a basic understanding of the operation and programming requirements of the Central Control Unit, the Type 1 and Type 2 Communication Scanners, and the Type 1 and Type 2 Channel Adapters.

Chapter 10: Control Panel
Describes the lights and switches of the control panel pertinent to the operation and modification of the control program.

The appendixes contain (A) External register functions, (B) Input/Output instruction bit definitions, (C) Input/Output instruction summary, and (D) Line Interface Base and Line Set types.

First Edition (April 1972)
Changes are periodically made to the information herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

This manual has been prepared by the IBM Systems Development Division, Publications Center, Department E01, P.O. Box 12275, Research Traingle Park, North Carolina 27709. A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be sent to the above address.

Contents

Figures

Abbreviations

ABAR	attachment buffer address register
ACR	add character register instruction
ACR	abandon call and retry
ACU	automatic calling unit
AHR	add halfword register instruction
ALU	arithmetic logic unit
AR	add register instruction
ARI	add register immediate instruction
ATT	attention
B	branch instruction
BAL	branch and link instruction
BALR	branch and link register instruction
BB	branch on bit instruction
BCL	branch on C latch instruction
BCT	branch on count instruction
BSC	binary synchronous communication
BZL	branch on Z latch instruction
CA	channel adapter
CACHKR	channel adapter check register
CACR	channel adapter control register
CADB	channel adapter data buffer
CAMR	channel adapter mode register
CASNSR	channel adapter sense register
CASTR	channel adapter status register
CBODR	channel bus out diagnostic register
CCDS	control command data status register
CCR	compare character register instruction
CCU	central control unit
CE	channel end
CHR	compare halfword register instruction
CMDR	channel adapter command register
COS	call originate status
CPU	central processing unit
CR	compare register instruction
CRC	cyclic redundency check
CRI	compare register immediate instruction
CRQ	call request
CSAR	cycle steal address register
CTDR	channel tag diagnostic register
CTRL	control
CW	control word
CWAR	control word address register
CWCNTR	control word byte count register
DE	device end
DLO	data line occupied
DPR	digit present
DTR	data terminal ready
ESC	emulator subchannel
EXIT	exit instruction
FDX	full duplex
HDX	half duplex
HEX	hexadecimal
HIO	halt I/O
IAR	instruction address register
IC	insert character instruction
ICT	insert character and count instruction
ICW	interface control word
IN	input instruction
INCWAR	in-bound control word address register
I/O	input/output
IPL	initial program load
IPR	interrupt priority register
L	load instruction
LA	load address instruction
LAR	lagging address register
LCD	line control identifier
LCOR	load character with offset register instruction
LCR	load character register instruction
LH	load halfword instruction
LHOR	load halfword with offset register instruction
LHR	load halfword register instruction
LIB	line interface base
LOR	load with offset register instruction
LR	load register instruction
LRI	load register immediate instruction
NCR	and character register instruction
NHR	and halfword register instruction
NR	and register instruction
NRI	and register immediate instruction
NSC	native subchannel
OCR	or character register instruction
OHR	or halfword register instruction
OR	or register instruction
ORI	or register immediate instruction
OUT	output instruction
OUTCWAR	out-bound control word address register
PCF	primary control field
PCI	program controlled interrupt
PDF	parallel data field
PND	present next digit
RA	register and immediate address operation
RE	register and external register operation
RI	register and immediate operand operation
ROS	read-only-storage
RR	register to register operation
RS	register and storage operation
RSA	register and storage with count operation
RT	branch/operation
RTS	request to send
SAR	storage address register
SCF	secondary control field
SCR	subtract character register instruction
SDF	serial data field
SDR	storage data register

SHR	subtract halfword register instruction
SIO	start I/O
SM	status modifier
SR	subtract register instruction
SRI	subtract register immediate instruction
ST	store instruction
STC	store character instruction
STCT	store character and count instruction
STH	store halfword instruction
SVC	supervisor call
TAR	temporary address register
TIC	transfer in channel
TIO	test I/O
TRM	test register under mask instruction
UC	unit check
UE	unit exception
XCR	exclusive or character register instruction
XHR	exclusive or halfword register instruction
XR	exclusive or register instruction
XRI	exclusive or register immediate instruction

Chapter 1: Introduction

The 3705 Communications Controller is a transmission control unit with processing capabilities that offer advantages not available in other IBM transmission control units. A control program residing in 3705 storage performs many of the functions previously performed by the central processing unit. The assumption of these functions by the controller increases the availability of the CPU to process other programs and to perform more involved message-processing functions for the teleprocessing system.

The controller is priority-interrupt driven. This allows the control program to handle service requests at five different priority levels.

The communications controller performs, under program control, the normal transmission control unit functions such as line-control, character recognition, line timeout, character assembly and disassembly, and redundancy checking. The control program can also (1) handle all polling and addressing of communications lines to determine if a line is ready to send or receive data, (2) take over data link control, (3) add framing characters to the beginning and end of blocks of data, and (4) translate from line code into code recognizable to a message-processing program, and vice versa. Most error recovery procedures can be handled by the communications controller, thus relieving the CPU of a time-consuming and storage-consuming telecommunications function. The control program can also provide dynamic buffering for incoming data and basic message-processing functions.

Host System Interface

The 3705 may be attached to a System/360, Model 30, 40, 50, 65, 67, 75, or 195, or System/370 Models 135 through 195 with the proper channel adapter. With a Type 2 Channel Adapter, only one subchannel address is required, and the interface may be connected to a byte multiplexer, block multiplexer, or selector channel. The Type 1 Channel Adapter operates only on a byte multiplexer channel and may require more than one subchannel address, depending on the mode of operation.

System Support Programs

IBM provides control programs that schedule and control all 3705 system resources. Also provided are system support programs. These are host processor programs used primarily to generate or assemble a user's control program and to provide IPL and dump facilities for the controller. The support programs operate under control of the host supervisor.

Description

The 3705 is available in 20 models, based on the amount of storage and physical communication-line attachment capability. Each model is specified by a letter and a number, such as Model C3. The letter indicates the maximum line attachment capability, and the number indicates the amount of storage. All C-models, for example can attach up to 256 lines, and all 3-models have 80K of storage.

Figure 1 lists the models of the 3705 with their line attachment capability and storage capacity. Note that the line attachment indicated is the physical line attachment. The actual number of lines that the 3705 can support depends on many factors, such as line speed and throughput capability of the control program.

The 'A' model of the 3705 is a standalone basic module that contains a Central Control Unit, a control panel, at least 16K of storage and provisions for a channel adapter, a two-channel switch, a communication scanner, and line interface bases (LIB) and line sets to attach up to 64 communication lines.

As the models increase in size, the available hardware options also increase. This is done by adding expansion modules to the basic module. For example, all 'B' models are made up of one basic module and one expansion module; 'C' models--one basic and two expansion modules; and 'D' models--one basic and three expansion modules.

Each expansion module is an add-on frame that can contain either 0, 32, or 64K bytes of additional storage and a Type 2 Communication Scanner capable of supporting up to six LIBs. The first expansion module can also contain a Type 2 Channel Adapter and a two-channel switch feature. Figure 2 shows the maximum configuration of the major components of the controller hardware.

The following paragraphs briefly describe the functions of each hardware component.

Storage

The 3705, like most central processing units, contains its own internal storage array. This storage provides a residence for the 3705 control program and a temporary storage area for data as it is being assembled or disassembled in preparation for transfer to the channel or a terminal.

The range of storage can be from 16K bytes (16,384) to 240K bytes (245,760) in 32K byte increments. This storage is completely separate from host

Model	*Maximum Line Attachment*	*Amount Of Storage*
A1	64	16K
A2	64	48K
B1	160	16K
B2	160	48K
B3	160	80K
B4	160	112K
C1	256	16K
C2	256	48K
C3	256	80K
C4	256	112K
C5	256	144K
C6	256	176K
D1	352	16K
D2	352	48K
D3	352	80K
D4	352	112K
D5	352	144K
D6	352	176K
D7	352	208K
D8	352	240K

Figure 1. 3705 Models

processor storage and has a storage cycle time of 1.2 microseconds.

Bytes of storage are handled separately or grouped together in fields. A *halfword* is a group of two consecutive bytes and is the basic building block of instructions. A *word* is a group of four consecutive bytes. The location of any field or group of bytes is specified by the address of its leftmost byte.

Central Control Unit

The Central Control Unit (CCU) contains the circuits and data flow paths to execute the instruction set, and to control storage, the communication scanners, and the channel adapters. The CCU operates under control of the programs residing in storage.

3705 Adapters

Two different adapters are required within the communications controller to interface the Central Control Unit with the host channel and the communication lines. These are the communication scanner (interface between CCU and communication line), and the channel adapter (interface between CCU and host channel). Both the communication scanners and the channel adapters (CA) for the 3705 come in two versions. The Type 1 Scanner and Type 1 CA have low functional hardware capabilities and require more program control than the Type 2 adapters. The Type 2 Scanner and Type 2 CA have more functional hardware capabilities and therefore require less program control while providing increased performance.

Figure 3 shows all possible combinations of communication scanners and channel adapters, including the maximum number that can be installed for each model of the 3705.

Channel Adapters

Two types of channel adapters (CAs) are available for the 3705--the Type 1 CA and the Type 2 CA. The Type 1 CA provides local attachment to a System/360 or System/370 byte multiplexer channel. It contains the hardware circuits necessary to assist the program in emulating an IBM 2701, 2702, or 2703 as well as to allow native 3705 attachment. When attached in the emulation mode, multiple subchannel addresses are required (one for each device address and one for native mode IPL).

The Type 2 CA provides local attachment to a System/370 byte multiplexer, block multiplexer, or selector channel. With the Type 2 CA, the 3705 appears as a single control unit on the host processor channel and uses a single channel address. The Type 2 CA operates in the native mode only.

All data transfer between the Type 2 CA and 3705 storage is by cycle steal. That is, when the CA has data to put in storage, it steals (under hardware control) the necessary machine cycles to transfer the data. Data transfers from storage to the channel adapter are also by cycle steal. A cycle steal operation is accomplished by hardware circuits and does not affect the logical operation of the program.

A second channel adapter can be installed in the first expansion module of a 3705 Model B, C, or D. When two Type 2 Channel Adapters are installed, they are referred to as Type 2 CA-1 and Type 2 CA-2. With the proper programming support, the two adapters can operate concurrently. Figure 3 shows the possible combinations of channel adapters. Data transferred between the System 360 or System 370 processor and the 3705 is transparent to either type of channel adapter.

Two-Channel Switch Feature

A two-channel switch can be installed for both the Type 1 and Type 2 Channel Adapters. With this feature, the communications controller can be attached to two host processor channels through a single channel adapter. (Both channels can be on the same CPU, or they can be on two different CPUs.) However, only one of the channels can be enabled for operation at a time. The enabled channel is selected by means of a manual switch on the 3705 control panel.

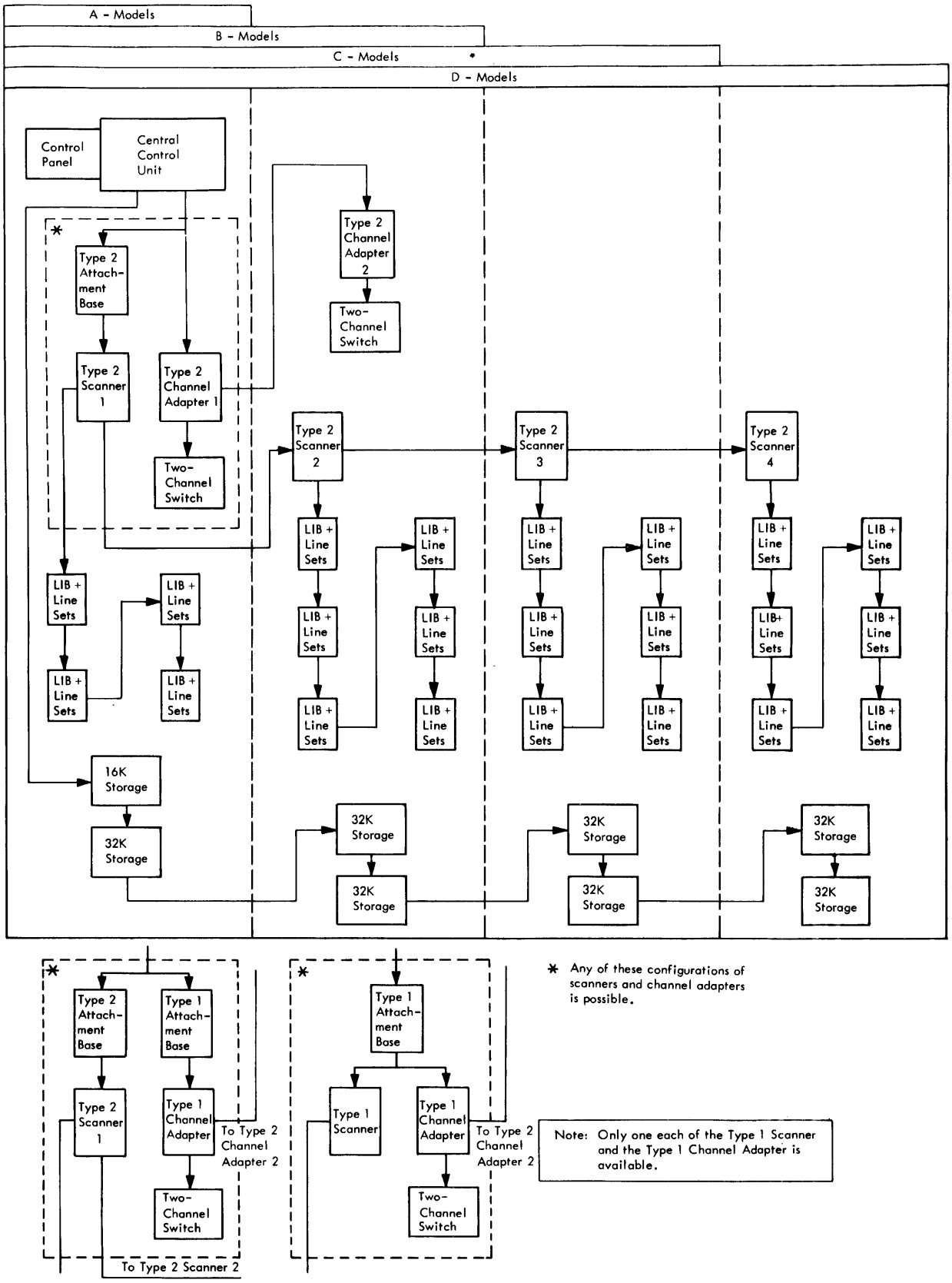


Figure 2. Maximum Configuration of Major 3705 Components

Combination	Number in 3705	Models
Type 1 CA Type 1 Scanner	1 1	All Models
Type 1 CA Type 2 Scanner	1 1 - 4	1 Type 2 Scanner - all models 2, 3, or 4 Type 2 Scanners - B, C, or D models
Type 2 CA Type 2 Scanner	1 - 2 1 - 4	1 of each - all models More than one of either or both - B, C, and D models
Type 1 CA Type 2 CA Type 1 Scanner	1 1 1	B, C, & D Models
Type 1 CA Type 2 CA Type 2 Scanner	1 1 1 - 4	B, C, & D Models

Figure 3. Channel Adapter/Communication Scanner Combinations

If two channel adapters are installed, both can have a two-channel switch. This allows the controller to be attached to four CPU channels, of which two (one interface on each channel adapter) can operate concurrently.

Communication Scanners

Two types of communication scanners are available for the 3705--the Type 1 Scanner and the Type 2 Scanner. The communication scanners provide the interface between the communication-line attachment hardware (line interface bases and line sets) and the Central Control Unit via the appropriate attachment base. The primary function of the scanner is to periodically scan the hardware associated with each communication line for service requests.

The Type 1 Scanner supports four LIBs with a maximum capability of 64 half-duplex lines and can transfer only one bit at a time to the interface. The LIBs within this scanner are specified as LIB-1, LIB-2, LIB-3, and LIB-4. Only one Type 1 Scanner can be installed in a 3705 and also prohibits installation of a Type 2 Scanner. This limits the communications controller to a maximum of 64 lines.

The Type 2 Scanner supports either four or six LIBs with a maximum capability of 64 or 96 half-duplex lines respectively. This scanner transfers a full byte to or from the interface. Depending on the model of 3705, from one to four Type 2 Scanners can be installed. These scanners are specified as Scanner-1, Scanner-2, Scanner-3, and Scanner-4. Scanner-1 is located in the basic module and can support up to four LIBs with 64 lines. Scanners-2, -3, and -4 are located in the expansion modules, and each can support up to six LIBs with 96 lines. The LIBs are specified as LIB-1, LIB-2, ...LIB-5, and LIB-6 in each scanner, with the exception of Scanner-1, which has only four LIBs.

Only one type of communication scanner (Type 1 or Type 2) can be installed within a 3705 Communications Controller. Each scanner supports both synchronous and asynchronous communication lines operating at different line speeds. The line type (BSC, Start-Stop, etc.), character length, bit clocking mechanism (business machine or modem), installed business machine clock speed, and interrupt priority are selected by the program for each line interface.

Attachment Bases

An attachment base is a required feature for support of the 3705 adapters. Two types of attachment bases are available: the Type 1 Attachment Base and the Type 2 Attachment Base.

The Type 1 Attachment Base provides common interface controls to the Central Control Unit for both the Type 1 Scanner and the Type 1 Channel Adapter.

The Type 2 Attachment Base provides common interface controls to the Central Control Unit and line addressing controls for the Type 2 Scanners.

One or both of the attachment bases are required, depending on the type of scanner and channel adapter installed in the 3705. The requirements are as follows:

Hardware Installed	*Attachment Base Required*
Type 1 Scanner/Type 1 CA	Type 1 Attachment Base
Type 2 Scanner/Type 1 CA	Type 1 & Type 2 Attachment Base
Type 2 Scanner/Type 2 CA	Type 2 Attachment Base

Line Interface Bases and Line Sets

Communication lines to and from the teleprocessing terminals or devices are attached to the 3705 through a Line Interface Base (LIB). The primary functions of the LIB are to drive and terminate all signals between the communication scanners and the line sets and to provide bit clocking.

The line interface base is transparent to the data transferred and has no effect on the control program except for bit clock control.

The number and type of LIBs vary according to the system configuration. As many as 22 LIBs can be installed in a 3705. The A models, containing either a Type 1 or Type 2 Communication Scanner, are capable of supporting four LIBs. The B, C, or D models, containing Type 2 Communication Scanners, can support up to 10, 16, or 22 LIBS respectively (four LIBs in the basic module and six LIBs in each of the expansion modules).

Four different LIB types are available to meet the needs of a wide variety of line and terminal types. Each LIB type operates identically and is controlled by the communication scanner that it is attached to. However, the design of each LIB type differs in order to support the many line sets and line configurations that can be attached to them. Refer to Appendix D for a description of the individual LIB and line set types.

The *line set* is the hardware connection between the LIB and the communication line. A given line set type may support attachment of many different terminals and devices; therefore, different data sets or modems may be required. A single line set provides the interface for one or two communication lines, depending on the type of interface.

Control Panel

The control panel contains the switches and lights necessary for manual control of the 3705. Some of the functions provided by the control panel are the ability to store and display information in storage and registers; the control and indication of power; indications of the controller status; operator/controller communication controls; and diagnostic controls.

General Programming Concepts

The control program that resides in 3705 storage controls the transfer of data as it passes through the controller between the terminals in the teleprocessing network and the host processor.

The primary functions of the control program are related to transmitting and receiving data. But it can also edit and process the data as it passes through the controller.

System Data Flow

In performing its functions, the control program interacts with the communication scanners and the channel adapters to control the flow of data through the teleprocessing system.

Data entered at a terminal is received by the line set and line interface base. The communication scanner recognizes that service is required and receives data from the LIB. The program places the data in storage, where it is then available to the channel adapter to be sent to the host processor channel.

When data is to be transmitted from the host processor to a terminal, the process is reversed. The host channel sends the data to the controller channel adapter which, under hardware or program control depending on the type adapter, places the data in storage. An interrupt request to send the data to the LIB and finally to the terminal is then signaled to the control program.

Some processing of the data may be accomplished while the data is in storage. Interaction between the controller adapters is through interrupts and input/output instructions. Figure 4 illustrates the data flow to and from the terminal and the CPU through the various components of the 3705.

Input/Output Instructions

The 3705 uses the concept of input and output instructions. These instructions are the primary link between the hardware and the control program.

All control information and data as it enters or exits the controller passes through the CCU and adapter external registers. These external registers are not directly accessible by the control program; therefore, input and output instructions are used to obtain or change the external register contents. When the control program executes an input instruction, the contents of the specified external register are loaded into a general register. The program now has direct access to that information and can act accordingly.

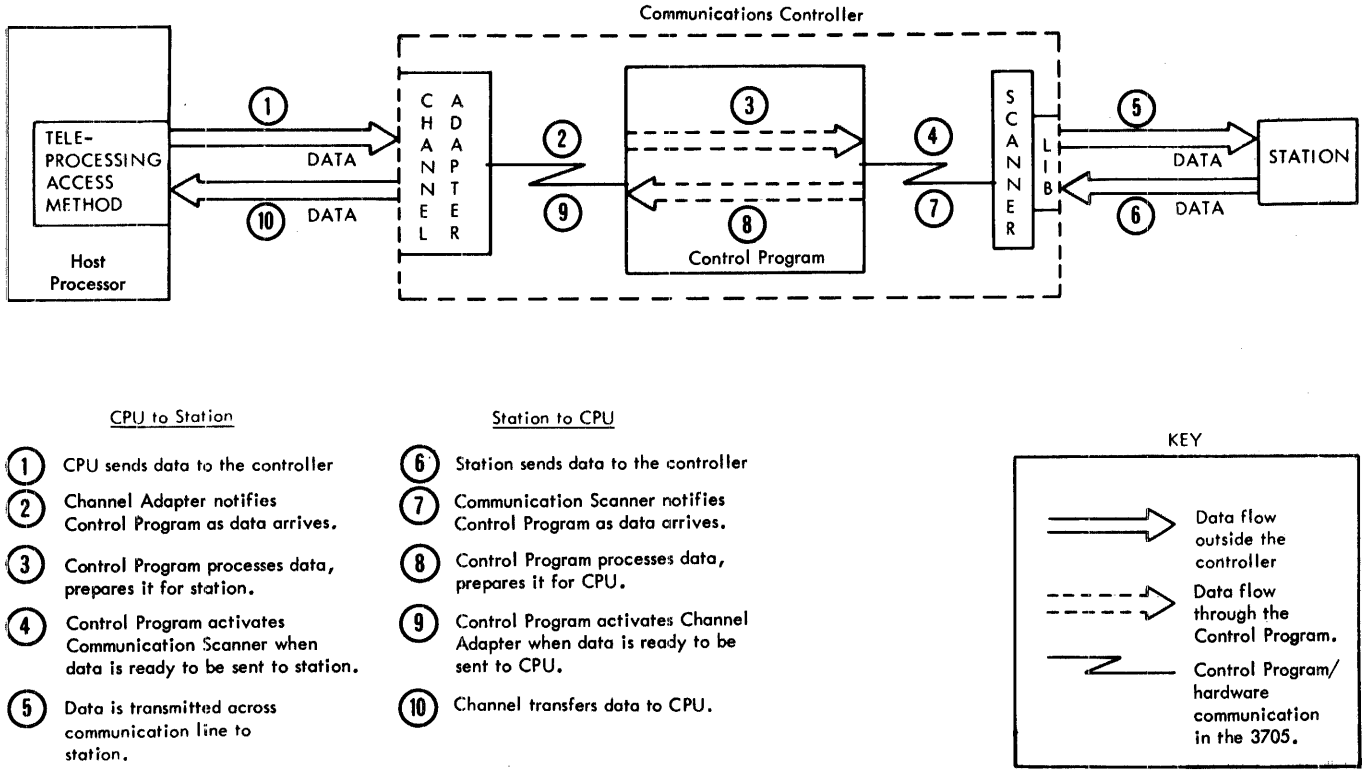


Figure 4. System Data Flow

In the same manner, the control program can load a general register with control information or data for a particular adapter. When that general register is used with the specified external register and an output instruction executed, the contents of the general register are transferred into the external register. The subject adapter can then take the the action specified by the settings of the register bits.

Any data-transfer and adapter-control operations that the control program can perform within the controller are accomplished through input and output instructions.

Chapter 2: System Structure

This chapter describes the registers, interrupt scheme, and program levels used in the 3705. The user needs a thorough understanding of these facilities in order to program the 3705 efficiently.

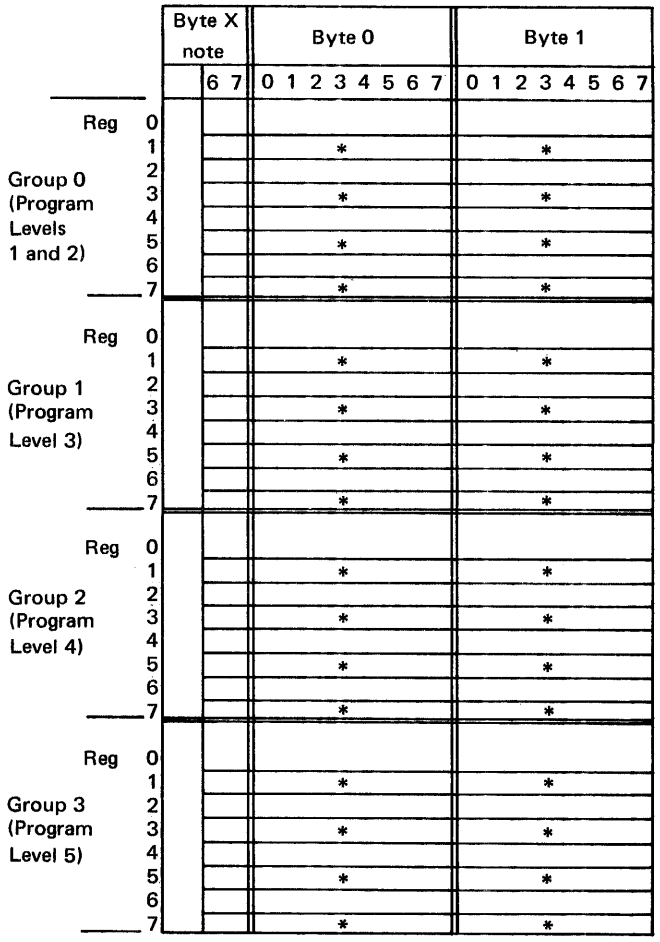
Registers

The controller has two types of registers -- general and external. These registers vary is size and location according to how they are used. They can range from one bit to eighteen bits. The following paragraphs give a brief description of the types, size, and usage of the registers.

General Registers

There are 32 general registers in the controller that are available for program use. These registers are located in a local storage array so as not to occupy usable storage locations. The basic size of each register is one halfword (16 bits). The bits are designated from left to right as byte 0, bits 0-7 and byte 1, bits 0-7. With Extended Addressing (see Chapter 3), each register contains 18 information bits. The 18 information bits are designated from left to right as byte X, bits 6-7; byte 0, bits 0-7; and byte 1, bits 0-7. Without Extended Addressing, byte X is not present, and any reference to it is ignored.

As shown in Figure 5, the 32 general registers are divided into 4 groups of 8 registers each. Each group is assigned to a specific program level, except for group 0, which is shared by program levels 1 and 2. (See *Program Levels* in this chapter.) This allows the control program at one level to be interrupted by another level without the need to save registers. The general registers are numbered 0-7 within each group. Only one group of general registers is active at a time—the group associated with the active program level. The registers within the currently active group are directly addressable with program instructions. The control program can gain access to the general registers in a nonactive group by specifying them as external registers in input and output instructions.



* Indicates selectable bytes of general registers

Note: This byte present only in machines with Extended Addressing

Figure 5. General Register Groups

Instruction Address Register

General register 0 in each group is the instruction address register (IAR). This register is an implied base register and contains the address of the next instruction to be executed for the associated program level. Register 0 of the active group is always incremented to point to the next sequential instruction before the current instruction is executed. In most cases, the next halfword is the next instruction to be executed. Sometimes, however, the contents of the instruction address register are changed as the result of the instruction being executed. Execution of a branch instruction, for example, can cause the IAR to be loaded with a storage address other than the one immediately following the current instruction. Refer to the individual instruction definitions in Chapter 4 for the precautions and results of the use of register 0.

External Registers

Each functional unit of the communications controller (Central Control Unit, communication scanner, channel adapter) has a number of hardware registers that

are used to store information required for communication between the control program and hardware circuits. These registers are called external registers.

Many of the external registers contain information pertinent to the operation of the hardware and/or the program. By using input instructions, the control program can load the contents of the external registers into a general register where it can operate on the data. Output instructions load an external register with the contents of the general register specified in the instruction.

The external registers that the control program can use are described in the Central Control Unit chapter and in each of the communication scanner and channel adapter chapters in this publication. Appendix A contains a summary of the external register addresses and functions and/or operations associated with each register.

Program Levels

The controller hardware has five operational program levels. Each program level operates similarly to a subroutine and is responsible for particular phases of the system operation. Figure 6 shows (1) the program levels in order of priority, (2) the interrupt requests causing entry, and (3) the general register group associated with each level. Program level 1 has the highest priority, and program level 5 has the lowest. Program levels 1, 2, 3, and 4 - referred to as interrupt program levels - provide the program interface between the hardware functional units and program level 5 (referred to as the background program level).

The following is a brief description of each of the five program levels.

- Background Program Level 5: This level is the lowest priority level and is active when none of the other four levels requires program cycles. Functions performed by this level should normally include (1) line management (host command interpretation, control of polling and addressing), (2) data and message handling, and (3) control command decoding and execution. This level can not interrupt another program level.
- Interrupt Program Level 4: The functions performed by this level should normally include (1) overall management of the system resources, (2) buffer management, (3) queue manipulation, and (4) the dispatching of program level 5 tasks. Certain program-controlled interrupt requests and the supervisor call request (generated when the Exit instruction is executed at level 5) are assigned to this program level.
- Interrupt Program Level 3: Level 3 should be used for most of the host processor/channel adapter interaction. This level handles interrupt requests from the channel adapter(s), the interval timer, the control panel, and the communication processing that can be deferred from level 2. In addition to hardware interrupts, level 3 can be called by program controlled interrupts (PCI) for initiating I/O and for any other services desired by the user. Level 3 interrupts are less critically time-dependent than those assigned to program level 2.
- Interrupt Program Level 2: Because of its high priority, this level only services interrupts from the communication lines for character or bit service. The control program can request a level 2 interrupt, but for the most part, it is entirely hardware interrupt driven. Normal operational interrupt requests from the communication scanner include (1) Type 1 Scanner bit service interrupts, (2) Type 1 Scanner character service interrupts, and (3) Type 2 Scanner character service interrupts. Only critically time-dependent processing should be done at this level.
- Interrupt Program Level 1: This is the highest-priority program level. It can be masked for channel adapter and scanner checks only when the Central Control Unit is in the test mode. A level 1 interrupt is invoked mainly to service "trouble" indications and is hardware-interrupt driven. Conditions that cause a level 1 request include all critical check conditions, such as (1) program checks, (2) addressing exceptions, (3) Central Control Unit (CCU) checks, and (4) scanner and channel adapter checks. Initial program load (IPL) procedures and address compare interrupts are also handled in this level.

  Because the same group of general registers are used for both level 1 and level 2, the level 1 program should take precautions to save the group 0 registers. A Store instruction with register 0 specified in the R field must be the first instruction executed in program level 1. For this special case, the Store instruction is modified (by hardware) to allow the contents of register 0 to be stored at the second operand location.

Interrupts

The Communications Controller operates in response to requests from either the control program or the hardware. Since these requests may have varying degrees of urgency, a priority system is used. Each program, CCU, and adapter request is assigned a particular priority level. A request for use of the controller

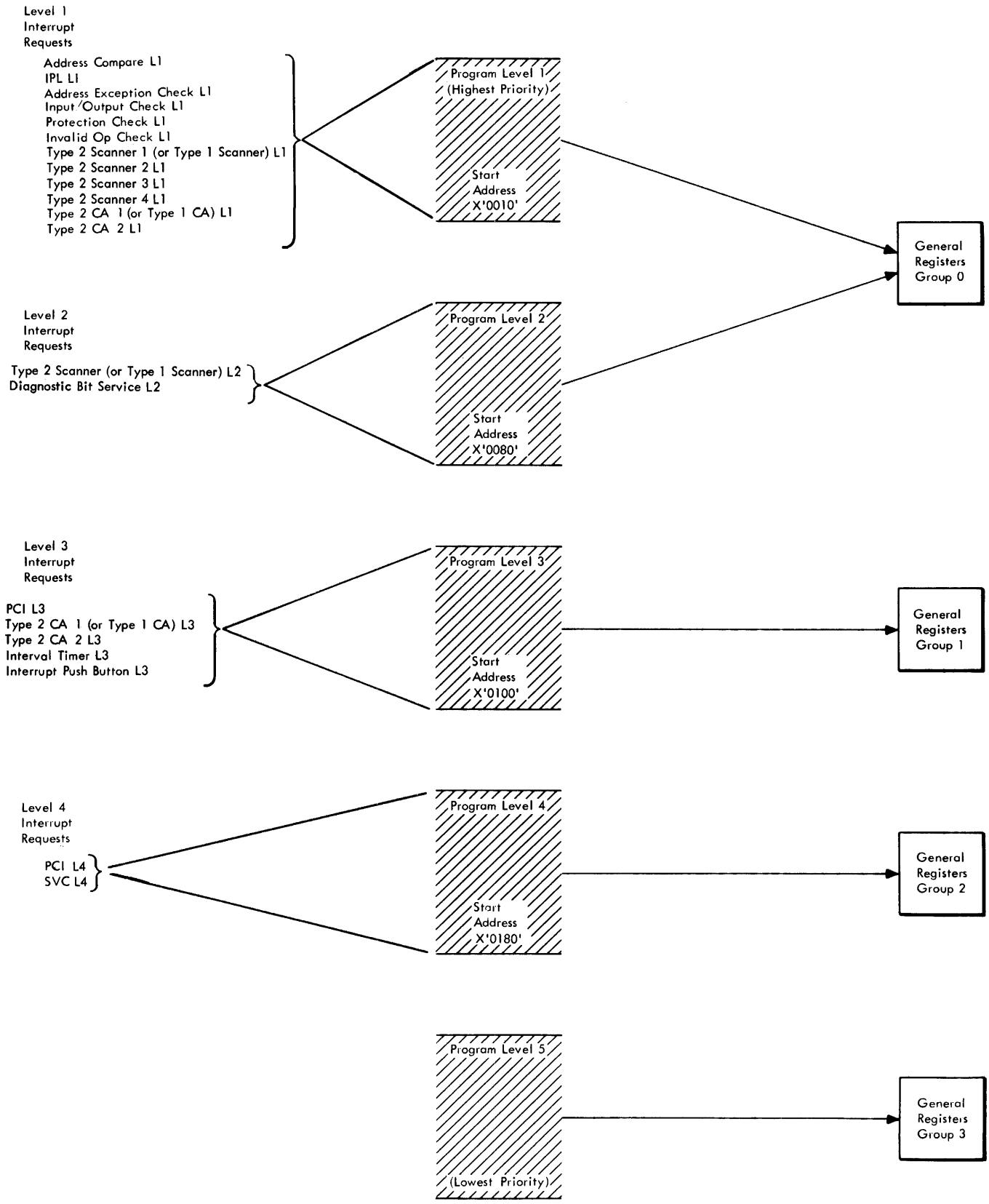


Figure 6. Program Levels

by the control program or hardware functions is called an *interrupt request*.

Each interrupt request is assigned to a *program level*. These program levels are numbered from one to five and determine the priority structure. Program level 1 has the highest priority, and the priority level decreases as the program number increases.

The machine priority controls determine when an interrupt can occur. If the interrupt request is to be allowed, the change from the active program level to the interrupting program level takes place immediately after completion of the current instruction. If several interrupt requests having different priorities are present at the same time, the one with the highest priority obtains use of the controller. When an interrupt request is granted use of the controller, it can be interrupted in that use by another request having a higher priority.

When an interrupt occurs, instruction execution at the lower-priority program level is suspended until instruction execution is completed at the higher-priority level. An interrupt to a specific program level prevents future interrupt requests assigned to either that level or to lower-priority program levels from causing another interrupt until the servicing of the first interrupt is complete.

The controller does not allow an interrupt if any of the following conditions exist.

1. A higher-priority interrupt request is present.
2. The program level to be interrupted is already entered ('interrupt entered' latch is on).
3. The interrupt request or the program level to be interrupted is masked.
4. A Type 2 Channel Adapter cycle-steal request exists.

At the time an interrupt is honored, the 'interrupt entered' latch for that program level turns on. The 'interrupt entered' latch is a hardware latch that signals the controller that the associated program level has been entered. As long as this latch is on, no other interrupt requests to that level are honored. The 'interrupt entered' latch is not turned off when its associated program level is interrupted by a higher priority level. It is turned off only by an Exit instruction or by a reset condition to the controller, therefore prohibiting interrupts that could destroy necessary information.

For an example of the interrupt facility, refer to Figure 7. The program at level 4 is being executed, and a level 2 interrupt request occurs (1). The controller hardware forces a branch to the starting address of program level 2 (2), and the program at that level begins servicing the interrupt. A level 3 interrupt request occurs (3), but it is not honored because program level 2 has a higher priority. When the level 2 interrupt has been serviced, the program executes an Exit instruction (4). The controller now allows the next highest-priority interrupt to be serviced. In this example, control is passed to program level 3 at its starting address (5). If, before the level 3 interrupt has been completely serviced another level 3 interrupt request occurs (6), no action is taken because the level 3 'interrupt entered' latch is on. However, as soon as program level 3 executes an Exit instruction (7), signaling the completion of processing and turning off its 'interrupt entered' latch, the controller can honor the second level 3 interrupt request and returns control to the starting address of program level 3 (8). When servicing of the latest interrupt is complete and the Exit instruction is executed (9), control is again passed to the highest-priority level that is able to execute. In this case program level 4 is the highest-priority level requiring service (10), so control is returned to it at the instruction following the point of interruption.

At times it may be desirable not to interrupt a particular operation by a higher-priority request. For such cases, a mask can be set to prevent interrupts to a particular program level. See *Masking Program Level Priorities* in Chapter 5.

When an interrupt occurs, instruction execution at that level begins with the instruction located at the starting address of that level. The starting address of each interrupt level is a permanently assigned storage location. The instruction(s) beginning at these fixed locations must direct the control program to the correct routine(s) to handle the particular interrupt. The remainder of the instructions for a particular program level need not be located in any specific storage area. In addition, some routines may be used by more than one program level. However in this case, the execution of that routine will be at the priority level of the currently active program level.

The starting addresses for the four program levels that can cause an interrupt are:

Level	*Starting Address (Hex)*
Program Level 1	X'0010'
Program Level 2	X'0080'
Program Level 3	X'0100'
Program Level 4	X'0180'

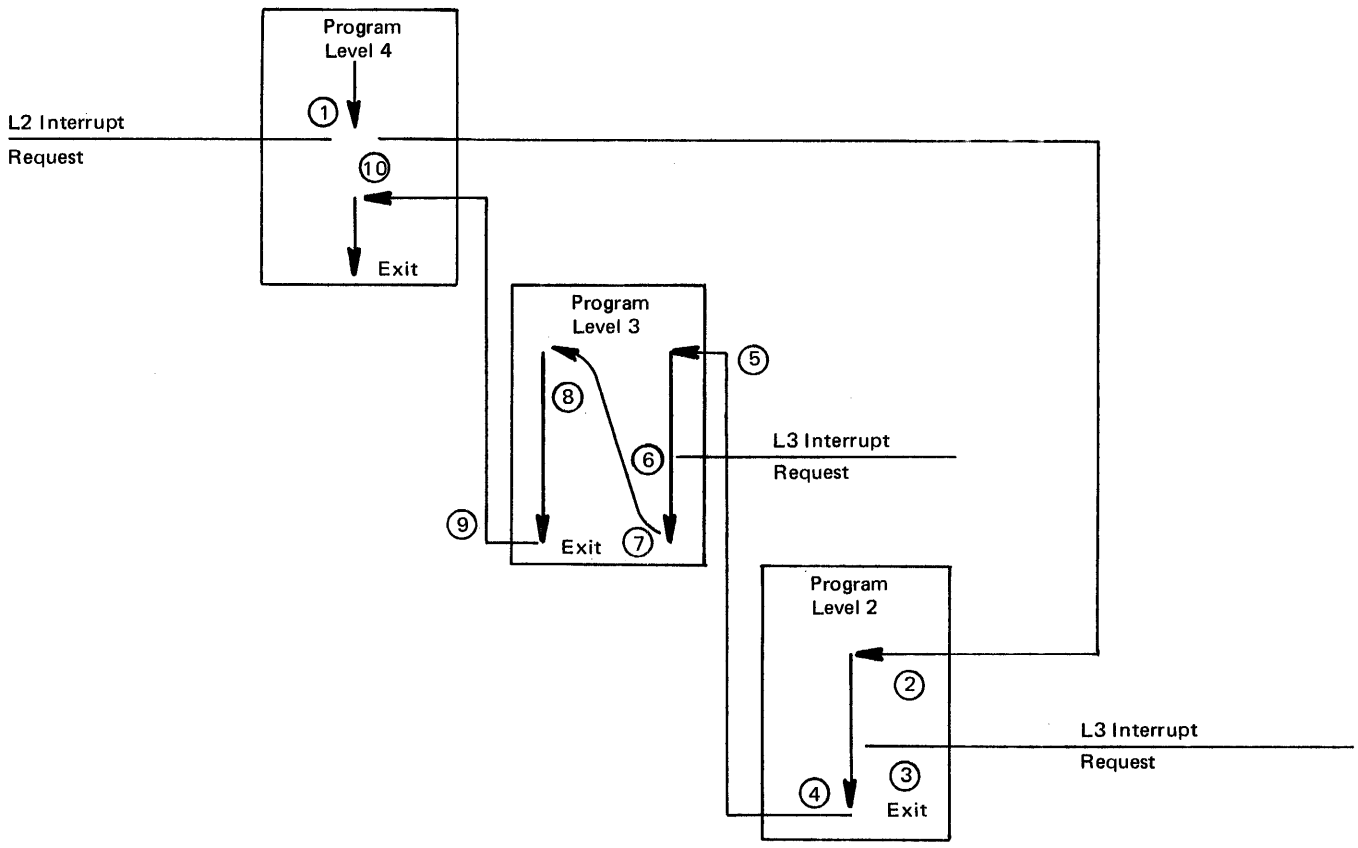


Figure 7. Interrupt Priority Example

When a program level has completed its interrupt servicing, it must execute an Exit instruction. The Exit instruction causes the 'interrupt entered' latch for that level to be reset and allows control to be passed to the next highest-priority program level requiring service.

When the Exit instruction is executed at program level 5, a supervisor call interrupt request for level 4 (SVC L4) is set. This is the only case in which program level 5 can generate an interrupt request.

Chapter 3: Storage and Line Addressing

This chapter describes the storage addressing scheme and the format for addressing the individual communication lines.

Storage Addressing

Byte locations in storage are consecutively numbered starting with 0; each number is considered the address of the corresponding byte. A group of bytes in storage is addressed by the high-order byte of the group. The number of bytes in the group is either implied or explicitly defined by the operation.

The basic addressing scheme uses a 16-bit binary address to accommodate a maximum byte address of 65,535. The two bytes of the halfword used for addressing are specified from left to right as byte 0 and byte 1. The bits within these two bytes are numbered left to right from 0 to 7. All general registers and the CCU external registers involved in addressing storage are two bytes in length and the bit positions correspond to the basic addressing scheme.

Figure 8 shows the storage address bit positions as they are used in the basic addressing scheme.

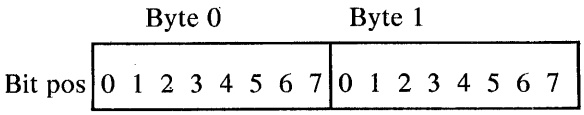


Figure 8. Storage Address Bit Positions (With Out Extended Addressing).

Storage addressing wraps at the maximum byte address of 65,535. This means that if, in the formation of a storage address, the binary representation of the address is *greater* than 16 bits in length, the actual address used will be only the address formed in the low-order 16 bits. For example, if the formation of a storage address uses a base address of X'A080' and a displacement of X'6010', the combined address does not include the carry of the high-order bit and therefore generates an effective address of X'0090'.

An addressing exception is recognized if the storage address is greater than the amount of installed storage locations but less than the point of storage wrap. In the basic controller, any attempt to address storage between 16,384 (or 49,152, whichever model is installed) and 65,535 results in an addressing exception. See *Program Checks* in Chapter 5.

Instructions and halfword operands must be located on integral halfword boundaries in storage. A boundary is called integral for a unit of information when its storage address is a multiple of the length of the unit in bytes. For example, words (four bytes) must be located in storage so that their address is a multiple of the number 4. A halfword (two bytes) must have an address that is a multiple of the number 2.

Storage addresses are expressed in binary form. In binary, integral boundaries for halfwords and words can be specified only by an address in which one or two low-order bits, respectively, are zero. For example, the integral boundary for a word is a binary address in which the two low-order positions are zero.

Extended Addressing

When a controller contains more than 48K bytes of storage, the basic 16-bit address structure is not sufficient. To address the storage positions above 64K, two additional address bits are required. These two bits are designated as byte X, bits 6 and 7, and allow address generation up to the maximum of 245,760 bytes. This addition of byte X is referred to as Extended Addressing. Figure 9 illustrates the storage address bit positions used by the affected registers with Extended Addressing.

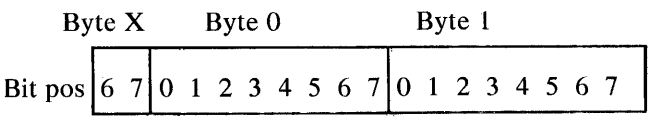


Figure 9. Storage Address Bit Positions (With Extended Addressing).

With Extended Addressing, the Central Control Unit data flow registers and all general registers are expanded to 18 bits. The two additional bits (byte X, bits 6-7) must be handled as an integral part of the register regardless of the address being operated on. The only exceptions are: (1) byte X is ignored for output instructions not involved in addressing storage and can be set to either 1 or 0, and (2) byte X is set to zeros for input instructions not involved in addressing storage.

Storage wrap and addressing exception also apply to Extended Addressing. However, with Extended Addressing, the point of storage wrap is 262,144, and a carry from this value wraps back to zero. The range of addresses that cause an addressing exception is from the maximum number of storage locations installed to the point of storage wrap. For example, if the installed storage is 180,224 bytes, then any address generated between 180,224 and 262,144 causes an addressing exception.

Note: *This manual is based primarily on the basic 16-bit addressing scheme. However, any exceptions in operation due to Extended Addressing are pointed out and explained by programming notes.*

Interface Addressing

The lines attached to a communication scanner are assigned interface addresses when the machine is installed. The interface address assigned to a given line is determined by the physical location of the line interface hardware and by the type of scanner (Type 1 or Type 2) installed in the controller.

Sixteen interface addresses are assigned to each Line Interface Base position, permitting a total of 352 lines on the 3705 with the maximum configuration. To uniquely address each of these lines requires nine address bits. Figures 10 and 11 show the address bits and their interpretation. The S field specifies which scanner the interface is attached to and is used when one or more Type 2 Scanners are installed. The S field is not used with a Type 1 Scanner, since only one Type 1 Scanner can be installed in a controller. The L field specifies the Line Interface Base number within the specified scanner. Only two bits of the L field (bits 3 and 4) are used with a Type 1 Scanner since four is the maximum number of LIBs that can be installed in a Type 1 Scanner. The I field is the position of the interface attachment in the specified LIB.

Whether all 16 interface addresses assigned to a given LIB position are usable depends on the type of LIB installed in that LIB position, and the type and number of line sets installed in that LIB.

With a Type 1 Scanner: When an interface address is passed to the program by an Input X'41' instruction, the interface address bits 3-8 are placed in byte 0, bits 6-7, and byte 1, bits 0-3, of the general register specified in the instruction. Byte 0, bit 4 is set to 1, and the remaining bits are set to 0.

This alignment associates a specific 2-byte storage address with each interface address. These storage addresses point to consecutive 16-byte blocks in storage. This area of storage can be directly addressed and contains a control block for control information and data handling routine addresses required for servicing an interface. Figure 12 shows the specific storage address associated with each interface position.

With a Type 2 Scanner: When an interface address is passed to the program by an Input X'40' instruction, the interface address bits 0-8 are placed in byte 0, bits 6-7, and byte 1, bits 0-6, of the specified general register. Byte 0, bit 4 is set to 1 and the remaining bits of the register are set to 0.

This alignment associates a specific 2-byte storage address with each interface address. These storage addresses are aligned on halfword boundaries and are arranged so that this area in storage can be used as a vector table to direct the control program to a routine that is designed to service the particular interface type. Figure 12 shows the storage address for each specific interface position.

For consistency, when the program provides an interface address to the Type 2 Scanner via an Output X'40' instruction, the same alignment must be maintained.

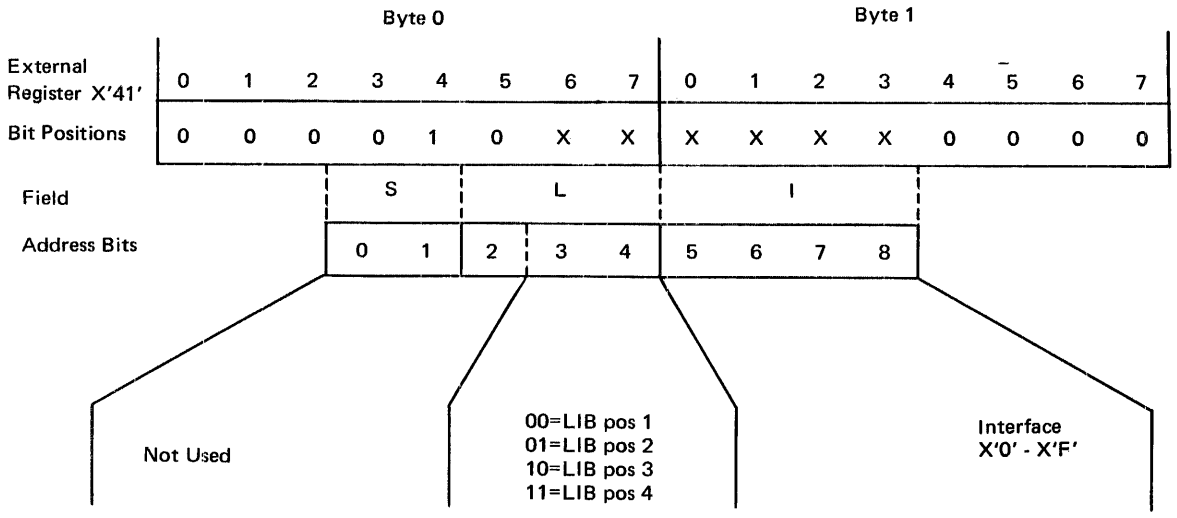


Figure 10. Type 1 Scanner Interface Address Bits

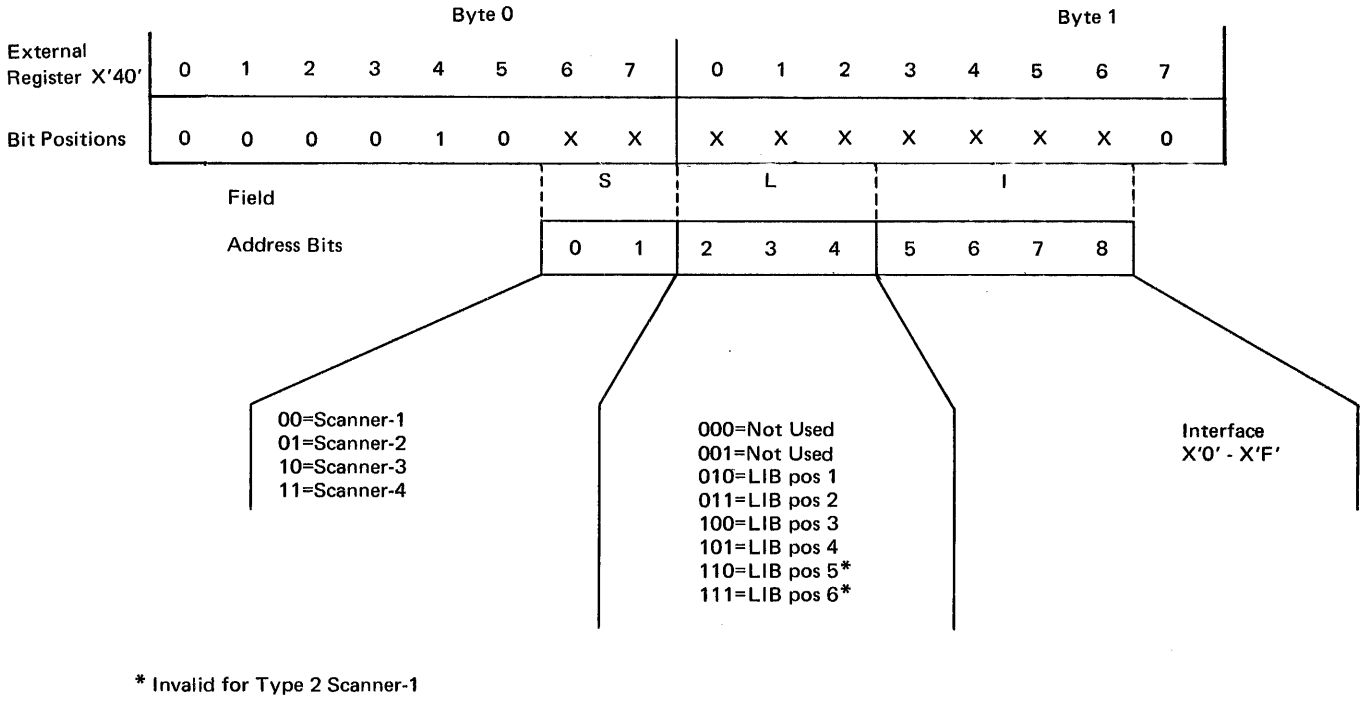


Figure 11. Type 2 Scanner Interface Address Bits

Lib and Line Addressing

Each line or autocall interface in the 3705 must be addressable for the following:

1. Scan addressing
2. Program addressing

Scan addressing and program addressing are performed differently for the Type 1 and the Type 2 Communication Scanners. These functions are described in detail in the communication scanner chapters.

		S/L (HEX)	INTERFACE ADDRESS ASSIGNMENTS (HEX) I → 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	BASIC MODULE		STORAGE ADDRESSES (HEX) *															
Type 1 Scanner	LIB position 1	00 —	800	810	820	830	840	850	860	870	880	890	8A0	8B0	8C0	8D0	8E0	8F0
	2	01 —	900	910	920	930	940	950	960	970	980	990	9A0	9B0	9C0	9D0	9E0	9F0
	3	02 —	A00	A10	A20	A30	A40	A50	A60	A70	A80	A90	AA0	AB0	AC0	AD0	AE0	AF0
	4	03 —	B00	B10	B20	B30	B40	B50	B60	B70	B80	B90	BA0	BB0	BC0	BD0	BE0	BF0
	BASIC MODULE		STORAGE ADDRESSES (HEX)															
Type 2 Scanner-1	LIB position 1	02 —	840	842	844	846	848	84A	84C	84E	850	852	854	856	858	85A	85C	85E
	2	03 —	860	862	864	866	868	86A	86C	86E	870	872	874	876	878	87A	87C	87E
	3	04 —	880	882	884	886	888	88A	88C	88E	890	892	894	896	898	89A	89C	89E
	4	05 —	8A0	8A2	8A4	8A6	8A8	8AA	8AC	8AE	8B0	8B2	8B4	8B6	8B8	8BA	8BC	8BE
	EXPANSION MODULE 1		STORAGE ADDRESSES (HEX)															
Type 2 Scanner-2	LIB position 1	0A —	940	942	944	946	948	94A	94C	94E	950	952	954	956	958	95A	95C	95E
	2	0B —	960	962	964	966	968	96A	96C	96E	970	972	974	976	978	97A	97C	97E
	3	0C —	980	982	984	986	988	98A	98C	98E	990	992	994	996	998	99A	99C	99E
	4	0D —	9A0	9A2	9A4	9A6	9A8	9AA	9AC	9AE	9B0	9B2	9B4	9B6	9B8	9BA	9BC	9BE
	5	0E —	9C0	9C2	9C4	9C6	9C8	9CA	9CC	9CE	9D0	9D2	9D4	9D6	9D8	9DA	9DC	9DE
	6	0F —	9E0	9E2	9E4	9E6	9E8	9EA	9EC	9EE	9F0	9F2	9F4	9F6	9F8	9FA	9FC	9FE
	EXPANSION MODULE 2		STORAGE ADDRESSES (HEX)															
Type 2 Scanner-3	LIB position 1	12 —	A40	A42	A44	A46	A48	A4A	A4C	A4E	A50	A52	A54	A56	A58	A5A	A5C	A5E
	2	13 —	A60	A62	A64	A66	A68	A6A	A6C	A6E	A70	A72	A74	A76	A78	A7A	A7C	A7E
	3	14 —	A80	A82	A84	A86	A88	A8A	A8C	A8E	A90	A92	A94	A96	A98	A9A	A9C	A9E
	4	15 —	AA0	AA2	AA4	AA6	AA8	AAA	AAC	AAE	AB0	AB2	AB4	AB6	AB8	ABA	ABC	ABE
	5	16 —	AC0	AC2	AC4	AC6	AC8	ACA	ACC	ACE	AD0	AD2	AD4	AD6	AD8	ADA	ADC	ADE
	6	17 —	AE0	AE2	AE4	AE6	AE8	AEA	AEC	AEE	AF0	AF2	AF4	AF6	AF8	AFA	AFC	AFE
	EXPANSION MODULE 3		STORAGE ADDRESSES (HEX)															
Type 2 Scanner-4	LIB position 1	1A —	B40	B42	B44	B46	B48	B4A	B4C	B4E	B50	B52	B54	B56	B58	B5A	B5C	B5E
	2	1B —	B60	B62	B64	B66	B68	B6A	B6C	B6E	B70	B72	B74	B76	B78	B7A	B7C	B7E
	3	1C —	B80	B82	B84	B86	B88	B8A	B8C	B8E	B90	B92	B94	B96	B98	B9A	B9C	B9E
	4	1D —	BA0	BA2	BA4	BA6	BA8	BAA	BAC	BAE	BB0	BB2	BB4	BB6	BB8	BBA	BBC	BBE
	5	1E —	BC0	BC2	BC4	BC6	BC8	BCA	BCC	BCE	BD0	BD2	BD4	BD6	BD8	BDA	BDC	BDE
	6	1F —	BE0	BE2	BE4	BE6	BE8	BEA	BEC	BEE	BF0	BF2	BF4	BF6	BF8	BFA	BFC	BFE

* Storage address X'6F0' is used for character service

Figure 12. Storage Address Assignment

Chapter 4: Instruction Set

The communications controller contains 51 executable instructions that can be used to tailor a control program to meet the specific needs of the teleprocessing system. The instruction set provides the greatest possible program flexibility within a minimum amount of storage.

This chapter gives the general instruction formats, and describes each individual instruction.

Figure 13 shows the op code bit structures, operand fields, and instruction execution times (in number of machine cycles). The asterisks in the C, Z column designate the instructions that can alter the C and Z condition latches.

Figure 14 shows the basic mnemonic designations and assembler operand field designations for the various instructions.

Any attempt at program levels 2, 3, 4, or 5 to execute an operation code other than one of the 51 specified instructions results in a level 1 interrupt with the invalid op check bit set on in the CCU interrupt request group 1 register. An attempt to execute an invalid op code in program level 1 sets the program check in L1 bit in the CCU check register along with the invalid op check bit and is handled as a CCU check. In all cases, instruction execution is suppressed. See *CCU Checks* in Chapter 5.

Instruction Format

The instruction length can be either one or two halfwords. All instructions must be located in storage on integral halfword boundaries.

The eight basic instruction formats are denoted by the format codes RR, RS, RT, RA, RSA, RE, RI and EXIT. The format codes express, in general terms, the operation to be performed. RR denotes a register-to-register operation; RS, a register-storage operation; RSA, a register-to-storage with addition operation; RT, a branch operation; RA, a register-to-immediate address operation; RE, a register-to-external register operation; RI, a register-to-immediate operand operation; and EXIT, a program level exit operation.

To help describe the execution of instructions, operands are designated as either first or second operands. For RR format instructions, the first and second operands are denoted by the number following the name of the field (for example R1, R2).

Instruction Operand Fields

Instruction operands are in four classes: (1) immediate operands in the instructions themselves, (2) operands in external registers, (3) operands in the active group of general registers, and (4) explicitly addressed operands in storage.

The following fields represent the operands in the instruction format.

Immediate Operands

I Field: The I field in RI format instructions contains an 8-bit immediate data field.

A Field: With Extended Addressing, the A field in RA format instructions contains an 18-bit immediate data field. Without Extended Addressing, the A field is treated as a 16-bit immediate data field, and bits 14 and 15 of the instruction are ignored.

External Register Operands

E Field: The E field in RE format instructions specifies the hexadecimal address of an external register operand. This field is used only for input and output instructions. Throughout this text, many references are made to input and output instructions. These references specify the value of the E field in the form X' *nn* ', where *nn* is the hexadecimal address of the external register. Appendix A shows the external register addresses and functions.

General Register Operands

R Field: The R field in RI, RS, RE, RA, RSA and some RT format instructions specifies the general register in the active group that contains the first operand. For byte operations, the register specified by this field must be an odd-numbered register. For all other operations, any one of the eight general registers in the active group may be specified.

For STH and ST instructions, if the R field is zero, a constant of all zeros is selected as the operand, rather than the contents of register 0.

R1 Field: The R1 field in RR format instructions specifies the general register containing the first operand. For byte operations, the register specified by this field must be an odd-numbered register. For all other operations, any one of the eight general registers in the active group may be specified.

R2 Field: The R2 field in the RR format instructions specifies the general register containing the second

Name	Instruction	C, Z	Cycles	FORMAT 0 1 2 3	4	5 6 7	8 9 10 11	12 13 14 15
B	Branch		1	1 0 1 0	1		T	± ¤
BCL	Branch on C Latch		1	1 0 0 1	1		T	± ¤
BZL	Branch on Z Latch		1	1 0 0 0	1		T	± ¤
BCT	Branch on Count		1	1 0 1 1	1	R N	1 T	± ¤
BB	Branch on Bit		1	1 1 M M	1	R N	M T	± ¤
LRI	Load Register Immediate	*	1	1 0 0 0	0	R N	I	
ARI	Add Register Immediate	*	1	1 0 0 1	0	R N	I	
SRI	Subtract Register Immediate	*	1	1 0 1 0	0	R N	I	
CRI	Compare Register Immediate	*	1	1 0 1 1	0	R N	I	
XRI	Exclusive Or Register Immediate	*	1	1 1 0 0	0	R N	I	
ORI	Or Register Immediate	*	1	1 1 0 1	0	R N	I	
NRI	And Register Immediate	*	1	1 1 1 0	0	R N	I	
TRM	Test Register under Mask	*	1	1 1 1 1	0	R N	I	
LCR	Load Character Register	*	1	0	0	R_1 N_1	0 0 0 0 1 0 0 0	
ACR	Add Character Register	*	1	0	0	R_1 N_1	0 0 0 1 1 0 0 0	
SCR	Subtract Character Register	*	1	0	0	R_1 N_1	0 0 1 0 1 0 0 0	
CCR	Compare Character Register	*	1	0 R_2 N_2	0	R_1 N_1	0 0 1 1 1 0 0 0	
XCR	Exclusive Or Character Register	*	1	0	0	R_1 N_1	0 1 0 0 1 0 0 0	
OCR	OR Character Register	*	1	0	0	R_1 N_1	0 1 0 1 1 0 0 0	
NCR	And Character Register	*	1	0	0	R_1 N_1	0 1 1 0 1 0 0 0	
LCOR	Load Character with Offset Register	*	1	0	0	R_1 N_1	0 1 1 1 1 0 0 0	
ICT	Insert Character and Count		2	0	0	R_1 N_1	0 0 0 1 0 0 0 0	
STCT	Store Character and Count		2	0	0	R_1 N_1	0 0 1 1 0 0 0 0	
IC	Insert Character	*	2	0	1		0 D	
STC	Store Character		2	0 B	1		1 D	
LH	Load Halfword	*	2	0	0	R	0 D	1
STH	Store Halfword		2	0	0	R	1 D	1
L	Load	*	2#	0	0	R	0 D	1 0
ST	Store		2#	0	0	R	1 D	1 0
LHR	Load Halfword Register	*	1	0	0	R_1	1 0 0 0 0 0 0 0	
AHR	Add Halfword Register	*	1	0	0	R_1	1 0 0 1 0 0 0 0	
SHR	Subtract Halfword Register	*	1	0	0	R_1	1 0 1 0 0 0 0 0	
CHR	Compare Halfword Register	*	1	0	0	R_1	1 0 1 1 0 0 0 0	
XHR	Exclusive Or Halfword Register	*	1	0	0	R_1	1 1 0 0 0 0 0 0	
OHR	OR Halfword Register	*	1	0	0	R_1	1 1 0 1 0 0 0 0	
NHR	And Halfword Register	*	1	0	0	R_1	1 1 1 0 0 0 0 0	
LHOR	Load Halfword with Offset Register	*	1	0 R_2	0	R_1	1 1 1 1 0 0 0 0	
LR	Load Register	*	1	0	0	R_1	1 0 0 0 1 0 0 0	
AR	Add Register	*	1	0	0	R_1	1 0 0 1 1 0 0 0	
SR	Subtract Register	*	1	0	0	R_1	1 0 1 0 1 0 0 0	
CR	Compare Register	*	1	0	0	R_1	1 0 1 1 1 0 0 0	
XR	Exclusive Or Register	*	1	0	0	R_1	1 1 0 0 1 0 0 0	
OR	OR Register	*	1	0	0	R_1	1 1 0 1 1 0 0 0	
NR	And Register	*	1	0	0	R_1	1 1 1 0 1 0 0 0	
LOR	Load with Offset Register	*	1	0	0	R_1	1 1 1 1 1 0 0 0	
*BALR	Branch & Link Register		2	0	0	R_1	0 1 0 0 0 0 0 0	
IN	Input		1	0 E	0	R	E	1 1 0 0
OUT	Output		1	0 E	0	R	E	0 1 0 0
BAL	Branch & Link		2	1 0 1 1	1		0 0 0 0	0 0 16–31 A
LA	Load Address		2	1 0 1 1	1		0 0 1 0	0 0 16–31 A
EXIT	Exit		1	1 0 1 1	1	0 0 0	0 1 0 0	0 0

* = Instructions that can alter condition latches.
¤ 1 = -
0 = +
\# = 3 Cycles with Extended Addressing

Figure 13. Instruction Bit Structure

Instruction	Format Code	Mnemonic	Operand Field Format
Add Character Register	RR	ACR	R1(N1),R2(N2)
Add Halfword Register	RR	AHR	R1,R2
Add Register	RR	AR	R1,R2
Add Register Immediate	RI	ARI	R(N),I
And Character Register	RR	NCR	R1(N1),R2(N2)
And Halfword Register	RR	NHR	R1,R2
And Register	RR	NR	R1,R2
And Register Immediate	RI	NRI	R(N),I
Branch	RT	B	T
Branch and Link	RA	BAL	R,A
Branch and Link Register	RR	BALR	R1,R2
Branch on Bit	RT	BB	R(N,M),T
Branch on Count	RT	BCT	R(N),T
Branch on C Latch	RT	BCL	T
Branch on Z Latch	RT	BZL	T
Compare Character Register	RR	CCR	R1(N1),R2(N2)
Compare Halfword Register	RR	CHR	R1,R2
Compare Register	RR	CR	R1,R2
Compare Register Immediate	RI	CRI	R(N),I
Exclusive Or Character Register	RR	XCR	R1(N1),R2(N2)
Exclusive Or Halfword Register	RR	XHR	R1,R2
Exclusive Or Register	RR	XR	R1,R2
Exclusive Or Register Immediate	RI	XRI	R(N),I
Exit	EXIT	EXIT	
Input	RE	IN	R,E
Insert Character	RS	IC	R(N),D(B)
Insert Character and Count	RSA	ICT	R(N),B
Load	RS	L	R,D(B)
Load Address	RA	LA	R,A
Load Character Register	RR	LCR	R1(N1),R2(N2)
Load Character with Offset Reg.	RR	LCOR	R1(N1),R2(N2)
Load Halfword	RS	LH	R,D(B)
Load Halfword Register	RR	LHR	R1,R2
Load Halfword with Offset Reg.	RR	LHOR	R1,R2
Load Register	RR	LR	R1,R2
Load Register Immediate	RI	LRI	R(N),I
Load with Offset Register	RR	LOR	R1,R2
Or Character Register	RR	OCR	R1(N1),R2(N2)
Or Halfword Register	RR	OHR	R1,R2
Or Register	RR	OR	R1,R2
Or Register Immediate	RI	ORI	R(N),I
Output	RE	OUT	R,E
Store	RS	ST	R,D(B)
Store Character	RS	STC	R(N),D(B)
Store Character and Count	RSA	STCT	R(N),B
Store Halfword	RS	STH	R,D(B)
Subtract Character Register	RR	SCR	R1(N1),R2(N2)
Subtract Halfword Register	RR	SHR	R1,R2
Subtract Register	RR	SR	R1,R2
Subtract Register Immediate	RI	SRI	R(N),I
Test Register Under Mask	RI	TRM	R(N),I

Figure 14. Instruction Format

operand. For byte operations, the register specified by this field must be an odd-numbered register. For all other operations, any one of the eight general registers in the active group may be specified.

N Field: Except for ARI, SRI and BCT instructions, the N field in RS, RI, and RT format instructions specifies whether byte 0 (N=0) or byte 1 (N=1) of the register specified in the R field is to be treated as the eight-bit first operand. For ARI, SRI and BCT instructions, the N field specifies whether byte 0 (N=0), or both byte 0 and byte 1 (N=1) are to be treated as the operand.

Programming Note
With Extended Addressing, byte X is also part of the first operand in an ARI or SRI instruction.

N1 Field: Except for ACR and SCR instructions, the N1 field in RR format instructions specifies whether byte 0 (N1=0) or byte 1 (N1=1) of the register specified in the R1 field is to be treated as the eight-bit operand. For ACR and SCR instructions, the N1 field specifies whether byte 0 (N1=0), or both byte 0 and byte 1 (N1=1) are to be treated as the operand.

Programming Note
With Extended Addressing, byte X is also part of the first operand in an ACR or SCR instruction.

N2 Field: The N2 field in RR format instructions specifies whether byte 0 (N2=0) or byte 1 (N2=1) of the register specified in the R2 field is to be treated as the eight-bit operand.

M Field: The binary value of the three-bit M field in the Branch on Bit instruction (BB) specifies the bit (bit 0-7) to be tested in the byte operand selected by the R and N fields of the instruction.

Explicitly Addressed Operands In Storage

B Field: The B field in RS and RSA format instructions specifies a general register in the active group that contains a base address. The base address is used to address second operand locations in storage. For RS format instructions, the storage address is formed by the addition of the base address (contained in the register specified by the B field) and a positive displacement specified in the D field of the instruction.

For ICT and STCT (RSA format) instructions, the base address contained in the register specified by the B field is used without modification for the storage address.

For RS format instructions, if the 3-bit B field is zero, a fixed address constant is used for the base address instead of the contents of register 0. This permits direct access to system parameters at preassigned storage locations. The address constants for the various instructions are:

Instruction	*Address Constant*
IC and STC	X'0680'
LH and STH	X'0700'
L and ST	X'0780'

D Field: The D field in RS format instructions contains the displacement that is added to the base address to form the storage address of the second operand. The D field is treated as a positive binary number that represents a byte displacement for IC and STC instructions, a halfword displacement for LH and STH instructions, and a fullword displacement for L and ST instructions. The displacement ranges that can be specified are:

Instruction	*Displacement*
IC and STC	0 to 127 bytes
LH and STH	0 to 126 bytes in multiples of two
L and ST	0 to 124 bytes in multiples of four

T Field: The T field in the RT format instructions is treated as a signed binary number. The number represents a halfword displacement from the address in register 0 of the active group of registers. Before the RT format instructions are executed, register 0 is incremented to point to the next sequential instruction (as is done for all instructions). This means that the displacement is with respect to the address of the next sequential instruction after the branch instruction. The low-order bit of the T field (instruction bit 15) is used as the sign bit. When this bit is zero, the displacement is positive. If the bit is a one, the displacement is negative. Thus, the following displacement ranges are allowed in the formation of the branch address:

Instruction	*T Field range in halfwords*	*Displacement from Branch instruction in halfwords*
BCL,BZL,B	+1023 to −1023	+1024 to −1022
BB, BCT	+63 to −63	+64 to −62

Condition Latches

A condition latch is a hardware latch that may be set or reset by instruction execution. Each of the five program levels has its own set of two condition latches. These condition latches are designated C and Z. The results of many instructions set the C and Z latches of the active group to 1 or 0. The states of these latches are described in the following instruction description sections. These condition latches can be inspected for decision making by Branch on C Latch (BCL) and Branch on Z Latch (BZL) instructions. A branch instruction never alters a condition latch. However, the program level 5 condition latches can be altered by executing an Output X'79' instruction. The state of these latches may be used as input via an Input X'79' instruction.

Programming Note

Since there is a separate set of C and Z condition latches for each of the five program levels, the state of the condition latches used by an interrupted program is not affected by other interrupting programs.

General Register Usage

Any register in the active group may be specified as containing an operand. However, in a byte operation, only the odd-numbered registers (that is, 1, 3, 5 and 7) in the active group may be specified as containing the byte operand. This is because only a two-bit field is allocated within the machine bit structure of the instruction to specify the register. Therefore, when the hardware decodes an instruction that contains a byte operation, a low-order one bit is added to access an odd-numbered register.

When a general register is specified for a halfword operation, byte 0 and byte 1 of the register are treated as a contiguous 16-bit binary operand. With Extended Addressing, byte X of the register is not used in a halfword operation and does not affect the setting of the condition latches.

When a general register is specified for an 18-bit operation (with Extended Addressing), byte X, byte 0, and byte 1 of the register are treated as a contiguous 18-bit binary operand.

When a general register is specified for a single byte operand, either byte 0 or byte 1 of the register may be specified as the operand. The other bytes in the register do not affect the setting of the condition latches. For ARI, SRI, ACR, SCR, and BCT instructions, byte 0 or both byte 0 and byte 1 can be specified as the first operand in the general register to contain the result of the arithmetic operation. For example, for the Add Character Register (ACR) instruction, the operand in R2 is added to the operand in R1, and the result is placed in R1. Since R1 is to contain the result, byte 0 or both bytes 0 and 1 of R1 could be specified as an operand, whereas only a single byte (0 or 1) of R2 can be specified as an operand.

Programming Note

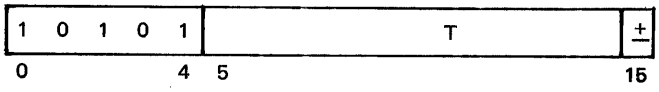
With Extended Addressing, byte X is included with the selected byte of the first operand in an ACR, ARI, SCR, or SRI instruction but does not affect the setting of the condition latches.

Instruction Descriptions

The following is a description of each of the 3705 instructions.

BRANCH

B T [RT]



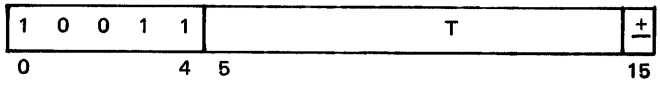
This instruction causes an unconditional branch to the branch address. The branch address is formed by adding the displacement value in the T field to the address of the next sequential instruction. Therefore, the branch address is always calculated relative to the next sequential instruction. The T field allows a displacement of +1023 to −1023 halfwords.

The low-order bit position (bit 15) of this instruction indicates whether the displacement is positive or negative. A 0 in bit 15 indicates that the displacement is in a positive direction, and a 1 indicates a negative direction.

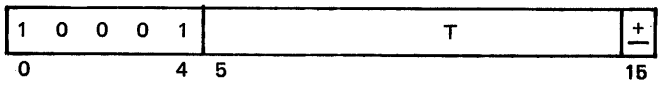
Resulting Condition Latches: Unchanged

BRANCH ON C LATCH
BRANCH ON Z LATCH

BCL T [RT]



BZL T [RT]



These instructions test the state of the C and Z condition latches associated with the active group of registers. If the tested latch is not set (0), the next sequential instruction is executed. If the tested latch is set (1), the next instruction to be executed is at the

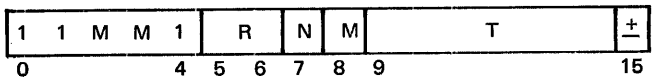
branch address. The branch address is formed by adding the displacement value in the T field to the address of next sequential instruction. Therefore, the branch address is always calculated relative to the next sequential instruction. The T field allows a displacement of +1023 to −1023 halfwords.

The low-order bit position (bit 15) of this instruction indicates whether the displacement is positive or negative. A 0 in bit 15 indicates that the displacement is in a positive direction, and a 1 indicates a negative direction.

Resulting Condition Latches: Unchanged

BRANCH ON BIT

BB R(N,M),T [RT]



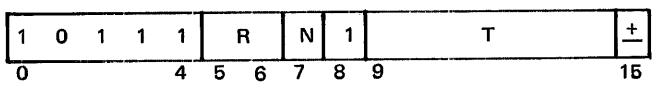
This instruction tests the state of a specified bit in a general register. If the bit tested is a zero, the next sequential instruction is executed. If the bit tested is a one, then the next instruction to be executed is at the branch address. The branch address is formed by adding the displacement value in the T field to the address of the next sequential instruction. Therefore, the branch address is always calculated relative to the next sequential instruction. The T field allows a displacement of −63 to +63 halfwords. The M field specifies which one of the eight bits of byte 0 (if N=0) or byte 1 (if N=1) of R is to be tested. The register specified by R must be an odd-numbered register.

The low-order bit position (bit 15) of this instruction indicates whether the displacement is positive or negative. A 0 in bit 15 indicates that the displacement is in a positive direction, and a 1 indicates a negative direction.

Resulting Condition Latches: Unchanged

BRANCH ON COUNT

BCT R(N),T [RT]



The count value in the register specified by R is decremented by one and then tested for zero. If the result is zero, the next sequential instruction is executed. If the result is non-zero, the next instruction to be executed is at the branch address. The count is contained in byte 0 (if N=0) or in both bytes 0 and 1 (if N=1) of the register. The branch address is formed by adding the displacement value in the T field to the address of the next sequential instruction. Therefore, the branch address is always calculated relative to the next sequential instruction. The T field allows for a displacement of −63 to +63 halfwords. The register specified by R must be an odd-numbered register.

The low-order bit position (bit 15) of this instruction indicates whether the displacement is positive or negative. A 0 in bit 15 indicates that the displacement is in a positive direction, and a 1 indicates a negative direction.

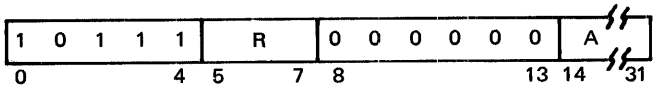
Resulting Condition Latches: Unchanged

Programming Note

If, before execution of this instruction, the count value (byte 0 or bytes 0 and 1) in the register is zero, the effective count value is 256 or 65,536, respectively.

BRANCH AND LINK

BAL R,A [RA]



This instruction is a 32-bit instruction that causes an unconditional branch. The address of the next sequential instruction is stored as link information in the register specified by R. Subsequently, the instruction address in register 0 is replaced by the branch address (address contained in the A field), and the branch is executed.

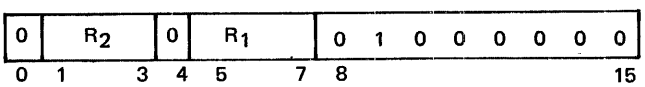
Resulting Condition Latches: Unchanged

Programming Notes

1. Since register 0 is the IAR, no linkage is provided if it is specified in the R field.
2. Bits 14 and 15 in the A field are used only with Extended Addressing.

BRANCH AND LINK REGISTER

BALR R1,R2 [RR]



The address of the next sequential instruction is stored as link information in the register specified by R1. Subsequently, the instruction address in register 0 is

replaced by the branch address (address in the register specified by R2), and the branch is executed. The branch address is obtained from R2 before the link information is stored in R1.

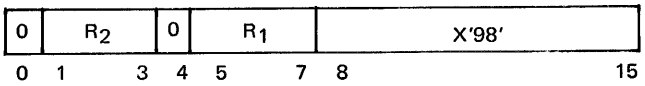
Resulting Condition Latches: Unchanged

Programming Note

Since register 0 is the IAR, no linkage is provided if it is specified in the R1 field, and no branch occurs if it is specified in the R2 field.

ADD REGISTER

AR R1,R2 [RR]



The second operand (R2) is added to the first operand (R1), and the sum is placed in the first operand location. Addition of all bits in the register operands is performed logically without regard to a sign, and the appropriate condition latches are set.

Resulting Condition Latches:

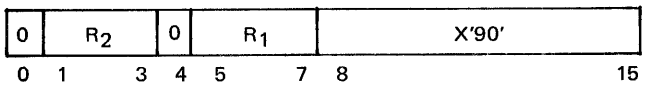
C An overflow occurred from R1
Z The result in R1 = 0

Programming Notes

1. If register 0 is specified by R1, a branch to the address formed in register 0 results, and the condition latches remain unchanged.
2. With Extended Addressing, this instruction operates on all 18 bit positions (bytes X, 0, and 1) of the registers. Without Extended Addressing, operation is the same as for the Add Halfword Register instruction.

ADD HALFWORD REGISTER

AHR R1,R2 [RR]



The second operand (R2, bytes 0 and 1) is added to the first operand (R1, bytes 0 and 1), and the sum is placed in the first operand location. Addition of the register operands is performed logically without regard to a sign, and the appropriate condition latches are set.

Resulting Condition Latches:

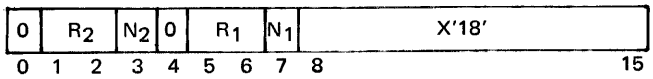
C An overflow occurred from byte 0 of R1
Z The result in bytes 0 and 1 of R1 = 0

Programming Note

If register 0 is specified as R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

ADD CHARACTER REGISTER

ACR R1(N1),R2(N2) [RR]



The second operand (R2, byte 0 if N2=0 or byte 1 if N2=1) is added to the first operandd (R1, byte 0 if N1=0 or bytes 0 and 1 if N1=1). The sum is placed in the first operand location. If N1 = 0, byte 1 of R1 remains unchanged. The registers specified by R1 and R2 must be odd-numbered registers.

Resulting Condition Latches:

N1 = 0
C An overflow occurred from byte 0 of R1
Z The result in byte 0 of R1 = 0

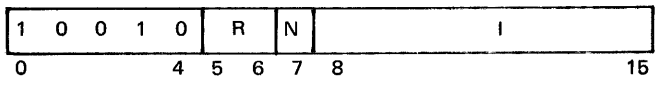
N1 = 1
C An overflow occurred from bytes 0 and 1 of R1
Z The result in bytes 0 and 1 of R1 = 0

Programming Note

With Extended Addressing, the first operand includes byte X of the register specified by R1. However, byte X does not affect the setting of the condition latches.

ADD REGISTER IMMEDIATE

ARI R(N),I [RI]



The second operand (I field) is added to the first operand (byte 0 if N=0 or bytes 0 and 1 if N=1 of the register specified by R). The sum is then placed in the first operand location. The register specified by R must be an odd-numbered register. If N=0, byte 1 of R remains unchanged.

Resulting Condition Latches:

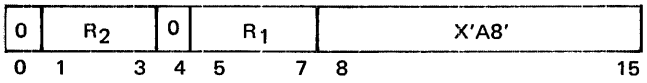
N = 0
C An overflow occurred from byte 0 of R
Z The result in byte 0 of R = 0

N = 1
C An overflow occurred from bytes 0 and 1 of R
Z The result in bytes 0 and 1 of R = 0

Programming Note
With Extended Addressing, the first operand includes byte X of the register specified by R. However, byte X does not affect the setting of the condition latches.

SUBTRACT REGISTER

SR R1,R2 [RR]



The second operand (R2) is subtracted from the first operand (R1), and the result is placed in the first operand location.

Subtraction is performed by adding the two's complement of the second operand to the first operand and setting the appropriate condition latch. If the difference is less than zero, the result is in the two's complement form.

Resulting Condition Latches:

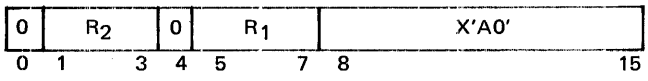
C The result in R1 < 0
Z The result in R1 = 0

Programming Notes

1. If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.
2. With Extended Addressing, this instruction operates on all 18 bit positions (bytes X, 0, and 1) of the registers. Without Extended Addressing, the operation is the same as for the Subtract Halfword Register instruction.

SUBTRACT HALFWORD REGISTER

SHR R1,R2 [RR]



The second operand (R2, bytes 0 and 1) is subtracted from the first operand (R1, bytes 0 and 1), and the result is placed in the first operand location.

Subtraction is performed by adding the two's complement of the second operand to the first operand and setting the appropriate condition latch. If the difference is less than zero, the result is in the two's complement form.

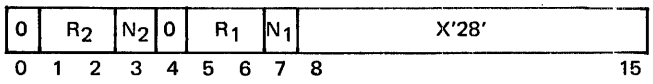
Resulting Condition Latches:

C The result in bytes 0 and 1 of R1 < 0
Z The result in bytes 0 and 1 of R1 = 0

Programming Note
If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

SUBTRACT CHARACTER REGISTER

SCR R1(N1),R2(N2) [RR]



The second operand (R2, byte 0 if N2=0 or byte 1 if N2=1) is subtracted from the first operand (R1, byte 0 if N1=0 or bytes 0 and 1 if N1=1). The result is placed into the first operand location. The registers specified by R1 and R2 must be odd-numbered registers.

Before the subtraction is performed, the second operand is expanded with high-order zeros to equal the size of the first operand. Subtraction is performed by adding the two's complement of the second operand to the first operand and setting the appropriate condition latch. If the difference is less than zero, the result is in the two's complement form.

Resulting Condition Latches:

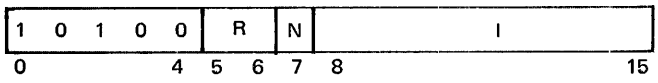
N1 = 0
C The result in byte 0 of R1 < 0
Z The result in byte 0 of R1 = 0

N1 = 1
C The result in bytes 0 and 1 of R1 < 0
Z The result in bytes 0 and 1 of R1 = 0

Programming Note
With Extended Addressing, the first operand includes byte X of the register specified by R1. However, byte X does not affect the setting of the condition latches.

SUBTRACT REGISTER IMMEDIATE

SRI R(N),I [RI]



The second operand (I field) is subtracted from the first operand (byte 0 if N=0 or bytes 0 and 1 if N=1 of the register specified by R). The result is placed in the first operand location. The register specified by R must be an odd-numbered register.

Before the subtraction is performed, the second operand is expanded with high-order zeros to equal the

size of the first operand. Subtraction is performed by adding the two's complement of the second operand to the first operand and setting the appropriate condition latch. If the difference is less than zero, the result is in the two's complement form.

Resulting Condition Latches:

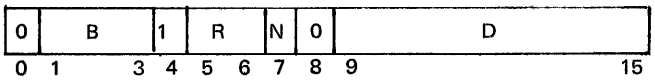
N = 0
C The result in byte 0 of R < 0
Z The result in byte 0 of R = 0

N = 1
C The result in bytes 0 and 1 of R < 0
Z The result in bytes 0 and 1 of R = 0

Programming Note
With Extended Addressing, the first operand includes byte X of the register specified by R. However, byte X does not affect the setting of the condition latches.

INSERT CHARACTER

IC R(N),D(B) [RS]



The eight-bit character at the second operand address is placed into byte 0 (if N=0) or byte 1 (if N=1) of the register specified by R. The remaining bits of the register are unchanged. The storage address of the second operand is formed by adding the displacement value in the D field to the contents of the base register specified by the B field. The D field allows for a displacement of 0 to +127 bytes. The register specified by R must be an odd-numbered register.

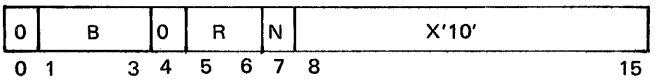
Resulting Condition Latches:

C The selected byte of R contains an even number of 1 bits
Z The selected byte of R = 0

Programming Note
If the B field is 0, address X'0680' is used as the base address instead of the contents of register 0. This permits direct addressing of the 128 bytes starting at address X'0680' without having to load a base register.

INSERT CHARACTER and COUNT

ICT R(N),B [RSA]



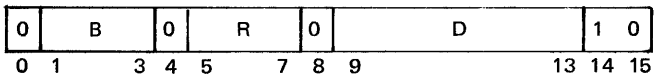
The eight-bit character at the second operand address is placed into byte 0 (if N=0) or byte 1 (if N=1) of the register specified by R. After the storage address has been obtained from the base register (B), the contents of the base register are incremented by 1. Therefore, after the execution of this instruction, the register specified by the B field normally contains an address one byte greater than before execution. (See note 2 below.) The register specified by R must be an odd-numbered register.

Resulting Condition Latches: Unchanged

Programming Notes
1. Register 0 should not normally be specified in the B field, because it contains the instruction address.
2. If the registers specified by R and B are the same, the contents of byte 1 of the register are incremented before the 8-bit character is inserted. Then if N=1, the inserted character overlays byte 1 of the same register, and the previous incrementing of this byte has no significance. If N=0, the character is inserted into byte 0 of the register, and byte 1 contains the original value plus 1.

LOAD

L R,D(B) [RS]



This instruction loads the data (second operand) from a four-byte field in storage into the first operand (register specified by R). The four-byte field containing the second operand must be on a halfword boundary. Since the general registers are not a fullword (32 bits) in length, only the low-order bits of that storage location are used. (See note 4 below.) The storage address is formed by adding the displacement value in the D field to the contents of the base register specified by B. The D field allows for a displacement of 0 to 124 bytes in multiples of 4 (31 fullwords).

Resulting Condition Latches:

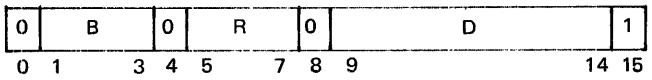
C The result in R ≠ 0
Z The result in R = 0

Programming Notes
1. The low-order bit of the storage address is ignored since storage is addressed on halfword boundaries with this instruction.
2. If register 0 (IAR) is specified in the R field, this instruction results in an unconditional branch to the address loaded in register 0, and the condition latches remain unchanged.

3. If the B field is 0, address X'0780' is used as the base address instead of the contents of register 0. This permits direct addressing of the 32 fullwords starting at address X'0780' without having to load a base register.
4. With Extended Addressing, this instruction loads the 18 low-order bits from the four-byte field addressed by the second operand into bytes X, 0, and 1 of R. The 14 high-order bits in storage are ignored. Without Extended Addressing, the low-order 16 bits of the addressed four-byte field are loaded into bytes 0 and 1 of R, and the 16 high-order bits in storage are ignored.

LOAD HALFWORD

LH R,D(B) [RS]



This instruction loads a halfword from storage into bytes 0 and 1 of the register specified by R. The storage address is formed by adding the displacement value in the D field to the contents of the base register (B). The D field allows for a displacement of 0 to 126 bytes in multiples of 2 (63 halfwords).

Resulting Condition Latches:

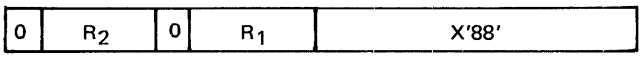
C The result in bytes 0 and 1 of R ≠ 0
Z The result in bytes 0 and 1 of R = 0

Programming Notes

1. For all Load Halfword instructions executed at program levels 2, 3, 4, or 5, or level 1 in IPL Phase 3, the halfword obtained from storage is loaded into both the specified general register and the old-CRC register. See *Cyclic Redundancy Check* in Chapter 5. For normal operations (non-CRC), the loading of data into the old-CRC register serves no function.
2. The low-order bit of the storage address is ignored since storage is addressed on halfword boundaries with this instruction.
3. If register 0 (IAR) is specified in the R field, this instruction results in an unconditional branch to the address formed in register 0, and the condition latches remain unchanged.
4. If the B field is 0, address X'0700' is used as the base address instead of the contents of register 0. This permits direct addressing of the 64 halfwords starting at address X'0700' without having to load a base register.
5. With Extended Addressing, byte X of the register specified by R is set to zero during the load operation.

LOAD REGISTER

LR R1,R2 [RR]



The second operand (R2) is loaded into the first operand (R1). All bits of the register specified by R2 are moved into the register specified by R1 and are not changed. Condition latches are set according to the result in the first operand.

Resulting Condition Latches:

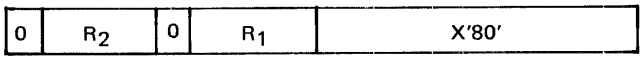
C The result in R1 ≠ 0
Z The result in R1 = 0

Programming Notes

1. If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.
2. With Extended Addressing, this instruction operates on all 18 bit positions (bytes X, 0, and 1) of the registers. Without Extended Addressing, operation is the same as for the Load Halfword Register instruction.

LOAD HALFWORD REGISTER

LHR R1,R2 [RR]



The second operand (R2, bytes 0 and 1) is loaded into the first operand (R1, bytes 0 and 1). The second operand is not changed, and the condition latches are set according to the result of the first operand.

Resulting Condition Latches:

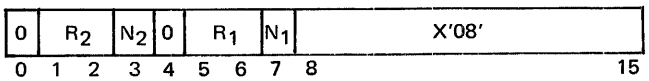
C The result in bytes 0 and 1 of R1 ≠ 0
Z The result in bytes 0 and 1 of R1 = 0

Programming Note

If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

LOAD CHARACTER REGISTER

LCR R1(N1),R2(N2) [RR]



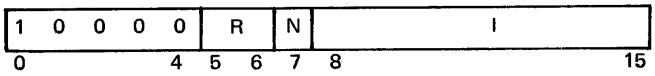
The second operand (R2, byte 0 if N2=0 or byte 1 if N2=1) is loaded into the first operand (R1, byte 0 if N1=0 or byte 1 if N1=1). The registers specified by R1 and R2 must be odd-numbered registers.

Resulting Condition Latches:

C The selected byte of R1 contains an even number of 1 bits
Z The selected byte of R1 = 0

LOAD REGISTER IMMEDIATE

LRI R(N),I [RI]



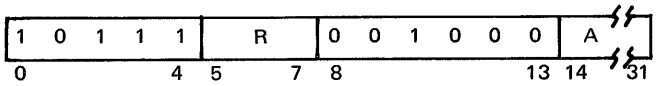
The second operand (I field) is loaded into the first operand (byte 0 if N=0 or byte 1 if N=1 of the register specified by R). The non-selected byte(s) of the register remain unchanged. The register specified by R must be an odd-numbered register.

Resulting Condition Latches:

C The result in the selected byte of R ≠ 0
Z The result in the selected byte of R = 0

LOAD ADDRESS

LA R,A [RA]



This instruction is a 32-bit instruction. The second operand (A field) is treated as an immediate operand and is loaded into the first operand (R).

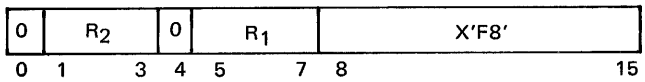
Resulting Condition Latches: Unchanged

Programming Notes

1. With Extended Addressing, instruction bits 14 and 15 in the A field are loaded into byte X of R. Without Extended Addressing, these bits are ignored.
2. If register 0 is specified by R, a branch results to the address contained in the A field.

LOAD with OFFSET REGISTER

LOR R1,R2 [RR]



The second operand (R2) is shifted right one bit position, and the result is loaded into the first operand (R1). A zero bit is inserted in the high-order bit position of R1.

Resulting Condition Latches:

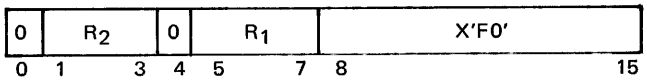
C A 1 bit shifted out of byte 1, bit 0
Z The result in R1 = 0

Programming Notes

1. If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.
2. With Extended Addressing, this instruction operates on all 18 bit positions (bytes X, 0, and 1) of the registers. Without Extended Addressing, the operation is the same as the Load Halfword with Offset Register instruction.

LOAD HALFWORD with OFFSET REGISTER

LHOR R1,R2 [RR]



The second operand (R2, bytes 0 and 1) is shifted right one bit position, and the result is loaded into the first operand (R1, bytes 0 and 1). A zero bit is inserted into the high-order bit position of R1, byte 0.

Resulting Condition Latches:

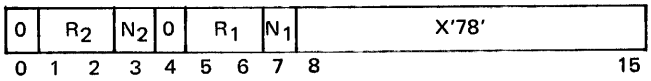
C A 1 bit shifted out of byte 1, bit 0
Z The result in bytes 0 and 1 of R1 = 0

Programming Note

If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

LOAD CHARACTER with OFFSET REGISTER

LCOR R1(N1),R2(N2) [RR]



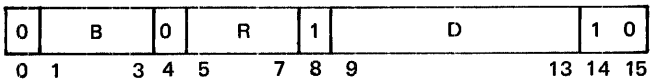
The second operand (R2, byte 0 if N2=0 or byte 1 if N2=1) is shifted right one bit position, and the result is loaded into the first operand (R1, byte 0 if N=0 or byte 1 if N=1). A zero bit is inserted into the high-order bit position of the selected byte of R1. The non-selected byte(s) of R1 remain unchanged. The registers specified by R1 and R2 must be odd-numbered registers.

Resulting Condition Latches:

C A 1 bit shifted out of the selected byte
Z The result in the selected byte of R1 = 0

STORE

ST R,D(B) [RS]



This instruction stores the contents of the first operand (the register specified by R) into the second operand in storage. The address of the second operand must be on a halfword boundary. Since the general registers are not a fullword (32 bits) in length, only the low-order bits of the four-byte field in storage will be affected. (See note 5 below.) The storage address is formed by adding the displacement contained in the D field to the contents of the base register specified by the B field. The D field allows for a displacement of 0 to 124 bytes in multiples of 4 (31 fullwords).

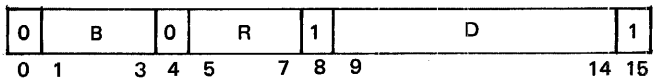
Resulting Condition Latches: Unchanged

Programming Notes

1. The low-order bit of the storage address is ignored because storage is addressed on halfword boundaries with this instruction.
2. If the B field is 0, address X'0780' is used as the base address instead of the contents of register 0. This permits direct addressing of the 32 fullwords starting at address X'0780' without having to load a base register.
3. If the R field is 0, zeros are stored, instead of the contents of register 0 (with the exception of note 4 below).
4. A Store instruction must be located at storage location X'0010' so that when a program level 1 interrupt occurs, this instruction is the first to be executed at that level. The normal function of the store instruction is modified in this special case to permit storing the contents of register 0. See the programming note under *Program Levels* in Chapter 2.
5. With Extended Addressing, the contents of the register (bytes X, 0, and 1) are stored into the low-order 18 bits of the four-byte field addressed in storage. The 14 high-order bits remain unchanged. Without Extended Addressing, bytes 0 and 1 of the register are stored into the low-order 16 bits of the four-byte field in storage, and the 16 high-order bits remain unchanged.

STORE HALFWORD

STH R,D(B) [RS]



This instruction stores bytes 0 and 1 of the register specified by R into the second operand in storage. The storage address of the second operand is formed by adding the displacement value in the D field to the contents of the base register specified by B. The D field allows for a displacement of 0 to 126 bytes in multiples of 2 (63 halfwords).

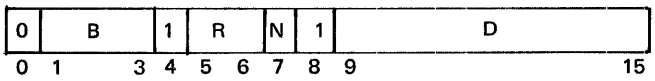
Resulting Condition Latches: Unchanged

Programming Notes

1. The low-order bit of the storage address is ignored because storage is addressed on halfword boundaries with this instruction.
2. If the R field is zero, X'0000' is stored at the storage address instead of the contents of register 0.
3. If the B field is 0, address X'0700' is used as the base address instead of the contents of register 0. This permits direct addressing of the 64 halfwords starting at address X'0700' without having to load a base register.

STORE CHARACTER

STC R(N),D(B) [RS]



The first operand (byte 0 if N=0 or byte 1 if N=1 of the register specified by R) is stored into the second operand in storage. The storage address of the second operand is formed by adding the displacement value specified by the D field to the contents of the base register specified by B. The D field allows for a displacement of 0 to 127 bytes. The register specified by R must be an odd-numbered register.

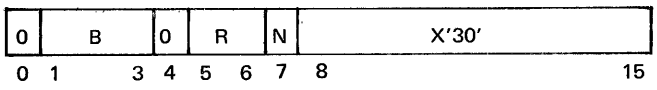
Resulting Condition Latches: Unchanged

Programming Note

If the B field is 0, address X'0680' is used as the base address instead of the contents of register 0. This permits direct addressing of the 128 bytes starting at address X'0680' without having to load a base register.

STORE CHARACTER and COUNT

STCT R(N),B [RSA]



The first operand is placed in the second operand in storage. The storage address of the second operand is contained in the register specified by the B field. The byte stored is byte 0 (if N=0) or byte 1 (if N=1) of the register specified by R. After the storage address has been obtained from the base register (B field), the contents of the register are incremented by 1. Therefore, at the completion of the execution of this instruction, the base register contains an address one byte greater than before execution. The register specified by R must be an odd-numbered register.

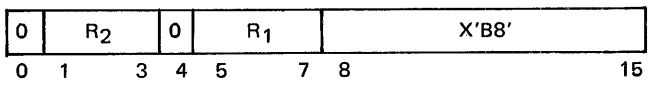
Resulting Condition Latches: Unchanged

Programming Notes

1. Register 0 should not normally be specified in the B field, because it contains the instruction address.
2. The contents of the base register are incremented before the selected byte of R is stored.

COMPARE REGISTER

CR R1,R2 [RR]



The second operand (R2) is compared to the first operand (R1), and the result sets the appropriate condition latch. This instruction performs a logical compare without regard to a sign bit. All bits of each operand participate in the comparison. The contents of the registers are not changed.

Resulting Condition Latches:

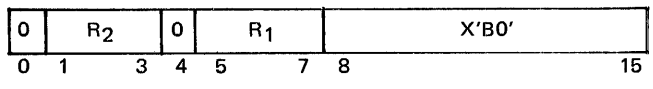
C Value in R1 < value in R2
Z Value in R1 = value in R2

Programming Note

With Extended Addressing, this instruction operates on all 18 bit positions (bytes X, 0, and 1) of the registers. Without Extended Addressing, operation is the same as the Compare Halfword Register instruction.

COMPARE HALFWORD REGISTER

CHR R1,R2 [RR]



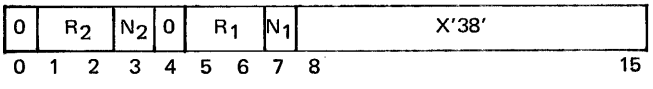
The second operand (R2, bytes 0 and 1) is compared to the first operand (R1, bytes 0 and 1), and the result sets the appropriate condition latch. This instruction performs a logical compare without regard to a sign bit. All bits of each operand participate in the comparison. The contents of the registers are not changed.

Resulting Condition Latches:

C Bytes 0 and 1 of R1 < bytes 0 and 1 of R2
Z Bytes 0 and 1 of R1 = bytes 0 and 1 of R2

COMPARE CHARACTER REGISTER

CCR R1(N1),R2(N2) [RR]



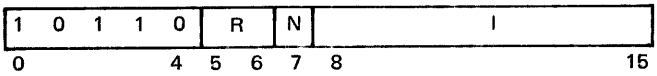
The second operand (R2, byte 0 if N2=0 or byte 1 if N2=1), is compared with the first operand (R1, byte 0 if N1=0 or byte 1 if N1=1), and the appropriate condition latch is set. This instruction performs a logical compare without regard to a sign bit. All bits of each operand participate in the comparison. The registers specified by R1 and R2 must be odd-numbered registers. The contents of the registers are not changed.

Resulting Condition Latches:

C The selected byte of R1 < the selected byte of R2
Z The selected byte of R1 = the selected byte of R2

COMPARE REGISTER IMMEDIATE

CRI R(N),I [RI]



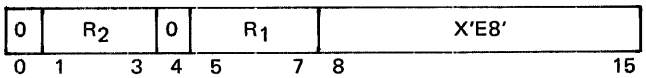
The second operand (I field) is compared with the first operand (byte 0 if N=0 or byte 1 if N=1 of the register specified by R), and the appropriate condition latch is set. This instruction performs a logical compare without regard to a sign bit. All bits of each operand participate in the comparison. The register specified by R must be an odd-numbered register, and its contents are not changed.

Resulting Condition Latches:

C The value in the selected byte of R < I
Z The value in the selected byte of R = I

AND REGISTER

NR R1,R2 [RR]



The second operand (R2) is ANDed with the first operand (R1), and the result is placed in the first operand location.

Operands are treated as unstructured logical quantities, and the connective AND is applied bit by bit. A bit position in the result is set to one if the corresponding bit positions in both operands contain a one; otherwise, the result bit is set to zero. All bits of each operandd participate in the operation. Any value in the operands or result is valid.

Resulting Condition Latches:

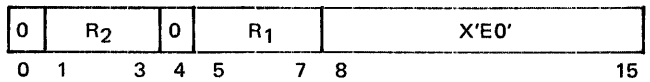
C The result in R1 ≠ 0
Z The result in R1 = 0

Programming Notes

1. If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.
2. With Extended Addressing, this instruction operates on all 18 bit positions (bytes X, 0, and 1) of the registers. Without Extended Addressing, the operation is the same as for the And Halfword Register instruction.

AND HALFWORD REGISTER

NHR R1,R2 [RR]



The second operand (R2, bytes 0 and 1) is ANDed with first operand (R1, bytes 0 and 1), and the result is placed in the first operand location.

Operation is performed in the same manner as the And Register instruction except that only the low-order 16 bits (bytes 0 and 1) of the registers are used.

Resulting Condition Latches:

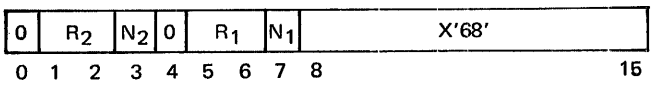
C The result in bytes 0 and 1 of R1 ≠ 0
Z The result in bytes 0 and 1 of R1 = 0

Programming Note

If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

AND CHARACTER REGISTER

NCR R1(N1),R2(N2) [RR]



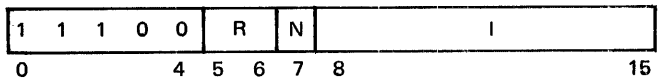
The second operand (R2, byte 0 if N2=0 or byte 1 if N2=1) is ANDed with the first operand (R1, byte 0 if N1=0 or byte 1 if N1=1). The operation is performed in the same manner as the And Register instruction, and the result is placed in the first operand location. The registers specified by R1 and R2 must be odd-numbered registers. The non-selected byte(s) of R1 remain unchanged.

Resulting Condition Latches :

C The result in the selected byte of R1 ≠ 0
Z The result in the selected byte of R1 = 0

AND REGISTER IMMEDIATE

NRI R(N),I [RI]



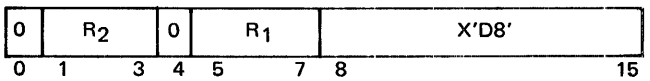
The second operand (I field) is ANDed with the first operand (byte 0 if N=0 or byte 1 if N=1 of the register specified by R). The result is placed into the first operand location with the remaining byte(s) of the register unchanged. The operation is performed in the same manner as the And Register instruction. The register specified by R must be an odd-numbered register.

Resulting Condition Latches:

C The result in the selected byte of R ≠ 0
Z The result in the selected byte of R = 0

OR REGISTER

OR R1,R2 [RR]



The second operand (R2) is ORed with the first operand (R1), and the result is placed in the first operand location. All bits of each operand participate in the operation.

Operands are treated as unstructured logical quantities, and the connective inclusive OR is applied bit by

bit. A bit position in the result is set to one if the corresponding bit position in either one or both of the operands contains a one. Otherwise the result bit is set to zero. Any value in the operands or result is valid.

Resulting Condition Latches:

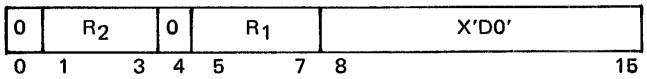
C The result in R1 ≠ 0
Z The result in R1 = 0

Programming Notes

1. If register 0 is specified by R1, a branch to the address formed in register 0 results, and the condition latches remain unchanged.
2. With Extended Addressing, this instruction operates on all 18 bit positions (bytes X, 0, and 1) of the registers. Without Extended Addressing, the operation is the same as the Or Halfword Register instruction.

OR HALFWORD REGISTER

OHR R1,R2 [RR]



The second operand (R2, bytes 0 and 1) is ORed with the first operand (R1, bytes 0 and 1), and the result is placed in the first operand location.

Operation is performed in the same manner as the Or Register instruction with the exception that only the low-order 16 bits (bytes 0 and 1) of the registers are used.

Resulting Condition Latches:

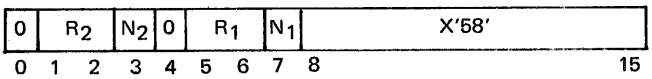
C The result in bytes 0 and 1 ≠ 0
Z The result in bytes 0 and 1 of R1 = 0

Programming Note

If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

OR CHARACTER REGISTER

OCR R1(N1),R2(N2) [RR]



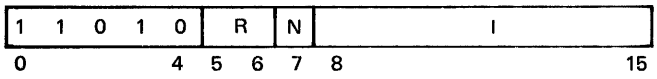
The second operand (R2, byte 0 if N2=0 or byte 1 if N2=1) is ORed with the first operand (R1, byte 0 if N1=0 or byte 1 if N1=1). The operation is performed in the same manner as the Or Register instruction, and the result is placed in the first operand location. The registers specified by R1 and R2 must be odd-numbered registers. The non-selected byte(s) of R1 remain unchanged.

Resulting Condition Latches:

C The result in the selected byte of R1 ≠ 0
Z The result in the selected byte of R1 = 0

OR REGISTER IMMEDIATE

ORI R(N),I [RI]



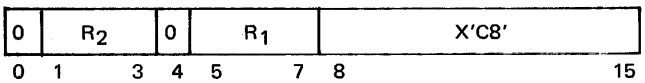
The second operand (I field) is ORed with the first operand (byte 0 if N=0 or byte 1 if N=1 of the register specified by R). The result is placed into the first operand location with the remaining byte(s) of R unchanged. The operation is performed in the same manner as the Or Register instruction. The register specified by R must be an odd-numbered register.

Resulting Condition Latches:

C The result in the selected byte of R ≠ 0
Z The result in the selected byte of R = 0

EXCLUSIVE OR REGISTER

XR R1,R2 [RR]



The second operand (R2) is exclusive ORed with the first operand (R1), and the result is placed in the first operand location. All bits of each operand participate in the operation.

Operands are treated as unstructured logical quantities, and the connective exclusive OR is applied bit by bit. A bit position in the result is set to one if the corresponding bit positions in the two operands are unlike; otherwise, the resulting bit is set to zero.

Resulting Condition Latches:

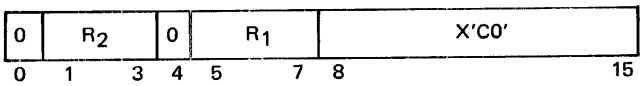
C The result in R1 ≠ 0
Z The result in R1 = 0

Programming Notes

1. If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.
2. With Extended Addressing, this instruction operates on all 18 bit positions (bytes X, 0, and 1) of the registers. Without Extended Addressing, the operation is the same as the Exclusive Or Halfword Register instruction.

EXCLUSIVE OR HALFWORD REGISTER

XHR R1,R2 [RR]



The second operand (R2, bytes 0 and 1) is exclusive ORed with the first operand (R1, bytes 0 and 1), and the result is placed in the first operand location.

The operation is performed in the same manner as the Exclusive Or Register instruction except that only the low-order 16 bits (bytes 0 and 1) of the registers are used.

Resulting Condition Latches:

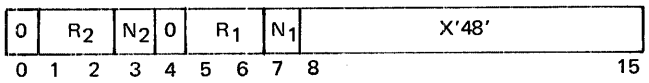
C The result in bytes 0 and 1 of R1 ≠ 0
Z The result in bytes 0 and 1 of R1 = 0

Programming Note

If register 0 is specified by R1, a branch results to the address formed in register 0, and the condition latches remain unchanged.

EXCLUSIVE OR CHARACTER REGISTER

XCR R1(N1),R2(N2)| [RR]



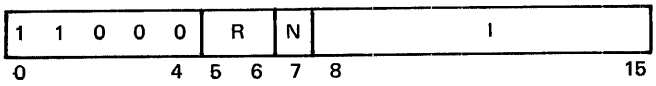
The second operand (R2, byte 0 if N2=0 or byte 1 if N2=1) is exclusive ORed with the first operand (R1, byte 0 if N1=0 or byte 1 if N1=1). The result is placed in the first operand location, and the appropriate condition latch is set. Operation is performed in the same manner as the Exclusive Or Register instruction. The registers specified by R1 and R2 must be odd-numbered registers.

Resulting Condition Latches:

C The result in the selected byte of R1 ≠ 0
Z The result in the selected byte of R1 = 0

EXCLUSIVE OR REGISTER IMMEDIATE

XRI R(N),I [RI]



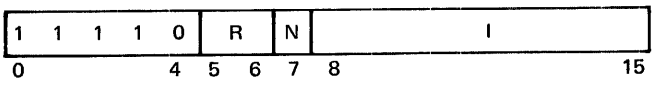
The second operand (I field) is exclusive ORed with the first operand (byte 0 if N=0 or byte 1 if N=1 of the register specified by R). The resulting byte is placed in the first operand location, and the appropriate condition latch is set. The operation is performed in the same manner as the Exclusive Or Register instruction. The register specified by R must be an odd-numbered register.

Resulting Condition Latches:

C The result in the selected byte of R ≠ 0
Z The result in the selected byte of R = 0

TEST REGISTER UNDER MASK

TRM R(N),I [RI]



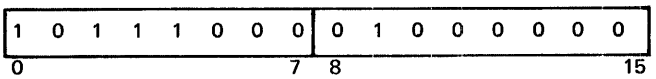
The state of the first operand bits selected by a mask is used to set the appropriate condition latch. The byte of immediate data (I field) is used as an eight-bit mask. The bits of the mask are made to correspond one for one with the bits of the first operand (byte 0 if N=0 or byte 1 if N=1 of the register specified by R). A mask bit of one indicates that the register bit is to be tested. When the mask bit is zero, the register bit is ignored. Testing is done by ANDing the selected byte of the register with the immediate operand. The contents of R are not altered. The register specified by R must be an odd-numbered register.

Resulting Condition Latches:

C The result ≠ 0
Z The result = 0

EXIT

EXIT [EXIT]



The Exit instruction is used to exit from the active program level. The interrupt-priority logic then determines which group of general registers to select as the

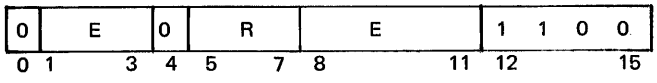
active group for the next program operation. If executed at program level 5, the level 4 supervisor call interrupt request (SVC L4) is set. Then the next instruction executed is normally the instruction at the starting address for program level 4. However, if other interrupt requests are present, the next instruction executed is the instruction at the starting address of the highest priority program level requesting an interrupt.

This instruction also resets the 'interrupt entered' latch for the program level that executes it.

Resulting Condition Latches: Unchanged

INPUT

IN R,E [RE]



This instruction loads the register specified by R with the contents of one of 128 input-addressable external registers, specified by the E field. Throughout this text the input instructions are referred to in the form: Input X *'nn'*, where nn is the hexadecimal address of the external register. Appendix A shows the hexadecimal addresses of the external registers. The 32 general registers can also be addressed as external registers. Appendixes B and C show the bit definitions for the external registers.

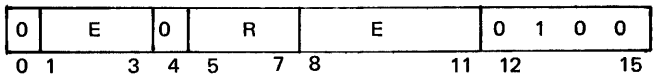
Resulting Condition Latches: Unchanged

Programming Notes

1. If register 0 is specified by R, this instruction results in a branch to the address formed in register 0.
2. This instruction is a *privileged* instruction executable only at program levels 1, 2, 3 or 4. Any attempt to execute this instruction at program level 5 causes the level 1 input/output check interrupt request (In/Out Check L1) to be set.
3. The input/output check request is set when the instruction is executed at program level 1, 2, 3, or 4 if the external register address either is not assigned or is not recognized by any adapter or the CCU.
4. The input/output check request is set if incorrect parity is detected on the CCU inbus when an input instruction is executed. This can occur for some input instructions if they are executed at an improper time. See the individual input instruction descriptions for details.
5. With Extended Addressing, byte X of the register specified by R is set to zeros with the following exceptions: (1) the E field is X'00' through X'1F' (signifying a general register), (2) the E field is X'74' (lagging address register), or (3) the E field is X'71' (panel address/data entry digits).

OUTPUT

OUT R,E [RE]



This instruction loads one of 128 output-addressable external registers specified by the E field with the contents of the register specified by R. Throughout this text the output instructions are referred to in the form: Output X *'nn'*, where *nn* is the hexadecimal address of the external register. Appendix A shows the hexadecimal addresses of the external registers. The 32 general registers can also be addressed as external registers. Appendixes B and C show the bit definitions for the external registers.

Resulting Condition Latches: Unchanged

Programming Notes

1. If register 0 of the active group of general registers is addressed as the external register, this instruction results in a branch to the address formed in register 0.
2. This instruction is a privileged instruction executable at program levels 1, 2, 3, or 4 only. Any attempt to execute this instruction at program level 5 causes the level 1 input/output check interrupt request (In/Out Check L1) to be set.
3. The input/output check request is set when this instruction is executed at program levels 1, 2, 3 or 4 and if the external register address either is not assigned or is not recognized by the CCU or any adapter.
4. If the R field is 0, and an external register from X'00' through X'1F' is specified in the E field, the parity bits of the external register are regenerated, but the data is not changed.
5. If any output instruction is executed at program levels 2, 3, or 4, or at program level 1 in IPL Phase 3, the output also causes the CRC data register in the CCU to be loaded with the contents of byte 1 of the register specified by R. See *Cyclic Redundancy Check* in Chapter 5.
6. With Extended Addressing, byte X of the register specified by R is ignored with the following exceptions: (1) the E field specifies an external register from X'00' through X'1F' (signifying a general register) or (2) the E field is X'71' or X'72' (display register 1 or 2).

Chapter 5: Central Control Unit

This chapter is intended to give the reader a basic understanding of the operation of the Central Control Unit and the requirements necessary to program its operation.

The Central Control Unit (CCU) contains the circuits and data flow paths needed (1) to execute the instruction set, (2) to address storage, (3) to perform arithmetic and logical processing of data, and (4) to control the attached adapters. Operation of the CCU is under control of the programs in storage.

The data flow in the CCU is of a general hardware nature. Data flow for a particular operation is determined by the instruction, cycle steal, or control operation being executed.

CCU Registers

The CCU contains the 32 general registers used by the control program for instruction execution and data handling. It also contains various hardware registers that are used to store and pass information essential to the controller operation. Some of these hardware registers are available to the control program as external register addresses through the use of input and output instructions. These registers are described in the following paragraphs.

BSC-CRC Register

When an Input X'7B' instruction is executed, this register is loaded with the new Cyclic Redundancy Check character and subsequently loaded into the general register specified by the input instruction. The contents of the BSC-CRC register is a current CRC character that is the combination of the old-CRC register and the CRC data register. See the *Cyclic Redundancy Check* section of this chapter for a further description of its use.

CRC Data Register

The CRC Data register is loaded with the next data character to be included in the calculation of a Cyclic Redundancy Check character. The contents of this register are then combined (by hardware) with the contents of the old-CRC register to form the new BSC-CRC character. See the *Cyclic Redundancy Check* section of this chapter for a complete description of its use.

Display Register 1 (DR1)

Display Register 1 is used as temporary storage for data to be displayed in the Display A lights on the control panel. This register can be loaded by pressing the Store or Set Address/Display push buttons on the control panel; or it can be loaded with data by executing an Output X'71' instruction. Its contents are displayed in the Display A lights whenever the Display/Function Select switch on the control panel is in the STORAGE ADDRESS, REGISTER ADDRESS, or any of the FUNCTION 1-6 positions. See *Display A and Display B Lights* in Chapter 10 for a summary of the operations that set this register.

Display Register 2 (DR2)

Display Register 2 is used as temporary storage for data that is to be displayed in the Display B lights on the control panel. This register can be loaded by pressing the Store or Set Address/Display push buttons on the control panel; or it can be loaded with data by executing an Output X'72' instruction. Its contents are displayed in the Display B lights whenever the Display/Function Select switch on the control panel is in the STORAGE ADDRESS, REGISTER ADDRESS, or any of the FUNCTION 1-6 positions. See *Display A and Display B Lights* in Chapter 10 for a summary of the operations that set this register.

Lagging Address Register (LAR)

The Lagging Address Register is a "came from" register. It normally contains the address of the last instruction executed prior to the instruction that is currently executing. The LAR is loaded from the storage address register each time an instruction is executed in program levels 2, 3, 4, and 5. A program level 1 interrupt prevents setting the LAR until the Exit instruction is executed and no other requests for level 1 are outstanding. This action preserves the address of the last instruction executed before a level 1 request. Since level 1 interrupts are caused primarily by the detection of an error condition, this register becomes extremely important for error recovery procedures. Figure 15 shows the contents of LAR after a check condition or a control panel operation other than normal instruction execution.

The control program can load the contents of LAR into a general register by executing an Input X'74' instruction. The control program can then either examine the contents of the general register or display the address on the control panel by using the general register as input to the display registers.

Old-CRC Register

The Old-CRC accumulation register is used as a temporary storage register in the calculation of a Cyclic

CONDITION	*LAR CONTENTS*
Invalid Op Code Check	Address of last instruction executed before the one that caused the check (see note)
Protection Check or Address Exception Check	Address of last instruction executed before the one that caused the check (see note)
	-or-
	Address of the instruction that caused the check
In/Out Check at Level 2, 3, or 4	Address of the input or output instruction that caused the check
In/Out Check at Level 5	Address of last instruction executed before the one that caused the check. (see note)
IPL (including CCU check)	Address of last instruction executed before IPL phase 1
Adapter Check	Unpredictable
-- Control Panel Operations --	
LOAD, ADDRESS COMPARE PROGRAM STOP or INTERRUPT	Address of last instruction executed before the one whose address is set in switches A to E (see note)
LOAD or STORE, ADDRESS COMPARE PROGRAM STOP or INTERRUPT	Address of instruction that loaded from or stored into the location set in switches A to E
INSTRUCTION STEP	Address of last instruction executed
STOP Push Button	Address of last instruction executed

Note: *The last instruction may have been an Exit instruction executing at a higher-priority program level than the level executing at the time the condition occurred. Therefore, LAR contains the address of that Exit instruction.*

Figure 15. Operations That Set The Lagging Address Register.

Redundancy Check (CRC) character. When an Input X'7B' instruction is executed, the contents of this register are combined with the character in the CRC data register to form a new BSC-CRC character. This register is loaded by a Load Halfword instruction. See the *Cyclic Redundancy Check* section of this chapter for a complete description of its use.

Operation Register (OP Reg)
The Operation Register is used to hold the first 16 bits of the instruction being executed. This register can be displayed in the Display B lights on the control panel when the Display/Function Select switch is in the TAR/OP REGISTER position.

Storage Address Register (SAR)
The Storage Address Register contains the storage address currently used by the program to read or write data to and from storage. This register is loaded from the temporary address register for instruction execution or from the cycle-steal address bus for cycle-steal operations.

Temporary Address Register (TAR)
The Temporary Address Register holds the storage addresses pertaining to instruction execution until the storage address register can be changed. This register normally contains the address of the next instruction to be executed. The contents of TAR are displayed in the DISPLAY A lights on the control panel when the Display/Function Select switch is in the TAR/OP REGISTER position.

Setting/Resetting Interrupt Requests

A particular interrupt request latch can be set as a result of a hardware-detected condition or, in some cases, by the program through the execution of an output instruction. The latch can be reset by either an input or an output instruction, depending upon the specific interrupt request. The procedures for setting/resetting individual adapter interrupt requests are described in the adapter sections.

For special service requests, program levels 1, 2, 3, and 4 may issue a program-controlled interrupt (PCI) request to program levels 3 and 4. Output instructions X'7C' (set PCI level 3) and X'7D' (set PCI level 4) set the PCI interrupt requests. (See Appendix B, Output X'7C' and X'7D'.) Certain bits in Output X'77' (miscellaneous control) reset the PCI requests and other Central Control Unit interrupt requests such as the interval timer level 3 request and the SVC level 4 request.

Determining the Cause of an Interrupt

Priorities between simultaneous interrupt requests assigned to the same interrupt program level are resolved by the order in which the program tests the

set/reset condition of the CCU and adapter interrupt request latches.

Interrupt requests from the CCU and the adapters are grouped together according to their source for ease of identification. The set/reset condition of a specific interrupt request latch can be determined by checking the interrupt request group to which it is assigned. Inputs X'76' and X'77' indicate the interrupt requests that are set by the adapters. A request for level 1 sets a bit in Input X'76', and a request for levels 2 or 3 sets a bit in Input X'77'. The CCU interrupt requests for level 1 or levels 2, 3, and 4 are set in Inputs X'7E' and X'7F' respectively. These inputs load the contents of the appropriate interrupt request group into an active general register. (See Appendix B for input instruction bit definitions of interrupt request groups.) The program may then test the general register to identify the request.

Masking Program Level Priorities

Programs at program levels 1, 2, 3, or 4 can selectively mask program levels 2, 3, 4, or 5, and level 1 adapter checks. Level 1 adapter checks, however, can only be masked while the CCU is in the *test mode*. (See *CCU Diagnostic Facilities* in this chapter.) The normal operational priority structure can be changed by Output instructions X'7E' and X'7F' (set/reset mask register). Refer to Appendix B for the mask register bit assignments.

When a program level is masked, use of machine cycles for instruction execution at that program level is suppressed until it is unmasked. Masking is normally used to prevent a higher-priority program level from interrupting a lower-priority level during execution of a time or data dependent routine. For example, if program level 3 contains a routine that should not be interrupted, the level 3 program can mask level 2 interrupts before entering this routine to ensure contiguous instruction execution. When the routine is completed, level 2 should be unmasked to allow interrupts.

If program level 2, 3, or 4 has already been entered, instruction execution at that level is allowed to complete before the masking of that level is effective. For example, if program level 2 interrupts program level 3 and subsequently masks level 3 interrupts, control can return to the level 3 program at the end of level 2 processing. However, once the level 3 program executes an Exit instruction, interrupts to level 3 cannot occur until it is unmasked. If program level 5 is interrupted by another level that sets the level 5 mask, control cannot be returned to level 5 until it is unmasked.

To selectively mask one or more program levels, one of the active general registers is loaded with the bits corresponding to those program levels to be masked. Output instruction X'7E' (set mask register) is then executed using the general register as input to the mask register. To selectively unmask one or more program levels, the same procedure is followed except that the Output X'7F' (reset mask register) instruction is executed.

Masking and subsequent unmasking should be handled with extreme caution to avoid disrupting the normal priority structure. If not used carefully, masking could cause overrun conditions or delay of hardware error indications.

Storage Protect

Storage Protect is a means of notifying the control program whenever the contents of storage are accessed for unauthorized modification. This facility monitors all attempts to modify storage and execute instructions. However, due to hardware restrictions, storage protection is effective only in program level 5 and for Type 2 CA cycle-steal operations.

Protection is achieved by a hardware comparison of separate keys assigned to the program and to storage. A user's ability to modify storage is identified by a 3-bit *protect key*. Each program level and cycle-steal mechanism is considered a user, and each has its own protect key. Storage is divided into blocks of 2048 bytes, and each block is assigned a 3-bit *storage key*.

When a protected area of storage is addressed, the storage key for that location is compared with the protect key associated with the user. Access to the location, for operands and instructions, is granted only when the two keys match. For attempts to execute an instruction, the two 3-bit keys *must* be equal. If the keys do not match, a protection exception L1 interrupt request is set. For attempts to modify a storage location, the keys match when (1) the keys are equal, (2) the protect key is X'0', or (3) the storage key is X'7' (unprotected storage).

The protect keys for program levels 1, 2, 3, and 4 are fixed at 0 and cannot be changed. The protect keys for the remaining users are set by the control program with an Output X'73' (set key) instruction. Byte 1, bit 3 of an Output X'73' instruction must be 0 to indicate protect key selection. The protect key address of the desired user is placed in byte 0, bits 4-6, and the protect key is placed in byte 1, bits 5-7. Figure 16 shows the protect key addresses and the users they apply to.

Note: *The protect key address is not an actual storage address. This key address is only a reference pointer for assigning a key to a user.*

*Protect Key Address (PKA)**	*Applies to*
Bits 0 1 2	
0 0 0	Protect Key - Program Level 5
0 0 1	Protect Key - Type 2 CA-1 Cycle Steals
0 1 0	Protect Key - Type 2 CA-2 Cycle Steals
0 1 1	Reserved
1 0 0	Reserved
1 0 1	Reserved
1 1 0	Reserved
1 1 1	Reserved
* Output X'73', byte 0, bits 4-6	

Figure 16. Protect Key Address Bits

To set any key, byte 1, bit 4 (set key) must also be 1. The set-key bit allows execution of the Output X'73' instruction to access a storage or protect key without changing the key.

The Output X'73' insturction also sets the storage keys. The storage key of an area is determined by dividing storage into 2K blocks. Each block is then given a storage block address, from 0 to 128, relative to its position in storage. Figure 17 shows the storage block address that is assigned to the storage blocks. To set a storage key, the storage block address must be placed in byte 0, bits 0-6 of a general register, and the storage key set in byte 1, bits 5-7. Byte 1, bit 3 (key address select) and bit 4 (set key) must also be 1 to select the storage key and allow it to be set when an Output X'73' is executed. See Appendix B for a definition of the bits used in the Output X'73' instruction.

The Input X'73' instruction (insert key) can be used to determine the key (storage or protect) that must be used for storage access. When executed, this input instruction places the key that was addressed by the last Output X'73' instruction into byte 1, bits 5-7 of the register specified by the R operand. Therefore, the program must know what type of key and what address was used in the preceding Output X'73'. By executing an Output X'73' with byte 1, bit 4 (set key bit) off, no keys are changed. However, this allows the address portion of the register to be used by an Input X'73'. Byte 1, bit 3 (key select bit) of the output instruction must also be known in order to determine the type of key (storage or protect) being accessed. See Appendix B for a definition of the bits used in the Input X'73' instruction.

Resetting the Central Control Unit disables storage protection. Therefore, any instruction fetch is valid, and any attempt to modify storage is permitted. The first Output X'73' instruction to be executed after a reset enables storage protect. This output instruction *must* set a storage key of 0 at the key address that corresponds to the storage block where the instruction execution is taking place. Otherwise a *protection check* occurs unless the storage key is already 0.

When the controller is powered on, the bits in all the protect keys except program levels 1-4 and all the storage keys assume a random bit pattern. Therefore, each key must be initialized by an Output X'73' containing its key address and key type (storage or protect). Until each key is fully initialized, caution must be exercised in the control of program levels and I/O activity that may depend on storage protection.

Interval Timer

The interval timer provides a program level 3 interrupt request (Interval Timer L3) approximately once every 100 milliseconds. The time interval is derived from the 50/60 cycle power line frequency, and its accuracy depends on the stability of the power source.

The interval timer may be used to maintain a real-time clock in storage, perform long and short I/O timeouts, and perform supervisory functions on a periodic basis. The interval timer interrupt request can be reset by executing an Output X'77' (miscellaneous control) instruction with byte 1, bit 1 set to 1.

Cyclic Redundancy Check (CRC)

Although any cyclic redundancy check (CRC) can be performed by a program algorithm, the communications controller has special hardware to speed the CRC accumulation function required for binary-synchronous (BSC) EBCDIC devices.

For all Load Halfword instructions executed at program levels 2, 3, 4, or level 1 in IPL phase 3, the halfword obtained from storage is loaded into both the specified general register and a register called the old-CRC register. For normal operations (non-CRC), the loading of data into the old-CRC register serves no function. However, when a CRC update is to be performed, an additional instruction is not required, because the old CRC accumulation is automatically loaded into the old-CRC register.

SKA Bits 0 to 2	SKA Bits 3 to 6															
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
000	0000	0800	1000	1800	2000	2800	3000	3800	4000	4800	5000	5800	6000	6800	7000	7800
001	8000	8800	9000	9800	A000	A800	B000	B800	C000	C800	D000	D800	E000	E800	F000	F800
010	10000	10800	11000	11800	12000	12800	13000	13800	14000	14800	15000	15800	16000	16800	17000	17800
011	18000	18800	19000	19800	1A000	1A800	1B000	1B800	1C000	1C800	1D000	1D800	1E000	1E800	1F000	1F800
100	20000	20800	21000	21800	22000	28000	23000	23800	24000	24800	25000	25800	26000	26800	27000	27800
101	28000	28800	29000	29800	2A000	2A800	2B000	2B800	2C000	2C800	2D000	2D800	2E000	2E800	2F000	2F800
110	30000	30800	31000	31800	32000	32800	33000	33800	34000	34800	35000	35800	36000	36800	37000	37800
111	38000	38800	39000	39800	3A000	3A800	3B000	3B800								

Notes:

1. Storage addresses are shown in hexadecimal.
2. Storage addresses shown are beginning addresses of block.
3. Storage Key Address (SKA) bits correspond to Output X'73' instruction byte 0, bits 0 to 6.

Figure 17. Storage Key Addresses

To generate a new CRC accumulation, a Load Halfword instruction should be used to load the old-CRC register. The next character to be included in the CRC accumulation should then be placed in the low-order byte of one of the active general registers. Any output instruction can then be executed using the general register as the source of input. This places the character from the general register into a register called the CRC data register.

Note: *When updating the CRC accumulation, the output instruction executed also performs its normal functions. Therefore, caution must be taken not to execute an output that can alter the status or state of either an adapter or an interface.*

After a character to be added is in the CRC data register and the old-CRC accumulation is in the old-CRC register, Input X'7B' (New BSC CRC) should then be used to load the new BSC CRC accumulation to a general register. During execution of the input instruction, the character to be included in the CRC accumulation and the old-CRC are combined by hardware circuits, and the new-CRC is stored in bytes 0 and 1 of the general register specified in the input instruction. With Extended Addressing, bits 6 and 7 of byte X are reset to 0. The new CRC accumulation may then be placed in storage by the program.

Programming Note

The Load Halfword instruction that loads the old-CRC accumulation, the output instruction that loads the character into the CRC data register, and the Input X'7B' (New BSC CRC) that loads the new BSC CRC into a general register do not have to be consecutive instructions. However, there must not be another Load Halfword instruction (executed at program levels 2, 3, 4, or level 1 in IPL phase 3) between loading the old CRC in the old-CRC register and loading the new BSC CRC to a general register. Also, another output instruction must not be executed between loading the CRC data register and storing the new BSC CRC.

Initial Program Load (IPL)

The initial program load (IPL) mechanism controls the loading of an initial program into the controller via a channel adapter (1) when the system is first powered up, (2) when further processing is not possible due to an error condition, or (3) when the channel adapter decodes a Write IPL command. Three phases of the IPL program (IPL Phase 1-3) control the loading operation. IPL is accomplished by successful completion of all three phases.

Phase 1 of the IPL operation is a general reset to the controller. During phase 2, a small bootstrap program is automatically loaded into storage from the controllers read-only-storage (ROS) array, and control passes to this program. Execution of the bootstrap program (phase 3) then controls channel operations until the first *load module* from the host successfully transfers into storage under a Write IPL channel command (X'05').

After successful transfer of the first program segment into storage, the initial program loading operation performed by the bootstrap program is complete. The controller is initialized, and the bootstrap program passes control to the loaded program segment. This program segment then controls the loading of whatever additional load modules are required to complete the program load operation.

Two of the lights in Display A on the control panel indicate the three phases of IPL. These lights are a binary representation of the phase number. That is, 01 = phase 1, 10 = phase 2, and 11 = phase 3.

The Load light on the control panel turns on when IPL is initiated. It does not turn off when IPL Phase 3

is completed. It must be turned off by executing an Output X'79' instruction when the program determines that the controller has been completely loaded.

IPL starts when either of the following occurs:

- The Diagnostic Control Switch is in the PROCESS, CCU CHECK HARD STOP or BYPASS CCU CHECK STOP position and either (1) the Load push button is pressed; (2) a power-on occurs; (3) a channel adapter decodes a Write IPL command, and the controller is either not already in IPL phase 1, 2, or 3 or not in a hard stop condition; or (4) the control program executes an Output X'79' instruction with byte 0, bit 2 (set IPL) set to 1 in the register specified by the R operand.
- The Diagnostic Control switch is in the PROCESS position, a Central Control Unit check occurs, and the controller is not in IPL phase 1, 2, or 3.

IPL Phase 1 - Reset: During this phase the Load light is turned on, the CCU is placed in the test mode and a general reset occurs in the controller. Unless the IPL is initiated by a power-on sequence, the phase 1 reset does not affect the state of the channel adapter(s).

The reset in the CCU:

- Sets the mask bits for program levels 2-5 and level 1 adapter requests.
- Resets all 'interrupt entered' latches.
- Resets all CCU interrupt requests.
- Resets program stop and hard stop.
- Prohibits storage references and instruction execution.
- Disables the storage protect facility until an Output X'73' instruction is executed.
- Turns *on* the Test light on the control panel.

IPL Phase 2 - Bootstrap Load: In this phase the bootstrap program is automatically loaded into storage from read-only-storage (ROS). Different ROS arrays are installed for the Type 1 CA and the Type 2 CA because the two channel adapters require different bootstrap programs. The bootstrap program for either channel adapter begins loading at address X'0000' and is 512 bytes long. When two channel adapters are installed, the bootstrap program is 1024 bytes to handle IPL from either channel adapter.

When the bootstrap load operation is complete, the IPL L1 interrupt request is set, and the controller enters IPL phase 3 (bootstrap execution).

IPL Phase 3 - Bootstrap Execution: The bootstrap program begins execution at address X'0010' as a result of the IPL level 1 interrupt request set by phase 2 and operates entirely at program level 1.

The bootstrap program is divided into two sections. The first section:

1. Saves the general registers of group 0 starting at storage location X'0780'. Each register location starts on a fullword boundary.
2. Verifies the operation of the controller instructions to be used in the second section of the bootstrap program.
3. If the program is allowed to continue, it saves external registers X'76', X'7D', and X'7E' at storage locations X'0702', X'0704', and X'0706' respectively.
4. Resets the test mode and turns the Test light on the control panel *off*.

The second section of the bootstrap program controls the channel adapter operations until the first program load module successfully transfers from the host processor.

If no outstanding command or final status is pending in the channel adapter, it generates an asynchronous status of Device End and Unit Check. If any command other than Write IPL is pending, the channel adapter generates a final status of Channel End, Device End, and Unit Check. In either case, sense bit 6 (Not Initialized) is set for the subsequent Sense command. A Write IPL command normally follows the Sense command. The Not Initialized bit is reset after the first program segment successfully transfers into storage from the host processor.

The Write IPL command causes the transfer of the first load module from the host processor into controller storage. Under the Write IPL command, the load module is stored in sequential locations starting at storage location X'0400'. The maximum size of this load module cannot exceed 1022 bytes. After successful completion of this transfer, the bootstrap program executes an Output X'77' instruction with byte 0, bit 0 (reset IPL L1) *on* in the register specified by the R operand. This resets the IPL L1 interrupt request and turns *off* the IPL Phase lights on the control panel. The bootstrap program then turns control over to the load module at program level 1 by branching to storage location X'0404'.

The IPL operation is complete when the IPL L1 interrupt request is reset and the IPL Phase lights are turned *off*. However, the Load light on the control panel remains on until reset by byte 1, bit 1 of Output X'79'. The program just loaded from the host should execute the Output X'79' when it is determined that the controller is completely loaded.

Programming Note
The first two halfwords of the load module must contain an IPL source identification (host processor ID) and the number of bytes in the load module (including the source ID and count bytes).

Check Conditions During IPL

Central Control Unit (CCU) checks are prohibited during IPL phase 1 since that phase performs a reset. If a CCU check occurs during IPL phase 2 or IPL phase 3, a *hard stop* occurs. If the bootstrap program determines that any of the bits are on in the central control unit check register (Input X'7D') when IPL phase 3 is entered, the bootstrap program causes the Equipment Check bit in the sense byte to be set along with other appropriate sense information.

If the bootstrap program does not reach completion for any of the following reasons, the controller either comes to a hard stop or enters a loop and attempts to display the cause of the check condition in the control panel display lights.

- CCU check hard stop condition.
- Improper operation of an instruction during instruction verification in the first section of phase 3.
- Program continuity check (Type 2 CA only). This check ensures that all instructions in the bootstrap program have executed in the correct sequence.
- Channel adapter disabled.
- Channel adapter hardware malfunction.
- Byte count in the second halfword of the load module does not compare with the number of bytes transferred from the host.

Input/Output Instructions

The control program uses input and output instructions to control and monitor the status of the CCU and the installed adapters. Appendix B defines the bits for each of the input and output instructions.

Input Instructions

The Central Control Unit has 16 input instructions. With these instructions, the control program monitors the status of the CCU, the communication scanners, the channel adapters, and the control panel, and is informed of any error conditions. Four of the input instructions (X'75', X'78', X'7A', and X'7C') are not used, and if executed, the bits in the general register are set to zero. The other twelve instructions set the bits of the general register according to the particular external register value. (Appendix B defines the bits within each input instruction.)

Input X'70' (Storage Size)
This instruction loads a general register with a combination of bits that indicates the amount of storage installed.

Input X'71' (Panel Address/Data Entry)
This instruction loads a general register with the bit combination that appears in the ADDRESS/DATA switches on the control panel. Through the use of this instruction, the program can accept information from the operator.

Input X'72' (Panel Display/Function Select Control)
This instruction loads a general register with a combination of bits to indicate the position of the control panel DISPLAY/FUNCTION SELECT switch. Through the use of this instruction, the program can accept information from the operator.

Note: *The STATUS and the TAR & OP REGISTER positions of the Display/Function Select switch are not included in the register.*

Input X'73' (Insert Key)
This instruction loads a general register with the storage key or protect key addressed by the last Output X'73' instruction executed. The program must know the bit setting of the the key address bits set in the last Output X'73' instruction before this input is meaningful.

Input X'74' (Lagging Address Register)
This instruction loads a general register with the contents of the lagging address register. When this input is executed at program levels 2, 3, or 4, the address transferred into the general register is that of the last instruction executed before the input instruction.

When executed in program level 1, the address transferred into the general register is that of the last instruction executed at the program level that was active before the level 1 interrupt.

Input X'76' (Adapter Interrupt Requests - Group 1)
This instruction loads a general register with information that can be used to determine which channel adapter or communication scanner caused a level 1 interrupt.

Input X'77' (Adapter Interrupt Requests - Group 2)
This instruction loads a general register with information that can be used to determine which channel adapter or communication scanner caused a level 2 or level 3 interrupt.

Input X'79' (Utility)
This instruction loads a general register with utility information. When executed in program level 1, byte 1, bits 0-3 designate the program level that was operating before the level 1 interrupt. When executed in program levels 2, 3, or 4 (or level 1 if level 1 is reentered immediately after a level 1 Exit), byte 1, bits 0-3 have no significance and are set to zero. When executed at any level, byte 0, bits 6 and 7 indicate the state of the program level 5 C and Z condition latches.

Input X'7B' (BSC CRC Register)
This instruction loads a general register with the new binary synchronous CRC character from the BSC CRC register. See the *Cyclic Redundancy Check* section in this chapter.

Input X'7D' (CCU Check Register)
This instruction loads a general register with the contents of the CCU check register. The program can test this register to determine the cause of a program level 1 interrupt due to a CCU check condition.

Input X'7E' (CCU Interrupt Requests - Group 1)
This instruction loads a general register with a configuration of bits to indicate the cause of a program check or a program level 1 interrupt due to an address compare or IPL. When a program check occurs, Input X'7D', byte 0, bit 3 should also be tested to determine if the check occurred while in program level 1 or in one of the other program levels.

Input X'7F' (CCU Interrupt Requests - Group 2)
This instruction loads a general register with a combination of bits to indicate the cause of various interrupts. Certain interrupt requests from program levels 2-4 set a bit associated with the type of request presented.

Output Instructions

The Central Control Unit has 16 output instructions to control its operation and data flow. However, the Output X'74', X'75', X'76', X'7A', and X'7B' instructions are not used, and if executed, the bit settings of the register specified by th R operand are ignored. The instructions that have an effect on an external register are as follows. (Appendix B defines the bits within each output instruction.)

Output X'70' (Hardstop)
This instruction causes the 'hardstop' latch in the CCU to turn *on*. This stops all machine activity and requires a reset and IPL to continue operation. Since this instruction performs a function and not an operation, the bit settings of the general register are ignored.

Output X'71' (Display Register 1)
This instruction loads the contents of the general register into display register 1. Whenever the Display/Function Select switch on the control panel is in the STORAGE ADDRESS, REGISTER ADDRESS, or any of the FUNCTION 1-6 positions, the bits of the register are displayed in the Display A lights on the control panel.

The Program Display light on the control panel also comes *on* to inform the operator that information has been placed in the display register.

Output X'72' (Display Register 2)
This instruction causes the contents of the general register to be loaded into display register 2. Whenever the Display/Function Select switch on the control panel is in the STORAGE ADDRESS, REGISTER ADDRESS, or any of the FUNCTION 1-6 positions, the bits of the register are displayed in the Display B lights on the control panel.

The Program Display light on the control panel also turns *on* to inform the operator that information has been placed in the display register.

Output X'73' (Set Key)
This instruction causes the contents of the general register to be used to address and/or set the storage and protect keys.

Output X'77' (Miscellaneous Control)
This instruction causes the contents of the specified general register to be used to set or reset various interrupt requests.

Output X'78' (Force CCU Checks)
This instruction provides the means for testing the CCU check circuits under diagnostic control. It causes the contents of the general register to be used to force certain error conditions in the CCU data flow. This instruction is valid only while the CCU is in the test mode. Refer to the *CCU Diagnostic Facilities* section of this chapter for details on forcing check conditions.

Output X'79' (Utility)
This instruction causes the contents of the specified general register to set and/or reset various hardware latches and lights.

Output X'7C' (Set PCI L3)
This instruction causes a Program Controlled Interrupt request to be set for program level 3 (PCI L3). This allows a program level to transfer a processing requirement to a different priority program level. Since this instruction performs a function, the bit settings of the general register are ignored.

Output X'7D' (Set PCI L4)
This instruction causes a Program Controlled Interrupt request to be set for program level 4 (PCI L4). This allows a program level to tranfer a processing requirement to a lower priority level. Since this instruction performs a function, the bit settings of the general register are ignored.

Output X'7E' (Set Mask Bits)
This instruction causes the mask bits of the program levels to be set according to the contents of the general register. When a mask bit is set on, interrupts for the program level that corresponds to that bit are not permitted.

Output X'7F' (Reset Mask Bits)
This instruction causes the mask bits of the program levels to be reset according to the contents of the general register.

CCU Checks

The Central Control Unit performs various hardware checks on the program operation and hardware circuits in the controller to ensure proper operation. When a check condition is detected, bits set in the CCU check register indicate the type of check. The control program can then execute an Input X'7D' instruction to load a general register with the information available in the check register.

The first occurrence of a check condition sets the CCU check register. Subsequent checks are not allowed to enter the check register until it is reset. The check register is reset by executing an Output X'77' instruction with byte 0, bit 1 set to 1, or by pressing the CCU Check Reset or Reset push button on the control panel.

When a CCU check condition is detected and the Diagnostic Control switch on the control panel is in the PROCESS position, an automatic IPL is initiated. If the check condition persists after the IPL sequence is initiated, the 'hard stop' latch is set, and the machine stops. Reset and re-IPL are the only means of resetting a hard stop condition.

If the Diagnostic Control switch is not in the PROCESS position, the action taken for a particular switch position is described in Chapter 10.

Program Checks

The Central Control Unit hardware monitors the control program operation for proper instruction execution and indicates to both the control program and the operator when a program check is detected.

Detection of a program check causes a level 1 interrupt request to be set. If the Display/Function Select switch is in the STATUS position, the type of check is indicated in the Display B lights on the control panel. The control program can determine the cause of the check by executing an Input X'7E' instruction and testing the register bits. Program action may vary according to the type of check detected; see the descriptions of each check (below) for appropriate action.

If error recovery for the program check is successful, the control program can reset the interrupt request by executing an Output X'77' instruction (miscellaneous control) with byte 1, bit 5 set to 1.

A program check while level 1 is active is indicated in the CCU check register (Input X'7D') by byte 0, bit 3 being set to 1, and the type of check is indicated in Input X'7E'. This condition causes an automatic IPL unless the Diagnostic Control switch on the control panel is in one of the check-control positions.

Below is a description of each of the four program checks.

In/Out Check
Input and output instructions are privileged instructions that can be executed only in program level 1, 2, 3, or 4. Any attempt to execute an input or output instruction at program level 5 causes a level 1 interrupt and sets the In/Out Check bit in the CCU interrupt request group 1 register (Input X'7E').

The In/Out Check bit is also set when an input or output instruction is executed at program level 1, 2, 3, or 4 with an external register address that is either not assigned or not recognized by any adapter. For example, issuing an Input X'38' instruction (non-existent) or an Output X'52' (for Type 2 Channel Adapter only) when a Type 1 Channel Adapter is installed sets the In/Out Check bit.

If an In/Out Check is detected while in program level 1, a no-op is performed, and the Program Check in Level 1 bit in the CCU check register is set along with the In/Out Check L1 interrupt request. The In/Out check is then handled as a CCU check.

Protection Exception
Whenever the control program attempts to modify a storage location that is protected and does not have a matching protect key, the result is a protection exception. A protection exception sets the Protection Check L1 interrupt request and suppresses storage access.

When a protection exception is detected during a cycle-steal operation, it is signaled to the Type 2 Channel Adapter involved as an addressing error. The adapter immediately terminates its cycle-steal operation and sets its check-interrupt request bit assigned to program level 1.

If a protection exception occurs when storage is

accessed to execute an instruction, an effective no-op is performed, and the instruction address register (IAR) is not updated. If the exception occurs in program level 2, 3, 4, or 5, a level 1 interrupt request is set. The control program can normally determine the address of the last instruction executed by examining the contents of the lagging address register (LAR).

A protection exception caused by an instruction attempting to modify a storage location can occur only when program level 5 is active. (The protect keys for the other program levels are fixed at 0 by the hardware design.) Such a protection violation causes the instruction to be suppressed and sets the level 1 interrupt request. The address of the instruction that caused the exception can be determined by examining the contents of the lagging address register (LAR).

If a protection exception is detected while in program level 1, the result is a no-op, and the Program Check in Level 1 bit in the CCU check register is set along with the Protection Check L1 interrupt request. The protection exception is then handled as a CCU check.

Invalid Op-code Check

The communications controller is limited to the 51 instructions described in Chapter 4. Each instruction has its own operation code bit structure. Whenever an attempt is made to execute an instruction with an op-code that does not compare to any of the 51 valid op-codes, the Invalid Op Check level 1 interrupt request is set and instruction execution is suppressed.

If an attempt is made at program level 2, 3, 4, or 5 to execute an invalid op code, an interrupt occurs to program level 1.

If an attempt is made at program level 1 to execute an invalid op code, the instruction execution is suppressed, and the Program Check in Level 1 bit in the CCU check register is set along with the Invalid Op Check L1 request. The invalid op code check is then handled as a CCU check.

Address Exception

An address exception occurs whenever an attempt is made to gain access to an uninstalled storage location for the given machine. For example, addressing storage location 52,304 when the controller contains only 49,152 bytes of storage causes an address exception.

When an address exception is recognized, an effective no-op is performed for the machine cycle in which it is detected. An Address Exception Check level 1 interrupt is then requested to inform the control program of the error.

An address exception during a cycle-steal operation is signaled to the adapter involved as an *address check* Upon the detection of an address check, the adapter immediately terminates its cycle-steal operation and sets its check interrupt request assigned to program level 1. This informs the control program of the error.

The following checks for address exceptions are made during instruction execution.

a. A check is made in the first cycle of each instruction as the storage address register (SAR) is loaded with the address of the instruction. If an address exception is detected, an effective no-op is performed and the instruction address register (IAR) is not updated. If an address exception occurs during the first cycle of an instruction executing at program level 2, 3, 4, or 5, a level 1 interrupt is taken and the address of the instruction last executed in that program level can normally be determined by examining the contents of the lagging address register (LAR).
b. A check is made in each subsequent cycle of two- or three-cycle instructions as the SAR is loaded with the address of the storage location to be accessed. If an address exception is detected, the operation in that cycle is suppressed. If an addressing exception occurs during the second or third cycle of an instruction executing at program level 2, 3, 4 or 5, a level 1 interrupt is taken, and the address of the instruction that caused the error can be determined by examining the contents of the lagging address register.

If an address exception is detected while program level 1 is active, the Program Check in Level 1 bit in the CCU check register and the Address Exception Check L1 bits are set to 1. The address exception is then handled as a CCU check.

CCU Diagnostic Facilities

The communications controller has diagnostic facilities designed to allow the control program to perform test procedures on the controller hardware. The diagnostic test routines can be either part of the online control program or a standalone control program used for testing purposes only. If these test routines are part of the online control program, the communication lines and adapters not being tested are allowed to continue operating.

To use the CCU diagnostic facilities of the control program, the CCU must be in the test mode. This mode of operation allows the use of certain bits and external registers that are not otherwise available to the control program. The test mode is entered by executing an Output X'79' instruction with byte 1, bit 2 (set test mode) set to 1 in the register specified by the R operand. Upon completion of testing, the control program should reset the test mode by executing an Output X'79' with byte 1, bit 3 (reset test mode) *on*.

While in the test mode, the following operations are available for program use:

1. Set and reset diagnostic L2 - Output X'77', byte 0, bit 6 sets a level 2 interrupt request that allows the program to execute a user diagnostic routine during normal program execution. When a level 2 interrupt occurs, the control program should test byte 1, bit 0 of the CCU interrupt request group 2 (Input X'7F') to determine if the interrupt is for diagnostic purposes. On completion of the diagnostic routine, the diagnostic L2 interrupt request must be reset by executing an Output X'77' with byte 0,bit 7 (reset diagnostic L2) set to 1.
2. Set and reset the bypass CCU check stop mode - Output X'79', byte 1, bit 4 set to 1 allows CCU check stops to be bypassed. If the Diagnostic Control switch on the control panel is in the PROCESS or the CLOCK STEP position, the bypass CCU check stop mode forces the check control hardware to operate as if the panel switch were in the BYPASS CCU CHECK STOP position. (See Chapter 10 for the function performed by the BYPASS CCU CHECK STOP position of the Diagnostic Control switch.) If the Diagnostic Control switch is in any other position, this operation is ignored. To reset the bypass mode, the control program must execute an Output X'79' instruction with byte 1, bit 5 set to 1.
3. Set and reset L1 adapter mask - Program level 1 interrupt requests caused by a check condition in an adapter (scanner or CA) can be masked by using an Output X'7E', byte 1, bit 1. This operation performs the same functions for the level 1 adapter checks as those described in the section on *Masking Program Level Priorities* in this chapter. To unmask the level 1 requests, the control program must execute an Output X'7F' (reset mask bits) with byte 1, bit 1 set to 1.
4. Force CCU checks - While in the test mode, the Output X'78' instruction is available for use by the control program. Setting of the bits of byte 0 in this output complements bits in the bytes being directed to the Arithmetic Logic Unit (ALU) to cause parity checks. The bits in byte 1 are used to complement storage data, Z bus parity, and cause A-register or Indata bus checks. See Appendix B, Output X'78', for a more complete description of the check conditions that can be forced by this output instruction.

Chapter 6: Type 1 Communication Scanner

This chapter gives the reader a basic understanding of the operation of the Type 1 Communication Scanner and the requirements necessary to program the scanner.

The Type 1 Communication Scanner (Type 1 Scanner) permits the control program to communicate with a line or autocall interface. This communication is done through the use of input and output instructions executed by the control program to the interface addressed by the communication scanner. The Type 1 Scanner makes data, status, and control information pertaining to each of the installed communication lines available to the program.

Most of the scanner functions are under direct control of the level 2 interrupt program and of processing routines that may be in other program levels. This permits increased flexibility by decreasing the number of restrictions caused by requirements of fixed hardware. The control program must assume the responsibility of the assembly and disassembly of characters, control character recognition, translation, and line control. Character assembly and disassembly are required of the program because the Type 1 Scanner transfers only one information bit at a time to or from the interface.

Operation and Data Flow

The Type 1 Communication Scanner hardware operates asynchronously with the other functional components of the controller. See Figure 18 for the Type 1 Scanner general hardware operation.

A *scan counter* sequentially addresses each interface in search of a service request. If a bit service request is detected, the scanner stops on that interface and requests a level 2 interrupt. All data and information for that interface is then made available to the control program through input instructions. When the program executes the proper output instructions, the scanner hardware passes information to the interface hardware.

When a character service request is detected, the control program is notified by a level 2 interrupt request. The scanner does not stop for a character service interrupt, but continues generating addresses in search of a bit service request.

Programming for the Type 1 Scanner should normally be done in the level 2 interrupt program because of the high priority placed on communication lines. However, the transferred bits and characters may be processed in a lower priority program level. See Figure 19 for a general flow of the Type 1 Scanner level 2 program.

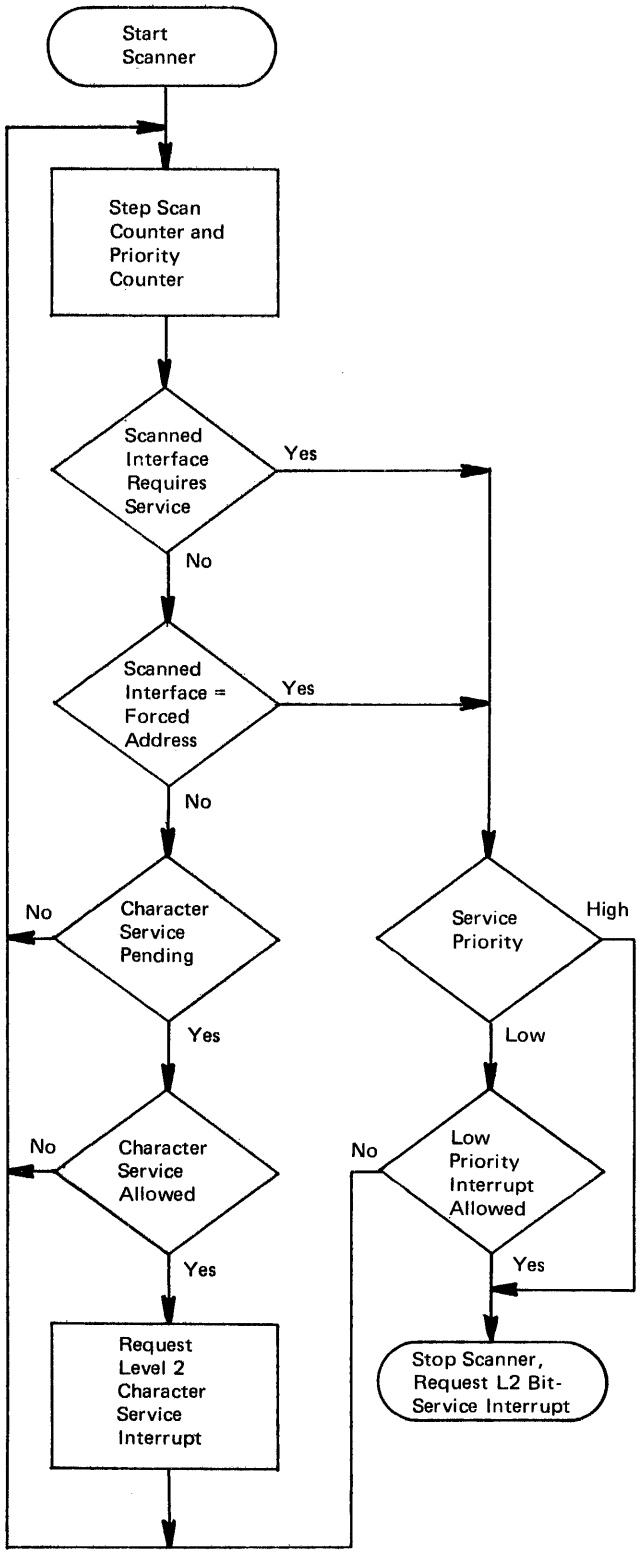


Figure 18. Level 2 Program Flow

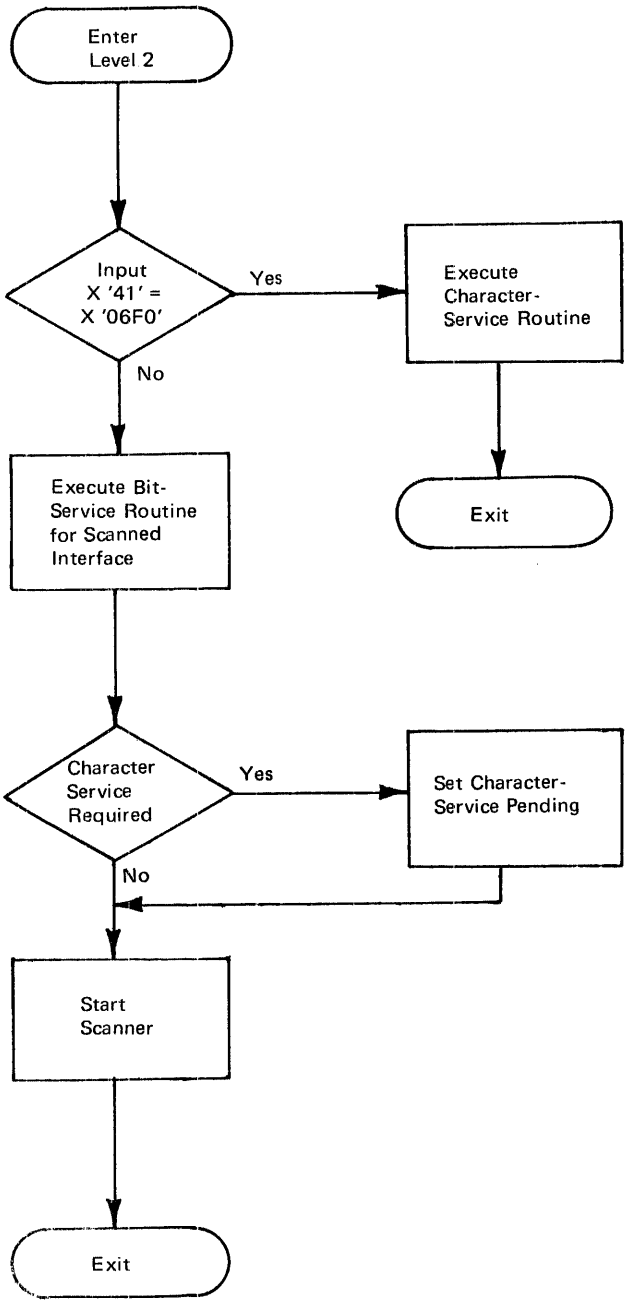


Figure 19. Level 2 Program Flow

When a level 2 interrupt occurs, an Input X'41' (interface address) instruction should be the first instruction executed to determine the cause of the interrupt. A branch can then be taken to the bit service or character service routine to handle the request.

The bit service routine determines if a bit has actually been received or must be transmitted. If a bit has been received, the routine places the received bit in the proper storage location allocated for the character to be received. If a bit is to be transmitted, the routine must remove a bit from the character and pass it to the interface. When the last bit of a character has been received or transmitted, the bit service routine must request a character service interrupt from the scanner hardware.

Before exiting the level 2 bit service routine, the program must restart the scanner by executing the proper output instruction (Output X'41' or Output X'46').

Interface Scanning

The 64 possible interface addresses (both used and unused) assigned to the four LIB positions are scanned sequentially. These lines are physically addressed by stepping a six position *scan counter* that generates an interface address. As the scan counter steps through each interface address, a check is made to see if that interface has a request for service. If a service request is present, the scanner checks the service priority and mode of the interface. If interrupts are allowed, the scanner determines whether the request is for bit service or character service. If bit service is required, the scanner *stops* on the interface and causes a level 2 interrupt request. If the request is for character service, the scanner signals the control program by requesting a level 2 interrupt.

When the bit service routine has obtained all necessary information pertaining to that line, it must execute an Output X'41' (or Output X'46' to request character service). These output instructions reset the level 2 bit service interrupt request and restart the scan counter.

Note: *The scan counter, when not stopped to allow a particular interface to be serviced, is stepped at a 2.5 MHz rate (400 nanoseconds per interface).*

Bit-Service Priority

Higher-speed communication lines should be serviced more frequently than lower-speed lines. To do this, the Type 1 Scanner allows the program to assign one of two (high or low) bit service priorities to each interface. An Output X'42' (control A) instruction is used to set the service priority for a particular interface.

When an interface is scanned, its assigned service priority is checked to determine whether it has high or low priority. If the service priority is high, and the interface requires service on this scan, the scanner stops and causes a level 2 bit service interrupt request. This permits the program to service that interface. A high-priority interface is serviced each time the line is scanned and bit service is required.

If the service priority is low, and the interface requires service, the scanner stops only if the interface

being addressed is the first low-priority line encountered on that cycle of the scan counter. (Refer to Figure 20.) To simplify this concept, Figure 20 shows only the first ten lines, and all are assumed to have a bit service request each time the scanner addresses them. Also assume for this example that a cycle is one complete scan of all interface addresses. When the scanner addresses the first low priority interface requiring service and the control program executes an Input X'41' instruction, the priority counter is reset to zero. The priority counter then adds a one to its count for each interface scanned (high or low priority) following the reset. As the scanner encounters the next and any other low priority interface, the priority counter is checked to see if a low priority interrupt is allowed. Low priority interrupts are not allowed until the counter reaches 65. When the scanner stops on the next low-priority interface and the control program executes an Input X'41' instruction, the *priority counter* is reset to zero and again, low-priority interrupts are not allowed until the counter again reaches 65.

Programming Note
All interfaces are set to high priority by a power-on-reset, IPL, machine reset, or a scanner disable (Output X'45' byte 0, bit 2).

Interrupt Requests

The Type 1 Scanner can initiate interrupt requests for either level 1 or level 2 service. Level 1 requests occur when the scanner detects an error condition affecting interface or scanner operation. Two different types of level 2 interrupt requests can occur for the purpose of handling normal service requests: the level 2 bit service request and the level 2 character service request. The following paragraphs describe the bit service and character service interrupt requests.

Bit Service Interrupt Request

The Type 1 Scanner level 2 bit service interrupt request occurs when the scanner stops on the address of an interface requesting service. Once the scanner has stopped on an interface, the program can identify the interface by executing an Input X'41' (interface address) instruction. Input X'41' loads a general register with the storage address associated with the interface requesting service. (See Figure 12 to determine the interface address and storage address association.) With the scanner stopped and the interrupting interface identified, a bit service routine can gain access to various control latches and status signals in the interface hardware to allow servicing of the bit request.

When enough information has been exchanged between the program and the interface hardware, the program must execute an output instruction to restart the scanner. An Output X'41' instruction is used to reset the interrupt request and start the scanner when the bit received is not the last bit of a character. An Output X'46' instruction sets the character service pending indication in addition to resetting the interrupt request and starting the scanner when character service is required.

Programming Note
The instruction to start the scanner should be executed as early as possible in the bit servicing program so the scanner delay (the time required to address another interface) can be overlapped with bit service processing.

Character Service Interrupt Request

When a bit service routine detects that character service is required, it should execute an Output X'46' instruction to set the 'character service pending' latch, reset the level 2 bit service interrupt request, and start the scanner.

Once the 'character service pending' latch is set, the scanner allows a character service interrupt request to be set when any of the following two events occurs four times, either separately or in combination.

1. The scanner makes a complete cycle, addressing all interfaces without detecting a bit service request.
2. The scanner passes an enabled high-priority interface that does not request bit service.

When a character service interrupt is set, the address provided by the Input X'41' instruction (interface address) is forced to X'06F0'. This address identifies the interrupt as being caused by a character service request and can be used to direct the control program to a character service routine.

The Type 1 Scanner has only one 'character service pending' latch, therefore, the control program should ensure that this latch remains set until the character service routine processes all pending character service requests. The control program should then execute an Output X'44' instruction with byte 1, bit 4 set to 1 to reset the 'character service pending' latch and the level 2 interrupt request.

Programming Note
A character service interrupt request does not stop the scanner; therefore, during the processing of a character, a bit service interrupt request can also be set. However, the request is not honored until an Exit instruction is executed in level 2 character service.

Scanner and Interface Initialization

Initialization is required to place the Type 1 Scanner and the attached interfaces into the proper mode for operation. This must be done any time the Type 1

Line Number	1	2	3	4	5	6	7	8	9	10
Priority	H	H	L	H	L	H	H	L	L	H
1st Cycle	X	X	X*	X		X	X			X
2nd Cycle	X	X		**	X*	X	X			X
3rd Cycle	X	X		X		**	X	X*		X
4th Cycle	X	X		X		X	X		X** *	X
5th Cycle	X	X		X		X	X			**
6th Cycle	X	X	X*	X		X	X			X

X = Interrupt Request Honored
* = Priority Counter Reset to Zero to Prevent Further Low Priority Interrupts
** = Priority Counter Incremented to 65 to Allow Low Priority Interrupts

Figure 20. Interfaces Serviced According to Priority

Scanner has become disabled so that all interfaces have been reset and no interrupts occur.

Conditions such as 'power on reset' and 'initial program load' cause the Type 1 Scanner and the attached interfaces to be disabled. The control program can also cause the same condition by executing an Output X'45' instruction with byte 0, bit 2 set to 1.

In the reset or disabled state, the scanner and all interfaces are reset to a non-operational state. The scan counter continues to run, but no interrupts are permitted, and all interfaces are set to high bit service priority. A minimum delay of 30 microseconds is required before the program can enable the scanner. The control program can enable the scanner and all the LIB positions by executing an Output X'45' instruction with byte 0, bit 1 set to 1 and byte 0, bits 4-set to 0.

Each interface address to be used for communications must be initialized. Once the scanner has been enabled, an individual interface may be enabled or disabled at any time.

When an interface is disabled, normal service interrupts for that interface are prohibited. To enable an interface, the control program must execute an Output X'47' (force bit service) with the associated storage address set in byte 0, bits 6-7 and byte 1, bits 0-3. This causes the scanner to stop when it reaches that interface address and to set a level 2 interrupt request. With the scanner stopped, the interface can be initialized for operation with an Output X'42' (control A) instruction.

The Output X'42' is a control instruction and includes (1) setting the interface mode, (2) selecting high or low priority, (3) selecting start-stop or synchronous/business machine or modem clocking, (4) selecting data rate, and (5) selecting an oscillator. (See Appendix B for Output X'42' bit definitions.) When the mode bits (byte 0, bits 6-7) are set to a non-zero value, the interface is enabled to accept interrupts. (See *Interface Modes of Operation* in this chapter.) The interface is then ready to send or receive data. If the mode is set to 11, an interrupt request is set for the interface at each bit interval when that interface is addressed by the scan counter.

Programming Considerations

When the control program has a record or message to send to a teleprocessing device, it must request a bit service interrupt for each bit to be transmitted by executing an Output X'47' instruction containing the storage address associated with that line interface. This causes that interface to set a level 2 bit service request. The next time the scanner services this interface, it stops, and the level 2 interrupt is requested. The bit service routine must then disassemble the character to be sent and execute a series of output instructions to transmit the bits. For each bit to be transmitted, the program must execute an Output X'43' (control B) instruction with byte 1, bit 4 set to 1 to indicate a transmit operation, and byte 1, bit 7 set to the proper bit value (mark or space). The Output X'43' is executed once each time a level 2 interrupt occurs for the particular interface until the complete character is transmitted.

Interface Modes of Operation

The Type 1 Scanner provides four modes of operation for the interface. The modes are set by the combination of bits 6 and 7 of byte 0 in the Output X'42' instruction. The setting of these bits are:

Bit 6	*Bit 7*	*Mode*
0	0	Level 2 interrupts disabled.
0	1	Monitor ring-indicator or data-set-ready.
1	0	Monitor Receive-Data-Space.
1	1	Allow level 2 interrupts.

In the *disabled mode* (00), no interrupts are allowed from the interface until an Output X'47' (force bit service request) is executed. This generates one bit service interrupt request. When the interrupt occurs, the mode can then be changed to allow interrupts (11) for normal operation. This applies to all transmit and receive operations.

The *monitor ring-indicator* or *data-set-ready mode* (01) allows the scanner to monitor for ring-indicator or data-set-ready on a low-priority interface without causing an interrupt each bit time. When either the ring-indicator lead or the data-set-ready lead becomes active, a bit service interrupt is requested. When the scanner stops on this interface to honor the request, the program should set the mode to allow interrupts (11) for normal operation. When the program has determined that the interface has completed the required action, the mode should be returned to a monitor mode to wait for the next request.

If the interface service priority is high, that interface is disabled from causing a level 2 interrupt in the monitor ring-indicator or data-set-ready mode. A force bit service (Output X'47') or a diagnostic bit service (Output X'44' byte 1, bit 0) is required to override the mode setting and cause a level 2 interrupt.

The *monitor receive-data-space mode* (10) allows the interface to remain idle in a 'mark' state without causing an interrupt each bit time. However, when the line changes to a 'space' level, signaling the start of activity, a bit service is requested. When the scanner stops on this interface to honor the request, the pro-
the interface to remain idle in a 'mark' state without causing an interrupt each bit time. However, when the line changes to a 'space' level, signaling the start of activity, a bit service is requested. When the scanner stops on this interface to honor the request, the program should set the allow interrupt mode (11) for normal operation. When the program has determined that the interface has completed the required action, the mode should be returned to a monitor mode to wait for the next request.

The *allow interrupt mode* (11) permits interrupt requests to be serviced at each bit interval when the scanner addresses that interface.

The *mode bit override* function is initiated by executing an Output X'40' instruction. Setting the 'mode bit override' latch overrides all mode settings except mode 01 at high-priority. When this latch is set, all interfaces set to any mode other than 01 and to high-priority, request a level 2 interrupt each bit-interval regardless of that interface's mode setting. This function allows the control program to force bit service on multiple interfaces without requiring an Output X'47' instruction for each interface address.

Programming Note

When in the monitor ring-indicator or data-set-ready mode, the scanner hardware does not service bit service requests. Therefore, a bit overrun/underrun may be indicated when a level 2 interrupt is forced. The bit overrun/underrun indication should be ignored and reset by the control program. When in the monitor for receive data space mode, the hardware services bits while monitoring for a space. Therefore, the control program should always test for a bit overrun/underrun indication.

Business Machine Clocks

The Type 1 Scanner must have at least one business machine clock installed and may have as many as four. If modem clocking is used with any of the lines, a business machine clock must be installed in the scanner with a speed one-half or less than that of the lowest speed modem clock. Figure 21 lists the business machine clocks available.

Clock Speed	*Power On Warm Up Period (Seconds)*
45.5 bps	5
50.0 bps	4
56.89 bps	20
74.2 bps	5
75.0 bps	5
100.0 bps	4
110.0 bps	3
134.5 bps	2
150.0 bps	less than 1
200.0 bps	less than 1
600.0 bps	less than 1
1200.0 bps	less than 1
2000.0 bps	less than 1
2400.0 bps	less than 1

Figure 21. Business Machine Clocks and Warm Up Period.

For line speeds greater than 2400 bps, an external data set must provide the clock pulses. Line Sets 1A, 1C, 1F, 2A, 3A, 3B, 4A, 4B, and 4C must operate with business machine clocking. Line Sets 1D and 1E can operate under business machine or modem clock control. Refer to Appendix D for a description of the individual LIB and line set types.

The installed business machine clock used for a given line is selected under program control by executing an Output X'42' (control A) instruction with byte 1, bits 6 and 7 set to indicate the desired clock. Figure 22 shows the proper setting of the oscillator select bits to assign an installed oscillator to a given interface.

Bit 6	7	*Selected Business Machine Clock*	
0	0	Lowest speed clock	(OSC0)
0	1	Next higher speed clock	(OSC1)
1	0	Next higher speed clock	(OSC2)
1	1	Highest speed clock	(OSC3)

Figure 22. Type 1 Scanner Business Machine Clock Selection

No business machine clock is selected if the Output X'42' bits are set to select an uninstalled oscillator (for example, bits 6 and 7 set to 11 when only two or three oscillators are installed).

Every interface must have a business machine clock assigned whether it is specified to be business machine or modem clocked. For autocall interfaces and for line interfaces that are to use modem clocking, the assigned business machine clock is used to ensure that the interface is periodically accessed. The lowest speed oscillator is always used for an autocall interface.

The oscillator select bits are set to 0 by a reset to the scanner. Therefore, the lowest speed clock is initially selected, and unless an Output X'42' is executed to select another clock for a given interface, the lowest speed clock is used.

After a power-on reset occurs, there is a warm-up period associated with the different clocks. (Refer to Figure 21.) During this warm-up period, a business machine clock cannot provide bit service requests.

Programming Notes

1. The oscillator select bits for a line interface can be changed without causing a switched network connection to be broken, if *data terminal ready* is up when the Output X'42' is executed.
2. The business machine clock selected for a modem-clocked line interface must be one-half or less than the rate of the modem clock.

Autocall Interface Operation

Programming the Type 1 Scanner for an autocall interface is essentially the same as for a line interface. The major difference is in the bit settings of Input instructions X'42' and X'43' and Output instructions X'42' and X'43'. Appendix B defines the affected instructions and the differences in the bit definitions.

Input/Output Instructions

The Type 1 Communication Scanner and the line interface bases it supports are controlled through the use of input and output instructions. These instructions allow the program to (1) control the attached interfaces, (2) send and receive data, and (3) monitor the status of the scanner and line interfaces to ensure proper operation. Appendix B defines the bits within each input and output instruction.

Certain input and output instructions should be executed only when the scanner is stopped. If an Input X'41', X'42', or X'43' instruction is executed when the scanner is running, the result in the general register specified in the instruction pertains to the interface addressed at the time of execution. However, there is no way to determine which interface was addressed. This may also cause a level 1 interrupt due to incorrect parity. If an Output X'41', X'42', X'43' is executed when the scanner is running, the mode or status of an unknown interface may be changed, and an error condition may result.

Input Instructions

Four functional input instructions (Input X'41', X'42', X'43', and X'44') are used with the Type 1 Communication Scanner. With these instructions, the control program receives data from the lines, monitors the status of the lines, and is informed of any error conditions. Execution of Input instructions X'40', X'45', X'46', and X'47' sets the bits in the general register to zero. (Appendix B defines the bits within each input instruction.)

Input X'41' (Interface Address)

This instruction loads a general register with the storage address associated with the interface the scanner is addressing. Each time the scanner stops because of a bit service interrupt, this input may be issued to determine which interface caused the request.

If the level 2 interrupt is the result of a character service request, the address loaded into the register will be X'06F0'. This address identifies the interrupt as being caused by a character service request and can be used to direct the control program to a character service routine.

Programming Note
An Input X'41' instruction should be the first instruction executed in level 2 to prevent the possible loss of a character service indication.

Input X'42' (Control A)
This instruction loads a general register with the 'control A' information as set by an Output X'42' instruction. An Input X'42' instruction is used to check the state of the control A bits. It is a direct bit-for-bit reflection of the last Output X'42' instruction to that interface.

The Input X'42' instruction should be executed only when the scanner is stopped.

Input X'43' (Control B/C)
This instruction can be executed in either a bit service or a character service routine. However, an Input X'41' instruction must be executed between the level 2 interrupt and execution of the Input X'43' instruction.

Executing an Input X'43' instruction as a result of a bit service interrupt loads a general register with the control B/C information. Data received from the interface, along with error information and line status, is set for the interface that caused the service request. If an Input X'43' is executed as the result of a character service interrupt, the specified general register (R) will contain X'0000'.

An Input X'43' instruction *must not* be executed if there is not a bit service or character service level 2 interrupt pending. Execution without an interrupt pending may cause a CCU In/Out level 1 check due to incorrect input parity.

Programming Note
This instruction cannot be executed immediately following an Output X'43'' for feedback checking. However, it can be executed after Output X'43' to obtain other status indications.

Input X'44' (Status Register)
This instruction loads a general register with the contents of the Type 1 Scanner status register. This register contains (1) indications of level 1 check interrupt requests from the scanner, (2) the scanner enable/disable condition, (3) character service pending information, and (4) mode bit override and override remember indications.

Output Instructions
Eight output instructions are used with the Type 1 Communication Scanner to control its operation and data transfer. (Appendix B defines the bits within each output instruction.)

Output X'40' (Set Mode Bit Override and Override Remember)
This instruction sets the 'mode bit override' latch and the 'override remember' latch. The function of the 'mode bit override' latch is described in the *Interface Modes of Operation* section of this chapter. The 'override remember' latch performs no hardware function but is available for use by the control program. Since this instruction performs a function, the bit settings of the register are ignored.

Output X'41' (Reset Bit Service Level 2 Request and Start Scanner)
This instruction resets the level 2 bit service interrupt request and starts the scanner. The bit service routine should issue this output after determining that all the information required to service the interface has been obtained or sent. An Output X'41' instruction should be the last instruction executed in the level 2 routine before exiting to a lower priority level for continued bit service processing. Since this instruction performs a function, the bit settings of the register are ignored.

The Output X'41' instruction should be executed only when the scanner is stopped. It does not reset the bit service request unless an Input X'41' was executed for that interface.

Output X'42' (Control A)
This instruction sets the mode of an interface according to the bit setting of the specified general register. In addition to setting the mode, it can also set service priority, clocking, data rate, osciallator selection, and diagnostic mode for the interface.

The Output X'42' instruction should be executed only when the scanner is stopped.

Output X'43' (Control B)
This instruction sets the interface into a transmit or receive mode and can activate or deactivate various line and autocall-interface leads as required. When the line is transmitting, byte 1, bit 7 of the general register must be loaded with the mark or space to be sent to the interface terminal.

The Output X'43' instruction should be executed only when the scanner is stopped.

Output X'44' (General Control)
This instruction sets or resets the diagnostic bit service request and resets (1) the mode bit override and override remember, (2) character service pending, and (3) outstanding error indications.

Output X'45' (Scanner Control)
This instruction enables or disables the scanner and/or

the line interface bases (LIBs). Normally, this instruction is used only during initialization or when an error has occurred that requires a LIB to be disabled.

Programming Note
Output X'45' can disable interrupts from any LIB attached to the Type 1 Scanner. This can be particularly useful when a bit-clock error occurs in a LIB and causes a level 1 interrupt. The failing LIB, including all lines attached to that LIB, can be disabled to prevent further error conditions while the remaining LIBs continue normal operation.

Output X'46' (Set Character Service Pending, Start Scanner, and Reset Level 2 Bit Service Request)
This instruction is normally used at the end of bit service processing when character service is required for that interface. It sets the 'character service pending' latch to signal the scanner that a character service interrupt is required. The instruction then resets the level 2 bit service interrupt request and starts the scanner. Since this instruction performs a function, the bit settings of the register are ignored.

The Output X'46' instruction should be executed only when the scanner is stopped. It resets the bit service request only if an Input X'41' was executed for that interface.

Output X'47' (Force Bit Service Request)
This instruction forces a bit service interrupt request for the interface address that is specified in the general register. This instruction stops the scanner on an interface and requests a bit service interrupt so that the program can enable the interface or access it to transmit a bit.

Programming Note
Forced bit service cannot be stacked. If an Output X'47' is executed before the previous Output X'47' has been serviced, the second address overlays the first.

Error Indications

Error conditions detected by the Type 1 Scanner are in one of two groups, depending on the type of error and the impact on the overall system operation. The first and most critical group (level 1 errors) causes a level 1 interrupt request and must be handled with high priority because the error may involve many lines. The second group (interface errors) can be handled at a lower-priority interrupt level because they have less system impact. The following paragraphs describe these errors and their detection.

Level 1 Errors
Failures in the Type 1 Communications Scanner or in a Line Interface Base can affect all communication lines attached to the controller or at least a group of lines within a particular LIB. The detection of one or more of these failures by the hardware check circuits causes a Type 1 Scanner L1 interrupt request (Input X'76', byte 0, bit 1). The level 1 interrupt routine, after determining that the interrupt request came from the Type 1 Scanner, should execute an Input X'44' (status register) instruction to further identify the error.

The error condition that caused a level 1 interrupt is indicated in byte 1, bits 2-7 of the Type 1 Scanner status register (Input X'44'). Bits 2-5 correspond to LIB positions 1-4 and are turned *on* respectively as a result of a LIB bit clock parity error. Bit 6 is turned *on* by the detection of a LIB select error. Bit 7 is turned *on* by a parity error on the CCU outbus, which is an internal interface between the Central Control Unit and the Type 1 Scanner See Appendix B, Input X'44' for a description of these bits and the error conditions.

If the level 1 interrupt routine can handle the error condition and processing can continue, the routine should then issue an Output X'44' instruction with byte 1, bit 5 on to reset the level 1 request. When the error is permanent, the LIB can be disabled via an Output X'45' so that processing can continue on the remaining LIBs.

Interface Errors
Line interface errors indicate intermittent or permanent internal logic faults and most problems with communciation facilities. The failure is detected at the interface level, but if failures are detected in a group of interfaces, the fault may be in either the (1) LIB logic, (2) Type 1 Scanner logic, (3) CCU input/output mechanism, or (4) program logic.

Interface errors normally are not critical enough to interrupt the entire system. Therefore, the Type 1 Scanner does not generate a level 1 interrupt when this type of error is detected. Instead, certain bits are set in the control B/C register to indicate the failure. By issuing an Input X'43' instruction when an interface requests service, the program can test for error conditions on that line without disrupting normal processing.

Byte 0, bit 2 of the Input X'43' instruction is a partial summary of interface errors. This interface-error summary bit is set *on* whenever the Type 1 Scanner detects (1) a feedback error, (2) bit overrun or underrun, or (3) that the 'data set ready' line is not up. If the control program checks this bit first, considerable

time can be saved in detecting errors.

A feedback error (byte 0, bit 1) is set on when the hardware circuits detect that the bit actually sent to the line set does not compare to the bit as it appears in byte 1, bit 7 of Output X'43'. This error is also set if the interface bit service failed to reset. A feedback check must be reset before the scanner can be restarted.

Bit overrun/underrun (byte 1, bit 7) is set when the Type 1 Scanner determines that a bit has been lost because of improper timings between the control program and the bit rate used by an interface.

Telegraph Echo Check (byte 1, bit 4) is set when the telegraph interface detects that an echo check has occurred.

Diagnostic Functions

The Type 1 Communication Scanner provides for three internal diagnostic functions: (1) diagnostic bit service, (2) diagnostic mode, and (3) IBM modem self-test. These tests run under the control of the scanner program and can provide online testing as described in the following sections. Diagnostic bit service can be issued to an autocall interface, but the diagnostic wrap and the modem self-test cannot.

Diagnostic Bit Service

The Type 1 Scanner diagnostic bit service provides a means for forcing level 2 bit service interrupt requests. This facility allows the control program, through the use of a diagnostic routine, to exercise program and/or hardware functions in a test environment. The diagnostic routine performed must be part of the control program. Diagnostic bit service causes continuous level 2 bit service requests for all 64 interface addresses whether the interface is used or not. An Output X'44' instruction with byte 1, bit 0 set to 1 indicates the diagnostic bit service function. When an Output X'44' is executed with this bit off (0), the diagnostic requests are terminated.

Diagnostic Wrap Mode

The Type 1 Scanner diagnostic wrap provides a means of testing and locating defects in the line control logic and in the line-interface transmit and receive logic. Diagnostic wrap can be performed online without affecting normal program operation or the lines not in diagnostic mode. The test requires one line interface to act as a transmit line and one or more line interfaces to act as receive lines. Any line can be a transmit or a receive line; however, only one diagnostic wrap transmit line may be present at any one time.

Diagnostic wrap is initiated by executing an Output X'42' instruction to each line to be tested with the following bits set in the register specified by the R operand.

Byte 0, bits 6-7: (Mode Bits 1 and 2) - These bits select the appropriate bit setting for the desired mode. bit setting for the desired mode. See *Interface Modes of Operation* in this chapter for the mode options.

Byte 1, bit 0: (Bit Service Priority) - This bit selects the appropriate service priority. See Output X'42' for service priority options.

Byte 1, bit 1: (Diagnostic Mode) - This bit must be 1.

Byte 1, bit 2: (Data Terminal Ready) - This bit must be 0.

Byte 1, bit 3: (Synchronous Clock) - This bit must be set according to the type of communication line to be tested. A 1 is set in this position for binary synchronous lines, and a 0 for start-stop lines.

Byte 1, bit 4: (External Clock) - This bit must be 0.

Byte 1, bit 5: (Data Rate Select) - This bit may be either 0 or 1. However, the same type line sets must use the same data rate.

Byte 1, bits 6-7: (Oscillator Select 1 & 2) - These bits select an available line oscillator (business machine clock). For bit clock options, see *Business Machine Clocks* in this chapter. All wrap-test lines must select the same oscillator.

After the Output X'42' instructions are executed, the affected lines can be used through any sequence of point-to-point or multipoint operations.

Diagnostic wrap mode simulates 'data set ready' and 'receive line signal detect' as active lines. 'Clear to send' is simulated as active if 'request to send' is active.

Programming Note

1. Only one line may be in a diagnostic wrap transmit state at any given time during the operation.
2. The line used for transmit should be the last line to be issued the Output X'42' instruction.

Modem Self-Test

Modem Self-Test tests the scrambler circuits of certain IBM modems under program control. The modem test can be performed online without affecting the normal operation of other lines. This test may also be performed simultaneously on any or all lines for which the

test function is initiated.

The modem self-test operation can be performed for IBM 3872 and 3875 modems externally attached to line interfaces provided by Line Set 1D.

Modem self-test is initiated by executing an Output X'42' (control A) instruction with the following bits set in the register specified by the R operand to each line to be tested.

Byte 0, bits 6-7: (Mode Bits 1 and 2) - These bits select the appropriate bit setting for the desired mode. See *Interface Modes of Operation* in this chapter for mode options.

Byte 1, bit 0: (Bit Service Priority) - This bit selects the appropriate service priority. See Output X'42' for service priority options.

Byte 1, bit 1: (Diagnostic Mode) - This bit must be 1.

Byte 1, bit 2: (Data Terminal Ready) - This bit must be 1 to cause the 'data terminal ready' latch to be set in the line interface. When this bit, together with diagnostic mode, is set on, the modem self-test performs instead of the diagnostic wrap test.

Byte 1, bit 3: (Synchronous Clock) - This bit must be 0.

Byte 1, bit 4: (External Clock) - This bit must be 1 to indicate modem clocking because the IBM modem provides the clock pulses.

Byte 1, bit 5: (Data Rate Select) - This bit may be either 0 or 1.

Byte 1, bits 6-7: (Oscillator Select 1 and 2) - These bits must select an internal oscillator whose speed is less then one-half the clock speed provided by the IBM modem. For the proper setting, see *Business Machine Clocks* in this chapter.

Chapter 7: Type 2 Communication Scanner

This chapter gives the reader a basic understanding of the operation of the Type 2 Communication Scanner and the requirements necessary to program the scanner.

The Type 2 Communication Scanner (1) scans the interface addresses assigned to the LIB positions it supports, (2) performs character assembly/disassembly, (3) provides character buffering, and (4) causes program interrupts when character service is required.

Up to four Type 2 Communication Scanners can be installed in the 3705 (Type 2 Scanner-1 through Type 2 Scanner-4). Type 2 Scanner-1 supports attachment of up to four LIBs with 64 half-duplex (HDX) lines. Type 2 Scanner-2, Type 2 Scanner-3, and Type 2 Scanner-4 can each support attachment of up to 6 LIBs with 96 HDX lines; thus, up to 352 HDX lines can be attached to the 3705 using four Type 2 Scanners. The Type 2 Scanners can be installed with either the Type 1 or Type 2 Channel Adapter features.

Operation and Data Flow

The interface addresses for all installed Type 2 Communication Scanners are generated from a common Type 2 Attachment Base. (See Figure 23.) A continuous running *scan counter* in the attachment base places the generated interface address on an address bus that goes to all scanners simultaneously. This address can be modified, under program control, by the attachment base or the scanner.

The interface address is then used to address an interface control word (ICW) which is loaded into the ICW work register where the scanner hardware determines if any action is to be performed for that interface. If no action is required, the ICW is replaced in local store and the next addressed ICW is loaded into the work register. If the scanner determines that program intervention is required, it requests a level 2 interrupt and loads the interface address into an interrupt priority register.

When the level 2 interrupt actually occurs, the address in the highest-priority interrupt priority register that is active is loaded into the attachment buffer address register (ABAR) and is then available to the control program along with the ICW in the ICW input register.

Type 2 Scanner Registers

The Type 2 Scanner contains various hardware registers that are used to store and pass information and data within the scanner and between the scanner and the control program. Some of these hardware registers are available to the control program as external register addresses through input and output instructions. The external registers required for control program access are described in the following paragraphs.

Local Storage

Each Type 2 Scanner contains a local storage array to store the interface control words not being used by the scanner hardware or the control program. This storage array holds 96 control words of 48 bits each (46 information bits and 2 parity bits).

ICW Work Register

The Type 2 Scanner control logic uses the ICW work register to access, monitor, and modify an interface control word (ICW). This register is loaded each time an ICW is read out of local storage.

ICW Input Register

The control program uses the ICW input register for access to the interface control words. This register is loaded from the ICW work register and reflects the status of the ICW at the time when it was read out of local storage.

Attachment Buffer Address Register

The attachment buffer address register is (ABAR) is physically located in the Type 2 Attachment Base and supplies the interface addresses to the control program. See *I/O Programming Considerations* in this chapter for a description of loading the ABAR.

Programming Note

The ABAR must be initialized by an Output X'40' instruction with an interface address associated with an installed Type 2 Scanner. The Output X'40' must be executed after the controller is powered on and before any other input or output instructions.

Display Register

The display register is a temporary storage register that can contain interface control information that the program can use. If bit 38 (display request) of an interface control word is *on*, control information for that interface is loaded into the display register each time that interface is scanned. The control program can then obtain this information by executing an Input X'46' instruction.

There is only one display register for each Type 2 Scanner; therefore, only one ICW should have its display request bit set to 1 at a time. Otherwise, the control program cannot determine which interface was the last one to cause the display register to be loaded.

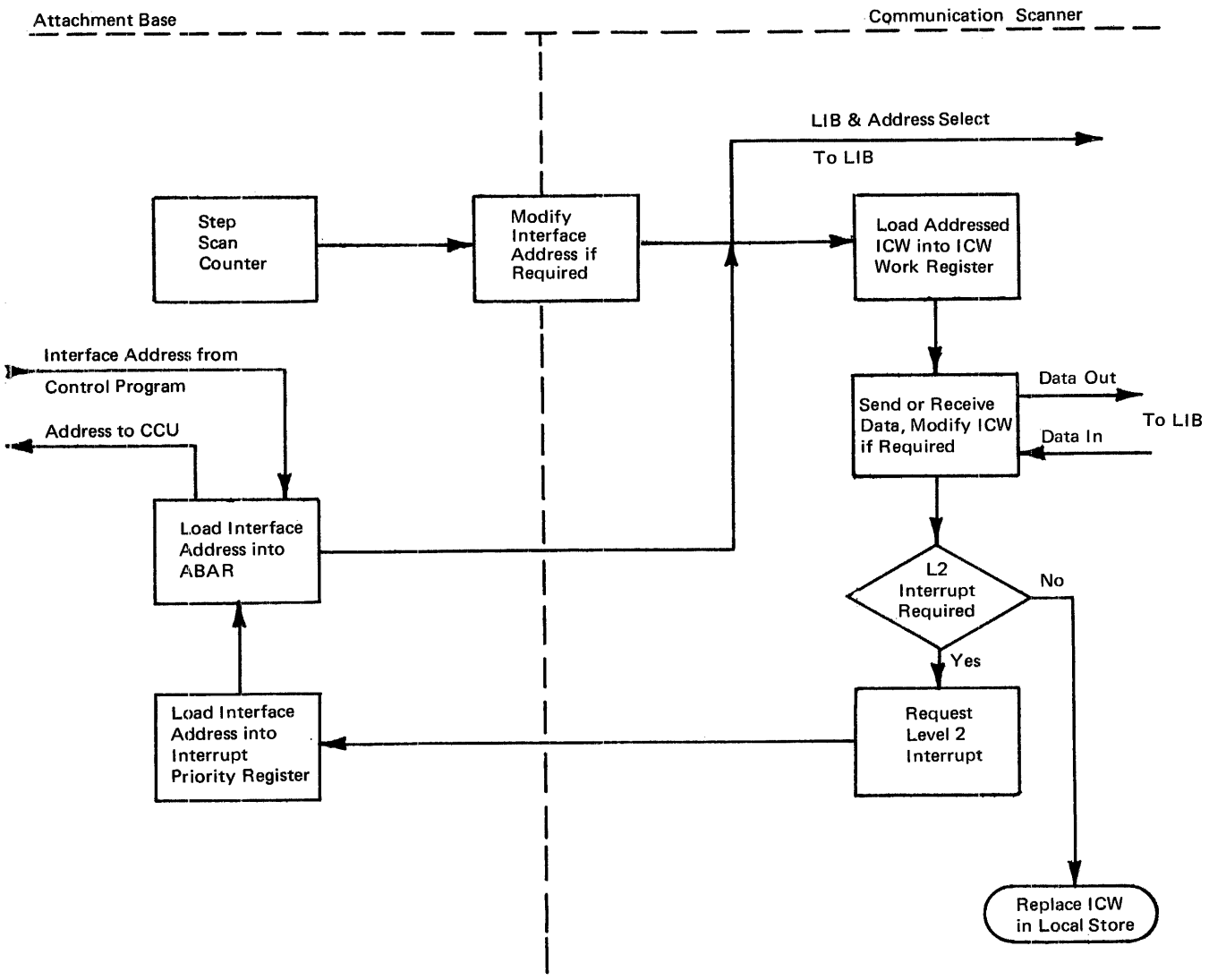


Figure 23. Type 2 Communication Scanner Operation

Type 2 Scanner Addressing

The Type 2 Communication Scanner scan-addressing and program-addressing mechanism is controlled by the Type 2 attachment base. The Attachment Base generates the basic scan address and places it on a 'line address bus' for availability to all installed Type 2 Communication Scanners. Refer to the *Interface Addressing* section of Chapter 3 for a detailed discussion of each interface address bit.

Scan Addressing

For scan addressing, an interface in each installed Type 2 Scanner is addressed simultaneously. Each scanner derives the address of the interface it is scanning from the 8-bit address that the Type 2 Attachment Base places on a 'line address bus'. The line address bus is an internal bus that carries the scan address from the attachment base to each of the communication scanners. This address, as modified by each scanner (see upper scan limit), is used not only to select a particular interface but also to address the associated interface control word (ICW) that the scanner maintains in local storage. (See *Interface Control Word* section in this chapter.) The Type 2 Scanner examines this ICW and, when an interface service function is required, performs that function; or, when a character service requires programming action, the Type 2 Scanner signals the attachment base that it needs a program level 2 interrupt.

Scan Counter

The Type 2 Attachment Base scan counter output provides the basic scan addresses for each Type 2 Scanner. If the scan counter output is not modified,

each Type 2 Scanner sequentially scans 96 interface addresses. Under these circumstances, the Type 2 Scanner cannot handle line speeds higher than 4800 bps without having the possibility of undetected bit overrun/underrun conditions. However, the ability to substitute some interface addresses (address substitution) and set a limit on the number of interfaces scanned (scan limit), greatly extends the capability of handling higher-speed lines. These mechanisms cause the scan counter output to be modified to allow certain interface addresses to be scanned at a different rate.

Upper Scan Limit

The Type 2 Scanners have an upper scan limit that can be set and reset under program control by an Output X'42' instruction. Each scanner maintains its own upper scan limit and is independent of the limits set by any of the other installed scanners. Based on the state of its 'upper scan limit' latches, a Type 2 Scanner may modify the 'scan counter' output from the Type 2 Attachment Base in such a way as to limit the number of interface addresses scanned.

The actual modification of the scan address is done by the Type 2 Scanner hardware as the line address bus enters the scanner from the attachment base. Figure 24 shows the number of interfaces scanned and the LIB position affected for each setting of the upper scan limit. When the upper scan limit is set to any value other than 00, the scanner modifies the addresses above the limit to start at the first address again. For example, if the upper scan limit is set to allow only 16 interface addresses to be scanned, when the scan counter output to that scanner reaches the 17th address, the address is modified to scan the first address again. This decreases the period of time between successive scans of the remaining interface addresses to accommodate higher-speed lines. In this case, the scanner with an upper scan limit of 11 scans the first 16 interfaces 4 times in the same period of time as another scanner with no limit scans 96 interfaces.

Address Substitution

The Type 2 Attachment Base can modify the output of its scan counter to cause certain addresses assigned to LIB position 1 to be substituted on the 'line address bus' in place of the normal scan addresses. As a result, those addresses that are substituted are scanned by each Type 2 Scanner more frequently than other addresses. Address substitution is controlled by the attachment base and therefore affects all installed scanners in the same manner. When operating in this manner, the attachment base forces each Type 2 Scanner to scan the substituted address, or addresses, with an effective scan period of 12.8 microseconds since address substitution occurs every eighth time the 'scan counter' changes state. This allows the substitution address, or addresses, in each scanner to handle higher line speeds, independent of the state of the scan limit.

Address substitution is controlled by the state of a 4-bit register in the Type 2 Attachment Base called the substitution control register. The bits of this register may be set under program control by Output X'41' byte 1, bits 2, 3, 4, and 5. Each bit of the substitution control register corresponds to one of four substitution addresses assigned to LIB position 1.

Programming Note

Any combination of the four substitution control register bits may be turned *on* to produce the desired substitutions. If address substitution is not used, Output X'41' must be executed with byte 1, bits 2 through 5 *off* in the register specified by the R operand.

When a given substitution control register bit is on, a corresponding address is substituted on the 'line address bus' every eighth time the scan counter

*Upper Scan Limit**	*Number of Interfaces Scanned*	*Interface Addresses Scanned*	*Interface Addresses Not Scanned*	*Scan Period (usec)*
00	96 **	Addr 0-F,LIB 1-6	-	153.6
10	48	Addr 0-F,LIB 1-3	Addr 0-F,LIB 4-6	76.8
11	16	Addr 0-F,LIB 1	Addr 0-F,LIB 2-6	25.6
01	8	Addr 0-7,LIB 1	Addr 8-F,LIB 1 & Addr 0-F,LIB 2-6	12.8

* Set by Output X'42' byte 1, bits 6-7
** Scanner-1 contains 96 addresses but only uses the first 64.

Figure 24. Upper Scan Limit

changes state. Combinations of bits on in the substitution control register results in fixed-address substitution for each corresponding bit. Figure 25 shows which address is substituted and which addresses are not scanned as a result of that substitution when the different substitution control register bits are on.

Output X'41' Byte 1, Bit:	*Fixed Address Substituted in Each Type 2 Scanner If Substitution Bit ON*	*Addresses Not Scanned In Each Type 2 Scanner If Substitution Bit ON*
2	Adr 0 LIB position 1	Adr E in LIB positions 1-6 Adr F in LIB positions 1-6
3	Adr 2 LIB position 1	Adr C in LIB positions 1-6 Adr D in LIB positions 1-6
4	Adr 4 LIB position 1	Adr A in LIB positions 1-6 Adr B in LIB positions 1-6
5	Adr 6 LIB position 1	Adr 8 in LIB positions 1-6 Adr 9 in LIB positions 1-6

Figure 25. Attachment Base Address Substitution Control

Program Addressing

Various input and output instructions exist that allow the program to control the operation of the Type 2 Scanners, Type 2 Attachment Base and the individual interfaces.

However, before the program can examine or modify fields in an interface control word (ICW) associated with a particular interface, the address of that interface must be placed in the attachment buffer address register (ABAR) of the attachment base. Similarly, before the program can access certain registers in a particular Type 2 Scanner or perform control functions in that Type 2 Scanner, the interface address in the ABAR must be one of those assigned to that scanner. Two distinct events can cause the contents of the ABAR to be changed: (1) a program level 2 interrupt, and (2) execution of an Output X'40' instruction.

When a program level 2 interrupt (Type 2 Scanner L2) occurs, the contents of the ABAR are automatically set by the Type 2 Attachment Base with the interface address from the highest interrupt priority register that is occupied. The control program can determine which interface address is in the ABAR by executing an Input X'40' instruction. The program can then examine and/or modify fields in the ICW associated with this interface. In the other interrupt program levels (1, 3, and 4), the program may find it necessary, at times, to examine certain functions in the Type 2 Scanner. Furthermore, in program level 3 or 4 the program may need to gain access to the ICW associated with a specific interface. By executing Output X'40' under such circumstances, the program can cause the ABAR to be set according to the interface address in the register specified by the R operand.

To avoid conflicts with the automatic mechanism that sets ABAR when a program level 2 interrupt occurs, programs executing at program level 3 or 4 should mask program level 2 interrupts before executing Output X'40'. If more than one program level is likely to execute an Output X'40', additional interlocking conventions must be established between those program levels by the user.

Interface Control Word (ICW)

The Interface Control Word (ICW) provides the normal means by which the control program communicates with the Type 2 Scanner and the interface hardware.

The ICW is made up of 46 information bits and 2 parity bits and is physically located in the scanner local storage. Each scanner contains one ICW for each possible interface. However, even though the scanner contains the maximum number of ICWs (96), only those ICWs associated with an attached and active interface are used.

ICW Access

The Type 2 Scanner hardware gains access to an ICW by using the interface address provided by the interrupt priority register in the Type 2 Attachment Base. When the level 2 interrupt occurs, the address from the interrupt priority register is loaded into the attachment buffer address register (ABAR). The program can then execute an Input X'40' instruction to get the storage address associated with the interface. Once the control program obtains the interface address, it has access to the various fields of the ICW through input and output instructions.

Programming Note

ICW access at program level 3 or 4 should be performed only when program level 2 interrupts are masked off; otherwise, the result is unpredictable.

ICW Modification

Program access to the various fields in the ICW is through the use of input and output instructions. When the Input X'44', X'45', and X'47' instructions are executed, the ICW bits assigned to those inputs are placed in the register specified in the R operand. See Appendix B for the input/output instruction bit definitions.

The information obtained by the input instructions comes from the ICW input register. There is one ICW input register in each communication scanner. This register is automatically loaded from the ICW work

register when a level 2 interrupt occurs or when an Output X'40' instruction is executed in any program level other than level 1.

The ICW input register does not necessarily reflect the current state of the ICW associated with the interface address in ABAR. The actual ICW may have been modified by the Type 2 Scanner during scan addressing after the ICW input register was set.

Also, the actual ICW may have been altered by the execution of an Output X'43', X'44', X'45', X'46', or X'47' instruction. Subsequent input instructions do not include these alterations since the ICW input register had been set by a previous Output X'40' instruction or Type 2 Scanner level 2 interrupt. In the event that an output instruction and scan addressing both occur during the same scan cycle, the output instruction is executed first; then the scanner performs its modification, if needed. This ensures that the latest modifications to the actual ICW will be included during the next scan addressing operation.

Refer to *Interface Control Word Format* in this chapter for a complete description of the individual ICW fields and the I/O instructions associated with each field.

Programming Note

Since the interface control words are asynchronously interrogated and modified by both the scanners and the control program, caution should be observed to ensure the ICW integrity when the program issues an output instruction. The Type 2 Scanner cannot check whether the control program has modified the ICW correctly. Therefore, errors in the modification itself may be difficult to isolate. To prevent control program modifications to the ICW (via output instructions) from being destroyed by the scanner, program modifications are not permitted during that portion of a scan when the Type 2 Scanner fetches, modifies, and restores the ICW for the line being scanned.

Interface Control Word Format

The following paragraphs describe the ICW fields and their bit meanings. (See Figure 26.)

ICW Bits 0-7 (Secondary Control Field)

The secondary control field (SCF) is used as a sense, status, and operation modifier field between the control program and the communication scanner. Bits 0-4 are set by the Type 2 Scanner hardware according to the conditions described below. Bits 6 and 7 are program controlled. This field may be tested by using the Input X'44' instruction. An Output X'44' instruction is used to reset bits 0-3 and 5 and to set and reset bits 6 and 7. Refer to Appendix B, *Input/Output Instruction Bit Definitions*.

Bit 0 -- Stop Bit Check/Receive Break: For start-stop lines in PCF state X'7', the receive data bit buffer is checked after each character is received. If the bit buffer contains a "0" (space) instead of a "1" (mark), the Type 2 Scanner signals this condition to the control routine by setting this bit.

For start-stop lines during transmit operations (PCF state X'9'), the 'receive data' line is checked for a space (0) condition every time the first bit of a character is placed in the transmit buffer. If a space condition is detected, this bit is set on. When the control program detects that this bit is set for two consecutive characters, it should be interpreted as a *receive break* signal.

If this bit is 1, the service request interlock (ICW bit 1) will be 0.

For autocall interfaces and binary synchronous line interfaces this bit is 0.

Bit 1 -- Service Request Interlock: This bit is set when the Type 2 Scanner detects that data transfer or control servicing is required between the control program and the 'parallel data field'. The control program must reset this bit after the interrupt is honored and all bits or bytes of the ICW have been modified. If this bit is already set when the scanner is prepared to set it on, and the PCF state is X'7' through X'A', a character overrun/underrun flag is set (ICW bit 2).

If this bit is 1, the stop bit check/receive break, character overrun/underrun, and modem check bits are 0.

Programming Note

The control program should reset the service request interlock before setting the PCF state for states that monitor modem or autocall unit control lines for the purpose of generating an interrupt.

Bit 2 -- Character Overrun/Underrun: This bit is set when the Type 2 Scanner attempts to set the service request interlock (ICW bit 1) and finds it already set. This error is normally caused by an instantaneous peak overload situation. Errors of this type should not occur in the average installation and should occur only infrequently in high throughput installations.

If a character overrun occurs, the next character received is placed into the PDF field overlaying the character that was to have been serviced. Thus, if an overrun occurs, a character is lost.

In the event of an underrun, the same character is transmitted until the program changes the PDF field to another character or the primary control field is changed from the transmit state.

If this bit is 1, the service request interlock (ICW bit 1) is 0.

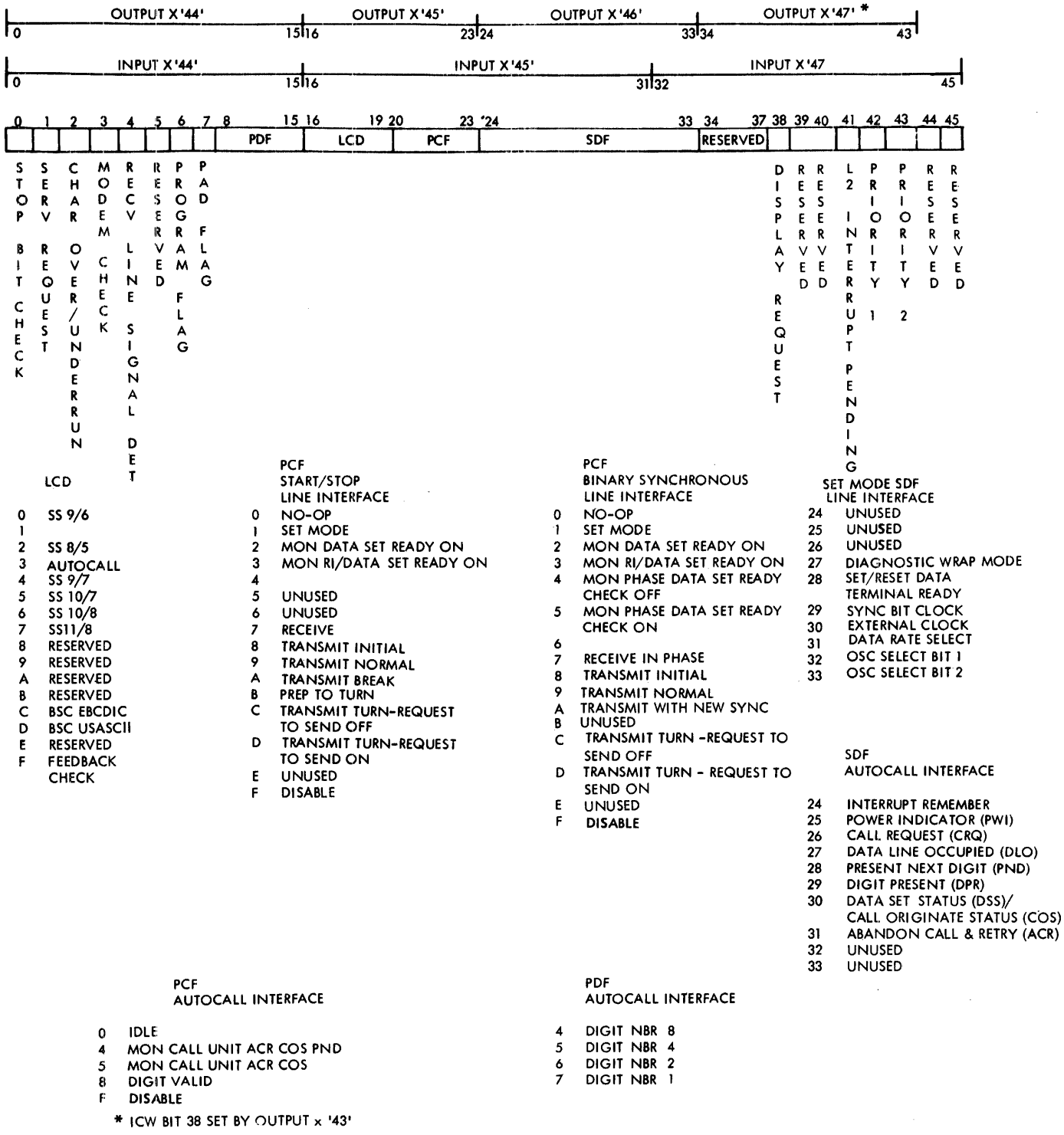


Figure 26. Interface Control Word

Bit 3 -- Modem Check: During each bit interval (bit service time), the Type 2 Scanner checks the line interface for the proper modem conditions. This bit is set to 1 to indicate the following conditions:

1. The 'data set ready' line is inactive when the PCF field of the ICW is in states X'5', X'7', X'8', X'9', X'A', X'B' or X'D'.

2. The 'clear to send' line is inactive when the PCF field of the ICW is in states X'9', X'A', X'B', or X'D'.
3. A TTY echo check has been detected.

If this bit is 1, the service request interlock (ICW bit 1) is 0.

Bit 4 -- Received Line Signal Detector: This bit is set when the line interface indicates that the data communication equipment is receiving a carrier signal that meets its requirement for receiving data. The program has no control over this bit.

Bit 5 -- This bit is reserved.

Bit 6 -- Program Flag: This bit provides a flag in the ICW that can be used by the program.

Bit 7 -- Pad Flag: For start-stop transmission, this bit is turned on by the control program when it wants the Type 2 Scanner to hold the output data line at a mark condition for one complete line-transmission character-time. This operation employs the normal transmit character serializing actions except that the start bit is sent as a mark in place of the normal space. The remainder of the character is deserialized as normal, and the control program must ensure that the PDF field is loaded with all bits on (X'FF'). Any number of *pad* characters may be sent by leaving the pad flag on and leaving the PDF field set to X'FF'. When pad transmission is to end, the control program must turn off the pad flag and resume placing normal characters in the PDF.

For autocall and binary synchronous interfaces, this bit is 0.

ICW Bits 8-15 (Parallel Data Field)
The parallel data field (PDF) is used as a character buffer. For a transmit operation, the characters to be sent to a terminal are placed in this field by the program with an Output X'44' instruction. Hardware circuits then transfer the character to the serial data field and transmit it to the interface. The format of the character loaded into the PDF depends on the state of the line control definer (LCD).

For receive operations, the character is assembled in the SDF from the line interface and then transferred to the PDF under hardware control. An Input X'44' instruction must then be executed to retrieve the character from this field for program use. The format of the character loaded into the PDF field and how it is used for various line control definer values can be found in the *LCD States* section in this chapter.

For an autocall interface, the digit number must be placed in the PDF as shown in Figure 26.

ICW Bits 16-19 (Line Control Definer)
The line control definer (LCD) field defines the type of interface associated with the ICW. The LCD, set under program control by an Output X'45' instruction, must be in agreement with the installed interface type and the common carrier or IBM equipment physically attached to the interface. The LCD field definition must also be in agreement with the basic line control procedures to be executed. An LCD is generally set at IPL time and remains static throughout normal operation.

The values of the LCD field are defined in detail along with its effect on the PDF and SDF fields in the section describing LCD states in this chapter.

The LCD field is also used to indicate that a feedback check has occurred. When the scanner sets the LCD field to X'F', it indicates an improper mode set or a hardware failure in either the Type 2 Scanner or the interface hardware.

Programming Note
During a diagnostic wrap operation, the LCD of the line or lines in diagnostic receive must agree with the LCD of the line in diagnostic transmit regardless of the common carrier or IBM equipment physically attached to the line.

ICW Bits 20-23 (Primary Control Field)
The primary control field (PCF) defines the state of the interface at any particular time. The interpretation of this field depends on the state of the LCD field. Figure 26 shows the PCF states for start-stop and BSC line control. See the section on *Autocall Interface Operation* for a description of the autocall PCF states.

The Output X'45' instruction can be used to set the state of the PCF. Also, an Input X'45' can be executed to test the state of this field. Refer to Appendix B for bit definitions of input/output instructions.

ICW Bits 24-33 (Serial Data Field)
The serial data field (SDF) is primarily used as a character deserializer/serializer field. On receive operations, the data coming in from a line is placed in this field bit-by-bit to assemble a character. When a character has been assembled, it is transferred, under hardware control, to the PDF for program access. For transmit operations, a character from the PDF is transferred into the SDF under hardware control and then transferred to the line interface hardware a bit at a time.

The SDF is also used for line interface mode setting and autocall operations. Its format depends on the values in the LCD and PCF fields. See *Interface Mode Set* and *Autocall Interface Operation* in this chapter for further definitions.

Program access to this field is through the Input X'45' and Input X'47' instructions and Output X'46' instruction. Refer to Appendix B for bit definitions of input/output instructions.

ICW Bits 34-37
These bits are reserved.

ICW Bit 38 (Display Request)
This bit allows the state of certain latches or signals to be loaded into the Type 2 Scanner display register when the interface associated with this ICW is scanned. The bit may be set and reset by an Output X'43' instruction when the interface address for that ICW is in the attachment buffer address register (ABAR) in the Type 2 Attachment Base.

The contents of the display register in a particular Type 2 Scanner can be accessed by an Input X'46' instruction when that scanner is selected.

Because only one display register is in each Type 2 Scanner, the program should ensure that the *display request* bit is never on in more than one ICW in each scanner. Only in this way can the information in the display register be meaningful.

Before executing an Input X'46', the program must also ensure that enough time has elapsed to guarantee that the interface has been scanned after setting the display request bit in the ICW associated with the interface.

ICW Bits 39-40
These bits are reserved.

ICW Bit 41 (Level 2 Interrupt Pending)
This bit is set when the interrupt priority register (IPR) assigned to this interface is already occupied by another interface. This stacks the new interrupt until the next time the line is scanned and the IPR is not occupied.

ICW Bits 42-43 (Priority Select bits 1 and 2)
These bits assign one of the four interrupt priority registers in the attachment base to the ICW for the interface. All combinations are valid with X'0' designating the lowest priority register and X'3' the highest priority register.

ICW Bits 44-45
These bits are reserved.

Line Control Definer

This section describes the various LCD (line control definer) states. The LCD field is used during normal transmit and receive operations to define the hardware line control required by the line set type. The variations in the PDF field due to different line set requirements are shown with each LCD state. The first information bit of a transmitted or received character is designated as X1, the second bit X2, and the nth and last bit Xn. For start-stop transmissions, the start and stop bits are not regarded as information bits and are inserted or deleted by the Type 2 Scanner hardware.

LCD State X'0' (Start-Stop 9/6 Bit Control)
This state should be set for start-stop transmission with a 9/6 format (that is, one start bit, six data bits, and two stop bits). When a character is sent to an interface, the six data bits must be placed into bits 2-7 of the PDF field as shown below. Characters received from the interface are in the same format when the scanner requests a character service interrupt.

PDF bit positions	0	1	2	3	4	5	6	7
Character bits	0	0	X6	X5	X4	X3	X2	X1

LCD State X'1'
(Reserved)

LCD State X'2' (Start-Stop 8/5 Bit Control)
This state is for start-stop interfaces with an 8/5 format (that is, one start bit, five data bits, and two stop bits). When a character is sent to an interface, the five data bits must be placed into bits 3-7 of the PDF field as shown below. Characters received from the interface are in the same format when the scanner requests a character service interrupt.

PDF bit positions	0	1	2	3	4	5	6	7
Character bits	0	0	0	X5	X4	X3	X2	X1

LCD State X'3' (Autocall)
This state is for autocall interfaces. See the *Autocall Interface Operation* section in this chapter.

LCD State X'4' (Start-Stop 9/7 Bit Control)
This state is for start-stop interfaces with a 9/7 format (that is, one start bit, seven data bits, and one stop bit). When a character is sent to an interface, the seven data bits are placed into bits 1-7 of the PDF field as shown below. Characters received from the interface are in the same format when the scanner requests a character service interrupt.

PDF bit positions	0	1	2	3	4	5	6	7
Character bits	0	X7	X6	X5	X4	X3	X2	X1

LCD State X'5' (Start-Stop 10/7 Bit Control)
This state is for start-stop interfaces with a 10/7 format (that is, one start bit, seven data bits, and two stop bits). When a character is sent to an interface, the seven data bits must be placed into bits 1-7 of the PDF field as shown below. Characters received from the interface are in the same format when the scanner requests a character service interrupt.

PDF bit positions	0	1	2	3	4	5	6	7
Character bits	0	X7	X6	X5	X4	X3	X2	X1

LCD State X'6' (Start-Stop 10/8 Bit Control)
This state is for start-stop interfaces with a 10/8 format (that is, one start bit, eight data bits, and one stop bit). When a character is sent to an interface, the eight data bits must be placed into bits 0-7 of the PDF field as shown below. Characters received from the interface are in the same format when the scanner requests a character service interrupt.

PDF bit positions	0	1	2	3	4	5	6	7
Character bits	X8	X7	X6	X5	X4	X3	X2	X1

LCD State X'7' (Start-Stop 11/8 Bit Control)
This state is for start-stop interfaces with an 11/8 format (that is, one start bit, eight data bits, and two stop bits). When a character is sent to an interface, the eight data bits must be placed into bits 0-7 of the PDF field as shown below. Characters received from the interface are in the same format when the scanner requests a character service interrupt.

PDF bit positions	0	1	2	3	4	5	6	7
Character bits	X8	X7	X6	X5	X4	X3	X2	X1

LCD States X'8' through X'B'
(Reserved)

LCD State X'C' (BSC EBCDIC Line Control)
This state is for binary synchronous interfaces using the EBCDIC SYN character. When a character is sent to an interface, the data bits must be placed into bits 0-7 of the PDF field as shown below. The SYN character (X'32') provides for automatic detection of the first phase character during a receive operation.

PDF bit positions	0	1	2	3	4	5	6	7
Character bits	X8	X7	X6	X5	X4	X3	X2	X1

LCD State X'D' (BSC USASCII Line cControl)
This state is for binary synchronous interfaces using the USASCII SYN character. When a character is sent to the interface, the data bits must be placed into bits 0-7 of the PDF field as shown below. The SYN character (X'16') provides for automatic detection of the first phase character during a receive operation.

PDF bit positions	0	1	2	3	4	5	6	7
Character bits	X8	X7	X6	X5	X4	X3	X2	X1

LCD State X'E'
(Reserved)

LCD State X'F' (Feedback Check)
This state is set by the Type 2 Scanner whenever a feedback check is detected (1) during scan addressing on any of the scanner 'data in' lines from the selected line interface base, or (2) when a bit service reset error is detected on the line from the selected LIB. A set mode to an interface that has been configurated incorrectly also sets this state.

Primary Control Field

This section describes the PCF states for start-stop and BSC line control. See the *Autocall Interface Operation* section in this chapter for a description of the autocall PCF states.

PCF State X'0' (No-Op)
This PCF state causes the Type 2 Scanner to take no action (active or passive) upon subsequent scans. The scanner hardware can request a Type 2 Scanner L2 interrupt and set this PCF state for an interface if it determines that new control information is required from the control program. This PCF state can be set by the control program; however, no interrupts are generated by the interface.

PCF State X'1' (Set Mode)
This PCF state causes the scanner to set and reset certain mode latches in the line interface hardware. These latches are specified by the SDF field. When setting this PCF state, the control program must ensure the integrity of the entire ICW. This may be done by first setting the PCF to state X'0' (no-op) so that the ICW will not be modified by a possible interrupt. The SDF can then be set to the proper value. Finally, state X'1' (set mode) can be set into the PCF field. Execution of a set mode does not require a bit service request from the addressed interface. However, a bit service request must occur to allow the scanner to request a Type 2 Scanner L2 interrupt to end the set

mode operation. The set mode operation ends when the scanner hardware sets the PCF state to X'0' (no-op).

A set mode can be executed to change the state of the data rate selector bit and the oscillator select bits without requiring a disable. However, data terminal ready must remain *on*.

PCF State X'2' (Monitor Data Set Ready)
This PCF state places the interface in a wait-for-incoming-call condition. For switched lines, this state should normally be set by the control program following a PCF state X'F' (disable) and PCF state X'1' (set mode with data terminal ready bit = 1). When an interface is in this state, the Type 2 Scanner tests the 'data set ready' lead from the common carrier or IBM line adapter for an active condition when the ICW is fetched. When data set ready is *on*, indicating that a call is established, the Type 2 Scanner sets PCF state X'0' (No-Op) for start-stop or PCF state X'4' (monitor phase--data set ready check off) for binary synchronous transmission and requests an L2 interrupt.

Though not necessary, this state can also be used for leased lines. Data set ready should be on at the first bit service request when the interface is scanned.

PCF State X'3' (Monitor Ring Indicator or Data Set Ready)
This PCF state, when set by the control program, places the line interface in a wait-for-incoming-call (ring indicator *on*) or wait-for-manual-call-out-connection condition (data set ready *on*). This state must be preceded by setting PCF state X'F' (disable), or a set mode that resets data terminal ready. When the PCF state is set to X'3', the Type 2 Scanner tests the 'ring indicator' and 'data set ready' leads from the common carrier equipment for an active condition of either lead. When 'ring indicator' is active a call is coming in and a pending connection is to be established. When either of these conditions occurs, the Type 2 Scanner sets PCF state X'0' (no-op) and places the line in a L2 interrupt pending state. This PCF state must be followed by PCF state X'1' (set mode) from the control program to set the 'data terminal ready' latch. After the Type 2 Scanner executes the set mode, it sets PCF state X'0' (no-op) and places the line in a L2 interrupt pending state. The interrupt handling program must then place the line in PCF state X'2' (monitor data set ready on), after which the operation proceeds as described in *PCF State X'2'--Monitor Data Set Ready*.

PCF State X'4' (Monitor Phase--Data Set Ready Check off)
This PCF state is identical to PCF state X'5' (BSC--monitor phase-data set ready check *on*) except that the inactive condition of 'data set ready' does not signal a check condition. PCF X'4' is intended to initialize the *first* receive operation after a switched network call connection has been established.

PCF State X'5' (Monitor Phase--Data Set Ready Check on)
This PCF state places a BSC line into a *hunt for phase* condition. The SDF field is shifted each bit interval time, and the contents are examined by the scanner hardware for a comparison with the bit configuration of the 8-bit SYN character. If a compare is sucessful, PCF state X'7' is set, and the tag bit is inserted in the SDF. A L2 interrupt request, however, is *not* generated at this time. The first interrupt request is at the next subsequent character time. When this interrupt request is presented, the control program must examine the character in the PDF to determine if the second SYN character has been received. If so, the PCF is left in state X'7'. If the second SYN has not been received, the program returns the PCF to state X'5'. The Type 2 Scanner also sets PCF state X'5' after completing a transmit turnaround (PCF state X'C' or X'D').

PCF State X'6'
(Reserved)

PCF State X'7' (Receive)
Start-Stop: In this PCF state, the Type 2 Scanner monitors for start bits (as described in the *Line Control Definer* section) according to the setting of LCD X'0', X'2', X'4'-X'7'. This state remains in effect until changed by the control program and is set by the Type 2 Scanner after the completion of a transmit turnaround (PCF state X'C' or X'D').

BSC: In this PCF state, the Type 2 Scanner frames consecutive 8-bit characters (as described inn this PCF state, the Type 2 Scanner frames consecutive 8-bit characters (as described in the *Line Control Definer* section) according to the setting of the LCD X'C' and X'D'. This state remains in effect until changed by the control program and is set by the Type 2 Scanner after one SYN character has been detected in PCF states X'4' or X'5' for BSC.

PCF State X'8' (Transmit Initial)
This PCF state is set by the control program. The Type 2 Scanner places the interface hardware into a transmit condition and transmitting begins when the 'clear to send' lead is activated from the carrier equipment. In addition to setting this state, the control program must perform the following sequence.

1. Set PCF state X'0' (no-op). This step may have been performed on any previous interrupt.
2. Set the SDF to the first character to be transmitted.
3. Store the second character to be transmitted into the PDF.
4. Set PCF state X'8' (transmit initial).

For start-stop initialization, the first character to be transmitted (set in the SDF) must be X'FF' and the second character (set in the PDF) must be the first information character.

For BSC initialization, when business machine clocking is used, 16 transitions of the 'received data' lead are required to ensure bit synchronization between the transmitting and receiving stations. Therefore, SDF Bits 1-9 must be set to an initial pad of X'AA', and the PDF must be set to X'1AA'. The next character to be transmitted must be a pad character of X'AA' followed by two SYN characters to enable character synchronization to be established.

BSC specification requires that the first character transmitted be an initial pad character. Therefore, when modem clocking is used, SDF bits 1-9 must be set to X'1AA'. Since the next two characters must be SYN characters, the PDF must be set to a SYN character. All characters stored into the SDF are assumed to be right-justified.

When the Type 2 Scanner begins transmitting (clear to send *on*), the hardware changes the PCF control state to X'9' (transmit data).

Note: *It may be desirable in certain applications (contention) to test the PCF state in order to determine if a transmit operation should be started. For example, a line may have just set PCF state X'7' (receiving-in-phase), and its subsequent interrupt has not been handled by the control program.*

PCF State X'9' (Transmit Data)
This PCF state is set by the Type 2 Scanner after completion of PCF state X'8' (transmit initial). Data is transmitted in this state until one of the transmit turnaround states (PCF X'B', X'C' or X'D') is set by the control program. PCF state X'A' (transmit data with new sync) should be used in place of state X'9' for synchronous modem equipment containing a 'new sync' lead.

All control and non-information characters must be supplied by the control program because the scanner does not perform character decoding, encoding, or insertion of any kind during a transmit operation. The scanner detects and signals underruns, but the control program has to take corrective action, (for example, BSC abort sequence).

For BSC transmission on a line with business machine clocking, the first two characters transmitted in the transmit data states (PCF states X'9' and X'A') must be X'AA' followed by two SYN characters. For BSC transmission on a line with modem clocking, the first character transmitted in PCF state 9 (transmit data) must be X'AA' followed by two SYN characters. Note, this may have already been done under PCF state X'8' (transmit initial).

After all information characters (EOB, EOT, ENQ, ACK, check characters, etc.) have been transmitted under the transmit data states (PCF states X'9' or X'A'), the control program must complete the transmit operation by setting one of the transmit turnaround states (PCF state X'B', X'C' or X'D').

PCF State X'A' (Transmit Break--Start-Stop)
This state is set by the control program instead of PCF state X'9' (Transmit Data) when transmitting a *break* signal to the remote destination. The following sequences of PCF states should be set by the control program for transmitting a *break* signal:

	SDF	*PDF*	*Pad Flag*
Transmit Initial (PCF State X'8')	X'FF'	X'FF'	1
Transmit Break		X'00'	0
Transmit Turnaround (PCF X'B' or X'C')		X'FF'	1

After the completion of this sequence, the *break* signal (continuous spacing) continues for n character times. The stop bits for the spaces (X'00') are inhibited from being transmitted as a mark so that the *break* signal is continuous spacing.

Note: *Two character-time delays exist before the actual break signal is transmitted.*

PCF State X'A' (Transmit Data with New Sync)
This state is identical to PCF state X'9' (transmit data) except that the 'new sync' line to the modem equipment is active. It must be used only with 4-wire duplex, multipoint leased-line modem equipment where the associated interface is designated as the master station. The control program must change PCF state X'A' to PCF state X'9' (transmit data) in the character service routine that places the last character to be transmitted into the PDF.

PCF State X'B' (Prepare to Turn for Start-Stop)
This state is set by the control program on the interrupt following the interrupt that placed the last character (pad or information) to be transmitted into the PDF. While bits are being transmitted, this state is the same as PCF state X'9' (transmit data).

When the character is completely transmitted, PCF state X'C' (transmit turnaround--request to send *off*) is set by the scanner. The SDF is set to X'00', and the line interface transmit data buffer is left in the 'mark' state. This action delays completion of the transmit operation to ensure that the stop bit remains on the interface transmit data output at least one bit time before 'request to send' can be turned off. At the next bit interval, if 'clear to send' is off, the line is placed in an interrupt pending condition as the final interrupt of the transmit operation. PCF state X'7' (Receive--Start-Stop) is set by the Type 2 Scanner, and the SDF is left at X'00'. If 'clear to send' is on, the scanner loops, and there is no change in the PCF state or no interrupt generated until 'clear to send' drops.

PCF State X'C' (Transmit Turnaround--Request To Send off)
For BSC line control (LCD=X'C', or X'D') this PCF state is set by the control program on the interrupt following the interrupt that placed the last pad character to be transmitted into the PDF. While bits are being transmitted, this state is the same as state X'9' (transmit data).

When the character is completely transmitted, 'request to send' is reset along with the 'transmit mode' latch in the interface hardware. PCF State X'C' is not changed until 'clear to send' is off. After 'clear to send' is off, PCF state X'5' (monitor phase - data set ready check *on*) is set, and the line is placed in an interrupt pending state. The control program should make sure 'clear to send' is off before the background timeout elapses. For a description of this state under start-stop line control, see PCF state X'B'.

When the control program wants to close a line that normally transmits with 'request to send' on, it must notify the scanner that 'request to send' is to be turned *off* by a PCF state X'C'. This must be done by sending a *pad message* using PCF state X'B' for start-stop, or PCF state X'C' for BSC, instead of PCF state X'D'. The pad message should result in a continuous marking condition on the line (for Start-Stop, see SCF Pad Flag). An alternative is to ensure that final outgoing transmissions use PCF state X'C'.

Programming Note
Some modems do not turn 'clear to send' off under the above conditions. Therefore, the control program should test this condition and may be required to set the PCF to X'D' and operate with 'request to send' on.

PCF State X'D' (Transmit Turnaround--Request To Send on)
This state is set by the control program on the interrupt following the interrupt that placed the last character (pad or information) to be transmitted into the PDF. While the bits are being serialized in the SDF, this state is the same as PCF state X'9' (transmit data).

When the character is fully serialized, the interface transmit control (not including 'request to send') is reset and the final interrupt request is set for the transmit operation. The PCF is set by the Type 2 Scanner to PCF state X'5' (monitor phase - data set ready check *on*) for BSC or to PCF state X'7' (receive) for start-stop.

The significance of 'request to send' on (PCF state X'D') is:

Start-Stop: 'Request to send' on is to be used with all common carrier equipment that provides duplex facilities and for IBM line adapter/modem equipment on duplex communication facilities.

BSC: 'Request to send' on is to be used on point-to-point 4-wire duplex and multipoint 4-wire duplex communication facilities where the 3705 serves as the master station. All BSC switched network communication facilities are half-duplex.

BSC and Start-Stop Local Attachments: Equivalent to 4-wire point-to-point communication facility.

PCF State X'F' (Disable)
This state is set by the control program and causes the Type 2 Scanner to turn off data terminal ready. A disable resets all control information in the line that was provided by the last set mode (PCF state X'1'). The scanner hardware then causes the interface to be placed in an interrupt pending state when the 'data set ready' lead and the 'receive line signal detector' lead are deactivated. For auto-dial applications, other conditions on the automatic calling unit must be satisfied before another dial operation can be attempted. Before the interrupt is requested, PCF state X'0' (no-op) is set by the scanner. Because all control information in the line set is reset, the control program must set up the proper control information again by a set mode (PCF state X'1') in the interrupt after the disable.

Interface Mode Set
The interface hardware latches set and reset according to the value in the SDF field during PCF state X'1'. The following paragraphs define the SDF values for a set mode.

SDF bits 0-2: ICW Bits 24-26 - Not Used

SDF bit 3: This bit is set to 1 to place the addressed interface in the diagnostic mode.

SDF bit 4: Data Terminal Ready - This bit controls the 'data terminal ready' lead which must be set to enable the line interfaces provided by Line Sets 1A, 1C, 1D, 1F, 1G, 3A, 3B, 4A, 4B, and 4C.

SDF bit 5: Synchronous Bit Clock - This bit determines whether synchronous or start-stop clocking is used for the addressed interface when business machine clocking is specified. If SDF bit 6 is 1 (external clock), this bit is ignored.

A feedback check occurs if this bit is on when the program executes a set mode for interfaces provided by Line Sets 1A, 1C, 2A, 3A, 3B, 4A, 4B, or 4C since these lines sets support only start-stop devices.

SDF bit 6: External Clock - This bit determines whether business machine or modem clocking is used for the addressed interface. A 1 = modem clock, and 0 = business machine clock.

A feedback check occurs if this bit is on when the program executes a set mode for interfaces provided by Line Sets 1A, 1C, 2A, 3A, 3B, 4A, or 4B, since these line sets allow only business machine clock control.

SDF bit 7: Data Rate Selector - This bit selects a high speed or low speed data rate for the attached modem. A 1 = high data rate, and 0 = low data rate. If modem clocking is specified, this bit selects which of the two clock speeds in the modem is to provide the clock pulses. The low rate usually equals one-half of the high rate. In this case the business machine clock selected by the oscillator select bits must not exceed one-half the clock speed selected in the modem.

A feedback check occurs if this bit is on when the program executes a set mode to a line interface provided by Line Sets 1A, 1C, 2A, 3A, 3B, 4A, 4B, or 4C.

SDF bits 8 and 9: OSC Select Bits 1 and 2 - The state of these two bits selects the business machine clock to be used by the addressed line interface. At least one business machine clock must be installed in each Type 2 Scanner. See the following section, *Business Machine Clocks*.

Programming Note
The oscillator select bits can be changed without causing a switched network connection to be broken if SDF Bit 4 (Data Terminal Ready) is set when the set mode is executed.

Business Machine Clocks

Each Type 2 Scanner must have at least one business machine clock installed and may have as many as four. If modem clocking is used with any of the lines, a business machine clock must be installed with a speed of one-half or less than that of the lowest speed modem clock. Figure 21 lists the business machine clocks available.

For line speeds greater than 2400 bps, a modem must provide the clock pulses. Line Sets 1A, 1C, 1F, 2A, 3A, 3B, 4A, 4B, and 4C must operate with business machine clocks. Line Sets 1D and 1E can operate under business machine or modem clock control. Refer to Appendix D for a description of the individual LIB and line set types.

The installed business machine clock used for a given line is selected under program control by executing a set mode (PCF state X'1') with SDF bits 8 and 9 set to indicate the desired clock. Figure 27 shows the proper setting of the oscillator select bits to assign an installed oscillator to a given interface.

SDF Bits 8 9	*Selected Business Machine Clock*
0 0	Lowest speed clock(OSC0)
0 1	Next higher speed clock(OSC1)
1 0	Next higher speed clock(OSC2)
1 1	Highest speed clock(OSC3)

Figure 27. Type 2 Communication Scanner Business Machine Clock Selection.

No business machine clock is selected if the oscillator select bits are set to select an uninstalled oscillator (for example, bits 8 and 9 set to 11 when only two or three oscillators are installed).

Every interface must have a business machine clock assigned whether it is specified to be business machine or modem clocked. For autocall interfaces and for line interfaces that are to use modem clocking, the assigned business machine clock is used to ensure that the interface is periodically accessed. The lowest speed oscillator is always used for an autocall interface.

The oscillator select bits are set to 0 by a reset to the scanner. Therefore, the lowest speed clock is initially selected, and unless a set mode is executed to select another clock for a given interface, the lowest speed clock is used.

After a power-on-reset occurs, there is a warm-up period associated with the different clocks. (Refer to Figure 21.) During this warm-up period, a business machine clock cannot provide service requests.

Programming Notes

1. The oscillator select bits for a line interface can be changed without causing a switched network connection to be broken, if 'data terminal ready' is up when the set mode is executed.
2. The business machine clock selected for a modem-clocked line interface must be one-half or less than of the rate of the modem clock.

I/O Programming Considerations

As a general rule, input/output instructions should be issued only when the status of the attachment buffer address register (ABAR) and the particular Type 2 Scanner ICW input register is known. An understanding of how those registers are set or loaded is needed for correct execution.

The ABAR is set under the following conditions:

1. The interface address in the highest priority program level 2 interrupt register located in the attachment base is loaded into ABAR just before program level 2 becomes the current program level.

   Therefore, if an Input X'40' is executed as the first instruction in program level 2, the register specified by the R operand contains the interface address for that interrupt.
2. When the program executes an Output X'40', the interface address in the register specified by the R operand is placed in ABAR.

The ICW input register of the selected Type 2 Scanner is loaded with the contents of the ICW associated with the interface address in the ABAR when:

1. The ABAR is loaded after a program level 2 interrupt occurs.
2. The Output X'40' instruction is executed in program level 3 or 4. This enables the level 3 or 4 routines to access any portion of the selected ICW associated with the interface address in the ABAR.

Figure 28 summarizes which program levels can set the ABAR in the attachment base and set the ICW input register in the selected scanner.

Program Level	ABAR	ICW Input Register
1	Output X'40'	Cannot be set
2	L2 Interrupts	L2 Interrupt
3 or 4	Output X'40'	Output X'40'

Figure 28. Setting ABAR and ICW Input Register

The following considerations are recommended for executing input/output instructions in the different program levels.

Program Level 1 - (Error Routines)

Input X'40' can be executed to obtain the methods of determining initial selection and data transfer addresses. attachment buffer address register (ABAR) in the Type 2 Attachment Base. This old interface address should be saved if a different address is required to select the Type 2 Scanner that has its L1 interrupt request set.

Output X'40' can be executed to select the appropriate Type 2 Scanner if needed. Only the selected Type 2 Scanner can decode the input/output instructions. However, the scanner input register is not changed if an Output X'40' is executed at program level 1.

After the Type 2 Scanner is selected, other input and output instructions may be executed as needed. Output instructions may be executed in any order, but all output instructions (Outputs X'43', X'44', X'45', X'46', and X'47') that set a portion of the ICW *must* be separated by at least one cycle. This is required because the output register in the Type 2 Scanner buffers the data from the general register and requires time to store the data in the ICW.

Before exiting from program level 1, the program may execute an Output X'40' to place the old interface address back in ABAR if it had been saved. However, one instruction cycle must separate it from any Output X'43'-X'47'. The selected Type 2 Scanner ICW input register is not changed as a result of Output X'40'.

At least one instruction cycle must separate the last output instruction and an Exit instruction.

Program Level 2 - (Character Service)
Input X'40' may be executed immediately to obtain the interface address. When Input X'40' is issued while in program level 2, the 'priority register occupied' latch associated with the interface address in ABAR is reset. This indicates that the character service request is being serviced and that the program level 2 interrupt priority register from which the ABAR was loaded is now available for another level 2 interrupt of the same priority. If all the 'priority register occupied' latches are reset, the Type 2 Scanner L2 interrupt request is reset.

Inputs X'44', X'45', or X'47' may be executed whenever necessary to obtain a portion of the ICW from the Type 2 Scanner ICW input register. Outputs X'43', X'44', X'45', X'46', or X'47' may be executed whenever necessary to set a portion of the ICW.

Output instructions may be executed in any order, but all subsequent Output X'43', X'44', X'46', or X'47' instructions must be separated by at least one cycle. These outputs must also be separated from an Output X'40' by at least one instruction.

At least one instruction cycle must separate the last output instruction and an Exit instruction.

Programming Note
Subsequent Input X'40' instructions within the same character service interrupt do not reset the 'priority register occupied' latches.

Program Levels 3 and 4 - (Lower Level Routines)
Output X'7E' may be executed with a 1 in byte 1 bit 2 of the register specified by the R operand. This will 'mask off' program level 2 interrupts that could change the contents of the attachment buffer address register (ABAR) in the Type 2 Attachment Base by a character service L2 interrupt.

Output X'40' may be executed to load ABAR with the interface address of a line to be acted upon. The contents of the ICW associated with this interface address are placed in that Type 2 Scanner ICW input register.

After the Type 2 Scanner is selected, (1) Output X'43', X'44', X'45', X'46', or X'47' may be executed (to alter the associated portion of the ICW), followed by some other instruction, or (2) some other instruction must be executed, followed by Input X'44', X'45', X'46', or X'47' (to obtain the associated portion of the ICW that was loaded by the Output X'40' into the ICW input register).

If Output X'43', X'44', X'45', X'46', or X'47' was executed as in (1) above, the ICW content was altered, but the ICW input register still contains the contents of the ICW as it was before the alteration.

Output instructions may be executed in any order, but all subsequent Output X'43', X'44', X'45', X'46', or X'47' instructions must be separated by at least one cycle.

It is recommended that all lines in the addressed Type 2 Scanner be disabled before executing an Output X'42' to change the upper scan limit.

Output X'7F' may be executed with a 1 in byte 1, bit 2 of the register specified by the R operand. This unmasks the program level 2 interrupts. This output instruction must be separated by at least one instruction cycle from the last Output X'43', X'44', X'45', X'46', or X'47' instruction.

Autocall Interface Operation

The Type 2 Communication Scanner supports operation of an autocall interface when the interface is attached to Line Set 1E and the line control definer (LCD) field of the associated ICW is set to X'3'. The primary control field (PCF) of the ICW is used to control the interface operation. Whenever an autocall interface that has a service request is scanned, the scanner interprets the PCF to determine what communication should occur between the scanner and the interface hardware. The scanner also determines, from the PCF, whether the interface should be placed in a level 2 interrupt pending state. The lowest speed business machine clock installed in each scanner is used to generate service requests for all autocall interfaces installed in that scanner. Service requests are generated at the same rate as the clock speed.

Interface Control Word for Autocall Operation
The following bits/fields are used for autocall operation.

ICW bit 1	Service Request
ICW bits 12-15	PDF bits 4-7
ICW bits 16-19	LCD
ICW bits 20-23	PCF
ICW bits 24-31	SDF bits 0-7
ICW bit 38	Display Request
ICW bit 41	L2 Interrupt Pending
ICW bits 42-43	Service Priority

Programming Note
A reset to the scanner sets the PCF state of each ICW to X'0' and resets ICW bits 34-38 and 41 when the interface is scanned. However, when power is turned on in the controller, all other ICW bits are unpredictable. Therefore the program must ensure that during initialization ICW bits 0-3 are reset.

Primary Control Field for Autocall Operation

Five primary control field (PCF) states are available for autocall operation. Setting the PCF to an undefined value may result in improper operation. Each time the control program changes the PCF state, it should also set the LCD to X'3'. Figure 26 shows the PCF states for this LCD value. The following PCF states are valid for autocall interfaces.

PCF State X'0': Idle - This state resets the call request and digit present indications in the autocall interface each time the interface is scanned and a bit service is present. If the control program sets this state, no interrupt requests result from that interface until the program changes the PCF to one of the other valid states. If the scanner sets this state as a result of ending a PCF state X'F' (disable), the interface is placed in a level 2 interrupt pending state.

PCF State X'4': Monitor Call ACR, COS, PND - When the Type 2 Scanner fetches an ICW for an autocall interface in this PCF state, the autocall interface is monitored for the active state of the following leads:

ACR--Abandon Call and Retry
COS--Call Originate Status
PND--Present Next Digit

When any of these leads are found to be active, the appropriate SDF bit is set, and the interface is placed in a level 2 interrupt request pending state.

PCF State X'5': Monitor Call ACR, COS - This state is the same as PCF state X'4' except the active condition of present next digit (PND) does not generate a level 2 interrupt request.

This PCF state can only be set by the control program.

Programming Note

The control program must ensure that the interrupt remember bit (SDF bit 0) is reset when it places the interface in this state. Otherwise no interrupt request can be generated because of active control leads.

PCF State X'8': Digit Valid - This state is set by the control routine after it has placed the next dial digit into the PDF. This digit is continuously presented to the ACU interface until, when PND falls, the scanner sets the PCF to X'4'.

PCF State X'F': Disable - This state is used to reset the dial interface at the end of the data transfer operation. After all the control leads from the Autocall unit (ACU) have been reset, the Type 2 Scanner sets the PCF to X'0' and places the interface in a level 2 interrupt request pending state.

Serial Data Field for Autocall Operation

The control program can monitor the autocall interface by interrogating the serial data field in the interface control word. Figure 26 shows the SDF for ACU interfaces. The serial data field is updated with the current status of the autocall interface each time the interface is scanned and a bit service request is present. SDF bits 1-9 relect the state of certain autocall control signals and have no effect on the interface operation. Bit 0 is the only SDF bit that affects operation. The following paragraphs describe each SDF bit and its meaning.

SDF Bit 0: Interrupt Remember (IR) - This bit is set by the scanner to indicate a level 2 interrupt pending state. In PCF state X'4' or X'5', the scanner monitors the respective autocall interface leads for an active condition. When one of the monitored leads becomes active, the scanner sets the interrupt remember bit to prevent further interrupts from that interface until the first interrupt has been serviced. This bit must be reset by the control program each time an interrupt is serviced in order to allow the next interrupt to be recognized.

Programming Note

The program should not reset the interrupt remember bit before changing the PCF state from X'4' or X'5' to some other state because an unexpected interrupt request may result.

SDF Bit 1: Power Indicator (PWI) - When this bit is 0, the automatic calling equipment is inoperative because of the lack of power.

SDF Bit 2: Call Request (CRQ) - A 1 in this position indicates a request to originate a call to the autocall interface. The scanner sets the CRQ in the autocall interface whenever (1) the interface is scanned, (2) a bit service request is present, and (3) the PCF State is X'4', X'5', or X'8'. If the condition of this bit does not agree with the state defined for the active PCF state, the LCD is set to X'F' to indicate a feedback check.

SDF Bit 3: Data Line Occupied (CLO) - A 1 in this position indicates that the autocall interface is in use. The program should not attempt to originate a call until this lead becomes inactive.

SDF Bit 4: Present Next Digit (PND) - A 1 in this position indicates that the autocall unit is ready to accept the next digit. The PND lead is used by the autocall unit to control the presentation of digits to the unit during a dialing operation.

When the 'present next digit' lead is active and the PCF state is X'4', the scanner sets the interrupt remember bit if it is not already on. If the PND lead is inactive and the PCF state is X'8', the scanner changes the PCF state to X'4'.

SDF Bit 5: Digit Present (DPR) - A 1 in this position indicates that a valid digit is present on the digit leads to the autocall unit. The scanner sets DPR via PCF state X'8' after present next digit comes on and the next dial digit has been placed into the parallel data field, bits 4-7. When the autocall unit turns PND off, the scanner changes the PCF state to X'4' and resets DPR. The scanner resets DPR when the PCF state is changed to X'0', X'4', X'5', or X'F'. If the condition of this bit does not agree with the state defined for the active PCF state, The LCD field is set to X'F' to indicate a feedback check.

SDF Bit 6: Data Set Status (DSS)/Call Originate Status (COS) - A 1 in this position indicates that a connection has been established and that the modem is in data mode and can be used for data communications. In PCF state X'4', the scanner sets the interrupt remember bit when the DSS/COS lead becomes active.

SDF Bit 7: Abandon Call and Retry (ACR) - A 1 in this position indicates that a pre-set time interval in the autocall unit has elapsed since the last change of the present next digit lead. This bit only alerts the control program to the timeout condition; it does not automatically abandon the call and retry. The control program is responsible for abandoning the call and retrying. In PCF state X'4', the scanner sets the interrupt remember bit when the ACR lead becomes active.

SDF Bits 8 and 9: These bits are not used.

Parallel Data Field for Autocall Operation
For autocall operation the parallel data field is used to present the dial digits to the automatic calling unit. When the autocall interface is in PCF state X'8', the parallel data field bits 4-7 must contain a valid digit. In any PCF state other than X'8', the PDF is treated as if it contained all zeros. The valid digits that can be loaded into the PDF are from X'0' to X'A'. X'0' to X'9' represent the value of the dial digit and X'A' is an *end of number* bit configuration used to inform the autocall unit that the last digit of the called number has been provided.

Input/Output Instructions

The Type 2 Scanner input/output instructions enable the program to communicate between line interface bases (LIBs), program interrupt levels, interface control words (ICWs), and Type 2 Scanner registers.

Some of the major functions of the I/O instructions are:

- Determines the interface address that caused a program level 2 interrupt
- Determines the cause of a program level 1 interrupt once the scanner causing the interrupt has been identified (Input X'76' has been executed)
- Determines the status of a particular ICW
- Determines the status of a Type 2 Scanner display register
- Sets the attachment buffer address register (ABAR) with the interface address required for:
  a) addressing a particular ICW in program levels 3 or 4.
  b) restoring an old ABAR address that had been saved while in the error routines of a program level 1 interrupt. This allows the program to resume normal operation with the same interface address in the ABAR as when the program level 1 interrupt occurred.
  c) addressing a particular scanner in program level 1, 3, or 4.
- Sets and resets bits in a particular ICW
- Sets scan limits in a particular scanner while setting the substitution control register (SCR) in the Type 2 Attachment Base
- Sets control bits in a particular scanner

Programming Note
Input/Output instructions are privileged instructions executable only at program level 1, 2, 3 or 4. Any attempt to execute these instructions improperly causes a program level 1 interrupt request by setting the 'input/output check L1' latch. Refer to *Input/Output Check* description in Chapter 5 for the conditions that cause the check to be set.

Input Instructions

Six input instructions allow the program to obtain the status of the ICW input register, display register, and error register in the Type 2 Scanner, and the interface address in the attachment base ABAR. (Appendix B defines the bits within each input instruction.)

Programming Notes
1. With Extended Addressing, byte X of all input instructions and external registers is set to zero.
2. When an autocall interface is used, some of the input instructions have different bit definitions. Refer to the individual instruction descriptions for these differences.

Input X'40' (Interface Address)
This instruction is used to obtain the line interface address from the ABAR in the attachment base. Con-

ditions that set the ABAR are described in the *I/O Programming Considerations* section in this chapter.

If Input X'40' is issued during program level 2, the 'priority register occupied' latch associated with the interface address in ABAR, is reset. This indicates that the character service request is being serviced and that the program level 2 interrupt priority register from which the ABAR was loaded is now available for another level 2 interrupt of the same priority. If all the 'priority register occupied' latches are reset, the Type 2 Scanner L2 interrupt request is reset.

Programming Note
Subsequent Input X'40' instructions within the same character service interrupt do not reset the 'priority register occupied' latches.

Input X'43' (Check Register)
This instruction may be used to obtain the status of the check register of a scanner. Since there is a possibility of four Type 2 scanners being installed in the controller, the check register selected is determined by the interface address in the ABAR at time of instruction execution.

Programming Note
If any of the check register bits in the Type 2 Scanner are set to 1, the Type 2 Scanner L1 interrupt request is set.

Input X'44' (ICW Input Register - Bits 0-15)
This instruction may be used to determine the state of the secondary control field (SCF) and the parallel data field (PDF) in the ICW that is set in the ICW input register. The interface address in the ABAR selects the proper scanner. Refer to *I/O Programming Considerations* for conditions that set the ICW input register. The SCF and PDF fields and bit definitions are described in the *Interface Control Word Format* section of this chapter.

Input X'45' (ICW Input Register - Bits 16-31)
This instruction may be used to determine the state of the LCD and PCF fields and SDF bits 0-7 in the ICW that is set in the ICW input register. The interface address in the ABAR selects the proper scanner. Refer to *I/O Programming Considerations* in this chapter for conditions that set the ICW input register. The LCD, PCF, and SDF fields and bit definitions are described in the *Interface Control Word Format* section of this chapter.

Input X'46' (Display Register)
This instruction may be used to determine the state of the display register in the Type 2 Scanner selected by the interface address in ABAR.

The program, under control of the *display request* (ICW bit 38), can cause status information for a particular interface to be placed into the Type 2 Scanner display register when the interface is scanned. Input X'46' can then be used to examine this status information.

Input X'47' (ICW Input Register - Bits 32-45)
This instruction may be used to determine the state of SDF bits 8-9, display request bit, L2 interrupt pending bit, and priority bits 1-2. The interface address in the ABAR selects the proper scanner and associated ICW. See *I/O Programming Considerations* in this chapter for conditions that set the ICW input register. For an interpretation of these bits, see the *Interface Control Word Format* section in this chapter.

Output Instructions

Seven output instructions allow the program to set the status of the ICW and to set the upper scan limit and certain other controls in the Type 2 Scanner. The interface address in the ABAR and the substitution control register in the Type 2 Attachment Base may also be set. (Appendix B defines the bits within each output instruction.)

Programming Note
With Extended Addressing, byte X of all output instructions and external registers has no significance and can be ignored.

Output X'40' (Interface Address)
This instruction may be used to set an interface address in the attachment buffer address register (ABAR) of the Type 2 Attachment Base. When this instruction is executed, byte 0, bit 6 through byte 1, bit 6, in the register specified by the R operand, are placed in the ABAR.

The interface address placed in ABAR selects the Type 2 Scanner and the ICW associated with that address. Each CCU clock time, the 46 bits of the ICW are placed in the ICW work register. If Output X'40' is executed in program level 3 or 4 the contents of the ICW work register are placed in the ICW input register where it is available for Inputs X'44', X'45', and X'47'.

Output X'41' (Scan Substitution Control)
This instruction must be used to set the substitution control register in the Type 2 Attachment Base. See *Address Substitution* in this chapter for the description and coding of the SCR bits.

Output X'42' (Upper Scan Limit Control)
This instruction must be used to set the upper scan limit in the selected Type 2 Scanner. At least one Output X'42' must be executed for each Type 2 Scanner available. The Scanner selected is determined by the interface address in the attachment buffer address register (ABAR) of the attachment base at the time of execution.

Output X'43' (Control)
This instruction may be executed to set or reset various control functions in a Type 2 Scanner. The Type 2 Scanner is selected by the interface address in the attachment buffer address register (ABAR) of the attachment base.

Output X'44' (ICW Bits 0-3 and 5-15)
This instruction may be used to reset the following secondary control field (SCF) bits in the ICW: stop bit check/receive break, service request interlock, character overrun/underrun, and modem error. It is also used to set or reset the program flag, pad flag, and parallel data field (PDF) in the ICW. The PDF field is used as a character buffer. The interface address in the attachment buffer address register (ABAR), located in the Type 2 Attachment Base, selects the Type 2 Scanner and the ICW associated with this address. Refer to the secondary control field of the ICW for an interpretation of the SCF bits (byte 0, bits 0-7). See *Interface Control Word Format* for the PDF format as it relates to various line control definer states.

Output X'45' (ICW Bits 16-23)
This instruction may be used to set the bits of the line control definer (LCD) and the primary control field (PCF) in the ICW. The interface address in the ABAR at execution time selects the proper scanner and the associated ICW. For a detailed interpretation of these bits, see the *Interface Control Word Format* section in this chapter.

Output X'46' (ICW Bits 24-33)
This instruction may be used to set the bits of the serial data field (SDF) in the ICW. The interface address in the ABAR at execution time selects the proper scanner and the associated ICW. For a detailed interpretation of these bits, see the *Interface Control Word Format* section in this chapter.

Output X'47' (ICW Bits 34-43)
This instruction is used to set the state of ICW bits 34-43. The interface address in the ABAR at execution time selects the proper scanner and the associated ICW. For a detailed interpretation of the *Interface Control Word Format* section in this chapter.

Diagnostic Functions

The Type 2 Communication Scanner has two diagnostic functions available to the control program: (1) the diagnostic wrap and (2) the IBM modem self-test. These two tests are run under the control of the scanner program and provide online testing as described in the following sections. These diagnostics cannot be issued to an autocall interface.

For line interfaces attached through any one scanner, either a diagnostic wrap operation or a modem self test can be performed, but not simultaneously. However, these tests can be performed on two different scanners simultaneously.

Diagnostic Wrap
The Type 2 Scanner diagnostic wrap provides a means of testing and locating defects in the line control logic and line-interface transmit and receive logic. It also provides a method of online program testing. Diagnostic wrap can be performed online without affecting the normal program operation or the lines not in diagnostic mode. The test requires one line interface to act as a transmit line and one or more line interfaces in the same scanner to act as receive lines. Any line in the Type 2 Scanner can be a transmit or a receive line; however, there may be only one diagnostic transmit line per scanner at any one time.

The diagnostic wrap is initiated under program control by executing Output X'45' and Output X'46' instructions to all lines to be tested. The Output X'45' instruction is executed with byte 1,bits 0-3 set for proper line control and byte 1, bit 7 set to 1 to indicate PCF state X'1' (set mode). The remaining bits of this input are set to 0. See PCF state X'1' in this chapter for further information on set mode.

Output X'46' must be set as follows:

Byte 0, bits 0-7: These bits are 0.

Byte 1, bit 0: This bit is 0.

Byte 1, bit 1: Diagnostic Mode (ICW bit 27) - This bit must be set to 1.

Byte 1, bit 2: Data Terminal Ready (ICW bit 28) - This bit must be set to 0.

Byte 1, bit 3: Synchronous Clock (ICW bit 29) - This bit must be set according to the type of Line Set tested. A 1 is placed in this position for binary synchronous lines, and a 0 for start-stop lines.

Byte 1, bit 4: External Clock (ICW bit 30) - This bit must be set to 0 to select a business machine clock.

Byte 1, bit 5: Data Rate Select (ICW bit 31) - This bit may be either 1 or 0.

Byte 1, bits 6-7: Oscillator Select 1 & 2 (ICW bits 32-33) - These bits are set to select an available line oscillator (business machine clock). For proper setting, see *Business Machine Clocks* in this chapter. All wrap test lines must select the same oscillator.

After the set modes are issued, the affected line interfaces can be exercised through any sequence of point-to-point or multipoint operations.

During diagnostic wrap operations both the 'data set ready' lead and the 'clear to send' lead are simulated active to prevent the Type 2 Scanner from setting a modem check (ICW bit 3). The 'clear to send' lead is not simulated active if the PCF is set to X'C' and the Type 2 Scanner detects that it has completely serialized the character in the SDF. This is to allow either the PCF state X'B' or X'C' to be used during a diagnostic wrap. The 'receive line signal detect' lead (ICW bit 4) is also simulated active to indicate a carrier signal.

Programming Note

1. Only one interface per Type 2 Scanner may be in a transmit state at any given time during the operation.
2. The line used for transmit should be the last line to be issued the mode set.
3. Diagnostic wrap cannot be executed on an autocall interface.
4. During a diagnostic wrap operation, the line control definer (LCD) of the line or lines in diagnostic receive must agree with the LCD of the line in diagnostic transmit regardless of the common-carrier or IBM equipment physically attached to the line.

Modem Self-Test

Modem self-test tests the scrambler circuits of certain IBM modems under program control. The modem test can be performed online without affecting the normal operation of other lines. This test may also be performed simultaneously on any or all interfaces for which the test function is defined.

The modem self-test operation can be performed for IBM 3872 and 3875 modems externally attached to line interfaces provided by Line Set 1D.

To execute the modem self-test, the LCD field of the ICW for the interface to be tested must be X'C' or X'D'. The control program must then execute an Output X'46' instruction to set the interface control word SDF as follows:

SDF bits 0-2: These bits are not used and should be 0.

SDF bit 3: This bit must be set to 1 to cause the 'diagnostic mode' latch to be set in the line interface hardware. This causes the modem self-test lead to be activated in the modem interface and is used to condition the modem for a test operation.

SDF bit 4: This bit must be set to 1 cause the 'data terminal ready' When this is set, together with the diagnostic mode, it enables a modem self-test operation to be performed instead of a diagnostic wrap.

SDF bit 5: This bit must be set to 1 to provide synchronous clocking.

SDF bit 6: This bit must be set to 1 since IBM modems provide the clock pulses.

SDF bit 7: This bit may be either 0 or 1.

SDF bits 8 and 9: These bits must select an internal oscillator whose speed is one-half or less than the clock speed provided by the IBM modem. For the proper setting, see *Business Machine Clocks* in this chapter.

After the set mode is executed, the interface should be put in PCF state X'8' (transmit initial) to cause 'request to send' to be raised. Before this is done, the PDF and SDF fields should be set to X'FF' and the pad flag (ICW bit 7) turned on to cause continuous marks to be transmitted. When the first interrupt occurs in PCF state X'9', the PCF should be changed to X'D' to cause the interface to turn around with 'request to send' on. When the turnaround is complete, the interface is in PCF state X'5', and a mark is sent.

The program can now test the operation of the modem scrambler circuits by setting the PCF to X'7' (receiving in-phase) and checking the received data for all marks for a period of at least one second.

Chapter 8: Type 1 Channel Adapter

This chapter gives the reader a basic understanding of the operation of the Type 1 Channel Adapter and the requirements necessary to program the adapter.

With the Type 1 Channel Adapter (Type 1 CA), the communications controller establishes and maintains communications with the byte-multiplexer channel of an IBM System/360 or System/370. With the proper programming support, the Type 1 CA allows the controller to operate in a native 3705 mode and/or in an emulation mode, emulating the 2701, 2702, or 2703 transmission control units.

The Type 1 CA relies heavily on the program to perform control operations. Data transfers across the channel interface occur in multibyte (up to 4) bursts with program intervention required before and after each burst. Channel I/O command decoding and interpretation, data transfers between the channel adapter and storage, ending status generation, and various other functions must generally be performed by the program.

Operation and Data Flow

The Type 1 Channel Adapter receives an address from the host processor and determines whether the host wants to communicate with the controller in native mode or emulation mode. The CA then requests an initial level 3 interrupt to make this information available to the control program via input instructions. The mode of operation is then set (native or emulation) and initial selection completed.

Data from the host channel interface is received into the data buffers where the control program must retrieve it by executing input instructions and placing the bytes in storage. Channel End status is normally sent to the host channel at the end of each data transfer sequence. However, Device End is generated by the control program when the complete message or block of data is received.

When data transfer is from the channel adapter to the host channel, the control program must load the data buffers with the bytes to be transferred and then request an attention interrupt from the host processor. The host processor must initiate an initial selection sequence so that the data from the data buffers can be transferred across the channel interface.

Type 1 CA Modes of Operation

The Type 1 CA has two modes of operation--Native Subchannel Mode (NSC) or Emulation Subchannel Mode (ESC). With the proper programming support, the Type 1 Channel Adapter allows the controller to operate in NSC and ESC modes concurrently or separately.

Native mode permits servicing any number of terminals up to 352, using only one host subchannel address. The terminal address decoding is handled by the control program. Initial Program Load must always be handled in NSC.

Emulation mode allows the controller to emulate the 2701, 2702, and 2703 transmission control units using existing host programs and subchannel addresses.

Note: *Most operations of the Type 1 CA are identical whether in native or emulation mode. Throughout this chapter, the exceptions and/or differences in operation due to native or emulation mode are noted by an **NSC** or **ESC** heading on the paragraph(s) that describe the particular operation. All text that is not designated as either NSC or ESC can be assumed to apply equally to both modes of operation.*

Channel Adapter States

The Type 1 CA operates in one of three active states under direction of the control program: (1) initial selection state, (2) data transfer state, or (3) final status state. When not active, the CA is in a ready mode; that is, the adapter can accept instructions but is not in one of the active states.

Initial Selection State

The initial selection state is entered when the host processor begins an initial selection sequence. The Type 1 CA continually monitors the channel interface for one of its assigned addresses. If one of these addresses is detected, the CA traps this address and proceeds with the initial selection. If the command is received without error (correct parity), an initial status of all zeros is returned unless the command is a Test I/O or a No-Op. (See *Type 1 CA Status Presentation* in this chapter for Test I/O and No-Op exceptions.) If the command has bad parity, the CA returns a Unit Check to the host channel.

When the command from the host is a No-Op, the CA returns Channel End, Device End, and performs no further action.

During initial selection, the I/O device address and the command are stored in their respective channel adapter registers (see below). A level 3 interrupt is then requested by the initial selection hardware, and control is passed to the level 3 interrupt program.

Data Transfer State
The data transfer state is entered when the control program initiates a data transfer sequence. Data transfer can be from the host processor to the controller or from the controller to the host processor. The data is transferred across the channel under hardware control. When the data transfer ends, the channel adapter hardware signals the control program with a level 3 interrupt request.

Status Transfer State
The status transfer state is entered when the control program initiates a status transfer sequence. The status information byte is transferred to the host during this sequence.

Type 1 CA Registers

Various hardware registers are available to the control program for interface between the CA hardware and the levels 1 and 3 program routines. The following sections briefly describe these registers and the method of program access. See Appendix B for the bit definitions of the I/O instructions and the registers.

Initial Selection Control Register (ISCR)
This one-byte register contains information identifying the event that caused the Type 1 CA Initial L3 interrupt request. The register is available to the control program with an Input X'60' instruction.

Data/Status Control Register (DSCR)
This two-byte register contains control information used when passing data and status bytes across the channel interface. The bits in this register are set by the control program (Output X'62') and the CA hardware as a result of either a channel data or status transfer. This register is available to the control program with an Input X'62' instruction.

Initial Selection Address and Command Register (ISACR)
This two-byte register contains the I/O device address and the command byte presented to the channel adapter during initial selection. This register is available to the control program with an Input X'61' instruction.

Initial Selection Status Register
This one-byte register contains the status byte generated and presented by the channel adapter hardware during initial selection sequences except under the following conditions.

NSC: The NSC status byte from the NSC status register is presented instead of the hardware generated status when (1) an initial selection sequence occurs for the native mode subchannel and (2) the NSC status byte provided by the control program has not been accepted by the host (as in the case of stacked status).

ESC: The ESC status byte provided by the control program is presented instead of the hardware generated status when (1) an initial selection sequence occurs for an emulation address (Test I/O initial selection and addresses compare) and (2) the control program has signaled that both an ESC status transfer sequence is required and that ESC Test I/O status is available.

NSC Status Register
This one-byte register contains the current status of the NSC and is gated over the channel interface during NSC status transfer sequences. The control program should set the NSC status by executing an Output X'66' instruction. This register can be accessed by executing an Input X'66' instruction.

Error/Control/Condition Register
This one-byte register contains detected hardware errors and asynchronous hardware control information. The register can be accessed by an Input X'67' instruction.

Local Store Registers
These two-byte registers provide buffering for the I/O device address used in all data and status transfer sequences initiated by the controller, and up to four bytes of data for inbound/outbound data transfer sequences. When in the emulation mode, the ESC final status is included along with the transfer address. These registers are as follows.

Transfer Address/ESC Status Register: This register can be accessed by executing an Input X'63' and loaded by executing an Output X'63'. The transfer address portion (byte 0) contains the I/O device address to be presented to the multiplexer channel when the controller initiates any data/status transfer sequence. The ESC status byte portion (byte 1) contains the status byte to be presented to the multiplexer channel on completion of an ESC command.

Data Buffer Byte 1/Data Buffer Byte 2 Register: This two-byte register contains the first and second data bytes transferred across the channel interface during an inbound or outbound data transfer sequence. The register can be loaded by executing an Output X'64' instruction or accessed by executing an Input X'64' instruction.

Data Buffer Byte 3/Data Buffer Byte 4 Register: This two-byte register contains the third and fourth data bytes transferred across the channel interface during an inbound or outbound data transfer sequence. The register can be loaded by executing an Output X'65' instruction or accessed by executing an Input X'65' instruction.

I/O Device Addresses

The channel adapter must determine whether to recognize the I/O device address presented when initial selection occurs. It must also determine which I/O device address to present when the control program signals the channel adapter to perform a data/status sequence. The following sections describe the methods of determining initial selection and data transfer addresses.

Initial Selection Address

The address byte presented during initial selection must have correct parity, or the channel adapter will not decode the address. The initial selection addresses that the channel adapter recognizes are determined by plug options wired by the customer engineer from information supplied by the user.

The address assigned for the native mode subchannel can be within the range of addresses assigned for ESC but cannot also be used for emulation. The assigned NSC address can be checked by executing an Input X'67'.

When power is turned on, the channel adapter is placed in such a state that it will not recognize any ESC addresses until after the interface is enabled and the control program executes an Output X'67' with byte 1, bit 5 = 1 (Set ESC Operational).

When a Type 1 CA Initial L3 interrupt request is caused by an initial selection, the program can determine the I/O device address by executing an Input X'61'. This interrupt request can be caused by an initial selection sequence for any operational ESC address or for the native mode subchannel address. The program should be prepared to handle initial selection sequences for all operational addresses.

NSC: The NSC address can be assigned any value in the range of 0-255. If the Two Channel Switch Feature is installed, the NSC addresses for interfaces A and B are assigned separately and can be either two different addresses or the same address used for both interfaces.

Because the native mode uses only one subchannel address, the terminal address must be transferred from the host as data. The location and format of these addresses must be coordinated between the host program and the control program in the controller.

ESC: The ESC addresses must be a group of contiguous addresses. The lowest address (L) in the group can be set to 0 or any even multiple of 16 from 16 to 240. The highest address (H) in the group can be set to any value greater than (L) that is one less than an even multiple of 4, from 3 to 255. The range of recognizable ESC addresses can be set to include a minimum of 4 or a maximum of 256 addresses. This range must be the same for both interface A and interface B if the Two Channel Switch feature is installed.

Note: *If emulation is not required, the machine can be wired (by the customer engineer) so that no ESC addresses are recognized.*

Transfer Sequence Addresses

When the control unit initiates a data/status transfer sequence, it must present the I/O device address associated with the transfer. Since this address is variable, it must be provided by the control program. This is done by executing an Output X'63'.

Since Output X'63' allows the control program to provide any I/O device address in the range of 0-255, and the channel adapter cannot determine if this is the address that should be presented, the control program must ensure that the address is correct. An incorrect address results in improper channel operation.

Type 1 CA Interrupt Requests

The Type 1 Channel Adapter can initiate interrupt requests for either level 1 or level 3 service. Level 1 requests are caused when the CA detects an error or check condition. Two different level 3 interrupt requests can be set for the purpose of controlling channel adapter operations: the Type 1 CA Initial L3 and the Type 1 CA Data/Status L3 interrupt request.

Level 1 Interrupt Request

When any of the following checks occur, a level 1 interrupt request is set and the corresponding bit is set in error/control/condition register. This register is made available to the control program by executing an Input X'67' instruction. See Appendix B for the placement of the bits within this register.

1. Channel Bus In Check: This error is set when the Type 1 CA hardware detects bad parity in the byte to be sent to the host.
2. In/Out Instruction Accept Check: This error is set

when the Type 1 CA hardware detects an invalid input/output instruction.
3. CCU Outbus Check: This error is set when the Type 1 CA hardware detects bad parity on the CCU Outbus.
4. Local Store Parity Check: This error is set when the Type 1 CA hardware detects bad parity on data being read out of the local store registers.

See the *Error Indications* section in this chapter for information on how to handle these errors.

Level 3 Interrupt Request
Two different level 3 interrupt requests can be set: an initial L3 and a data/status L3. These are identified to the control program when Input X'77' is executed. Either one or both of these requests can be set. Several different events can cause either to be set.

If the Type 1 CA initial L3 is the only interrupt request set, it can be serviced without being concerned that a Type 1 CA data/status L3 interrupt request will occur. Any pending data/status transfer sequence is inhibited until the Type 1 CA initial L3 request is reset. The level 3 interrupt can be reset by either an Output X'60' or Output X'62' instruction. If an Output X'62' is executed to reset the Type 1 CA Initial L3 request, the control program must be prepared to reinitiate a data/status transfer if one was pending at the time the initial L3 was reset.

If just the Type 1 CA data/status L3 interrupt request is set, the program must be aware that a system reset sequence can occur during servicing of the data/status interrupt request. If the system reset occurs before the Input X'62' is executed, the conditions that caused the interrupt are lost to the program. A system reset causes an initial L3 interrupt request and resets all hardware latches except the 'system reset' latch.

If both interrupt requests are set, the program must service both requests before executing the output instruction to reset them. if both are set and the Initial L3 interrupt request was caused by a system reset, the conditions that caused the data/status L3 will be lost due to the system reset.

Type 1 CA Initial L3: Type 1 CA initial L3 interrupt request is when (1) an initial selection sequence occurs, (2) a system reset sequence occurs, or (3) NSC status is cleared because of an initial selection sequence. When an initial L3 interrupt occurs, Input X'60' can be executed to determine the cause.

Once the Type 1 CA initial L3 interrupt request is set, the channel adapter automatically responds with a short control-unit-busy status (Control Unit End, Status Modifier, Busy) to all initial selection attempts. This status is indicated until the control program signals the channel adapter to reset the condition that caused the interrupt request to be set. During this period, no channel command can be accepted in either emulation or native mode. The control program should signal the channel adapter as soon as possible to reset the cause of the interrupt request.

Once the Type 1 CA initial L3 interrupt request is set, subsequent data/status transfer sequences are inhibited until the interrupt request is reset. This means that the Type 1 CA initial L3 interrupt request and the Type 1 CA data/status L3 interrupt request can both be set only if the data/status request occurs first, and only if the Initial L3 interrupt request is not the result of a system reset.

The Type 1 CA initial L3 interrupt request can be reset by executing an Output X'60', X'62', or X'67', depending on the cause of the interrupt. (See the preceeding section *Level 3 Interrupt Request* restrictions on resetting level 3 interrupts.)

Data/Status L3 Interrupt Request: The Type 1 CA data/status L3 can be set by (1) any of the five data/status transfer sequences, (2) a suppress out monitor interrupt, (3) a program requested interrupt, or (4) a CCU Outbus check when the adapter is in one of the five data/status transfer sequences.

The control program requests a data/status L3 interrupt by executing an Output X'62' or Output X'67' instruction with the proper bits set for the operation desired. See Appendix B for the output instruction bit definitions. The program should not initiate another transfer sequence or monitor for the inactive condition of the 'suppress out' line unless the Type 1 CA initial L3 or the Type 1 CA dataa/status L3 interrupt has been set. The control program can signal the channel adapter at any time to cause a program requested interrupt when conditions permit. When a Type 1 CA data/status L3 interrupt occurs, Input X'62' can be executed to determine the cause.

The Type 1 CA data/status L3 interrupt request can be reset by executing an Output X'62' with the appropriate bits set. Refer to Appendix B, Output X'62' for bit definitions.

Channel Commands
The channel adapter recognizes any I/O command byte combination as a valid command if odd parity is maintained on the I/O channel interface and there is no pending activity on the channel. Except for those commands listed below, the Type 1 Channel Adapter hardware accepts the command, returns an initial selection status of X'00' and causes a Type 1 CA initial L3 interrupt request. The Type 1 Channel Adapter hardware initially accepts the full range of commands (X'00' through X'FF'). It is the responsibility of the

control program to decode the command, determine its validity and respond accordingly.

The hardware recognizes the following commands and takes the action listed.

NSC: *I/O No-Op (X'03')*--When this command is presented to the NSC address, an immediate initial status of CE, DE is presented by the hardware if the channel adapter is free of commands. If the NSC has a pending status available (a previous NSC status byte has been stacked), the hardware presents it with the 'Busy' status bit as initial status to the No-Op command. it with the 'Busy' status bit as initial status to the I/O No-Op command.

Test I/O (X'00')--When this command is presented to the NSC address, the hardware presents the current status of the NSC. If the NSC is free of commands, the hardware responds with a X'00' status byte during the initial status presentation to the Test I/O. If the NSC hardware has a pending status available in the NSC status register, this status is presented during the initial status presentation to the Test I/O. The busy bit does not accompany this status.

Test I/O and No-Op must be recognized by the control program if these immediate commands are stacked.

Write IPL (X'05')--When this command is presented to the NSC address, the hardware accepts this command, returns the initial status byte of X'00', and causes a Type 1 CA initial L3 interrupt request. The hardware also notifies the CCU (Central Control Unit) of the reception of this command by raising an internal interface line to the CCU. See Chapter 5, *Initial Program Load* for an explanation of the IPL procedures.

ESC: *I/O No-Op (X'03')*--When this command is presented to a valid ESC address, the hardware presents an immediate CE, DE initial selection status.

Test I/O (X'00')--Refer to *Emulation Mode Test I/O* in this chapter for an explanation of this operation.

The following commands are standard channel I/O commands issued by the host access method for either native or emulation mode. The control program has the responsibility to decode the command and initiate the appropriate action. Refer to *Inbound Data Transfer* and *Outbound Data Transfer* in this chapter for the sequence of instructions that the control program must execute for proper data transfer.

Write (X'01') - The channel adapter hardware accepts this command and returns, an initial selection status of X'00'. The purpose of the command is to transfer data from the host processor to the communications controller. The data may be either user data or control information to inform the control program of a function to be performed.

Read (X'02') - The purpose of this command is to transfer data from the communications controller to the host processor. The channel adapter hardware accepts the command and returns an initial selection status of X'00'.

Sense (X'04') - This channel command should result in the transfer of one byte of sense data from the communications controller to the host. The channel adapter hardware accepts the command and returns an initial status of X'00'. Normal ending status is CE, De unless a Half I/O is detected when the adapter is not initialized. In this case, a CE, DE, UC is returned to the channel. It is the responsibility of the control program to recognize the command, load the sense byte into a data register, and send it to the host. The transfer is the same as a data transfer with one byte of data. The sense byte for the Type 1 Channel Adapter is the same as for the Type 2 Channel Adapter. See Input X'53' in Appendix B for a definition of each bit.

Type 1 CA Status Presentation

Initial Selection Status

Initial Selection status is generated by hardware and presented to the channel during the initial selection sequence. The host processor must examine the status byte to determine whether the command was accepted by the channel adapter. If the command was not accepted, the status byte contains bits indicating the reason.

The following paragraphs describe the status bytes that can be returned to the host processor channel and their meanings.

X'00' (Zero Status)

NSC: The hardware accepts a command byte, other than a No-Op, or a Test I/O, and the NSC is free of commands.

ESC: The hardware accepts a command byte other than a Test I/O or No-Op command, and there is no initial select L3 interrupt request pending.

X'02' (Unit Check)
The hardware detects even parity in the command byte.

X'0C' (Channel End, Device End)
This status is returned by the hardware as an immediate response to a No-Op command.

X'10' (Busy)

NSC: The NSC hardware is active with another command and has not presented final status to that command.

X'70' (Status Modifier, Control Unit End, Busy)
This status is returned as a short control-unit-busy sequence to any initial selection sequence when any of the following conditions are present:

- The Type 1 CA has an initial selection L3 interrupt request pending due to the acceptance of a previous command, and the control program has not yet reset the interrupt request.
- The Type 1 CA hardware has previously detected a system reset indication and caused a Type 1 CA initial L3 interrupt request, and the control program has not reset the initial select L3 interrupt request.
- The Type 1 CA hardware has previously detected a selective reset during a service transfer and caused a Type 1 CA data/status L3 interrupt request, and the control program had not reset this request.
- A program requested interrupt is pending, and the control program has not reset the request.

ESC: This is the normal initial selection status returned by the hardware when a Test I/O is issued to an ESC address.

Busy Bit and Status Information

NSC: When an initial selection occurs for the NSC and status is pending in the NSC status register, the Busy bit, in addition to the pending status, is returned.

Pending Status

NSC: When a Test I/O command is issued to the NSC and status is pending in the NSC status register, the pending status is returned without the Busy bit.

Final Status
The final status is a control program generated code and should be presented to the host processor upon completion of a command. Bit definitions for the final status byte must be coordinated between the control program and the host program for proper operation. Upon completion of a command, the control program should load the appropriate register with the status byte to be presented and initiate a final status transfer sequence. Refer to *Status Transfer Sequence* in this chapter for additional information.

Emulation Mode Test I/O
When a Test I/O command is issued to an ESC address, the hardware responds with initial selection status and also causes a Type 1 CA initial L3 interrupt request. Although the initial selection status presented to the host is X'70' (Status Modifier, Control Unit End, Busy), byte 0, bit 0 of the Type 1 CA initial selection control register is set to one (normal initial selection). The control program should recognize the command and address and execute:

1. Output X'63' - To load the ESC address and the status byte to be presented to this Test I/O command.
2. Output X'62' - To set ESC Final Status Transfer Sequence, Reset Initial Selection, Reset Data/Status Control, and Set ESC Test I/O Status Available.

When the next initial selection sequence occurs, the hardware compares the address presented to the adapter with the address that had received the Test I/O command.

If these addresses compare and the subsequent command is a Test I/O command, the hardware presents the status byte that was loaded by the above instructions. The hardware then causes a Type 1 CA data/status L3 interrupt request. When the control program executes an Input X'62', the ESC Final Status Transfer Sequence bit is on, indicating a successful completion of the Test I/O sequence.

If the addresses do not compare during the initial selection sequence, or if the command is not a Test I/O, the hardware resets out of the ESC Test I/O mode and handles this sequence as a standard initial selection. If this occurs, the control program must reset the ESC final status transfer sequence by executing an Output X'62' instruction.

Between the time the Test I/O command is first issued and the time the control program executes the above Output X'62' and X'63' instructions, the hardware responds with an initial status of X'70' to any initial selection sequence from the host processor. In this case, an Output X'60' should not be used to reset initial selection because the address compare uses the address in the initial selection address control register. Refer to *Initial Selection Status* in this chapter for an explanation of X'70' status.

Native Mode Asynchronous Status

The control program initiates the presenting of asynchronous status to notify the host of an unusual condition or a required action.

The control program initiates the action by requesting a Type 1 CA program interrupt. This is done by executing an Output X'67' instruction. When the requested interrupt is allowed, the hardware sets a Type 1 CA data/status L3 interrupt request.

The control program should then prepare to present the asynchronous status by:

1. Loading the address of the NSC (Output X'63').
2. Loading the NSC status byte to be presented (Output X'66').
3. Executing an Output X'62' with the NSC Final Status Transfer set to 1. Any Output X'62' instruction resets the program requested interrupt.

At the completion of the status transfer, the hardware causes a Type 1 CA data/status L3 interrupt to inform the control program of the results of the transfer. The control program should then execute an Input X'62' instruction. If the status byte is accepted, the NSC final status transfer sequence bit is one.

Stacked Initial Status

The host processor can stack all initial status responses except zero status (X'00') to a Start I/O. When an initial status is stacked, the hardware causes a Type 1 CA initial L3 interrupt request. The control program must determine what status was stacked by analyzing the contents of the initial selection control register and the initial selection address and command byte register. This information is obtained by executing an Input X'60' and X'61'. The status and address of the stacked device must be presented later in a final status transfer. The control program must be able to distinguish between the NSC address and the ESC addresses to be able to load and transfer the proper status.

The following initial status is presented by the Type 1 CA hardware and is capable of being stacked.

Channel End/Device End (X'0C') - The initial status is presented to an No-Op command. If the initial status is stacked, the initial status byte stacked bit is on in the initial selection control register (Input X'60'). The address and command byte can be obtained by executing an Input X'61' instruction.

Unit Check (X'02') - Unit Check is presented if the hardware detects even parity in the command byte. If this initial status is presented, the initial status byte stacked bit and the channel out bus check bit is on in the initial select control register (Input X'60'). The device address may be obtained by executing an Input X'61' instruction.

NSC: Zero or any pending status to a Test I/O addressed to the NSC.

If a Test I/O command is issued to the NSC and the status is stacked, the initial status byte stacked bit is on in the initial select control register (Input X'60'). The address and command may be obtained by executing an Input X'61' instruction. The control program should not execute an Output X'66' to load the stacked status in the NSC status register. The NSC hardware maintains the stacked status from a Test I/O command in the NSC status register until the host processor has accepted it.

ESC: Initial status byte to a Test I/O addressed to the ESC.

If a Test I/O command is issued to the ESC and the initial select status is stacked, the initial status byte stacked bit is on in the initial selection control register (Input X'60'). The address and command may be obtained by executing an Input X'61'. The control program should treat this the same as a stacked status while in an ESC final status transfer.

Stacked Final Status

When a final status or an NSC Channel End is stacked, the Type 1 CA hardware causes a Type 1 CA data/status L3 interrupt request. The control program should analyze the contents of the data/status control register (Input X'62') to determine which status was stacked. One of the following actions may be taken:

1. Test the level of the 'suppress out' line at the time the last Input X'62' was executed. If the 'suppress out' line was inactive, the control program can now attempt another transfer sequence.

NSC: Since the NSC stacked status is maintained in the NSC status register the stacked status should not again be placed in the NSC status register with an Output X'66' instruction.

ESC: The status byte along with the ESC address must be loaded by an Output X'63' when preparing an ESC final status transfer.

If the 'suppress out' tag line is active, the control program can queue the stacked status; or if no data services are required, it can set the suppress out monitor as described below.

2. The *suppress out monitor* may be used to notify the control program when the 'suppress out' line falls and when the status may be presented. To use this feature, the control program should execute an Output X'62' to reset the Type 1 CA data/status L3 interrupt request and execute an Output X'67' to set a suppress out monitor interrupt. The hardware then causes a Type 1 CA data/status L3 interrupt request when the 'suppress out' line becomes inactive. When the control program executes an Input X'62', the suppress out monitor interrupt bit is active. If the channel adapter is in the ESC mode, the suppressible status bit (byte 1, bit 3) must be on in Output X'62'. The stacked status can be transferred by loading the status (in the case of ESC final status) into a general register and executing an Output X'62' with the the type of transfer indicated and reset suppress out monitor. The suppressible status bit in Output X'62' must be on.

NSC: When operating in NSC mode only, the control program can initiate another NSC status transfer immediately after being notified of a stacked status by executing an Output X'62' with the type of transfer indicated and the reset data/status L3 interrupt request bit on. The hardware does not attempt to transfer the status until the 'suppress out' line becomes inactive. It then transfers the status across the channel to the host. The suppressible status bit does not have to be on since the channel adapter hardware monitors the channel 'suppress out' tag line.

Programming Note
Whenever the channel adapter is in the ESC mode and the control program is presenting suppressible status, the suppressible status bit (byte 1, bit 3) must be on when an Output X'62' is executed. Status is suppressible if 'stacked status' is received for a particular line or when the line has been issued an interface disconnect. Refer to *IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufactures Information* (GA22-6974) for further information on suppressible status.

Status Transfer Sequences

Upon completion of a command, the control program should present a final status associated with that command. The control program may choose to separate the Channel End and Device End on an NSC command. This status is transferred by executing the following output instructions.

ESC Final Status
Output X'63'--Loads the I/O device address that is presenting the status and the status byte that is to be presented.

Output X'62'--Sets the following control bits: (1) ESC final status transfer, (2) reset initial selection (if this status transfer is being initiated because of a stacked initial status) or (3) reset data/status interrupt (for final status transfer).

NSC Final Status
Output X'63'--Loads the address of the NSC (must be loaded only once at the beginning of the program if operating in NSC mode only).

Output X'66'--Loads the NSC status byte to be presented. Output X'62'--Sets the following control bits: (1) NSC final status transfer, (2) reset initial selection (if this status transfer is being initiated because of a stacked initial status) or (3) reset data/status interrupt (for a final status transfer).

NSC Channel End Status
Output X'63'--Loads the address of the NSC (can be loaded only once at the beginning of the program if operating in NSC mode only).

Output X'62'--Sets the following control bits: NSC channel end transfer sequence and reset data/status interrupt.

When the NSC channel end transfer sequence bit is active during the execution of an Output X'62', the CA hardware loads Channel End Status (X'08') in the NSC status register.

When the Output X'62' instruction is executed in the above sequence, the CA hardware notifies the I/O channel that it requires service. In a hardware-controlled sequence, the CA hardware indicates that status is being passed to the host and gates the status byte out to the I/O channel. After this sequence is completed, the CA hardware causes a data/status L3 interrupt to inform the control program of the results of the status transfer.

The control program should react to the data/status L3 interrupt request by executing an Input X'62' instruction. If the status byte is accepted by the host processor, the related final status bit is on, (that is, ESC or NSC final status transfer or Channel End transfer sequence).

If any unusual conditions occurred during this transfer (that is, status stacked, selective reset, or interface disconnect) the related bits are on when the Input X'62' instruction is executed.

Data Transfer Sequences

Outbound or inbound data is transferred through input and output instructions executed by the control program. As described in the following paragraphs, these I/O instructions must be in the correct sequence for proper transfer operation.

Outbound Data Transfer
When reacting to a read type I/O command, which requires data to be passed from the communications controller to the host processor, the control program must execute the following instructions.

Output X'63'--Loads the address of the I/O device sending the data.

NSC: If the Type 1 CA is operating in NSC mode only, it is necessary to load this address only once. It remains constant for subsequent NSC transfers.

ESC: When operating in ESC mode, this address may change and must be updated when required.

Output X'64'--Places the first and second bytes of outbound data in the Type 1 CA data buffer byte 1/data buffer byte 2 register.

Output X'65'--Places the third and fourth bytes of outbound data in the Type 1 CA data buffer byte 3/data buffer byte 4 register.

Output X'62'--Sets the following control bits: (1) outbound data transfer sequence, (2) reset initial selection (if this transfer is the first after an initial selection), or (3) reset data/status interrupt and the request byte count. A maximum of four bytes may be transferred during one sequence.

When Output X'62' is executed, the CA hardware notifies the I/O channel that it requires service. In a hardware controlled sequence, the CA then identifies itself by gating its I/O device address onto the I/O channel and passes the required data. The CA hardware then causes a Type 1 CA data/status L3 interrupt to inform the control program of the completion of the transfer.

The control program should react to the Type 1 CA data/status L3 interrupt by executing an Input X'62'. Assuming a normal data transfer, the outbound data transfer sequence bit is on and the transferred byte count field reflects the number of bytes passed across the channel during this transfer sequence.

If any unusual conditions occurred during this transfer (that is, selective reset, interface disconnect or channel stop) their related bits are on when the Input X'62' was executed.

Inbound Data Transfer
When reacting to a write type I/O command, which requires data to be passed from the host processor to the communications controller, the control program must execute the following instructions.

Output X'63'--Loads the address of the I/O device that is transferring the data. Output X'62'--Sets the following control bits: (1) inbound transfer sequence, (2) reset

initial selection (if this is the first transfer after an initial select), (3) Reset data/status interrupt, and (4) the transferred byte count. A maximum of four bytes may be transferred during one sequence.

When the Output X'62' instruction is executed, the CA hardware notifies the channel that it needs service. In a defined sequence, the CA then identifies itself by gating its I/O device address onto the I/O channel and accepts the required number of bytes of data. The CA hardware then causes a Type 1 CA data/status L3 interrupt to inform the control program of the completion of the transfer.

The control program should react to this interrupt by executing an Input X'62' instruction. Assuming normal data transfer, the inbound data transfer sequence bit is on and the transferred byte count field reflects the number of bytes of data passed from the host processor to the controller during this sequence.

If any unusual conditions occurred during this transfer (that is, selective reset, interface disconnect or channel stop) their respective bits are active when the Input X'62' is executed.

Input/Output Instructions
The X'60' through X'67' input/output instructions are used to manipulate data, status, and control information in the Type 1 Channel Adapter. Refer to Appendix B for the definitions of the bits within the input and output instructions.

Input Instructions
Eight input instructions are assigned to the Type 1 Channel Adapter to monitor its operation and data flow. Appendix B defines the bits within each input instruction.

Input X'60' (Initial Selection Control)
This instruction loads the contents of the initial selection control register into the general register specified by R. The bits of the control register are set as a result of initial selection and reflect the status of the operation when completed.

This should be the first instruction executed after determining that the level 3 interrupt was caused by a Type 1 Channel Adapter initial selection or a system reset.

Input X'61' (Initial Selection Address and I/O Command Byte)
This instruction loads the initial selection address and the I/O command into byte 0 and byte 1 respectively of the general register specified by R.

The control program should normally execute an Input X'61' instruction when servicing a Type 1 CA initial L3 interrupt request. If the cause of the interrupt (identified by executing an Input X'60') is found to be an initial selection sequence, the addressed subchannel and channel I/O command byte can be determined by executing an Input X'61' instruction.

Input X'62' (Data/Status Control)
This instruction loads the contents of the Type 1 CA data/status control register into the general register specified by R. An Input X'62' should normally be the first instruction executed by the control program when servicing a Type 1 CA data/status L3 interrupt. It is used to determine the cause of the interrupt.

Input X'63' (Transfer Address and ESC Status Bytes)
This instruction loads byte 0 and byte 1 of the general register specified by R with the I/O device address byte and ESC status byte respectively. These are the address byte and status byte given to the channel when the last Output X'63' was executed. This instruction can be executed for checking purposes immediately after Output X'63' is executed. When servicing a Type 1 CA data/status L3 interrupt request due to a transfer sequence, the program can execute this instruction either for checking purposes or to obtain the information if not retained elsewhere.

Input X'64' (Data Buffer Bytes 1 and 2)
This instruction loads the contents of data buffer bytes 1 and 2 into bytes 0 and 1 of the general register specified by R. The control program normally executes the Input X'64' instruction when servicing a Type 1 CA data/status L3 interrupt caused by the ending of an inbound data transfer sequence in which one or more data bytes were transferred.

Programming Note
Input X'62' should be executed before the Input X'64' instruction to determine the transferred byte count. Data buffer 1 contains valid information if the transfer count is greater than zero. Data buffer 2 contains valid information if the transfer count is two or more.

Input X'65' (Data Buffer Bytes 3 and 4)
This instruction loads the contents of data buffer bytes 3 and 4 into bytes 0 and 1 of the general register specified by R. The control program normally executes the Input X'65' instruction when servicing a Type 1 CA data/status L3 interrupt caused by the ending of an inbound data transfer sequence in which three or four data bytes were transferred.

Programming Note
Input X'62' should be executed before the Input X'65' instruction to determine the transferred byte count. Data buffer 3 contains valid information if the transfer count is three or four. Data buffer 4 contains valid information if the transfer count is four.

Input X'66' (NSC Status Byte)
This instruction loads the contents of the NSC status byte register into bytes 0 and 1 of the general register specified by R. These bits reflect the status bits loaded into the NSC status byte register when an Output X'66' instruction is executed. These bits are reset either when the channel accepts an NSC status byte presentation or when the control program executes an Output X'67' instruction with byte 1, bit 3 or byte 0, bit 4 set to one.

Programming Note
This instruction should be used only as a diagnostic aid and should not be used during active communication over the channel.

Input X'67' (Type 1 CA Controls)
This instruction loads a general register with the NSC address byte and the state of various check and control latches.

The control program should execute this instruction when (1) an asynchronous status sequence is required (determined by the control program and initiated by executing an Output X'67' with byte 1 bit 1 on) or (2) there is a Type 1 CA L1 interrupt request and the control program wants to determine the cause of the interrupt.

Output Instructions

Eight output instructions are assigned to the Type 1 Channel Adapter to control its operation and data flow. Appendix B defines the bits within each output instruction.

Programming Note
If the control program executes an Output X'61' instruction, (an assigned, but unused instruction) the hardware takes no action unless execution occurs during a data/status transfer. During data/status transfer, the hardware sets the 'in/out instruction accept check' and causes a Type 1 CA level 1 interrupt request.

Output X'60' (Reset Initial Selection)
When this instruction is executed, the hardware resets both the initial selection hardware latches and the Type 1 CA L3 interrupt request resulting from an initial selection sequence. Since this instruction performs a function instead of an operation, the bit settings of the register are ignored.

Output X'62' (Data/Status Control)
This instruction sets the data/status control register according to the contents of bytes 0 and 1 of the general register specified by R.

Output X'63' (Transfer Address and ESC Status Bytes)
This instruction loads the transfer address and ESC status registers with the contents of byte 0 and 1 respectively of the general register specified by R.

This instruction should be executed before signaling the Type 1 CA that a data/status sequence is required. The I/O device address provided by the last Output X'63' executed is presented to the channel in all subsequent transfer sequences. The Output X'63' instruction must be executed each time a transfer sequence is required for a different I/O device address. This instruction must also be executed before signaling the channel adapter that an ESC final status transfer is required. This presents the I/O device address in addition to the status byte to the host channel.

Programming Note
For compatibility, the program should ensure that the bits of the ESC status byte that are set are consistent with the bits set under similar conditions in the operation of the IBM 2701/2702/2703 transmission control units.

Output X'64' (Data Buffer Bytes1 and 2)
This instruction loads data buffer bytes 1 and 2 with the contents of the general register specified by R. This instruction should be executed before signaling the Type 1 CA that an outbound data transfer sequence is required. When an outbound data transfer sequence occurs, data buffer bytes 1 and 2 are the first and second data bytes transferred.

Programming Note
The request byte count (Output X'62', byte 1, bits 6-7) must be set with the total number of valid data bytes loaded into data buffers 1-4 (by Outputs X'64' and X'65') to ensure data integrity.

Output X'65' (Data Buffer Bytes 3 and 4)
This instruction loads data buffer bytes 3 and 4 with the contents of the general register specified by R. This instruction should be executed before signaling the Type 1 CA that an outbound data transfer sequence is required. When an outbound data transfer sequence occurs, data buffer bytes 3 and 4 are the third and fourth data bytes transferred.

Programming Note
The request byte count (Output X'62', byte 1, bits 6-7) must be set with the total number of valid data bytes loaded into data buffers 1-4 (by Outputs X'64' and X'65') to ensure data integrity.

Output X'66' (NSC Status Byte)
This instruction sets bits in the NSC status register according to the contents of the general register specified by R. The Output X'66' instruction should be executed before signaling the Type 1 CA that an NSC final status transfer sequence is required.

When the NSC final status transfer sequence occurs, the status byte provided by this output is presented to the channel. If the status byte has previously been given to the channel adapter (but was stacked by the channel), it need not be given again. This output may be used to present an asynchronous status (for example, Attention) or the final status, ending a channel I/O command in the native mode.

Programming Note
For compatibility, the control program should ensure that the bits of the NSC status byte that are set are consistent with the bits set under similar conditions in the operation of the Type 2 Channel Adapter.

Output X'67' (Type 1 CA Controls)
This instruction causes various control latches to be set or reset in the channel adapter according to the bits in the general register specified by R.

Programming Note
The CA Diagnostic Reset (byte 0, bit 4) resets the channel adapter only if both interface A and interface B are disabled.

Error Indications
The programmer should decide what specific action to take in attempting to recover from an error condition. The various ways to handle errors depend on the application and installation.

The following indications require error recovery procedures.

Channel Bus In Check
When the Type 1 CA hardware detects bad parity in the byte to be sent across the I/O Channel Interface, it generates good parity for that byte and gates it onto the 'I/O channel interface' bus. The hardware then causes a Type 1 CA level 1 interrupt. The control pro-

gram should interrogate the condition register by executing an Input X'67' and record the error conditions that occurred. The control program should then reset the level 1 interrupt and end the command by presenting a CE, DE, UC status.

In/Out Instruction Accept Check
An in/out instruction accept check is set when the control program executes an Input or Output X'60', X'62', X'63', X'64', X'65', or X'66' instruction when the Type 1 CA is in the process of handling any data/status transfer sequence. A level 1 interrupt request is also set, and the instruction is prohibited from setting any hardware control latches. The control program should interrogate the condition register by executing an Input X'67' and record the error indication. An Output X'67' should be issued to reset the L1 interrupt request and the 'in/out instruction accept check' latch.

CCU Outbus Check
When the Type 1 CA hardware detects bad parity on the CCU Outbus, it sets the 'CCU Outbus check' latch, causes a L1 interrupt, and prohibits reselection on the channel interface. The control program should interrogate the condition register by executing an Input X'67' and record the error indication. The control program should then issue an Output X'67' to reset the L1 interrupt and the 'CCU Outbus check' latches. If the CA was active on the channel in a data/status state when the error occurred, the hardware terminates the transfer and causes a Type 1 CA data/status L3 interrupt request. The control program must reissue the output instructions to allow reselection to a channel interface.

Local Store Parity Check
When the hardware detects bad parity on data bytes being gated out of local store registers, it causes a level 1 interrupt. The control program should interrogate the condition register by executing an Input X'67' and record the error indication. The control program should then issue an Output X'67' to reset the L1 interrupt request and end the command by presenting CE, DE, UE status.

Chapter 9: Type 2 Channel Adapter

This chapter gives the reader a basic understanding of the operation of the Type 2 Channel Adapter and the requirements necessary to program the adapter.

The Type 2 Channel Adapter (Type 2 CA) is a high-performance, cycle-stealing adapter that supports instantaneous channel data-transfer rates of up to 376 kilobytes per second.

All data transfer between the Type 2 CAs and controller storage is by cycle steal. Two bytes are transferred during each steal cycle execution. The maximum cycle steal request rate of each Type 2 CA is 188 thousand requests per second, which allows a maximum data transfer rate of up to 376 kilobytes per second.

The Type 2 CA is minimally dependent on the control program. Its operation is initiated by the control program in much the same manner that an I/O operation is initiated in System/370. Data transfer and control word chaining are handled without control program intervention. As in System/370, the Type 2 CA notifies the control program when a data transfer is complete with an "I/O interrupt". In the 3705, this "I/O interrupt" is denoted as a Type 2 CA initiated level 3 interrupt.

Operation and Data Flow

A host processor channel command is received by the Type 2 CA, decoded, and placed into the command register. If the operation is a data transfer type (Read or Write), the Type 2 CA checks the validity of the control word (CW) address contained in the in or out CWAR (control word address register). (See the *Channel Adapter Control Word* section in this chapter.) A valid control word address should have previously been placed into the CWARs by an output instruction from the control program.

If the address in the appropriate CWAR is not valid, Unit Exception initial status is returned to the channel, and no data transfer is attempted. If the CWAR is valid, a cycle-steal operation (1) places the information from the first halfword of the CW into the Type 2 CA, (2) loads the command and flag fields of the CW into the CW command/flag registers, and (3) loads the count field into the count register. In addition, the first two bits of the starting address of the storage area where data is to be read from, or written to, are placed into the cycle-steal address register (CSAR). Another cycle steal operation places the information from the second (and final) halfword of the CW into the Type 2 CA. The second halfword, containing the remainder of the starting data address, is placed into the CSAR.

For 3705-to-channel data transfer, a halfword is transferred via cycle steal from storage to the data 1 and data 2 registers and then sent to the channel, a byte at a time.

For channel-to-3705 data transfer, a byte at a time is received from the channel and alternately loaded into the data 1 and data 2 registers. The contents of both registers (one halfword total) are then transferred to storage via a cycle steal using the CSAR address.

Each byte transferred to or from the channel causes the count register to be decremented by one. Each halfword transferred to or from storage causes CSAR to be incremented by two in order to address the next halfword storage location.

When the count register is equal to 0, the data transfer for that CW is complete, and the next sequential CW is transferred to the Type 2 CA if CW chaining is indicated. Otherwise the operation ends by causing a level 3 interrupt.

On the byte-multiplexer channel, the Type 2 CA disconnects from the host processor channel after two bytes have been transferred in burst mode across the channel. After a Type 2 CA-to-storage cycle-steal operation, the Type 2 CA reconnects to the channel to transfer another two bytes.

On selector and block-multiplexer channels, the total data transfer, from Initial Selection to Channel End status, is in burst mode.

Cycle Steal Operation

To relieve the system control program of the responsibility of transferring data and information between the channel adapter and storage, the Type 2 CA uses cycle steal. Under hardware control, the Type 2 CA "steals" machine cycles from the Central Control Unit. Cycle steal allows overlap of channel operations with control operations. When the channel adapter needs data from storage or has data to be stored, it requests a cycle steal from the CCU. A cycle-steal request has the highest priority (except for maintanence cycles) in the controller. Therefore, the CCU, at the end of the current machine cycle, permits the CA cycle-steal request to be serviced. Two bytes of information are transferred into or out of storage during each cycle steal operation.

Programming Note

Transferring an odd number of bytes to the Type 2 CA from the host processor or starting a data transfer on a non-halfword boundary should be avoided. Otherwise, coding must exist in both the host processor and the controller to recognize an odd boundary address or

an odd number of characters and handle them appropriately.

1. When the address starts on an even boundary but there is an odd byte count, the next to the last byte received is duplicated in storage to fill the last halfword addressed by the cycle steal address register.
2. When the address starts on an odd boundary, the text is shifted to the next lower even-byte boundary. If there is an odd byte count, the condition in 1 above also applies.

Type 2 Channel Adapter States

Because of the problems associated with the handling of two asynchronous interfaces--the Central Control Unit and the host processor channel interface--the Type 2 CA must be in a certain state with respect to one interface before access is allowed from the other interface.

A thorough understanding of these states is required to write or modify a Type 2 CA control program or a host processor access method. More than one of these states may be present at the same time.

- CA Active State - The Channel Adapter is in the process of transferring data or information between itself and the host processor channel. The active state exists from the time the CA accepts channel command to the time the channel accepts Device End status for that command.
- CA Level 3 Interrupt State - This state may have been initiated by the control program or by the completion of either a control word or a channel command. If the Type 2 CA *is not* in the active state, the control program can gain access to all Type 2 CA registers except the channel sense and status registers. If the Type 2 CA *is* in the active state, all registers are available to the control program. See Output X'57' in Appendix B for an additional restriction on the channel adapter mode register.
- CA Level 1 Interrupt State - The Channel Adapter enters this state only when an error condition is detected during the execution of an input or output instruction directed at the CA, during a cycle steal operation, or during transfer of information across the channel interface.

  If the channel adapter is executing a channel command when a check occurs, the command is ended with a hardware generated Channel End, Device End, and Unit Check status when the L1 interrupt is requested.
- CA Busy State - This state exists when the channel attempts an initial selection sequence and the channel adapter level 1 or level 3 'interrupt' latch is set. A Busy status is returned to the channel for all commands. Busy status is also returned when the controller is online and in a program stop state.
- Diagnostic Wrap State - This state is for diagnostic purposes and is entered when the control program executes an Output X'57' instruction with byte 1, bit 7 set to 1. In the diagnostic wrap state, the CA is forced to an offline status regardless of the position of the control panel Enable/Disable switch. All Type 2 CA registers, including the IPL portion of Output X'57', are available to the control program to be used for test functions. Output instructions X'58' and X'5B' can be used to simulate the channel interface lines. The diagnostic wrap state remains in effect until Output X'57' is executed with byte 1, bit 7 off (0).
- Hard Stop State - This state is entered when a the command currently executing ends with a Channel End, Device End, Unit Check indication. The hard stop state also causes the adapter to go offline. However, the Channel Enable light on the control panel remains *on*.

Type 2 Channel Adapter Registers

The Type 2 CA contains 13 external registers that are used by the control program for normal operations. Input/output instructions (X'50' to X'5F') are used to gain access to these registers. See Appendix B for I/O instruction bit position assignments.

With Extended Addressing, byte X, bits 6 and 7, are used in the cycle-steal address register to present a full 18-bit address to the Central Control Unit during a cycle-steal operation. This is the only register with more than 16 bits used by the Type 2 CA. For input instructions, byte X contains zeros; for output instructions, it is ignored.

Inbound Data Control Word Address Register (INCWAR)

The INCWAR (inbound data control word address register) contains the storage address of the control word (CW) to be fetched by the Type 2 CA cycle-steal hardware when a channel Write, Write Break, or Write IPL command is decoded.

The control program can load this register with an Output X'50' instruction or use it as input by executing an Input X'50' instruction. The Type 2 CA recognizes these input/output instructions only after a CA interrupt request.

The contents of the INCWAR are increased by 4 (bytes) each time a CW fetch is executed by the Type 2 CA cycle-steal hardware.

Programming Note

All CWs must begin on a halfword boundary and reside in the lower 64K bytes of storage. CW chaining

or a TIC (transfer in channel) command to an address above 64k causes a CW error condition resulting in a CA level 1 interrupt.

Outbound Data Control Word Address Register (OUTCWAR)
The OUTCWAR (outbound data control word address register) contains the storage address of the control word (CW) to be fetched by the Type 2 CA cycle-steal hardware when a channel Read command is decoded.

The control program can load this register with an Output X'51' instruction or use it as input by executing an Input X'51' instruction. The Type 2 CA recognizes these input/output instructions only after a CA interrupt request.

The contents of the OUTCWAR are increased by 4 (bytes) each time a CW fetch is executed by the Type 2 CA cycle-steal hardware.

Programming Note
All CWs must begin on a halfword boundary and reside in the lower 64K bytes of storage. CW chaining or a TIC (transfer in channel) command to an address above 64k causes a CW error condition resulting in a CA level 1 interrupt.

Control Word Byte Count Register (CWCNTR)
The control word byte count register (CWCNTR) contains the value of the byte count that was loaded from the control word currently being executed. As each byte is transferred across the channel interface, the CWCNTR decreases by 1. When the count reaches zero, either CW chaining or a CA level 3 interrupt is initiated, depending upon the state of the chain flag and zero count override flag of the control word just ended.

The count contained in CWCNTR is precise and should be used by the control program to determine the location of the end of data in storage when a channel data transfer is complete. If an odd number of bytes is transferred from the channel during a channel Write, Write Break, or Write IPL command, the next to the last byte received is duplicated in storage to fill the last halfword addressed by the cycle steal address register.

The contents of CWCNTR are available to the control program by executing an Input X'52' instruction (control word byte count).

Programming Note
The Input X'52' instruction can be recognized by the Type 2 CA only following the setting of a CA interrupt request.

Channel Adapter Sense Register (CASNSR)
The Type 2 CA sense register (CASNSR) provides data for the channel Sense command. The CASNSR is one byte long and conforms to the System/370 standard definition of sense bits 0 through 3. Bits 6 and 7, however, are unique to the 3705. All bits can be used as input to the control program by an Input X'53' instruction (sense register). These input/output instructions are recognized by the Type 2 CA only following the setting of a CA interrupt request while the Type 2 CA is in the CA active state. If Output X'53' is executed while the CA is in a CA interrupt state and the CA is *not* active, the output is ignored, and no indication is returned to the control program. The programmer should always determine the active/inactive state of the CA before issuing an output instruction to this register.

Programming Note
The setting of any CASNSR bit causes the Unit Check bit to be set in the Type 2 CA status register and also causes the termination of any data transfer that may have been in progress. CASNSR is reset during initial selection whenever the Type 2 CA accepts a command other than Sense, Test I/O, or No-Op.

Channel Adapter Status Register (CASTR)
The channel adapter status register (CASTR) contains the standard System/370 status byte. The control program can load this register by executing an Output X'54' instruction (set status register bits) and can use it as input by executing an Input X'54' instruction (status register). These input/output instructions are recognized by the Type 2 CA only following the setting of a CA interrupt request while the Type 2 CA is in the CA active state.

Channel Adapter Control Register (CACR)
The channel adapter control register (CACR) is a collection of control latches to be used by the control program when initiating or terminating a Type 2 CA operation. The control program can load this register with an Output X'55' instruction (set control register bits) and can use it as input by executing an Input X'55' instruction (control register). These input/output instructions are recognized by the Type 2 CA only following the setting of a CA select and a CA interrupt request. The Type 2 CA need not be in the CA active state.

Channel Adapter Check Register (CACHKR)
The channel adapter check register (CACHKR) is a collection of latches that cause the level 1 Type 2 CA check interrupt request to be set. By checking the register, the level 1 interrupt check routine can deter-

mine the exact cause of the Type 2 CA level 1 check. All the latches in this register are automatically reset at the end of the first level 1 interrupt that occurs after they have been set. The control program can use this register as input by executing an Input X'56' instruction (check register). This input instruction can be recognized only following the setting of a Type 2 CA interrupt request.

Channel Adapter Mode Register (CAMR)
The channel adapter mode register (CAMR) can be loaded by the control program by an Output, X'57' (channel adapter mode register) at any time to initialize or halt the Type 2 CA. This register cannot be used as input to the control program.

Channel Bus Out Diagnostic Register (CBODR)
The channel bus out diagnostic register (CBODR) is used to simulate I/O 'bus out' when the Type 2 CA is in the diagnostic wrap state. The control program can load this register by executing an Output X'58' instruction (channel bus out diagnostic register) and can use it as input by executing an Input X'58' instruction. If used by the control program when the adapter is not in the diagnostic state, the register indicates the current state of the host processor channel 'bus out'.

Cycle Steal Address Register (CSAR)
This register is the interface to the 'cycle steal address bus'. It contains the current data address while data transfer is in progress. With each data halfword (two bytes) transferred to the channel, this address increases by two. The register is initially loaded with the control word (CW) address at the beginning of a CW-fetch operation and then is loaded with the starting data address when the CW fetch is complete.

The control program can use this register as input by executing an Input X'59' instruction. The CSAR is not available for output.

Channel Adapter Data Buffer (CADB)
The channel adapter data buffer register (CADB) forms the CA buffer for all channel data being transferred through the CA for either normal or diagnostic operations. This register physically shares a local store array with the INCWAR and OUTCWAR. The control program can load this register or use it as input by executing an Output or Input X'5A' instruction. In the diagnostic state, the control program should ensure good parity in this register by executing an Output X'5A' instruction before attempting an Input X'5A' instruction.

Channel Tag Diagnostic Register (CTDR)
The channel tag diagnostic register (CTDR) contains two bytes used to simulate 'tag out' and interrogate 'tag in' signals for diagnostic purposes. If accessed by the control program, when the channel adapter is not in the diagnostic state, the register indicates the current status of the host processor/3705 'tag' interface. The control program can load byte 0 of this register with an Output X'5B' instruction, and it can use the entire register as input by executing an Input X'5B' instruction.

Channel Adapter Command Register (CMDR)
The CA command register (CMDR) indicates either the current or the last command and control word executed.

The control program can use this register as input by executing Input X'5C' (command register) instruction. The CMDR is not available for output.

Channel Adapter Control Word

The channel adapter control word (CW) specifies the operation to be performed by the CA in conjunction with a host processor channel operation. Control words are built by the control program according to the operation to be performed. See Figure 29, Type 2 Channel Adapter Control Word.

The following paragraphs describe the various fields of the CA control word.

Command Code: Bits 0-1 specify the operation to be performed. See *Control Word Command Codes* in this chapter.

Zero Count Override: Bit 2 is used with the command chain flag to determine what the Type 2 CA should do when the current CW count is exhausted. A level 3 interrupt request is set when the CW count reaches zero and this flag is on. When the chain flag is off and this bit is on, the Type 2 CA disconnects from the byte-multiplexer channel without giving Channel End or remains connected to the burst channels without giving Channel End. The CA resumes data transfer when the proper 'CWAR valid' latch comes on. If the chain flag is on, a level 3 interrupt is requested before CW chaining is executed. Thus, the CA can transfer multiple buffers under a single host processor channel command with a minimum assignment of buffers to the Type 2 CA.

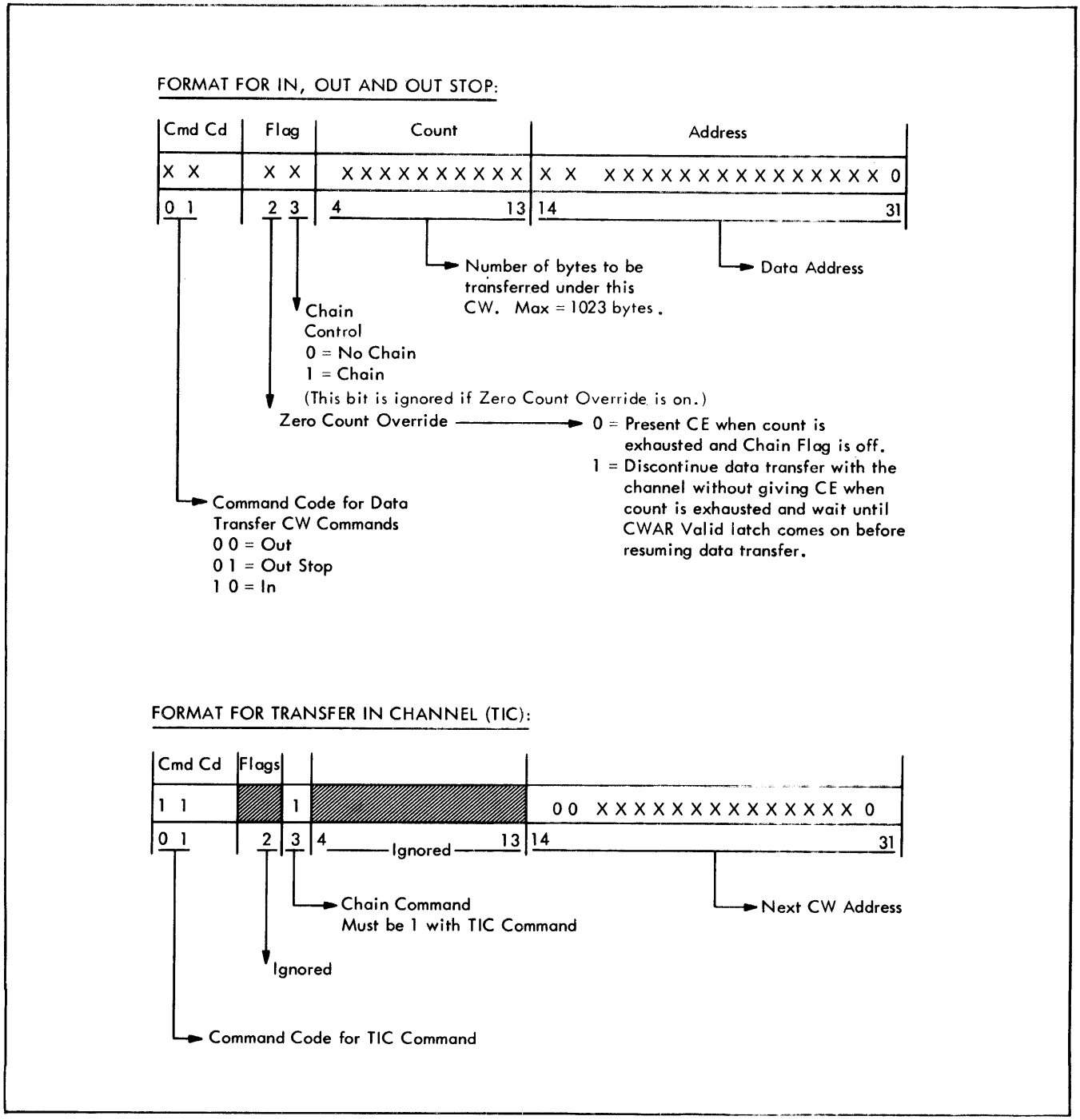


Figure 29. Type 2 Channel Adapter Control Word

Chain Flag: Bit 3 can cause multiple CWs to be used for one operation. When this bit is on, the appropriate 'CWAR valid' latch remains set during the CW-fetch operation. When the CW count reaches 0 while the Type 2 CA is executing a Read or Write channel command, control word chaining to the next sequential CW occurs automatically. This bit must be on for a TIC (transfer in channel) control word.

Count: Bits 4-13 specify the number of bytes to be transferred across the channel interface by the CW. The maximum count is 1024 bytes. This field is ignored for TIC control words.

Data Address: Bits 14-31 contain the storage address where data is to be written or read from, depending on the command code. This address is restricted to halfword addresses; therefore, all data buffers and control words must begin on a halfword boundary. For a TIC control word, this field contains the address of the next CW to be executed.

Control Word Command Codes
The following four CW commands are recognized by the Type 2 CA. The first three are operational and indicate the direction of data movement between the controller and the host processor. The fourth is used for chaining control words.

Out (X'00'): This control word command is used to control data transfer from the communications controller to the channel. If the count register reaches to 0 and CW chaining is indicated, the CA immediately chains to the next CW without ending the channel command. If CW chaining is not indicated and the zero count override bit is not on, the current channel command is ended with CE status, and a level 3 interrupt request is set. If the zero count override bit is on, a level 3 interrupt is requested without presenting any status to the channel. If both zero count override and CW chaining are on, a level 3 interrupt is requested before chaining is executed.

Out Stop (X'01'): This control word command is also used to control data transfer from the communications controller to the channel and is used to facilitate data blocking. If the count register decrements to 0 and CW chaining is indicated, the current channel command ends immediately with CE, DE. A level 3 interrupt is *not* requested. If CW chaining is not indicated, the current channel command is ended with CE status, and a level 3 interrupt is requested. If both zero count override and CW chaining are on, a level 3 interrupt is requested before chaining is executed.

In (X'10'): This control word command is used to control data transfer from the channel to the communications controller. If the count register reaches 0 and CW chaining is indicated, the Type 2 CA chains immediately to the next CW without ending the channel command. If CW chaining is not indicated and the zero count override bit is not on, a level 3 interrupt request is set after the current channel command is ended with Channel End status. If the zero count override bit is on, a level 3 interrupt is requested without presenting any status to the channel. If both zero count override and CW chaining are on, a level 3 interrupt is requested before chaining is executed.

TIC (X'11'): The Transfer-In-Channel command causes the CA cycle-steal hardware to load the address field of the CW into the appropriate CWAR. This causes a transfer to another string of CWs. The Type 2 CA must complete another CW fetch request to resume data transfer across the channel interface.

Control Word Conventions
All control words must begin on a halfword boundary and reside in the lower 64K bytes of storage. CW chaining or a TIC (transfer in channel) to an address above 64K causes a CW error condition resulting in a CA level 1 interrupt.

The control word names include "In" or "Out" to denote the direction of data transfer with respect to the communications controller. Thus an "In" control word must be coupled with a channel Write, Write Break, or Write IPL command, and the "Out" and "Out Stop" CWs must be coupled with a channel Read command.

Channel Adapter Interrupt Requests

The Type 2 CA has two interrupt requests assigned. A check interrupt request for level 1, and a normal service interrupt request for level 3. All programming operations for the Type 2 CA are performed in these two interrupt levels with the use of input/output instructions.

A level 1 interrupt request is caused when a Type 2 CA hardware or program error is detected. The error condition is set in the channel adapter check register. This register is available to the control program with an Input X'56' instruction, and error recovery may then be attempted.

A level 3 interrupt request occurs whenever the channel adapter requires service. The following conditions cause a level 3 interrupt. The first two conditions are the method by which the control program is informed of the completion of a Type 2 CA operation. The last condition is used to initialize the channel adapter for operation.

1. An Out Stop control word is being ended where CW chaining is not indicated, and channel command chaining is indicated.
2. An In, Out, or Out Stop control word is being executed, the count has reached zero, and either the zero count override CW flag is on, or no CW chaining is indicated.
3. The Type 2 CA is in an inactive state without stacked status or pending status, and one of the following has occurred.
   a. The 'program-requested level 3' latch has been set. A level 3 interrupt request may be set by executing an Output X'57' instruction with byte 1, bit 0 on along with the appropriate channel adapter (byte 1, bit 4).
   b. A program stop has been indicated by an output instruction to the channel adapter mode register while the adapter was active.
   c. A selective reset or system reset has been initiated.

Channel Commands

Because the 3705 control program handles most of the functions previously handled by the host access method, the number of commands accepted by the controller is been greatly reduced. Only the following six commands are accepted. All others cause a command reject to be returned to the host.

Write (X'01')

This channel command executes a data transfer from the host processor channel to the controller. The data may be user data or a control message to inform the control program of a function to be performed. The data transfer is terminated by (1) a Channel Stop indicated in response to a data service request, or (2) the control word count becoming zero and no further CW chaining being indicated.

Write Break (X'09')

This channel command causes the CA to perform the same functions as a Write command (X'01'). The only difference is that this command sets byte 1, bit 1 (Write Break command remember) in the Type 2 CA control register (Input X'55'). This permits the host processor to inform the control program of the point it has reached in the host CCW chain.

Write IPL (X'05')

This channel command transfers object code from the host processor to controller storage. When the command is decoded during initial selection, the Type 2 CA presents a level 3 interrupt request and causes the Central Control Unit to initiate IPL phase 1. When the CCU read-only-storage code resets the interrupt request, the Type 2 CA initiates a CW fetch and begins data transfer across the channel. Each time two bytes are received from the channel, a Type 2 CA cycle-steal operation places the data in storage and increments the CSAR by 2.

The communications controller ends this command in the same manner as a normal Write command; that is, Channel End is generated by the CA hardware, and Device End is generated by interrupt-driven code. This implies that the level 3 interrupt program must have been loaded by the Write IPL command just concluded. If a Halt I/O is received while a Write IPL command is being executed, the host processor must reissue the Write IPL command.

Refer to Chapter 5 for further information on IPL.

Read (X'02')

This channel command executes a data transfer from the communications controller to the host processor channel. The operation is normally ended when a Channel Stop command is issued or the CW count register reaches zero, and either of the following occurs:

1. The Type 2 CA is under control of an Out Stop control word with CW chaining indicated. With this condition, the Type 2 CA presents Channel End and Device End status to the channel and does not request a level 3 interrupt.
2. The Type 2 CA is under control of either an Out CW or Out Stop CW with no CW chaining indicated and with the zero count override bit off. With this condition, the Type 2 CA presents Channel End status to the host processor channel and requests a level 3 interrupt. The control program must then signal the CA hardware to present Device End and any other appropriate status to the channel.

If the Read command is terminated by a Channel Stop or Halt I/O, the Type 2 CA presents Channel End status immediately and requests a CA level 3 interrupt. The control program must signal the Type 2 CA to present Device End and other appropriate status.

Sense (X'04')

The Sense command transfers one byte of sense data from the controller to the host processor channel. This byte is described in the channel adapter sense register (CASNSR). The normal ending status for the Sense command is Channel End, Device End after the channel has accepted the sense byte. No interrupt request is generated.

Except for the not-initialized bit, the sense byte is reset during an initial selection when any host processor command, other than Sense, No-Op, or Test I/O, is presented.

No-Op (X'03')
The controller responds immediately to this command with Channel End, Device End in the initial status. No further action occurs, and an interrupt request is not made.

Test I/O (X'00')
The controller presents the following status indications as initial status in response to a Test I/O.

1. Zero status indicates that the CA is command-free, and there is no pending or stacked status.
2. Busy status is indicated from the acceptance of a command until the generation of Device End. If CE or DE is pending, Busy and CE or DE are presented together.
3. Device End (and any other status) is presented if the ending status is pending or stacked from the previous host processor command.

Once the initial status has been accepted, the Test I/O operation is complete. An interrupt request is not set, and no further action is taken.

Status Servicing

The status byte informs the host processor channel if the channel adapter is available to communicate. This status byte is generated by either the Type 2 CA hardware or the control program and is stored in the channel adapter status register (CASTR).

The five acceptable status configurations for the communications controller are described below.

Initial Status
Initial status is generated by the Type 2 CA hardware without control program intervention. The valid initial status bits are:

All zeros: The channel command is accepted.

Channel End, Device End: This is an unconditional response to a No-Op.

Unit Check: An error condition has occurred, and the channel command cannot be executed because:

a. The command is invalid.
b. The channel adapter is not initialized.
c. A bus-out parity error was detected in the command byte.

Further information on the error can be obtained by examining the sense byte.

Unit Exception: This status occurs when the appropriate control word address register (CWAR) does not contain a valid address. This status is only valid for a Read, Write, Write Break, or Write IPL command.

Unit Exception: This status occurs when the appropriate control word address register (CWAR) does not contain a valid address. This status is only valid for a Read, Write, Write Break, or Write IPL command.

Busy: The controller is already in the process of executing a channel command, or the channel adapter level 1 or level 3 interrupt request latch is set. If ending status has been generated for the command currently in execution, this ending status is presented with Busy as initial status to initial selection. The only exception occurs when the channel command indicated in the initial selection is a Test I/O.

Ending Status
At the termination of a channel command, the Type 2 CA generates the following status bits to be passed to the host processor.

Channel End: This status is hardware-generated under normal circumstances for host processor Read, Write, Write Break and Write IPL commands. Whenever the Type 2 CA generates CE alone, it also requests a level 3 interrupt. The control program is thus informed of the circumstances that caused the Channel End. The control program must complete the channel operation by signaling the Type 2 CA to present Device End and any other appropriate status.

Channel End, Device End: Channel End, Device End presented together indicates that the current command has been terminated normally. This status is normally generated as the ending status for a channel Read command if combined with an Out Stop control word. Channel End, Device End is also the normal ending status for a No-Op or Sense command.

Channel End, Device End, Unit Check: This status is generated by the Type 2 CA when any one of the following three conditions exist.

a. When a sense bit is set during the execution of a Read, Write, Write Break, or Write IPL command. The Type 2 CA also requests a level 1 interrupt to inform the control program of the error.
b. When a program stop is indicated by the control program before Channel End status is set during any data transfer command. CASNSR bit 6 (abort sense) is also set, and a level 3 interrupt request is generated.
c. When the 'hard stop' latch is set.

Device End: This status is present to the host processor when the level 3 interrupt is reset after a CE status has been presented. The status bits generated by the control program are presented along with DE. This status occurs only for host processor data-transfer commands.

Stacked Status

Stacked status occurs when the channel adapter has status information to present to the channel, but the channel cannot except it at that time.

When the channel indicates stacked status, the controller status is retained in the channel adapter status register until accepted by the host processor channel. All status except all zeros to a Test I/O is subject to being stacked by the channel.

Stacked ending status is handled in the same manner as stacked initial status. All ending status conditions can be stacked.

Asynchronous Status

The Type 2 CA presents asynchronous status to the host channel under one of three conditions:

1. The channel adapter receives an Output X'55' instruction from the control program with byte 0, bit 6 set to 1 (set program requested attention), and the CA is not active. Attention status is then presented to the host channel. Attention status is then presented to the host channel.
   This causes the CA to present an asynchronous status of Device End and Unit Check to the host channel and indicates that the controller requires an IPL from the host processor.
3. An asynchronous Device End is presented to the host channel when an interrupt is reset that had caused a Busy initial status to be presented in response to an initial selection.

Two Channel Operation

A maximum of two Type 2 Channel Adapters can be installed in the 3705, one in the base module and in the first expansion module. Both Type 2 CAs can be connected to the same channel, or each to a different channel. Both Type 2 CAs can operate simultaneously.

The control program must issue an Output X'57' instruction to select the channel adapter it wants to use. Setting byte 1, bit 4 of this output to 1 selects Type 2 CA-1. Setting this bit to 0 selects Type 2 CA-2.

Input/Output Instructions

The Type 2 Channel Adapter input/output instructions enable the control program to communicate with the host processor I/O channel. These instructions are available to the control program only when the Type 2 CA is selected and (1) a level 1 or level 3 interrupt has been requested or (2) the CA is in the diagnostic wrap state. Type 2 CA instructions are specified by X'50' to X'5F'.

Input Instructions

Twelve input instructions allow the control program to obtain the status of various channel adapter registers. Listed below are the usable Type 2 CA input instructions and the registers associated with them. The register descriptions are found elsewhere in this chapter. See Appendix B for the input instruction bit definitions.

When the adapter is selected and a CA L1 or L3 interrupt request is set, or the CA is in the diagnostic wrap state, execution of Input X'57', X'5D', X'5E', or X'5F" results in loading all zeros into the general register specified by the R operand. If the CA is not selected or an interrupt has not been requested, any attempt to execute an Input X'50' through X'5F' results in an in/out check L1 interrupt request.

Input X'50': Inbound Data Control Word Address Register (INCWAR).

Input X'51': Outbound Data Control Word Address Register (OUTCWAR).

Input X'52': Control Word Byte Count Register (CWCNTR).

Input X'53': Channel Adapter Sense Register (CASNSR).

Input X'54': Channel Adapter Status Register (CASTR).

Input X'55': Channel Adapter Control Register (CACR).

Input X'56': Channel Adapter Check Register (CACHKR).

Input X'58': Channel Bus Out Diagnostic Register (CBODR).

Input X'59': Cycle Steal Address Register (CSAR).

Input X'5A': Channel Adapter Data Buffer (CADB)

Input X'5B': Channel Tag Diagnostic Register (CTDR)

Input X'5C': CA Command Register (CMDR)

Output Instructions

Ten output instructions allow the control program to set the bits of various channel adapter registers. Listed below are the usable Type 2 CA output instructions and the registers associated with them. The register descriptions are found elsewhere in this chapter. See Appendix B for output instruction bit definitions.

When the adapter is selected and a CA L1 or L3 interrupt request is set or the CA is in the diagnostic wrap state, the execution of an Output X'52', X'59', X'5C', X'5D', X'5E', or X'5F' instruction is ignored. If the CA is not selected or an interrupt has not been requested, any attempt to execute an Output X'50' through X'5F' results in an in/out check L1 interrupt request.

Output X'50': Inbound Data Control Word Address Register (INCWAR).

Output X'51': Outbound Data Control Word Address Register (OUTCWAR).

Output X'53': Channel Adapter Sense Register (CASNSR).

Output X'54': Channel Adapter Status Register (CASTR).

Output X'55': Channel Adapter Control Register (CACR).

Output X'56': Channel Adapter Control Register (CACR).

Output X'57': Channel Adapter Mode Register (CAMR).

Output X'58': Channel Bus Out Diagnostic Register (CBODR).

Output X'5A': Channel Adapter Data Buffer (CADB).

Output X'5B': Channel Tag Diagnostic Register CTDR).

Error Indications

When the Type 2 Channel Adapter hardware detects an error condition, a level 1 interrupt request is automatically set. With this request, a bit is also set in the channel adapter check register to indicate the nature of the error. The level 1 interrupt handling routine, using an Input X'56' instruction, should then analyze the result of this input to determine the cause of the error and execute appropriate recovery procedures. Refer to the channel adapter check register (CACHKR) and Appendix B for the error bit definitions and register reset.

The following is a list of valid Type 2 CA error checks.

Invalid CWAR Address
Invalid Control Word Format
Data Address Error
Buffer Parity Error
Outbus Parity Error
Inbus Parity Error
Channel Bus In Parity Error

The control panel contains the switches and lights necessary to operate and control the communications controller. The need for operator control is held to a minimum by controller design and the governing control program. The result is fewer and less serious operator errors.

This chapter describes only those push buttons, switches, and lights that apply to the control program operation and/or program debugging. See Figure 30 for the control panel layout. For a complete description of the control panel functions, refer to the *3705 Operators Guide*, GA27-3055.

Display A and Display B Lights

The two rows of lights referred to as Display A and Display B are used to display the contents of certain Central Control Unit (CCU) registers and indicate CCU check and status information. The information displayed is selected by the Display/Function Select rotary switch.

Check and Status Information: When the Display/Function Select switch is not in the STATUS position, Display A and Display B lights show the contents of display registers 1 and 2. See the *Display/Function Select Switch* section in this chapter.

When the Display/Function Select switch is in the STATUS position, the lights of Display A and Display

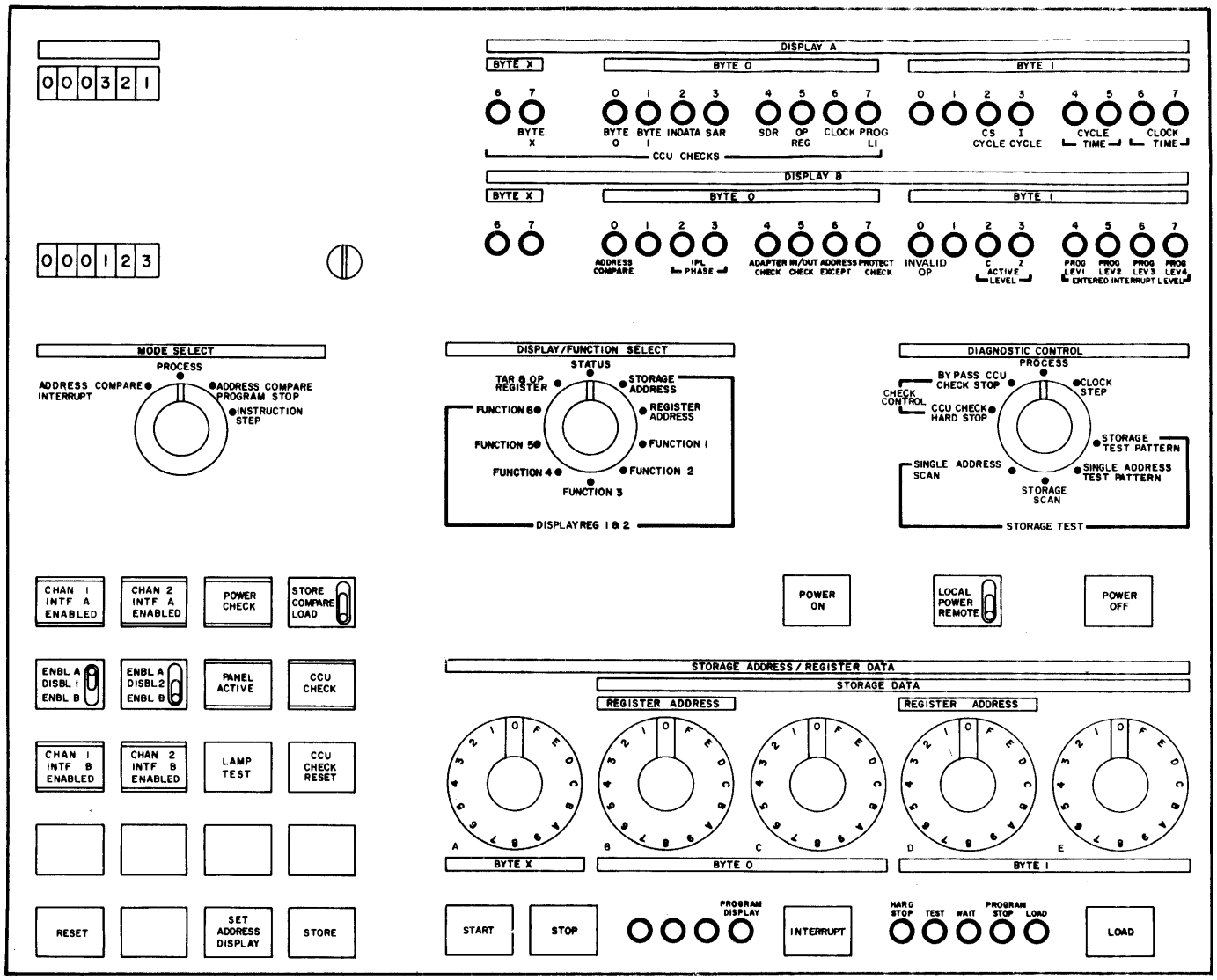


Figure 30. Control Panel

B show the check and status information designated by the labeling beneath the light positions.

The CCU check lights in Display A are driven by the CCU Check Register. The light associated with a particular CCU check comes *on* when the check is detected. In addition, the CCU Check light (lower left panel) comes *on*. Both remain on until one of the following occurs:

- The operator presses the CCU Check Reset or Reset push button, and no further checks are detected.
- The program executes an Output X'77' with byte 0, bit 1 (reset CCU check) on in the register specified by the R operand, and no further checks are detected.

Refer to the *CCU Checks* section in Chapter 5.

Display B check and status lights are as follows:

Address Compare: When the Diagnostic Control switch is in the LOAD ADDRESS COMPARE or STORE ADDRESS COMPARE position, this light comes on when an address comparison occurs. If the Mode Select switch is in the ADDRESS COMPARE INTERRUPT position, the light goes *off* when a reset occurs or when Output X'77' is executed with byte 1, bit 4 (reset address compare L1) on in the register specified by the R operand. If the Mode Select switch is not in this position, the Address Compare light goes *off* at the beginning of the next instruction executed after the address compare operation is performed.

IPL Phase: When an IPL operation is initiated, these two lights give a binary designation of which of the three IPL phases (IPL Phase 1-3) is in progress.

Adapter Check: This light comes *on* when any adapter requests a program level 1 interrupt. It remains on until the interrupt request is reset.

IN/OUT Check: This light comes *on* if an IN/OUT Check is detected by the CCU during the execution of an input or output instruction. It remains on until a reset occurs or Output X'77' is executed with byte 1, bit 5 (reset program checks L1) on in the register specified by the R operand.

Address Exception: This light comes *on* when an address exception check is detected. It remains on until a reset occurs or Output X'77' is executed with byte 1,bit 5 (reset program checks L1) on in the register specified by the R operand.

Protect Check: This light comes *on* when a storage protection check is detected. It remains on until a reset occurs or Output X'77' is executed with byte 1, bit 5 (reset program checks L1) on in the register specified by R.

Invalid Op: This light comes *on* when an invalid operation code is detected. It remains on until a reset occurs or Output X'77' is executed with byte 1, bit 5 (reset program checks L1) on in the register specified by R.

C, Z Active Levels: These two lights represent the state of the C and Z condition latches for the active program level.

Entered Interrupt Levels: These four lights display which of the four interrupt levels (program levels 1-4) are entered. Any or all four lights may be on simultaneously. The light associated with a given program level comes *on* when an interrupt occurs for that program level. It remains on until an Exit instruction is executed at that program level, or the 'interrupt entered' latches are reset because of a machine reset. If one or more of these lights are on, the one associated with the highest program level designates the active program level. If none of these lights are on, program level 5 is the active program level.

Panel Active Light

The Panel Active light comes *on* when power comes *on* unless the Mode Select and Diagnostic Control switches are not in the PROCESS position. When this light is *off*, the settings of these switches are overridden, and the controller functions are if they were in the PROCESS position. When the panel is in the inactive state, no push button are effective except Power Off.

In addition, if the light is off, no channel interface can be enabled. (Channel Interface Enable/Disable switches are overridden.)

Start Push Button

Pressing and releasing this push button causes the 'hard stop' and 'program stop' latch to be reset. The controller resumes operation in the mode selected by the Mode Select and Diagnostic Control switches.

The Start push button is effective only if the 'hard stop' latch or 'program stop' latch is on (unless the Diagnostic Control switch is in the CLOCK STEP position). The Panel Active light must also be *on*.

Stop Push Button

Pressing this push button sets the 'program stop' latch, which stops program execution at the next instruction boundary. This push button does not stop adapter cycle steal operations or maintenance cycle steal operations (for example, if the Diagnostic

Control switch is in STORAGE SCAN position). This push button is effective only if the Panel Active light is *on*.

Interrupt Push Button
The Interrupt push button is pressed to request a program level 3 interrupt. To control the action taken by the program, the operator can set the Display/Function Select switch and Address/Data switches according to whatever convention is established by the program handling the interrupt request.

Pressing and releasing this push button causes the Interrupt push button L3 request to be set. To reset this request, the program should execute Output X'77' with byte 0, bit 2 (reset Interrupt push button L3) on in the register specified by the R operand.

This push button is effective only when the Panel Active light is *on*.

Load Push Button
The Load push button is pressed to manually cause the start of Initial Program Load (IPL). This event also causes a reset to the controller. Refer to the *Initial Program Load* section in Chapter 5.

This push button is effective only when the Panel Active light is *on*.

Load Light
The Load light comes *on* whenever an IPL operation is initiated and remains on throughout the IPL sequence. It should be turned *off* by the execution of the Output X'79' instruction in the final phase of IPL.

See *Initial Program Load* in Chapter 5.

Program Stop Light
This light comes *on* when the execution of the control program has stopped. Conditions that cause a program stop are:

- An address compare occurs when the Mode Select switch is in the ADDRESS COMPARE PROGRAM STOP position.
- Any of the conditions that cause a 'hard stop' occurs.
- The Stop push button is pressed.
- The Mode Select switch is in the INSTRUCTION STEP position, and an instruction is executed.
- The Diagnostic Control switch is in a storage test position, and the Start push button is pressed. Program execution resumes when the condition that caused the light to come on is cleared, and the Start push button is pressed.

Wait Light
The Wait light is *on* when the CCU is in the Wait state (that is, the CCU is running, but instruction cycles and cycle steal cycles are not being taken). The light goes *off*, and the controller exits the Wait state if an interrupt occurs or a cycle steal cycle is taken.

Test Light
The Test light is *on* when the Mode Select or Diagnostic Control switches are not in the PROCESS position. It can also be turned *on* and *off* by executing Output X'79' (byte 1, bit 2, set test light; byte 1, bit 3, reset test light).

Hard Stop Light
When this light is *on*, the controller has ceased all operations, and manual intervention is required. Conditions that cause a hard stop are:

1. A CCU check is detected during IPL, or the Diagnostic Control switch is in the CCU CHECK HARD STOP or STORAGE SCAN position.
2. The RESET push button is pressed.
3. The Diagnostic Control switch is turned to the CLOCK STEP position.
4. An Output X'70' instruction is executed.

In the case of CCU checks, normal operation may be resumed by pressing CCU Check Reset and Start if there are no further check conditions. Otherwise, an IPL is required for this and all other conditions.

Program Display Light
The Program Display light is *on* when either display register 1 or display register 2 contains program output. (Output X'71' or Output X'72' has been executed.) If on, the operator should place the Display/Function Select switch in a position other than TAR/OP REG or STATUS in order to display the data loaded into display register 1 and/or display register 2 by the program.

Mode Select Switch
The Mode Select switch controls the normal mode of operation. The Mode Select switch can be switched between positions with no effect on normal operation, except as described for the switch settings. The Mode Select switch is overridden when the Diagnostic Control switch is in one of the storage test positions.

The Mode Select and Diagnostic Control switches must be switched to the PROCESS position at least once after power is turned on in the controller in order to activate the control panel (see Panel Active light).

The following paragraphs describe the Mode Select switch settings.

Process: This is the normal operating position for this switch. When the switch is taken out of the PROCESS position, the controller is put into a test state, and the Test light turns *on*.

Address Compare Interrupt: In this position, an address compare causes the address compare L1 interrupt request to be set and the Address Compare light to come *on*. An address compare is detected when the address in the storage address register compares with the address in the Address/Data switches A-E on the control panel. The operation that causes the compare is determined by the position of the Load/Store Compare switch. (See Load/Store Compare switch).

Address Compare Program Stop: In this position, an address compare causes the controller to be placed in the program stop state. An address compare is detected when the address in the storage address register compares with the address in the Address/Data switches A-E on the control panel. The operation that causes the compare is determined by the position of the Load/Store Compare switch. (See Load/Store Compare switch).

Instruction Step: In this position, one instruction is executed each time the Start push button is pressed and released. After each instruction is executed, the 'program stop' latch is set. Cycle steal requests are handled in the normal manner.

This position inhibits all interrupts, except program level 1 interrupts and PCI interrupts to higher-priority program levels, until an Exit instruction is executed. After an Exit instruction is executed, the machine-cycle priority controls then determine which program level is active until the next Exit instruction or PCI to a higher-priority level is executed, or a program level 1 interrupt request is set.

Unless already set, the Interval Timer L3 interrupt request is also prevented from being set when the Mode Select switch is in this position.

Display/Function Select Switch
The following positions of the Display/Function Select switch determine whether display registers 1 and 2 or CCU check and status information is displayed in the Display A and Display B lights. The contents of display registers 1 and 2 are dependent on program or control panel operations. Figure 31 shows the various operations that set the display registers.

Status: In this position, the Display A and Display B lights display the check and status information designated by the labeling beneath the light positions. See Display A and Display B.

TAR/OP Reg: In this position, the contents of the temporary address register are displayed in Display A, and the contents of the operation register are displayed in Display B. In any other position of this switch, Display A shows the contents of display register 1, and Display B shows the contents of display register 2.

Storage Address and Register Address: These two positions are used in conjunction with the Set Address/Display push button and the Store push button. They determine whether a storage location or a register is to be accessed in these push button operations.

Function Select 1-6: These positions can be used to give directional information to the program. When the switch is placed in a given position and the operator presses the Interrupt push button, the program can sample the switch setting by executing an Input X'72' instruction and taking whatever action it associates with that setting.

Used in this manner the Display/Function Select switch, in conjunction with the Interrupt push button and Address/Data switches, could allow the operator to perform such functions as online parameter changes, program controlled displays, and diagnostic test programs if these functions are program-supported.

Address/Data Switches (rotary)
These five switches (A, B, C, D, and E) are 16-position rotary switches labeled with the hexadecimal characters 0 to F. The values set in these switches are placed in the register specified by the R operand when an Input X'71' instruction is executed.

Note: *With Extended Addressing, Switch A only has the logical values of X'0' through X'3'. Without Extended Addressing, this switch is ignored.*

These switches can be used in conjunction with the Display/Function Select switch and Interrupt push button to provide control information or addresses to the program (Input X'71'). These switches are also used in the following control panel operations to provide data/addresses.

Control Panel Operation	*Address/Data Switch Used*
Set Address/Display push button	Register address (B,D) or storage address (A-E)
Store push button	Register data (A-E) or storage data (B-E)
STORAGE TEST PATTERN	Storage data (B-E)
SINGLE ADDRESS TEST PATTERN	Storage data (B-E)
SINGLE ADDRESS SCAN	Storage address (A-E)

Operation	Display Register 1	Display Register 2
Output X'71'	Contents of R	Not affected
Output X'72'	Not affected	Contents of R
Set Address/Display push button		
a. REGISTER ADDRESS	Register address (Address/Data switches B & D)	Contents of addressed input register
b. STORAGE ADDRESS	Storage address (Address/Data switches A-E)	Contents of addressed storage location
Store push button		
a. REGISTER ADDRESS	Register address (contents of TAR)	Data stored (Address/ Data switches A-E)
b. STORAGE ADDRESS	Storage address + 2 where data was stored	Contents of storage at address in DR1
Diagnostic Control switch		
a. STORAGE SCAN	Storage address + 2 (TAR + 2)	Contents of storage at address in DR1
b. SINGLE ADDRESS SCAN	Storage address (Address/Data sw A-E)	Contents of storage at address in DR1
c. STORAGE TEST PATTERN	Storage address (TAR stepping)	Contents of storage location (Address/ Data sw B-E)
d. SINGLE ADDRESS TEST PATTERN	Storage address that is currently in TAR	Contents of storage location (Address/ Data sw B-E)

Figure 31. Operations that Set the Display Registers.

CCU Check Light

The CCU Check light comes *on* when any of the CCU check conditions are detected. The specific check may be shown in the Display A lights by using the Display/Function Select switch. Also, the CCU check register is available to the control program through an Input X'7D' instruction.

CCU Check Reset Push Button

Pressing this push button causes the CCU check register and the CCU Check light to be reset if there are no longer any CCU checks present.

This push button is effective only if the Panel Active light is *on*.

Reset Push Button
Pressing the Reset push button causes a general reset to the controller which:

- Sets the 'hard stop' and 'program stop' latches.
- Signals reset to the controller adapter across the adapter interface.
- Signals the Not Initialized state to the adapters until turned off as a result of IPL.
- Resets the CCU check register.
- Masks program levels 2-5 and level 1 adapterrequests.
- Resets the 'interrupt entered' latches.
- Resets all CCU interrupt requests.
- Sets the 'test mode' latch.

Set Address/Display Push Button
This push button can be used to cause the contents of a register or a storage location to be moved to display register 2 for the purpose of displaying it on the control panel.

This push button must be used before the first Store push button operation in order to provide the address of the register or storage location into which data is to be stored.

The Set Address/Display push button is effective only when the Display/Function Select switch is in the STORAGE ADDRESS or REGISTER ADDRESS position. It is not effective during IPL Phase 1 or IPL Phase 2 or when the Diagnostic Control switch is in one of the storage test positions, and the START push button has been pressed. It is also effective only when the Panel Active light is *on*.

Whether a register or a storage location is addressed in a Set Address/Display operation depends upon whether the Display/Function Select switch is in the REGISTER ADDRESS or STORAGE ADDRESS position.

Set Address and Display Register: If the Display/Function Select switch is in the REGISTER ADDRESS position, pressing and releasing the set Address/Display push button loads the contents of Address/Data switches B and D into byte 0, bits 0-3, and byte 1, bits 0-3, of display register 1.

The registers that can be displayed are the same registers addressable via input and output instructions. Since only 128 register addresses are possible, the high-order register address bit, byte 0, bit 0, is ignored. If the register address is that of an assigned input register, the contents of that register are loaded into display register 2. If the register address is an unassigned input address, the contents of display register 2 are set to zero, but the In/Out Check L1 interrupt request is not set.

If the controller is in the program stop state, the register address placed in the operation register from the Address/Data switches B and D, is also placed in the temporary address register. In this case the address placed in the operation register is also used for register addressing if a subsequent STORE operation is performed.

Note: *Most input registers can be displayed by the Set Address/Display push button without disturbing normal program operation. However, certain input registers (Input X'53', X'54', and X'7B') also perform control functions and therefore cannot be displayed without affecting normal operation. The sections describing the individual input registers should be referred to for more detailed information.*

Set Address and Display Storage: If the Display/Function Select switch is in the STORAGE ADDRESS position, pressing and releasing the Set Address/Display push button loads the contents of Address/Data switches A-E into the storage address register and into display register 1.

The contents of the addressed storage halfword are loaded into byte 0 and byte 1 of display register 2, and byte X of display register 2 is reset.

If the storage address in the Address/Data switches is valid for the machine configuration, the Address Exception L1 interrupt request is not set. Instead, all zeroes are loaded into display register 2.

If the controller is in the program stop state, the storage address in the Address/Data switches is also gated to the temporary address register for use as the initial storage address in a subsequent Store operation.

Note: *Storage locations may be displayed by the Set/Address Display push button without affecting normal program or cycle steal operation.*

Store Push Button
If the controller is in the program stop state, this push button can be used to store data from the Address/Data switches into a storage location or into a register.

Whether a register or a storage location is addressed in a Store operation depends upon whether the Display/Function Select switch is in the REGISTER ADDRESS or STORAGE ADDRESS position.

The Store push button is effective only when the Display/Function Select switch is in the STORAGE ADDRESS or REGISTER ADDRESS position and the controller is in the program stop state. It is not effective during IPL phase 1 or 2 or when the Diagnostic Control switch is in one of the storage test posi-

tions. It is also effective only when the Panel Active light is *on*.

The first Store push button operation performed must be preceded by a Set Address/Display operation, which provides the address of the register or storage location into which data is to be stored. If the Display/Function Select switch is in the REGISTER ADDRESS position, each Store push button operation must be preceded by setting the desired register address in switches B and D and pressing Set Address/Display. If the Display/Function Select switch is in the STORAGE ADDRESS position, data can be stored into sequential storage locations without intervening Set Address/Display operations. The storage address placed in the temporary address register by the first Set Address/Display operation is automatically incremented with each Store.

Load/Store Compare Switch
This switch is used in conjunction with the address compare positions of the Mode Select switch and the Address Compare light.

Load Compare: When the switch is in this position, the storage address placed in Address/Data switches A-E is compared with the contents of the storage address register. This occurs during each instruction cycle. If an address match is detected, the Address Compare light in Display B turns *on*. Also, if the Mode Select switch is in the ADDRESS COMPARE PROGRAM STOP or ADDRESS COMPARE INTERRUPT position, action is taken as described for those switch settings.

The LOAD COMPARE switch setting allows the operator to determine if a specific storage location is ever being used to load the operation register with an instruction or to load data or an address into a general register.

In general the following procedures should be followed.

- To determine if an instruction at a specific address is ever executed, the address placed in the Address/Data switches may be the address of byte 0 or byte 1 of the instruction. If the selected byte of the storage location is ever loaded in a general register by ICT, IC, LH, L, LA or BAL, an address compare results.
- To determine if an instruction ever loads a specific byte into byte X, byte 0 and byte 1 of a general register, the address placed in the Address/Data switches must be the address of the specific byte. If that byte is ever loaded into byte 0 or byte 1 of the operation register (part of an instruction), an address compare results.
- To determine if a specific halfword is ever loaded into byte 0 and byte 1 of a general register by LA, BAL, L, or LH, the address placed in the Address/Data switches may be the address of byte 0 or byte 1 of the halfword. If the selected byte is ever loaded into (1) byte 0 or byte 1 of a general register by an IC or ICT, (2) byte X of a general register by a L, or (3) byte 0 or byte 1 of the operation register (part of an instruction), an address compare results.

Storage Compare: When the switch is in this position, thee storage address placed in the Address/Data switches A-E is compared with the contents of the storage address register. This occurs during each instruction cycle that causes data in a general register to be stored in a storage location. If a match is detected, the Address Compare light in Display B turns *on*. Also, if the Mode Select switch is in the ADDRESS COMPARE PROGRAM STOP or ADDRESS COMPARE INTERRUPT position, action is taken as described for those switch settings.

In general the following procedures should be followed.

- To determine if an instruction ever stores data from a general register into a specific byte, the address of that byte must be placed in the Address/Data switches.
- To determine if a ST or STH instruction stores both byte 0 and byte 1 of a general register into a storage halfword, the address of either byte may be placed in the Address/Data switches. If either byte of the general register is stored into the specified byte in storage, an address compare results. (N Character times)

Appendix A: External Register Addresses

Input Register Addresses

E Field	Register/Function	
00	Gen Reg, Group 0	Reg 0
01		Reg 1
02		Reg 2
03		Reg 3
04		Reg 4
05		Reg 5
06		Reg 6
07		Reg 7
08	Gen Reg, Group 1	Reg 0
09		Reg 1
0A		Reg 2
0B		Reg 3
0C		Reg 4
0D		Reg 5
0E		Reg 6
0F		Reg 7
10	Gen Reg, Group 2	Reg 0
11		Reg 1
12		Reg 2
13		Reg 3
14		Reg 4
15		Reg 5
16		Reg 6
17		Reg 7
18	Gen Reg, Group 3	Reg 0
19		Reg 1
1A		Reg 2
1B		Reg 3
1C		Reg 4
1D		Reg 5
1E		Reg 6
1F		Reg 7
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
2A		
2B		
2C		
2D		
2E		
2F		
30	See Note 1	
31		
32		
33		
34		
35		
36		
37		
38		
39		
3A		
3B		
3C		
3D		
3E		
3F		

E Field	Register/Function	
	Type 2 CS	**Type 1 CS**
40	Interface Address	Unused
41	Unused	Interface Address
42	Unused	CNTL A
43	Check Register	CNTL B/C
44	ICW Input Reg 0-15	Status
45	ICW Input Reg 16-31	Unused
46	Display Register	Unused
47	ICW Input Reg 32-45	Unused
48		
49		
4A		
4B		
4C	See Note 1	
4D		
4E		
4F		
50	INCWAR	
51	OUTCWAR	
52	Control Word Byte Count	
53	Sense Register	
54	Status Register	
55	Control Register	
56	Check Regsiter	
57	See Note 2	Type 2 CA
58	Channel Bus Out Diag Reg	
59	Cycle Steal Address Reg	
5A	Data Buffer	
5B	Tag Diagnostic Reg	
5C	Command Register	
5D	See Note 2	
5E	See Note 2	
5F	See Note 2	
60	Initial Selection Control	
61	Initial Selection Address & Command	
62	Data/Status Control	
63	Address and ESC Status	Type 1 CA
64	Data Buffer Bytes 1,2	
65	Data Buffer Bytes 3,4	
66	NSC Status Byte	
67	Controls	
68		
69		
6A		
6B	See Note 1	
6C		
6D		
6E		
6F		
70	Storage Size Installed	
71	Panel Address/Data Digits	
72	Panel Function Select Controls	
73	Insert Key	
74	LAR	
75	See Note 2	
76	Adapter Interrupt Request Group 1	
77	Adapter Interrupt Request Group 2	
78	See Note 2	
79	Utility	
7A	See Note 2	
7B	New BSC CRC	
7C	See Note 2	
7D	CCU Check Register	
7E	CCU Interrupt Requests Group 1	
7F	CCU Interrupt Requests Group 2	

Note 1:
A constant of all zeros is loaded into R and the Input/Output check L1 request is set.

Note 2:
A constant of all zeros is loaded into R.

Output Register Addresses

E Field	Register/Function	
00	Gen Reg, Group 0	Reg 0
01		Reg 1
02		Reg 2
03		Reg 3
04		Reg 4
05		Reg 5
06		Reg 6
07		Reg 7
08	Gen Reg, Group 1	Reg 0
09		Reg 1
0A		Reg 2
0B		Reg 3
0C		Reg 4
0D		Reg 5
0E		Reg 6
0F		Reg 7
10	Gen Reg, Group 2	Reg 0
11		Reg 1
12		Reg 2
13		Reg 3
14		Reg 4
15		Reg 5
16		Reg 6
17		Reg 7
18	Gen Reg, Group 3	Reg 0
19		Reg 1
1A		Reg 2
1B		Reg 3
1C		Reg 4
1D		Reg 5
1E		Reg 6
1F		Reg 7
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
2A		
2B		
2C		
2D		
2E		
2F		
30	Set Note 1	
31		
32		
33		
34		
35		
36		
37		
38		
39		
3A		
3B		
3C		
3D		
3E		
3F		

E Field	Register/Function	
	Type 2 Scanner	**Type 1 Scanner**
40	Interface Address	Set Mode Bit Override
41	Address Substitution Control	Start Scanner & Reset Bit L2
42	Upper Scan Limit Control	CNTL A
43	Control	CNTL B
44	ICW 0-15	General Control
45	ICW 16-23	Scanner Control
46	ICW 24-33,44	Set Char Service Pending
47	ICW 34-43	Force L2 Bit Request
48		
49		
4A		
4B		
4C	See Note 1	
4D		
4E		
4F		
50	INCWAR	
51	OUTCWAR	
52	See Note 2	
53	Set Sense Register Bits	
54	Set Status Register Bits	
55	Set Control Register Bits	
56	Reset Control Register Bits	
57	Channel Adapter Mode Reg	/Type 2 CA
58	Channel Bus Out Diag Reg	
59	See Note 2	
5A	Channel Adapter Data Buffer	
5B	Tag Diagnostic Reg	
5C	See Note 2	
5D	See Note 2	
5E	See Note 2	
5F	See Note 2	
60	Reset Initial Selection	
61	Unused	
62	Data/Status Control	
63	Address and ESC Status	
64	Data Buffer Bytes 1,2	Type 1 CA
65	Data Buffer Bytes 3,4	
66	NSC Status Byte	
67	Control	
68		
69		
6A		
6B		
6C	See Note 1	
6D		
6E		
6F		
70	Hardstop	
71	Display Reg 1	
72	Display Reg 2	
73	Set Key	
74	See Note 2	
75	See Note 2	
76	See Note 2	
77	Miscellaneous Control	
78	Force CCU Checks	
79	Utility	
7A	See Note 2	
7B	See Note 2	
7C	Set PCI Level 3	
7D	Set PCI Level 4	
7E	Set Mask Bits	
7F	Reset Mask Bits	

Note 1:
The bits of R are ignored and the Input/Output check L1 request is set.

Note 2:
The bits of R are ignored.

Appendix B: Input/Output Instruction Bit Definitions

Type 1 Scanner Input Instructions

Input X'41' (Interface Address)
When operating in the bit service mode, this instruction causes a storage address associated with an interface to be loaded into the specified general register. The selected interface is where the scanner has stopped because of an interrupt request. The storage address is a fixed location, determined by the LIB position and the interface address assigned to the line causing the interrupt. Figure 12 shows the storage addresses associated with each interface address.

When operating in the character service mode, a fixed address of X'06F0' is loaded into the specified register.

Input X'42' (Control A)
The Input X'42' instruction should be executed only when the scanner is stopped on an interface. This input instruction checks the state of the Output X'42' (control A) instruction. When executed for a particular line, the result in the register specified by the R operand is a bit-for-bit reflection of the control information set by the Output X'42' instruction for that interface. Refer to the Output X'42' instruction for the definition of each bit position.

Input X'43' (Control B/C)
This instruction cannot be executed immediately following an Output X'43' for feedback checking. However, it can be executed after an Output X'43' to obtain other status indications.

Byte 0, bit 0: Receive Data Bit Buffer - This position indicates a mark or space condition received from the line interface (1 = a mark, 0 = a space).

Byte 0, bit 1: Feedback Check - This bit is set on if a bit to be transmitted is not transferred to the interface correctly or if the interface bit service request fails to reset.

Programming Note
After a feedback check occurs, the bit-service interrupt request cannot be reset and the scanner cannot be restarted until the control program has reset the check indication with an Output X'44' with byte 1, bit 6 set to 1.

Byte 0, bit 2: Interface Check Summary - This bit is set on to indicate (1) the detection of a feedback error, (2) bit overrun/underrun, or (3) 'data set ready' line not active. It is a summary of these check conditions and is set to 1 when byte 0, bit 1, or byte 1, bit 2 or bit 7 of this input is on.

Byte 0, bit 3: Received Data Lead - This bit indicates the level of the 'receive data' signal from the modem. A 1 indicates a space level and 0 indicates a mark level.

Byte 0, bit 4: Transmit Mode - A 1 in this position indicates that the line interface is set for transmit mode; a 0 indicates that it is set for receive mode.

Byte 0, bit 5: New Sync - This position indicates the state of the 'new sync' line to the modem. A 1 indicates that the line is active. A 0 indicates that the line is inactive.

Byte 0, bit 6: Request to Send (RTS) - A 1 in this position indicates that the 'request to send' line to the modem is active. A 0 indicates that the line is inactive.

Byte 0, bit 7: Send Data Bit Buffer - This position shows the setting of the line interface send data buffer. A 1 indicates a mark, and a 0 indicates a space.

Byte 1, bit 0: Not Clear to Send - A 1 in this position indicates that the 'clear to send' line from the modem is *inactive*. A 0 indicates that it is active.

Byte 1, bit 1: Ring Indicator - If this bit is 1, the 'ring indicator' line from the modem is active. A 0 indicates that it is inactive.

Byte 1, bit 2: Not Data Set Ready - When this position is A 1, the 'data set ready' line from the modem is *inactive*. A 0 indicates that it is active.

Byte 1, bit 3: Received Line Signal Detector - When the 'receive line signal detector' line from the modem is active, this bit is 1. A 0 indicates that the line is inactive.

Byte 1, bit 4: Telegraph Interface Echo Check - If this bit position is 1, a TTY echo check has occurred.

Byte 1, bit 5: Diagnostic Mode - A 1 in this position indicates that the line interface is set for the diagnostic wrap mode, and a 0 indicates that it is set for the normal read/write mode. See Type 1 Scanner

Diagnostic Wrap in Chapter 6. If this bit is a 1, then byte 1, bit 2 of this input is forced to 0 and byte 1, bit 3 of this input is forced to a 1. If this bit is 1 and byte 0, bit 6 is 1, byte 1, bit 0 is forced to 0.

Byte 1, bit 6: Bit Service - A 1 in this position indicates that the scanner has requested a program level 2 bit service interrupt.

Byte 1, bit 7: Bit Overrun/Underrun - A 1 in this position indicates that a bit overrun or underrun has occurred.

Autocall Interface Bits for Input X'43'

Byte 0, bit 0: This bit is 0.

Byte 0, bits 1-2: These bits are the same as for line interface.

Byte 0, bit 3: Digit Present (DPR) - A 1 in this position indicates that the 'digit present' signal to the autocall unit is active; a 0 indicates that the signal is inactive.

Byte 0, bits 4-7: These bit positions represent the autocall dial digit in BCD form.

Byte 1, bit 0: Not Abandon Call and Retry - A 1 in this position indicates that the 'abandon call and retry' signal from the autocall unit is inactive. A 0 indicates that it is active.

Byte 1, bit 1: Present Next Digit (PND) - A 1 in this position indicates that the 'present next digit' signal from the autocall unit is active; a 0 indicates that the signal is inactive.

Byte 1, bit 2: Not Data Line Occupied (Not DLO) - A 1 in this position indicates that the 'data line occupied' signal from the autocall unit is *inactive*. A 0 indicates that it is active.

Byte 1, bit 3: Power Indicator (PWI) - A 1 in this position indicates that the 'power indicator' signal from the autocall unit is active; a 0 indicates that the signal is inactive.

Byte 1, bit 4: Call Request (CRQ) - A 1 in this position indicates that the 'call request' signal to the autocall unit is active; a 0 indicates that the signal is inactive.

Byte 1, bit 5: Call Originating Status (COS) - A 1 in this position indicates that the 'call originating status' signal from the autocall unit is active; a 0 indicates that the signal is inactive.

Byte 1, bits 6-7: These bits are the same as for a line interface operation.

Input X'44' (Status)

Byte 0, bit 0: Mode Bit Override - When this position is a 1 the 'mode bit override' latch is set. When this latch is set, all interface mode settings except 01 with high-priority are overridden. See *Interface Modes of Operation* in Chapter 6.

Byte 0, bit 1: This bit is 0.

Byte 0, bit 2: Override Remember - When this position is a 1, the 'override remember' latch is set. See *Interface Modes of Operation* in Chapter 6.

Byte 0, bit 3: Scanner Enabled - When the scanner is enabled, this bit is 1. The bit is 0 when the scanner is disabled.

Byte 0, bit 4: Character Service Pending - This position indicates that the program has a character service request pending.

Byte 0, bits 5-7: These bits are 0.

Byte 1, bits 0-1: These bits are 0.

Byte 1, bit 2: LIB 1 Bit Clock Check - This bit is a 1 if a LIB-1 bit clock parity check has occurred.

Byte 1, bit 3: LIB 2 Bit Clock Check - This bit is a 1 if a LIB-2 bit clock parity check has occurred.

Byte 1, bit 4: LIB 3 Bit Clock Check - This bit is a 1 if a LIB-3 bit clock parity check has occurred.

Byte 1, bit 5: LIB 4 Bit Clock Check - This bit is a 1 if a LIB-4 bit clock parity check has occurred.

Byte 1, bit 6: LIB Select Check - This bit is a 1 if more than one LIB or more than one interface is addressed at the same time.

Byte 1, bit 7: CCU Outbus Check - This bit is A 1 when a CCU outbus parity check has been detected.

Type 1 Scanner Output Instructions

Output X'40' (Set Mode Bit Override and Override Remember)

This output is used to set the 'mode bit override' and the 'override remember' latches. Setting these latches causes the scanner to override all interface mode settings except 01 with high-priority. This instruction performs a function, and therefore, the bit settings of the register specified by the R operand are ignored. See *Interface Modes of Operation* in Chapter 6.

Output X'41' (Start Scanner & Reset Bit Service L2 Request)

The Output X'41' instruction should be executed only when the scanner is stopped on an interface. An Input X'41' instruction *must* precede execution of an Output X'41'. Otherwise, the output instruction is ignored.

This instruction starts the scanner at the completion of the line interface servicing and resets the bit service request for the interface the scanner is addressing. It also resets the program level 2 bit service interrupt request. This instruction performs a function, and therefore the bit settings of the register specified by the R operand are ignored.

Output X'42' (Control A)

The Output X'42' instruction should be executed only when the scanner is stopped on an interface.

Byte 0, bits 0-5: These bits are unused.

Byte 0, bits 6-7: Mode bits - Each interface may be set to one of four level 2 interrupt modes as follows:

00 Disable level 2 interrupts - This mode disables all level 2 interrupts for a given interface.

01 Monitor for ring indicator or data set ready - This mode allows level 2 interrupts if 'ring indicator' or 'data set ready' becomes active for at least one bit time.

10 Monitor for receive data space - This mode allows a level 2 interrupt each time a space bit is received. An interrupt also occurs if 'data set ready' drops for at least one bit time.

11 Enable level 2 interrupts - This mode allows all level 2 interrupts to be indicates for a given interface.

For further discussion on the use of the mode bit setting, refer to *Interface Modes of Operation* in Chapter 6.

Byte 1, bit 0: Bit Service Priority - This bit determines the service priority for the interface the scanner is addressing. A 1 in this position sets a low priority; a 0 sets a high priority.

Byte 1, bit 1: Diagnostic Mode - A 1 in this position sets the line interface to the diagnostic mode, and a 0 sets it to the normal read/write mode. See *Diagnostic Wrap* and *Modem Self Test* in Chapter 6 for the description of operation in this mode.

Byte 1, bit 2: Data Terminal Ready (DTR) - This bit activates or deactivates the 'data terminal ready' line from the line interface to the modem. A 1 activates the line, and a 0 deactivates it.

Byte 1, bit 3: Synchronous Bit Clock - A 1 in this position sets the line interface for synchronous clocking, and a 0 sets it for start-stop clocking.

Byte 1, bit 4: External Clock - This bit position selects modem (data set) clocking, or business machine (data terminal equipment) clocking. 1 = modem, and 0 = business machine.

Byte 1, bit 5: Data Rate Selector - A 1 in this position selects the high data rate for the attached modem, and a 0 selects the low data rate.

Byte 1, bits 6-7: Oscillator Select Bits - These two bit positions select which one of the four *internal bit rates* available in the Type 1 Scanner is to be assigned to the line interface.

The internal bit rates available are determined by the bit clocks that have been installed in the Type 1 Scanner in conjunction with the line interface base types and line sets. See the description of *Business Machine Clocks* in Chapter 6.

Autocall Interface Bits for Output X'42'

The Output X'42' instruction for autocall is the same as for *line interface* except that byte 1, bits 1-7 are ignored.

Note: *The lowest speed oscillator is always assigned to autocall interfaces.*

Output X'43' (Control B)
The Output X'43' instruction should be executed only when the scanner is stopped on an interface.

Byte 0, bits 0-7: These bits are unused.

Byte 1, bits 0-3: These bits are unused.

Byte 1, bit 4: Transmit/Receive Mode - A 1 in this position sets the line interface to transmit mode; a 0 sets it to receive mode.

Byte 1, bit 5: New Sync - A 1 in this position activates the 'new sync' line to the modem; a 0 deactivates the line.

Byte 1, bit 6: Request to Send (RTS) - A 1 in this position activates the 'request to send' line to the modem; a 0 deactivates the line.

Byte 1, bit 7: Send Data - A 1 in this position sends a mark to the line interface 'send data buffer'; a 0 sends a space.

Autocall Interface Bits for Output X'43'

Byte 0, bits 0-7: These bits are unused.

Byte 1, bits 0-1: These bits are unused.

Byte 1, bit 2: CRQ - A 1 in this position activates the 'call request' line to the autocall interface; a 0 deactivates the line.

Byte 1, bit 3: DPR - A 1 in this position activates the 'digit present' line to the autocall interface; a 0 deactivates the line.

Byte 1, bits 4-7: These bit positions represent the dial digit in BCD form.

Output X'44' (General Control)

Byte 0, bits 0-7: These bits are unused.

Byte 1, bit 0: Diagnostic Bit Service - A 1 in this position sets a latch that causes continuous bit service requests for all 64 line interfaces (used and unused). A 0 resets the latch.

Byte 1, bit 1: This bit is unused.

Byte 1, bit 2: Reset Mode Bit Override - A 1 in this position resets the 'mode bit override' latch as described in the *Interface Modes of Operation* in Chapter 6.

Byte 1, bit 3: Reset Override Remember - A 1 in this position resets the 'override remember' latch in the Type 1 Scanner. See *Interface Modes of Operation* in Chapter 6.

Byte 1, bit 4: Reset Character Service Pending - A 1 in this position resets the 'character service pending' latch set by Output X'46'.

Byte 1, bit 5: Reset Level 1 Checks - A 1 in this position resets all level 1 check conditions resulting from the Type 1 Scanner.

Byte 1, bit 6: Reset Feedback Check - A 1 in this position resets the 'feedback error' latch.

Byte 1, bit 7: Reset bit Overrun/Underrun - A 1 in this position resets the 'bit overrun/underrun' latch for the interface the scanner is addressing.

Output X'45' (Scanner Control)
This instruction may be used to disable one or more LIBs. A LIB should be disabled only when it is causing solid errors or solid level 1 interrupts.

Byte 0, bit 0: This bit is unused.

Byte 0, bit 1: Set Scanner Enable - A 1 in this position sets the 'scanner enable' latch, allowing normal operation of the scanner.

Byte 0, bit 2: Reset Scanner Enable - A 1 in this position resets the 'scanner enable' latch. This disables the scanner, prevents any further interrupts, forces all interfaces to be reset, and allows no bit serviices.

A 1 in both bit 1 and bit 2 of this byte is invalid, and the result is unpredictable. The scanner may be either enabled or disabled.

Byte 0, bit 3: This bit is unused.

Byte 0, bit 4: Disable LIB 1 - A 1 in this position disables L1 and L2 interrupts and bit service from LIB 1. A 0 resets the disable condition.

Byte 0, bit 5: Disable LIB 2 - A 1 in this position disables L1 and L2 interrupts and bit service from LIB 2. A 0 resets the disable condition.

Byte 0, bit 6: Disable LIB 3 - A 1 in this position disables L1 and L2 interrupts and bit service from LIB

3. A 0 resets the disable condition.

Byte 0, bit 7: Disable LIB 4 - A 1 in this position disables L1 and L2 interrupts and bit service from LIB 4. A 0 resets the disable condition.

Byte 1, bits 0-7: These bits are unused.

Output X'46' (Set Character Service)
The Output X'46' instruction sets the 'character service pending' latch. It also starts the scanner and resets the level 2 bit service interrupt request. This instruction performs a function, and therefore, the bit settings of the register specified by the R operand are ignored. An Input X'41' instruction *must* precede execution of an Output X'46'. Otherwise, the output instruction is ignored.

Output X'47' (Force Bit Service Request)
This instruction is executed to force one level 2 bit service interrupt request. The storage address associated with the interface that is to cause the interrupt must be loaded into the register specified by the R operand. When this instruction is executed, the scanner determines the interface address from the storage address and requests a level 2 interrupt for that interface.

Programming Note
Forced bit service cannot be stacked. If an Output X'47' is executed before the previous Output X'47' has been serviced, the second address overlays the first address.

Byte 0, bits 0-5: These bits are unused.

Byte 0, bits 6-7, Byte 1, bits 0-3: Interface Address - These bit positions are loaded with the storage address associated with the interface that is to cause a bit service interrupt.

Byte 1, bits 4-7: These bits are unused.

Type 2 Scanner Input Instructions

Input X'40' (Interface Address)
This instruction obtains the line interface address from the ABAR in the attachment base. Conditions that set the ABAR are described in the *I/O Programming Considerations* section in Chapter 7. When this instruction is executed, the interface address from the ABAR is placed in byte 0, bit 6 through byte 1, bit 6 of the register specified by the R operand. Byte 0, bit 4 is always set to 1. The other register bit positions are set to zero.

Input X'43' (Check Register)
This instruction obtains the status of the check register in the scanner. Since there can be up to four Type 2 scanners, the check register selected is determined by the interface address in the ABAR at the time of instruction execution.

When this instruction is executed, the check register bits are placed in the register specified by the R field.

Programming Note
If any of the check register bits in the scanner are set to 1, the Type 2 Scanner L1 interrupt request is set.

Byte 0, bit 0: LIB 1 Bit Clock Check - This bit is set to 1 if a LIB position 1 bit clock control check is detected by the Type 2 Scanner; otherwise, it is set to 0.

Byte 0, bit 1: LIB 2 Bit Clock Check - Same as above.

Byte 0, bit 2: LIB 3 Bit Clock Check - Same as above.

Byte 0, bit 3: LIB 4 Bit Clock Check - Same as above.

Byte 0, bit 4: LIB 5 Bit Clock Check - Same as above.

Byte 0, bit 5: LIB 6 Bit Clock Check - Same as above.

Byte 0, bit 6: LIB Select Check - This bit is set to 1 if the Type 2 Scanner has detected a LIB address parity check on either LIB string 1 (LIB pos 1, 2, or 3) or LIB string 2 (LIB pos 4, 5, or 6); otherwise, it is set to 0.

Byte 0, bit 7: ICW Input Register Check - This bit is set to 1 if the Type 2 Scanner has detected a parity error in the ICW input register; otherwise, it is set to 0.

Byte 1, bit 0: ICW Work Register Check - This bit is set to 1 if the Type 2 Scanner has detected a parity error in the ICW work register; otherwise, it is set to 0.

Byte 1, bit 1: Priority Register Available Check - This bit is set to 1 if the Type 2 Scanner has

Priority Register Available Check - This bit is set to 1 if the Type 2 Scanner has detected even parity on one of the four *priority register available* lines; otherwise, it is set to 0. These four lines from the attachment base are parity-checked in the Type 2 Scanner.

Byte 1, bit 2: CCU Outbus Check - This bit is set to 1 if the Type 2 Scanner has detected even parity on the CCU Outbus. Otherwise, it is set to 0.

Byte 1, bit 3: Line Address Bus Check - This bit is set to 1 if the Type 2 Scanner has detected a scanner buffer address register bus parity error during program addressing (refer to discussion on *Program Addressing* in Chapter 7); otherwise, it is set to 0.

Byte 1, bits 4-7: These bits are 0.

Input X'44' (ICW Input Register - Bits 0-15)
This instruction may be used to determine the state of the 'secondary control field' (SCF) and the 'parallel data field' (PDF) in the ICW that is set in the ICW input register. The interface address in the ABAR selects the proper scanner. When this instruction is executed, the contents of the ICW input register, bit positions 0-15, are placed in the register specified by the R field. Refer to *I/O Programming Considerations* for conditions that set the ICW input register. The *Interface Control Word Format* section of Chapter 7 describes the SCF and PDF fields and their bit definitions.

Input X'45' (ICW Input Register - Bits 16-31)
This instruction may be used to determine the state of the LCD and PCF fields, and SDF bits 0-7 in the ICW that is set in the ICW input register. The interface address in the ABAR selects the proper scanner. When this instruction is executed, the contents of the ICW input register bit positions 16-31 are placed in the register specified by the R field. Refer to *I/O*

Programming Considerations in Chapter 7 for conditions that set the ICW input register. The LCD, PCF, and SDF fields and bit definitions are described in the *Interface Control Word Format* section of Chapter 7.

Programming Note
This input does not contain the complete serial data field. Only bits 0-7 of the field are available with this input. The remainder of the SDF (bits 8-9) are accessed by the Input X'47' instruction.

Input X'46' (Display Register)
This instruction may be used to determine the state of the display register in the Type 2 Scanner selected by the interface address in ABAR. When this instruction is executed, the contents of the Type 2 Scanner display register are placed in the register specified by the R field.

The hardware, because of the *display request* (ICW bit 38), can cause status information for a particular interface to be placed into the Type 2 Scanner display register when the interface is scanned. Input X'46' can then be used to examine this status information. Refer to Output X'43', which must be used to set/reset the display request bit in the ICW.

Before accessing the display register with an Input X'46', the program must ensure that enough time has elapsed to guarantee that the interface has been scanned at least once after it set the display request bit in the ICW.

Byte 0, bit 0: Clear To Send - This bit is set to 1 if the 'clear to send' line from the modem is on, or if the diagnostic wrap forces 'clear to send' on; otherwise, it is set to 0.

Byte 0, bit 1: Ring Indicator - This bit is set to 1 if the 'ring indicator' line from the modem is on; otherwise, it is set to 0.

Byte 0, bit 2: Data Set Ready - This bit is set to 1 if the 'data set ready' line from the modem is on, or if the diagnostic wrap forces data set ready on; otherwise, it is set to 0.

Byte 0, bit 3: Receive Line Signal Detector - This bit is set to 1 if the 'receive line signal detected' line from the modem is on, or if the diagnostic wrap state forces 'receive line signal detected' on; otherwise, it is set to 0.

Byte 0, bit 4: Receive Data Bit Buffer - This bit is set to 1 if the line interface receive data buffer contains a mark (1). If the buffer contains a space (0), this bit is set to 0.

Byte 0, bit 5: Diagnostic Wrap Mode - This bit is set to 1 if the line interface is in diagnostic wrap state; otherwise, it is set to 0.

Byte 0, bit 6: Bit Service Request - This bit is set to 1 if the line interface 'bit service request' is on; otherwise, it is set to 0. Normally, bit service must be on before the Type 2 Scanner can access the line associated with the interface address or initiate transmit or receive operations.

Byte 0, bit 7: This bit is 0.

Byte 1, bits 0-7: These bits are 0.

Autocall Interface Bits for Input X'46'

Byte 0, bit 0: Abandon Call and Retry - This bit is set to 1 if the autocall unit 'abandon call and retry' (ACR) is active; otherwise it is set to 0.

Byte 0, bit 1: Present Next Digit - This bit is set to 1 if the autocall unit 'present next digit' (PND) is active; otherwise, it is set to 0.

Byte 0, bit 2: Data Line Occupied - This bit is set to 1 if the autocall unit 'data line occupied' (DLO) is active; otherwise, it is set to 0.

Byte 0, bit 3: Power Indicator - This bit is set to 1 if the autocall unit 'power indicator' (PWI) is active; otherwise, it is set to 0.

Byte 0, bit 4: This bit is 0.

Byte 0, bit 5: Call Originating Status - This bit is set to 1 if the autocall unit 'call originating status' (COS) is active; otherwise, it is set to 0.

Byte 0, bit 6: Bit Service Request - This bit is the same as for line interface.

Byte 0, bit 7: This bit is 0.

Byte 1, bits 0-7: These bits are 0.

Input X'47' (ICW Input Register - Bits 32-45)
This instruction may be used to determine the state of SDF bits 8-9, the display request bit, the L2 interrupt pending bit, and priority bits 1-2. The interface address in the ABAR selects the proper scanner and associated ICW. When this instruction is executed, the contents of the ICW input register bit positions 32 through 45 are placed in the register specified by the R operand. Byte 0, bits 2-5 and 7, and byte 1, bits 0 and 6-7, are set to 0. See *I/O Programming Considerations* in Chapter 7 for conditions that cause the ICW input register to be set. For an interpretation of these bits, see the *Interface Control Word Format* section in Chapter 7.

Type 2 Scanner Output Instructions

Output X'40' (Interface Address)
This instruction may be used to set an interface address in the attachment buffer address register (ABAR) of the Type 2 Attachment Base. When this instruction is executed, byte 0, bit 6 through byte 1, bit 6, in the register specified by the R operand are placed in the ABAR.

The interface address, placed in ABAR, selects the Type 2 Scanner and the ICW associated with that address. When accessed, the ICW is placed in the ICW work register by the scanner hardware. If Output X'40' is executed in program levels 3 or 4, the contents of the ICW work register are placed in the ICW input register where they are available for access by Inputs X'44', X'45', and X'47'.

Output X'41' (Address Substitution Control)
This instruction must be used to set the substitution control register in the Type 2 Attachment Base. See *Address Substitution* in Chapter 7 for a description and coding of the substitution control bits.

Execution of Output X'41' causes byte 1, bits 2-5 from the register specified by R to be placed into the substitution control register.

Programming Note
If address substitution is not used, Output X'41' *must* be executed with byte 1, bits 2-5 off in the register specified by R.

Output X'42' (Scan Limit Control)
This instruction must be used to set the 'scan limit' in the selected Type 2 Scanner. At least one Output X'42' must be executed for each Type 2 Scanner available. The scanner selected is determined by the interface address in the attachment buffer address register (ABAR) of the attachment base at the time of execution.

When this instruction is executed, byte 1, bits 6 and 7, in the register specified by the R operand are placed in the 'scan limit' latches of the scanner.

Byte 0, bits 0-7: These bits are unused.

Byte 1, bits 0-5: These bits are unused.

Byte 1, bits 6-7: Scan Limit Select Bits 0 and 1 - These bits are set as follows to indicate the desired scan limit for each Type 2 Scanner.

Bit 6 7	*Scan Limit*
0 1	8
1 1	16
1 0	48
0 0	96

Output X'43' (Control)
This instruction may be executed to set or reset various control functions in a Type 2 Scanner. The Type 2 Scanner is selected by the interface address in the attachment buffer address register (ABAR) of the attachment base. When this instruction is executed, the bit configuration in the register specified by the R field determines which control functions are set or reset.

Byte 0, bit 0: Set Function - A 1 in this position causes the functions of byte 0, bits 2-7, and byte 1, bits 0-7, of this output to be set when the corresponding bit is 1. This bit should not be 1 if byte 0, bit 1 is 1.

Byte 0, bit 1: Reset Function - A 1 in this position causes the functions of byte 0, bits 2-7 and byte 1, bits 0-7 of this output to be reset when the corresponding bit is 1. This bit should not be 1 if byte 0, bit 0, is 1.

Byte 0, bit 2: Display Request - A 1 in this position causes the display request (ICW bit 38) to be set or reset according to byte 0, bits 0 and 1.

Byte 0, bits 3-6: These bits are unused.

Byte 0, bit 7: Disable LIB Position 1 - A 1 in this position causes LIB position 1 of the addressed scanner to be disabled or enabled. When this bit is on and byte 0, bit 0 (set function) is on, the LIB position is

disabled. When this bit is on and byte 0, bit 1 (reset function) is on, the LIB position is enabled.

Byte 1, bit 0: Disable LIB Position 2 - This bit is associated with LIB position 2 and functions the same as byte 0, bit 7.

Byte 1, bit 1: Disable LIB Position 3 - This bit is associated with LIB position 3 and functions the same as byte 0, bit 7.

Byte 1, bit 2: Disable LIB Position 4 - This bit is associated with LIB position 4 and functions the same as byte 0, bit 7.

Byte 1, bit 3: Disable LIB Position 5 - This bit is associated with LIB position 5 and functions the same as byte 0, bit 7. This bit should be set for a Type 2 Scanner-1.

Byte 1, bit 4: Disable LIB Position 6 - This bit is associated with LIB position 6 and functions the same as byte 0, bit 7. This bit should be set for a Type 2 Scanner-1.

Byte 1, bit 5: Type 2 Scanner Level 1 Request - This bit is for diagnostic purposes. A 1 in this position along with the set function (byte 0, bit 0) causes a level 1 interrupt request and sets the following check latches.

LIB positions 1-6 Bit Clock Checks.
LIB Select Check.
ICW Input Register check.
ICW Work Register check.
Priority Register Available check.
CCU Outbus check.
LINEADDBUS check.

A 1 in this position along with the reset function (byte 0, bit 1) resets the level 1 interrupt request and resets the above check latches.

Byte 1, bit 6: Disable Interrupt Requests - This bit is for diagnostic purposes. A 1 in this position along with the set function (byte 0, bit 0) sets the 'power-on-reset' latch and resets the scanners and LIBs. A 1 in this position along with the reset function (byte 0, bit 1) resets the 'power-on-reset' latch, thereby ending the reset state.

Programming Note

During any 'power-on' sequence, the POR latch is set in each Type 2 Scanner. The program *must* reset this latch with an Output X'43' instruction for each Type 2 Scanner before the scanner can be initialized.

Byte 1, bit 7: This bit is unused.

Output X'44' (ICW Bits 0-15)

This instruction may be used to reset the following secondary control field (SCF) bits in the ICW: stop bit error, service request, character overrun/underrun, and modem error. It is also used to set or reset the program flag, pad flag, and parallel data field (PDF) in the ICW. The PDF field is used as a character buffer. The interface address in the attachment buffer address register (ABAR), located in the Type 2 Attachment Base, selects the Type 2 Scanner and the ICW associated with this address.

When this instruction is executed, the bit configurations in the register specified by the R field determine what bits are to be set, reset, or left unchanged in ICW bits 0-3 and 5-15. Refer to the *Secondary Control Field* of the ICW (in Chapter 7) for a description of SCF bits (byte 0, bits 0-7). See *ICW Format* for the PDF as it relates to various LCD states.

Byte 0, bit 0: When this bit is a 1, ICW bit 0 (stop bit check) is reset to 0; otherwise, it is unchanged.

Byte 0, bit 1: When this bit is a 1, ICW bit 1 (service request) is reset to 0; otherwise, it is unchanged.

Byte 0, bit 2: If this bit is a 1, ICW bit 2 (character overrun/underrun) is reset to 0; otherwise, it is unchanged.

Byte 0, bit 3: If this bit is a 1, ICW bit 3 (modem check) is reset to 0; otherwise, it is unchanged.

Byte 0, bit 4: This bit is unused.

Byte 0, bit 5: This bit is reserved and must be set to zero.

Byte 0, bit 6: If this bit is a 1, ICW bit 6 (program flag) is set to 1; otherwise, it is reset to 0.

Byte 0, bit 7: If this bit is a 1, ICW bit 7 (pad flag) is set to 1; otherwise, it is reset to 0.

Byte 1, bits 0-7: The settings of these bits are placed into the ICW bit positions 8-15 (PDF bits 0-7).

Output X'45' (ICW Bits 16-23)
This instruction may be used to set the bits of the 'line control definer' (LCD) and the 'primary control field' (PCF) in the ICW. When this instruction is executed, byte 1, bits 0-7 in the register specified by the R field are placed in the LCD and PCF fields. The interface address in the ABAR at execution time selects the proper scanner and the associated ICW. For a detailed description of these bits, see the *Interface Control Word Format* section in Chapter 7.

Byte 0, bits 0-7: These bits are unused.

Byte 1, bits 0-3: The settings of these bits are placed into the ICW bit positions 16-19 (LCD bits 0-3).

Byte 1, bits 4-7: The settings of these bits are placed into the ICW bit positions 20-23 (PCF bits 0-3).

Output X'46' (ICW Bits 24-33)
This instruction may be used to set the bits of the 'serial data field' (SDF) in the ICW. When this instruction is executed, byte 0, bits 6-7 and byte 1, bits 0-7 in the register specified by the R field are placed in the SDF of the ICW. The interface address in the ABAR at execution time selects the proper scanner and the associated ICW. For a detailed description of these bits, see the *Interface Control Word Format* section in Chapter 7.

Byte 0, bit 0: The setting of this bit is placed into the ICW bit position 45.

Byte 0, bits 1-5: These bits are unused.

Byte 0, bits 6-7: The settings of these bits are placed into the ICW bit positions 24-25 (SDF bits 0 and 1).

Byte 1, bits 0-7: The settings of these bits are placed into the ICW bit positions 26-33 (SDF bits 2-9).

Output X'47' (ICW Bits 34-37 and 39-43)
This instruction sets the state of ICW bits 34-37 and 39-43. Execution of this instruction places bits from the register specified by R into the appropriate ICW bit positions. The interface address in the ABAR at execution time selects the proper scanner and the associated ICW. For a detailed description of these bits, see the *Interface Control Word Format* section in Chapter 7.

Byte 0, bits 0-5: These bits are unused.

Byte 0, bits 6-7: The settings of these bits are placed into the ICW bit positions 34-35. (These bits are reserved and should be 0.)

Byte 1, bits 0-1: The settings of these bits are placed into the ICW bit positions 36-37. (These bits are reserved and should be 0.)

Byte 1, bit 2: This bit is unused.

Byte 1, bits 3-4: The settings of these bits are placed into the ICW bit positions 39-40. (These bits are reserved and should be 0.)

Byte 1, bit 5: L2 Interrupt Pending - A 1 in this position sets ICW bit 41, which forces an interrupt for a particular interface without requiring the interface to have a service request set. A 0 leaves the ICW bit unchanged.

Byte 1, bits 6-7: The settings of these bits are placed into the ICW bit positions 42-43 (priority bits 1 and 2).

Type 1 CA Input Instructions

Input X'60' (Initial Selection Control)
This input instruction loads the register specified by R with the contents of the initial selection control register. The bits of this register are normally set at the completion of initial selection and identify the cause of a Type 1 CA initial L3 interrupt request.

Byte 0, bit 0: Initial Selection Interrupt - This bit is set by hardware and causes a Type 1 CA initial L3 interrupt request when a Start I/O command is accepted by the adapter and clear initial status is presented to the host channel. This bit is also set when a Test I/O command is received for an emulation subchannel address and a status of X'70' is returned. If this bit is zero, the interrupt request was caused by an unusual condition and can be further defined by the remaining bits of this input. An Output X'62', byte 0, bit 5 or an Output X'60' resets this bit.

Byte 0, bit 1: Interface Disconnect - This bit is set by hardware when an interface disconnect condition (Halt I/O command) is detected during an initial selection sequence. An Output X'62', byte 0, bit 5 or an Output X'60' resets this bit.

Byte 0, bit 2: Selective Reset - This bit is set by hardware when a selective reset condition is detected during an initial selection sequence. An Output X'62', byte 0, bit 5 or an Output X'60' resets this bit.

Byte 0, bit 3: Channel Bus Out Check - This bit is set by hardware when incorrect parity is detected in the channel command byte during initial selection. This causes the Type 1 CA to automatically respond with Unit Check (UC) initial status. An Output X'62', byte 0, bit 5 or an Output X'60' resets this bit.

Byte 0, bit 4: This bit is 0.

Byte 0, bit 5: Stacked Initial Status - This bit is set by hardware when a stacked status condition is detected during initial selection. An Output X'62', byte 0, bit 5 or an Output X'60' resets this bit.

Byte 0, bit 6: NSC Status Byte Cleared - This bit is set to indicate that a status byte for the native mode subchannel (NSC) has been transferred as the initial status byte in an initial selection. Therefore, the NSC status byte has been cleared, and this resulted in the setting of the Type 1 CA initial level 3 interrupt request. An Output X'62', byte 0, bit 5 and Output X'60' resets this bit.

Byte 0, bit 7: System Reset - This bit is set by hardware and causes a level 3 initial selection interrupt when a system reset is detected on the channel interface. An Output X'67', byte 1, bit 3 resets this bit.

Byte 1, bits 0-7: These bits are 0.

Input X'61' (Initial Selection Address and Command)

Byte 0, bits 0-7: Initial Selection Address - These bits are set during an initial selection sequence and contain the address of the line selected.

Byte 1, bits 0-7: Initial Selection Command - These bits are set during an initial selection sequence and contain the command as presented to the channel adapter from the channel interface.

Input X'62' (Data/Status Control)
This input loads the register specified by R with the contents of the data/status control register. The bits of this register identify the cause of a Type 1 CA data/status L3 interrupt request.

Byte 0, bit 0: Outbound Data Transfer Sequence - This bit indicates that the channel adapter hardware is transferring data to the host processor channel. The control program can set or reset this bit with an Output X'62' instruction.

Byte 0, bit 1: Inbound Data Transfer Sequence - This bit indicates that the channel adapter hardware is transferring data from the host processor channel. The control program can set or reset this bit with an Output X'62' instruction.

Byte 0, bit 2: ESC Final Status Transfer Sequence - This bit indicates that the channel adapter is transferring a 2701/2702/2703 type status byte to the host processor channel. The control program can set or reset this bit with an Output X'62' instruction.

Byte 0, bit 3: NSC Channel End Status Transfer Sequence - This bit indicates that the channel adapter is transferring NSC Channel End status to the host processor channel. The control program can set or reset this bit with an Output X'62' instruction.

Programming Note
The channel adapter automatically generates a Busy initial status byte if the host channel accepts the NSC Channel End status byte. This occurs in all subsequent initial selection sequences on the native mode

subchannel until the control program initiates an NSC final status transfer by executing Output X'62'.

Byte 0, bit 4: NSC Final Status Transfer Sequence - This bit indicates that the channel adapter is tranferring final status to the host processor channel. The control program can set or reset this bit with an Output X'62' instruction.

Byte 0, bit 5: Channel Stop or Interface Disconnect - This bit indicates that a channel stop or interface disconnect was detected when the channel adapter was in a data transfer sequence. This bit is set by hardware and reset by an Output X'62', byte 0, bit 6.

Byte 0, bit 6: Suppress Out Monitor Interrupt - This bit is set by hardware and causes a Type 1 CA data/status L3 interrupt request when the 'suppress out monitor' latch (see Output X'67') is on and the 'suppress out' tag line drops. This bit is reset by an Output X'62', byte 0, bit 6.

Byte 0, bit 7: Program-Requested Interrupt - This bit indicates that the program has requested a Type 1 CA data/status L3 interrupt by executing an Output X'67' instruction. Execution of an Output X'62' instruction with any bit combination resets this bit.

Byte 1, bit 0: Channel Bus Out Check - This bit is set by hardware to indicate incorrect parity on the channel interface during a Write command. This bit is reset by an Output X'62', byte 0, bit 6.

When this check occurs, the byte with incorrect parity is placed in the data buffer, and data transfer is terminated. The transfer byte count (byte 1, bits 5-7) reflect the byte that caused the check.

Byte 1, bit 1: Selective Reset - This bit is set by hardware to indicate the detection of a selective reset when the CA was in a service transfer sequence. This bit is reset by an Output X'62', byte 0, bit 6.

Byte 1, bit 2: Suppress Out - This bit is set to indicate that the 'suppress out' tag line on the channel interface is active.

Byte 1, bit 3: Stacked Ending Status - This bit is set to indicate that the ending status has been stacked by the host processor channel during a status transfer sequence. This bit is reset by an Output X'62', byte 0, bit 6.

Byte 1, bit 4: I/O Command Chaining - This bit indicates command chaining when the CA is in a status transfer sequence. This bit is reset by an Output X'62', byte 0, bit 6.

Byte 1, bits 5-7: Transferred Byte Count Bits 1-3 - These bits are set by the CA hardware and reflect the number of bytes transferred across the channel in the current data transfer operation or the last-completed data transfer operation (if none is in progress). Up to four bytes can be transferred in one data transfer sequence.

Bit 5	Bit 6	Bit 7	Number of Bytes Transferred
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4

Input X'63' (CA Address and Emulator Status Bytes)
This input loads a general register with the CA line address and the status bits for the line that were set by the last Output X'63' instruction.

Byte 0, bits 0-7: Address Byte - These bits contain the address of the last communication line to be serviced by the channel adapter for data or status transfer.

Byte 1, bit 0: ESC Attention.

Byte 1, bit 1: ESC Status Modifier.

Byte 1, bit 2: ESC Control Unit End.

Byte 1, bit 3: ESC Busy.

Byte 1, bit 4: ESC Channel End.

Byte 1, bit 5: ESC Device End.

Byte 1, bit 6: ESC Unit Check.

Byte 1, bit 7: ESC Unit Exception.

Input X'64' (Data Buffer Bytes 1 and 2)
This input loads a general register with the contents of the data buffer bytes 1 and 2 as received from the channel interface. It can also be used to verify the

contents of the data buffers following an Output X'63' instruction.

Byte 0, bits 0-7: Data Buffer Byte 1 - These bits represent the first byte of data received from the channel during a channel Write command.

Byte 1, bits 0-7: Data Buffer Byte 2 - These bits represent the second byte of data received from the channel during a channel Write command.

Input X'65' (Data Buffer Bytes 3 and 4)
This input loads a general register with the contents of the data buffer bytes 3 and 4 as received from the channel interface. It can also be used to verify the contents of the data buffers following an Output X'64' instruction.

Byte 0, bits 0-7: Data Buffer Byte 3 - These bits represent the third byte of data received from the channel during a channel Write command.

Byte 1, bits 0-7: Data Buffer Byte 4 - These bits represent the fourth byte of data received from the channel during a channel Write command.

Input X'66' (NSC Status Byte)
This input loads a general register with the contents of the NSC status register. These bits reflect the status bits loaded into the status register by the last Output X'66' instruction. This instruction should be used only as a diagnostic aid.

Byte 0, bit 0: Attention

Byte 0, bit 1: Status Modifier

Byte 0, bits 2-3: These bits are 0.

Byte 0, bit 4: Channel End.

Byte 0, bit 5: Device End.

Byte 0, bit 6: Unit Check.

Byte 0, bit 7: Unit Exception.

Byte 1, bits 0-7: These bits are 0.

Input X'67' (CA Controls)
This input loads a general register with various CA control information. Byte 1, bits 0-3 (check indications) are reset by an Output X'67', byte 1, bit 2.

Byte 0, bits 0-7: These bits are set to the NSC address for the enabled A or B interface.

Byte 1, bit 0: Channel Bus In Check - This bit indicates that incorrect parity was detected on the channel bus in. When this is detected, the hardware generates good parity and causes a level 1 interrupt.

Byte 1, bit 1: In/Out Instruction Accept Check - This bit indicates that the control program executed an Input or Output X'60', X'61', X'62', X'63, X'64', X'65', or X'66' instruction when the CA was in the process of handling a data/status transfer. Detection of this condition also causes a level 1 interrupt request.

Byte 1, bit 2: CCU Outbus Check - This bit indicates that the CA hardware detected incorrect parity on the CCU outbus. When this is detected, the hardware causes a level 1 interrupt request and prohibits reselection on the channel interface until this bit is reset.

Byte 1, bit 3: Local Store Check - This bit indicates that the CA hardware detected incorrect parity on data bytes gated out of local store. The control program should place good parity in local store by executing an Output X'63', X'64', or X'65' instruction.

Byte 1, bit 4: Channel Interface Enabled - This bit indicates that either interface A or interface B is enabled. When no interface is enabled, the bit is 0.

Byte 1, bit 5: NSC Address Active - This bit indicates that the native subchannel has been selected and is active. The bit is reset when the host channel accepts the final status from the CA.

Byte 1, bits 6-7: These bits are unused.

Type 1 CA Output Instructions

Output X'60' (Reset Initial Selection)
This instruction is executed to reset the Type 1 CA initial selection latches and the level 3 interrupt request resulting from an initial selection. Since this instruction performs a function, the bit settings of the register specified by the R operand are ignored.

Output X'62' (Data/Status Control)
This output instruction unconditionally resets the CA program-requested interrupt and is used to set the following bits in the data/status control register.

Byte 0, bit 0: Outbound Data Transfer Sequence - This bit is set by the control program to initiate an outbound data transfer sequence. For example, a channel Read or Sense command initiates an outboard data transfer sequence that sends data to the host processor.

Byte 0, bit 1: Inbound Data Transfer Sequence - This bit is set by the control program to initiate an inbound data transfer sequence. For example, a channel Write command initiates an inbound data transfer sequence that receives data from the host processor.

Byte 0, bit 2: NSC Final Status Transfer Sequence - This bit is set by the control program and signals the hardware to initiate an ESC status transfer using the address and status information previously loaded into the address and status register by an Output X'63'.

Byte 0, bit 3: NSC Channel End Status Transfer - This bit is set by the control program and signals the CA hardware to present Channel End only for the associated NSC address.

Byte 0, bit 4: NSC Final Status Transfer - This bit is set by the control program and signals the CA hardware to initiate an NSC final status transfer sequence.

Byte 0, bit 5: Reset Initial Selection - This bit is set by the control program to reset byte 0, bits 0-6 of the initial selection control register (Input X'60'). This reset allows the Type 1 CA to accept an initial selection. The bit position is reset to 0 at the completion of the Output X'62' instruction.

Note: *This bit resets the Type 1 CA initial L3 interrupt request unless the interrupt was caused by a system reset.*

Byte 0, bit 6: Reset Data/Status Interrupt - This bit is set by the control program to reset the Type 1 CA data/status L3 interrupt request and the following data/status service register bits.

Interface disconnect
Selective reset
Bus out check
Command reject
Channel stop
Stacked status

When this bit is on along with one of the transfer sequence bits (byte 0, bits 0-4), the CA hardware raises the 'request in' tag line on the channel interface, except when ESC Test I/O Status is available (Output X'62', byte 1, bit 4).

This bit position is reset to 0 at the completion of the Output X'62' instruction.

Byte 0, bit 7: This bit is unused.

Byte 1, bit 0: This bit is unused.

Byte 1, bit 1: Reset Suppress Out Monitor Interrupt - This bit is set by the control program to reset the 'suppress out monitor' latch. This bit position is reset to 0 at the completion of the Output X'62' instruction.

Programming Note
If the suppress out monitor interrupt (Input X'62', byte 0, bit 6) is active, this bit should be set to 1 when initiating the next transfer sequence.

Byte 1, bit 2: This bit is unused.

Byte 1, bit 3: Set Suppressible Status - This bit should be set when the control program is presenting suppressible status to the host channel in ESC mode. Status is suppressible if 'stacked status' is received for a particular line or when the line has been issued an interface disconnect. Refer to *IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturers Information* (GA22-6974) for further information on suppressible status.

Byte 1, bit 4: Set ESC Test I/O Status Available - When this bit is set to 1, the controller hardware responds to a Test I/O command given to a 2701/2702/2703 address. The control program must have previously loaded the correct address and status byte into the CA address and emulator status byte register (Output X'63') before setting this bit. This bit is reset by the emulator hardware when it presents this status to the channel.

When this bit is set to 1, byte 0, bit 2 of this instruction must also be set to 1.

Byte 1, bit 5: This bit is unused.

byte 1, bits 6-7: Request Byte Count 1 and 2 - These bits are set accordingg to the number of bytes to be transferred during inbound or outbound data transfer.

Bit 6	Bit 7	Number of Bytes Transferred
0	0	4
0	1	1
1	0	2
1	1	3

Output X'63' (CA Address and Emulator Status Bytes)
The control program can set this output to indicate the line address (NSC or ESC) and the status of the line to be serviced next by the CA in emulation mode.

Byte 0, bits 0-7: Address Byte - These bits are set by the control program to indicate the I/O device address to be serviced.

Byte 1, bit 0: Set ESC Attention status.

Byte 1, bit 1: Set ESC Status Modifier.

Byte 1, bit 2: Set ESC Control Unit End status.

Byte 1, bit 3: Set ESC Busy status.

Byte 1, bit 4: Set ESC Channel End status.

Byte 1, bit 5: Set ESC Device End status.

Byte 1, bit 6: Set ESC Unit Check status.

Byte 1, bit 7: Set ESC Unit Exception status.

Output X'64' (Data Buffer Bytes 1 and 2)
This output instruction is used only for outbound data transfer.

Byte 0, bits 0-7: Data Buffer Byte 1 - These bits represent the first data byte to be sent across the channel during a channel Read command. This buffer must be loaded by the control program.

Byte 1, bits 0-7: Data Buffer Byte 2 - These bits represent the second data byte to be sent across the channel during a channel Read command. This buffer must be loaded by the control program.

Output X'65' (Data Buffer Bytes 3 and 4)
This output instruction is used only for outbound data transfer.

Byte 0, bits 0-7: Data Buffer Byte 3 - These bits represent the third data byte to be sent across the channel during a channel Read command. This buffer must be loaded by the control program.

Byte 1, bits 0-7: Data Buffer Byte 4 - These bits represent the fourth data byte to be sent across the channel during a channel Read command. This buffer must be loaded by the control program.

Output X'66' (CA NSC Status Byte)
This output loads the native mode status byte with the bits that are set in the register specified by R. The control program sets these bits to indicate the status to be presented across the channel interface when the Type 1 CA is in the NSC status transfer sequence. Hardware resets these bits when the status byte is accepted by the channel.

Byte 0, bits 0-7: These bits are unused.

Byte 1, bit 0: Set Attention status.

Byte 1, bit 1: Set Status Modifier.

Byte 1, bits 2-3: These bits are unused.

Byte 1, bit 4: Set Channel End.

Byte 1, bit 5: Set Device End.

Byte 1, bit 6: Set Unit Check.

Byte 1, bit 7: Set Unit Exception.

Output X'67' (CA Controls)
This instruction causes various control latches to be set or reset in the channel adapter according to the states of the bits in the register specified by R.

Byte 0, bits 0-3: These bits are unused.

Byte 0, bit 4: Diagnostic Reset - This bit resets the channel adapter when the 'channel interface disable' latch is set. This bit should be set only by a diagnostic program and not by in-line code.

Byte 0, bits 5-7: These bits are unused.

Byte 1, bit 0: Suppress Out Monitor Interrupt - A 1 in this position causes the CA to monitor the 'suppress out' tag line for the inactive state. When this is detected, the Type 1 CA data/status L3 interrupt request is set. After servicing the request, the control program

should execute an Output X'62' with byte 1, bit 2 on to reset the request.

Programming Note
Following a stacked status condition, the control program can use this bit to cause the channel adapter to signal when the suppress status indication is removed. This bit should not be on if Output X'67' is executed when a Type 1 CA L3 interrupt request is not set.

Byte 1, bit 1: Set Program-Requested Interrupt - A 1 in this position indicates that a Type 1 CA data/status L3 interrupt is requested. If a data/status transfer or initial selection is in progress, the interrupt request is held until the sequence is complete.

Byte 1, bit 2: Reset Level 1 Checks - A 1 in this position resets the Type 1 CA level 1 check latches and interrupt requests.

Byte 1, bit 3: Reset System Reset and NSC Address Active - A 1 in this position resets the level 3 interrupt request caused by a system reset. It also resets the NSC address active indication.

Byte 1, bit 4: Allow Channel Interface Enable - A 1 in this position causes the 'channel interface enable' latch to be set. This enables the CA to communicate with the host processor. The bit must be 0 if byte 1, bit 7 of this output is 1.

Programming Note
The channel interface cannot be enabled following a power-off to power-on-reset until an Output X'67 is executed with this bit on. The IPL Bootstrap program performs this operation in IPL Phase 3.

Byte 1, bit 5: ESC Command Free - A 1 in this position resets the emulator subchannel (ESC) addresses to be operational. The channel interface must be enabled before the emulator subchannels can become operational.

Byte 1, bit 6: ESC Command Free - A 1 in this position resets the 'ESC command active' latch. The 'ESC command active' latch is set by initial selection.

Programming Note
The Type 1 CA cannot be disabled until it is free of commands; therefore, the control program must ensure that the 'ESC command active' latch is reset before a disable attempt is made.

Byte 1, bit 7: Allow Channel Interface Disable - A 1 in this position sets the 'channel interface disable' latch. This latch overrides the I/O Channel 1 Enable/Disable switch on the control panel and allows the channel to become disabled if (1) the channel adapter is free of commands, (2) commands are not chained, and (3) the adapter is not in initial selection.

The 'channel interface disable' latch is reset by Output X'67', byte 1, bit 4.

Type 2 CA Input Instructions

Input X'50' (Inbound Data Control Word Address Register - INCWAR)
This input instruction loads a general register with the contents of the INCWAR. The INCWAR contains the storage address of the next control word to be used when the channel adapter receives a Write, Write Break, or Write IPL command. The Type 2 CA recognizes this instruction only after setting a CA interrupt request.

Byte 0, bits 0-7: INCWAR bits 0-7.

Byte 1, bits 0-6: INCWAR bits 8-14.

Byte 1, bit 7: INCWAR bit 15 - Ignored by cycle-steal operations, otherwise 0.

Input X'51' (Outbound Data Control Word Address Register - OUTCWAR)
This input instruction loads a general register with the contents of the OUTCWAR. The OUTCWAR contains the storage address of the next control word to be used when the channel adapter receives a Read command.. The instruction is recognized by the Type 2 CA only following the setting of a CA interrupt request.

Byte 0, bits 0-7: OUTCWAR bit 0-7.

Byte 1, bits 0-6: OUTCWAR bits 8-14.

Byte 1, bit 7: OUTCWAR bit 15 - Ignored by cycle-steal operations, otherwise 0.

Input X'52' (Control Word Byte Count Register - CWCNT)
This input instruction loads a general register with the number of bytes remaining to be transferred under the last control word that was fetched. See *Control Word Byte Count Register* in Chapter 9. The Type 2 CA recognizes this instruction only after setting a CA interrupt request.

Byte 0, bits 0-5: These bits are 0.

Byte 0, bits 6-7: CWCNT bits 0-1.

Byte 1, bits 0-7: CWCNT bits 2-9.

Input X'53' (Channel Adapter Sense Register - CASNSR)
This input is used to load a general register with the contents of the channel adapter sense register.

The sense register is accessible for input only when the CA is active, selected, and has a level 1 or level 3 interrupt request set, or the CA is in the diagnostic wrap state. If these conditions are not met, an Input X'53' causes a CCU Inbus parity check. If an interrupt request is set and the CA is not active, an Input X'53' causes the general register to be loaded with zeros.

Byte 0, bit 0: Command Reject - This bit is set when the host processor channel command presented to the CA during initial selection is not a valid command for the controller. When the channel adapter is not initialized, this bit, along with Unit Check, is set during the initial selection sequence for all commands other than the Write IPL command.

Byte 0, bit 1: Intervention Required - This bit indicates programming errors detected by the CCU or CA hardware, or the control program. It is set under hardware control for any one of the following conditions:

1. The CCU 'hard stop' latch is set while the CA is transferring data under a Read, Write, or Write Break command.
2. An addressing exception or a protection check was caused by the address used by the CA for a cycle-steal operation.
3. A TIC command or command chaining to a control word address above 64K has been detected during a CW fetch cycle steal.
4. An OUT or OUT STOP control word was decoded when executing a channel Write, Write Break, or Write IPL command during a CW fetch cycle steal.
5. An IN control word was decoded when executing a channel Read command during a CW fetch cycle steal.
6. An IN, OUT, or OUT STOP control word was decoded with a byte count of zero during a CW fetch cycle steal.

Programming Note
Condition 3 above also sets byte 0, bit 0 (invalid CWAR address) of the channel adapter check register. Conditions 4, 5, and 6 above also set byte 0, bit 1 (invalid control word format) of the channel adapter check register.

Byte 0, bit 2: Bus Out Check - This bit is set when a parity error is detected on the channel bus out during the initial selection command transfer or during host processor-to-controller data transfer. The control program cannot set this bit.

Byte 0, bit 3: Equipment Check - This bit is set any time the Type 2 CA detects an internal hardware error or a parity error on the Inbus or Outbus between the Central Control Unit and the CA.

Byte 0, bits 4-5: These bits are 0.

Byte 0, bit 6: Not Initialized - This bit is on when the channel adapter has not been initialized. The controller read-only-storage sets the not-initialized condition when the CA goes offline. This condition is reset only when the CA accepts an IPL command from the host.

Byte 0, bit 7: Abort - This bit indicates that the channel adapter has halted its channel operation abnormally.

Byte 1, bits 0-7: These bits are 0.

Input X'54' (Channel Adapter Status Register - CASTR)

This input is used to load a general register with the contents of the channel adapter status register. The bits of the status register can be set by the control program and/or the CA hardware. See Output X'54' in this appendix for the method of setting each bit.

The status register is accessible for input only when the CA is active, selected, and has a level 1 or level 3 interrupt request set, or the CA is in the diagnostic wrap state. If these conditions are not met, an Input X'54' causes a CCU Inbus parity check. If an interrupt request is set and the CA is not active, an Input X'54' causes the general register to be loaded with zeros.

Byte 0, bit 0: Attention - This bit indicates that Attention has been set.

Byte 0, bit 1: Status Modifier - This bit indicates that the Status Modifier has been set.

Byte 0, bit 2: This bit is 0.

Byte 0, bit 3: Busy - This bit indicates that Busy status has been set.

Byte 0, bit 4: Channel End - This bit indicates that Channel End status has been set.

Byte 0, bit 5: Device End - This bit indicates that Device End status has been set.

Byte 0, bit 6: Unit Check - This bit indicates that a Unit Check has occurred.

Byte 0, bit 7: Unit Exception - This bit indicates that Unit Exception status has been set.

Byte 1, bits 0-7: These bits are 0.

Input X'55' (Channel Adapter Control Register - CACR)

This input instruction loads a general register with the status of various control latches in the Type 2 CA. This instruction is recognized by the Type 2 CA only following the setting of a CA select and a CA interrupt request. The Type 2 CA need not be in the CA Active state.

Byte 0, bit 0: Diagnostic Wrap Mode - This bit indicates that the controller is offline and in the diagnostic wrap state.

Byte 0, bit 1: Zero Count Override - This bit indicates the condition of the zero count override flag in the control word just executed. It is reset either when a control word is fetched with zero count override off, or when Channel End is generated for the current command.

Byte 0, bit 2: INCWAR Valid - This bit indicates that the control word address register for inbound data transfer (channel Write command) points to the storage location containing the control word to be used for controlling this type of data transfer.

The control program sets this bit during a Type 2 CA-requested level 3 interrupt via an Output X'55' instruction. However, once data transfer across the channel begins, this bit is controlled as follows.

After an IN-CW fetch operation, the bit reflects the status of the chain flag of the IN-CW fetched for the Type 2 CA. During the CW fetch operation, the address in the INCWAR register is incremented by 4 (fullword address). The chain flag in the fetched CW indicates whether or not the updated INCWAR points to a valid CW. If the chain bit is off in the fetched CW, the 'INCWAR valid' latch is reset.

Byte 0, bit 3: OUTCWAR Valid - This bit indicates that the control word address register for outbound data transfer (that is, a channel READ command) points to the storage location containing the control word to be used for controlling this type of data transfer. The control program sets this bit during a CA-requested level 3 interrupt by an Output X'55' instruction. However, once data transfer across the channel has started, this bit is controlled as follows.

After an OUT-CW fetch operation, this bit reflects the status of the chain flag in the OUT or OUTSTOP control word fetched for the CA. During the CW fetch operation, the address in the OUTCWAR is incremented by 4 (fullword address). The chain flag in the fetched CW indicates whether or not the updated OUTCWAR points to a valid CW. If the chain bit is off in the fetched CW, the 'OUTCWAR valid' latch is reset.

Byte 0, bit 4: Program Requested Level 3 Interrupt - This bit indicates that the Type 2 CA L3 interrupt was initiated because the control program set the CA mode register byte 1, bit 0 (set Type 2 CA L3 request).

Byte 0, bit 5: Program Requested Abort/Level 3 Interrupt - This bit indicates that a level 3 interrupt was caused while the adapter was in the active state. The interrupt was caused by executing an Output X'57' with byte 1, bit 1 on. This bit is reset when Output X'57' is executed with byte 1, bit 3 on (reset L3 request).

Byte 0, bit 6: Program Requested Attention - This bit indicates that the program has requested Attention by executing an Output X'55' instruction with byte 0, bit 6 on. It is reset when the host channel accepts the status byte containing Attention.

Byte 0, bit 7: Channel Adapter Active - This bit indicates that the Type 2 CA is currently executing a channel command. It is set by completion of the initial selection for the command and is reset when the host channel accepts the Device End status for that command.

Byte 1, bit 0: Command Chaining - This bit is set by the Type 2 CA hardware when the 'suppress out' tag line is up at the time the channel accepts ending status from the Type 2 CA. It is reset, if the adapter is not active, at the end of the first level 3 interrupt to occur after the latch has been set; or it is reset each time a Read, Write, Write Break, or Write IPL command is decoded during initial selection.

Byte 1, bit 1: Write Break Command Remember - This bit is set by the channel adapter when a Write Break command (X'09') is received. It is reset when the host channel accepts Device End status for that command.

Byte 1, bit 2: Channel Stop/Interface Disconnect - This bit is set by the channel adapter when a Channel Stop or an Interface Disconnect is detected on the channel interface. This bit is reset by an Output X'57', byte 1, bit 6.

Byte 1, bit 3: Selective/System Reset - This bit is set by the Type 2 CA hardware when a system reset or a selective reset terminates a host processor channel command. It is reset by Output X'57', byte 1, bit 5.

Byte 1, bit 4: This bit is 0.

Byte 1, bit 5: Channel Read Command Remembrance - This bit is set when a Read command is accepted by the Type 2 CA. It is reset when the next Write command is accepted.

Byte 1, bit 6: Type 2 CA-2 Selected - This bit indicates that the second channel adapter (Type 2 CA-2) has been selected for operation by an Output X'57'. This bit is 0 if only one channel adapter is installed.

Byte 1, bit 7: Type 2 CA-1 Selected - This bit indicates that the first channel adapter (Type 2 CA-1) has been selected for operation by an Output X'57'.

Input X'56' (Channel Adapter Check Register - CACHKR)

This input instruction loads a general register with the contents of the CA check register. All the bits of the check register set a level 1 interrupt request. By executing this input, the level 1 interrupt check routine can determine the exact cause of the Type 2 CA level 1 check. All the latches in this register are automatically reset at the end of the first level 1 interrupt that occurs after the latches have been set. This input instruction is recognized only after a Type 2 CA interrupt request is set.

Byte 0, bit 0: Invalid CWAR Address - This bit is set when the CWAR associated with the current channel operation points to a storage address above 64k bytes.

Byte 0, bit 1: Invalid Control Word Format - This bit is set when (1) an OUT STOP control word is fetched when executing a channel Write command, (2) an IN control word is fetched when executing a channel Read command, or (3) an IN, OUT, or OUT STOP control word containing a data count of zero is fetched.

Byte 0, bit 2: Cycle Steal Address Check - This bit is set when the CCU signals the channel adapter that an address error has occurred during a cycle steal operation. It indicates that the cycle steal address (1) is beyond the storage capacity of the machine, (2) is of incorrect parity, or (3) points to a protected area of storage.

Byte 0, bit 3: CWAR/Data Buffer Check - This bit indicates that either the INCWAR, OUTCWAR, data 1, or data 2 register contained incorrect parity when access was attempted for either an input instruction or data transfer to the channel during a Read command.

Byte 0, bit 4: CCU Outbus Check - This bit is set (1) when data from an output instruction to the Type 2 CA has incorrect parity on the CCU Outbus, or (2) when, during a cycle steal, the data from storage contained incorrect parity.

Byte 0, bit 5: CCU INBUS Check - This bit indicates that incorrect parity was present on the Inbus during a Type 2 CA cycle steal or input instruction.

Byte 0, bit 6: Channel Bus Out Check - This bit indicates that a parity check was detected on the channel bus out lines during initial selection or data transfer.

Byte 0, bit 7: This bit is 0.

Byte 1, bits 0-3: These bits are 0.

Byte 1, bit 4: Channel Bus In Check (Interface A) - This bit indicates that the sense, status, data, or address byte presented to interface A did not have correct parity.

Byte 1, bit 5: Channel Bus In Check (Interface B) - This bit indicates that the sense, status, data, or address byte presented to interface B did not have the correct parity.

Byte 1, bits 6-7: These bits are 0.

Input X'58' (Channel Bus Out Diagnostic Register - CBODR)
This input loads a general register with the current state of the host processor 'bus out' lines when in the diagnostic wrap state.

Byte 0, bits 0-7: Channel Bus Out bits 0-7.

Byte 1, bit 0: Channel Bus Out Parity Bit - This bit may be used in diagnostic mode to check the Type 2 CA error-detection circuits.

Byte 1, bit 1: This bit is 0.

Byte 1, bit 2: Transfer Byte 1 - This bit indicates that the Type 2 CA is currently transferring an odd numbered byte across the channel interface.

Byte 1, bit 3: Transfer Byte 2 - This bit indicates that the Type 2 CA is currently transferring an even numbered byte across the channel interface.

Byte 1, bit 4: Interface A Enabled - This bit is 1 when channel interface A of the Channel Adapter is currently enabled.

Byte 1, bit 5: Interface B Enabled - This bit is 1 when channel interface B of the Channel Adapter is currently enabled.

Byte 1, bit 6: CSAR Byte X, Bit 6 - This bit is 1 when byte x, bit 6 of the cycle-steal address register (with Extended Addressing) is 1.

Byte 1, bit 7: CSAR Byte X, Bit 7 - This bit is 1 when byte X, bit 7 of the cycle-steal address register (with Extended Addressing) is 1.

Input X'59' (Cycle-Steal Address Register - CSAR)
This instruction is used to load a general register with the current storage data address from the cycle-steal address register while data transfer is in progress.

Byte X, Bits 6-7: CSAR byte X, bits 6-7 (with Extended Addressing only).

Byte 0, bits 0-7: CSAR byte 0, bits 0-7.

Byte 1, bits 0-7: CSAR byte 1, bits 0-7.

Input X'5A' (CA Data Buffer - CADB)
This input is used to load a general register with incoming data from the channel adapter data buffer.

Byte 0, bits 0-7: Data buffer byte 0, bits 0-7.

Byte 1, bits 0-7: Data buffer byte 1, bits 0-7.

Input X'5B' (Channel Adapter Tag Diagnostic Register)
This input is used to load a general register with a combination of bits to indicate the state of the channel tag lines for diagnostic purposes. When used as input to the control program, and the adapter is not in the diagnostic mode, the register indicates the current status of the host processor/controller interface A tag lines. Interface B is not accessible.

Byte 0, bit 0: Select Out/Hold Out - This bit indicates the state of the inbound channel 'select out/hold out' tag line.

Byte 0, bit 1: Address Out - This bit indicates the state of the channel 'address out' tag line.

Byte 0, bit 2: Command Out - This bit indicates the state of the channel 'command out' tag line.

Byte 0, bit 3: Service Out - This bit indicates the state of the channel 'service out' tag line.

Byte 0, bit 4: Operational Out - This bit indicates the state of the channel 'operational out' tag line.

Byte 0, bit 5: Suppress Out - This bit indicates the state of the channel 'suppress out' tag line.

Byte 0, bits 6-7: These bits are 0.

Byte 1, bit 0: Select Out - This bit indicates the state of the outbound channel 'select out' tag line.

Byte 1, bit 1: Request In - This bit indicates the state of the channel 'request in' tag line.

Byte 1, bit 2: Operational In - This bit indicates the state of the channel 'operational in' tag line.

Byte 1, bit 3: Address In - This bit indicates the state of the channel 'address in' tag line.

Byte 1, bit 4: Status In - This bit indicates the state of the channel 'status in' tag line.

Byte 1, bit 5: Service In - This bit indicates the state of the channel 'service in' tag line.

Byte 1, bit 6: This bit is 0.

Byte 1, bit 7: Generate Busy - This bit indicates that the channel is busy.

Input X'5C' (CA Command Register - CMDR)
This instruction loads a general register with the current or last command executed. Byte 1 also indicates the current or last control word type executed.

Byte 0, bit 0: Test I/O - This bit is 1 when the command received from the host processor was a Test I/O (X'00').

Byte 0, bit 1: Write - This bit is 1 when the command received from the host processor was a channel Write (X'01') command.

Byte 0, bit 2: Read - This bit is 1 when the command received from the host processor was a channel Read (X'02') command.

Byte 0, bit 3: No-Op - This bit is 1 when the command received from the host processor was a No-Op (X'03').

Byte 0, bit 4: Sense - This bit is 1 when the command received from the host processor was a Sense (X'04') command.

Byte 0, bit 5: This bit is 0.

Byte 0, bit 6: Write Break - This bit is 1 when the command received from the host processor was a Write Break (X'09') command.

Byte 0, bit 7: This bit is 0.

Byte 1, bit 0: OUT Control Word - This bit is 1 when the current or last control word in use was an OUT control word.

Byte 1, bit 1: OUT STOP Control Word - This bit is 1 when the current or last control word in use was an OUT STOP control word.

Byte 1, bit 2: IN Control Word - This bit is 1 when the current or last control word in use was an IN control word.

Byte 1, bit 3: TIC Control Word - This bit is 1 when the current or last control word in use was a TIC control word.

Byte 1, bits 4-6: These bits are 0.

Byte 1, bit 7: Write IPL - This bit is 1 when the command received from the host processor was a Write IPL (X'05') command.

Type 2 CA Output Instructions

Output X'50' (Inbound Data Control Word Address Register - INCWAR)
This output instruction is used to load the INCWAR with the storage address of the control word (CW) to be fetched by the Type 2 CA cycle-steal hardware when a channel Write, Write Break, or Write IPL command is decoded. The Type 2 CA recognizes this instruction only after setting a CA interrupt request.

Programming Note
All CWs must begin on a halfword boundary and reside in the lower 64k bytes of storage. CW chaining or a TIC (transfer in channel) to an address above 64k causes a CW error condition resulting in a CA level 1 interrupt.

Byte 0, bits 0-7: INCWAR bits 0-7.

Byte 1, bits 0-6: INCWAR bits 8-14.

Byte 1, bit 7: INCWAR bit 15 - Ignored by cycle-steal operations, otherwise 0.

Output X'51' (Outbound Data Control Word Address Register - OUTCWAR)
This output instruction loads the OUTCWAR with the storage address of the control word (CW) to be fetched by the Type 2 CA cycle-steal hardware when a channel Read command is decoded. The instruction is recognized by the Type 2 CA only following the setting of a CA interrupt request.

Programming Note
All CWs must begin on a halfword boundary and reside in the lower 64k bytes of storage. CW chaining or TIC (transfer in channel) to an address above 64k causes a CW error condition resulting in a CA level 1 interrupt.

Byte 0, bits 0-7: OUTCWAR bits 0-7.

Byte 1, bits 0-6: OUTCWAR bits 8-14.

Byte 1, bit 7: OUTCWAR bit 15 - Ignored by cycle-steal operations, otherwise 0.

Output X'53' (Channel Adapter Sense Register - CASNSR)
This output instruction is used to set the abort indication in the CA sense register. The Type 2 CA recognizes this instruction only after setting a CA interrupt request while the Type 2 CA is in the CA active state. If Output X'53' is issued while in a CA interrupt state and the CA is not active, the output is ignored, and no indication of this is returned to the control program. The control program should always determine the active/inactive state of the CA before issuing this output instruction.

Programming Note
The setting of any CASNSR bit causes the Unit Check bit to be set in the Type 2 CA status register and also causes the termination of any data transfer that may have been in progress. CASNSR is reset during initial selection whenever the Type 2 CA accepts a command other than Sense, Test I/O, or No-Op.

Byte 0, bit 0: This bit is unused.

Byte 0, bit 1: Intervention Required - This bit should never be set by the control program during normal operation.

Byte 0, bits 2-6: These bits are unused.

Byte 0, bit 7: Abort - This bit can be set by the control program during a Type 2 CA interrupt if the CA active state exists. Abort indicates to the host processor that the control program has halted its channel operation abnormally.

Byte 1, bits 0-7: These bits are unused.

Output X'54' (Channel Adapter Status Register - CASTR)
This instruction sets the channel adapter status register bits. The instruction is recognized by the Type 2 CA only following the setting of a CA level 1 or level 3 interrupt request while the Type 2 CA is in the CA active state or in the diagnostic wrap state.

Byte 0, bit 0: Attention - This bit may be set indirectly when the Type 2 CA is in the CA inactive state.

Asynchronous Attention status is initiated by an output to the CA mode register requesting a CA level 3 interrupt. In the subsequent CA level 3 interrupt, the program should execute an output to the CA control register (byte 0, bit 6) to request Attention status. Asynchronous Attention indicates that the controller requires service from the channel. This bit may be set directly when the CA active and CA interrupt states exist.

Byte 0, bit 1: Status Modifier - This bit may have special applications in controlling channel data transfers. The status modifier bit can be presented only with Device End and should convey to the host processor a unique indication for the particular command being executed.

Byte 0, bit 2: This bit is unused.

Byte 0, bit 3: Busy - This bit is presented as initial status to all host processor channel commands if (1) initial selection is attempted before Device End status has been signaled to the host processor for the command currently being executed, or (2) the channel adapter has its level 1 or level 3 interrupt request latch set.

If status is available at this time, it is presented, except for Test I/O, along with Busy status. If ending status has been stacked, that ending status is also presented along with Busy.

Byte 0, bit 4: Channel End - This bit is set to indicate that the controller has completed the data transfer of the channel command in progress. Channel End is set by Type 2 CA hardware for each host processor channel command. It may also be set when the CA interrupts an active state and Channel End has not already been presented to the channel for the command being executed.

Byte 0, bit 5: Device End - This bit is set to indicate that the controller has finished with the current host processor channel command and is ready to accept another command from the channel. CE and DE are not always presented together. The conditions under which they are presented separately are discussed under *Status Servicing* in Chapter 9. When CE and DE are presented separately, the CA makes a level 3 interrupt request after the channel accepts CE. When the interrupt request is reset (Output X'57'--channel adapter mode register), the CA sends DE and any other status (Attention, Status Modifier, Unit Exception) that may have been set during execution of the interrupt handling routine.

Byte 0, bit 6: This bit is unused.

Byte 0, bit 7: Unit Exception - This bit is set during initial status to notify the channel that the control routine has not set up a Type 2 CA control word to handle this particular host processor channel command. UE is presented with DE to more efficiently notify the channel that a CW is invalid. Unit Exception can be used in this manner to break host processor channel command chaining without going through another selection sequence when the controller no longer needs service.

Byte 1, bits 0-7: These bits are unused.

Output X'55' (Channel Adapter Control Register - CACR)

This output sets the CA control register bits. The instruction is recognized by the Type 2 CA only following the setting of a CA select and a CA interrupt request. The Type 2 CA need not be in the CA active state.

Byte 0, bits 0-1: These bits are unused.

Byte 0, bit 2: INCWAR Valid - This bit is set by the control program when the control word address register for inbound data transfer (channel Write command) points to the storage location containing the control word to be used for controlling this type of data transfer.

To initialize the CA, the control program sets this bit during a CA-requested level 3 interrupt.

Byte 0, bit 3: OUTCWAR Valid - The control program sets this bit when the control word address register for outbound data transfer (that is, a channel READ command) points to the storage location containing the control word to be used for controlling this type of data transfer. To initialize the Type 2 CA, the control program sets this bit during a CA-requested level 3 interrupt.

Byte 0, bits 4-5: These bits are unused.

Byte 0, bit 6: Program Requested Attention - A 1 in this position causes Attention status (either synchronous or asynchronous) to be presented to the channel at the earliest appropriate time. The bit is reset

when the channel accepts the status byte containing the attention bit.

Byte 0, bit 7: This bit is unused.

Byte 1, bits 0-7: These bits are unused.

Output X'56' (Reset Control Register-CACR)
This output is executed to reset the 'INCWAR valid' and the 'OUTCWAR valid' latches.

Byte 0, bits 0-1: These bits are unused.

Byte 0, bit 2: Reset INCWAR Valid - A 1 in this position causes the 'INCWAR valid' latch to be reset.

Byte 0, bit 3: Reset OUTCWAR Valid - A 1 in this position causes the 'OUTCWAR valid' latch to be reset.

Byte 0, bits 4-7: These bits are unused.

Byte 1, bits 0-7: These bits are unused.

Output X'57' (Channel Adapter Mode Register - CAMR)
This output loads the channel adapter mode register with the bits set in the general register. Byte 1 of this register can be loaded at any time to initialize or halt the Type 2 CA. Byte 0 is accessible only when the CA is 'not initialized' or is in the diagnostic wrap state.

Byte 0, bit 0: Set IPL Attention - A 1 in this position cuases the channel adapter 'attention' latch to be set. This is done as soon as the adapter hardware determines that the channel is not examining the status register.

Byte 0, bit 1: Set IPL Channel End - A 1 in this position causes the channel adapter 'channel end status' latch to be set once the adapter hardware determines that the 'channel end remembrance' latch is not on.

Byte 0, bit 2: This bit is unused.

Byte 0, bit 3: Set IPL Device End - A 1 in this position causes the channel adapter 'device end status' latch to be set.

Byte 0, bit 4: Set IPL Unit Check - A 1 in this position causes the channel adapter 'unit check status' latch to be set. This indication should not be set without setting byte 0, bit 7 (IPL not initialized).

Byte 0, bits 5-6: These bits are unused.

Byte 0, bit 7: Set IPL Not-Initialized - A 1 in this position causes the channel adapter not-initialized sense bit to be set when the channel is not executing a sense command. The not-initialized sense bit can be reset only by the adapter accepting a Write IPL command.

Byte 1, bit 0: Set Type 2 CA Level 3 Interrupt Request - A 1 in this position causes the Type 2 CA to request a level 3 interrupt once the Type 2 CA is in the inactive state, with no outstanding status and no channel chaining indicated. The latch is reset at the end of the first level 3 interrupt to occur after it has been set.

Byte 1, bit 1: Set Program Abort - A 1 in this position causes the Type 2 CA immediately terminate whatever channel operation may be in progress. If the CA is executing a channel Write or Read command, the ending status is CE, DE, and UC with the abort sense bit set. Once the status is accepted by the host processor channel, the CA sets the level 3 interrupt request. Program abort is reset when the CA level 3 interrupt request is reset.

Note: *When Program Abort is indicated, both CWAR-valid flags are reset. They must be set by an Output X'55' (channel adapter control register) before another channel data transfer can be initiated.*

Byte 1, bit 2: Reset Type 2 CA Level 1 Interrupt Request - A 1 in this position resets the Type 2 CA 'level 1 interrupt request' latch. This should be the last action the control program takes before leaving a Type 2 CA initiated level 1 interrupt. All check latches, except 'bus out check', contained in Input X'55', are reset when this output occurs.

Byte 1, bit 3: Reset Type 2 CA Level 3 Interrupt Request - A 1 in this position resets the Type 2 CA 'level 3 interrupt request' latch.

Byte 1, bit 4: Select Type 2 CA-1 - A 1 in this position selects Type 2 CA-1. A 0 in this position selects Type 2 CA-2. The control program must ensure that the state of this bit is correct any time an Output X'57' is issued.

Byte 1, bit 5: Reset Selective/System Reset - A 1 in this position resets the selective system reset condition. However, byte 1, bit 3 of this output must also be 1 to reset the resulting level 3 interrupt request.

page missing from original document

page missing from original document

the address from the LAR is that of the last instruction executed before the input instruction. If this input is executed in program level 1, the address from the LAR is that of the last instruction executed before entering level 1.

Input X'76' (Adapter Interrupt Requests Group 1)
Input X'76' is associated with program level 1 interrupt requests. Execution of this instruction loads the register specified by R with a combination of bits to indicate the origin of an adapter level 1 interrupt request.

Byte 0, bit 0: This bit is 0.

Byte 0, bit 1: Type 2 Scanner-1 (or Type 1 Scanner) L1 - This bit indicates that a Type 1 Scanner or a Type 2 Scanner-1 has requested a program level 1 interrupt.

Byte 0, bit 2: Type 2 Scanner-2 L1 - This bit indicates that a Type 2 Scanner-2 has requested a program level 1 interrupt.

Byte 0, bit 3: Type 2 Scanner-3 L1 - This bit indicates that a Type 2 Scanner-3 has requested a program level 1 interrupt.

Byte 0, bit 4: Type 2 Scanner-4 L1 - This bit indicates that a Type 2 Scanner-4 has requested a program level 1 interrupt.

Byte 0, bit 5: Type 2 CA-1 (or Type 1 CA) L1 - This bit indicates that a Type 1 CA or a Type 2 CA-1 has requested a program level 1 interrupt.

Byte 0, bit 6: Type 2 CA-2 L1 - This bit indicates that a Type 2 CA-2 has requested a program level 1 interrupt.

Byte 0, bit 7: This bit is 0.

Byte 1, bits 0-7: These bits are 0.

Input X'77' (Adapter Interrupt Requests Group 2)
Input X'77' is associated with program level 2 and level 3 interrupt requests. Execution of this instruction loads the register specified by R with a combination of bits to indicate the origin of an adapter interrupt request.

Byte 0, bit 0: This bit is 0.

Byte 0, bit 1: Type 2 Scanner (or Type 1 Scanner) L2 - This bit indicates that a Type 1 Scanner or a Type 2 Scanner has requested a program level 2 interrupt.

Byte 0, bits 2-7: These bits are 0.

Byte 1, bits 0-1: These bits are 0.

Byte 1, bit 2: Type 2 CA-2 L3 - This bit indicates that a Type 2 CA-2 has requested a program level 3 interrupt.

Byte 1, bit 3: Type 1 CA Data/Status L3 - This bit indicates that a Type 1 CA has requested a program level 3 data service interrupt.

Byte 1, bit 4: Type 2 CA-1 (or Type 1 CA) L3 - This bit indicates that a Type 2 CA-1 has requested a program level 3 interrupt, or a Type 1 CA has requested a program level 3 initial selection interrupt.

Byte 1, bits 5-7: These bits are 0.

Input X'79' (Utility)
Execution of Input X'79' causes the register specified by R to be loaded with information indicating (1) the state of the program level 5 C and Z condition latches, (2) the last program level to be active before a level 1 interrupt, and (3) the state of the IPL escape control.

Byte 0,bits 0-5: These bits are 0.

Byte 0, bit 6: Program Level 5, C Condition - This bit indicates that the 'C condition' latch for program level 5 is on.

Byte 0, bit 7: Program Level 5, Z Condition - This bit indicates that the 'Z condition' latch for program level 5 is on.

Byte 1, bit 0: Program Level 2 - This bit indicates that program level 2 was interrupted by level 1. (See note below.)

Byte 1, bit 1: Program Level 3 - This bit indicates that program level 3 was interrupted by level 1. (See note below.)

Byte 1, bit 2: Program Level 4 - This bit indicates that program level 4 was interrupted by level 1. (See note below.)

Byte 1, bit 3: Program Level 5 - This bit indicates that program level 5 was interrupted by level 1. (See note below.)

Byte 1, bits 4-6: These bits are 0.

Byte 1, bit 7: IPL Escape Control - This bit is 0 as a result of a jumper wire installed by a Customer Engineer. The jumper causes a bypass of the part of the bootstrap program that actually controls the initial program load by forcing a branch to storage location X'06FC'. This is a diagnostic facility for read-only-storage maintenance.

Note to byte 1 bits 0-3: *One of these bits is 1 when an Input X'79' is executed in program level 1. The bit that is set indicates the program level that was operating when control was passed to program level 1. All other positions are set to 0. These bits are all 0 if Input X'79' is executed when not in program level 1.*

Input X'7B' (BSC CRC Register)
This instruction causes the old CRC character and the data character to be added to the CRC accumulation to be combined and loaded into the BSC CRC register. Then bytes 0 and 1 of the general register specified by R are loaded with the new BSC CRC accumulation character from the BSC CRC register. With Extended Addressing, byte X of the general register is set to 0. See *Cyclic Redundancy Check* in Chapter 5.

Input X'7D' (CCU Check Register)
The bits of this input are set when the Central Control Unit detects an error condition. Execution of this instruction sets the bits in the general register specified by R to correspond to the CCU check register.

Byte 0, bit 0: Byte X Check - This bit indicates a byte X parity error (for Extended Addressing only).

Byte 0, bit 1: Byte 0 Check - This bit indicates a byte 0 parity error.

Byte 0, bit 2: Byte 1 Check - This bit indicates a byte 1 parity error.

Byte 0, bit 3: L1 Program Check - This bit indicates that a program check occurred while in level 1.

Byte 0, bit 4: SAR Check - This bit indicates a storage address register parity check. Byte 0, bit 0, 1, or 2 is also 1 to indicate which byte of the SAR caused the check.

Byte 0, bit 5: SDR Check - This bit indicates a storage data register parity check. Byte 0, bit 0, 1, or 2 is also 1 to indicate which byte of the SDR caused the check.

Byte 0, bit 6: Op Reg Check - This bit indicates an operation register parity check. Byte 0, bit 0, 1, or 2 is also 1 to indicate which byte of the Op Reg caused the check.

Byte 0, bit 7: Indata Bus Check - This bit indicates an indata bus check. Byte 0, bit 0, 1, or 2 is also 1 to indicate which byte of the indata bus caused the check.

Byte 1, bit 0: Cycle Counter Check - This bit indicates that the CCU cycle counter signaled an even number of time slots to the CCU. This check is one of three that set the Clock Check light on the control panel.

Byte 1, bits 1-4: These bits are 0.

Byte 1, bit 5: CCU Checks - This bit indicates the presence of a CCU check. The CCU check bit is a summary type bit that is set to 1 whenever any one or more of the CCU check bits of this input are on. When checking for a CCU check condition, this bit should be tested first.

Byte 1, bit 6: Type 2 Attachment Base Clock Check - This bit indicates a Type 2 Attachment Base clock check. This check indicates that the Type 2 Scanner clock has signaled an incorrect number of time slots. The check is one of three checks that set the Clock Check light on the control panel.

Byte 1, bit 7: CCU Clock Check - This bit indicates a Central Control Unit clock check. A CCU clock check indicates that the CCU clock has signaled an incorrect number of time slots. This check is one of three checks that set the Clock Check light on the control panel.

Input X'7E' (CCU Interrupt Requests Group 1)
This input sets the bits in the register specified by R to indicate which group 1 interrupt request is set. Byte 1, bits 1-4 are set as the result of a program check in any level and cause an interrupt to level 1.

Byte 0, bits 0-7: These bits are 0.

Byte 1, bit 0: Address Compare Interrupt L1 - This bit indicates a level 1 address compare interrupt.

Byte 1, bit 1: Address Exception L1 - This bit indicates that an addressing exception has occured.

Byte 1, bit 2: In/Out Check L1 - This bit indicates that an input/output check has occurred.

Byte 1, bit 3: Protection Check L1 - This bit indicates that a protection check has occurred.

Byte 1, bit 4: Invalid Op Check L1 - This bit indicates that an invalid op-code check has occurred.

Byte 1, bit 5: This bit is 0.

Byte 1, bit 6: IPL L1 - This bit indicates that program level 1 has requested an IPL.

Byte 1, bit 7: This bit is 0.

Input X'7F' (CCU Interrupt Requests Group 2)
This input is used to set bits in the register specified by R to indicate which group 2 interrupt requests are set.

Byte 0, bit 0: Diagnostic L2 - This bit indicates a diagnostic level 2 interrupt request.

Byte 0, bits 1-5: These bits are 0.

Byte 0, bit 6: Panel Interrupt Request L3 - This bit indicates an interrupt request from the Interrupt push button on the control panel.

Byte 0, bit 7: PCI L4 - This bit indicates a level 4 program-controlled interrupt (PCI L4) request.

Byte 1, bits 0-4: These bits are 0.

Byte 1, bit 5: Interval Time L3 - This bit indicates a level 3 interval timer interrupt request.

Byte 1, bit 6: PCI L3 - This bit indicates a level 3 program-controlled interrupt (PCI L3) request.

Byte 1, bit 7: SVC L4 - This bit indicates a level 4 supervisor call interrupt (SVC L4) request.

CCU Output Instructions

Output X'70' (Hardstop)
When this output is executed, it causes the 'hardstop' latch to be set. This causes the controller to come to a complete stop and requires an IPL to continue processing. Since this instruction performs a function, the bit settings of the register are ignored.

Output X'71' (Display Register 1)
When this output is executed, the contents of the register specified by R are transferred to display register 1. The Program Display light on the control panel is also turned *on*.

Output X'72' (Display Register 2)
When this output is executed, the contents of the register specified by R are transferred to display register 2. The Program Display light on the control panel is also turned *on*.

Output X'73' (Set Key)
This output is associated with the storage protect mechanism. Refer to *Storage Protect* in Chapter 5. It is normally used to set either a storage key or protect key with the contents of byte 1, bits 5-7 of the register specified by R.

Byte 1, bit 3 (key select), controls the selection of either a storage key or protect key. If byte 1, bit 4 (set key) of the register is on, the addressed key is set according to byte 1, bits 5-7. If byte 1, bit 4 is off, the addressed key is not set. In either case, the key addressed with the last Output X'73' executed can be accessed with a subsequent Input X'73'.

Byte 0, bits 0-3: Key Address - If byte 1, bit 3 of this output is 1, these bits, along with byte 0, bits 4-6 are set to the storage block number to be referenced when setting a storage key. If byte 1, bit 3 is 0, these bits are ignored.

Byte 0, bits 4-6: Key Address - If byte 1, bit 3 of this output is 1, these bits along with byte 0, bits 0-3 form a storage block number. If byte 1, bit 3 is a 0, these bits are set to the protect key address of the desired user.

Byte 0, bit 7: This bit is unused.

Byte 1, bits 0-2: These bits are unused.

Byte 1, bit 3: Key Address Select - This bit is used to indicate which key this instruction will be used for. If this bit is 1, a storage key is set, and the storage block number should be in byte 0, bits 0-6. If this bit is 0, a protect key is set, and the address of the user should be in byte 0, bits 4-6.

Byte 1, bit 4: Set Key - If this bit is set to 1, the key set in bits 5-7 is inserted for the addressed key. If this bit is 0, no keys are changed.

Byte 1, bits 5-7: Key - These bits contain the storage key or protect key as indicated by byte 1, bit 3 to be assigned to the storage block or user addressed by byte 0, bits 0-6. If byte 1, bit 4 is zero, these bits are ignored.

Output X'77' (Miscellaneous Control)
This output contains miscellaneous controls used to set or to reset various interrupt requests.

Byte 0, bit 0: Reset IPL L1 - This bit is set to 1 to reset the level 1 IPL request.

Byte 0, bit 1: Reset CCU Checks - This bit is set to 1 to reset all CCU checks.

Byte 0, bit 2: Reset Panel Interrupt Request L3 - This bit is set to 1 to reset the external interrupt request at program level 3.

Byte 0, bits 3-5: These bits are unused.

Byte 0, bit 6: Set Diagnostic L2 - This bit is set to 1 to set the level 2 diagnostic interrupt request. It is ignored if the CCU is not in the test mode.

Byte 0, bit 7: Reset Diagnostic L2 - This bit is set to 1 to reset the level 2 diagnostic interrupt request. It is ignored if the CCU is not in the test mode.

Byte 1, bit 0: This bit is unused.

Byte 1, bit 1: Reset Interval Time L3 - This bit is set to 1 to reset the program level 3 interval timer interrupt request.

Byte 1, bit 2: Reset PCI L3 - This bit is set to 1 to reset the level 3 program-controlled interrupt request (PCI L3) set by Output X'7C'.

Byte 1, bit 3: This bit is unused.

Byte 1, bit 4: Reset Address Compare L1 - This bit is set to 1 to reset the program level 1 address compare interrupt request.

Byte 1, bit 5: Reset Program Checks L1 - This bit is set to 1 to reset all program check interrupt requests to program level 1.

Byte 1, bit 6: Reset PCI L4 - This bit is set to 1 to reset the level 4 program-controlled interrupt request (PCI L4) set by Output X'7D'.

Byte 1, bit 7: Reset SVC L4 - This bit is set to 1 to reset the level 4 supervisor call (SVC L4) request.

Output X'78' (Force CCU Checks)
This output provides a means for testing the CCU check circuits under diagnostic control by forcing checks in the CCU data flow. This output instruction is ignored if the CCU is not in the test mode.

Programming Note
Although the action specified by this output is taken only once, the result may be "permanent" until corrective action is taken to store data with the correct parity into the affected register or storage location.

Byte 0, bits 0-7: Complement Bits 0-7 - A 1 in any of these positions causes the corresponding bit positions of each byte (byte X, 0, and 1) of the input to the ALU check detection circuits to be complemented.

Byte 1, bit 0: Complement Storage Parity - A 1 in this position causes incorrect parity in both byte 0 and byte 1 of the halfword accessed in storage on the next instruction cycle.

Byte 1, bit 1: Complement Z Bus Parity - A 1 in this position causes incorrect parity to be stored into a general register to exercise parity checking circuits.

Byte 1, bit 2: A-Register Check - A 1 in this position causes incorrect parity in the A-register.

Byte 1, bit 3: Indata Bus Check - A 1 in this position causes the Indata Bus Check - A 1 in this position causes the Indata parity bits to be complemented. This results in a B-register parity check when an Input X'70', X'73'-X'75', or X'78'-X'7F' instruction is executed. Input instructions X'71' and X'72' do not give checks.

Byte 1, bits 4-7: These bits are unused.

Output X'79' (Utility)
This output is used to set or reset various CCU latches.

Byte 0, bits 0-1: These bits are unused.

Byte 0, bit 2: Set IPL - This bit is set to 1 to initiate an IPL operation.

Byte 0, bits 3-4: These bits are unused.

Byte 0, bit 5: Inhibit Prog Level 5, C & Z Replacement - If this bit is set to 0, the program level 5 C and Z condition latches are set according to byte 0, bits 6-7. If this bit is 1, no action is taken.

Byte 0, bit 6: Program Level 5, C Condition - If byte 0, bit 5 of this output is 1, the 'C condition' latch for program level 5 is set according to the state of this bit.

Byte 0, bit 7: Program Level 5, Z Condition - If byte 0, bit 5 of this output is 1, the 'Z condition' latch for program level 5 is set according to the state of this bit.

Byte 1, bit 0: Reset CCU Check Hard Stop Mode - This bit is set to 1 to reset a CCU check hard stop. It is ignored if the CCU is not in the test mode.

The 'check stop mode' latch is set during IPL phase 1. As long as the Diagnostic Control switch on the control panel is in the PROCESS or CLOCK STEP position, the controller operates as if the switch is in the CCU CHECK HARD STOP position. See *Diagnostic Control Switch* in Chapter 10. This state can be overridden by the bypass CCU check Stop state.

Byte 1, bit 1: Reset Load Indicator - This bit is set to 1 to reset the Load light on the control panel and reset the 'load' latch, to indicate the completion of IPL. The 'load' latch and the Load light are set during IPL.

Byte 1, bit 2: Set Test Mode - This bit is set to 1 to set the CCU in the *test mode*. This bit turns on the Test light on the control panel and enables the following diagnostic functions to be performed:

(1) Force CCU checks - Output X'78'.
(2) Set/reset Diagnostic Level 2.
(3) Set/reset Bypass CCU Check Stop.
(4) Set/reset adapter level 1 interrupt request mask.
(5) Reset CCU check hard stop.

See the *CCU Diagnostic Facilities* section in Chapter 5. This bit should not be 1 if byte 1, bit 3 of this output is 1.

Byte 1, bit 3: Reset Test Mode - This bit is set to 1 to reset the test mode. Reset Test Mode - This bit is set to 1 to reset the test mode. The Test light on the control panel is turned off if the Mode Select and the Diagnostic Control switches are both in the PROCESS position. Any of the test functions allowed by the test mode are also reset. This bit should not be 1 if byte 1, bit 2 of this output is 1.

Byte 1, bit 4: Set Bypass CCU Check Stop Mode - This bit is set to 1 to bypass the CCU check stop. When this bit is set, the operation is the same as if the Diagnostic Control switch on the control panel were in the BYPASS CCU CHECK STOP position. This bit should not be 1 if byte 1, bit 5 of this output is 1. It is ignored if the CCU is not in the test mode.

Byte 1, bit 5: BRset Bypass CCU Check Stop Mode - This bit is set to 1 to reset the bypass CCU check stop. This bit should not be 1 if byte 1, bit 4 of this output is 1. It is ignored if the CCU is not in the test mode.

Byte 1, bit 6: Scope Sync Pulse 1 - This bit is set to 1 to generate the scope sync pulse 1. This is an oscilloscope synchronization pulse of 100 ns for maintenance purposes.

Byte 1, bit 7: Scope Sync Pulse 2 - This bit is set to 1 to generate the scope sync pulse 1. This is an oscilloscope synchronization pulse of 100 ns for maintenance purposes.

Output X'7C' (Set PCI L3)
This output sets the program-controlled interrupt request for level 3. Since this instruction performs a function, the bit settings of the register are ignored.

Output X'7D' (Set PCI L4)
This output sets the program-controlled interrupt request for level 4. Since this instruction performs a function, the bit settings of the register are ignored.

Output X'7E' (Set Mask bits)
This output is used to set mask bits that prevent interrupts to a certain program level during processing. See *Masking Program Level Priorities* in Chapter 5.

Byte 0, bits 0-7: These bits are unused.

Byte 1, bit 0: This bit is unused.

Byte 1, bit 1: Adapter Requests L1 - A 1 in this position sets a mask to prevent CA and Scanner interrupts to program level 1 (for diagnostic test mode only).

Byte 1, bit 2: Program Level 2 - A 1 in this position sets a mask to prevent interrupts to program level 2.

Byte 1, bit 3: Program Level 3 - A 1 in this position sets a mask to prevent interrupts to program level 3.

Byte 1, bit 4: Program Level 4 - A 1 in this position sets a mask to prevent interrupts to program level 4.

Byte 1, bit 5: Program Level 5 - A 1 in this position sets a mask to prevent instruction execution in program level 5.

Byte 1, bits 6-7: These bits are unused.

Output X'7F' (Reset Mask bits)
This output is used to reset the mask bits for program level interrupts. See *Masking Program Level Priorities* in Chapter 5.

Byte 0, bits 0-7: These bits are unused.

Byte 1, bit 0: This bit is unused.

Byte 1, bit 1: Adapter Requests L1 - A 1 in this position unmasks CA and scanner interrupts to level 1 (for diagnostic test modes only).

Byte 1, bit 2: Program Level 2 - A 1 in this position unmasks interrupts to program level 2.

Byte 1, bit 3: Program Level 3 - A 1 in this position unmasks interrupts to program level 3.

Byte 1, bit 4: Program Level 4 - A 1 in this position unmasks interrupts to program level 4.

Byte 1, bit 5: Program Level 5 - A 1 in this position unmasks program level 5 to allow instruction execution at that level.

Byte 1, bits 6-7: These bits are unused.

Appendix C. Input/Output Instruction Summary Charts

INPUT X'00' thru X'1F' — GENERAL REGISTERS

Gen Reg (R)	Reg/Function (E)
BYTE X, BIT 6	BYTE X, BIT 6 (with EA only)
BIT 7	BIT 7 (with EA only)
BYTE 0, BIT 0	BYTE 0, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	BYTE 1, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7

EA = Extended Addressing

OUTPUT X'00' thru X'1F' — GENERAL REGISTERS (Note)

Gen Reg (R)	Reg/Function (E)
BYTE X, BIT 6	BYTE X, BIT 6 (with EA only)
BIT 7	BIT 7 (with EA only)
BYTE 0, BIT 0	BYTE 0, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	BYTE 1, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7

Note: If R field=0, data in addressed reg is not changed but parity is regenerated.

	OUTPUT X'40' TYPE 1 SCANNER - SET MODE BIT OVERRIDE and OVERRIDE REMEMBER Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 * BYTE 1, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 *
INPUT X'41' TYPE 1 SCANNER-INTERFACE ADDRESS (Note) Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 0 BIT 1 0 BIT 2 0 BIT 3 0 BIT 4 1 BIT 5 0 BIT 6 Interface Address BIT 3 BIT 7 BIT 4 BYTE 1, BIT 0 BIT 5 BIT 1 BIT 6 BIT 2 BIT 7 BIT 3 BIT 8 BIT 4 0 BIT 5 0 BIT 6 0 BIT 7 0 Note: Bytes 0 & 1, 1=X'06F0' For Char Serv	OUTPUT X'41' TYPE 1 SCANNER-START SCANNER and RESET L2 BIT SERVICE REQUEST Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 * BYTE 1, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 *
INPUT X'42' TYPE 1 SCANNER-CNTRL A--LINE INTERFACE (Autocall Interface shown below) Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 0 BIT 1 0 BIT 2 0 BIT 3 0 BIT 4 0 BIT 5 0 BIT 6 Mode Bit 1 BIT 7 Mode Bit 2 BYTE 1, BIT 0 1=LO;0=HI Bit Svc Priority BIT 1 Diagnostic Mode BIT 2 Data Terminal Ready (DTR) BIT 3 Synchronous Bit Clock BIT 4 External Clock BIT 5 Data Rate Selector BIT 6 OSC Select Bit 1 BIT 7 OSC Select Bit 2	OUTPUT X'42' TYPE 1 SCANNER-CNTRL A--LINE INTERFACE (Autocall Interface shown below) Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 1=Set; 0=Rst Mode Bit 1 BIT 7 1=Set; 0=Rst Mode Bit 2 BYTE 1, BIT 0 1=LO;0=HI Bit Svc Priority BIT 1 1=Set; 0=Rst Diagnostic Mode BIT 2 1=Set; 0=Rst Data Terminal Ready (DTR) BIT 3 1=Set; 0=Rst Synchronous Bit Clock BIT 4 1=Set; 0=Rst External Clock BIT 5 1=Set; 0=Rst Data Rate Selector BIT 6 1=Set; 0=Rst OSC Select Bit 1 BIT 7 1=Set; 0=Rst OSC Select Bit 2
INPUT X'42' TYPE 1 SCANNER-CNTRL A--AUTOCALL INTERFACE (Line Interface shown above) Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 0 BIT 1 0 BIT 2 0 BIT 3 0 BIT 4 0 BIT 5 0 BIT 6 Mode Bit 1 BIT 7 Mode Bit 2 BYTE 1, BIT 0 1=LO;0=HI Bit Svc Priority BIT 1 0 BIT 2 0 BIT 3 0 BIT 4 0 BIT 5 0 BIT 6 0 BIT 7 0	OUTPUT X'42' TYPE 1 SCANNER-CNTRL A--AUTOCALL INTERFACE (Line Interface shown above) Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 1=Set; 0=Rst Mode Bit 1 BIT 7 1=Set; 0=Rst Mode Bit 2 BYTE 1, BIT 0 1=LO;0=HI Bit Svc Priority BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 *

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

INPUT X'43' — TYPE 1 SCANNER-CNTRL B/C-LINE INTERFACE
(Autocall Interface shown below)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Receive Data Bit Buffer
BIT 1	Feedback Check
BIT 2	Interface Check Summary (note)
BIT 3	Received Data Lead (1=SPACE)
BIT 4	Transmit Mode
BIT 5	New Sync.
BIT 6	Request to Send (RTS)
BIT 7	Send Data Bit Buffer
BYTE 1, BIT 0	Not Clear to Send (CTS)
BIT 1	Ring Indicator (RI)
BIT 2	Not Data Set Ready (DSR)
BIT 3	Received Line Signal Detector
BIT 4	Telegraph Interface Echo Check
BIT 5	Diagnostic Mode
BIT 6	Bit Service
BIT 7	Bit Overrun/Underrun

Note: OR of Byte 0 Bit 1, Byte 1 Bits 2 & 7

OUTPUT X'43' — TYPE 1 SCANNER-CNTRL B-LINE INTERFACE
(Autocall Interface shown below)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	1=Transmit Mode; 0=Receive Mode
BIT 5	New Sync
BIT 6	Request to Send (RTS)
BIT 7	Send Data (1=mark)

INPUT X'43' — TYPE 1 SCANNER-CNTRL B/C-AUTOCALL INTERFACE
(Line Interface shown above)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	0
BIT 1	Feedback Check
BIT 2	Interface Check Summary (note)
BIT 3	Digit Present (DPR)
BIT 4	Digit NBR 8
BIT 5	Digit NBR 4
BIT 6	Digit NBR 2
BIT 7	Digit NBR 1
BYTE 1, BIT 0	Not Abandon Call & Retry (ACR)
BIT 1	Present Next Digit (PND)
BIT 2	Not Data Line Occupied (DLO)
BIT 3	Power Indicator (PWI)
BIT 4	Call Request (CRQ)
BIT 5	Call Originating Status (COS)
BIT 6	Bit Service
BIT 7	Bit Overrun/Underrun

Note: OR of Byte 0 Bit 1, Byte 1 Bits 2 & 7

OUTPUT X'43' — TYPE 1 SCANNER-CNTRL B-AUTOCALL INTERFACE
(Line Interface shown above)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	Call Request (CRQ)
BIT 3	Digit Present (DPR)
BIT 4	Digit NBR 8
BIT 5	Digit NBR 4
BIT 6	Digit NBR 2
BIT 7	Digit NBR 1

INPUT X'44' — TYPE 1 SCANNER-STATUS

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Mode Bit Override
BIT 1	0
BIT 2	Override Remember
BIT 3	Scanner Enabled
BIT 4	Char Svc Pending
BIT 5	0
BIT 6	0
BIT 7	0
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	LIB Pos1 Bit Clock Check
BIT 3	LIB Pos2 Bit Clock Check
BIT 4	LIB Pos3 Bit Clock Check
BIT 5	LIB Pos4 Bit Clock Check
BIT 6	LIB Select Check
BIT 7	CCU OUTBUS Check

OUTPUT X'44' — TYPE 1 SCANNER-GENERAL CONTROL

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*
BYTE 1, BIT 0	1=Set; 0=Rst Diagnostic Bit Svc
BIT 1	*
BIT 2	Reset Mode Bit Override
BIT 3	Reset Override Remember
BIT 4	Reset Character Service Pending
BIT 5	Reset Type 1 Scan L1 Checks
BIT 6	Reset Feedback Check
BIT 7	Reset Bit Overrun/Underrun

OUTPUT X'45' — TYPE 1 SCANNER-SCANNER CONTROL

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	Set Scanner Enabled
BIT 2	Reset Scanner Enabled
BIT 3	*
BIT 4	1=Set; 0=Rst Disable LIB Pos 1
BIT 5	1=Set; 0=Rst Disable LIB Pos 2
BIT 6	1=Set; 0=Rst Disable LIB Pos 3
BIT 7	1=Set; 0=Rst Disable LIB Pos 4
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

	OUTPUT X'46' — TYPE 1 SCANNER SET CHAR SVC PENDING, START SCANNER, RST L2 BIT REQUEST Gen Reg (R) — Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 * BYTE 1, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 *
	OUTPUT X'47' — TYPE 1 SCANNER-FORCE BIT SERVICE L2 REQUEST Gen Reg (R) — Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 Interface Address BIT 3 BIT 7 BIT 4 BYTE 1, BIT 0 BIT 5 BIT 1 BIT 6 BIT 2 BIT 7 BIT 3 BIT 8 BIT 4 * BIT 5 * BIT 6 * BIT 7 *

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

INPUT X'40'	TYPE 2 SCANNER - INTERFACE ADDRESS
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	1
BIT 5	0
BIT 6	Interface Address BIT 0
BIT 7	BIT 1
BYTE 1, BIT 0	BIT 2
BIT 1	BIT 3
BIT 2	BIT 4
BIT 3	BIT 5
BIT 4	BIT 6
BIT 5	BIT 7
BIT 6	BIT 8
BIT 7	0

OUTPUT X'40'	TYPE 2 SCANNER - INTERFACE ADDRESS
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	Interface Address BIT 0
BIT 7	BIT 1
BYTE 1, BIT 0	BIT 2
BIT 1	BIT 3
BIT 2	BIT 4
BIT 3	BIT 5
BIT 4	BIT 6
BIT 5	BIT 7
BIT 6	BIT 8
BIT 7	*

OUTPUT X'41'	TYPE 2 SCANNER - ADDRESS SUBSTITUTION CONTROL
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	SUB CTRL REG BIT 1
BIT 3	SUB CTRL REG BIT 2
BIT 4	SUB CTRL REG BIT 3
BIT 5	SUB CTRL REG BIT 4
BIT 6	*
BIT 7	*

OUTPUT X'42'	TYPE 2 SCANNER - UPPER SCAN LIMIT CONTROL
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	Scan Limit Select BIT 0 (note)
BIT 7	Scan Limit Select BIT 1 (note)

Note: 00=96; 01=8; 10=48; 11=16

INPUT X'43'	TYPE 2 SCANNER - CHECK REGISTER
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	LIB Pos 1 Bit Clock Check
BIT 1	LIB Pos 2 Bit Clock Check
BIT 2	LIB Pos 3 Bit Clock Check
BIT 3	LIB Pos 4 Bit Clock Check
BIT 4	LIB Pos 5 Bit Clock Check (Note)
BIT 5	LIB Pos 6 Bit Clock Check (Note)
BIT 6	LIB Select Check
BIT 7	ICW IN Reg Check
BYTE 1, BIT 0	ICW Work Reg Check
BIT 1	Priority Reg Avail Check
BIT 2	CCU OUTBUS Check
BIT 3	LINEADBUS Check
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0

Note: Not applicable for Type 2 Scanner-1

OUTPUT X'43'	TYPE 2 SCANNER - CONTROL
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Set Function
BIT 1	Reset Function
BIT 2	ICW Bit 38 (Display Request)
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	Disable LIB Pos 1
BYTE 1, BIT 0	Disable LIB Pos 2
BIT 1	Disable LIB Pos 3
BIT 2	Disable LIB Pos 4
BIT 3	Disable LIB Pos 5 (Note)
BIT 4	Disable LIB Pos 6 (Note)
BIT 5	Type 2 Scan L1 Request
BIT 6	Disable Interrupt Requests
BIT 7	*

Note: Not applicable for Type 2 Scanner-1

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

INPUT X'44' — TYPE 2 SCANNER-ICW Input Reg Bits 0-15 (Note)

Gen Reg (R)	Reg/Function (E)	
BYTE 0, BIT 0	ICW BIT 0	Stop Bit Check
BIT 1	BIT 1	Service Request
BIT 2	BIT 2	Char Over/Underrun
BIT 3	BIT 3	Modem Check
BIT 4	BIT 4	Rcvd Line Sig Det (PWI)
BIT 5	BIT 5	*
BIT 6	BIT 6	Program Flag
BIT 7	BIT 7	Pad Flag
BYTE 1, BIT 0	BIT 8	PDF BIT 0
BIT 1	BIT 9	BIT 1
BIT 2	BIT 10	BIT 2
BIT 3	BIT 11	BIT 3
BIT 4	BIT 12	BIT 4
BIT 5	BIT 13	BIT 5
BIT 6	BIT 14	BIT 6
BIT 7	BIT 15	BIT 7

Note: Autocall Interface Lines are shown in parenthesis.

OUTPUT X'44" — TYPE 2 SCANNER-ICW Bits 0-15

Gen Reg (R)	Reg/Function (E)	
BYTE 0, BIT 0	RST ICW BIT 0	Stop Bit Check
BIT 1	RST ICW BIT 1	Service Request
BIT 2	RST ICW BIT 2	Char Over/Underrun
BIT 3	RST ICW BIT 3	Modem Check
BIT 4	*	
BIT 5	ICW BIT 5	(must be 0)
BIT 6	BIT 6	Program Flag
BIT 7	BIT 7	Pad Flag
BYTE 1, BIT 0	BIT 8	PDF BIT 0
BIT 1	BIT 9	BIT 1
BIT 2	BIT 10	BIT 2
BIT 3	BIT 11	BIT 3
BIT 4	BIT 12	BIT 4
BIT 5	BIT 13	BIT 5
BIT 6	BIT 14	BIT 6
BIT 7	BIT 15	BIT 7

INPUT X'45' — TYPE 2 SCANNER-ICW Input Reg Bits 16-31

Gen Reg (R)	Reg/Function (E)	
BYTE 0, BIT 0	ICW BIT 16	LCD BIT 0
BIT 1	BIT 17	BIT 1
BIT 2	BIT 18	BIT 2
BIT 3	BIT 19	BIT 3
BIT 4	BIT 20	PCF BIT 0
BIT 5	BIT 21	BIT 1
BIT 6	BIT 22	BIT 2
BIT 7	BIT 23	BIT 3
BYTE 1, BIT 0	BIT 24	SDF BIT 0
BIT 1	BIT 25	BIT 1
BIT 2	BIT 26	BIT 2
BIT 3	BIT 27	BIT 3
BIT 4	BIT 28	BIT 4
BIT 5	BIT 29	BIT 5
BIT 6	BIT 30	BIT 6
BIT 7	BIT 31	BIT 7

OUTPUT X'45' — TYPE 2 SCANNER-ICW Bits 16-23

Gen Reg (R)	Reg/Function (E)	
BYTE 0, BIT 0	*	
BIT 1	*	
BIT 2	*	
BIT 3	*	
BIT 4	*	
BIT 5	*	
BIT 6	*	
BIT 7	*	
BYTE 1, BIT 0	ICW BIT 16	LCD BIT 0
BIT 1	BIT 17	BIT 1
BIT 2	BIT 18	BIT 2
BIT 3	BIT 19	BIT 3
BIT 4	BIT 20	PCF BIT 0
BIT 5	BIT 21	BIT 1
BIT 6	BIT 22	BIT 2
BIT 7	BIT 23	BIT 3

INPUT X'46' — TYPE 2 SCANNER DISPLAY REG (Note)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Clear to Send (ACR)
BIT 1	Ring Indicator (PND)
BIT 2	Data Set Ready (DLO)
BIT 3	Rcv Line Sig Det (PWI)
BIT 4	Receive Data Bit Buffer
BIT 5	Diagnostic Wrap Mode (COS)
BIT 6	Bit Service Request
BIT 7	0
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0

Note: Autocall Interface Lines are shown in parenthesis.

OUTPUT X'46' — TYPE 2 SCANNER-ICW Bits 24-33 & 45

Gen Reg (R)	Reg/Function (E)	
BYTE 0, BIT 0	ICW BIT 45	
BIT 1	*	
BIT 2	*	
BIT 3	*	
BIT 4	*	
BIT 5	*	
BIT 6	ICW BIT 24	SDF BIT 0
BIT 7	BIT 25	BIT 1
BYTE 1, BIT 0	BIT 26	BIT 2
BIT 1	BIT 27	BIT 3
BIT 2	BIT 28	BIT 4
BIT 3	BIT 29	BIT 5
BIT 4	BIT 30	BIT 6
BIT 5	BIT 31	BIT 7
BIT 6	BIT 32	BIT 8
BIT 7	BIT 33	BIT 9

INPUT X'47' — TYPE 2 SCANNER-ICW Input Reg Bits 32-45

Gen Reg (R)	Reg/Function (E)	
BYTE 0, BIT 0	ICW BIT 32	SDF BIT 8
BIT 1	BIT 33	SDF BIT 9
BIT 2	BIT 34	0-Unused
BIT 3	BIT 35	0-Unused
BIT 4	BIT 36	0-Unused
BIT 5	BIT 37	0-Unused
BIT 6	BIT 38	Display Request
BIT 7	BIT 39	0-Unused
BYTE 0, BIT 0	BIT 40	0-Unused
BIT 1	BIT 41	L2 Interrupt Pending
BIT 2	BIT 42	Priority Bit 1
BIT 3	BIT 43	Priority Bit 2
BIT 4	BIT 44	0-Unused
BIT 5	BIT 45	0-Unused
BIT 6	0	
BIT 7	0	

OUTPUT X'47' — TYPE 2 SCANNER-ICW Bits 34-43

Gen Reg (R)	Reg/Function (E)	
BYTE 0, BIT 0	*	
BIT 1	*	
BIT 2	*	
BIT 3	*	
BIT 4	*	
BIT 5	*	
BIT 6	ICW BIT 34	
BIT 7	BIT 35	
BYTE 1, BIT 0	BIT 36	
BIT 1	BIT 37	
BIT 2	*	
BIT 3	ICW BIT 39	
BIT 4	BIT 40	
BIT 5	BIT 41	Set L2 Interrupt Pending
BIT 6	BIT 42	Priority Bit 1
BIT 7	BIT 43	Priority Bit 2

With Extended Addressing Byte X is set to zero for input instructions, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

INPUT X'50' — TYPE 2 CA-INCWAR (For Channel Write, Write Break, & Write IPL)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	INCWAR BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	BIT 8
BIT 1	BIT 9
BIT 2	BIT 10
BIT 3	BIT 11
BIT 4	BIT 12
BIT 5	BIT 13
BIT 6	BIT 14
BIT 7	BIT 15

OUTPUT X'50' — TYPE 2 CA-INCWAR (For Channel Write, Write Break, & Write IPL)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	INCWAR BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	BIT 8
BIT 1	BIT 9
BIT 2	BIT 10
BIT 3	BIT 11
BIT 4	BIT 12
BIT 5	BIT 13
BIT 6	BIT 14
BIT 7	BIT 15

INPUT X'51' — TYPE 2 CA-OUTCWAR (For Channel Read)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	OUTCWAR BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	BIT 8
BIT 1	BIT 9
BIT 2	BIT 10
BIT 3	BIT 11
BIT 4	BIT 12
BIT 5	BIT 13
BIT 6	BIT 14
BIT 7	BIT 15

OUTPUT X'51' — TYPE 2 CA-OUTCWAR (For Channel Read)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	OUTCWAR BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	BIT 8
BIT 1	BIT 9
BIT 2	BIT 10
BIT 3	BIT 11
BIT 4	BIT 12
BIT 5	BIT 13
BIT 6	BIT 14
BIT 7	BIT 15

INPUT X'52' — TYPE 2 CA-CONTROL WORD BYTE COUNT (CWCNT)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	0
BIT 6	CWCNT BIT 0
BIT 7	BIT 1
BYTE 1, BIT 0	BIT 2
BIT 1	BIT 3
BIT 2	BIT 4
BIT 3	BIT 5
BIT 4	BIT 6
BIT 5	BIT 7
BIT 6	BIT 8
BIT 7	BIT 9

INPUT X'53' — TYPE 2 CA-SENSE REGISTER (CASNSR)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Command Reject
BIT 1	Intervention Required
BIT 2	Bus Out Check
BIT 3	Equipment Check
BIT 4	0
BIT 5	0
BIT 6	Not Initialized
BIT 7	Abort
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0

OUTPUT X'53' — TYPE 2 CA-SENSE REG (CASNR)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	Set Intervention Required
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	Set Abort
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

INPUT X'54'	TYPE 2 CA-STATUS REGISTER (CASTR)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Attention (ATT)
BIT 1	Status Modifier (SM)
BIT 2	0
BIT 3	Busy
BIT 4	Channel End (CE)
BIT 5	Device End (DE)
BIT 6	Unit Check (UC)
BIT 7	Unit Exception (UE)
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0

OUTPUT X'54'	TYPE 2 CA-STATUS REGISTER (CASTR)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Set Attention (ATT)
BIT 1	Set Status Modifier (SM)
BIT 2	*
BIT 3	Set Busy
BIT 4	Set Channel End (CE)
BIT 5	Set Device End (DE)
BIT 6	*
BIT 7	Set Unit Exception (UE)
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*

INPUT X'55'	TYPE 2 CA-CONTROL REGISTER (CACR)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Diagnostic Wrap Mode
BIT 1	Zero Count Override
BIT 2	INCWAR Valid
BIT 3	OUTCWAR Valid
BIT 4	Prog Requested L3 Interrupt
BIT 5	Prog Requested Abort/L3 Interrupt
BIT 6	Prog Requested Attention
BIT 7	CA Active
BYTE 1, BIT 0	Command Chaining
BIT 1	Write Break Command Remember
BIT 2	Channel Stop/Intf Disconnect
BIT 3	Selective/System Reset
BIT 4	0
BIT 5	I/O Read Command Remembrance
BIT 6	Type 2 CA 2 Selected
BIT 7	Type 2 CA 1 Selected

OUTPUT X'55'	TYPE 2 CA-CONTROL REGISTER (CACR)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	Set INCWAR Valid
BIT 3	Set OUTCWAR Valid
BIT 4	*
BIT 5	*
BIT 6	Set Prog Requested Attention
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*

INPUT X'56'	TYPE 2 CA-CHECK REGISTER (CACHKR)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Invalid CWAR Address
BIT 1	Invalid CW Format
BIT 2	Cycle Steal Address Check
BIT 3	CWAR/Data Buffer Check
BIT 4	CCU OUTBUS Check
BIT 5	CCU INBUS Check (cycle steal)
BIT 6	Chan Bus Out Check (Sense Bit 2)
BIT 7	0
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	Channel Bus In Check (Intf A)
BIT 5	Channel Bus In Check (Intf B)
BIT 6	0
BIT 7	0

OUTPUT X'56'	TYPE 2 CA-RESET CONTROL REGISTER
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	Reset INCWAR Valid
BIT 3	Reset OUTCWAR Valid
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*

OUTPUT X'57'	TYPE 2 CA-CHNL ADAPTER MODE REGISTER (CAMR)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Set IPL Attention
BIT 1	Set IPL Channel End
BIT 2	*
BIT 3	Set IPL Device End
BIT 4	Set IPL Unit Check
BIT 5	*
BIT 6	*
BIT 7	Set IPL Not-Initialized
BYTE 1, BIT 0	Set Type 2 CA L3 Interrupt
BIT 1	Set Program Abort
BIT 2	Reset Type 2 CA L1 Req
BIT 3	Reset Type 2 CA L3 Req
BIT 4	1 = Select Type 2 CA-1; 0 = Type 2 CA-2
BIT 5	Reset Selective/System Reset
BIT 6	Reset Channel Stop/Intf Disconnect
BIT 7	1 = Set; 0 = Rst Diagnostic Wrap Mode

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

INPUT X'58'	TYPE 2 CA-CHANNEL BUS OUT DIAG REG (CBODR)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Channel Bus Out BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	Parity Bit
BIT 1	0
BIT 2	Transfer Byte 1
BIT 3	Transfer Byte 2
BIT 4	Channel Intf A Enabled
BIT 5	Channel Intf B Enabled
BIT 6	CSAR Byte X, Bit 6
BIT 7	CSAR Byte X, Bit 7

OUTPUT X'58'	TYPE 2 CA-CHANNEL BUS OUT DIAGNOSTIC REG (CBODR)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Channel Bus Out BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	Parity Bit
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	Reset CA

INPUT X'59'	TYPE 2 CA-CYCLE STEAL ADDRESS REGISTER (CSAR)
Gen Reg (R)	Reg/Function (E)
BYTE X, BIT 6	CSAR BYTE X, BIT 6
BIT 7	BIT 7
BYTE 0, BIT 0	CSAR BYTE 0, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	CSAR BYTE 1, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7

INPUT X'5A'	TYPE 2 CA-DATA BUFFER (CADB)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Data Buffer BYTE 0, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	Data Buffer BYTE 1, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7

OUTPUT X'5A'	TYPE 2 CA-DATA BUFFER (CADB)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Data Buffer BYTE 0, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	Data Buffer BYTE 1, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7

INPUT X'5B'	TYPE 2 CA-TAG DIAGNOSTIC REGISTER (CTDR)
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Select Out/Hold Out Inbound
BIT 1	Address Out
BIT 2	Command Out
BIT 3	Service Out
BIT 4	Operational Out
BIT 5	Suppress Out
BIT 6	0
BIT 7	0
BYTE 1, BIT 0	Select Out Outbound
BIT 1	Request In
BIT 2	Operational In
BIT 3	Address In
BIT 4	Status In
BIT 5	Service In
BIT 6	0
BIT 7	Generate Busy

OUTPUT X'5B'	TYPE 2 CA-TAG DIAGNOSTIC REGISTER
Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Select Out/Hold Out Inbound
BIT 1	Address Out
BIT 2	Command Out
BIT 3	Service Out
BIT 4	Operational Out
BIT 5	Suppress Out
BIT 6	*
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*

With Extended Addressing Byte X is set to zero for input instructions, ignored for output instructions except where noted.
* = Bit positions unused - may be 1 or 0.

INPUT X'5C' TYPE 2 CA-COMMAND REGISTER (CMDR)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Test I/O X'00'
BIT 1	Write Command X'01'
BIT 2	Read Command X'02'
BIT 3	No-op Command X'03'
BIT 4	Sense Command X'04'
BIT 5	0
BIT 6	Write Break Command X'09'
BIT 7	0
BYTE 1, BIT 0	OUT Control Word
BIT 1	OUT STOP Control Word
BIT 2	IN Control Word
BIT 3	TIC Control Word
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	Write IPL Command X'05'

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions.
* = Bit Positions unused - may be 1 or 0.

INPUT X'60' — TYPE 1 CA-INITIAL SELECTION CNTRL

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Initial Selection Interrupt
BIT 1	Interface Disconnect
BIT 2	Selective Reset
BIT 3	Channel Bus Out Check
BIT 4	0
BIT 5	Stacked Initial Status
BIT 6	NSC Status Byte Cleared
BIT 7	System Reset
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0

OUTPUT X'60' — TYPE 1 CA-RESET INITIAL SELECTION

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*

INPUT X'61' — TYPE 1 CA-INITIAL SELECTION ADDRESS and COMMAND

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Address Byte BIT 0 (Initial Sel)
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	I/O Cmd Byte BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7

INPUT X'62' — TYPE 1 CA-DATA/STATUS CNTRL

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Outbound Data Transfer Seq
BIT 1	Inbound Data Transfer Seq
BIT 2	ESC Final Status Transfer Seq
BIT 3	NSC Chan End Status Transfer Seq
BIT 4	NSC Final Status Transfer Seq
BIT 5	Channel Stop/Intf Disconnect
BIT 6	Suppress Out Monitor Interrupt
BIT 7	Prog Requested Interrupt
BYTE 1, BIT 0	Channel Bus Out Check
BIT 1	Selective Reset
BIT 2	Suppress Out
BIT 3	Stacked Ending Status
BIT 4	I/O Command Chaining
BIT 5	Transferred Byte Count BIT 1
BIT 6	BIT 2
BIT 7	BIT 3

OUTPUT X'62' — TYPE 1 CA-DATA/STATUS CNTRL (Note)

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	1 = Set; 0 = Rst Outbound Data Transfer Seq
BIT 1	1 = Set; 0 = Rst Inbound Data Transfer Seq
BIT 2	1 = Set; 0 = Rst ESC Final Status Transfer Seq
BIT 3	1 = Set; 0 = Rst NSC Chan End Status Transfer Seq
BIT 4	1 = Set; 0 = Rst NSC Final Status Transfer Seq
BIT 5	Reset Initial Selection
BIT 6	Reset Data/Status Interrupt
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	Reset Suppress Out Monitor Interrupt
BIT 2	*
BIT 3	Set Suppressible Status
BIT 4	Set ESC Test I/O Status Available
BIT 5	*
BIT 6	Request Byte Count BIT 1
BIT 7	BIT 2

Note: This output resets a program requested interrupt

INPUT X'63' — TYPE 1 CA-ADDR. & ESC STATUS BYTES

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Addr. Byte BIT 0 (Data/Stat Transfer)
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	ESC Status Byte BIT 0 (Attn)
BIT 1	BIT 1 (SM)
BIT 2	BIT 2 (CUE)
BIT 3	BIT 3 (Busy)
BIT 4	BIT 4 (CE)
BIT 5	BIT 5 (DE)
BIT 6	BIT 6 (UC)
BIT 7	BIT 7 (UE)

OUTPUT X'63' — TYPE 1 CA-ADDR. & ESC STATUS BYTES

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Addr. Byte BIT 0 (Data/Stat Transfer)
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	ESC Status Byte BIT 0 (Attn)
BIT 1	BIT 1 (SM)
BIT 2	BIT 2 (CUE)
BIT 3	BIT 3 (Busy)
BIT 4	BIT 4 (CE)
BIT 5	BIT 5 (DE)
BIT 6	BIT 6 (UC)
BIT 7	BIT 7 (UE)

With Extended Addressing, Byte X is set to zero fro input instructions, ignored for output instructions.
* = Bit Positions unused - may be 1 or 0.

INPUT X'64'	TYPE 1 CA-DATA BUFFER BYTES 1,2	OUTPUT X'64'	TYPE 1 CA-DATA BUFFER BYTES 1,2
Gen Reg (R)	Reg/Function (E)	Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Data Buffer Byte 1, BIT 0	BYTE 0, BIT 0	Data Buffer Byte 1, BIT 0
BIT 1	BIT 1	BIT 1	BIT 1
BIT 2	BIT 2	BIT 2	BIT 2
BIT 3	BIT 3	BIT 3	BIT 3
BIT 4	BIT 4	BIT 4	BIT 4
BIT 5	BIT 5	BIT 5	BIT 5
BIT 6	BIT 6	BIT 6	BIT 6
BIT 7	BIT 7	BIT 7	BIT 7
BYTE 1, BIT 0	Data Buffer Byte 2, BIT 0	BYTE 1, BIT 0	Data Buffer Byte 2, BIT 0
BIT 1	BIT 1	BIT 1	BIT 1
BIT 2	BIT 2	BIT 2	BIT 2
BIT 3	BIT 3	BIT 3	BIT 3
BIT 4	BIT 4	BIT 4	BIT 4
BIT 5	BIT 5	BIT 5	BIT 5
BIT 6	BIT 6	BIT 6	BIT 6
BIT 7	BIT 7	BIT 7	BIT 7

INPUT X'65'	TYPE 1 CA-DATA BUFFER BYTES 3,4	OUTPUT X'65'	TYPE 1 CA-DATA BUFFER BYTES 3,4
Gen Reg (R)	Reg/Function (E)	Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Data Buffer Byte 3, BIT 0	BYTE 0, BIT 0	Data Buffer Byte 3, BIT 0
BIT 1	BIT 1	BIT 1	BIT 1
BIT 2	BIT 2	BIT 2	BIT 2
BIT 3	BIT 3	BIT 3	BIT 3
BIT 4	BIT 4	BIT 4	BIT 4
BIT 5	BIT 5	BIT 5	BIT 5
BIT 6	BIT 6	BIT 6	BIT 6
BIT 7	BIT 7	BIT 7	BIT 7
BYTE 1, BIT 0	Data Buffer Byte 4, BIT 0	BYTE 1, BIT 0	Data Buffer Byte 4, BIT 0
BIT 1	BIT 1	BIT 1	BIT 1
BIT 2	BIT 2	BIT 2	BIT 2
BIT 3	BIT 3	BIT 3	BIT 3
BIT 4	BIT 4	BIT 4	BIT 4
BIT 5	BIT 5	BIT 5	BIT 5
BIT 6	BIT 6	BIT 6	BIT 6
BIT 7	BIT 7	BIT 7	BIT 7

INPUT X'66'	TYPE 1 CA-NSC STATUS BYTE	OUTPUT X'66'	TYPE 1 CA-NSC STATUS BYTE
Gen Reg (R)	Reg/Function (E)	Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	NSC Status Byte BIT 0 (Attn)	BYTE 0, BIT 0	*
BIT 1	NSC Status Byte BIT 1 (SM)	BIT 1	*
BIT 2	0	BIT 2	*
BIT 3	0	BIT 3	*
BIT 4	NSC Status Byte BIT 4 (CE)	BIT 4	*
BIT 5	NSC Status Byte BIT 5 (DE)	BIT 5	*
BIT 6	NSC Status Byte BIT 6 (UC)	BIT 6	*
BIT 7	NSC Status Byte BIT 7 (UE)	BIT 7	*
BYTE 1, BIT 0	0	BYTE 1, BIT 0	Set NSC Status Byte BIT 0 (Attn)
BIT 1	0	BIT 1	Set NSC Status Byte BIT 1 (SM)
BIT 2	0	BIT 2	*
BIT 3	0	BIT 3	*
BIT 4	0	BIT 4	Set NSC Status Byte BIT 4 (CE)
BIT 5	0	BIT 5	Set NSC Status Byte BIT 5 (DE)
BIT 6	0	BIT 6	Set NSC Status Byte BIT 6 (UC)
BIT 7	0	BIT 7	Set NSC Status Byte BIT 7 (UE)

INPUT X'67'	TYPE 1 CA-CONTROLS	OUTPUT X'67'	TYPE 1 CA-CONTROLS
Gen Reg (R)	Reg/Function (E)	Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	NSC Address Byte BIT 0 (Hdw)	BYTE 0, BIT 0	*
BIT 1	BIT 1	BIT 1	*
BIT 2	BIT 2	BIT 2	*
BIT 3	BIT 3	BIT 3	*
BIT 4	BIT 4	BIT 4	CA Diagnostic Reset
BIT 5	BIT 5	BIT 5	*
BIT 6	BIT 6	BIT 6	*
BIT 7	BIT 7	BIT 7	*
BYTE 1, BIT 0	Channel Bus In Check	BYTE 1, BIT 0	Set Suppress Out Monitor Interrupt
BIT 1	In/Out Instruction Accept Check	BIT 1	Set Prog Requested Interrupt
BIT 2	CCU OUTBUS Check	BIT 2	Reset Type 1 CA L1 Checks
BIT 3	Local Store Check	BIT 3	Reset System Reset/NSC Address Active
BIT 4	Channel Interface Enabled	BIT 4	Set Allow Channel Interface Enable
BIT 5	NSC Address Active	BIT 5	Set ESC Operational
BIT 6	0	BIT 6	Set ESC Command Free
BIT 7	0	BIT 7	Set Allow Channel Interface Disable

With Extended Addressing, Byte X is set to zero for input instruction, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

INPUT X'70' — STORAGE SIZE INSTALLED

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	144,176,208, or 240K
BIT 1	80,112,208, or 240K
BIT 2	48,112,176, or 240K
BIT 3	16,48,80,112,144,176,208, or 240K
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0

OUTPUT X'70' — HARDSTOP

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	*
BIT 7	*

INPUT X'71' — PANEL ADDR/DATA ENTRY DIGITS

Gen Reg (R)	Reg/Function (E)
BYTE X, BIT 6	DIGIT A, BIT 2 (with EA only)
BIT 7	BIT 3 (with EA only)
BYTE 0, BIT 0	DIGIT B, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	DIGIT C, BIT 0
BIT 5	BIT 1
BIT 6	BIT 2
BIT 7	BIT 3
BYTE 1, BIT 0	DIGIT D, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	DIGIT E, BIT 0
BIT 5	BIT 1
BIT 6	BIT 2
BIT 7	BIT 3

EA = Extended Addressing

OUTPUT X'71' — DISPLAY REGISTER 1 (DR 1) (Note)

Gen Reg (R)	Reg/Function (E)
BYTE X, BIT 6	DR1 BYTE X, BIT 6 (with EA only)
BIT 7	BIT 7
BYTE 0, BIT 0	DR1 BYTE 0, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	DR1 BYTE 1, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7

Note: Sets Program Display Light EA = Extended Addressing

INPUT X'72' — PANEL DISPLAY/FUNCTION SELECT SWITCH CONTROLS

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	Storage Address
BIT 4	Register Address
BIT 5	0
BIT 6	0
BIT 7	0
BYTE 1, BIT 0	0
BIT 1	Function Select 1
BIT 2	Function Select 2
BIT 3	Function Select 3
BIT 4	Function Select 4
BIT 5	Function Select 5
BIT 6	Function Select 6
BIT 7	0

OUTPUT X'72' — DISPLAY REGISTER 2 (DR2) (Note)

Gen Reg (R)	Reg/Function (E)
BYTE X, BIT 6	DR2 BYTE X, BIT 6 (with EA only)
BIT 7	BIT 7 (with EA only)
BYTE 0, BIT 0	DR2 BYTE 0, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	DR2 BYTE 1, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7

Note: Sets Program Display Light EA = Extended Addressing

INPUT X'73' — INSERT KEY

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	Key Bit 0
BIT 6	Key Bit 1
BIT 7	Key Bit 2

OUTPUT X'73' — SET KEY

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	SKA Bit 0
BIT 1	SKA Bit 1
BIT 2	SKA Bit 2
BIT 3	SKA Bit 3
BIT 4	SKA Bit 4 or PKA Bit 0
BIT 5	SKA Bit 5 or PKA Bit 1
BIT 6	SKA Bit 6 or PKA Bit 2
BIT 7	*
BYTE 1, BIT 0	*
BIT 1	*
BIT 2	*
BIT 3	Key Address Select 1 = SKA; 0 = PKA
BIT 4	Set Key
BIT 5	Key Bit 0
BIT 6	Key Bit 1
BIT 7	Key Bit 2

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions except where noted.
* = Bit positions unused - may be 1 or 0.

INPUT X'74' — LAGGING ADDRESS REGISTER (LAR)

Gen Reg (R)	Reg/Function (E)
BYTE X, BIT 6	LAR BYTE X, BIT 6 (with EA only)
BIT 7	BIT 7 (with EA only)
BYTE 0, BIT 0	LAR BYTE 0, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7
BYTE 1, BIT 0	LAR BYTE 1, BIT 0
BIT 1	BIT 1
BIT 2	BIT 2
BIT 3	BIT 3
BIT 4	BIT 4
BIT 5	BIT 5
BIT 6	BIT 6
BIT 7	BIT 7

EA = Extended Addressing

INPUT X'76' — ADAPTER INTERRUPT REQUESTS GROUP 1

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	0
BIT 1	Type 2 Scan-1 (or Type 1 Scan) L1
BIT 2	Type 2 Scan-2 L1
BIT 3	Type 2 Scan-3 L1
BIT 4	Type 2 Scan-4 L1
BIT 5	Type 2 CA-1 (or Type 1 CA) L1
BIT 6	Type 2 CA-2 L1
BIT 7	0
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0

INPUT X'77' — ADAPTER INTERRUPT REQUESTS GROUP 2

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	0
BIT 1	Type 2 Scan (or Type 1 Scan) L2
BIT 2	0
BIT 3	0
BIT 4	0
BIT 5	0
BIT 6	0
BIT 7	0
BYTE 1, BIT 0	0
BIT 1	0
BIT 2	Type 2 CA-2 L3
BIT 3	Type 1 CA Data/Status L3
BIT 4	Type 2 CA-1 (or Type 1 CA) L3
BIT 5	0
BIT 6	0
BIT 7	0

OUTPUT X'77' — MISCELLANEOUS CONTROL

Gen Reg (R)	Reg/Function (E)
BYTE 0, BIT 0	Reset IPL L1
BIT 1	Reset CCU Checks
BIT 2	Reset Panel Interrupt Reg L3
BIT 3	*
BIT 4	*
BIT 5	*
BIT 6	Set Diagnostic L2 (note)
BIT 7	Reset Diagnostic L2 (note)
BYTE 1, BIT 0	*
BIT 1	Reset Interval Timer L3
BIT 2	Reset PCI L3
BIT 3	*
BIT 4	Reset Address Compare L1
BIT 5	Reset Program Checks L1
BIT 6	Reset PCI L4
BIT 7	Reset SVC L4

Note: Ignored if not in Test Mode.

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions except where noted.
* = Bit positions unused - may be 1 or 0.

	OUTPUT X'78' — FORCE CCU CHECKS (Ignored if not in Test Mode) Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 — Complement Byte 0,1 -- Bit 0 BIT 1 — Complement Byte 0,1 -- Bit 1 BIT 2 — Complement Byte 0,1 -- Bit 2 BIT 3 — Complement Byte 0,1 -- Bit 3 BIT 4 — Complement Byte 0,1 -- Bit 4 BIT 5 — Complement Byte 0,1 -- Bit 5 BIT 6 — Complement Byte X,0,1 -- Bit 6 BIT 7 — Complement Byte X,0,1 -- Bit 7 BYTE 1, BIT 0 — Complement Storage Parity BIT 1 — Complement Z Bus Parity BIT 2 — A-Reg Check BIT 3 — INDATA Bus Check BIT 4 — * BIT 5 — * BIT 6 — * BIT 7 — *
INPUT X'79' — UTILITY Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 — 0 BIT 1 — 0 BIT 2 — 0 BIT 3 — 0 BIT 4 — 0 BIT 5 — 0 BIT 6 — Prog Level 5 C Condition BIT 7 — Prog Level 5 Z Condition BYTE 1, BIT 0 — Prog Level 2 Interrupted (note) BIT 1 — Prog Level 3 Interrupted (note) BIT 2 — Prog Level 4 Interrupted (note) BIT 3 — Prog Level 5 Interrupted (note) BIT 4 — 0 BIT 5 — 0 BIT 6 — 0 BIT 7 — IPL Escape Control NOTE: This bit = 0 if not in Level 1 or if entered immed. after exiting Level 1.	OUTPUT X'79' — UTILITY Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 — * BIT 1 — * BIT 2 — Set IPL BIT 3 — * BIT 4 — * BIT 5 — Inhibit Prog Level 5 C, Z Replacement BIT 6 — Prog Level 5 C Condition BIT 7 — Prog Level 5 Z Condition BYTE 1, BIT 0 — Reset CCU Check Hard Stop Mode BIT 1 — Reset Load Light BIT 2 — Set Test Mode BIT 3 — Reset Test Mode BIT 4 — Set Bypass CCU Check Stop Mode (note) BIT 5 — Reset Bypass CCU Check Stop Mode (note) BIT 6 — Scope Sync Pulse 1 BIT 7 — Scope Sync Pulse 2 NOTE: Ignored if not in Test Mode.
INPUT X'78' — BSC CRC REGISTER Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 — BSC CRC BIT 0 BIT 1 — BIT 1 BIT 2 — BIT 2 BIT 3 — BIT 3 BIT 4 — BIT 4 BIT 5 — BIT 5 BIT 6 — BIT 6 BIT 7 — BIT 7 BYTE 1, BIT 0 — BIT 8 BIT 1 — BIT 9 BIT 2 — BIT 10 BIT 3 — BIT 11 BIT 4 — BIT 12 BIT 5 — BIT 13 BIT 6 — BIT 14 BIT 7 — BIT 15	

With Extended Addressing, Byte X is set to zero for input instruction, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

	OUTPUT X'7C' SET PCI L3 Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 * BYTE 1, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 *
INPUT X'7D' CCU CHECK REGISTER Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 Byte X Check BIT 1 Byte 0 Check BIT 2 Byte 1 Check BIT 3 Program Check in Level 1 BIT 4 SAR Check BIT 5 SDR Check BIT 6 OP Reg Check BIT 7 INDATA Bus Check BYTE 1, BIT 0 Cycle Counter Check BIT 1 0 BIT 2 0 BIT 3 0 BIT 4 0 BIT 5 0 = No CCU Checks; 1-CCU Check(s) BIT 6 TYPE 2 Attach Base Clock Check BIT 7 CCU Clock Check	OUTPUT X'7D' SET PCI L4 Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 * BYTE 1, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 *
INPUT X'7E' CCU INTERRUPT REQUESTS GROUP 1 Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 0 BIT 1 0 BIT 2 0 BIT 3 0 BIT 4 0 BIT 5 0 BIT 6 0 BIT 7 0 BYTE 1, BIT 0 Address Compare Interrupt L1 BIT 1 Address Exception (note) BIT 2 In/Out Check (note) BIT 3 Protection Check (note) BIT 4 Invalid Op Check (note) BIT 5 0 BIT 6 IPL L1 BIT 7 0 Note: Prog Check	OUTPUT X'7E' SET MASK BITS Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 * BYTE 1, BIT 0 * BIT 1 Adapter Requests L1 (note) BIT 2 Program Level 2 BIT 3 Program Level 3 BIT 4 Program Level 4 BIT 5 Program Level 5 BIT 6 * BIT 7 * Note: Ignored if not in Test Mode
INPUT X'7F' CCU INTERRUPT REQUESTS GROUP 2 Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 Diagnostic L2 BIT 1 0 BIT 2 0 BIT 3 0 BIT 4 0 BIT 5 0 BIT 6 Panel Interrupt Request L3 BIT 7 PCI L4 BYTE 1, BIT 0 0 BIT 1 0 BIT 2 0 BIT 3 0 BIT 4 0 BIT 5 Interval Timer L3 BIT 6 PCI L3 BIT 7 SVC L4	OUTPUT X'7F' RESET MASK BITS Gen Reg (R) / Reg/Function (E) BYTE 0, BIT 0 * BIT 1 * BIT 2 * BIT 3 * BIT 4 * BIT 5 * BIT 6 * BIT 7 * BYTE 1, BIT 0 * BIT 1 Adapter Requests Level 1 (note) BIT 2 Program Level 2 BIT 3 Program Level 3 BIT 4 Program Level 4 BIT 5 Program Level 5 BIT 6 * BIT 7 * Note: Ignored if not in Test Mode

With Extended Addressing, Byte X is set to zero for input instructions, ignored for output instructions.
* = Bit positions unused - may be 1 or 0.

Appendix D: Line Interface Bases and Line Sets

Four different Line Interface Base (LIB) types are available for the 3705 to meet the needs of a wide variety of line and terminal types. Each LIB type operates identically and is controlled by the communication scanner that it is attached to. However, the design of each LIB type differs in order to support the many line sets and line configurations that can be attached to them.

LIB Type 1 Voltage Interface, EIA

The LIB Type 1 provides for the attachment of up to eight of the following line set types in any combination.

Line Set 1A. (Low-Speed External Modem)
This line set provides for the attachment of two start-stop communication lines at speeds up to 1200 bps with external modems. The control program must condition these line interfaces for business machine clock control.

Line Set 1C. (Low-Speed Local Attachment)
This line set provides for the local attachment of two start-stop communication lines at speeds up to 1200 bps. The control program must condition these line interfaces for business machine clock control. Modems are not required.

Line Set 1D. (Medium-Speed External Modem)
This line set provides for the attachment of two start-stop or synchronous communication lines at speeds up to 7200 bps with external modems. The control program may condition either one or both of these line interfaces for modem clock or business machine clock control (if the line speed does not exceed 2400 bps).

Line Set 1E. (Autocall Adapter)
This line set provides for the attachment of two external Automatic Calling Units. The control program may condition the line interface for either modem clock or business machine clock control (if the line speed does not exceed 2400 bps). The two interfaces are completely independent. Each interface and attached Automatic Calling Unit can be connected by external cabling with any of the line interfaces provided by Line Sets 1A, 1D, and 1G.

Line Set 1F. (Medium-Speed Local Attachment)
This line set provides for the local attachment of two half-duplex terminals at speeds up to 2400 bps (limited by business machine clock speed). The control program must condition these line interfaces for business machine clock control. No external modem is required.

Line Set 1G. (High-Speed External Modem)
This line set provides for the attachment of one synchronous communication line that has a digital interface for attachment to external modems for 19,200 bps, 40,800 bps, or 50,000 bps wideband communication facilities. The control program must condition this line interface for modem clock control. This line set may not be installed with a Type 1 Communication Scanner.

LIB Type 2 Single-Current Telegraph

The LIB Type 2 provides for the attachment of up to eight line sets for telegraph termination.

Line Set 2A. (Telegraph Single Current)
This line set provides for the attachment of two telegraph lines at speeds up to 100 bps. The control program must condition these line interfaces for business machine clock control.

LIB Type 3 IBM Type-1 Line Adapter

The LIB Type 3 provides for the attachment of up to six of the following limited distance line sets in any combination.

Line Set 3A. (IBM Limited Distance Type-1 Line Adapter, 2-wire)
This line set provides for the attachment of two half-duplex two-wire start-stop lines at speeds up to 134.5 bps. The control program must condition these line interfaces for business machine clock control. This line set includes two IBM limited distance Type-1 line adapters; therefore, no external modem is required.

Line Set 3B. (IBM Limited Distance Type-1 Line Adapter, 4-wire)
This line set provides for the attachment of two four-wire start-stop communication lines at speeds up to 134.5 bps. The control program must condition these line interfaces for business machine clock control. This line set includes two IBM limited distance Type-1 line adapters; therefore, no external modem is required.

LIB Type 4 IBM Type-2 Line Adapter

The LIB Type 4 provides for the attachment of either one or two limited distance or leased-line line sets in any combination.

Line Set 4A. (IBM Limited Distance Type-2 Line Adapter)
This line set provides for the attachment of two 2-wire limited distance start-stop lines at speeds up to 600 bps. The control program must condition these line interfaces for business machine clock control. This line set includes two IBM limited distance Type-2 line adapters; therefore, no external modem is required.

Line Set 4B. (IBM Leased Line Line Adapter, 2-wire)
This line set provides for the attachment of two half-duplex two-wire leased start-stop communication lines at speeds up to 600 bps. The control program must condition these line interfaces for business machine clock control. This line set includes two leased-line adapters; therefore, no external modem is required.

Line Set 4C. (IBM Leased-Line Adapter, 4-wire)
This line set provides for the attachment of two four-wire leased start-stop communication lines at speeds up to 600 bps. The control program must condition these line interfaces for business machine clock control. This line set includes two leased-line adapters; therefore, no external modem is required.

Glossary

Access method. A data management technique for transferring data between storage and an input/output device.

Addressing. The means whereby the originator or control unit selects the teleprocessing device to which it is going to send a message.

Address substitution. A Type 2 Attachment Base feature that modifies the scan counter output to replace certain pairs of interface addresses with one other address.

Attachment base. A communications controller hardware feature that provides the interface controls to the Central Control Unit for the 3705 adapters.

Bit service. The process of character assembly or disassembly.

Buffer. A temporary storage area for data.

Central Control Unit. The communications controller hardware unit that contains the circuits and data flow paths needed to execute the instruction set and to control storage and the attached adapters.

Channel adapter (CA). A communications controller hardware unit that provides attachment of the controller to a System/360 or System/370 channel.

Character assembly. The process by which bits are put together to form characters as the bits arrive on a communication line. In the communications controller, character assembly is performed either by the control program or by the communication scanner, depending on the type of scanner installed.

Character disassembly. The process by which characters are broken down into bits for transmission over a communication line. In the communications controller, character disassembly is performed either by the control program or by the communication scanner, depending on the type of scanner installed.

Character service. The process by which a character is moved to a buffer from the storage area where it was assembled.

Communication scanner. A communications controller hardware unit that provides the interface between line interface bases and the Central Controller Unit. The communication scanner monitors the communication lines for service requests.

Component. An independently addressable part of a station that performs either an input or an output function for the terminal, but not both.

Control Character. (See Line Control Character.)

Cycle steal. The process by which the Type 2 Channel Adapter acquires machine cycles from the control program for data transfer.

Device. (See Teleprocessing device.)

Extended Addressing. The addition of two high order bits to the basic addressing scheme to permit installation of larger storage capacities.

Hard stop. Immediate termination of controller operation without the execution of orderly closedown procedures.

Hardware check. A failure in a hardware unit that halts operation.

Host processor. The central processing unit to which the communications controller is attached by a channel and that executes the teleprocessing access method that supports the controller.

Interrupt. A break in the normal sequence of instruction execution. It causes an automatic transfer to a preset storage location where appropriate action is taken.

Interrupt priority. The order in which the control program processes interrupts received sumultaneously from two or more communication lines.

Line control character. A special character that controls transmission of data over a communication line. For example, line control characters are used to start or end a transmission, to cause transmission-error checking to be performed, and to indicate whether a terminal has data to send or is ready to receive data.

Line group. A group of communication lines by which stations supported by the same line-control discipline are connected to the communications controller.

Line interface base (LIB). A communications controller hardware unit that provides for the attachment of up to 16 communication lines to the controller.

Line scanner. (see Communication scanner.)

Line set. A communications controller hardware unit through which one or two lines are attached to a line interface base.

Program check. An error in a program that suspends execution.

Subchannel. The channel facility required for sustaining a single I/O operation.

Teleprocessing. A form of information handling in which a data processing system utilizes communication facilities.

Upper Scan Limit. A Type 2 Communication Scanner feature that allows the control program to limit the maximum number of lines that a particular scanner addresses.

Index

M

N

O

READER'S COMMENT FORM

IBM 3705 Communications Controller
Principles of Operation

Order No. GC30-3004-0

- How did you use this publication?

  As a reference source ☐
  As a classroom text ☐
  As ..... ☐

- Based on your own experience, rate this publication . . .

As a reference source:	.....	....	.....	.....	.....
	Very Good	Good	Fair	Poor	Very Poor
As a text:	.....	....	.....	.....	.....
	Very Good	Good	Fair	Poor	Very Poor

- What is your occupation? .....

- We would appreciate your other comments; please give specific page and line references where appropriate. If you wish a reply, be sure to include your name and address.

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS, PLEASE . . .

Your answers to the questions on the back of this form, together with your comments, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in using your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold Fold

FIRST CLASS
PERMIT NO. 569
RESEARCH TRIANGLE PARK
NORTH CAROLINA

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 12275
Research Triangle Park
North Carolina 27709

Attention: Publications Center, Dept. E01

Fold Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

Cut Along Line

IBM 3705 Communications Controller Principles of Operation Printed in U.S.A. GC30-3004-0