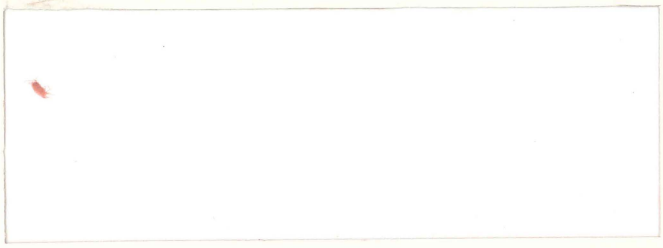
KD11-B processor maintenance manual

11/05.

pdp11

digital



KD11-B processor maintenance manual

11/05

digital equipment corporation · maynard, massachusetts

1st Edition, January 1975

Printed in U.S.A.

CONTENTS

CONTENTS (Cont)

ILLUSTRATIONS

TABLES

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This manual describes the KD11-B Processor, which is the basic component of the PDP-11/05/10 and PDP-11/05S computer systems. The processor is connected to the Unibus as a subsystem and controls time allocation of the Unibus for peripherals, and performs arithmetic and logic operations through instruction decoding and execution. The information contained in this manual pertains primarily to the processor itself.

Table 1-1 lists other manuals that are necessary for a complete understanding of the basic PDP-11/05 or PDP-11/05S system.

Table 1-1
Related Documents

Title	Document Number	Remarks
PDP-11/05/10 Computer Manual	DEC-11-H05AA-B-D	Describes overall PDP-11/05/10 computer and includes sections on installation, operation, and programming.
PDP-11/05S System Manual	DEC-11-H05SS-A-D	Describes overall PDP-11/05S system and includes sections on installation, operation, and programming.
BA11-K Mounting Box Manual	DEC-11-HBKEF-A-D	Describes power system for KD11-B when it is mounted in the 10-1/2 in. mounting box.

1.2 ORGANIZATION

The discussion of the KD11-B Processor is divided into four major sections: microprogramming (Chapter 2), console description (Chapter 3), KD11-B detailed description (Chapter 4), and KD11-B and console maintenance (Chapter 5).

Chapter 2 discusses the processor first then briefly covers the conventional method of implementing the instruction set. The remainder of the chapter is devoted to a discussion of microprogrammed implementation, the basic microprogram memory, and the structure of the microprogram word.

Chapter 3 provides a general and a detailed description of the console logic. The general description is keyed to the block diagram level, the detailed description covers the theory of operation of the console logic. The function and use of the console controls are discussed in the PDP-11/05S System Manual.

Chapter 4 describes the logic and physical implementation of the KD11-B data path (DP), data path control (DPC), Unibus control, serial communications line (SCL), and the line clock. Extensive use is made of bipolar, medium, and large scale integrated circuits in the processor.

Chapter 5 describes techniques for isolating and repairing failures in the KD11-B and the console. The basic procedures are aimed at differentiating between failures in the processor and the remainder of the computer.

CHAPTER 2
MICROPROGRAM CONTROL

2.1 INTRODUCTION

This chapter describes the microprogram control implemented in the KD11-B processor. The flow notation used in the microprogram flow section of the prints is described in Paragraph 2.5.1. The difference between microprogram control and conventional control in a computer processor is described in Paragraph 2.2. Paragraph 2.3 describes the KD11-B control store (CS) structure; Paragraph 2.4 describes the technique of branching within microroutines in the CS; and Paragraph 2.5 describes the microprogram flow, including instruction interpretation, Unibus control coordination, interrupts, traps, and console functions.

2.2 MICROPROGRAMMED VERSUS CONVENTIONAL CONTROL

The control section of a conventional computer is a complex collection of specialized logic circuits. These circuits generate the timing signals that constitute the major and minor time states of a machine cycle. During each time state, these control signals configure the data path (DP), determine function performed within the arithmetic/logic units (ALU), influence the Unibus control (BC), etc. Major disadvantages associated with this conventional approach are its complexity, the large amount of logic required, its inflexibility, and difficulty of making modifications.

A microprogrammed processor such as the KD11-B results in a reduction in the amount and complexity of the control logic, while facilitating a systematically implemented and easily modified control section. Basically, a microprogram involves the execution of a sequence of microsteps from the control store (ROMs). Execution of a microstep causes the assertion of a set of control signals specified in the control store word associated with that microstep. By executing appropriate sequences of microsteps (known as a microroutine), the KD11-B can be made to interrupt PDP-11 instructions. Other functions such as console functions, interrupts, and traps are also accomplished by specialized microroutines.

2.3 CONTROL STORE

Figure 2-1 shows the format of the KD11-B control store (CS) word. There are 256 such words, each having the same fields. The fields, the possible values they may contain, and the significance of each value are described in Table 2-1. The CS is shown on prints CONF and CONG.

An explanation of the notation will aid in relating the CS word to the reset of the print set. Each field within the CS has been given a name (e.g., BUT, BRG, ALG, . . . ,ALU, NXT). These field names are used throughout documentation of the microprogram.

The signal coming from each bit is named according to the convention used in the print set. Several signals may be associated with a single field (e.g., the BUT field controls four signals: CONG BUT 01 L, CONG BUT 00 L, CONG BUT 02 L, and CONG BUT 03 L).

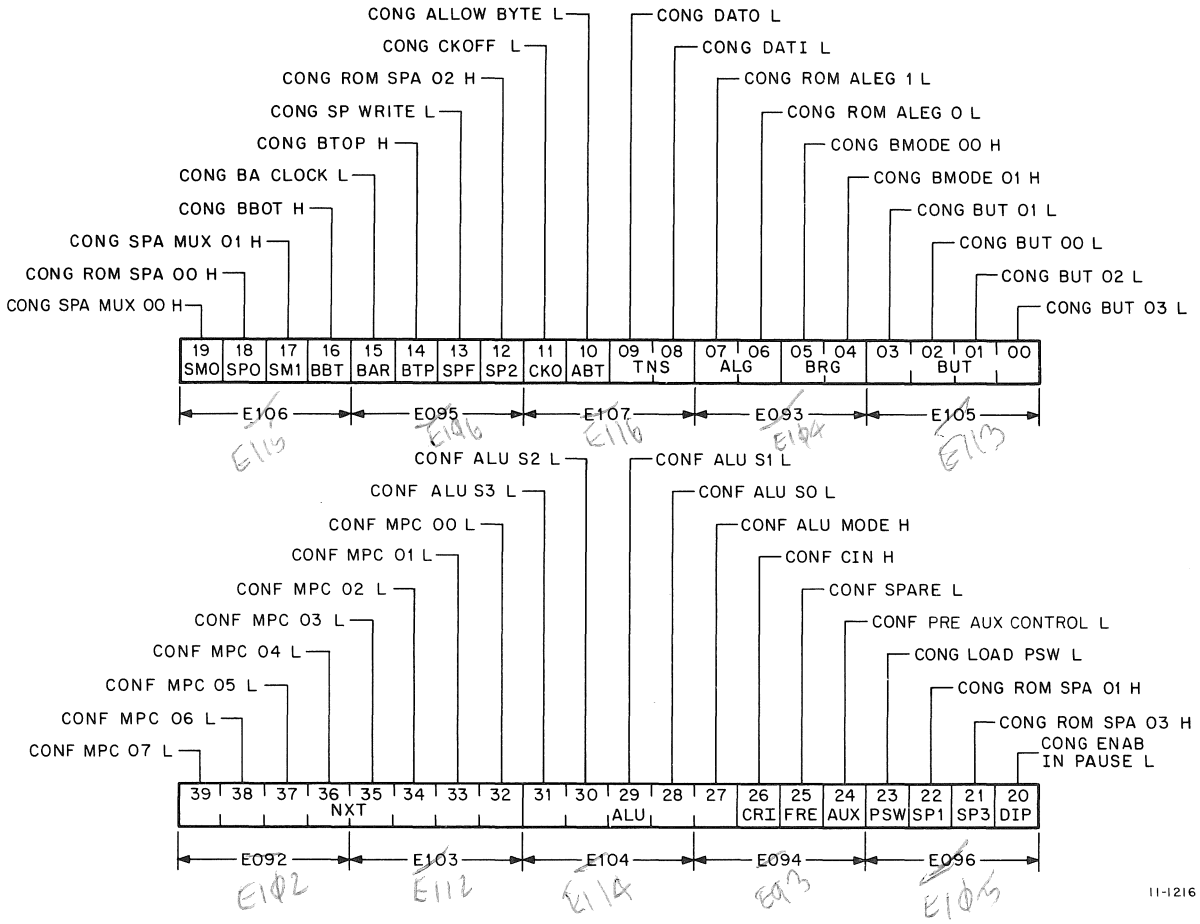


Figure 2-1 Control Store Word Bit and Field Format

Table 2-1
KD11-B Control Store Fields

Field	Description
BUT	Branch on microtest. The BUT field has two uses: a) specify microprogram conditional branches, and b) as an encoded miscellaneous field. The values this field can assume are grouped by these two uses. Branching within the microprogram is accomplished by wiring conditional signals with the open-collector outputs of the NXT field of the CS. Each BUT condition has the minimum number of control bits required. This makes the range of branching restrictive, but it minimizes logic (print CONE). Table 2-2 lists the microstep in which each BUT is performed, the possible conditions, and resulting destination of the microprogram branch. Microprogram conditional branches:
NON	No effect

Table 2-1 (Cont)
KD11-B Control Store Fields

Field		Description
BUT (Cont)	JSRMP	Microprogram branch on JMP or JSR instruction
	IRD	Microprogram branch on results of Instruction Register Decode
	BYT	Microprogram branch to distinguish: a) byte and non-byte instructions, and b) odd/even byte references
	DST	Microprogram branches on destination mode IR (5:3)
	MOV	Microprogram branch to distinguish both MOV and MOVB from other instructions
	INT	Microprogram branch on interrupt to be processed
	UNY	Microprogram branch to distinguish unary instructions
	SW	Microprogram branch dependent on console switch action
	NMD	Microprogram branches to distinguish non-modifying instructions (e.g., CMP, TST, etc.)
	SRV	Microprogram branch at end of instruction sequence to determine if any condition requires service before going off to fetch next instruction
		Miscellaneous encoded field:
	CON	Enable the constants ROM on the A-leg
	INI	Trigger BUS INIT L during the RESET instruction
	SVS	Set SSYN on Unibus during the interrupt sequence
	ENO	Enable the stack overflow detection logic
	IRC	Clock data into the instruction register
BRG		Control the B register
	H	Hold, do not modify.
	L	Load.
	SR	Shift right once.
	SL	Shift left once.
ALG		A-leg control; determines what is enabled onto the A-inputs of the ALU

Table 2-1 (Cont)
KD11-B Control Store Fields

Field		Description
ALG (Cont)	SP	Scratch pad
	NUL	Nothing
	SPR	Low orders eight bits (right half) of the scratch pad
	PSW	Program Status Word
TNS		Initiation of Unibus transfer
	NON	No effect
	I	Initiate DATI
	O	Initiate DATO
	IP	Initiate DATIP
ABT		Allow byte reference on current Unibus transfer.
	NO	
	YES	
CKO		Inhibit the processor clock until pending Unibus transfer is complete.
	OFF	No effect
	ON	
SPA		Scratch pad address. This field is physically split in the control store word. It is made up of: SPA = SP0 = CS (18) SP1 = CS (22) SP2 = CS (12) SP3 = CS (21) Scratch pad address (R0 through R17)
SPF		Scratch pad control function
	REA	Scratch pad contents not modified
	WRI	Write into scratch pad
BLG		B-leg control. Determines what is enabled onto the B-input of the ALU. This field is physically split in control store word.

Table 2-1 (Cont)
KD11-B Control Store Fields

Field		Description
BLG (Cont)		BLG = BTP (B Top - Upper Byte) = CS (14) BBT (B Bottom - Lower Byte) = CS (16)
	BRG	B register
	SEX	B register sign extended. Bit 7 of the B register is propagated from bit 7 to bit 15.
	+1	The constant +1
BAR		Bus Address Register Control
	H	Hold, do not modify.
	L	Load.
SAM		Scratch pad address multiplexer control. This field is physically split in the control store word. SAM = SM0 (19) SM1 (17)
	ROM	Scratch pad address taken from control store word (see SPA field)
	IRS	Scratch pad address taken from source register bits of Instruction Register, IR (8:6).
	IRD	Scratch pad address taken from destination register bits of Instruction Register, IR (2:0).
	BAR	Scratch pad address taken from Bus Address Register low order three bits, BA (2:0).
PSW		Program Status Word control
	H	Hold
	L	Load
AUX		Auxiliary ALU control enabled
	OFF	
	ON	
CRI		Enable carry in to ALU
	OFF	
	ON	

Table 2-1 (Cont)
KD11-B Control Store Fields

Field		Description	
ALU		ALU function	
	AL	A logical	
	AA	A arithmetic	
	AB	A and B	
	ABBAR	A and ones complement of B	
	ZERO	Output zero	
	A OR B	A or B	
	BL	B logical	
	A + B	A plus B	
	AXORB	A exclusive or B	
	A-B-1	A minus B minus 1	
	BBAR	1's complement of B	
	-1	Output the constant minus one	
	A-1	A minus one	
	ABAR	1's complement of A	
	ASL	Arithmetic shift B left	These are used during shift and rotate instructions to control the serial shift inputs to the B register.
	ROL	Rotate B left	
	ASR	Arithmetic shift B right	
	ROR	Rotate B right	

A field may contain any one of a number of different alternative bit patterns. To facilitate microprogramming, these alternatives have been given symbolic names, making it possible to work with the microprogram at a symbolic level rather than in binary. For example (Table 2-1), one of the alternative values that can be assigned to the ALU field is OR (A or B). This value corresponds to a bit pattern of 01001 [CS (37:33) = 01001].

The data word output from the CS is determined by the contents of the MPC registers (E111 and E101 shown on print CONF).

2.4 BRANCHING WITHIN MICROROUTINES

A microroutine is composed of a sequence of microsteps. Every microstep specifies the location of the next microstep in a sequence, namely, the NXT field. During the execution of a microstep, the signals resulting from the NXT field are loaded into the MPC (microprogram counter). The MPC specifies the location from which the next microstep will be executed (print CONF). Conditional branching within a microroutine is accomplished by wire-ORing signals into those signals coming from the NXT field, while they are being loaded into the MPC. Each branch condition controls the minimum number of bits required. This restricts the range of branching, but it minimizes the logic (print CONE). This provides control for all the bits in the MPC.Table 2-2 shows the location of each microcode branch, the destination, and associated conditions.

In general, microsteps are not executed from numerically sequential locations. This extra degree of complexity (and an extra eight bits in each CS word to specify the NXT location) enables the minimization of logic.

Table 2-2
Microprogram Branches (BUT)

BUT	Source	Destination	Comment
IRD (IR decode)	F-5	S0-1 through S7-1	All double operand instructions
		D0-1 through D7-1	Single operand instructions
		B-1	Branch, change PC
		B2-2D	Branch, PC unchanged
		MCC-1	Set or 0 clear condition codes
		R1-1	RTS
		R2-1	RTI
		W-1	WAIT
		H-1	HALT
		ET-1	EMT
		BT-1	Break Point Trap
		IT-1	IOT
		T-1	Trap
		RT-1	Reserved instruction
		RST-1	RESET
DST (destination)*	S0-2, SBE-2	D0-1 through D7-1	
	CCM-2	CC-1	Clear condition codes
		SC-1	Set condition codes

*Always has a branching destination (i.e., NXT field always modified).

Table 2-2 (Cont)
Microprogram Branches (BUT)

BUT	Source	Destination	Comment
BYT (byte)	S0-1	SBE-1	Byte source data (Mode 0)
	S1-2	SBE-1	Even byte source data
		SBO-1	Odd byte source data
MOVE	D0-1	DBO-1	Byte instruction other than MOVE
		MB-0	MOVB instruction (BYTE)
		D0-3A	MOV instruction (NOT BYTE)
NMD (non-modifying)	D0-3, D0-3A	B2-2A	Non-modifying instruction TST, CMP, Bit
	D1-4	B2-2B	
	DBO-2	B2-2	
	DO-10	B2-2C	
SRV (service)	B-3, B2-2 B2-2A, B2-2B B2-2C, B2-2D, CC-1, CS-3, D0-4, DB0-3, J1-2, J2-8, MB-2, SC-1		In order of priority highest to lowest
		BT-1	T-bit trap
		ERT-IA	Stack overflow trap
		PF-1	Power fail
		BG-1	BR 7 (bus request level)
		BG-1	BR 6
		LC-1	Internal line clock
		BG-1	BR 5
		BG-1	BR 4
		URTR	UART Receive
		URTX	UART Transmit

Table 2-2 (Cont)
Microprogram Branches (BUT)

BUT	Source	Destination	Comment
SRV (Cont)		H-1	Console STOP
		F-1	None of the above
		W-1	When executing WAIT instruction, LOOP ON W-1 instead of going to F-1.
SW (switch)	H-2	CS-1	Start
		CCS-1	Continue
		CE1-1	Examine 1st.
		CE2-1	Examine
		CD1-1	Deposit 1st.
		CD2-1	Deposit
		CL-1	Load
		H-2	None
		CE1-1	Loop until examine is released.
INT (interrupt)	BG-1	INT-1	Interrupt service
JSRMP	D1-1, D2-3, D3-5, D6-5	J1-1	JMP instruction mode of operation to change PC.
	D6-5	J2-1	JSR instruction
INITIALIZE	RST-1		Initialize computer RESET instruction.
UNY (unary)	D0-2	ERT-1	JMP or JSR Mode 0 - illegal instruction
		SB1-1	SWAB
		U1-1	Other unary
	D1-3	SB2-1	SWAB
		U2-1	Other unary
	DB0-1	U3-1	Unary other than JMP, JSR, or SWAB

Table 2-2 (Cont)
Microprogram Branches (BUT)

BUT	Source	Destination	Comment
UNY (Cont)	DE-1	U5-1	Unary other than JMP, JSR, or SWAB
	DO-9	U4-1	Unary other than JMP, JSR, or SWAB
NON (none)			No branch test

2.5 MICROPROGRAM FLOW

The microprogram flow chart is shown in full detail in engineering drawing K-MP-KD11-B-1. Figure 2-2 is a simplified flow that provides an overview and aids in using the detailed flow. No attempt is made in this manual to trace through each path of the microcode. An explanation of the detailed flow notation is provided along with examples to illustrate instruction interpretation, interrupts and traps, and console functions.

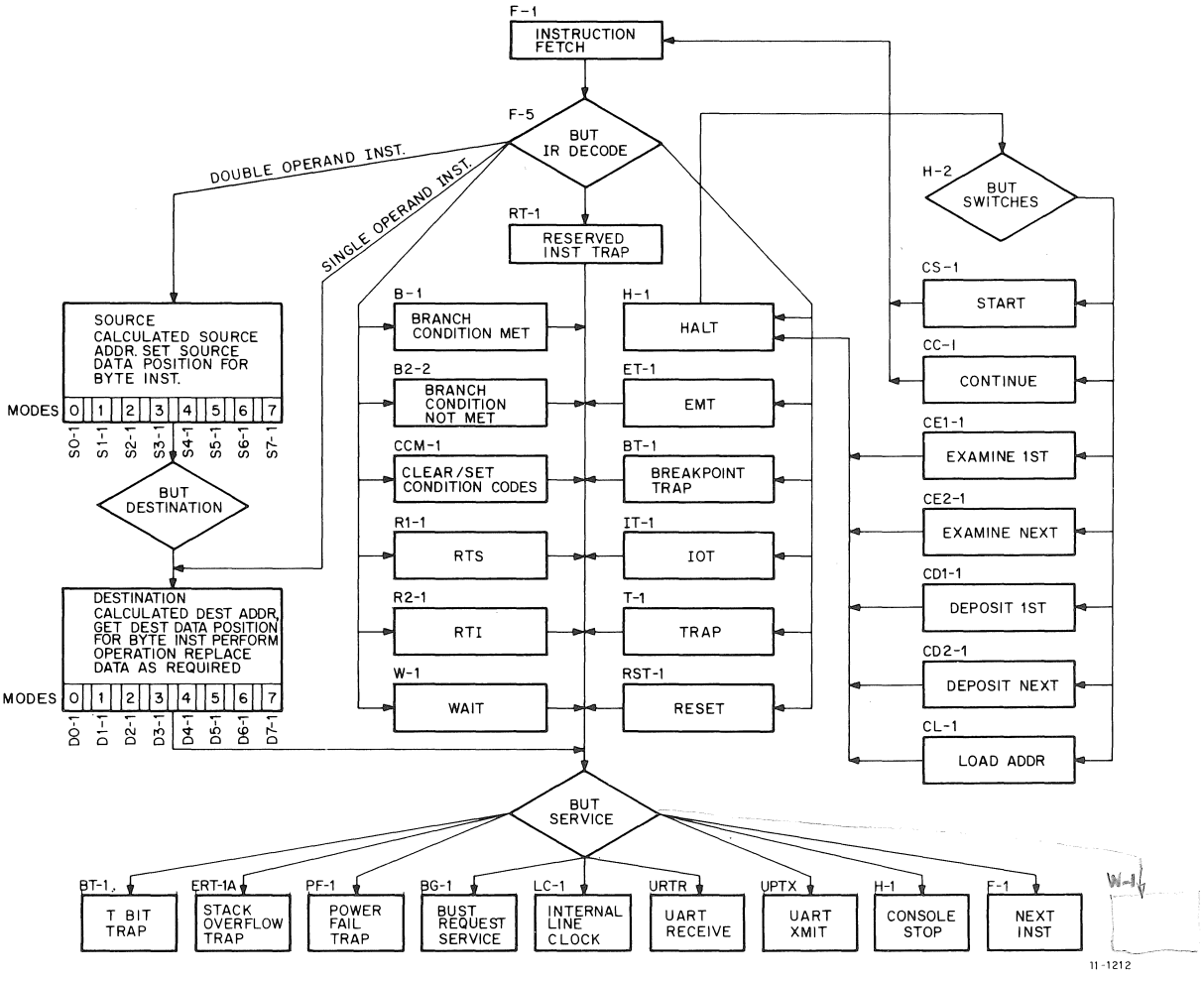


Figure 2-2 KD11-B Simplified Flow Diagram

2.5.1 Flow Chart Notation

Figure 2-3 illustrates an excerpt from the microprogram flow section of the prints. Notice that the listing is grouped into microroutines (source mode 0 through mode 3); these microroutines start with an identifying comment, the first character of which (disregarding the LOC and NXT columns) is an asterisk. Other comment lines begin with a slash.

```
LOC     NXT     * SOURCE MODE 0 (REGISTER), GET SOURCE DATA
                / GET TO S0-1 FROM F-5 VIA BUT IR DECODE IR 11:9 = 0
201     007     S0-1 B  R[S] ; BUT BYTE
                / IF BYTE INST GOTO SBE-1 (MUST BE EVEN BYTE)
007     001     S0-2  R[10]  B; BUT DESTINATION
                / IF IR 5:3   = 0 GOTO   D0-1
                /             = 1        D1-1
                /             = 2        D2-1
                /             = 3        D3-1
                /             = 4        D4-1
                /             = 5        D5-1
                /             = 6        D6-1
                /             = 7        D7-1

LOC     NXT     * SOURCE MODE 1 (REG. DEFERRED) GET SOURCE DATA
                / GET TO S1-1 FROM F-5 VIA BUT IR DECODE IR 11:9
203     244     S1-1  BA  R[S] ; DATI; CKOFF; ALBYT
                / GET TO S1-2 FROM S2-3 VIA GOTO
                /          "          S3-5    "
                /          "          S6-5    "
244     007     S1-2  B  UNIBUS DATA; BUT BYTE; GOTO S0-2
                / IF ODD BYTE GOTO SBO-1
                / IF EVEN BYTE GOTO SBE-1
                / IF NOT BYTE FALL THROUGH TO S0-2

LOC     NXT     * SOURCE MODE 2 (AUTO-INC.) GET SOURCE DATA
                / GET TO S2-1 FROM F-5 VIA BUT IR DECODE IR  11:9 = 2
205     301     S2-1  BA  R[S] ; DATI; ALBYT
301     014     S2-2  B  R[S]+1+BYTE. BAR
                / GET TO S2-3 FROM S4-1 VIA GOTO
214     244     S2-3  R[S]  8; CKOFF; GOTO S1-2

LOC     NXT     * SOURCE MODE 3 (AUTO-INC DEFERRED) GET SOURCE DATA
                / GET TO S3-1 FROM F-5 VIA BUT IR DECODE IR  11:9 = 3
207     016     S3-1  BA  R[S] ; DATI (MUST BE AN EVEN ADDRESS HERE)
216     017     S3-2  B  R[S]+2
```

Figure 2-3 Excerpt from Microprogram Flow (KMP-KDL-B-1)

All microsteps have mnemonic names such as S0-1 (source mode 0, step 1), S2-2 (source mode 2, step 2), etc. A microroutine will often weave back and reuse part of another. For example, the source mode 1 routine weaves back into the source mode 0 routine by the GOTO S0-2 in S1-2 (Figure 2-3).

To the left of every microstep is the location of that step in the CS (in octal) and the contents of the NXT field. Observe the microprogram counter (MPC) while single stepping through the microprogram. The LOC and NXT columns provide useful information relating to the path taken by the microprogram.

The flow is well commented and should be self-explanatory. Table 2-3 is a useful glossary of flow notation.

Table 2-3
Flow Notation Glossary

Designation	Definition
BA	Bus Address Register
←	Assignment operator
;	Separator
DATI	Initiate DATI operation on Unibus.
+	Plus, the arithmetic operator
PC	Program Counter = R 7
CKOFF	Set the Clock Off bit of the control store.
B	B-leg register
IR	Instruction Register
B Sex	B-leg register sign extended (bit 7 repeated in bits 8 through 15)
R [S]	Scratch Pad Register specified by the source portion of the current instruction [IR (8:6)].
R [D]	Scratch Pad Register specified by the destination portion of the current instruction [IR (2:0)].
R [n]	Scratch Pad Register n specified by the control ROM
BUT	Branch on microtest
ALBYT	Allow byte Unibus reference
BYTE.BAR	A signal indicating the absence of a byte in instruction
ENABOVER	Enable the stack overflow detection logic (working BUT)
DATO	Initiate DATO operation on Unibus.
DATIP	Initiate DATIP operation on Unibus.
INIT	Initialize the logic (working BUT).

Table 2-3 (Cont)
Flow Notation Glossary

Designation	Definition
SVS	Set slave sync (working BUT).
IRC	Clock the Instruction Register (working BUT).
K [n]	That location of the constants chip (on the data path A-leg) containing the value n.
R [10] OP B	ALU function determined by the auxiliary ALU control logic as a function of the instruction currently in the Instruction Register.
GOTO X	NXT field is to contain the address of X. Unconditional GOTO.

To illustrate the interpretation of PDP-11 instructions, the execution of a CMP instruction is traced through the microcode. The machine is in the RUN state (i.e., the machine is executing instructions) and the instruction is located in memory location 1000.

Location	Assembler Symbolic	Octal
1000	CMP #15, CHAR	022767
1002		000015
1004		000100
.		
.		
.		
1106	CHAR: WORD 0	

This instruction compares the literal 15 to the contents of CHAR and sets the condition code accordingly. Source mode is immediate (mode 2, register 7 = PC) and destination mode is relative (mode 6, register 7 = PC). Figure 2-4 shows the simplified flow for the CMP example.

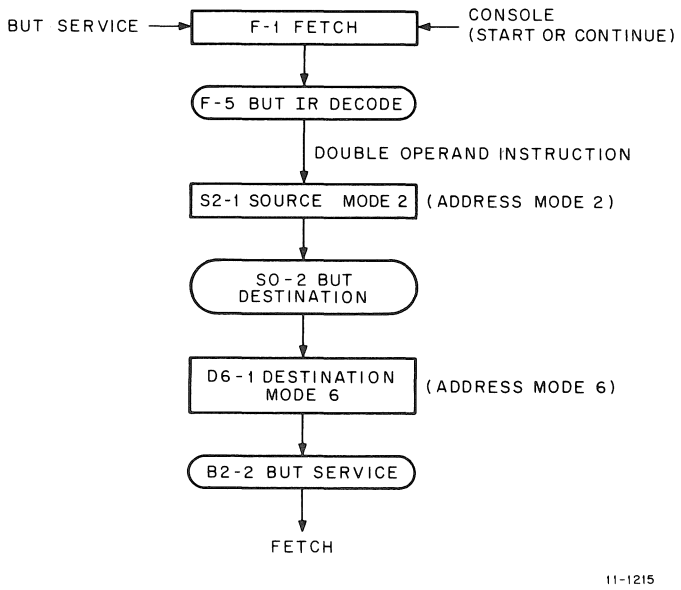


Figure 2-4 CMP # 15, CHAR (022767), Simplified Flow Diagram

First the instruction is fetched from memory (microsteps F-1 through F-5). This is the same fetch microroutine used to get each instruction from memory and update the PC.

Location	NXT	Microstep Name	Action	Comment
062	053	F-1	BA ← PC: DATI	/Load the Bus Address Register (BA) with the contents of the PC (R7) and initiate a DATI by the Unibus control (BC).
053	365	F-2	B ← PC+2	/Load the B register with the contents of the PC+2.
365	364	F-3	PC ← B; CKOFF	/Update the PC. CKOFF inhibits execution of the next microstep until the pending Unibus transfer (DATI, initiated in F-1) is complete.
364	061	F-4	B, IR ← UNIBUS DATA	/Load the data from the Unibus (instruction fetched from memory) into the B register and Instruction Register (IR).
061	001	F-5	B ← B SEX; BUT IR DECODE	/Sign extend the low order eight bits of the copy of the instruction in the B register (used in branch instruction interpretation) and branch on microtest (BUT) determined by the IR decode logic. Note that NXT (F-5) = 1 which is the CS location of the RESERVED instruction microroutine. If the IR decode logic does not recognize the instruction, no signals are wire-ORed into the MPC and the RESERVED instruction microroutine (RT-1) is executed by the microprogram. In this example, CMP is recognized (by the IR decode logic) and 204 is wire-ORed with NXT (F-5 = 1) to cause the MPC to be loaded with 205, the location of the microroutine which operates on source mode 2 (S2-1).

Since the instruction is of the double operand group, the next step is to get the source data. Source mode 2 is autoincrement. (Autoincrement implies one level of deferred addressing.) When used with R7 (the PC), it becomes an immediate mode.

Location	NXT	Microstep Name	Action	Comment
205	301	S2-1	BA ← R [S]; DATI; ALBYT	/Load the BA with the contents of the register specified by IR (08:06). The register will contain the location of the source data (1002) in this example. Initiate a Unibus DATI to actually get the data. ALBYT will allow an odd Unibus transfer, if the IR

Location	NXT	Microstep Name	Action	Comment
				contains a byte instruction and the BA contains an odd address. Without the ALBYT, a Unibus transfer that addresses an odd BA results in a bus error (Paragraph 2.3).
301	014	S2-2	B ← R [S] + 1 + BYTE . BAR	/For byte instructions, the autoincrement is by one, for non-byte instructions, auto-increment is by two. BYTE BAR indicates that BLG (S2-2) = +1, and this signal is conditioned by the logic, such that it is true (+1) only when the IR does not contain a byte instruction. So actually, R [S] is on the A-leg of the ALU, CARRY IN is enabled, and the +1 constant (enabled only if the IR does not contain a byte instruction) is on the B-leg. The ALU function is A + B.
014	244	S2-3	R [S] ← B;	/Update the register which is to be auto-incremented. Inhibit the processor clock until the DATI initiated in S2-1 is complete. From here, the microroutine is woven back into S1-2 [i.e., NXT (S2-3) = S1-2].
244	007	S1-2	B ← UNIBUS DATA;	/Load the source data which has come in from memory into the B register. The microcode at this point joins the micro-routine associated with source mode 0 (S0-2). Not a byte instruction, so go to S0-2.
007	001	S0-2	R [10] ← B; BUT DESTINATION	/Source data is stored in the scratch pad register, R [10], while the destination data is retrieved. BUT DESTINATION will cause a microcode branch dependent on IR (3:5). In this case, the destination mode of 6 will cause 114 to be wire-ORed into the NXT (S0-2) = 1, such that the MPC will be loaded with 115 = LOC (D6-1).

The microroutine starting in D6-1 will get the destination data and perform the operation indicated by the OP code of the instruction. Mode 6, when used with the PC, requires that the index contained in the word currently pointed to by the PC be added to the updated PC (address of the index word plus two) to get the location of the source data.

Location	NXT	Microstep Name	Action	Comment
115	075	D6-1	BA ← PC; DATI	/Initiate the Unibus transfer to get the index word from memory.
075	077	D6-2	B ← PC + 2	/Prepare to update PC to next word.

Location	NXT	Microstep Name	Action	Comment
077	057	D6-3	PC ← B; CKOFF	/Update the PC and inhibit the processor clock until the Unibus DATI initiated in D6-1 is complete.
057	300	D6-4	B ← UNIBUS DATA	/Receive the index word into the B register.
300	200	D6-5	B, BA ← B+R (D); DATI; BUT JSRMP; ALBYT; CKOFF; GOTO D1-2	/The actual location of the destination data is formed by adding the index (in the B register) to the destination register [IR (2:0)], which is the PC in the example. This address is loaded into the BA, and a DATI is issued to retrieve the data from memory. As in S2-1, ALBYT makes odd byte Unibus transfers legal. BUT JSRMP involves a collection of logic which examines the contents of the IR to see if the instruction is a JMP or JSR. If either of these instructions are present, the appropriate bit is wire-ORed with NXT (D6-5) = D1-2 into the MPC, such that the MPC is loaded with J1-1 or J2-1, respectively for JMP or JSR instruction. In the example, neither of these instructions are present and the MPC is loaded with NXT (D6-5) = D1-2. CKOFF inhibits the processor clock until the DATI initiated in this microstep is complete. Note that this is the first time in this example that memory reference has not been overlapped with microprograms.
200	210	D1-2	D ← UNIBUS DATA; BUT BYTE	/Receive the destination data from memory. If the instruction had been a byte instruction (e.g., CMPB), the microprogram would be diverted to D0-1 (for odd byte address) to get the byte operand into the right half of the B register. This is not the case in this example.
210	143	D1-3	R [11] ← B; BUT UNARY	/It is at this point in the microroutine that a branch occurs for unary instructions (e.g., SWAB, CLR, COM, etc.). Unary instructions would have caused the BUT IR DECODE done in F-5 to take the appropriate destination microroutine (there is no source field in a unary instruction). R [11] is used in unary instruction interpretation.
163	334	D1-4	B ← R [10] OP B; BUT NON MOD	/This microstep allows the AUX ALU control ROM (print DPF) to: 1) cause the ALU to perform the appropriate function, and 2) set or clear condition codes in

Location	NXT	Microstep Name	Action	Comment
				accordance with the instruction in the IR and the results of the ALU operation. In the example, it is the setting of condition codes which count. Since CMP is an instruction that does not modify memory, (NONMOD), the microprogram is ready to branch to the microstep in which a BUT SERVICE is done. If the instruction requires a memory modification (e.g., MOV, ADD, INC., etc.), D1-5 and D1-6 are executed before going to BUT SERVICE.
335	040	B2-2B	BUT SERVICE	/At the end of each instruction, various situations that attempt to intervene before the next instruction are tested. Their priorities are arbitrated in the F101 ROM shown on print CONE. These conditions and their relative priorities are as follows: High priority 1. T-bit trap 2. Stack overflow 3. Power fail 4. Bus request level 7 5. Bus request level 6 6. Internal line clock 7. Bus request level 5 8. Bus request level 4 9. UART receive 10. UART transmit 11. Console stop Low priority 12. Next instruction If no condition with a higher priority exists, the microprogram proceeds to F-1 and commences with the fetch of the next instruction.

This completes the example of the microprogram interpretation of CMP #5, CHAR. It may be useful to trace this or some other instruction through the detailed flow (K-WL-KD11-B-1).

2.5.2 Interrupts and Traps

Interrupts and traps are also accomplished by the microprogram. Interrupts are sent from Unibus devices; bus requests (BR) are received by the BC. At the end of each instruction (not microstep), if a BR is present, and if it has the highest priority (see microstep B2-2 in previous example), the microprogram goes to BG-1. In BG-1, a BUT INTERRUPT is done to distinguish BRs that are associated with interrupts from those that are not. If an interrupt is required, the microcode is diverted to INT-1 where the interrupt vector location is loaded into R (12) from the Unibus data lines. At this point, the microprogram joins the ET-2 microroutine, which stacks the PSW and PC and retrieves a new PSW and PC from the interrupt vector words. At the end of microroutine ET-13, another BUT SERVICE is done to determine if anything (e.g., another higher priority interrupt or the occurrence of stack overflow) is asserted. If none are, the microprogram proceeds to F-1 where it commences to fetch the next instruction.

Power fail trap, stack overflow trap, and T-bit traps are also recognized during BUT SERVICE. Each of these routines has a microroutine associated with it that loads the B register with the appropriate trap vector location (from the constants ROM, E44 on print DPB). In each case, the microprogram joins the ET-2 microroutine which stacks the PSW and PC and loads the new PSW and PC, just as with external interrupts. The main difference is that the vector location comes from the constants ROM rather than from the UNIBUS DATA.

Bus error traps are treated differently since they may prevent an instruction from being completed. When a bus error is detected, the NXT field of the CS (E102 and E112 on print CONH) is disabled, and the microprogram is forced to ERT-1. This microroutine picks up the respective trap vector location from the constants ROM, and from that point on, operates like all other traps. The difference is the method in which the microprogram gets to ERT-1.

2.5.3 Console Functions

When the processor is in the HALT state, the microprogram is looping on microstep H-2 doing BUT SWITCH. As a console switch is activated, the microprogram branches to an associated microroutine. Additional logic intervenes to distinguish the first of a sequence of examines or deposits. This is illustrated in the following examples.

Assume that the console operator wants to examine locations 1000 and 1002. The processor is in the HALT state, with the microprogram looping on microstep H-2. First the operator must set the switches to 1000 and depress LOAD ADRS. The BUT SWITCH then causes the microprogram to branch to CL-1.

Location	NXT	Microstep Name	Action	Comment
302	300	H-2	BUT SWITCH	/Loop on H-2 waiting for switch action. When LOAD ADRS is depressed, branch to CL-1.
311	375	CL-1	BA ← K [207]. BAR *; DATI; CKOFF	/The SR is logically on the Unibus at location 177570. This constant is obtained from the 8-bit wide constants ROM (F25 on DPB print) by taking 207 and forming the complement through the ALU on the way to the BA. A request for the contents of the SR is initiated (DATI) and the processor clock is inhibited until the data is available (CKOFF).
375	367	CL-2	B ← UNIBUS DATA	/Since the SR is physically on the A-leg of the data path (DP) (prints DPA, DPB, DPC, and DPD), it cannot be written directly into register 17 of the scratch pad; instead, it is first loaded into the B register.
367	302	CL-3	R [17] ← B; GOTO H-2	/Load SR into the Load Address Register, R [17]. Microprogram goes to H-2.
		H-2	BUT SWITCH; GOTO, H-2	/Loop here looking for switch activity. The microprogram loops on CL-1, CL-2, CL-3, and H-2 as long as LOAD ADR is depressed.

*(207). BAR = 1's complement of 207

Now the operator has loaded 1000 from the SR into the Load Address Register R [17]. The lights are attached to the B register and will display the loaded address.

To examine location 1000, the operator depresses EXAM. As long as the EXAM switch is depressed, the location to be examined is displayed in the lights. When it is released, the contents of that location are displayed.

Location	NXT	Microstep Name	Action	Comment
317	307	CE1-1	BA, B ← R [17]; BUT SWITCH	/The lights are connected to the B-leg. By loading the B register with the contents of the Load Address Register, R [17], the address of the location is displayed. The BA is also loaded for subsequent retrieving of the data. BUT SWITCH causes the microprogram to loop on CE1-1 until EXAM is released.
307	326	CE1-2	DATI; CKOFF	/When the switch is released, the data is requested from the Unibus, and the processor clock is inhibited until it is available.
326	302	CE1-3	B ← UNIBUS DATA; GOTO H-2	/Display the data by loading it into the B register and return to the H-2 microprogram loop to await the next switch action.

While the microprogram loops in H-2, the B register remains unchanged and the contents of location 1000 are displayed. When EXAM is depressed a second time, the logic associated with F100 (print CONE) causes BUT SWITCH in H-2 to branch the microcode to CE2-1. In this case, the Load Address Register must be incremented before using its contents.

Location	NXT	Microstep Name	Action	Comment
302	300	H-2	BUT SWITCH	/Loop waiting for switch action.
315	371	CE2-1	B ← R [17] + 2	/Increment the Load Address Register so that sequential words can be examined.
371	317	CE2-2	R [17] ← B; GOTO CE1-1	/Update R [17]. The rest of this microroutine merges with CE1-1.
317	307	CE1-1	BA, B ← R [17]; BUT SWITCH	
307	326	CE1-2	DATI; CKOFF	
326	302	CE1-3	B ← UNIBUS DATA; GOTO H-2	

This completes the example of console function microroutines. The remaining console functions are quite similar.

2.6 MICROPROGRAM SYMBOLIC LISTING

The microprogram section of the prints (K-MP-KD11-B-1 through 4) contains four useful tools. Paragraph 2.5 describes the microprogram flow. Flow is probably the most useful level to work with the microprogram when tracing through processor action on any specific operation. Flow tells what happens in each microstep and why. To determine how a microstep accomplishes its task, refer to the Microprogram Symbolic Listing (K-MP-KD11-B-2), an excerpt of which is shown in Figure 2-5. In this listing, microsteps are listed alphabetically (e.g., F-1, F-2 . . .). Each of the CS fields described in Table 2-1 is listed along with its symbolic values. For example, in F-2 of the example in Paragraph 2.5.1, flow indicates:

F-2 B ← PC + 2

The symbolic listing is useful for determining how this is to be accomplished in terms of CS fields (e.g., ALU function). Refer to the excerpt in Figure 2-5 and scan the alphabetically-ordered list of names for F-2.

A-leg	(ALG)	= SP (scratch pad)
ALU function	(ALU)	= A + B
B-leg	(BLG)	= +1 (the constant)
B Register	(BRG)	= L (load)
Carry In	(CRI)	= ON
Scratch Pad Address	(SPA)	= R7 (the PC)
Scratch Pad Function	(SPF)	= REA (read)
Next MPC	(NXT)	= F-3 (go to F-3 next)

B ← PC + 2 is accomplished by gating register 7 (the PC) onto the A-leg of the ALU, gating + 1 onto the B-leg, and causing the ALU to perform an A + B operation (=R7 + 1) with Carry In enabled (=R7 + 1 + carry in). The B register is loaded with the results, and the MPC is loaded with the address of F-3, which is the next microstep.

Only eight of a total of eighteen fields are described in the above example. The rest of the fields have values but they are not of immediate interest.

2.7 MICROPROGRAM BINARY LISTING

In addition to the flow and symbolic listing, a binary listing of the CS is included in the microprogram section of the prints (K-MP-KD11-B-3). An excerpt is shown in Figure 2-6. As in the symbolic listing, the binary listing is alphabetically ordered by microstep name. The fields are located across the top of the listing; however, they relate closely to the actual signals (Figure 2-1). A high is represented by a 1 in this listing.

From the previous example, flow indicates F-1 B ← PC + 2. The symbolic listing shows that the ALU function to accomplish this is A + B; the binary listing shows the actual logic level value of CONF ALU S3 L, CONF ALU S2 L, CONF ALU S1 L, CONF ALU S0 L, and CONF ALU MODE H (Figure 2-1 and 2-6). Notice that these five signals are grouped together under the heading ALU. They physically come from chips E103 and E104. The binary listing is spaced to show signals grouped both by field (ALU) and chip (E103 and E104).

If the PC is not being properly incremented during program execution, the flow may be used to determine what is supposed to happen during the fetch microroutine; the symbolic listing is used to determine how it is to be accomplished. If the symbolic listing does not identify the problem, use the binary listing and an oscilloscope probe to locate the incorrect signal and/or malfunctioning chip.

KD11-B MICROPROGRAM SYMBOLIC LISTING

NAME	LOC	ABT	ALG	ALU	AUX	BAR	BLG	BRG	BUT	CON	CKO	CRI	PSW	SAM	SPA	SPF	TNS	NXT
DO-9	132	NO	SP	BL	OFF	H	SEX	L	UNY	NON	OFF	OFF	H	ROM	R11	WRI	NON	A145
ERT-1	010	NO	NUL	AL	OFF	H	BRG	L	CON	4	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
ERT1A	046	NO	NUL	AL	OFF	H	BRG	L	CON	4	OFF	OFF	H	ROM	R0	WRI	NON	ET2-2
ERT1B	153	NO	NUL	AL	OFF	H	BRG	L	CON	4	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
ET-1	011	NO	NUL	AL	OFF	H	BRG	L	CON	30	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
ET-10	254	NO	SP	AL	OFF	L	BRG	H	IRC	NON	ON	OFF	H	ROM	R12	REA	I	ET-11
ET-11	255	NO	SP	AL	OFF	L	BRG	L	NON	NON	OFF	OFF	H	ROM	R7	WRI	NON	ET-12
ET-12	256	NO	SP	A+B	OFF	L	+1	L	NON	NON	ON	ON	H	ROM	R12	REA	I	ET-13
ET-13	257	NO	SP	AL	OFF	H	BRG	L	NON	NON	OFF	OFF	L	ROM	R0	REA	NON	B2-2
ET-2	245	NO	SP	BL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R12	WRI	NON	ET-3
ET-3	246	NO	SP	A-B-1	OFF	L	+1	L	END	NON	OFF	OFF	H	ROM	R6	REA	NON	ET-5
ET-5	247	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R6	WRI	O	ET-6
ET-6	226	NO	PSW	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R0	REA	NON	ET-7
ET-7	251	NO	SP	A-B-1	OFF	L	+1	L	END	NON	OFF	OFF	H	ROM	R6	REA	NON	ET-8
ET-8	252	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R6	WRI	O	ET-9
ET-9	253	NO	SP	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R7	REA	NON	ET-10
ET2-2	003	NO	SP	BL	OFF	H	BRG	L	NON	NON	OFF	OFF	H	ROM	R12	WRI	NON	ET2-3
ET2-3	004	NO	SP	A-B-1	OFF	L	+1	L	NON	NON	OFF	OFF	H	ROM	R6	REA	NON	ET2-5
ET2-5	036	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R6	WRI	O	ET2-6
ET2-6	037	NO	PSW	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R0	REA	NON	ET2-7
ET2-7	051	NO	SP	A-B-1	OFF	L	+1	L	NON	NON	OFF	OFF	H	ROM	R6	REA	NON	ET-8
F-1	062	NO	SP	AL	OFF	L	BRG	H	NON	NON	OFF	OFF	H	ROM	R7	REA	I	F-2
F-2	053	NO	SP	A+B	OFF	H	+1	L	NON	NON	OFF	ON	H	ROM	R7	REA	NON	F-3
F-3	365	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R7	WRI	NON	F-4
F-4	364	NO	NUL	AL	OFF	H	BRG	L	IRC	NON	OFF	OFF	H	BAR	R0	REA	NON	F-5
F-5	061	NO	SP	BL	OFF	H	SEX	L	IRD	NON	OFF	OFF	H	ROM	R0	REA	NON	RT-1
H-1	041	NO	SP	AL	OFF	H	BRG	L	NON	NON	OFF	OFF	H	ROM	R7	REA	NON	H-2
H-2	302	NO	SP	AL	OFF	L	BRG	H	SW	NON	OFF	OFF	H	ROM	R17	REA	NON	D6-5
INT-1	325	NO	SP	AL	OFF	H	BRG	H	SVS	NON	OFF	OFF	H	ROM	R12	WRI	NON	ET-3
IT-1	273	NO	NUL	AL	OFF	H	BRG	L	CON	20	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
J1-1	204	NO	SP	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R0	REA	NON	J1-2
J1-2	260	NO	SP	BL	OFF	H	BRG	H	SRV	NON	OFF	OFF	H	ROM	R7	WRI	NON	BG-1
J2-1	212	NO	SP	AL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R0	REA	NON	J2-1A
J2-1A	261	NO	SP	BL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	ROM	R11	WRI	NON	J2-2
J2-2	262	NO	SP	A-B-1	OFF	L	+1	L	NON	NON	OFF	OFF	H	ROM	R6	REA	NON	J2-3
J2-3	214	NO	SP	BL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R6	WRI	O	J2-4
J2-4	206	NO	SP	AL	OFF	H	BRG	H	ENO	NON	OFF	OFF	H	IRS	R0	REA	NON	J2-5
J2-5	216	NO	SP	AL	OFF	H	BRG	L	NON	NON	OFF	OFF	H	ROM	R7	REA	HON	J2-6
J2-6	263	NO	SP	BL	OFF	H	BRG	H	NON	NON	OFF	OFF	H	IRS	R0	WRI	NON	J2-7
J2-7	264	NO	SP	AL	OFF	H	BRG	L	NON	NON	OFF	OFF	H	ROM	R11	REA	NON	J2-8
J2-8	265	NO	SP	BL	OFF	H	BRG	H	SRV	NON	OFF	OFF	H	ROM	R7	WRI	NON	BG-1
LO-1	042	NO	NUL	AL	OFF	H	BRG	L	CON	100	OFF	OFF	H	ROM	R0	WRI	NON	ET-2
MB-0	154	NO	SP	AL	OFF	H	BRG	H	NON	NON	ON	OFF	H	ROM	R0	REA	NON	MB-1
MB-1	242	NO	SP	ABAR	ON	H	BRG	L	NON	NON	OFF	ON	H	ROM	R10	REA	NON	MB-2

Figure 2-5 Excerpt of (K-WL-KD11-B-2) Microprogram Symbolic Listing

NAM	LOC	NXT	ALU	CFA RRU IEX	PSSD SPPI W13P	SSSB MPMB 001T	BBSS ATPP RPF2	CAT KBN OTS	AB LR GG	BUT
DO-1B	143	1100 1010	0000	1001	1001	1011	1110	0001	11 01	1111
DO-2	123	1010 1011	0000	1001	1001	1011	1110	1111	11 10	1111
DO-3	124	1010 1010	0000	1001	1001	1011	1110	11 11	11 00	1111
DO-4	125	1010 1001	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-5	126	1010 1000	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-6	127	1010 0111	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-7	132	1010 0110	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-8	131	1010 0101	0000	1001	1001	1011	1110	11 11	11 10	1111
DO-9	132	1001 1010	0101	1001	1011	1111	1000	11 11	11 11	1010
ERT-1	012	0101 1010	0000	1001	1101	1011	1101	11 11	10 11	1101
ERT1A	046	1111 1100	0000	1001	1101	1011	1101	11 11	10 11	1101
ERT1B	153	0101 1010	0000	1001	1101	1011	1101	11 11	10 11	1101
ET-1	011	0101 1010	0000	1001	1101	1011	1110	11 11	10 11	1101
ET-10	254	0101 0010	0000	1001	1111	1011	0110	01 10	11 00	0000
ET-11	255	0101 0001	0000	1001	1101	1111	0101	11 11	11 11	1111
ET-12	256	0101 0000	0110	0101	1111	1010	0110	01 10	11 11	1111
ET-13	257	0011 1010	0000	1001	0001	1011	1110	11 11	11 11	1111
ET-2	245	0101 1001	0101	1001	1111	1011	1100	11 11	11 00	1111
ET-3	247	0110 1001	0101	1001	1101	1011	1101	01 01	11 00	1111
ET-5	247	0110 1001	0101	1001	1101	1011	1101	01 01	11 00	1111
ET-6	226	0101 0110	0000	1001	1001	1011	1110	11 11	00 00	1111
ET-7	251	0101 0101	1001	0001	1101	1010	0111	11 11	11 11	0100
ET-8	252	0101 0100	0101	1001	1101	1011	1101	01 01	11 00	1111
ET-9	253	0101 0011	0000	1001	1101	1111	1111	11 11	11 00	1111
ET2-2	003	1111 1011	0101	1001	1111	1011	1100	11 11	11 11	1111
ET2-3	004	1110 0001	1001	0001	1101	1010	0111	11 11	11 11	1111
ET2-5	036	1110 0000	0101	1001	1101	1011	1101	01 01	11 00	1111
ET2-6	037	1101 0110	0000	1001	1001	1011	1110	11 11	00 00	1111
ET2-7	051	0101 0131	1001	0001	1101	1010	0111	11 11	11 11	1111
F-1	062	1101 0100	0000	1001	1101	1111	0111	11 10	11 00	1111
F-2	053	0000 1010	0110	0101	1101	1110	1111	11 11	11 11	1111
F-3	365	0000 1011	0101	1001	1101	1111	1101	01 11	11 001	1111
F-4	364	1100 1110	0000	1001	1001	0001	1110	11 11	10 11	0000
F-5	061	1111 1110	0101	1001	1001	1011	1010	11 11	11 11	0111
H-1	041	0011 1101	0000	1001	1101	1111	0111	11 11	11 11	1111
H-2	302	0011 1111	0000	1001	1111	1111	0111	11 11	11 00	0110
INT-1	325	0101 1001	0000	1001	1111	1011	1100	11 11	11 00	1000
IT-1	273	0101 1010	0000	1001	1001	1011	1111	11 11	10 11	1101
J1-1	204	0100 1111	0000	1001	1001	1011	1110	11 11	11 00	1111
J1-2	260	1101 1111	0101	1001	1101	1111	1101	11 11	11 00	1100

Figure 2-6 Excerpt of Microprogram Binary Listing (K-W-KD11-B-3)

The following example is used to show the interrelation of the microprogram listings and the logic prints in checking the generation of control signals from the CS ROMs. Specifically, the example deals with the generation of the SPM enabling signals during a microprogram step that requires an SPM read operation. It is a simplified example because only the SPM enabling signals are examined in detail. The address lines, input data, and output data are not examined.

The example chosen is step CCM-2 of the microprogram symbolic listing (print KD11-B-2, sheet 2). The items of interest in this example are as follows.

Name	Location	ALG	CKO	SPF
CCM-2	350	SP	OFF	REA

The step's name is CCM-2 and its location in the CS ROMs is 350_8. Table 2-1 describes the CS fields.

The ALG field is the A-leg control and determines what is enabled onto the A-inputs of the ALU. In this case, SP indicates the contents of the scratch pad memory.

The CKO field determines whether the processor clock is running or is inhibited until the pending Unibus transfer is complete. The notation OFF means that the CKO field has no effect, so the processor clock is running.

The SPF field determines the scratch pad memory control function. The notation REA means that a read operation is selected.

The binary equivalents of the CS field bits are obtained from the microprogramming binary listing (print KD11-B-3, sheet 2).

Name	Location	SPF	CKO	ALG
CCM-2	350	1 (13)	1 (11)	1 1 (07,06)

The CS field bit is shown below the binary representation. The CS field bits run from 00-39, starting with 00 at the right.

In this example, the four signals generated from CS fields ALG, CKO, and SPF are all logical 1.

Refer to Figure 2-1 to determine the signal associated with these fields and the designation of the CS ROM that generates the signal.

The following information is obtained from Figure 2-1.

Field	Bit	Signal	ROM
ALG	06	CONG ROM ALEG 0 L	E104
ALG	07	CONG ROM ALEG 1 L	E104
CKO	11	CONG CKOFF L	E116
SPF	13	CONG SP WRITE L	E106

As previously stated, the purpose of this example is to verify the generation of the SPM enabling signals during a microprogram step that requires an SPM read operation. The last step in the procedure is to trace the signal flow through the logic prints. As a visual aid, all the logic involved is shown in a simplified logic diagram (Figure 2-7). The one exception is the processor clock logic which is not discussed in detail.

In accordance with the function table for the 7489 SPM, during a read operation the E-input is low and the W-input is high (Figure 4-11). This means that signal DPA SP WRITE L must be high and CONG ENAB SPL L and CONG ENAB SPR L must both be low because a word is being read. Signal DPA SP WRITE L is generated by NAND gate E029 (print DPA) which has two inputs. One input is CONJ PROC CLOCK H which is active because the clock is

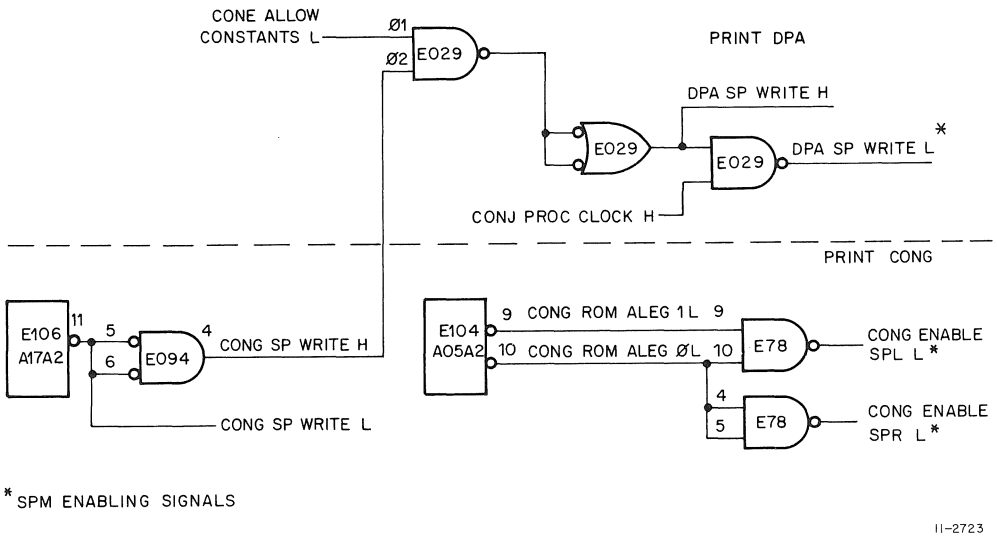


Figure 2-7 Generation of SPM Enabling Signals

running; therefore, it is high when a clock pulse is generated. The other input is CONG SP WRITE H and must be low to generate DPA SP WRITE L = 1. Signal CONG SP WRITE H is the inversion of CONG SP WRITE L which is generated by CS ROM E106. In this example, the microprogramming binary listing states that CONG SP WRITE L = 1. This signal is inverted by E094 to give CONG SP WRITE H = 0 which is the desired input to E029 (print DPA).

Signal CONG ENAB SPR L is the inversion of CONG ROM ALEG 0 L, which is generated by CS ROM E104. In this example, the microprogramming binary listing states that CONG ROM ALEG 0 L = 1. This signal is inverted by E78 to give CONG ENAB SPR L = 0, which is the desired signal.

Signal CONG ROM ALEG 0 L = 1 is also an input to NAND gate E78. The other input is CONG ROM ALEG 1 L = 1 as stated in the microprogramming binary listing. These signals produce CONG ENAB SPL L = 0 at the output of E78, which is the desired signal.

The signals generated by the CS ROMs can be checked by examining the ROM listing (print M7261-0-8). The listing is by ROM part number.

In this example, examine octal address 350 for each ROM referenced to determine the state of the desired signal. The ROM part numbers are listed below.

ROM Part No.	ROM Designation	Output Signal
23-A05A2	E04	CONG ROM ALEG 0 L CONG ROM ALEG 1 L
23-A07A2	E106	CONG SP WRITE L
23-A14A2	E116	CONG CKOFF L

2.8 MICROPROGRAM CROSS REFERENCE LISTING

The microstep name (e.g., F-2) is the key that ties the flow, symbolic, and binary listings together. When working with the processor, it is often useful to determine the name of a microstep from a location or vice versa. This information is provided in the cross reference listings (K-MP-KD11-B-4 in the microprogram section of the prints).

CHAPTER 3
CONSOLE DESCRIPTION

3.1 INTRODUCTION

This chapter provides a general and a detailed description of the console logic. The general description is keyed to the block diagram level, the detailed description covers the theory of operation of the console logic. The function and use of the console controls are discussed in the PDP-11/05S System Manual.

3.2 GENERAL DESCRIPTION

The console logic is divided into two sections: address/data register logic, and control switch logic. All the console logic is contained on one printed circuit board, which also contains the switches and indicators.

3.2.1 ADDRESS/DATA Register Logic

During manual console operation, data and addresses are generated by positioning the 16 ADDRESS/DATA Register switches. The switches are 2-position toggle type: the down position grounds the switch and provides a low signal to the processor logic; the up position provides a high signal to the processor logic by connecting the switch to +5 V.

The ADDRESS/DATA Register logic samples the 16 bits (address or data) from the B-leg of the processor data section and displays them via the ADDRESS/DATA Register indicators (Figure 3-1). The address/data multiplexer scans the processor 16-bit B-leg signals and provides a serialized output to the Buffer Register. The output of the register consists of 16 signals that are buffered and sent to the 16 ADDRESS/DATA indicators. The buffer has two modes of operation that are controlled by the SHIFT/HOLD signal from the 16-bit synchronous counter. In the shift (scan) mode, serialized data from a scan operation is shifted into the register; this operation takes 16 μs. At the end of this time, the register enters the hold (display) mode for 240 μs, during which time the register contents are displayed. This process is continuous and a scan pulse display sequence takes 256 μs. The information that is scanned (multiplexer input) remains stable for a long time compared to the 256-μs cycle for the register; therefore, the multiplexer scans relatively stable information that can be displayed.

In addition to supplying the SHIFT/HOLD signal that controls the buffer register, the counter also generates the four scan address signals that select the multiplexer inputs.

The clock provides pulses to clock the counter and Buffer Register. It starts when power is applied and is self-sustaining thereafter.

3.2.2 Control Switch Logic

The six console control switches allow programming functions to be performed manually. They are: load address (LOAD ADRS), examine (EXAM), continue (CONT), deposit (DEP), START, and HALT/ENABLE. The switches provide signals to the processor logic, which actually controls the functions.

A bounce buffer is connected across the output contacts of each switch to eliminate interruptions of the output signal due to contact bounce when the switch is activated. The bounce buffer is a latch constructed of two cross-coupled inverters.

The control switch logic senses a power-up signal (PUP) and PANEL LOCK signal to ensure control switch lockout during the panel lock mode, and to eliminate program interruption after a power interruption with the HALT/ENABLE switch left inadvertently in the HALT position during operation in the panel lock mode.

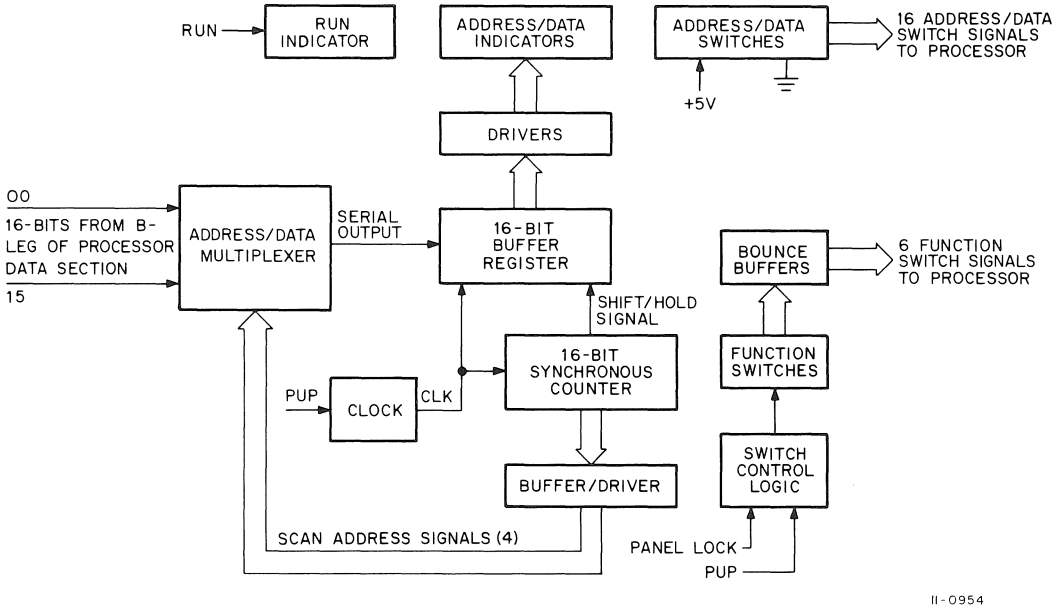


Figure 3-1 Console Functional Block Diagram

3.3 DETAILED DESCRIPTION

This paragraph provides a detailed description of the console logic. Each major functional unit is discussed separately and with regard to its interrelation with other functional units.

Both detailed and simplified logic diagrams are used to support the text. The simplified logic diagrams are included in this chapter; however, the detailed logic diagrams are part of the print set that is supplied with each computer. Three drawings are referenced, and they are identified as D-CS-5409766-0-1, sheets 1, 2, and 3. In this discussion, the drawings are referenced by the C-numbers located in the title box and shown below:

Sheet 1 – Display Buffer and Driver (C-1)
Sheet 2 – Control Keys (C-3)
Sheet 3 – Scan Control and Switch Register (C-2)

3.3.1 Multiplexer

The multiplexer, located on the processor M7260 module, scans the 16 bits in the B-leg of the processor data section. The information on these lines can be data bits or address bits. It is serialized in the multiplexer and transmitted over the console cable to the buffer.

The multiplexer is a Type 74150 Data Selector/Multiplexer (1-of-16). It has 16 inputs (D_0 through D_{15}) and a single output. Four SCAN ADRS lines from the counter are the data select lines for the multiplexer: 4 bits give 16 unique combinations. A low strobe signal enables the selected input to the output; however, the signal is inverted at the output.

The four SCAN ADRS lines select the input lines on an equivalent number basis. For example, if the SCAN ADRS lines represent decimal 5, input 5 is selected and enabled to the serial output. The SCAN ADRS lines from the counter are inverted before being sent to the multiplexer. When the counter state is zero (0000), the SCAN ADRS lines indicate 15 (1111) and multiplexer input 15 is selected. This ensures that input 15 is shifted into the proper bit position in the buffer after a scan operation is complete. Table 3-1 shows the relationship between the counter state and SCAN ADRS signals.

Table 3-1
Scan Address Signal Generation

Counter State	SCAN ADRS	MUX Line Scanned
0000 (0)	1111 (15_{10})	15
0001 (1_{10})	1110 (14_{10})	14
0010 (2_{10})	1101 (13_{10})	13
.	.	.
.	.	.
1110 (14_{10})	0001 (1_{10})	1
1111 (15_{10})	0000 (0)	0

3.3.2 Clock

The console clock provides pulses to clock the Counter and Shift Register (drawing C-2). It is a simple oscillator that generates high level clock pulses. Two retriggerable monostable multivibrators (Type 74123) are connected back-to-back to form a simple oscillator (Figure 3-2). The Q output of each is used to trigger the other. The clock starts when power is applied to the processor and is self-sustaining thereafter.

One 74123 IC package (E4) contains two separate and identical units identified as 1 and 2. Output 1Q (pin 13) is connected to input 2A (pin 9) and output 2Q (pin 5) is fed back to input 1A (pin 1). The complementary $\overline{Q}$ outputs are not used, nor are the CLEAR inputs. Input 2B is held high by application of +5V via resistor R18; therefore, unit 2 can be triggered only by a high-to-low level transition at input 2A (see truth table in Figure 3-2). Input 1B (pin 2) is connected to signal PUP from the processor. This signal is low when power is off and is high when power is on. When PUP is low, the clock output is inhibited regardless of the state of input 1A. When PUP goes high during the power-up sequence, it triggers the first high level pulse at output 1Q. The high-to-low level transition of this pulse triggers the first high level pulse at output 2Q (see timing diagram in Figure 3-2). Because both B-inputs are high, the feedback connection (2Q to 1A) allows each unit to trigger on the high-to-low transition at its A input. This produces a continuous string of positive pulses (CLK signal) at output 2Q. Pulse generation is self-sustaining as long as PUP is high.

The counter is clocked on the low-to-high clock pulse transition and the Shift Register is clocked on the high-to-low clock pulse transition. The period between clock pulses allows time for the serial data from the multiplexer to settle down. This is important because the serial data is sent to the Shift Register via a cable connection.

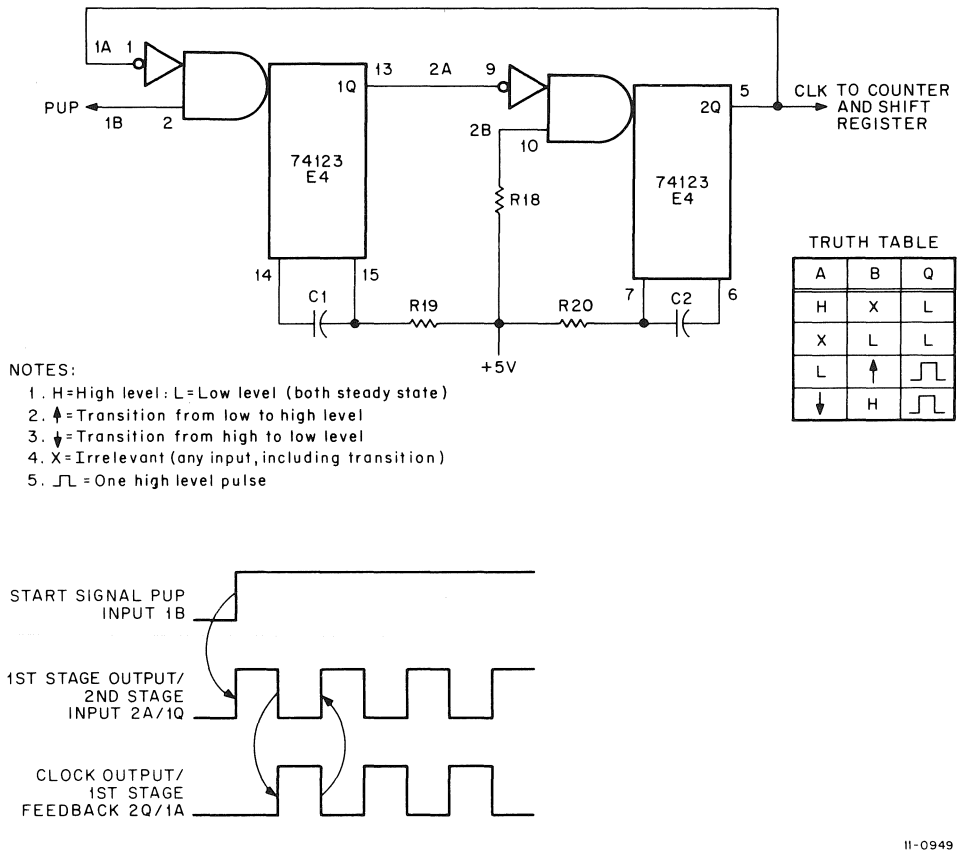


Figure 3-2 Console Clock, Schematic and Timing Diagram

3.3.3 Counter

The counter provides four scan address lines that are the data select lines for the data/address multiplexer (drawing C-2). It also provides a control signal (SHIFT DISPLAY) to the Shift Register, which places it in the hold mode.

Two Type 74193 Synchronous 4-Bit Up/Down Counters (E6 and E8) are cascaded to provide an 8-bit counter (Figure 3-3). Cascading is accomplished by connecting the CARRY output (pin 12) of the first counter to the COUNT UP input (pin 5) of the second counter. The counter is used only in the count-up mode; therefore, the COUNT DOWN input is disabled by connecting it to +5 V, and the BORROW output is not used.

The preset feature is not used; thus, the LOAD input (pin 11) is disabled by connecting it to +5 V. The CLEAR input is not used so that the counter cannot be forced to 0 by an outside signal.

When the clock starts, the counter starts counting through its 256 states. It counts continuously as long as the clock is running.

Two modes of operation occur during one complete counting sequence (256 states) before overflow (all 0s) occurs and the sequence repeats. Output A of the first 4-bit section (E6) is the least significant bit; output D_2 of the second 4-bit section (E8) is the most significant bit. The first section advances from 0 through 15 (16 counts), overflows (goes to 0), and starts over. At overflow, a pulse from the CARRY output of the first section is sent to the COUNT UP input of the second section, which increments the second section by 1. After 16 overflows, the counter is full (all 8 bits are 1s) which represents 255_{10} or 256 counts. The next clock pulse causes both sections to overflow, which sets the counter to 0 and the sequence repeats.

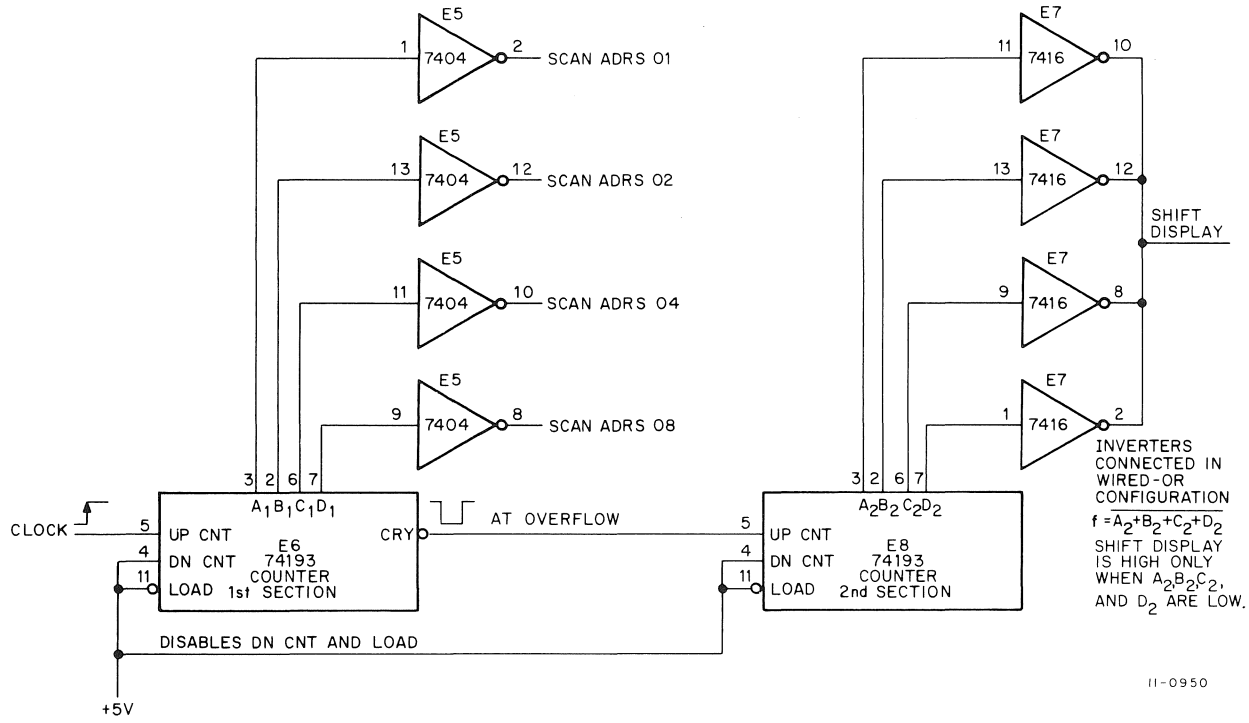


Figure 3-3 Counter, Simplified Logic Diagram

The output of the first counter section is the 4-bit scan address [SCAN ADRS 01 (1) L, 02 (1) L, 04 (1) L, and 08 (1) L]. The lines go to four Type 7404 Inverters (E5) and then to the select inputs of the data/address multiplexer. As the first section of the counter sequences through its 16 states, these lines cause the multiplexer to scan its 16 input lines and send the data serially to the Shift Register. Each of the four outputs of the second counter section goes to a Type 7416 Inverter Driver (E7). The open collectors of these inverters are connected together in a wired-OR configuration. The output is the SHIFT DISPLAY H signal, which is high only when all inverter inputs (E8 counter outputs) are low (0). The SHIFT DISPLAY H signal is a control signal input to the Shift Register. When it is low, the register is in the hold mode; when it is high, the register shifts serial data in to the right. The second counter section is 0 only during states 0 through 15. During this period, SHIFT DISPLAY H is high, and the Shift Register accepts serial data from the multiplexer and shifts it right. This data represents a complete scan of the 16 inputs to the multiplexer that are placed in the Shift Register. At state 16, a 1 is present in the second counter section. From this state, up to and including state 255, one or more 1s are present in the second counter section. The counter states are shown in Table 3-2. During this period, SHIFT DISPLAY H is low, and the Shift Register is in the hold mode. The data is static and is available for display.

A counter state change occurs in approximately 1 μs; therefore, the scan mode takes 16 μs and the hold mode lasts for 240 μs. During manual console operation, data and addresses are generated by positioning switches. The information on the multiplexer input remains stable for a long time in comparison to the 256 μs required for a counter scan/hold cycle. In effect, the multiplexer continually scans relatively stable information that can be displayed as static rather than transient information.

Table 3-2
Counter States

Counter State (Decimal)	Counter States								Remarks
	2nd Section				1st Section				
	D	C	B	A	D	C	B	A	
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	1	
2	0	0	0	0	0	0	1	0	States 0--15 are
3	0	0	0	0	0	0	1	1	scan mode. Data
4	0	0	0	0	0	1	0	0	is obtained and
5	0	0	0	0	0	1	0	1	loaded into Shift
.			.				.		Register.
.			.				.		
15	0	0	0	0	1	1	1	1	
16	0	0	0	1	0	0	0	0	
17	0	0	0	1	0	0	0	1	
18	0	0	0	1	0	0	1	0	
.			.				.		
.			.				.		
31	0	0	0	1	1	1	1	1	States 16--255 are
32	0	0	1	0	0	0	0	0	hold mode. Data
.			.				.		is held in Shift
.			.				.		Register for
239	1	1	1	0	1	1	1	1	display.
240	1	1	1	1	0	0	0	0	
.			.				.		
.			.				.		
255	1	1	1	1	1	1	1	1	
0	0	0	0	0	0	0	0	0	← Counter overflow

3.3.4 Display Buffer and Driver

The display buffer and driver logic consists of a 16-bit buffer and 16 inverter drivers for the ADDRESS/DATA Register lights (drawing C-1).

The 16-bit buffer is composed of four Type 8271 4-Bit Registers (E11, E10, E13, and E15). They are connected in a shift-right configuration with a serial data input; the last bit output (DO) of the preceding section is connected to the series data input (DS) of the following section (Figure 3-4). The parallel data inputs are not used. The reset input (RD) is disabled by connecting it to +5V. The LOAD input (pin 10) is connected to ground (logic 0), and the SHIFT input (pin 13) is connected to the SHIFT DISPLAY H signal from the counter. With the LOAD input held low, the operating mode of the buffer is controlled by the SHIFT input. When the SHIFT DISPLAY H signal is high, the buffer accepts data and shifts right; this is the console scan mode. When the SHIFT DISPLAY signal is low, the buffer holds the data; this is the console display mode.

Each Shift Register output signal is sent to a Type 7416 Inverter Driver to illuminate an associated light-emitting diode (LED). The 16 LEDs are the indicators for the ADDRESS/DATA Register display. A high output from the buffer causes the LED to illuminate, which indicates that the associated bit is a logical 1. The final stage of a 7416 inverter has an open collector that is connected to an LED, which in turn is connected to +5V via a current-limiting resistor (Figure 3-5). When the inverter input is low (logical 0 = 0V), no current can flow through the LED because there is no conducting path to ground through the transistor; therefore, the LED is not illuminated. A high (logical 1) inverter input puts a positive voltage on the base of the transistor, which turns it on. Current flows from the +5V source through the resistor, LED, and transistor emitter to ground, which illuminates the LED.

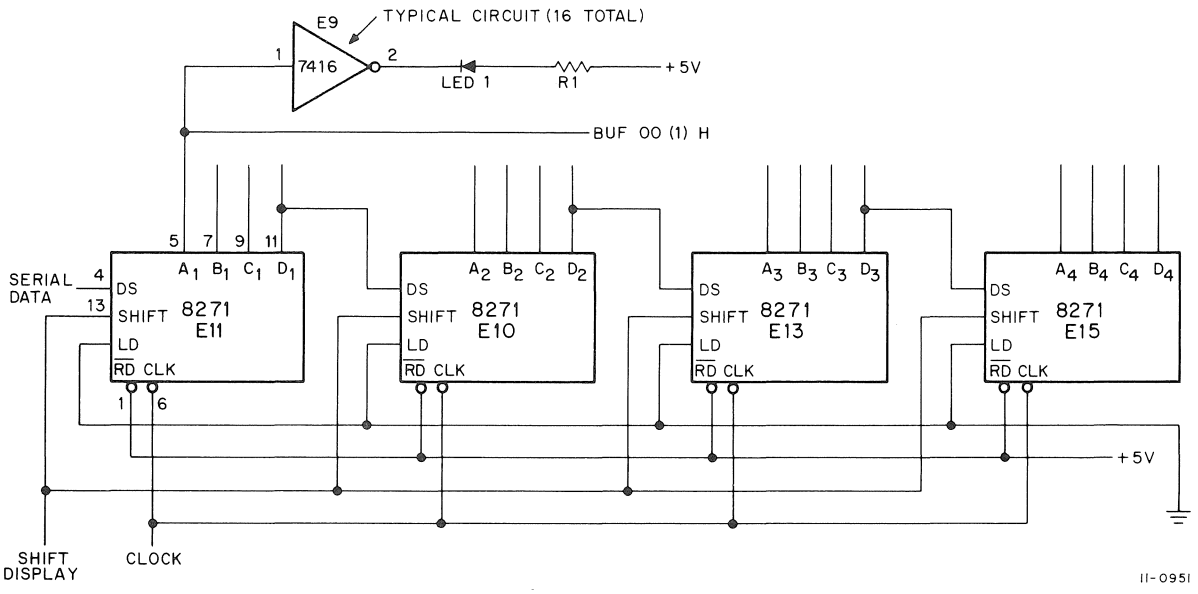


Figure 3-4 Display Buffer and Driver, Simplified Logic Diagram

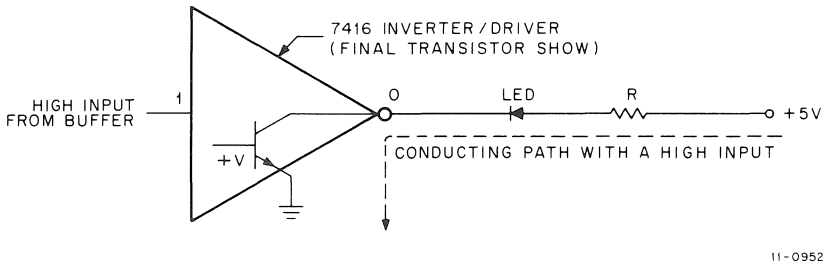


Figure 3-5 LED Driver Circuit

3.3.5 Control Switches and Logic

The console contains six control switches (drawing C-3). The HALT/ENABLE switch is a 2-position toggle type; HALT is the down position and ENABLE is the up position. The other five switches are momentary action type. They are: load address (LOAD ADRS), examine (EXAM), continue (CONT), deposit (DEP) and START. The DEP switch is activated when it is lifted; the others are activated when they are depressed.

A bounce buffer is connected across the output contacts of each switch to eliminate interruptions of the output signal due to contact bounce when the switch is activated. The bounce buffer is a latch constructed of two cross-coupled 7416 inverter buffer/drivers. When the switch is activated, the output signal is latched and any contact bounce, with accompanying signal loss, does not alter the output signal.

For the momentary action switches, the output is asserted low (logical 0) when the switch is activated. For the HALT/ENABLE switch, the output is asserted low when the switch is in the HALT position.

The input of each switch is connected to the output of a 7417 Open-Collector Non-Inverting Buffer. The inputs of all the 7417 buffers are connected to the output of a very simple logic network that detects power on/off and panel lock on/off. Power is sensed by monitoring the power-up signal (PUP L) from the processor. Panel lock is sensed by monitoring the PANEL LOCK signal from the OFF/POWER/PANEL LOCK switch on the console front panel. Panel lock is a mode of operation that disables all console control switches, it prevents inadvertent switch operation from disturbing a running program.

3.3.5.1 **Normal Operating Mode** – Normal operation is performed with PANEL LOCK off. This discussion is referenced to engineering drawing C-3 and Figure 3-6, which is a simplified logic diagram.

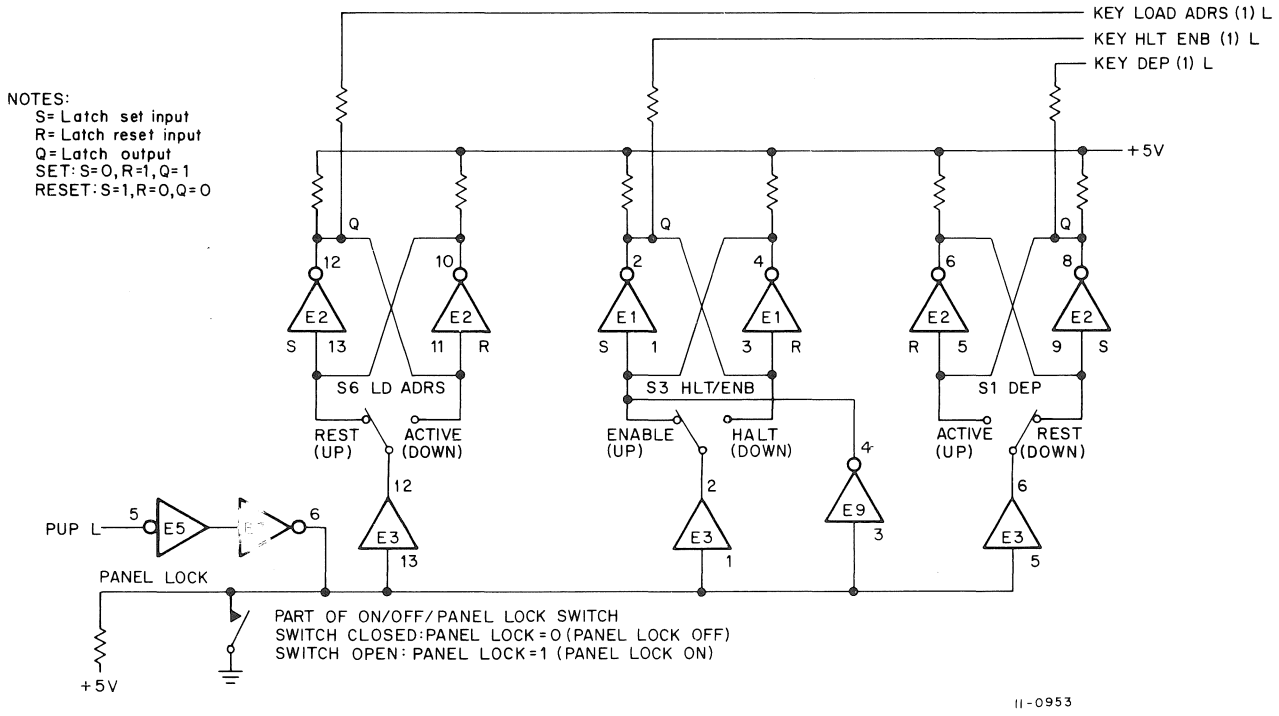


Figure 3-6 Control Switches and Bounce Buffers, Logic Diagram

The switch input logic network is composed of one 7404 Inverter (E5) and one 7416 Open-Collector Inverter (E7). The output of E7 is connected to PANEL LOCK, which is controlled by the key-operated ON/OFF/PANEL LOCK switch on the console panel. When the switch is in the PANEL LOCK position, the panel lock mode is activated and the PANEL LOCK signal is high (logical 1). When the switch is in the ON position, the PANEL LOCK signal is low (logical 0). This is accomplished by grounding the PANEL LOCK signal in this switch position. The output of E7 (pin 6), which represents the state of input PUP L, is connected to the PANEL LOCK signal line. This point is the input to all switch 7417 buffers (E3). It is high only when PUP L and PANEL LOCK are both high.

With PANEL LOCK off, the PANEL LOCK signal is low and the input to each switch is low. [Refer to momentary action switch S6 (LOAD ADRS), which is typical of the five switches of this type.] The set input of the latch is the rest terminal, and the reset input is the active terminal. With S6 in the rest position, a 0 is placed on the set input of the latch (E2 pin 13). The latch is set (S=0, R=1, Q=1) and the output (E2 pin 12) is high, which is the non-asserted state of the switch output [KEY LOAD ADRS (1) L = 1]. With S6 in the active position, a 0 is placed on the reset input of the latch (E2 pin 11). The latch is reset (S=1, R=0, Q=0) and the output (E2 pin 12) is low, which is the asserted state of the switch output [KEY LOAD ADRS (1) L = 0]. Note that the DEP switch (S1) is electrically identical to S6 but its active position is up rather than down.

With the HALT/ENABLE switch (S3) in the ENABLE (up) position, the latch is set and the switch output signal KEY HLT ENB (1) L = 1, which is its non-asserted state. This state allows a program to run. In the HALT (down) position, KEY HLT ENB (1) L = 0 is the asserted state and halts an operating program. Type 7416 Open-Collector Inverter E9 is used for power loss compensation and is described in a subsequent paragraph. In the normal operating mode, it has no effect on the switch operation.

3.3.5.2 **Panel Lock Mode** – In the panel lock mode, the PANEL LOCK signal is high (+5V via resistor R42). All switch inputs are now high. Panel lock is applied after a program has started in the normal operating mode. All momentary action switches are in the rest position; switch outputs are high (not asserted) because the latches have been set previously (S=0, R=1, Q=1). The HALT/ENABLE switch is in the ENABLE position; the switch output is high (not asserted) because the latch has been set previously (S=0, R=1, Q=1). With respect to the momentary action switches, the high on the switch input has no effect if the switch is moved to the active position, because it puts a 1 on the reset input of the latch whose reset input is already a 1. With respect to the HALT/ENABLE switch, the high on the switch input has no effect if the switch is moved to the HALT position because it puts a 1 on the reset input of the latch whose reset input is already a 1. Remember that the momentary action switch latches had been set (S=0, R=1, Q=1), and the HALT/ENABLE switch latch had also been set.

In this mode of operation, inadvertent switch operation cannot halt or otherwise alter a running program.

3.3.5.3 **Power Loss During Operation** – The processor contains a power fail circuit that allows the computer to tolerate an ac power loss without adverse effects. If a power loss occurs in the normal operating mode (panel lock off), the switches perform the functions determined by their current positions as soon as the +5V logic supply voltage is reestablished. PANEL LOCK = 0 is the signal that provides normal switch operation in this case.

If a power loss occurs in the panel lock mode, a forcing signal is required to ensure that the latches are driven to the states commensurate with the switch positions before the PANEL LOCK signal is applied again. Without the forcing signal, the latches could be set or reset in a random manner not related to switch position as the +5V logic supply voltage is reestablished.

As ac power is restored, PUP L is forced low for approximately 70 ms. This applies a 0 to the switch inputs to force the latches to the states commensurate with the switch positions: all momentary action switches are not asserted and KEY HLT ENB (1) L is not asserted (HALT/ENABLE switch in ENABLE position). The processor resumes operation and when PUP L goes high again, the panel lock mode is reestablished.

If the HALT/ENABLE switch is inadvertently placed in the HALT position during processor operation in the panel lock mode, the processor does not halt. However, if a power loss occurs with the switch in the HALT position, PUP L going low during the power-up sequence resets the latch and its output, KEY HLT ENB (1) L, is low, which halts the program. When PUP L goes high again and the panel lock mode is reestablished, the 1 on the switch input does not set the latch or eliminate the HALT signal.

Open-collector inverter E9 solves this problem. When PUP L goes high during the power-up sequence, the 1 on the switch input is also inverted by E9 and a 0 is placed on the set input (E1 pin 1) of the latch. The latch is set and its output is not asserted [KEY HLT ENB (1) L = 1], which allows the processor to resume operation even though the switch is in the HALT position.

CHAPTER 4
KD11-B DETAILED DESCRIPTION

4.1 INTRODUCTION

This chapter describes the logic and physical implementation of the KD11-B data path (DP), data path control (DPC), Unibus control, serial communications line (SCL), and the line clock. Extensive use is made of bipolar, medium and large scale integrated circuits in the processor. There are 28 read-only memories (ROMs) used in the KD11-B. Details of the microprogram are described in Chapter 2.

4.2 ROMs AS GENERALIZED GATES

With the increasing availability of inexpensive bipolar ROMs, it is possible to replace rather complex combinational logic structures with one or two 16-pin dual in-line integrated circuits. In the processor, extensive use is made of two different ROM formats. As shown in Figure 4-1, one format stores 256 bits (b), arranged in 32 words of 8 bits each.

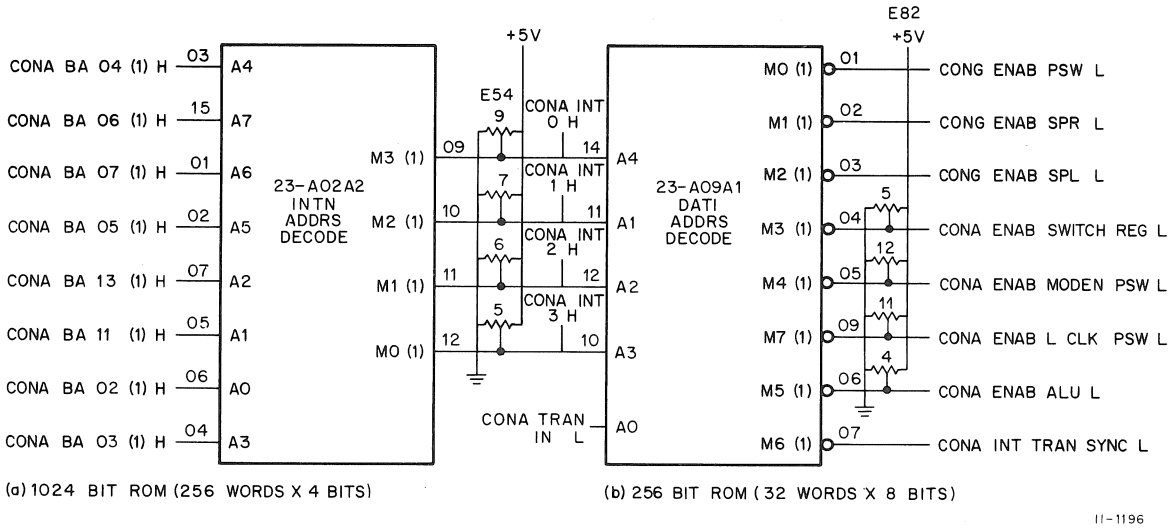
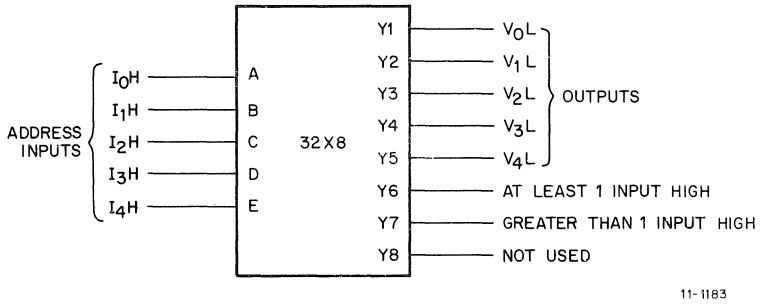


Figure 4-1 1024-Bit and 256-Bit ROMs

The other format stores 1024 bits(a), arranged in 256 words of 4 bits each. The 32-word ROM has 5 address lines, 1 output enable line, and 8 outputs. The 256-word ROM has 8 address lines, 2 output enable lines, and 4 outputs. Both devices have open-collector outputs.

Figure 4-2 illustrates the use of a 32 X 8 ROM as a generalized gate. In the example, a 32 X 8 ROM is used as a 5-input priority encoder.

A similar priority encoder is used in the KD11-B on print CONE where it is necessary to decide which switch function to perform if more than one console switch is depressed.



1 of 5 Priority Encoder

Truth Table

E	D	C	B	A	Address Octal	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	1	0	0
0	0	0	1	0	2	0	1	0	0	0	1	0	0
0	0	0	1	1	3	0	1	0	0	0	1	1	0
0	0	1	0	0	4	0	0	1	0	0	1	0	0
					.								
					.								
0	0	1	1	1	7	0	0	1	0	0	1	1	0
0	1	0	0	0	8	0	0	0	1	0	1	0	0
					.								
					.								
0	1	1	1	1	17	0	0	0	1	0	1	1	0
					.								
					.								
					.								
1	0	1	1	1	27	0	0	0	0	1	1	1	0
					.								
					.								
					.								
1	1	1	1	1	37	0	0	0	0	1	1	1	0

Figure 4-2 32 X 8 ROM used as Generalized Gate

Many situations arise in which five or fewer input conditions result in combinations of eight or fewer output conditions where a 32 X 8 ROM is used for implementing the function. Similar applications apply to 256 X 4 ROMs. For example, the KD11-B uses one 256 X 4 ROM to test all of the PDP-11 conditional branch instructions against the C, N, V, and Z condition code bits. The branch decode ROM may be found on print DPG in position E072.

4.3 KD11-B DATA PATH, SIMPLIFIED DESCRIPTION

Figure 4-3 contains a simplified diagram of the KD11-B data path. The heart of the DP is an arithmetic-logic unit (ALU), which is capable of performing 16 Boolean operations and 16 different arithmetic operations on two 16-bit binary variables. The inputs to the ALU are storage registers on the A-leg input and the B-leg input. The output of the ALU feeds into a switch that is capable of introducing external data into the DP from the Unibus.

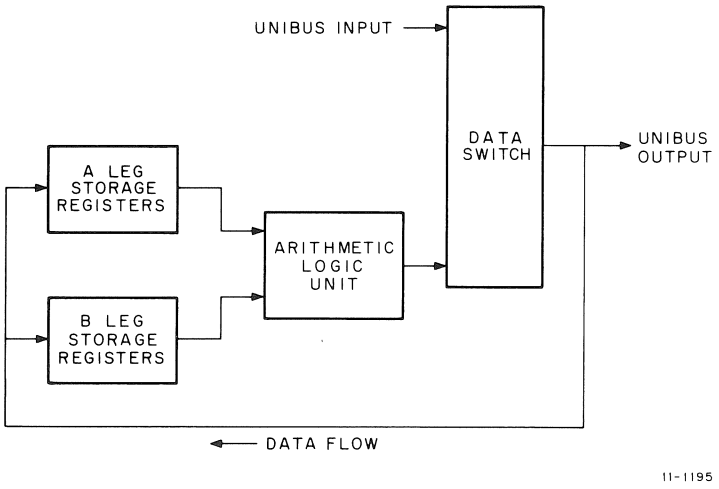


Figure 4-3 KD11-B Simplified Data Path Block Diagram

4.3.1 Data Path (DP) Detailed Description

Figure 4-4 is a detailed block diagram of the KD11-B.

It is important to recognize that this DP consists of a number of interconnected registers that are capable, when properly controlled, of executing the PDP-11 instruction set.

4.3.2 DP Data Polarities

It is useful to note the data polarity at various places in the processor. There are two signal levels used in the KD11-B. A high signal is represented by a voltage of +3 V to +5 V. A low signal is represented by a voltage between 0 V and 0.4 V. Positive and negative data polarities are defined as follows:

Negative Data Polarity: Logic 1 = Low Signal = 0–0.4 V
Logic 0 = High Signal = 3–5 V

Positive Data Polarity: Logic 1 = High Signal = 3–5 V
Logic 0 = Low Signal = 0–0.4 V

Data polarity is negative on the Unibus and within the dotted lines surrounding the ALU as shown in Figure 4-4. Throughout the remainder of the processor the data polarity is generally positive. In the KD11-B print set, the polarity of the asserted logic signal is given. For example, the signal DPF LOAD IR L is asserted, true, or logic 1, when it is at 0 V (low signal).

4.3.3 Control Logic and Microprogramming (CON)

The DPC is shown in Figure 4-4 at the left side of the drawing. All functions performed by the processor, including instruction interpretation, trap handling, and Switch Register (SR) function execution, depend upon the contents of the control store (CS). For each PDP-11 action performed by the KD11-B, the DPC executes a sequence of microsteps stored in the CS.

The microprogram contained in the CS consists of a series of microroutines which, when executed in the proper sequence, enable the KD11-B to perform as a PDP-11 processor. Details of the microprogram are described in Chapter 2.

The CS consists of ten 256 X 4 bipolar ROMs, shown on prints CONF and CONG. The outputs of the ROMs are used to control the registers and arithmetic elements in the DP. The current control step (microstep) is stored in a microprogram counter (MPC). The MPC is an 8-bit latch that is loaded at intervals of approximately 310 ns with a number generated by the output of the NXT field of the CS, wire-ORed with the outputs of the microbranch network.

4.3.4 A-Multiplexer

The A-Multiplexer (AMUX) is a 2-word, 16-bit multiplexer composed of four Type 8266 2-Input, 4-Bit Digital Multiplexers. The AMUX representation is contained in four logic prints as shown below.

AMUX Designation	Print
E010	DPA
E009	DPB
E008	DPC
E007	DPD

The A-word input to the AMUX is the output of the ALU, and the B-word input consists of the Unibus data lines D (15:00). These data signals are taken from the Unibus via four Type 8838 single input inverters (called bus receivers). The receiver designations and locations are listed below.

Receiver Designation	Unibus Data Bits	Print
E004	BUS D00–D03	DPA
E003	BUS D04–D07	DPB
E002	BUS D08–D11	DPC
E001	BUS D12–D15	DPD

The AMUX A-input is inverting and the B-input is non-inverting. Word selection is based on the state of select signal inputs S0 and S1 as shown in the following truth table.

Select Signals		Output
S1	S0	f3, f2, f1, f0
L	L	B
L	H	A

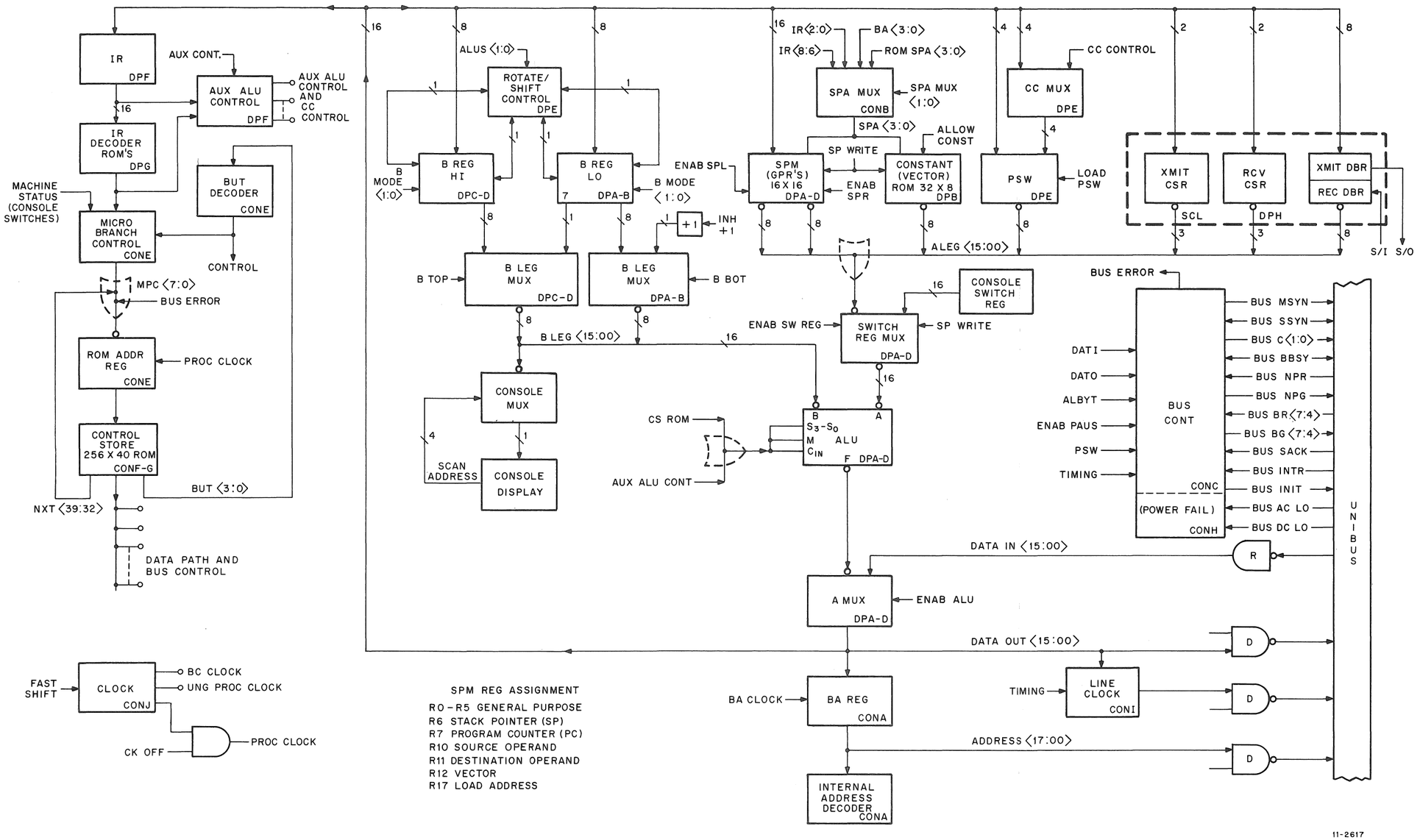


Figure 4-4 KD11-B Detailed Block Diagram

Select signal S0 is CONA ENAB ALU H and S1. S1 is connected to ground so it is always low; therefore, signal CONA ENAB ALU H controls the selection of the input. When it is low, the B-input is selected, and when it is high, the A-input is selected.

The 16-bit output of the AMUX is sent to several places as shown below.

Destination	Print
Bus Address Register	CONA
Unibus Drivers	DPA-DPD
Instruction Register	DPF
B-Register	DPA-DPD
Scratch Pad Memory	DPA-DPD
PSW Logic	DPE
SCL Control Logic	DPH
Line Clock	CONI

4.3.5 Arithmetic Logic Unit (ALU)

The arithmetic logic unit (ALU) is the heart of the data path logic. It performs 16 Boolean operations and 16 arithmetic operations on two 16-bit words. Not all of these arithmetic and logic operations are in the KD11-B; Table 2-1 lists the operations that are used. The truth table for the 74181 ALU (Appendix A) shows all the operations that are available.

The ALU is composed of four Type 74181 Arithmetic Logic Unit/Function Generators and one Type 74182 Look-Ahead Carry Generator. The symbolic representation of the ALU is contained on four logic prints as shown below.

Device	Component Designation	Print
74181	E027	DPA
74181	E026	DPB
74181	E025	DPC
74181	E024	DPD
74182	E031	Sections shown on prints DPA, DPB, DPC, and DPD.

Figures 4-5 and 4-6 show the signal and pin designations for the 74181 and 74182, respectively.

For clarity, a detailed block diagram of the ALU is shown in Figure 4-7. The 16-bit A-word input is fed by ALEG (15:00) and the 16-bit B-word input is fed by BLEG (15:00). The ALEG is fed by the Switch REG MUX. The Switch REG MUX sources are: scratch pad memory, constants generator, processor status word logic, console Switch Register, and serial communications line. The BLEG sources are: B register, sign extension logic and +1 logic. Each of these sources is discussed in subsequent paragraphs. The 16-bit ALU output is sent to the A-word input of the AMUX. The AMUX is a 2-word, 16-bit multiplexer composed of four Type 8266 2-Input, 4-Bit Digital Multiplexers. The source of the B-word input to the AMUX is Unibus data bits D (15:00).

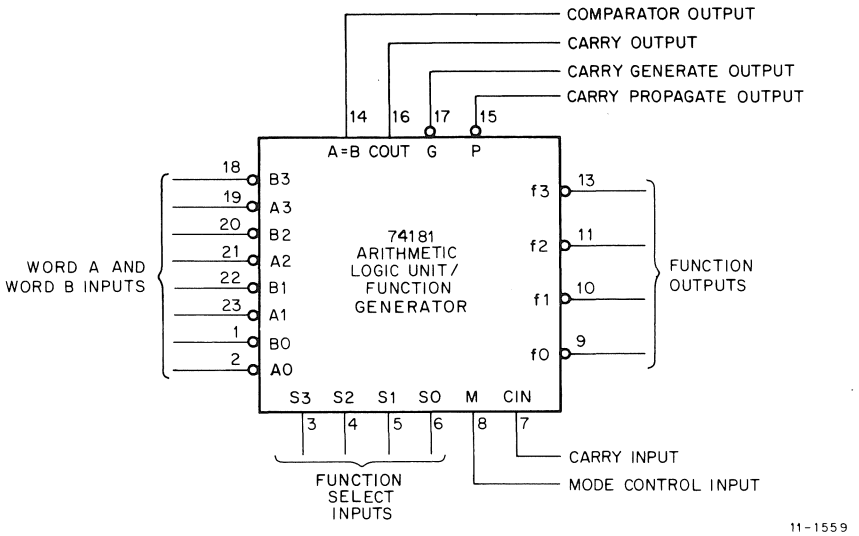


Figure 4-5 74181 Pin and Signal Designations

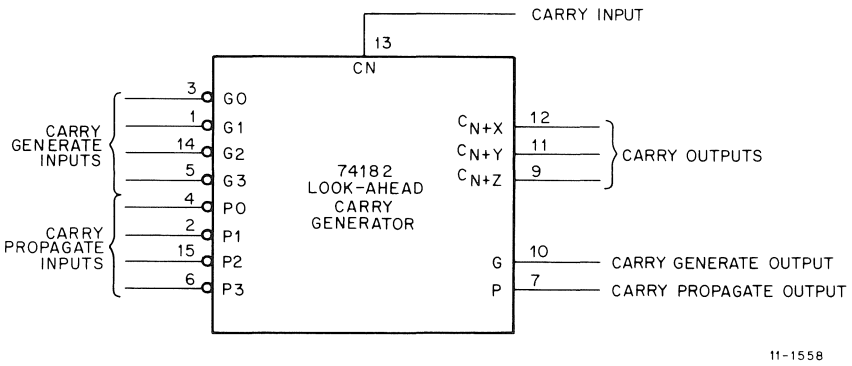


Figure 4-6 74182 Pin and Signal Designations

The ALU is controlled by six input signals (Table 4-1) that select the mode (logic or arithmetic) and the desired function. The primary source for the control signals is the control store logic (print CONF). A wire-ORed connection allows the control signals to also be obtained from the aixiliary ALU control (print DPF) and the IR decoding logic (print DPG). The four function select signals (CONF ALU S0 L-S3 L) are buffered and inverted by Type 7437 NAND Buffers (print DPA) before they are sent to the ALU select inputs. After buffering, they are identified as DPA ALU S0 H-S3 H. Mode signal CONF ALU MODE H is sent directly to the ALU. The carry input signal (CONF CIN H) is sent to one input of exclusive-OR gate E068 (print DPA). The output of this gate is signal DPA ALU CIN 00 H and is sent to the carry inputs of both the 74181 ALU and 74182 carry generators. This signal is also controlled by signals DPF COP L and DPE COUT (1) L. They are inputs to NOR gate E017 which supplies the other input to exclusive-OR gate E068. Signal DPF COP L is an output of E054 SOP AUX CTL ROM 23-A03A1 (print DPF) in the auxiliary ALU control logic. Signal DPE COUT (1) H is an output of E052 COUT flip-flop (print DPE) in the PSW logic.

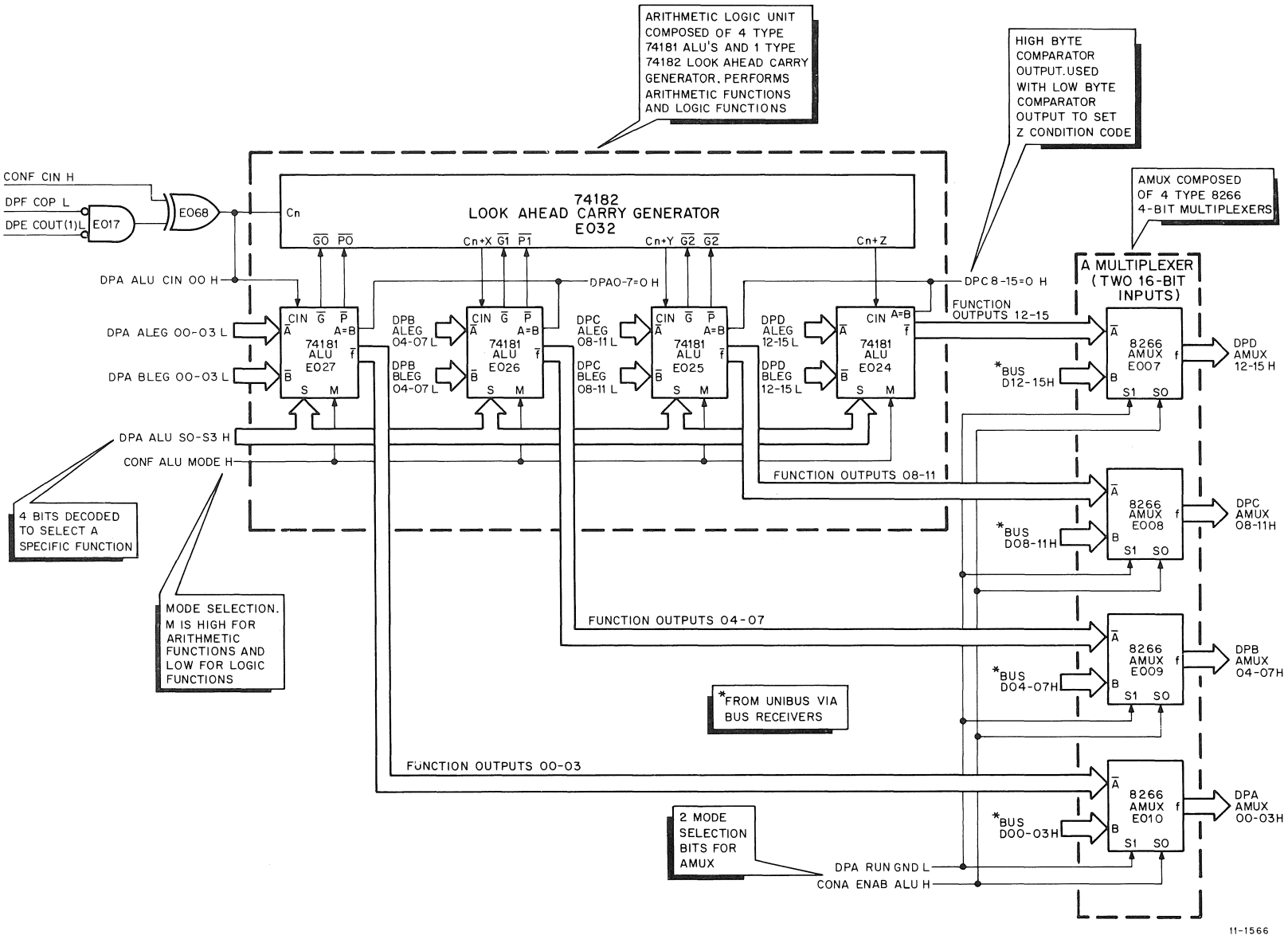


Figure 4-7 Arithmetic Logic Unit Block Diagram

Table 4-1
ALU Control Signals

Control Signal	Control Store A11A2 E114 CONF	Control Store A20A2 E103 CONF	DOP Aux Cont ROM23-A02A1 E053 DPF	SOP Aux Cont ROM23-A03A1 E054 DPF	IR Decode ROM23-A05A1 E059 DPG
CONF ALUS3 L	X		X	X	
CONF ALUS2 L	X		X	X	X
CONF ALUS1 L	X		X	X	
CONF ALUS0 L	X		X	X	X
CONF ALU MODE H		X	X	X	
CONF CIN H		X	X	X	

The A-B terminal of each 74181 ALU is an open-collector comparator output that is high when the input words are equal and the ALU is in the subtract mode. These outputs are wire-ANDed for each data byte to generate equality signals that are used in forming the Z condition code. Signal DPA 0-7=0H indicates that the inputs to the low data byte are equal to zero. Signal DPA 8-15=0 H indicates that the inputs to the high data byte are equal to zero.

4.3.6 B Register

4.3.6.1 Functional Description – The B register (BREG) is the only storage register in the B-leg of the ALU. The BREG is used as a general purpose register to store the results of any operation that requires data to be read from the SPM. It is used as a shift-left/shift-right register to perform rotate, shift, and byte instructions.

The BREG output is attached to additional logic to permit its lower byte to be sign-extended during execution of byte and branch instructions. Logic is also provided to place the constant +1 on the B-leg. This operation is used in the process of incrementing or decrementing the general registers by 2. This discussion covers the BREG and the additional logic.

The BREG consists of four Type 74194 4-Bit Bidirectional Universal Shift Registers. The register designations and locations are listed below.

74194 Designation	Print
E015	DPA
E014	DPB
E013	DPC
E012	DPD

The additional logic required for the sign extension and +1 operations is listed below.

Device	Component Designation	Print
74S158	E021	DPA
7404	E058	DPA
7400	E020(4)	DPB
74S158	E019	DPC
74S158	E018	DPD

For clarity, a simplified logic diagram of the BREG and associated logic is shown in Figure 4-8. The inputs to the BREG consist of the 16 bits from the AMUX. They are identified as:

DPA AMUX 00 H - 03 H
DPB AMUX 04 H - 07 H
DPC AMUX 08 H - 11 H
DPD AMUX 12H - 15H

The corresponding BREG outputs are:

DPA BREG 00 (1) H - 03 (1) H
DPB BREG 04 (1) H - 07 (1) H
DPC BREG 08 (1) H - 11 (1) H
DPD BREG 12 (1) H - 15 (1) H

The BREG is clocked by DPA PROC CLOCK L, which is processor clock signal CONJ PROC CLOCK H that has been buffered and inverted by gate E089 pin 03 (print DPA).

The type of operation performed by the BREG is determined by the states of mode control inputs S1 and S0 as shown below.

Mode Control S1	S0	Operation
H	H	Parallel Load
L	H	Shift Right (towards LSB)
H	L	Shift Left (towards MSB)
L	L	Hold (clock inhibited)

The BREG is loaded when both mode control inputs (S1 and S0) are high. These inputs are selected by control store field BRG (CS word bits 04 and 05). The signals are: CONG B MODE 01 H (bit 04) that is sent to input S1; and CONG B MODE 00 H (bit 05) that is sent to input S0. Shift-right, shift-left, and hold operations are also controlled by the BRG field (see CS word format in drawing D-CS-M7261-0-1, sheet 14). The shift-right (SR) and shift-left (SL) inputs are the serial data inputs that are used only during shifting operations. They are discussed in subsequent paragraphs.

The eight bits that constitute the low byte of the BREG are sent to the BLEG MUX and the NAND gates whose outputs are sent to the B-word input of the ALU. Bits DPA BREG 00 (1) H–03 (1) H are connected to Type 74S158 2-line-to-1-line multiplexer. The multiplexer is used to select +1 or bits 00 through 03 from the BREG. Bits DPB BREG 04 (1) H–07 (1) H are connected to 7400 NAND gates. All eight bits are enabled by signal CONG BBOT H which is one of two bits of the B-leg control field of the control store word. It is generated by CS ROM E115 (print CONG).

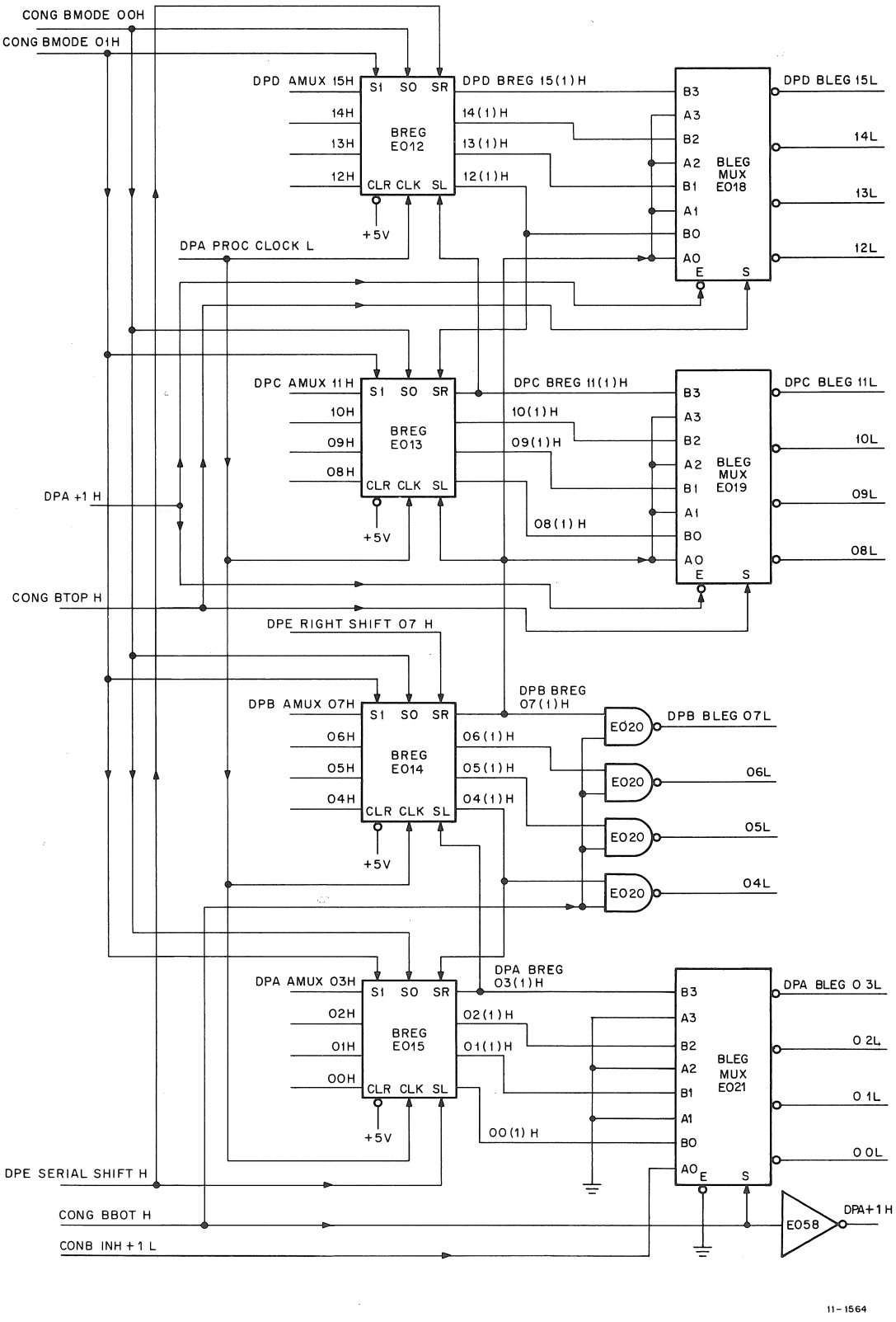


Figure 4-8 B Register and Output Logic

The eight bits that constitute the high byte of the BREG are sent to the B-word inputs of the BLEG MUX. The A-word inputs of the BLEG MUX are connected in common to DPB BREG 07 (1) H. The BLEG MUX is composed of two Type 74S158 2-line-to-1-line Multiplexers. Input word selection is controlled by the select (S) input when the enable (E) input is asserted low as shown below.

Enable (E) Input	Select (S) Input	Output
H	X	H
L	L	A-word
L	H	B-word

The select (S) signal is CONG BTOP H, which is the other bit of the B-leg control field of the control store word. It is generated by CS ROM E106. The enable (E) signal is DPA+1 H from the +1 logic.

4.3.6.2 **BLEG Operations That Provide Input to the ALU** – The following discussion covers the three operating modes of the BLEG that provide input data to the ALU. The modes are: BREG unmodified, BREG sign extended, and generation of the constant +1. The output of the BREG is gated onto the BLEG in a manner dictated by the BLEG mode of operation. The circuits involved are the BLEG MUX, +1 logic, and gate E020 on the outputs of BREG bits 00–07. The discussion is not concerned with the operation of the BREG.

Control of the BLEG operating mode is provided by control store field BLG. This field is physically split in the CS word as shown in Table 4-2.

Table 4-2
Control Store Signals for BLEG Operations

Control Store		ROM	Output Signal	
Bit	Field	No.	Name	Function
14	BTP	E106	CONG BTOP H	Controls BREG output bits 08–15 (high byte)
16	BBT	E115	CONG BBOT H	Controls BREG output bits 00–07 (low byte)

The following truth table shows the states of CS bits 14 and 16 for the three BLEG modes.

Bit 16 BBT	Bit 14 BTP	BREG Mode
H	H	BREG Unmodified
H	L	Sign Extend BREG
L	L	Generate Constant +1

In the BREG unmodified mode, it is desired to place the unmodified contents of the BREG on the BLEG. Control store field BLG makes both CONG BTOP H and CONG BBOT H high. Signal CONG BBOT H enables the low byte of the BREG onto the BLEG via gate E020 and BLEG MUX E021. Signal CONG BBOT H is inverted by E058 and is identified as DPA +1 H. It is the enabling signal for the BLEG MUX (E018 and E019). In this case, it is low and enables the BLEG MUX. The select (S) signal for the BLEG MUX is CONG BTOP H, which is high. This selects the B-word input of the BLEG MUX which is the high byte of the BREG output. Thus, the 16-bit output of the BREG is transferred to the BLEG unmodified.

In the BREG sign-extended mode, it is desired to place the unmodified low byte of the BREG on the BLEG and sign extend the high byte, which makes bits 08–15 the same as bit 07. The sign-extended high byte is also placed on the BLEG. Control store field BLG makes CONG BBOT H high and CONG BTOP H low. Signal CONG BBOT H enables the low byte of the BREG onto the BLEG via gate E020 and BLEG MUX E021.

Bit 07 [DPB BREG 07 (1) H] from the BREG is sent to all eight A-inputs of the BLEG MUX. The BLEG MUX is enabled by DPA +1 H which is low. Select signal CONG BTOP H is low which selects the A-word of the BLEG MUX. This is the high byte of the B-leg but all eight bits are identical and equal to the state of DPB BREG 07 (1) H. The sign of bit 07 is extended to bits 08–15 and these bits, along with unmodified bits 00–07, are transferred to the B-leg.

In the +1 operation, it is desired to place the constant +1 on the BLEG. The constant +1 is placed in the LSB position (bit 00) and all other bits (01–15) are forced to 0. Control store field BLG makes CONG BBOT H and CONG BTOP H both low. Signal CONG BBOT H disables BLEG MUX E021 and gate E020, which drives BLEG bits 00–07 high: Signal CONG BBOT H is inverted by G058 to produce DPA +1 H, which is high. Signal CONB INH +1 L controls the activation of the +1 logic. Signal DPA +1 H is high, so it disables the BLEG MUX and drives all its outputs high (bits 08–15). This operation places the constant +1 into the BLEG.

The BLEG is the input to the B-word of the ALU that uses negative logic. The +1 operation can be readily seen by looking at the BLEG bits with respect to their logical states as follows:

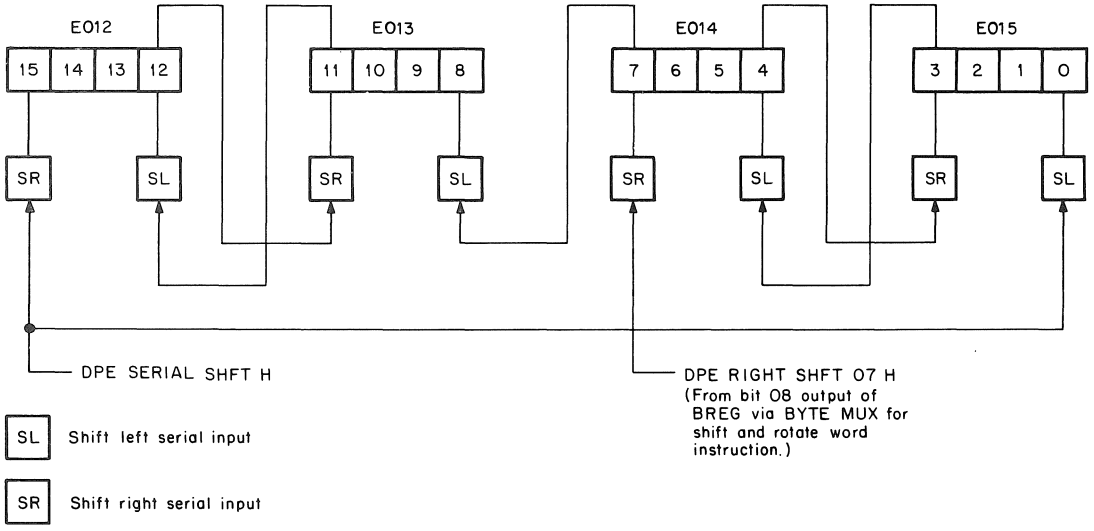
BLEG 00 = Low = Logical 1
BLEG 01–15 = High = Logical 0

4.3.6.3 BREG Shifting Operations – The BREG is used as a left/right shift register to perform rotate, shift, and odd byte instructions. The following discussion covers the shifting process and generation of the serial shift input for the ASL, ASR, ROL, and ROR instructions. An explanation of the BREG operation during the performance of byte instructions is discussed separately.

The key to this discussion is the symbolic representation of the bit structure of the BREG as shown in Figure 4-9. Each of the four 74194 Shift Registers that make up the BREG has a shift-left (SL) serial input and a shift-right (SR) serial input. The SL input is connected to the lowest order bit and the SR input is connected to the highest order bit. The four devices are interconnected to provide shift-left and shift-right paths for the 16-bit BREG. For clarity, the parallel inputs and outputs are not shown.

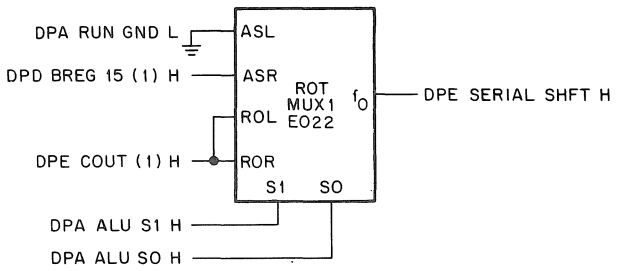
Mode control signals CONG BMODE 00 H and CONG BMODE 01 H select a shift-left or shift-right operation. In shifting operations that deal with instructions ASL, ASR, ROL, and ROR, the same source is used for serial input data for a shift left or a shift right.

The SL serial input goes to BREG bit 00 and the SR serial input goes to BREG bit 15. When SL or SR is enabled, the other input is disabled. The signal name for the serial data input is DPE SERIAL SHFT H which is the f_1 output of ROT MUX 1 E022 (print DPE). Depending on the instruction being processed, DPE SERIAL SHFT H can load the appropriate BREG serial input with a 0 (for ASL), bit 15 of the BREG output (for ASR), or the C-bit (for ROL and ROR).



B Register Bit Structure

Value of SERIAL SHFT H



TRUTH TABLE

S1	SO	Funct. (f_1)
L	L	ROR
L	H	ASR
H	L	ROL
H	H	ASL

Generation of DPE SERIAL SHFT H

11-1567

Instruction	Value of DPE SERIAL SHFT H	Remarks
ASL	DPA RUN GND L	0 to BREG bit 0 via SL input
ASR	DPD BREG 15 (1) H	Bit 15 of BREG output to bit 15 of BREG via SR input
ROL	DPE COUT (1) H	C bit to BREG bit 0 via SL input
ROR	DPE COUT (1) H	C bit to BREG bit 15 via SR input

Figure 4-9 B Register Shift Signal Inputs

The values of DPE SERIAL SHFT H and the truth table for the ROT MUX 1 are shown in Figure 4-9.

This register handles byte shifting also as required by instructions ASLB, ASRB, ROLB, and RORB. Signal DPE RIGHT SHFT 07 H is used as a serial right (SR) input to bit 07 to handle replication of bit 07 for an ASRB instruction and to load the previous contents of the C-bit for an RORB instruction. This signal is also required to perform the word shifting for instructions ASR and ROR because there is no direct connection between bits 08 and 07 for a shift-right operation. Signal DPE RIGHT SHFT 07 H is generated by BYTE MUX E023 and it represents BREG output bit 08 (DPC BREG 08 H) during word instructions ASR and ROR.

The shifting requirements for the ASL, ASR, ROL, and ROR instructions are described briefly below.

Arithmetic Shift Left (ASL) – Shifts all bits left one place. Bit 0 loaded with a 0.

The BREG is shifted left one place. ROT MUX 1 selects ASL input (DPA RUN GND L) which is logical 0 because it is connected to ground. DPE SERIAL SHFT H = 0 and is loaded into BREG bit 00 via the SL input.

Arithmetic Shift Right (ASR) – Shifts all bits right one place. Bit 15 is loaded with BREG output bit 15.

The BREG is shifted right one place. ROT MUX 1 selects ASR input [DPD BREG 15 (1) H], which is output bit 15 of the BREG. DPE SERIAL SHFT H equals the bit 15 output of the BREG and is loaded into BREG bit 15 via the SR input. This is replication of bit 15. DPE RIGHT SHFT 07 H equals the bit 08 output of the BREG and is loaded into BREG bit 07 via the SR input to provide the connection from bit 08 to bit 07.

Rotate Left (ROL) – Rotates all bits left one place. Bit 00 loaded with C-bit.

The BREG is shifted left one place. ROT MUX 1 selects ROL input [DPE COUT (1) H], which is the value of the C-bit prior to execution of the instruction. DPE SERIAL SHFT H equals this value of the C-bit and is loaded into BREG bit 00 via the SL input.

Rotate Right (ROR) – Rotates all bits right one place. Bit 15 loaded with C-bit.

The BREG is shifted right one place. ROT MUX 1 selects ROR input [DPE COUT (1) H], which is the value of the C-bit prior to execution of the instruction. DPE SERIAL SHFT H equals this value of the C-bit and is loaded into bit 15 via the SR input. DPE RIGHT SHFT 07 H equals the bit 08 output of the BREG and is loaded into BREG bit 07 via the SR input to provide the connection from bit 08 to bit 07.

In each of these instructions, the C-bit is loaded with a new value from the BREG. This function is discussed in the description of the PSW logic.

4.3.7 Byte Instructions

For the correct execution of all instructions that operate on data, the least significant bit of both the source and destination must line up with bit 0 of the A-leg and B-leg, respectively. This same rule applies even if the instruction being executed is a byte operation. For even bytes this is no problem, since the data received from the Unibus has the least significant bit of the low order byte lined up properly. For odd bytes, it is necessary to shift the data word right eight bit positions to properly line up the data. Then if the destination is an odd byte, the data must be shifted eight bits left before it is restored to its proper memory location. This operation is illustrated in the example in Figure 4-10 with the associated processor flow. Note that bytes are always sign extended (data path bits 15:8 duplicating bit 7); often they are lined up in the low order position.

4.3.8 Scratch Pad Memory

The scratch pad memory (SPM) is a 16-word by 16-bit random access read/write bipolar memory composed of four Type 7489 16-word by 4-Bit Memory Units. A block diagram of the SPM is shown in Figure 4-11. The SPM representation is contained on four logic prints as shown below.

SPM Designation	Print
E040	DPA
E039	DPB
E038	DPC
E037	DPD

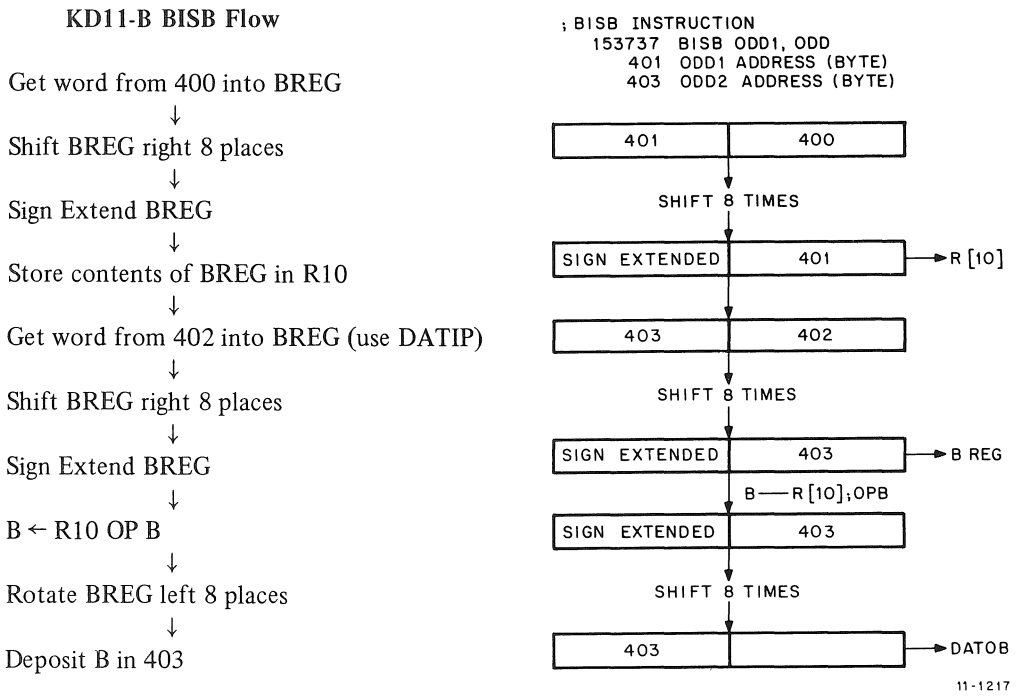


Figure 4-10 Byte Format for Shifting Instructions

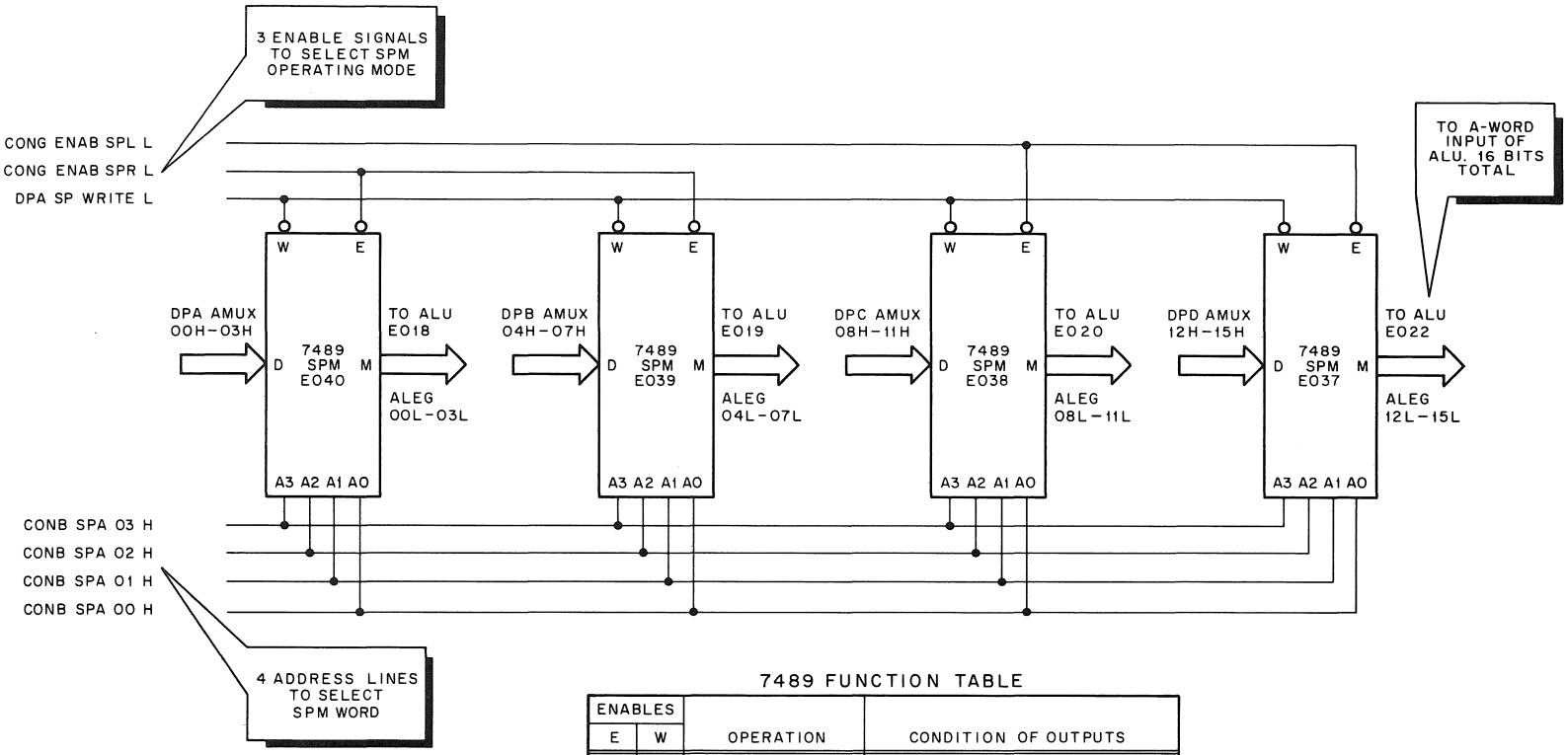
The 16-word by 16-bit organization of the SPM provides 16 storage registers that are utilized as shown in Table 4-3.

Table 4-3
Register Utilization in SPM

Register Number	Designation
R0–R5	General Purpose
R6	Processor Stack Pointer
R7	Program Counter
R8 and R9	Unused
R10	Source Operand Storage
R11	Destination Operand Storage
R12	Interrupt Vector
R13–R16	Unused
R17	Load Address Storage

The SPM data inputs are the AMUX outputs: DPA AMUX 00 H–03H, DPB AMUX 04 H–07 H, DPC AMUX 08 H–11 H, and DPD AMUX 12H–15 H. The SPM outputs are the 16 ALEG bits (ALEG 00–15) which are sent to the A-word input of the ALU.

The SPM address line input signals are generated by the scratch pad address multiplexers (SPAM) as shown in Table 4-4.



7489 FUNCTION TABLE

ENABLES		OPERATION	CONDITION OF OUTPUTS
E	W		
L	L	WRITE	COMPLEMENT OF DATA INPUTS
L	H	READ	COMPLEMENT OF SELECTED WORD
H	L	INHIBIT STORAGE	COMPLEMENT OF DATA INPUTS
H	H	DO NOTHING	HIGH

11-1563

Figure 4-11 Block Diagram and Function Table for Scratch Pad Memory

Table 4-4
SPM Address Line Signals

Signal	Source
CONB SPA 00 H	SPA MUX E056 (print CONB)
CONB SPA 01 H	SPA MUX E056 (print CONB)
CONB SPA 02 H	SPA MUX E057 (print CONB)
CONB SPA 03 H	SPA MUX E057 (print CONB)

Two enable inputs control the SPM mode of operation: input W (memory enable) and input E (write enable). The SPM function table is shown in Figure 4-11. Signal DPA SP WRITE L is the W-input. There are two E-input signals: CONG ENAB SPR L for the low byte (bits 00–07), and CONG ENAB SPL L for the high byte (bits 08–15). This allows word or byte operations to be performed on the SPM.

4.3.9 Scratch Pad Memory Address Multiplexer

The SPAM generates the four address signals that select the desired SPM word. The SPAM consists of two Type 74153 Dual 4-Line-to-1 Line Data Multiplexers. The SPAM is shown in print CONB (E058 and E059). Each of the four 4-line-to-1-line multiplexers (two per 74153 package) has a common strobe input signal (CONH RUN GND L) and common address input signals (CONG SPA MUX 00 H and CONG SPA MUX 01 H). Four data input sources are used and they are connected so that when the SPAM is addressed and strobed, it generates one 4-bit output, selected from one of the four sources. Table 4-5 lists the sources of the SPAM input data that are a function of the state of the processor.

Table 4-5
SPAM Input Data Sources

Function	SPAM Input	Source	Source Print
Source Operand Register Selection	B	Instruction Register Bits 06–08	DPF
Destination Operand Register Selection	C	Instruction Register Bits 00–02	DPF
General Purpose Register Selection From Console	A	Bus Address Register Bits 00–03	CONA
Register Selection By Microprogram	D	Control Store ROM Bits 12, 18, 21, 22	CONG

The SPAM address inputs are S1 (signal CONG SPA MUX 01 H) and S0 (signal CONG SPA MUX 00 H). They are generated by CS ROM A13A2 (E115).

The data input selected is a function of the states of S1 and S0 as shown below.

Address Inputs		
S1	S0	Output
L	L	A
L	H	B
H	L	C
H	H	D

4.3.10 Processor Status Word Register

The processor status word register (PSW) contains information on the current priority of the processor, the result of the previous operation, and indicates a processor trap during debugging. The PSW bit assignments and use are shown in Table 4-6.

Table 4-6
Processor Status Word Bit Assignments

Bit	Name	Use
07–05	Priority	Set the processor priority.
04	Trace	When set, the processor traps the trace trap vector. Used for program debugging.
03	N	Set when the result of the last data manipulation is negative.
02	Z	Set when the result of the last data manipulation is zero.
01	V	Set when the result of the last data manipulation produces an overflow.
00	C	Set when the result of the last data manipulation produces a carry from the most significant bit.

The PSW is loaded as a result of instruction execution, program traps, I/O interrupts, and returns to main-line code. In the case of a program trap, interrupt, or return, the PSW is loaded with the second word of the vector from the Unibus data lines via the AMUX. Otherwise, the PSW is loaded through a network of multiplexers and combinational logic that is controlled by the particular instruction being executed.

The PSW is an 8-bit flip-flop register (print DPE). The condition code bits (N, Z, V, and C) are stored in 74175 and 7474 D-type flip-flops (E056 and E052). The priority bits and T-bit are stored in a 74175 D-type flip-flop called PSW 7:4 (E055). The output of the T-bit flip-flop is sent to another flip-flop (T DEL) which is used as the trap flag.

The input source for the condition code bits is the output of the condition code multiplexer (CC MUX). The CC MUX (E062 print DPE) is a Type 74157 Quad 2-Line-to-1-Line Multiplexer. One of the two 4-bit inputs is selected by the state of the (S) input. When S is high, the B-input is selected to the D-inputs of the condition code flip-flops (NEG, ZERO, VBIT, and COUT). The B-input consists of AMUX outputs DPA AMUX 00 H–03 H. When S is low,

the A-input is selected. The A-input consists of signals from the ROT CC MUX (E028 print DPE) and the C and V BIT ROM (E069 print DPF). These devices are part of the logic used in setting the condition codes as a function of instruction execution and are described in detail in subsequent paragraphs.

The input source for the priority bits (PSW 05–07) consists of AMUX outputs DPB AMUX 05 H–07 H which are sent to D-inputs D1, D2, and D3 of E055. Signal DPB AMUX 04 H is sent to D-input D0 of E055 as the source of the T-bit.

The PSW is loaded when the flip-flops are clocked. Each bit is clocked by the processor clock signal CONJ PROC CLOCK H which is free running as long as the clock is not inhibited. Clock control is provided by gating other signals with CONJ PROC CLOCK H. These signals and the PSW bits that they control are shown below.

Control Signal	PSW Bit
CONG LOAD PSW L	N, Z, V, C, T and Priority
DPF AUX DEL (1) L	N, Z, and V
DPF C CLK DEL (1) L	C

The logic that determines the condition code bits (C, V, N, and Z) and loads them in the PSW register is shown in prints DPE and DPF.

This discussion covers the determination and loading of the condition code bits as a result of instruction execution. Two categories of instructions are discussed: normal arithmetic instructions, and rotate and shift instructions.

Before discussing specific examples in these categories, the functional units of the logic are described briefly.

CC MUX

As mentioned previously, the condition code bits are loaded from the outputs of the CC MUX (E062, print DPE). The CC MUX is a Type 74157 Quad 2-Line-to-1-Line Multiplexer. Only the 4-bit A-input is used during instruction execution. The A-input is selected when the E-input and the S-input are both low.

The S-input selects the input (A or B). It is controlled CONF AUX CONTROL L bit. Signal CONF AUX CONTROL L is the AUX field (bit 24) of the control store word. It is generated by CS ROM E103 and 74175 E13 (print CONF) and enables the auxiliary ALU control when it is low. This is the desired condition to select input A of the CC MUX.

C and V BIT ROM

The C and V BIT ROM (E069, print DPF) calculates the values of the C-bit and V-bit for normal arithmetic instructions. The DPF SET COUT H output is modified by 3-input NAND gate E061 only during the execution of a subtract instruction. For shift and rotate instructions, the C-bit is determined by the two NAND gates wire-ORed to the DPF SET COUT L output of the C and V BIT ROM. For these instructions, the V-bit is determined by the exclusive-OR gate and NAND gate connected to the DPF SET V L output of the C and V BIT ROM.

ROT CC MUX

The ROT CC MUX (E028, print DPE) determines the value of the N-bit and Z-bit. It is a Type 74153 Dual 4-Line-to-1-Line Multiplexer. The outputs are DPE NEG H (for the N-bit) and DPE SET Z H (for the Z-bit). The inputs of both sections of the ROT CC MUX are a function of the category of instruction being executed.

Instruction Category	Input Designation
Rotate Byte	BR
Byte (not Rotate)	BR
Rotate (not Byte)	BR
Not Byte or Rotate	BR

The enabling or strobe (STB1 and STB2) inputs for the ROT CC MUX are both connected to ground which enables the multiplexer. Inputs S1 and S2 are the address inputs and are common to both sections. The data inputs are selected by the states of S1 and S2 according to the following truth table.

Address Inputs S1	S0	Selected Input
L	L	BR
L	H	BR
H	L	BR
H	H	BR

Input S1 is connected to DPG BYTE H, which is an inverted output of IR decoding ROM E078. Input S0 is connected to DPF ROTATE H, which is an inverted output of SOP AUX CTL ROM E066. This signal is a function of the instruction register output.

The data signal inputs to the ROT CC MUX are shown below.

ROT CC MUX Input Signals For DPE NEG H Output Section			ROT CC MUX Input Signals For DPE SET Z H Output Section		
Pin	Desig	Signal	Pin	Desig	Signal
13	BR	} DPE ROT NEG H	04	$B\overline{R}$	DPA 0–7 = 0H
11	$\overline{BR}$		06	$\overline{BR}$	DPE 0–15 = 0H
12	$B\overline{R}$	DPB AMUX 07H	03	BR	Output of inverter E058 pin 10 (zero detector for BLEG bits 01–06
10	$\overline{BR}$	DPD AMUX 15H			
			05	$\overline{BR}$	Output of gate E017 pin 13 (zero detector for BLEG bits 01–14)

ROT MUX 1 and 2

The ROT MUX 1 and 2 (E016 and E022 print DPE) generates outputs which are used in computing the C-bit and N-bit for arithmetic shift instructions ASL and ASR, and rotate instructions ROL and ROR. It is a Type 74153 Dual 4-Line-to-1-Line Multiplexer. Output DPE ROT COUT H (E048) is sent to exclusive OR E068 and positive NAND gate E070 (print DPF). Output DPE ROT NEG H is sent to input BR (pin 11) and input BR (pin 13) of the ROT CC MUX (E028). The inputs of both sections of the ROT MUX 2 are a function of the specific type of instruction being executed. The instructions are listed below.

Instruction	Input
Arithmetic Shift Left	ASL
Arithmetic Shift Right	ASR
Rotate Left	ROL
Rotate Right	ROR

The enabling inputs (STB1 and STB0) are both connected to ground which enables the multiplexer. The data inputs are selected by the states of address inputs S1 and S0 according to the following truth table.

Address Inputs S1	Address Inputs S0	Selected Inputs
L	L	ROR
L	H	ASR
H	L	ROL
H	H	ASL

The address inputs are connected to DPA ALU S1 H (input S1) and DPA ALU S0 H (input S0) which are inverted and buffered control store bits 28 and 29 from CS ROM E114 (print CONF). The actual signals are CONF ALU S1 L and CONF ALU S0 L which are part of the ALU field that picks the function to be performed by the ALU. These signals can be used by ROT MUX 1 and ROT MUX 2 because the ALU and these multiplexers are never used simultaneously. The data signal inputs to the ROT MUX 2 are shown below.

ROT MUX 2 Input Signals For DPE SERIAL SHIFT H Output Section			ROT MUX 2 Input Signals For DPE ROT NEG H Output Section		
Pin	Desig	Signal	Pin	Desig	Signal
03	ASL	GND	13	ASL	DPE L SHIFT SIGN H
			12	ROL	
04	ROL	DPE COUT (1) H			
06	ROR		11	ASR	DPD BREG 15(1) H
05	ASR	DPD BREG 15(1) H	10	ROR	DPE COUT (1) H

The ROT MUX 1 (E016, print DPE) generates the enabling signal for the rotate and shift zero-detection logic and generates a serial input signal for the BREG. It is a Type 74153 Dual 4-Line-to-1-Line Multiplexer. Output F_0 (pin 07) is the enabling signal for the BLEG zero-detection gates E005 and E011. The inputs to the ROT MUX 1 are a function of ASL, ASR, ROL, and ROR instruction execution. This multiplexer uses the same enabling signals and truth table as the ROT MUX 2. The data signal inputs to the ROT MUX 1 are shown below.

ROT MUX 1 Input Signals For DPE SERIAL SHFT H Output Section			ROT MUX 1 Input Signals For Output Section That Enables Gate E029		
Pin	**Desig**	**Signal**	**Pin**	**Desig**	**Signal**
13	ASL	DPD BREG 15(1) H	03	ASL	DPA BLEG 00 L
12	ROL		04	ROL	Output of gate E017 pin 04 ($\overline{BR00}\cdot\overline{COUT}$)
10	ROR	DPA BREG 00(1) H			
11	ASR		05	ASR	DPD BLEG 15 L
			06	ROR	Output of gate E017 pin 01 ($\overline{BR15}\cdot\overline{COUT}$)

The following example covers the determination of the condition code bits for the negate (NEG) instruction. When the NEG instruction is executed, the disposition of the condition code bits is as follows:

C – cleared if the result is 0; set otherwise
V – set if the result is 100000; cleared otherwise
Z – set if the result is 0; cleared otherwise
N – set if the result is less than 0; cleared otherwise

The condition code bits depend on the result produced by the instruction. For this example, assume that the result is 001000_8.

The C-bit should be set because the result is not zero, and the V-bit should be cleared because the result is not 100000_8. The C and V BIT ROM calculates the C and V bits:DPF SET COUT H is high and DPF SET V H is low. Verification of these signals is accomplished by checking the outputs of the C and V BIT ROM back to their sources, which are the associated ROM maps.

The N-bit should be cleared because the result is greater than zero. The NEG instruction is not a byte and not a rotate instruction: Therefore, the BR input of the ROT CC MUX is selected, which is DPD AMUX 15 H for the top section of this dual multiplexer. This signal is low so ROT CC MUX output DPE NEG H is low. This signal is sent to the CC MUX to load a 0 into flip-flop E056 (N-bit is cleared).

The Z-bit should be cleared because the result is not zero.

The BR input of the ROT CC MUX is selected: DPE 0–15 = 0 H for the bottom section of the multiplexer. This signal is low because all bits of the result are not zero; therefore, output DPE SET Z H is low. This output is sent to the CC MUX to load a 0 into flip-flop E056 (Z-bit cleared).

The following example covers the determination of the condition bits for the arithmetic shift-right (ASR) instruction. This instruction requires the use of some additional PSW logic. When the ASR instruction is executed, the disposition of the condition code bits is as follows:

C – loaded from the low order bit (00)
V – loaded with the exclusive-OR of the N-bit and the C-bit at completion of the shift operation.
Z – set if the result is 0; cleared otherwise.
N – set if the result is less than 0; cleared otherwise.

The condition code bits depend on the result produced by the instruction. For this example, assume that the result is 000002_8.

For this instruction, input ASR is selected for both sections of ROT MUX 1 and ROT MUX 2. Input BR is selected for ROT CC MUX because the ASR instruction is not a byte but it is considered to be a rotate instruction.

First, consider the C-bit, which is loaded from the low order bit. The output of the top section of ROT MUX 1 is DPE ROT COUT H. In this example, the selected input is DPD BREG 00 (1) H, which is low because bit 00 of the result is low: this makes output DPE ROT COUT H low. For the ASR instruction, C and V BIT ROM E069 is disabled by holding its enabling signal high.

Determination of the C-bit is handled by NAND gates E070, E061 and exclusive-OR gate E068 (Figure 4-12). The E070 gate is wire-ORed with the DPF SET COUT L output of the C and V BIT ROM. For the ASR instruction there is a high at the output of E070 which is wire-ORed to E068.

When the exclusive-OR output is low, signal DPF SET COUT H is sent to the CC MUX to reset COUT flip-flop E056. Thus, the C-bit is loaded with a 0 from the low order bit (00).

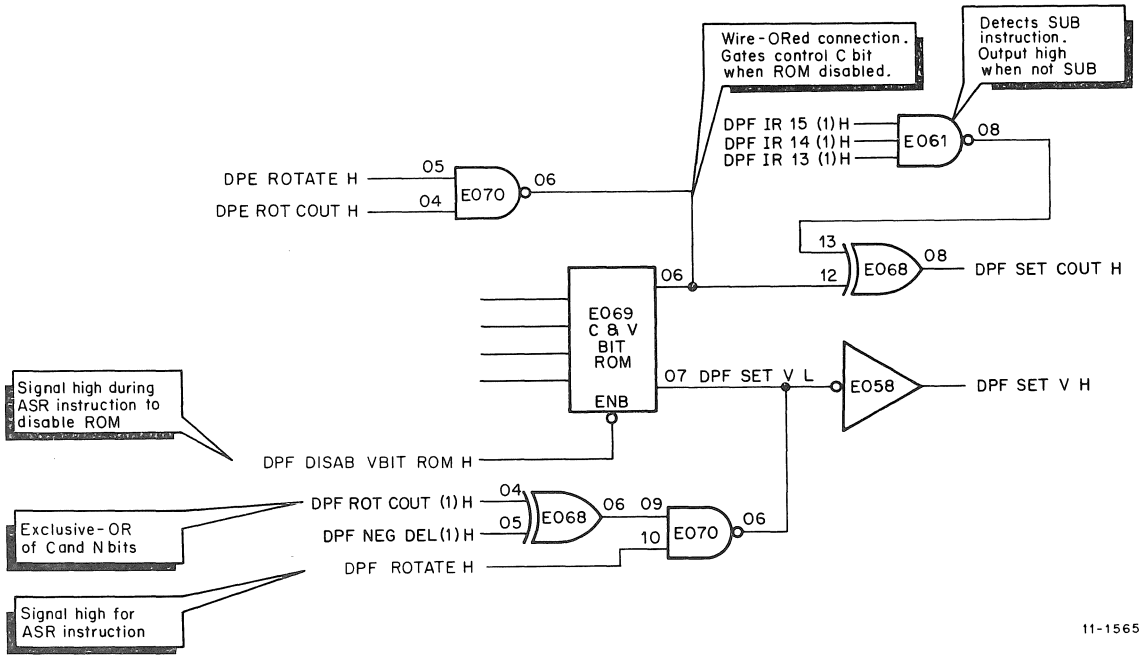


Figure 4-12 Logic For Determining C and V Bits
(Example Shown for ASR Instruction)

Next, consider the N-bit, which is cleared because the result is greater than 0 (bit 15 is a 0). Output f_1 of ROT CC MUX is DPE NEG H. In this example, the selected input (BR) is DPE ROT NEG H. This is output f_1 of ROT MUX 2, and in this case, the selected input (ASR) is DPD BREG 15 (1) H which is low because bit 15 of the result is low. Output f_1, which is DPE ROT NEG H, is low and as a result, DPE NEG H from the ROT CC MUX is low. This signal is sent to the CC MUX to reset flip-flop E056 (N-bit cleared). Signal DPF NEG H is also sent to the logic that determines the V-bit.

Next, consider the V-bit, which is the exclusive-OR of the N-bit and the C-bit. As mentioned, the C and V BIT ROM is disabled during the execution of the ASR instruction. Determination of the V-bit is handled by exclusive-OR gate E068. NAND gate E070 and inverter E058. The exclusive-OR gate performs the exclusive-OR function of the N and C bits and its output is connected to the DPF SET V L output of the C and V BIT ROM. This output is inverted by E058 to produce DPF SET V H. The inputs to exclusive-OR gate E068 are DPF ROT COUT H and DPF NEG H which are both low (refer to previous discussions of C-bit and N-bit). The output (pin 06) of the exclusive-OR gate is low and is sent to pin 09 of NAND gate E070. The other input of this gate is DPF ROTATE H and it is high during execution of the ASR instruction. The output of this gate is high and it is inverted by E058 to produce DPF SET V H which is low. This signal is sent to the CC MUX to reset flip-flop E056. Thus, the V-bit is loaded with the exclusive-OR of the C and N bits, which is zero.

Finally, consider the Z-bit, which is cleared because the result is not zero. Output f_0 of ROT CC MUX is DPE SET Z H. In this example, the selected input (BR) comes from gate E017 pin 13, which is an output of the rotate and shift zero-detection logic. Gate E017 produces a low because the enabling signal for this logic is not asserted. The enabling signal comes from output f_0 of ROT MUX 1 that selects DPD BLEG 15 L for an ASR instruction. In this case, bit 15 is low; therefore, DPE SET Z H is low. This signal is sent to the CC MUX to clear the Z-bit out of flip-flop E056.

4.3.11 Constants Generator

The constants generator consists of a single 32-word by 8-bit ROM attached to the A-leg of the ALU. It is identified as E044 CONSTANTS (part number 23-A01A1) and is shown on print DPB. The outputs of the constants generator are addresses of trap vectors and the complement of the address of the console switch register. Each output is an 8-bit word that is sent to the low order byte of the ALU A-leg. The eight bits are identified as DPB ALEG 00 L–07 L.

Four of the five inputs to the constants generator are CONB SPA 00 H–03 H which are generated by the SPAM (E058 and E059, print CONB). It is possible for both the constants generator and the SPM to use these signals because they are never used simultaneously. The fifth input is CONG SP WRITE H, which is also used by the SPM. It is an inverted output of control store ROM E106 (part number A17A2) in print CONG.

The enabling input for the constants generator is CONE ALLOW CONSTANTS L which is an output of the BUT DECODE multiplexer E091 (print CONE). This multiplexer is driven by control store ROM E113 (part number A18A2) in print CONG.

4.3.12 Console Switch Register

The settings of the 16 switches in the console Switch Register are transmitted in parallel via a cable to the Berg connector on the M7260 Data Paths Module. On the console (print 5409766-0-1, sheet 3), these signals are identified as SW00 (1) H–SW15 (1) H. On the M7260 module, these signals are identified as EXTRA SWITCH REG 00H–15 H.

4.3.13 Switch Register Multiplexers

Four 8266 2-input, 4-bit multiplexers are located between scratch pad memories and the arithmetic logic units (ALU). Each multiplexer is identified as SWITCH REG MUX. They are designated E033, E034, E035, and E036. Refer to drawing D-CS-M7260-0-1, sheets 4, 5, 6, and 7.

Figure 4-13 shows SWITCH REG MUX E036 for ALEG bits 00–03. The switch register signals from the console cable are sent directly to the A inputs of E036.

The outputs of SPM E040 are sent to the B inputs of SWITCH REG MUX E036. Input selection (A or B) is made by select lines S1 and S0. The multiplexer outputs are sent to the A inputs of ALU E027.

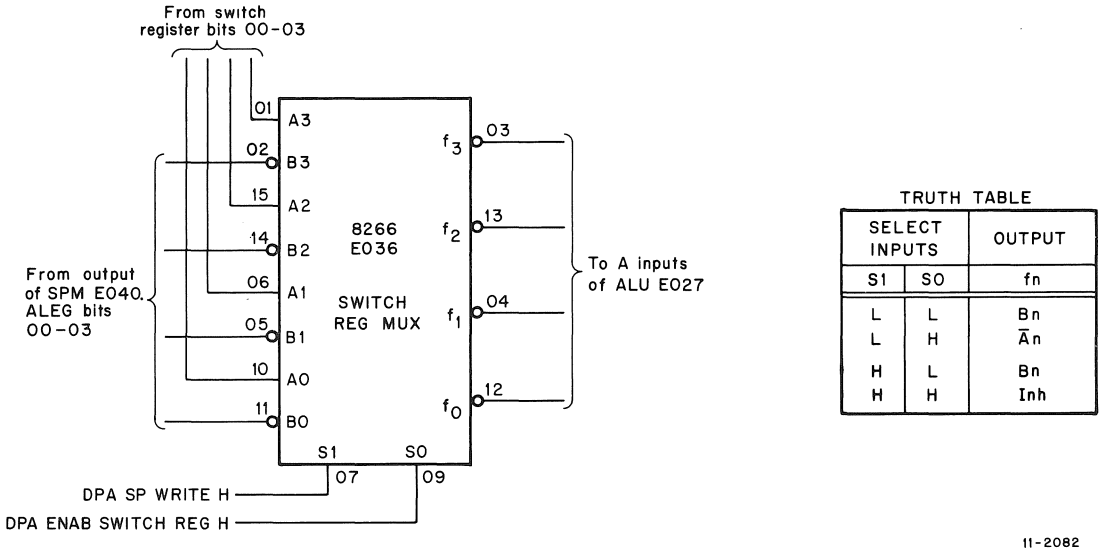


Figure 4-13 Typical Switch Register Multiplexer

4.3.14 Console Multiplexer

The console multiplexer scans the 16 bits in the BLEG, serializes the information, and transmits it via a cable to the console Buffer Register. It is a 74150 data/selector multiplexer and is identified as CONSOLE MUX (print DPE). Figure 4-14 is a block diagram of the console multiplexer and its input source.

The CONSOLE MUX (E006 print DPE) selects one of sixteen inputs in accordance with the states of the four data select lines (S0–S3). A low strobe signal enables the selected input to the output in the inverter state.

The high byte input (D8–D15) of the CONSOLE MUX comes from the output of the BLEG multiplexer. These signals are DPC BLEG 8–11 and DPD BLEG 12–15 L. The low byte input D0–D3) comes from the output of the BLEG multiplexer (DPA BLEG 00–03 L). The low byte inputs D4–D7 comes from the open-collector NAND gates (E036, print DPB) that are driven by the B register. The four data select inputs (S0–S3) are the inverted outputs of the console counter (E6 print 5409766-0-1, sheet 3). These signals are EXTA SCAN ADDRS 01 (1) H, -02 (1) H, -04 (1) H, and -08 (1) H. They are decoded as a BCD number.

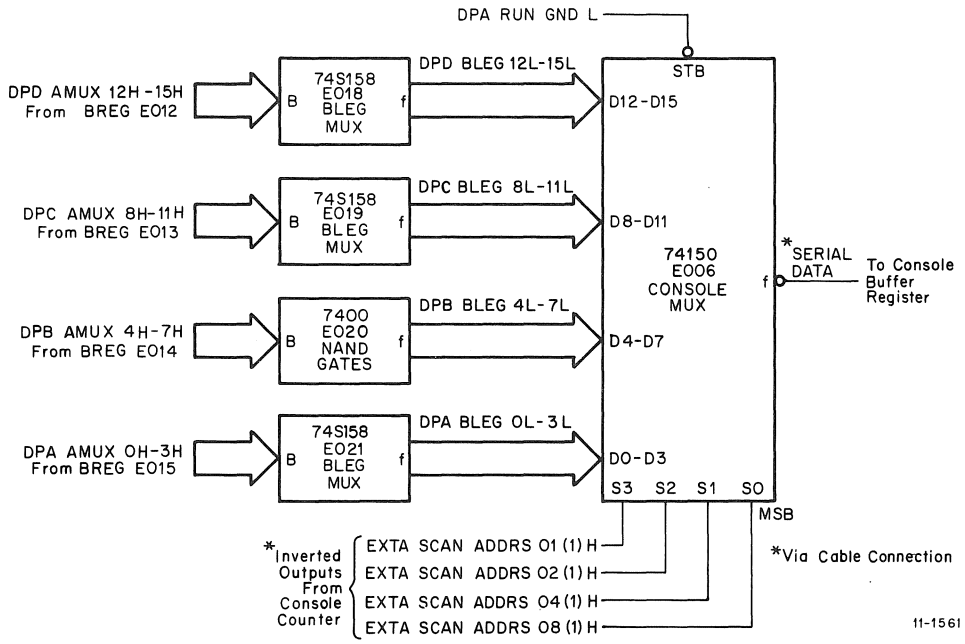


Figure 4-14 Console Multiplexer Block Diagram

4.4 INSTRUCTION DECODING

4.4.1 Introduction

Two methods are used to control instruction decoding. One uses microroutine selection and the other uses auxiliary ALU control. Dual control is required because of the large number of instructions that require source/destination calculations. Auxiliary ALU control is evoked whenever the microcode executes the action B← R10 OP B as a result of a specific instruction.

There are two prerequisites to a thorough understanding of the instruction decoding procedure. One is a knowledge of the microbranching process (Chapter 2) and the other is a knowledge of the PDP-11 instruction format (PDP-11/05S, 11/10S System Manual).

Certain facts concerning the PDP-11 instruction set are listed below.

a. In general, the PDP-11 operation code is variable from 4 to 16 bits.

b. Instructions are decoded from the most significant part of the word towards the least significant part of the word beginning with the most significant four bits.

c. There are a number of instructions that require two address calculations and a larger number that require only one address calculation. There are also a number of instructions that require address calculations, but do not operate on data.

d. All OP codes that are not implemented in the KD11-B processor must be trapped.

e. There are illegal combinations of instructions and address modes that must be trapped.

f. There exists a list of exceptions in the execution of instructions having to do with both the treatment of data and the setting of condition codes in the program status word.

4.4.2 Double Operand Instructions

Double operand instructions are decoded by ROM E059 (print DPG). Four inputs to E059 are DPF IR 12 (1) H–DPF IR 15 (1) H; these are outputs of the Instruction Register (E043, print DPF) which represent the OP code of a double operand instruction. The fifth input is CONE BUT DESTINATION L which is an output of the E091 BUT DECODE demultiplexer. The inputs to E091 are CONG BUT 00 L – CONG BUT 03 L which are the four bits of the BUT field of the control store word. When a double operand instruction is decoded, E059 output signal DPG CAL SOURCE L is asserted. This signal is ANDed with CONE BUT IR DECODE L at gate E085 to produce signal CONF MPC 07 L at pin 11 of quad NAND gate E071. The output of gate E079 is ANDed with DPF IR 09 (1) H, DPF IR 10 (1) H, and DPF IR 11 (1) H to produce CONF MPC 01 L, CONF MPC 02 L, and CONF MPC 03 L at the three remaining sections of quad NAND gate E071 (lower center section of print DPG). These four signals represent four bits of the 8-bit NXT field of the control store word and cause a microcode branch.

ROM E059 also generates DPG CMP + BIT L which indicates that the instruction does not modify the destination operand. Output signals DPG MOVE L and DPG BYTE L are used in the microbranch logic (print CONE). Table 4-7 explains the use of these signals.

Table 4-7
Effect of E066 Outputs DPG CMP+BIT L, DPG MOVE L, and DPG BYTE L

Instruction	E066 Output Signal	Effect	Remarks
CMP	DPG CMP+BIT L	Set condition codes	Destination is not modified; therefore, DATIP is not required.
BIT	DPG CMP+BIT L	Set condition codes	Destination is not modified; therefore, DATIP is not required.
MOVB	DPG MOV L DPG BYTE L		If the destination is a register, (i.e., destination mode 0) the result is sign extended; i.e., the sign of the low order byte is extended through the upper byte.
(ANY) BYTE	DPG BYTE L		Bit 0 of the address word must be used in determining which microroutine to use position source and destination data. See Chapter 9, for details.

For a binary operand instruction, the source operand is stored in R10 and the destination operand is temporarily stored in the B register. Then the control step B← R10 OP B is performed. The ALU can perform the operation A-leg minus B-leg, but not the converse. The CMP instruction requires the operation source minus destination, which is equivalent to A-leg minus B-leg; however, the SUB instruction requires the operation destination minus source. This is accomplished by storing the complement of the source in R10 for the SUB instruction only. The signal CONE BUT DESTINATION L is an input to E059. The microprogram issues CONE BUT DESTINATION L, whenever the SOURCE operand is stored in R10. If the current instruction is a SUB, E059 issues the signals DPG DIS ALU S BITS H, CONF ALU S0 L, and CONF ALU S2 L. This causes the complement of the BREG to be sorted in R10. When control step B← R10 OP B is performed for the subtract instruction, the ALU operation is A-leg plus B-leg plus 1, which is equivalent to destination minus source.

When the microprogram has completed the source calculation and retrieved the source operand for a binary operand instruction, it generates the signal CONE BUT DESTINATION L. This signal is ORed and inverted to produce CONE BUT DESTINATION H. The MOV, MOVB, CMP and BIT instructions are detected at the control steps listed below:

Bit Patterns	Instruction Class	Asserted Signals
(11) = (9) = (8) = 1 + (10) = 0	Unary Potential TST	DPG CAL DEST L + DPG 54 L
(10:08) = 0 + (11) = 0	Branch	DPG CAL BRANCH L
(15:08) = 0	Other	DPG ODD BYTE = OL

Two instructions in the other class require destination calculations: JMP and SWAB. These instructions are detected by ROM E083 shown in the lower left-hand corner of DPG. Standard unary instructions that affect or test the destination (with the exception of SWAB) are treated as binary instructions; i.e., the instruction is fetched, the operand is fetched, the operation is performed, and the operand is returned. The logic that decodes the operation for B← R10 OP B is shown on print DPF. For unary operand instructions, the destination operand is copied into both R10 and B.

4.4.3 Branch On Unary

There are three formats of instructions that require destination address calculations. The majority of the microcode destination routines are shared by all of the instructions that have destination fields. ROM E077, shown in upper right-hand corner of print DPG, is used to differentiate between the various instructions that use the microcode destination routines.

E077 is also used to detect illegal instruction combinations, which are defined as JMP or JSR and used with destination mode 0. The microcode flow chart shows that in microstep D0-2 a test is made for unary and illegal instructions by asserting the signal CONE BUT UNARY L. CONE BUT UNARY L produces the signal CONE ENAB UNARY L, which enables E077 (print DPG) to cause a microprogram branch. At other points in the microprogram such as D2-3, a test is made for a legal JSR or JMP instruction by the assertion of the signal CONE JMP + JSR L. The asserted signal CONE JMP + JSR L alters the input to E077 such that microroutines for legal JSR and JMP instructions are used. Signal CONE JMP + JSR L also causes the generation of the signal CONE ENAB UNARY L, which enables E077.

The effect of ROM E077 (part number 23-A10A1) is determined by observing its data pattern shown in drawing K-RL-M7260-0-8, sheet 9.

4.4.4 PDP-11 Branch Instruction

PDP-11 conditioned branch instructions are completely decoded by E072 shown on print DPG. E072 is enabled by the signal DPG CAL BRANCH L, which is asserted by E078 according to a previously discussed algorithm. IR (15) and IR (10:08) along with the condition codes N, Z, V, and C completely determine the branch instruction disposition. The offset of a branch instruction is sign-extended in microstep F-5 and shifted left one place in microstep B-1. All successful branch instructions are interpreted by the microroutine that begins in B-1, while all unsuccessful branch instructions are interpreted by the microroutine that begins in B2-1.

4.4.5 Operate Instructions

Operate instructions and instructions that set and clear condition codes are decoded by E083 and E065. NOPS, set condition code instructions, and clear condition code instructions all proceed from step F-5 to step CCM-1 in the microprogram. At step CCM-2, the microprogram performs a BUT DESTINATION to examine IR (4). Set condition

code instructions and the NOP-260 proceed with step SC-1 while clear condition code instructions and the NOP-240 proceed with step CC-1. Also in step CCM-1, the B register is loaded with the contents of the instruction ANDed with 17_8. This procedure zeroes all but the least significant four bits of the instruction copy contained in the B register. Remember that the instruction is loaded into both the IR and B register in step F-4. If the instruction is a SET COND CODE type, the operation is PSW ← B or PSW in step SC-1. Similarly, for clear condition code instructions, PSW ← B and not PSW is performed in step CC-1. Even though the entire PSW is reloaded, only the least significant four bits are effected by the sequence just described.

Other operate instructions such as WAIT, RTI, and HALT are decoded completely when BUT IR DECODE is issued during microstep F-5.

4.4.6 Auxiliary ALU Control

The auxiliary ALU control consists of the ROMs E053, E054, and E066 shown on print DPF. These ROMs determine the operation to be performed whenever the microcode executes the action B← R10 OP B. E053 decodes binary operand instructions while the other two ROMs decode unary operand instructions. Table 4-8 shows the auxiliary control outputs for binary and unary instructions.

4.5 PROCESSOR CLOCK

The KD11-B processor clock is shown in print CONJ. A single astable oscillator is used to generate a pulse train to which the entire processor is synchronized. Since it is a fully clocked processor, events that result in the alteration of storage registers occur only on a defined edge of the processor clock pulse.

The logic diagram for the processor clock is shown in print CONJ. A timing diagram is shown in Figure 4-15. NAND Schmitt trigger E039 is connected as an astable multivibrator (oscillator). It does not require a trigger and is free running as soon as it is gated on. The period of the oscillator pulse output should be set for 155 ns. Adjustable resistor R10 is used to set the period. The oscillator can be disabled by a low signal from NAND gate E13 pin 13. This low signal is asserted during the time that a Unibus transaction is in process. The processor clock is disabled during this time. The oscillator output is sent to the triggering input of one-shot E057. During maintenance, with the KM11 Maintenance Module installed, the CONJ MAN CLK L input to E039 pin 12 permits the processor clock to be single stepped. A switch on the maintenance module grounds the CONJ S CLK ON L to provide a positive transition at the output of E083 to trigger the one-shot (E57). This action provides a single processor clock pulse.

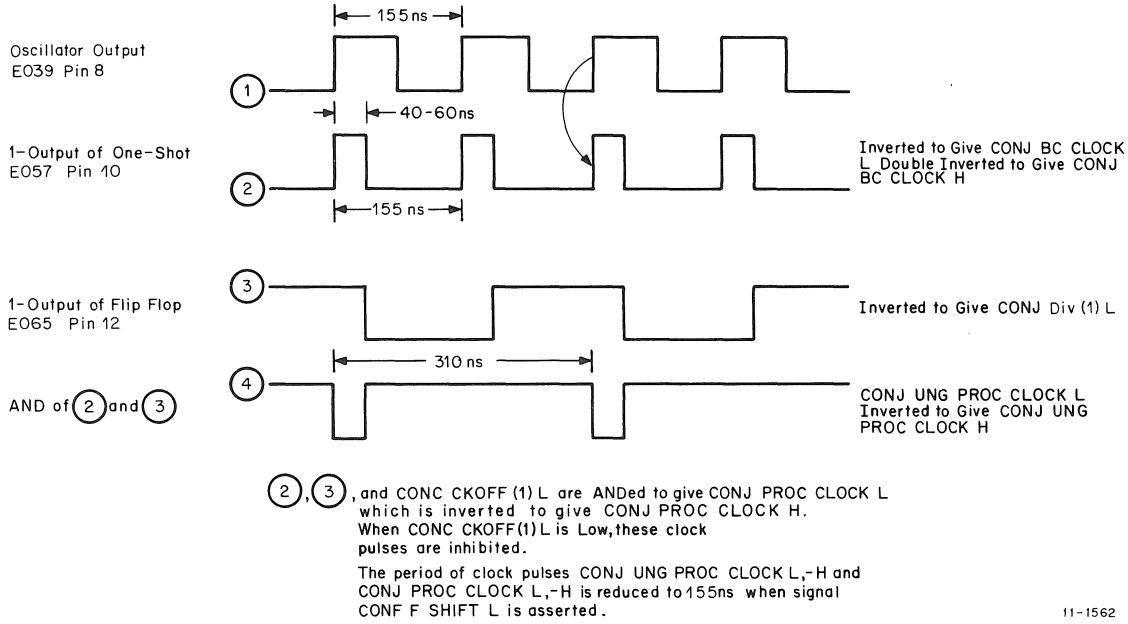


Figure 4-15 Processor Clock Timing Diagram

Table 4-8
Auxiliary Control for Binary and Unary Instructions

Inst.	Condition Codes			ALU Function	CIN	B
	N and Z	V	C			
MOV (B)	Load	Cleared	Not effected	A Logical	0	Load
CMP (B)	Load	Load like SUBTRACT	Load like SUBTRACT	A - B - 1	+1	Load
Bit (B)	Load	Cleared	Not effected	A + B	0	Load
BIC (B)	Load	Cleared	Not effected	~A + B	0	Load
BIS (B)	Load	Cleared	Not effected	AB	0	Load
ADD	Load	Set if OP's same sign and result different.	Set if carry out	A plus B	0	Load
SUB	Load	+ - (-) = - - (-) (+) = + } Set	Set if Carry	A plus B	+1	Load
CLR (B)	Load	Cleared (like ADD)	Clear	0	0	Load
COM (B)	Load	Cleared	Set	~B Logical	0	Load
INC	Load	Set if dst held 077777 before OP or 177(B)	Not effected		+1	Load
INC (B)	Load					
NEG (B)	Load	Set if result is 100000	Cleared if result is 0; set otherwise	A - B - 1	+1	Load
ADC (B)	Load	Set if dst was 077777 and C = 1.	Set if dst was 177777 and C = 1.	A Arithmetic	+C	Load
SBC (B)	Load	Set if dst was 100000.	Cleared if dst was 0 and C = 1; set otherwise	A - B	~C	
DEC	Load	Set if dst was 100000	Not effected			
DEC (B)	Load	Set if dst was 200(B)	Not effected			
TST (B)	Load	Cleared	Cleared	A Logical	0	Load
ROR (B)	Z ← (C:01) N ← C	N ⊕ C	(0)			Shift Right
ROL (B)	Z (14:C) N ← (14)	N ⊕ C	(15) B (7)			Shift Left
ASR (B)	Z ← (15:01) N ← N	N ⊕ C	C ← (15)			Shift Right
ASL (B)	Z ← (14:01 N ← (14)		C ← (15)			Shift Left

With pin 11 high, a positive transition at pin 12 triggers one-shot E057. A positive pulse is generated at output pin 10 and a negative pulse is generated at output pin 9. These pulses are 40–60 ns wide. The positive pulse from th one-shot is sent to high speed NAND buffer E10. The inverted pulse from the output (pin 06) of E10 is called CONJ BC CLOCK L. This signal is inverted and buffered by another E10 high speed NAND buffer whose output (pin 08) is called CONJ BC CLOCK H. These two clock signals have 155 ns periods and are buffered to provide increased fanout capability.

The negative pulse from the one-shot is sent to the clock (C) input of flip-flop E65. The 1-input of the flip-flop is fed back to its D-input. Normally, the other input of this gate (CONF F SHFT L) is high so the 1-output is inverted before being fed back to D-input. The clear input (pin 2) of the flip-flop is kept disabled by XOR SYNC L, which is connected to +5 V via E54. This is a toggle configuration and the flip-flop changes state on every positive transition of the clock pulse. The 1-output of the flip-flop represents a division by 2 of the oscillator output; i.e., a bipolar pulse train with a period of 310 ns. This signal is sent to high speed NAND Buffers E64 and E84. At gate E084, the flip-flop 1-output is ANDed with the positive output of the one-shot to generate a 40–60 ns negative pulse every 310 ns that is called CONJ UNG PROC CLOCK L. This signal is inverted by NOR gate E64 to produce CONJ UNG PROC CLOCK H. At gate E055, the flip-flop 1-output, the positive output of the one-shot, and signal CON CKOFF (1) L are ANDed to generate another 40–60 ns negative pulse every 310 ns. This signal is called CONJ PROC CLOCK L and is buffered to provide increased fanout capability. Signal CONJ PROC CLOCK L is inverted by NOR gate E64 to produce CONJ PROC CLOCK H. These clock signals are inhibited when CON CKOFF (1) L is low. This occurs when the processor is awaiting the completion of a Unibus interrupt or a RESET instruction. Signal CONC CKOFF (1) L is an output of CKOFF flip-flop E076 (print CONC) and is low when the flip-flop is set. This redefined flip-flop is set when its D-input (CONG CKOFF L) is low. This signal is the CK0 field of the control store word and is generated by CS ROM E116 (print CONG).

As previously mentioned, the 1-output of flip-flop E64 is sent to 1-input of NAND gate E84. The other input is CONF FSHFT L, which was previously identified as CONF SPARE L. It is generated by CS ROM E103 and is a field of the control store word. Normally, this signal is high and flip-flop E64 performs its divide-by-2 function to provide a 310 ns period for clock signals CONJ UNG PROC CLOCK L and H, and CONJ PROC CLOCK L and H. During a non-processor request (NPR) transaction, the NPR latency time is reduced by speeding up the B register shifting operations by decreasing the period of the CONJ PROC CLOCK L and H pulses from 310 ns to 155 ns. This is accomplished by asserting CONF FSHFT L at the input to E84. When this signal is low, the D-input to flip-flop E64 is always high so that the divide-by-2 function is not enabled. The periods of CONJ PROC CLOCK L and H are now the same as the period of the one-shot output which is 155 ns.

4.6 UNIBUS CONTROL

The Unibus control (BC) is found in prints CONC and CONC1, and the majority of Unibus drivers are found in print COND. The microprogram requests the BC to perform DATI, DATIP, DATO, and DATOB operations and to retrieve interrupt vectors. At the request of peripherals attached to the Unibus, the BC arbitrates BRs and NPRs. The BC is also responsible for detecting and causing a trap, whenever there is an attempt by the processor to address non-existent memory or to access odd addresses illegally.

The BC operates in parallel with the DP. The microprogram may request a DATI and then perform other tasks, such as incrementing R7, as long as the Bus Address Register is unchanged. The Unibus control proceeds with the DATI until the slave sync signal (SSYN) is returned from the slave device. At this point, the BC waits for the microprogram to set the CKOFF flip-flop shown on print CONC. This signal indicates that the microprogram is ready to accept Unibus data. If the microprogram sets CKOFF before SSYN is received, the BC inhibits the oscillator until SSYN is received or a Unibus timeout occurs.

4.6.1 DATI Timing

A DATI is used by the processor to retrieve data from devices attached to the Unibus. Figure 4-16 contains a timing diagram of the Unibus control signals for a DATI bus operation. Signals BBSY, C0, C1, and the address lines may be set by the processor or bus master, whenever it is determined that the Unibus is free for use. The Unibus is free for use by the processor when the following equation is true:

BUS FREE = (~BBSY) (~NPR) (~SACK) (~SSYN)

Once BBSY, C0, C1, and the address lines are asserted, the master device must wait at least 150 ns before issuing MSYN. During this time, the address and control lines of the Unibus are settling, so that when MSYN is issued, there will be no confusion regarding the device addressed or the direction of the data transfer. After MSYN is asserted, the BC must wait until SSYN returns from the Unibus and CKOFF is asserted. This indicates that data is available on the Unibus and the microprogram is ready to accept that data. Once the processor has strobed the data from the Unibus into a storage element, normally the B register, the signal MSYN is not asserted. BBSY, C1, C0, and the address are maintained for 155 ns after MSYN is not asserted.

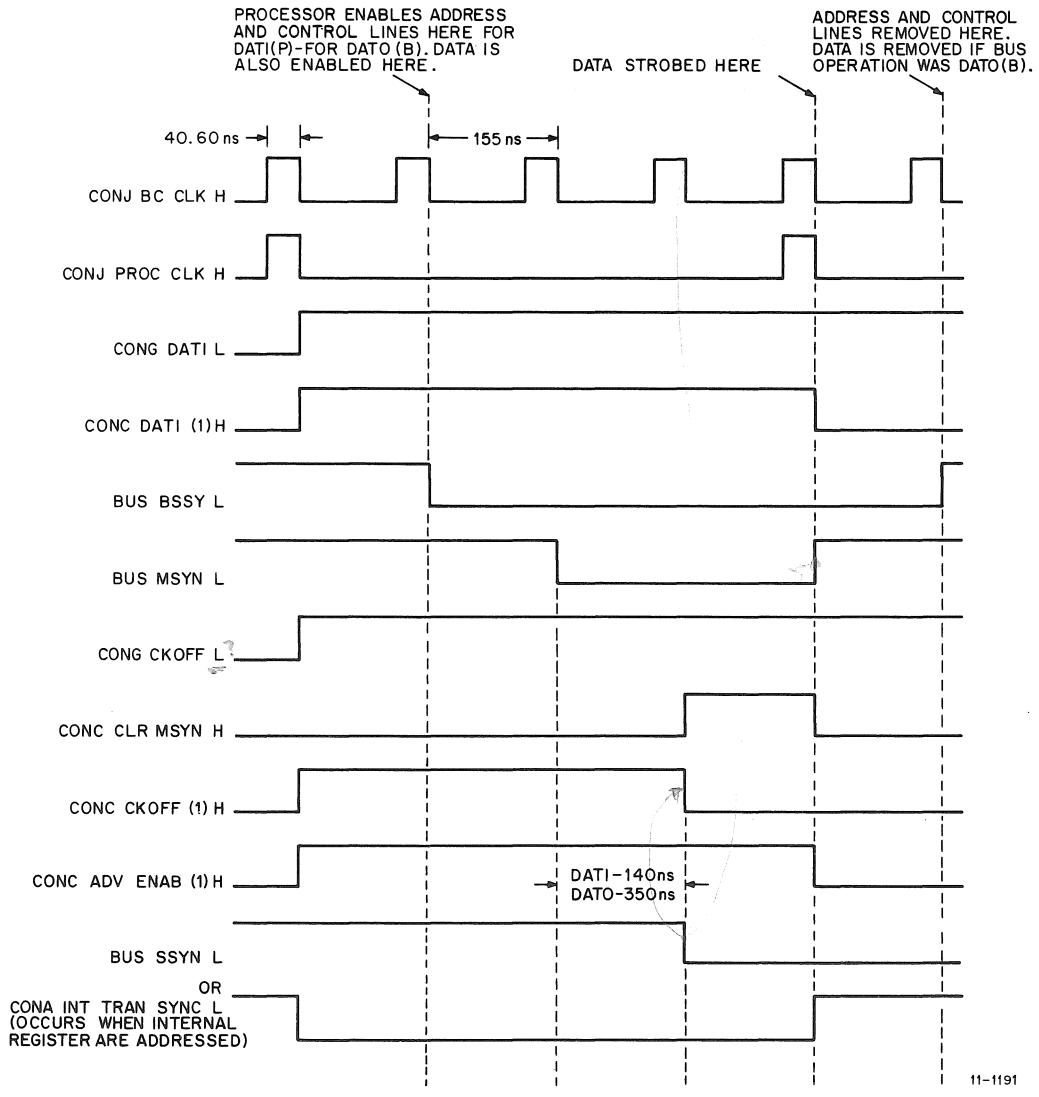


Figure 4-16 DATI and DATO Timing

4.6.2 DATI Operation

The microprogram requests a DATI by asserting the signal CONG DATI L, which is the input to E18 pin 8 on print CONC. On the next processor clock following the assertion of CONG DATI L, the flip-flop DATI E8 on CONC is set. If the Unibus is free, BBSY is set.

Simultaneous with the assertion of CONC BBSY (1) L, the bus address drivers (print COND) enable the contents of the Bus Address (BA) Register onto the Unibus address lines. The bus drivers for BUS A16 and BUS A17 are automatically enabled by the following equation:

BUS A16 and BUS A17 = (A15) (A14) (A13) (BBSY)

This allows PDP-11 processors, such as the KD11-B, that do not have extensive memory management facilities, to address peripheral registers that are located between 124K and 128K in the address space.

The MSYN flip-flop, E15 on print CONC, is normally set 160 ns after the issuance of BBSY. The setting of MSYN triggers a 9602 one-shot E29, shown at the lower left side of print CONC. This one-shot, which has a pulse width of 22 μs, is used to detect attempts at addressing non-existent memory by the processor. If SSYN does not appear on the bus before the signal CONC DAT TO (1) L is asserted by E28, the microprogram is forced to execute an error trap sequence.

SSYN is strobed into flip-flop E36 (print CONC) and generates the signal CONC SSYN (1) H. CONC SSYN (1) H enables an NOR gate (E35 shown in the center of print CONC). At this point, the following conditions exist:

a. BBSY, C0, C1, and MSYN are being applied to the Unibus by the KD11-B.

b. An address is enabled on the bus address lines by the processor.

c. Data is being driven onto the Unibus data lines by the addressed device or memory location.

d. SSYN is being generated by the addressed device.

The addressed peripheral device must maintain both its data and SSYN on the bus as long as MSYN is asserted. The Unibus control removes MSYN from the bus within 310 ns after SSYN and CKOFF are both set. The gating structure for removing MSYN can be traced back from the K-input to the MSYN flip-flop (E15 on print CONC).

If MSYN, CKOFF, and the oscillator divider flip-flop are all set, and the BC is waiting for SSYN, the oscillator input is inhibited and the oscillator stops. When SSYN is asserted, the input is released and MSYN is cleared. This method of synchronization causes no extra delay or flip-flop setup problem.

4.6.2.1 DATIP Operation – Note that the sequence for DATI and DATIP are almost identical. DATIP is used by the processor to prevent the modification of a memory location by a device other than the processor, while the processor is operating on that memory location. To further understand the need for DATIP, consider the operation of the DM11, a 16-line Asynchronous Serial Line Multiplexer (DEC-11-HOMA-D). The Buffer Active Register in the DM11 indicates status information and initiates message transmission. To begin the transmission of a message, the processor sets a 1 in the DM11 Buffer Active Register. When the message has been transmitted, the DM11 performs an NPR transfer to its own status register and clears the appropriate channel status bit.

Typically, the program to set an appropriate bit in the DM11 status register will use a BIS instruction. To execute this instruction, the processor must first execute a DATIP to the address of the DM11 status register and obtain a copy of the current contents of the status register. The specified bit is then set in the copy of the DM11 status register that is held by the processor. Finally, the processor performs a DATO to the status register and returns the altered copy of the status register to the DM11.

If, for instance, at the time of the DATIP, channels 0, 1, and 2 were active, the processor would retrieve a status word of 000007_8. Suppose the program desired to activate channel 4; the return status word would equal 000027_8. If channels 0, 1, or 2 completed their transmission between the time the processor issued the DATIP and the DATO and the processor permitted the DM11 to clear its status register before the DATO cycle of the BIS instruction, it is obvious that the copy of the DM11 status register held by the processor would be invalid.

Most core memories manufactured by Digital inhibit the normal restore cycle when a DATIP is issued. Therefore, when the following DATO is issued, the memory does not have to wait for the completion of the previous restore cycle before continuing with the DATO operation. However, the processor must inhibit NPRs from issuing a DATIP to the completion of the following DATO. Therefore DATIP operations lengthen the worst case NPR latency of the processor.

4.6.2.2 **DATIP Logic** – The BC executes a DATIP whenever the flip-flops DATI and DATIP (E8 and E16 on print CONC) are simultaneously set by the microprogram. The equation for setting DATIP, E8, is as follows:

(SET DATIP E46, pin 12) = (CONG ENAB IN PAUSE L) ← (DPG ENAB NON MOD H)

(CONJ DIV (1) L)

Signal number 1 is an indication that the microprogram anticipates the need for a DATIP. Signal number 2 confirms that the current instruction in the IR is one that requires the destination to be restored. The instructions TST, CMP, BIT, JMP, and JSR can never result in the modification of the destination by the processor. Therefore, it is not necessary to use the DATIP operation during the execution of these instructions. Signal number 3 ensures that DATIP is set on a processor clock rather than a BC clock. DATIP remains set following the transfer and inhibits the setting of NPG flip-flop E44. It is directly cleared when the processor enables the destination data during the next DATO, and NPRs are again allowed to be granted.

4.6.3 DATO

DATO differs from DATI in that for a DATO the Unibus data lines are driven by the processor. Figure 4-15 shows that data is maintained on the bus for the duration of BBSY. In the KD11-B, a DATO operation requires cooperation between the BC and the microprogram. The steps executed by the microprogram for a DATO operation are illustrated in flow chart example shown below. Note that CK OFF and DATO must be set simultaneously, and that the microprogram control step that follows the DATO specification must enable the data from the appropriate storage register through the ALU and AMUX.

	LOC	NXT	
DATO Starts	334	065	D1–5 DATO; ALBYT; CKOFF /GET TO D1–6 FROM D0–18 VIA GOTO
DATA Put on Unibus	065	305	D1–6 DRIVERS B; GOTO B2-2 (BUT SERVICE)

The microprogram initiates a DATO operation by setting the DATO flip-flop (E8 on print CONC). The 7400 gate, E47, generates the signal CONC DAT ENAB L, which enables the data drivers shown on prints DPA, DPB, DPC, and DPD, and also clears DATIP.

4.6.4 Byte Operations

Byte operations have the following significance to the KD11-B Unibus control (BC):

a. An odd address may be placed on the Unibus.
b. For a DATOB, both C0 and C1 are enabled.

Byte operations have the following significance on the Unibus slave:

a. No significance for DATIP operations.
b. For DATOB operations, only the upper or lower eight bits of the addressed location should be altered.

NOTE

The master must properly position the data during a DATOB operation. For instance, if the operation is a DATOB to the odd byte of a location, the data must appear on Unibus data lines (15:08).

In the processor, the ALLOW BYTE flip-flop (E16 on print CONC0 permits odd addresses and generates the appropriate C0 and C1 signals. The microprogram attempts to set the ALLOW BYTE flip-flop, whenever the possibility of a legal odd address or DATOB is anticipated, by asserting the signal, CONG ALLOW BYTE L. The signal DPG BYTE L (shown as an input to E46 pins 3 and 4 on CONC) confirms that the current instruction (IR) is a byte operation.

4.6.5 Bus Errors

The following situations cause the bus error trap sequence to be executed:

a. An attempt to illegally address an odd location in the memory space. For instance if the contents of R7 is odd at the beginning of an instruction fetch, a bus error trap will be executed because instructions must start at even addresses.

b. An attempt to access non-existent locations in the memory space. A non-existent location is recognized when SSYN does not appear on the bus within 25 μs of the setting of MSYN by the processor.

Either type of bus error causes the BE flip-flop, E7 on print CONC, to be set. The BE flip-flop inhibits the signal CONC MSYN OUT H which removes MSYN from the Unibus whenever a bus error is detected. The signal CONC BUS ERROR (1) H causes the 256 X 4 ROMs (E102 and E101 on print CONF) that generate the next address for the microprogram to be disabled. This forces the microprogram to execute its next control step from microaddress 010_8.

A double bus error is defined by two successive unsuccessful attempts at addressing the memory. On the second successive bus error, the microprogram is forced to location 110_8 by the simultaneous setting of the BE and DBE flip-flops (E7 on print CONC). The microprogram in the KD11-B is designed to cause a processor halt after two successive bus errors.

4.7 INTERNAL UNIBUS ADDRESSES

All presently implemented PDP-11 processors, including the KD11-B, contain internal registers that have associated addresses in the Unibus address space. To the program executed by the processor, the internal registers are indistinguishable from peripheral or memory registers. However, access to the internal registers is not available to devices attached to the Unibus other than the processor.

In the KD11-B, the concept of internal registers has been expanded to include the serial communications line control and the line clock. Table 4-9 lists the internal Unibus addresses.

Attempts to address internal Unibus addresses are detected by the logic detailed on print CONA and illustrated in Figure 4-17. A characteristic of all addresses listed in Table 4-9 is that the odd byte of the address is equal to 377_8. The signal CONA INT BUS ADDRS L, generated by E60, indicates that the odd byte of the currently addressed register is 377_8 and that the bus address may be that of an internal register.

Table 4-9
Unibus Addresses

Octal Address	Function
177700 177701 . . 177707	General registers R0 through R7
177710 177717	Hidden registers used by the microprogram
177776	Program status register
177570	Console switch register
177571	Odd byte of console switch register
177560	Receiver or keyboard status register
177562	Receiver or keyboard buffer
177564	Transmitter or printer status register
177566	Transmitter or printer buffer
177546	Line clock status register

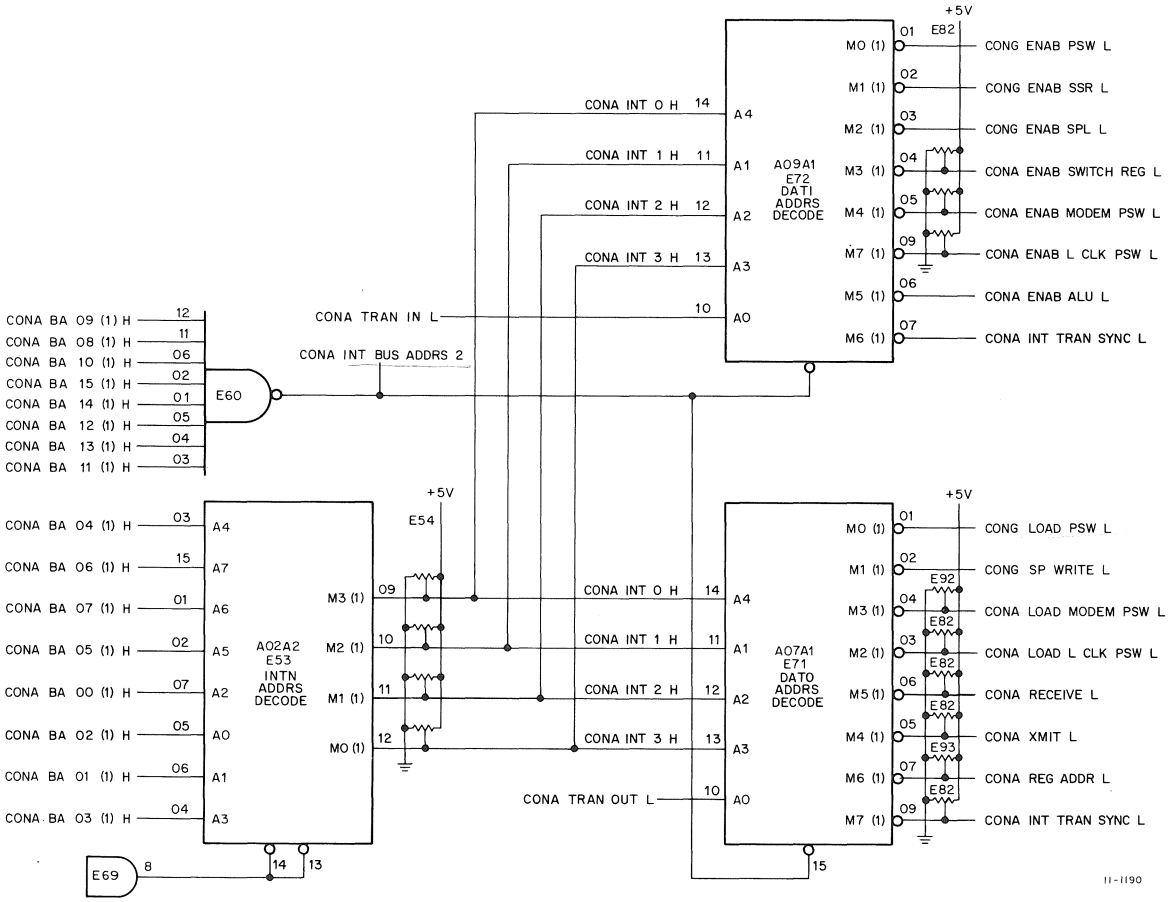


Figure 4-17 Unibus Address Decoding

The read-only memories of ICs E53, E72, and E71 decode the least significant eight bits of the Unibus address to determine which, if any, of the internal registers are currently being accessed.

The timing diagram contained in Figure 4-18 shows that the signal CONA INT TRAN SYNC L replaces SSYN for internal registers. Note that bus addresses, C1, C0, and MSYN are driven onto the Unibus during attempts to address internal registers. However, the signals generated by ROMs E71 and E72 reconfigure the data path (DP) such that during a DATI from 177776, for example, the PSW is enabled onto the DP.

During transfers to and from the processor, to registers and to memory, data to or from the BREG is normally inhibited. The reason is that most of the elements contained on the A-leg may be addressed with their corresponding Unibus address. Therefore, almost any data transfer may be from or to the DP. Since it is not possible to both read and write into the DP on the same clock pulse, it is necessary for the microprogram to receive and transmit Unibus data from the BREG.

In order to understand the decoding sequence for ROMs E53, E71, and E72, it is necessary to refer to the ROM maps.

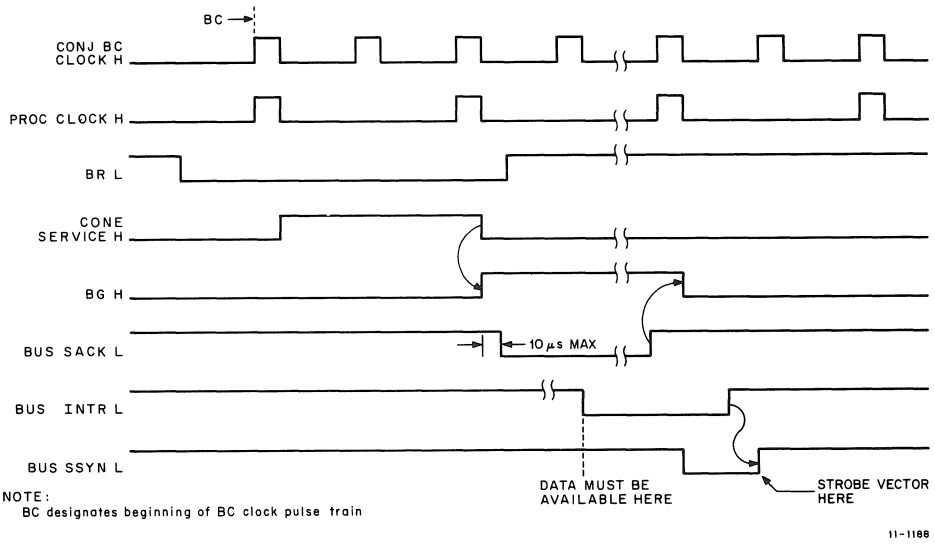


Figure 4-18 Bus Request (BR) Timing

4.8 BUS REQUESTS

The KD11-B responds to bus requests (BRs) in a manner similar to that of the other PDP-11 processors. Peripherals may request the use of the Unibus in order to make data transfers or to interrupt the current processor program by asserting a signal on one of four BR lines, numbered 4, 5, 6, and 7 in order of increasing priority. For example, if two devices, one at priority 5 and the other at priority 7, assert BRs simultaneously, the device at priority 7 is serviced first. Furthermore, if the processor priority, determined by (07:05) of the PSW, is at level 4, only devices that request BRs at levels higher than 4, such as BR 7, BR 6, or BR 5, are serviced. Table 4-10 contains the order of priorities for all BRs and other traps.

Table 4-10
Trap Priorities

Priority	Service Priorities
Highest	1. T-bit trap
	2. Stack overflow
	3. Power fail
	4. BR7
	5. BR6
	6. Internal line clock
	7. BR5
	8. BR4
	9. UART receive
	10. UART transmit
	11. Console stop
Lowest	12. Next instruction fetch

Since a BR can cause a program interrupt, it may be serviced only after the completion of the current instruction in the IR. A device that requests a program interrupt must at the appropriate time place a vector address on the Unibus data lines. The processor first stacks away the current contents of PSW and R7; then a new R7 is loaded from the contents of the vector address, and a new PSW is loaded from the contents of the vector address plus two. An example of the flow that handles a BR is as follows:

```
LOC     NXT     *BUS GRANT SERVICE
                /GET TO BG-1 FROM BUT SERVICE

040     305     BG-1 BUT INTERRUPT; GO TO B2-2 (BUT SERVICE
                              ↓          /IF INTERRUPT GO TO INT-1
                                         /IF NO INTERRUPT FALL THROUGH TO B2-2

LOC     NXT     *INTERRUPT SERVICING
                /GET TO INT-1 FROM BG-2 VIA BUT INT (TRUE)

325     246     INT-1 R(12) ← UNIBUS DATA; SET SLAVE SYNC; GO TO ET-3
                              ↓
LOC     NXT
246     247     ET-3 B, BA ← R(6) - 2; ENAB OVER
247     226     ET-5 R(6) ← B; CK OFF; DATO
226     251     ET-6 DRIVERS ← PS
251     252     ET-7 B, BA ← R(6) - 2; ENAB OVER
252     253     ET-8 R(6) ← B; CK OFF; DATO
253     254     ET-9 DRIVERS ← PC
254     255     ET-10 BA ← R(12); DATI; CK OFF
255     256     ET-11 PC ← UNIBUS DATA
256     257     ET-12 BA ← R(12) + 2; DATI, CK OFF
257     305     ET-13 PS ← UNIBUS DATA; GO TO B2-2 (SERVICE)
```

The microprogram indicates the end of instruction execution by asserting the signal CONE BUT SERVICE L. BRs are arbitrated by the ROM E24 (shown on print CONC1). If there is an impending BR, the signal CONC BR GRANT H is asserted by E34 pin 8 (print CONC1). When CONE BUT SERVICE L is issued, the appropriate BG is clocked into the storage register (E026). Simultaneously, the microprogram address is forced to the bus grant sequence by the logic shown on print CONE.

In the KD11-B, interrupts for the SCL and the line clock are not entered the same way as interrupts from other devices attached to the Unibus. Interrupts from the SCL and line clock are handled in the same manner as power fail and stack overflow traps. For all of these events, the microprogram address is altered when CONC BUT SERVICE L is issued to force the microprogram into the appropriate routine, which simulates the appropriate interrupt or trap.

The appropriate vector address for SCL and line clock interrupts are generated by the constants generator, which is the E44 ROM shown on print DPB.

4.9 NON-PROCESSOR REQUESTS (NPR)

NPRs are a facility of the Unibus that permit devices on the Unibus to communicate with each other with minimal participation of the processor. The processor's function in servicing an NPR is simply to give up control of the bus in a manner that does not disturb the execution of an instruction by the processor. For example, the processor may not relinquish the bus following a DATIP.

An NPR is received through a bus receiver (print COND) and gated into flip-flop E13 (print CONC1). If conditions are appropriate to permit an NPG to be issued by the KD11-B, the signal CONC SET NPG H is issued by E45 pin 6. CONC SET NPG H is generated according to the following equation:

$$\text{CONC SET NPG H} = (\leftarrow \text{DATIP}) \cdot (\leftarrow \text{SACK DELAYED}) \cdot \text{RUN}$$

The signal CONC SET NPG H causes flip-flop E44 to be set, which in turn causes NPG to be placed on the Unibus. Note that both NPGs and BGs will be issued by the KD11-B for a period of 10 μs. If the requesting device does not respond with SACK within this period, the 9602 timer IC (shown in the upper right-hand corner of CONC1) trips, causing flip-flop E28 to be set. This in turn causes the pending BG or NPG to be cancelled, and the processor to continue operation.

4.10 SERIAL COMMUNICATIONS LINE DESCRIPTION (SCL)

The SCL of the KD11-B is essentially program compatible with the KL11 teletype control. The heart of the serial communications line logic (prints DPH and DPH1) is the Universal Asynchronous Receiver Transmitter (UART), an MOS-LSI IC. The UART is easily recognized on the M7260 module because it is the only 40-pin dual in-line package used in the KD11-B. The UART is the only IC in the processor that requires two supply voltages, +5 V and −12 V. The −12 V supply (print DPH) for the UART is generated by placing four diodes in series with the −15 V supplied by the power supply.

The additional circuitry other than the UART (prints DPH and DPH1) serves the following purposes:

a. Generation of the reader RUN signal that is used to control the low speed paper-tape reader found on Model ASR 33 Teletypes.

b. Generation of status bits and interrupts to make the KD11-B SCL program compatible with the KL11.

c. Generation of the 20-mA current loop necessary to operate Model ASR 33 Teletypes, VT05s, and LA30s.

An important feature of the KD11-B SCL is double buffering. An understanding of double buffering (Figure 4-19) may be gained by studying the programming example provided. In order to receive or transmit data at the maximum rate, it is only necessary to empty or fill the appropriate UART buffer once every character time. Conversely, on single-buffered devices such as the KL11, it is necessary to empty or fill the appropriate buffer in one bit time.

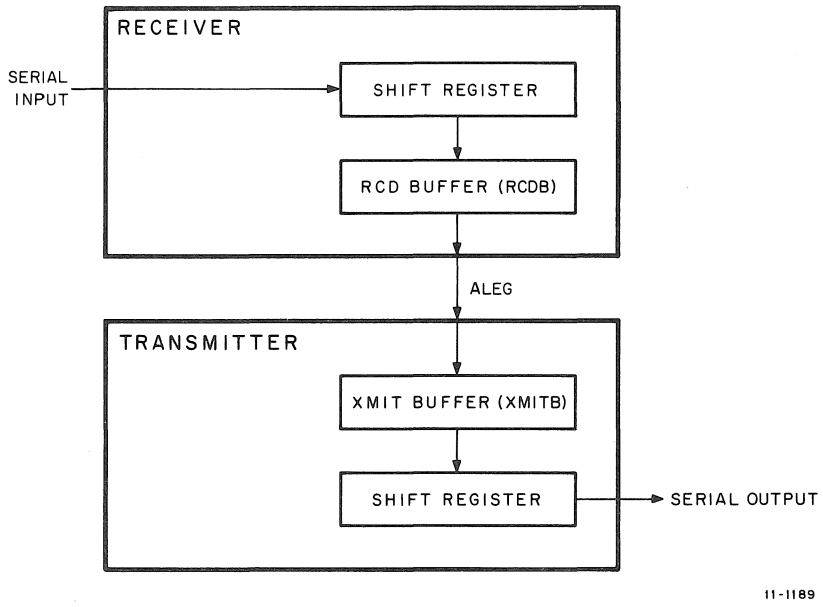


Figure 4-19 Double-Buffering Data Flow

The following programs are sample programs which utilize the UART. The first program simply echoes a character received from the UART into the transmitter of the UART. The second program illustrates the proper use of the RESET instruction, following an instruction that caused the SCL to transmit a character. RESET should not be issued until the last desired character has cleared the UART transmitter shift register.

UART Sample Programs

```
LOOP:   TSTB    RCDSTA      ; Test for a received character
        BPL                 ; Go to Loop if no character
        TSTB    XMITST      ; Test the XMIT condition
        BPL
        MOVB    RCDB, XMIT  ; echo character
                            ; If at this point it is desirable to issue a
                            ; RESET it is necessary to send a null
                            ; character to ensure that the desired
                            ; character has been completely transmitted
        TSTB    XMITST
        BPL
        MOVB    NULL, XMIT
        TSTB    XMITST      ; When null character is
        BPL                 ; clear of the
        RESET               ; XMITB
```

On print DPH, the transmitter DONE flag is seen only as an indication that the transmitter buffer is empty (TBMT). The TBMT flag will set at least one character time before the UART has finished transmitting the last character received. A RESET instruction that occurs while a character is in the process of being transmitted aborts that character transfer. Therefore, the only safe way to issue RESET instructions, following an instruction that has transmitted a character through the UART, is to transmit a null character prior to issuing the RESET instruction. Some care must be used in selecting the null character since it may be garbled by the RESET instruction. When the null character clears the UART transmitter buffer, it is safe to issue the RESET instruction. When the SCL maintenance mode is enabled by setting the transmitter status bit (2), the serial output is fed back into the serial input just as in a standard KL11. The transmitter status register address is 177564_8. The SCL always appears to the program as the last device at the BR4 interrupt level.

Two type 9602 retriggerable one-shots are interconnected to form a simple oscillator that generates the SCL clock signal (Figure 4-20). The oscillator starts when signal DPH B INIT L is released (goes high) and is self-sustaining thereafter. It has a gated input that allows initialize signal DPH B INIT L to inhibit the clock when it is asserted.

Both 9602s are in the same package (E100) and are identified as A and B in this discussion. To ensure stability, the +5 V power to these one-shots is filtered by R22, R27, C111, and C112. The clear inputs (pins 03 and 13) are not used so they are permanently pulled up to +5 V.

Each one-shot triggers on a negative (falling) edge at input pin 05/11 while input 04/12 is held permanently low (connected to ground).

The output pulse width is determined by the RC network connected across pins 14/2 and 15/1. For one-shot B (pins 2 and 1), the values are chosen to provide a pulse width of 10 μs. For one-shot A (pins 14 and 15), the pulse width is variable from about 22 to 44 μs. Potentiometer R21 is used to perform the adjustment.

As power is applied and DPH B INIT L is asserted, both one-shots are at rest; therefore, the 0 output (pin 09) of one-shot A is high. This signal is sent to the triggering input of one-shot B and is also fed back to pin 13 of NAND gate E097. When DPH B INIT L is released (goes high), the output of E099 goes low and triggers one-shot A. A negative pulse is generated at the 0 output of one-shot A. Assume that R21 is adjusted to give a pulse width of 26 μs. The negative-going leading edge of this pulse triggers one-shot B, which produces a 10 μs positive pulse at its 1 output (pin 06). When one-shot A times out, its output goes high and, because of the feedback connection, it is triggered again. There is a delay of approximately 25 ns before it triggers again and subsequently causes one-shot B to trigger again. The pulse width of one-shot A determines the period of clock signal DPH TTY CLK (1) H. In this case, 10 μs clock pulses are produced at a period of 26 μs, which is a frequency of 38.4 kHz.

A simplified timing diagram is shown in Figure 4-20.

Signal DPH TTY CLK (1) H is used to create the clock for the universal asynchronous receiver/transmitter (UART). This signal is first sent to a counter whose outputs are selected by a switch to provide five different UART clock rates. The UART receiver and transmitter clocks must operate at a rate 16 times the value of the desired baud rate because the UART samples incoming data 16 times per second. The frequency in Hz of a particular UART clock is found by multiplying the desired baud rate by 16.

Signal DPH TTY CLK (1) H from the oscillator is sent to the clock 1 (C1) input of 4-bit binary counter E095 (Figure 6). It is a type 74197 that is connected as a 4-bit ripple-through counter. The clock input (C1) is divided by 2, 4, 8, and 16 at outputs R0(1), R1(1), R2(1) and R3(1), respectively. Load inputs D0, D1, D2, and D3 are all connected to ground so the counter cannot be preset. The counter is cleared (all outputs go to 0) when DPH B INIT L is asserted.

TRUTH TABLE FOR
9602 ONE-SHOT

INPUT PIN NO.		OUTPUT PIN NO.	
05/11	04/12	06/10	07/09
H	↑	⎍	⎎
↓	L	⎍	⎎

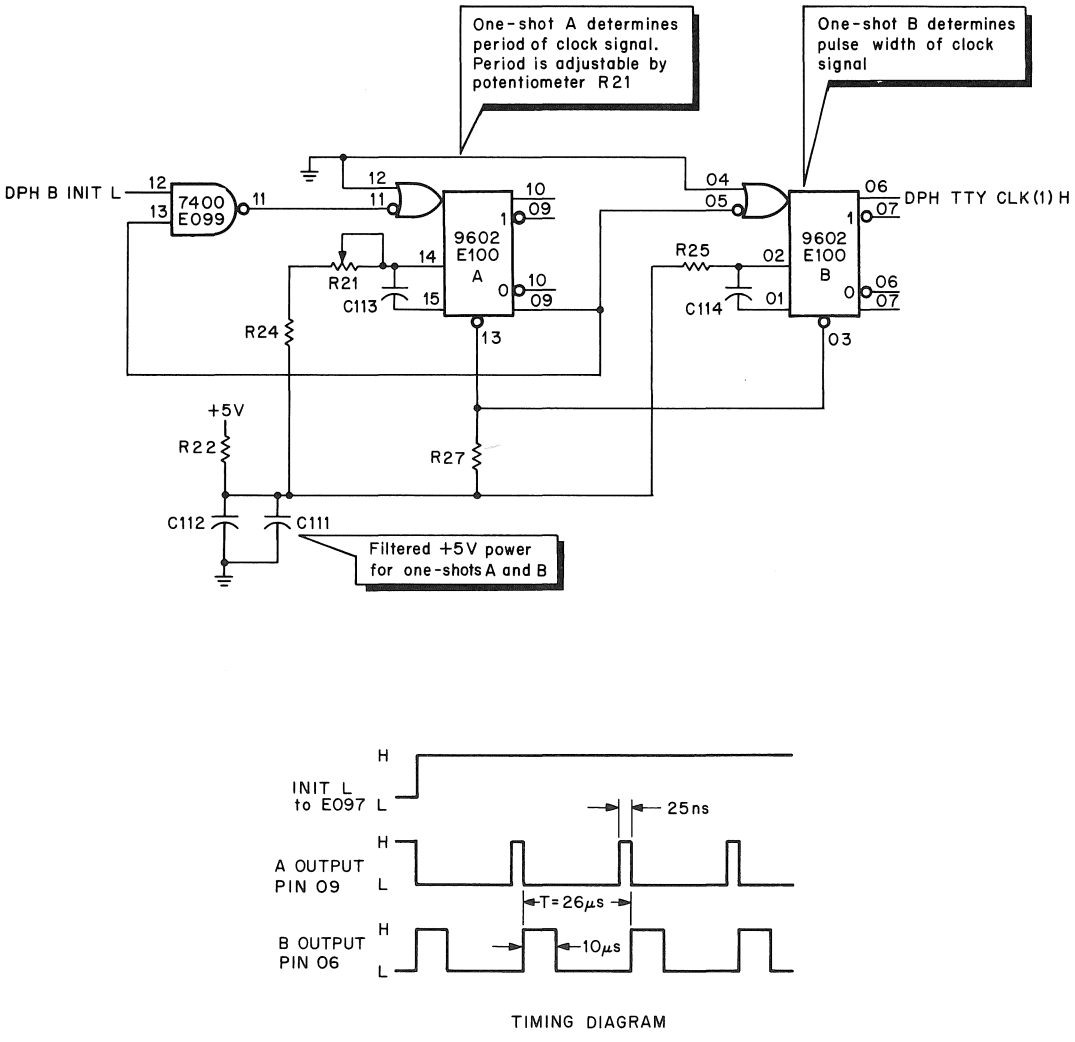


Figure 4-20 SCL Oscillator Schematic and Timing Diagram

The oscillator output, DPH TTY CLK (1) H, and counter outputs are connected to a single pole rotary switch whose output is one of five frequencies as shown below:

Switch Position	Output Frequency
01	Oscillator
02	Oscillator ÷ 2
03	Oscillator ÷ 4
04	Oscillator ÷ 8
05	Oscillator ÷ 16

The switch output is sent to pin 01 of NAND gate E099. The other input (pin 02) of this gate is pulled high except when an external clock is used rather than the one on the M7260 module. Under these conditions, the clock signal as chosen by the switch appears inverted at the output (pin 03) of E099. This clock signal is sent to pin 04 of another E099 NAND gate, which is shown as the logically equivalent negative input OR gate. The other input (pin 05) of this gate is also high except when an external clock is used; therefore, the clock signal is inverted again and appears as DPH TTY CLK H at pin 06 of the second E099 gate. The signal clocks the UART receiver and transmitter and counter E088 which generates signal DPH START H.

Two commonly used baud ranges are obtained by selecting 26 μs or 35.5 μs as the period of the oscillator output (Table 4-11).

An external clock can be used instead of the clock on the M7260 module. The module clock is disabled by a low level on pin FH2 (Figure 4-21). The disabling signal is called FS CLK DISAB L. The external clock is connected to pin FH1 and is called FS CLK L. The inverted external clock signal appears at pin 06 of E099.

Table 4-11
Baud Selection

Switch Position	Range 1		Range 2	
	Baud	Period*	Baud	Period*
1 (OSC output)	2400	26 μs	1760	35.5 μs
2 (OSC ÷ 2)	1200	52 μs	880	71 μs
3 (OSC ÷ 4)	600	104 μs	440	142 μs
4 (OSC ÷ 8)	300	208 μs	220	284 μs
5 (OSC ÷ 16)	150	416 μs	100	568 μs

*Period of clock signal DPH TTY CLK H observed on backplane pin E02D2.

4.11 BAUD RATE ADJUSTMENT

Use the following procedure to adjust the baud rate for the serial communications line.

1. Turn power off.
2. Extend the computer out of the cabinet until the chassis slide lock clicks.
3. Remove the mounting box top and bottom covers.

4. Remove the M7260 module. Using a small blade type screwdriver, turn the baud adjustment switch (S1) fully counterclockwise, which is position 1. Refer to Table 1 to determine the switch position for the selected baud rate. Set the switch to the proper position and insert the M7260 module into the backplane.

5. Turn power on.

6. Connect an oscilloscope probe to backplane pin E02D2 to observe signal DPH TTY CLK H (Figure 4-21). This is the UART clock signal; its frequency is 16 times the baud rate.

7. Adjust the SCL oscillator frequency to match the selected baud rate in accordance with Table 4-11. This table lists the period of clock signal DPH TTY CLK H versus baud rate for two adjustment ranges. For example, assume that in step 4 the baud switch was set to position 4 to select 300 baud in range 1. The corresponding clock signal period is 208 μs. It is set by adjusting potentiometer R21 on the M7260 module, which is accessible with the module installed. The clock period is displayed on the oscilloscope. Clockwise adjustment decreases the period and counterclockwise adjustment increases the period.

NOTE

Once this adjustment is made, other baud rates within the same range can be selected by changing only the baud switch position. If the new baud rate is in the other range, steps 1 through 7 must be performed because switching ranges requires a change in the period of the clock signal.

8. Remove the oscilloscope probe, replace the mounting box top and bottom covers and push the computer back in the rack. Signal DPH TTY CLK (1) H is the output of the SCL oscillator. After the adjustment procedure is completed, the period of this signal is 26 μs for range 1 and 35.5 μs for range 2. This signal is accessible for observation only by extending the M7260 module and directly probing E100 pin 06.

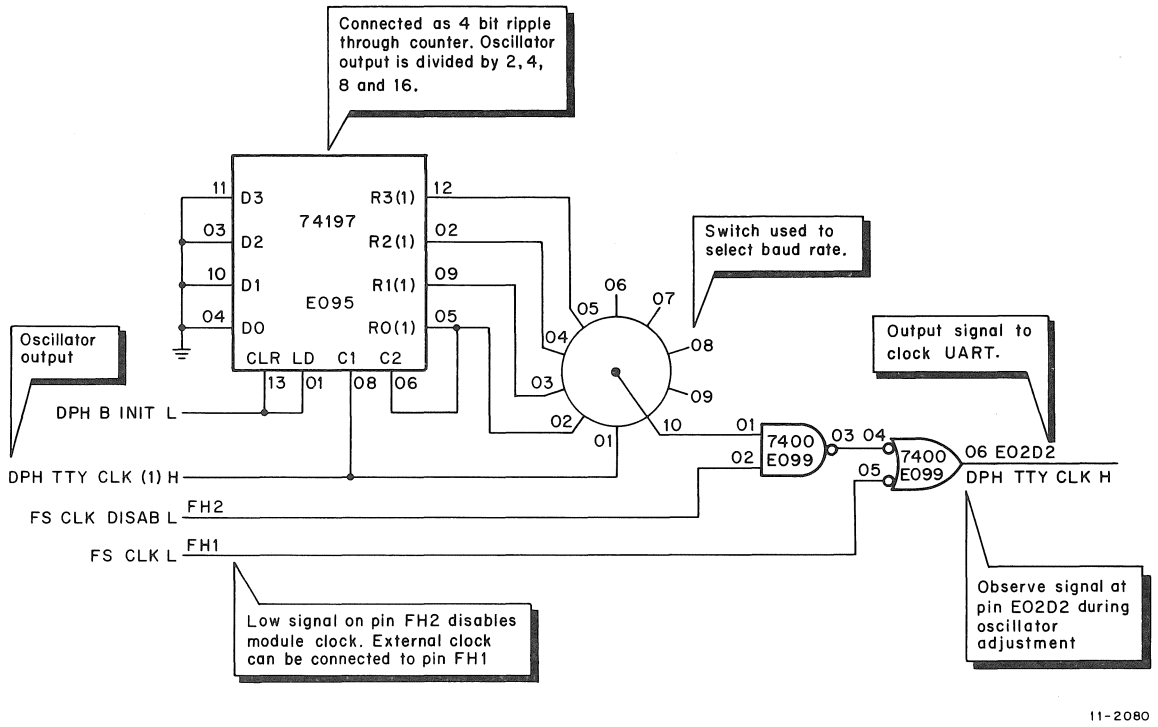


Figure 4-21 Baud Selection Logic

4.12 LINE CLOCK

4.12.1 Introduction

The line clock allows the program to measure time by sensing the frequency (50 Hz or 60 Hz) of the ac input power. The sensing signal is generated by the power supply. It is a positive, approximate square wave developed from the ac input waveform. For a 60-Hz supply, this signal occurs at a rate of 16.7 ms; the rate for a 50-Hz supply is 20 ms. Each sensing signal generates a flag that can be read on Unibus data bit 07 and can be cleared only by program control. A line clock interrupt signal is generated concurrent with the flag signal, provided the interrupt enable bit is set by the program. This interrupt signal is used in the processor priority arbitration logic. An interrupt enable flag is generated and can be read on Unibus data bit 06.

The line clock is not connected to the Unibus. It uses an internal bus and can be accessed only by the processor and console; however, to the operating program, the line clock is indistinguishable from other devices that are attached to the Unibus. This is accomplished by logic that decodes the address of the line clock on the Unibus as an internal address.

The line control logic is shown in print CONI. It can be divided into two sections: flag control and interrupt control. Each section is described in detail in subsequent paragraphs.

4.12.2 Flag Control

The flag control logic is shown in the top of print CONI. Signal PWR SUPPLY L CLK INT H is the sensing signal from the power supply. This signal is generated by a resistor-Zener clipper circuit. The positive half cycle is clipped at approximately +4 V and the negative half cycle is clipped at approximately −1 V. This produces a clipped sine wave (approximate square wave) with a pulse height of nearly +5 V (base line at −1 V). These positive pulses occur every 16.7 ms for a 60-Hz input and every 20 ms for a 50-Hz input.

Signal PWR SUPPLY L CLK INT H is sent to the input of NAND Schmitt trigger E39. Each positive input is converted to a clean negative square pulse at the output. The positive-going edge of this pulse clocks the CLOCK flip-flop E73. This is a redefined flip-flop with its D-input connected to ground (CONH RUN GND L). When clocked, the flip-flop is set and its 1-output (pin 08) is high. This signal is sent to the input (pin 11) of Unibus driver E1. The output of this driver is the line clock flag signal BUS D07L. It can be read during a DATI operation by providing an enabling signal to pin 12 from gate E14. This gate is enabled when both inputs are low. The inputs are: CONC BBSY (1) L which is asserted when the BBSY flip-flop E15 (print CONC) is set; and CONA ENAB L CLK PSWL which is generated by DATI ADDRS DECODE ROM E72 (print CONA).

The flag can be cleared only by program control. It is accomplished by signal CONI CLR CLOCK L via the clear input (pin 10) of the CLOCK flip-flop. This signal is generated at the output (pin 6) of NAND gate E85. One input is a resultant of DPB AMUX 07 H and CONA LOAD L CLK PSW L from E74. The other input is CONT PROC CLOCK H. Signal CONA LOAD L CLK PSW L is generated by DATO ADDRS DECODE E71 (print CONA). It is low when the line clock address is decoded. When this signal is ANDed with DPB AMUX 07 H, input 4 of E85 is high. If the flag is to be cleared, CONJ PROC CLOCK H will also be high, creating a low at the output of E85. A low at the output of E85 will clear flip-flop E73.

4.12.3 Interrupt Control

The interrupt control logic is shown on the bottom of print CONI. The Schmitt trigger output also clocks LC INT flip-flop E87 which is a redefined flip-flop with its D-input connected to ground. When clocked, the flip-flop is set and its 1-output (pin 06) is high. This signal is sent to the D-input of LC INT SYNC flip-flop E86. When clocked, the 1-output of the LC INT SYNC flip-flop is high. This signal is sent to pin 1 of NAND gate E85. This gate generates signal CONI L CLK INT L that is used in the processor priority arbitration logic. It can be regarded as an internal BR signal. Signal CONI L CLK INT L is asserted when both inputs (pins 1 and 2) are high. Pin 1 is high as discussed above and pin 2 is high when the program sets the interrupt enable bit and the processor priority is not 6 or 7. The logic for qualifying pin 2 is discussed below.

The INT ENAB flip-flop E73 (CONI) is clocked by the CONA LOAD L CLK PSW L, and CONJ PROC CLOCK H signal via inverter E83 and NAND gate E85.

To set the interrupt enable bit, the program generates a high on DPB AMUX 06 H which is the D-input of the INT ENB flip-flop. When clocked, the INT ENB flip-flop is set and its 0-output (pin 06) is low. This signal is sent to pin 12 of gate E74. The other input of this gate comes from the output of NOR gate E74 which is low when either or both of its inputs are high. The inputs are DPE PSW 07 (0) H and DPE PSW 06 (0) H. They come from the 0-outputs of PSW (07:04) flip-flop E55 (print DPE). This flip-flop stores the current priority of the processor in bits 07, 06, and 05. The two outputs used are the complement of bits 07 and 06 of the priority word. The qualifying condition for the interrupt control logic is that the processor priority not be 6 or 7. That is, the output of gate E74 pin 10 is low when the processor priority is not 6 or 7. This condition is verified as shown below.

Priority Bits DPB AMUX 07 H, 06 H and 05 H to PSW (07:04) flip-flop				**Complement of Bits 07 and 06 from PSW (07:04) flip-flop**		
	PSW Bits					
Priority	**07**	**06**	**05**	**07**	**06**	
7	1	1	1	0	0	NOR Gate E74 Disabled (Output High)
6	1	1	0	0	0	
5	1	0	1	0	1	NOR Gate E74 Enabled (Output Low)
4	1	0	0	0	1	
3	0	1	1	1	0	
2	0	1	0	1	0	
1	0	0	1	1	1	
0	0	0	0	1	1	

The low output of NOR gate E74 is sent to input pin 11 of E74. The other input is low because the INT ENAB flip-flop is set. This generates a high at the output of E74 (pin 13) which is sent to pin 2 of E85. The other input (pin 1) of E85 is also high which asserts CONI L CLK INTL.

The 1-output of INT ENAB flip-flop E73 is sent to pin 2 of Unibus driver E1. The output of this driver is the state of the interrupt enable bit; however, it is designated BUS D06L. This bit can be read during a DATI operation by enabling the gate with the output of E14, which is explained in the flag control discussion.

When the interrupt is serviced, the microprogram performs a BUT SERVICE which asserts signal CONE L CLK SER L from the INT INTR ACK ROM (E110 print CONE). This low signal is gated with a low clock pulse (CONJ PROC CLOCK H) at pins 09 and 10 of E96 to produce a low output (pin 8). The low output from pin 8 is used to clear the LC INT flip-flop via its clear input (pin 10). When the LC INT flip-flop is cleared, it sends a low signal to the D input of the LC INIT SYNC flip-flop. On the next clock pulse (CONJ PROC CLOCK L), the LC INT SYNC flip-flop is reset.

4.13 POWER FAIL

The KD11-B power fail/auto restart circuitry (print CONH) serves the following purposes:

a. Initializes the microprogram, the Unibus control (BC), and the Unibus to a known state immediately after power is applied to the computer.

b. Notifies the microprogram of an impending power failure.

c. Prevents the processor from responding to an impending power failure for 2 ms after initial startup.

The actual power fail/auto restart sequences are microprogram routines. The operation of the power fail/auto restart circuitry depends on the proper sequencing of two bus signals: AC LO and DC LO. Because of the electrical properties of the Unibus drivers and receivers, the entire computer system must be powered up for the machine to operate. Therefore, the processor is notified of a power fail in peripherals as well as in its own ac source.

The notification of power status of any PDP-11 system component is transmitted from each device by the signals BUS AC LO L and BUS DC LO L (Figure 4-22). The power-up sequence shows that BUS DC LO L is unasserted before BUS AC LO L is unasserted. When BUS DC LO L is not asserted, it is assumed that the power in every component of the system is sufficient to operate. When BUS AC LO L is not asserted, there is sufficient stored energy in the regulator capacitors of the power supply to operate the computer for 5 ms, should power be shut down immediately.

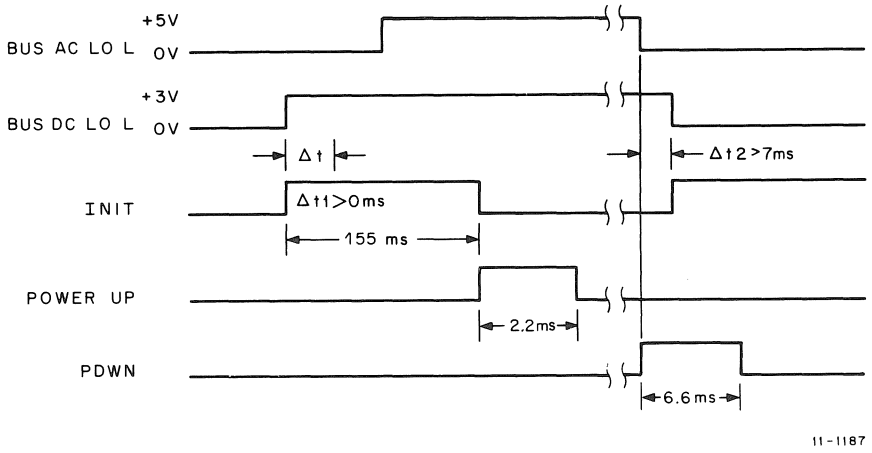


Figure 4-22 BUS AC LO and BUS DC LO Timing Diagram

As power is shut down, note that BUS AC LO L is asserted first. BUS AC LO L is an indicator that warns the processor of an impending power failure. When BUS DC LO L is asserted, it must be assumed that the computer system can no longer operate predictably. Memories manufactured by Digital use BUS DC LO L as a switch signal. When BUS DC LO L is asserted, these memories turn themselves off even if power is available. Time Δ +2 (Figure 4-22) is the time delay between the assertion of BUS AC LO L and the assertion of BUS DC LO L; it must be greater than 7 ms. This allows for power to be rapidly cycled on and off. According to PDP-11 specifications, upon system startup, a minimum of 2-ms run time is guaranteed before a power fail trap occurs, even if the line power is removed simultaneously with the beginning of the power-up sequence. After the power fail trap occurs, a minimum of 2-ms run time is guaranteed before the system shuts down. Given the tolerances permitted in the timing circuitry used in most equipment, Δ +2 must be greater than 7 ms.

When an impending power fail is sensed, a program trap occurs that causes the present contents of R7 and the PSW to be pushed onto the memory stack, as determined by the contents of R6. R7 is then loaded with the contents of memory location 24_8, and the PSW is loaded with the contents of location 26_8. Processing is continued with the new R7 and PSW. The program must prepare for the impending power failure by storing away volatile registers and reloading location 24_8 and 26_8 with a power-up vector. This vector points to the beginning of a restart routine.

When power is restored, the processor loads R7 with the contents of location 24_8 and the PSW with the contents of location 26_8. Note that no stacking is performed on an auto restart. The HALT switch is also ignored if the console lock is set. After loading R7 and the PSW, processing continues if the HALT switch is not depressed. Presumedly, the program will prepare locations 24_8 and 26_8 for another power failure. If the HALT switch is depressed and the console lock is not enabled, the processor powers up in the halt state.

Schematics of the power fail, auto restart, and bus reset logic are found on print CONH. As shown on Figure 4-19, E6707 generates a 150 ms processor INIT pulse as soon as BUS DC LO L is nonasserted after power is applied to the computer. At the end of 150 ms, the PUP one-shot, IC E5707, is fired if BUS AC LO L is not asserted. At this point, the processor begins to load R7 and the PSW if the HALT switch is not depressed. The PUP one-shot generates a 2.2-ms pulse, during which time the assertion of BUS AC LO L is not recognized.

After PUP has been reset, the assertion of BUS AC LO L fires the one-shot E6710. Flip-flop E8708 is set by the leading edge of the one-shot's pulse. Note that E8708 is not synchronized to the processor clock. Flip-flop E8613 generates the signal CONH PDWN SYNC (1) L, which is synchronized to the processor clock. A power fail trap can be recognized by the microprogram whenever CONE BUT SERVICE L is issued. The various traps are arbitrated by the ROM F101 (print CONE).

If a momentary power failure occurs which causes the assertion of BUS AC LO L but does not cause the assertion of BUS DC LO L, the processor will restart when the PDWN (0) L one-shot times out, retriggering the INIT one-shot simultaneously with DC LO H becoming nonasserted.

CHAPTER 5
KD11-B AND CONSOLE MAINTENANCE

5.1 INTRODUCTION

This chapter describes techniques for isolating and repairing failures in the KD11-B and the console. The basic procedures are aimed at differentiating between failures in the processor and the remainder of the computer. If the processor is at fault, it is necessary to determine which of the two KD-11B modules is defective. The KM11 maintenance panel may be used in conjunction with the KD11-B documentation to isolate failure to specific integrated circuits.

The easiest method of isolating failures and determining if a system will function under worst-case conditions is to use diagnostic programs that have been designed by DEC to test the processor and memory. For most DEC computers, it is possible to assemble a hierarchy of diagnostics that progressively tests more and more of the computer. With large systems, it is possible to test the KD11-B beyond its performance specifications. Diagnostic programs are written and commented in a manner that guides the user in determining computer malfunctions.

This chapter also describes special techniques for troubleshooting the KD11-B. The exact determination of failures and their repair requires careful application of the tools described in this chapter, in addition to a general knowledge of PDP-11 systems. A section on console maintenance provides a console troubleshooting procedure.

5.2 DIAGNOSTICS

The diagnostic programs supplied by DEC provide a rigorous test of the computer that can indicate the need for service even before a failure occurs. Preventive maintenance is especially important on machines that include mechanical components, such as line printers or tape drives.

5.3 TYPES OF FAILURES

Failures can be broken down into three classes: basic, complex, and peripheral. A basic failure of the processor, memory, or program read-in device does not permit diagnostic software to be loaded; thus, fault isolation and repair in a computer with a basic failure requires an elementary approach. A complex failure typically occurs only with programs that generate interaction on the Unibus between several peripherals and the processor. DEC provides a number of system diagnostics, such as the General Test Program (GTP) and the Communications Test Program (CTP), that are useful in isolating complex failures.

Often the failure is caused by a peripheral problem that is unrelated to the processor or memory. In this case, the processor itself may be used as a troubleshooting tool. For example, a diagnostic program is available that tests the alignment of a TU10 Magnetic Tape Drive and reports significant parameters via the serial communications line (SCL).

5.4 SUGGESTED EQUIPMENT

Table 5-1 provides a list of test equipment, maintenance devices, and tools used to perform the processor maintenance procedures and adjustments.

Table 5-1
Test Equipment and Tools

Equipment		Description
Test Equipment:	Oscilloscope	Tektronix Model 453 (or equivalent)
	Volt-ohmmeter	Triplett Model 630 (or equivalent)
Devices:	Extender Board	Three W984A Double Extender Boards
	Maintenance Module Set	One W130 (two are desirable) One W131 (two are desirable)
	Maintenance Module Overlays	KM1-DEC Part No. 55-09081-9 KM2-DEC Part No. 55-09081-10
	IC Test Clip	
Tools:	Small Flatblade Screwdriver	

5.5 PROCEDURES

It is useful to know the precise condition of the computer at the time of the failure. The user is advised to record the state of the computer, in as much detail as needed, to reproduce the problem when a failure occurs. At least the following information should be noted:

a. Any peripherals attached to the Unibus not usually present.
b. The name of the program running when the failure occurred.
c. The state of the processor indicators (console) when the failure occurred.
d. If possible, the sequence of events preceding the failure should be noted.

When running a program on the KD11-B for the first time, it should be noted that certain subtle differences exist among the several PDP-11 processors that can cause problems when non-standard programming practices are used.

Once it is established that a hardware failure exists, the following checks are advised before dismantling the computer:

1. Verify that the power supply is attached to a live ac source and is functioning normally.

2. Verify that the Unibus is properly routed.

3. Be certain that grant continuity cards are properly placed whenever missing peripherals would break the BUS GRANT lines.

4. Be certain that no Unibus address conflicts exist.

Programs can be executed from the scratch pad memory (SP) locations, and if processor problems are suspected, this procedure should be tried to isolate the problem. Communication between the console and processor must be functioning properly in order to use this procedure, and is the first thing to check when a processor problem is suspected. Executing programs from the SP is advantageous for troubleshooting or checking the processor. When executing a program from the SP, the PC (R7) is incremented by one; however, BR instructions always modify the PC by multiples of two. Consequently, a BR instruction must be carefully used in a program to prevent the PC from being modified to an incorrect address. An example of a simple program that loops on two SP register locations is as follows:

Address	Instruction	Octal
177700 (R0)	NOP	000240
177701 (R1)	BR.-1*	000777

To load the above program from the console, perform the following steps:

1. Enter 177700 in the Switch Register and depress LOAD ADRS (this is the address of register 0).
2. Enter 000240 in the Switch REgister and lift DEP. (This places a NOP instruction in R0.)
3. Enter 000777 in the Switch Register and lift DEP.
4. Enter 177700 in the Switch Register and depress LOAD ADRS (this specifies the starting address).
5. Lift ENABLE/HALT to the ENABLE position.
6. Depress START. The RUN light should come on. The program is now being executed.
7. If ENABLE/HALT is pressed, the ADDRESS/DATA display should contain either 177700 or 177701.

When executing programs from the SP (registers), do not use the registers used by the processor (R6, R7, R10, R11, R12, and R17).

5.6 ADJUSTMENTS

Adjustments to the processor are as follows:

1. The processor clock should have a 310 ns period. Adjust, if necessary, performing the following procedures:
   a. Extend the M7261 module.
   b. With an oscilloscope, observe the processor clock at E010 pin 8.
   c. Adjust the potentiometer on the M7261 until the processor clock period is 310 ns.
   d. Remove oscilloscope probe and reinsert the M7261 module.

* 777_8 is normally a BR self-instruction. However, when executed from the SP, it is a BR. −1, because the SP registers are located on BYTE ADDRESSES.

2. The SCL clock frequency should be 16 times the desired baud rate. Adjust, if necessary, using the following procedure:

   a. Extend the M7260 module.

   b. With an oscilloscope, observe the SCL clock at E099 pin 6.

   c. Adjust potentiometer R74 for 16 times the desired baud rate, according to Table 5-2.

   d. Remove oscilloscope probe and reinsert the M7260 module.

Table 5-2
Baud Rate Adjustment

Baud Rate	Period (μs)	Frequency (Hz)
110	568	1760
150	416	2400
300	208	4800

An alternate method for adjusting the SCL clock that does not require extending the module, is to run any program, such as T-17, that causes a continuous stream of characters to be printed on the console. The potentiometer on the M7260 should then be adjusted to the center of the range for which satisfactory characters are printed.

5.7 KD11-B PRINT FUNCTION TABLE

The principles of operation of the KD11-B logic are described in Chapter 4. The microprogram is described in Chapter 3. The KD11-B print set is described in Chapter 3. Table 5-3 lists each engineering drawing for the KD11-B processor and describes the functions of the items shown on that drawing.

Table 5-3
Engineering Drawing Print List and Functions

Print Designation	Print Title	Function of Logic on Print
D-CS-M7260-0-01		
DPA	Data Path (3:0)	This print contains the least significant four bits of the DP components, including the ALU, the scratch pad, the B register, the AMUX, Unibus data drivers and receivers, and additional A-leg gating for the PSW and console switches. Prints DPB, DPC, and DPD contain the three other 4-bit parts of the data path.
DPB	Data Path (7:4)	In addition to the items mentioned above, DPB contains the constants generator.

Table 5-3 (Cont)
Engineering Drawing Print List and Functions

Print Designation	Print Title	Function of Logic on Print
D-CS-M7260-0-01 (Cont)		
DPC	Data Path (11:8)	Same as DPA
DPD	Data Path (15:12)	Same as DPA
DPE	PSW	DPE contains the 8-bit PSW and the multiplexers required to load it. Rotate multiplexers are also shown on DPE. The console (MUX shown in the lower right-hand corner of DPE) converts the data presented on the B-leg into a serial bit stream for the console display.
DPF	AUX ALU CONTROL	In addition to the auxiliary ALU control, the Instruction Register (IR) and the C- and V-bit encoder are shown on DPF.
DPG	IR DECODE	The major elements of the IR decoder are shown on DPG.
DPH and DPH1	SCL CONTROL	The UART and other elements of the SCL control are shown on DPH and DPH1.
D-CS-M7261-0-01		
CONA	INT ADDR	The Bus Address Register (BR) is shown on the left side of CONA. On the right half of the print, the logic required to detect reference to internal registers is diagrammed.
CONB	STACK FLOW AND SPAM	The left half of CONB contains the Scratch Pad Address Multiplexer (SPAM) while the right side contains the stack overflow and RUN flip-flops.
CONC	UNIBUS CONTROL (BC)	Data requests flip-flops are shown towards the left edge of CONC. The lower right-hand quarter of the print contains bus error and CKOFF flip-flops. The 9602 that detects non-existent memory is shown in the lower left-hand corner of CONC.
D-CS-M7261-0-01		
CONC1	PRIORITY ARBITRATION	The priority arbitration logic for bus requests is shown along the bottom edge of CONC1. Towards the left and top of CONC1 are three 4-bit latches used to hold signals received from the Unibus. The 9602 shown on the upper right of CONC1 is used to clear the bus if SACK is not received 22 μs after NPG or BG.
COND	DRIVE AND RECEIVERS	COND contains all of the Unibus drivers and receivers except those used for the data lines and two drivers used for the line clock circuit.

Table 5-3 (Cont)
Engineering Drawing Print List and Functions

Print Designation	Print Title	Function of Logic on Print
D-CS-M7261-0-01 (Cont)		
CONE	MICRO BRANCH LOGIC	A 4-to-16 line decoder associated with the BUT field of the microprogram is located in the upper left of CONE. The function switch buffers and decoders are shown in the upper middle and upper right. Two flip-flops associated with the console EXAM and DEP keys are shown in the center of CONE. The gates which affect the MPC during a BUT are located in the lower left.
CONF	MPC	Sixteen bits of the CS are shown on the left side of CONF. The MPC is along the right edge of the print.
CONG	CONTROL STORE (CS)	The remaining 23 bits of the CS are shown on CONG.
CONH	POWER FAIL	Initialize and power fail circuitry is shown on CONH. The 9602 contained in the lower left-hand corner of CONH generates bus instructions during the RESET instruction.
CONI	LINE CLOCK	The circuit equivalent to the KW11-L is contained on CONI.
CONJ	PROCESSOR CLOCK	The circuit consisting of E39, R6, R7, and C109 comprises the oscillator that generates the processor clock. The input to E03912 is used by the KM11 to control clock signals manually.

5.8 EXTERNAL CLOCK INPUTS

External clock inputs and corresponding internal clock disables are provided for the serial communications line (SCL) clock and the processor clock. The external input for the SCL clock permits the reception and transmission of serial asynchronous data at rates up to 10,000 baud. High baud rate signals should be input on pin FM1 of the M7260, rather than the low frequency input on pin FN1. The SCL clock, its external disable, and external clock input are shown on print DPH.
The external clock input for the processor clock permits the synchronization of two processors or the use of a manual clock. The manual clock input and the internal processor clock disable are shown on print CONJ.

5.9 KM11 MAINTENANCE PANEL

The discussion to this point has not considered the backplane or configuration. Every KD11-B contains the necessary logic to permit single step operation; however, the use of these facilities depends on the specific configuration. Two module slots are provided in the computer for the maintenance panel. Figure 5-1 contains a diagram of the KM11 overlays for slots KM-1 and KM-2 in the computer backplane. Table 5-4 provides description of the overlay designations. Note the following:

a. The KM11 switches have the same function in slots KM-1 and KM-2.

b. When the manual clock is enabled, bus error timeouts are disabled. Nonexistent memory trap cannot occur in manual mode.

c. Each actuation of the manual clock with line EV1 of the M7261 grounded produces bus control (BC) clock. It normally requires two BC clock pulses to advance the microprogram counter (MPC) to the next address.

d. The MPC is duplicated on both KM11 slots. This permits the user who has only one KM11 to plug the unit into either KM-1 or KM-2.

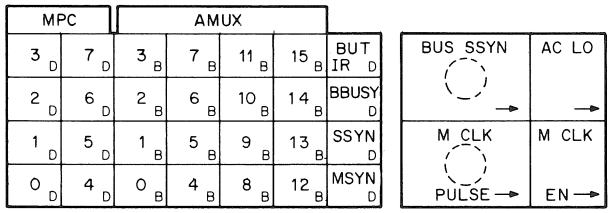
e. The MPC displayed on the KM11 is the address of the next microstep to be selected and not the present one.

f. Some lights on the maintenance panel indicate the assertion of a signal when illuminated and others indicate nonassertion when illuminated. This fact is indicated on the KM11 overlay drawing by the letter B for bright or D for dim appearing under each indicator light.

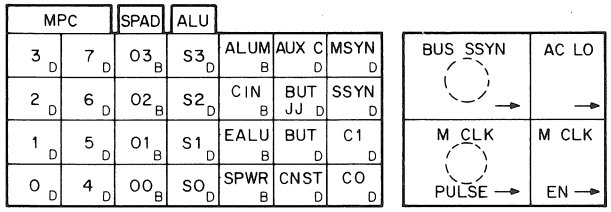
B = bright for assertion (logic 1)
D = dim for assertion (logic 1)

g. The wiring for KM-1 appears in slot A2, and the wiring for KM-2 appears in slot B2 for a Configuration 2 backplane. KM-1 and KM-2 are wired to slots A1 and B1, respectively, for a Configuration 1 backplane.

KM-1 is the more useful configuration and should be used to begin any repair attempts requiring the use of the maintenance panel. The console indicators display the B-leg input to the ALU, and the KM-1 configuration maintenance panel displays the output of the AMUX. If the ALU and AMUX are functioning, it is possible to deduce the contents of the A-leg by observing the console and the maintenance panel.



(a) KM-1 OVERLAY



(b) KM-2 OVERLAY

NOTE:
D = Dim when asserted.
B = Bright when asserted.

11-1271

Figure 5-1 KM11 Maintenance Module, KD11-B Overlays

Table 5-4
KM-1 and KM-2 Overlay Designations

Display	Definition
	KM-1 OVERLAY
MPC (7:0)	The address of the next microinstruction to be executed.
AMUX (15:0)	The 16-bit output of the AMUX.
BUT IR	BUT IR DECODE signal. When asserted, the microprogram is at F-5 and does a branch on the contents of the IR.
BBUSY	BUS BUSY. When asserted, BBSY indicates that a device has control of the Unibus.
SSYN	BUS SLAVE SYNC. When asserted, SSYN indicates that the Unibus slave device has responded to the master.
MSYN	BUS MASTER SYNC. When asserted, MSYN indicates that the master device on the Unibus is informing the selected slave that address and control information are present.
BUS SSYN	When actuated in the direction of the arrow (ON), SWITCH BUS SSYN asserts BUS SLAVE SYN as long as the switch is ON.
AC LO	When actuated in the direction of the arrow (ON), AC LO asserts BUS AC LO as long as the switch is ON.
M CLK PULSE (Manual Clock Pulse)	Each actuation in the direction of the arrow (ON), the processor generates one bus control clock, provided that CLK EN switch has been actuated. Two actuations will generate a processor clock.
M CLK EN	When actuated in the direction of the arrow (ON), it disables the processor clock logic and allows the M CLK PULSE switch to generate processor clocks.
	KM-2 OVERLAY
MPC 7 through 0	The address of the next microinstruction to be executed.
SPAD (Scratch Pad Address)	The address of the register (location) in the scratch pad memory.
ALU S3 through S0 ALU M	These five signals together indicate the function that the ALU is performing.
CIN	Carry in signal to bit 0 of the ALU.
E ALU	Enable ALU is the signal that switches the AMUX from inputting the Unibus data lines to inputting at the output of the ALU.

Table 5-4 (Cont)
KM-1 and KM-2 Overlay Designations

Display	Definition
SP WR	Scratch Pad Write indicates that the SPM is doing a write function as opposed to a read.
AUX CNTRL	Auxiliary Control enables the AUX ALU ROMs on print DPF.
BUT J J	Signifies that a branch test for a JMP or JSR instruction is occurring.
BUT UN	Signifies that a branch test for a unary instruction is occurring.
CNST	Signifies that the constants ROM, F025 on M7260, is enabled.
MSYN	Same as MSYN on KM-1
SSYN	Same as SSYN on KM-1
C1 and C0	BUS C1 and C0 together signify the type of Unibus cycle that is occurring:

C1	C0	
0	0	DATI
0	1	DATIP
1	0	DATO
1	1	DATOB

5.10 USING KM11 MAINTENANCE PANEL

Assume that the maintenance panel is plugged into slot A2 for the KM-1 overlay configuration. The M CLK EN switch must be activated in the direction of the arrow, which disables the processor M CLK PULSE. The following example uses the sequence of microsteps described in Paragraph 2.5.3.

With the HALT switch depressed, hold down the START switch. Toggle the M CLK PULSE switch advance two times, then release the START switch and toggle two more times. The processor should now be in microstep CS-2. The MPC should read 321_8, which is the contents of the NXT field of LOC 100_8 of the CS. Repeated actuation of the M CLK PULSE switch should cause the microprogram to proceed as follows:

Location	NXT (MICRO PC)	Step Name
100	322	CS-1
322	321	CS-2
321	40 + 1*	CS-3
41	302	H-1
302	300 + 2	H-2

*In step CS-3 the NXT field contains 40. However, if the HALT switch is depressed, a 1 is ORed into the NXT field to cause a branch to H-1.

5.11 CONSOLE MAINTENANCE

If any malfunctions are suspected in the console display logic, the console may be put into service mode. This mode of operation induces known data into the serial data line from the computer to verify that the counters, clock, and shift registers of the console logic on the console board are functioning properly. If the data on the console display does not match the known data, then the closed loop can be probed with an oscilloscope to determine the faulty area.

The procedure takes the four Scan Address lines and feeds them one at a time into the serial data output line, the address/data multiplexer is bypassed. Since the clock is free running, each scan address line displays a known data pattern in the console lights. The troubleshooting procedure for the console is as follows:

1. Make certain the computer power is off.

2. Disconnect the console cable connector from the M7260 module and then turn on the computer power.

3. After Step 2 is completed, the data pattern 177777_8 should be displayed on the console lights.

4. At the Berg cable connector that plugs into the M7260 module, use a piece of small gauge wire and jumper pin F (signal DAK, serial output line) to pin BB (ground). All the console lights should be off. Remove the jumper before proceeding to the next step.

5. At the cable connector, jumper pin F (signal DAK) to pin N (SCAN ADDRESS 01). The pattern displayed on the lights, should be 125252_8. Remove the jumper before proceeding to the next step.

6. At the cable connector, jumper pin F to pin L (SCAN ADDRESS 02). The pattern displayed on the lights should be 146314_8. Remove the jumper before proceeding to the next step.

7. At the cable connector, jumper pin F to pin J (SCAN ADDRESS 04). The pattern displayed on the lights should be 170360_8. Remove the jumper before proceeding to the next step.

8. At the cable connector, jumper pin F to pin D (SCAN ADDRESS 08). The pattern displayed on the lights should be 177400_8. Remove the jumper after completing the step.

KD11-B PROCESSOR
MAINTENANCE MANUAL
DEC-11-HKDBB-A-D

Reader's Comments

Your comments and suggestions will help us in our continuous effort to improve the quality and usefulness of our publications.

What is your general reaction to this manual? In your judgment is it complete, accurate, well organized, well written, etc.? Is it easy to use? _____

_____

_____

_____

What features are most useful? _____

_____

_____

_____

What faults do you find with the manual? _____

_____

_____

_____

Does this manual satisfy the need you think it was intended to satisfy? _____

Does it satisfy *your* needs? _____ Why? _____

_____

_____

_____

Would you please indicate any factual errors you have found. _____

_____

_____

_____

Please describe your position. _____

Name _____ Organization _____

Street _____ Department _____

City _____ State _____ Zip or Country _____

CUT OUT ON DOTTED LINE

— — — — — — — — — — — — Fold Here — — — — — — — — — — —

— — — — — — — — — — Do Not Tear - Fold Here and Staple — — — — — — — — —

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

Digital Equipment Corporation
Technical Documentation Department
146 Main Street
Maynard, Massachusetts 01754

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASSACHUSETTS 01754