EK-KS10-TM-PRE

KS10 TECHNICAL MANUAL (PRELIMINARY)

Interim Release

First Release (Review) 1/16/78
Second Release (Interim) 2/24/78

The material in this manual is for informational purposes and is subject to change without notice. Digital Equipment Corporation assumes no responsibility for any errors which may appear in this manual.

Printed in U.S.A.

PREFACE

This document is an interim version of the KS10 Technical Manual (Preliminary) that has been released for the first Field Service Training Seminar. Note that it is not complete and that not all the existing material has been validated. A final version of the document will be released prior to FCS.

CONTENTS

SECTION 1

INTRODUCTION

The DECSYSTEM-2020 is the hardware base for the new low end member of the DECsystem-10 and DECSYSTEM-20 families of computers. The system runs the TOPS20 operating system (Release 3) such that no differences from 2040/2050 appear at the user level. This capability provides a new kind of mainframe computer; that is, a machine with large computer software power in the mini-mid computer price range.

1.1 OVERVIEW

The configuration for the end-user version of the 2020 KS10 system is shown in Figure 1-1 and listed in Table 1-1. (Note that systems supported by DEC Field Service require a magtape.)

Table 1-1 2020 Configurations

Item	Min System	Typical System	Max System
CPU (KS10)	1	1	1
MEMORY	128K words	256K words	512K words
RM03 or RP06	1	2	8
TU45	0 (see note)	1	4

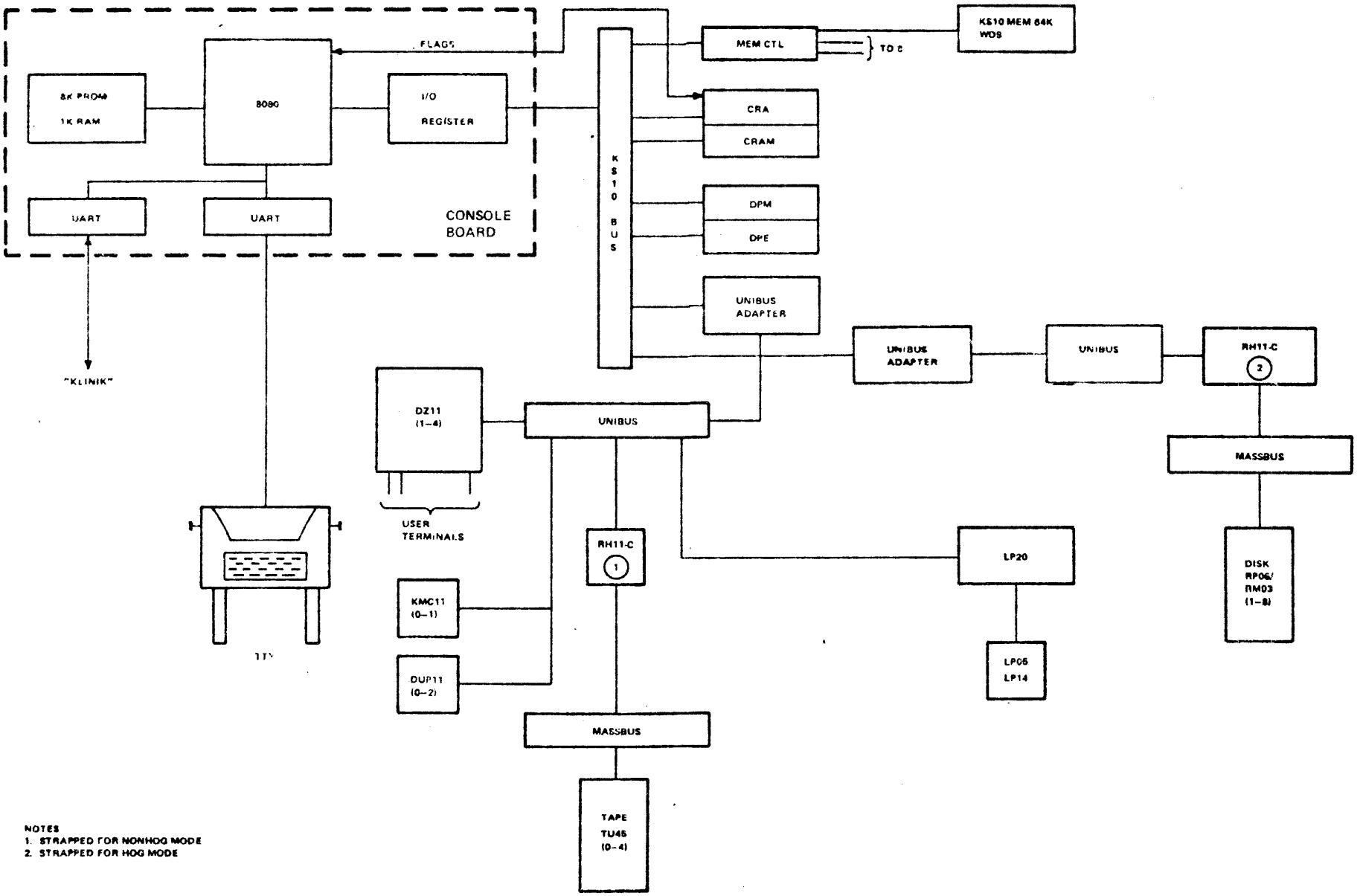


Figure 1-1 2020 System Configuration

SYNC LINES	0	1	2
LP05/LP14	0	1	1
TERMINAL LINES	8	16	32

NOTE

OEM-serviced systems only. DEC Software/Hardware Support will not maintain systems that do not have TU45 Magtape.

The KS10 has an internal backplane bus that provides a control and data path between the processor, memory, console, and peripheral devices (via Unibus adapters). It is a multiplexed 2-cycle bus that allows command and address information to be transmitted by one bus device to another during one bus cycle; data is then transferred to/from the addressed device during a following bus cycle.

The KS10 processor consists of 4 extended hex modules (i.e., data path modules DPE and DPM, and control-store modules CRA and CRM). The processor uses low power Schottky TTL and features the AM2901 4-bit data path slice. Other features include:

1. 512-word virtual-address cache memory.

2. 8 blocks of 16 fast general purpose registers.

3. Parity checking in micro-store, on data paths, and on backplane bus.

4. Fast byte operations on 7-bit ASCII characters.

5. 2K word (96-bits/word) writable RAM micro-store with address provision for 4K words.

6. Basic micro-instruction cycle time of 300 NS.

The KS10 memory system consists of a single extended hex control module that connects to the backplane bus and to 2-8 storage (array) modules. Each storage module contains 64K of MOS memory. Memory features include:

1. 1.050 μsec cycle time.

2. Single bit error correction.

3. Double bit error detection.

4. 128K words minimum capacity and up to 512K words maximum capacity.

The console consists of a single extended hex module that uses an 8080 microprocessor to perform console and diagnostic functions. Provision is made for a KLINIK connection that operates in parallel with the CTY and allows diagnosis of the system via a remote link.

KS10 peripheral devices are selected Unibus devices that interface to the system through Unibus adapters (UBAs). A UBA is a single extended hex module connecting to both the backplane bus and a Unibus. Up to three UBAs may be installed in the KS10 although two UBAs are standard in the end-user 2020 configuration. One UBA (and Unibus) is reserved for disks only. The second UBA (and Unibus) is used for all other devices; that is, for tape, line printer, and synchronous and asysnchrous communications lines. Characteristics and features of the devices supported on the UBAs are as follows:

DISKS

RP06

1. Average access time of 36.3 ms.

2. Average seek time of 28 ms.

3. Formatted capacity of 176 MB.

4. Maximum data transfer rate of 166K 36-bit words/second.

5. 128 36-bit words/sector.

6. Removable (20-surface) disk pack.

7. 18-bit (NPR) data transfers over Unibus; 36-bit (NPR) data transfers over backplane bus.

RM03

1. Average access time of 38.3 ms.

2. Average seek time of 30 ms.

3. Formatted capacity of 67 MB.

4. Maximum data transfer rate of 250K 36-bit words/second.

5. 128 36-bit words/sector.

6. Removable (5-surface) disk pack.

7. 18-bit (NPR) data transfers over Unibus; 36-bit (NPR) data transfers over backplane bus.

TM03/TU45

1. Tape speed of 75 IPS.

2. Density of 800/1600 BPI.

3. 9-track format

4. Maximum data transfer rate of 60K 18-bit words/second.

5. Uses 1/2 inch industry-standard tape.

6. 18-bit (NPR) data transfers over Unibus; 18-bit (NPR) data tranfers over backplane bus.

SYNCHRONOUS COMMMUNICATIONS INTERFACE

1. DUP11 single-line controller (1 per line).

2. Bit rate of 2000-19200 BPS.

3. DDCMP data protocol.

4. KMC11 NPR microprocessor (one per system).

5. 2 lines/system (maximum).

6. 8 or 16 bit (NPR) data transfers over Unibus; 8 or 16 bit (NPR) data transfers over backplane bus.

ASYNCHRONOUS COMMUNICATIONS INTERFACE

1. DZ11 8-line controllers.

2. RS 232C interface standard.

3. Baud rates of 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, and 9600.

4. 8, 16, 24, or 32 lines per system.

5. Character lengths of 5, 6, 7, or 8 bits.

6. 1, 1.5, or 2 stop bits.

7. Carrier, ring, data, terminal ready, and break MODEM control.

8. Odd/even parity.

9. Full duplex.

10. 64 character silo receive buffer (alarm at 16

characters).

11. 8-bit (register I/O) data transfers over Unibus; 8-bit (register I/O) data transfers over backplane bus.

1.2 PHYSICAL DESCRIPTION

The KS10 is compactly configured in a single width corporate high-boy cabinet (H7502H-7). This cabinet, shown in Figure 1-2, houses the KS10PA, BA11K drawer, power system, MASSBUS transition plate, asynchronous communication panel, and operator's switch panel.

1.2.1 KS10PA

The KS10PA assembly card cage is a hybrid style card cage; that is, it contains both extended hex and standard hex modules. It is located in the lower front portion of the KS10 cabinet as shown in Figure 1-3. This assembly contains the KS10 CPU, the MOS memory (128K words minimum, 512K words maximum), two Unibus adapters (UBAs), and the RH11C Unibus disk controller. Module utilization is shown in Figure 1-4.

1.2.2 BA11-K

The BA11-K drawer (Figure 1-3) contains the KS10 system's I/O peripheral controllers. It has dedicated locations for the following:

1. DZ11 asynchronous communications controllers: 1 minimum

FRONT VIEW

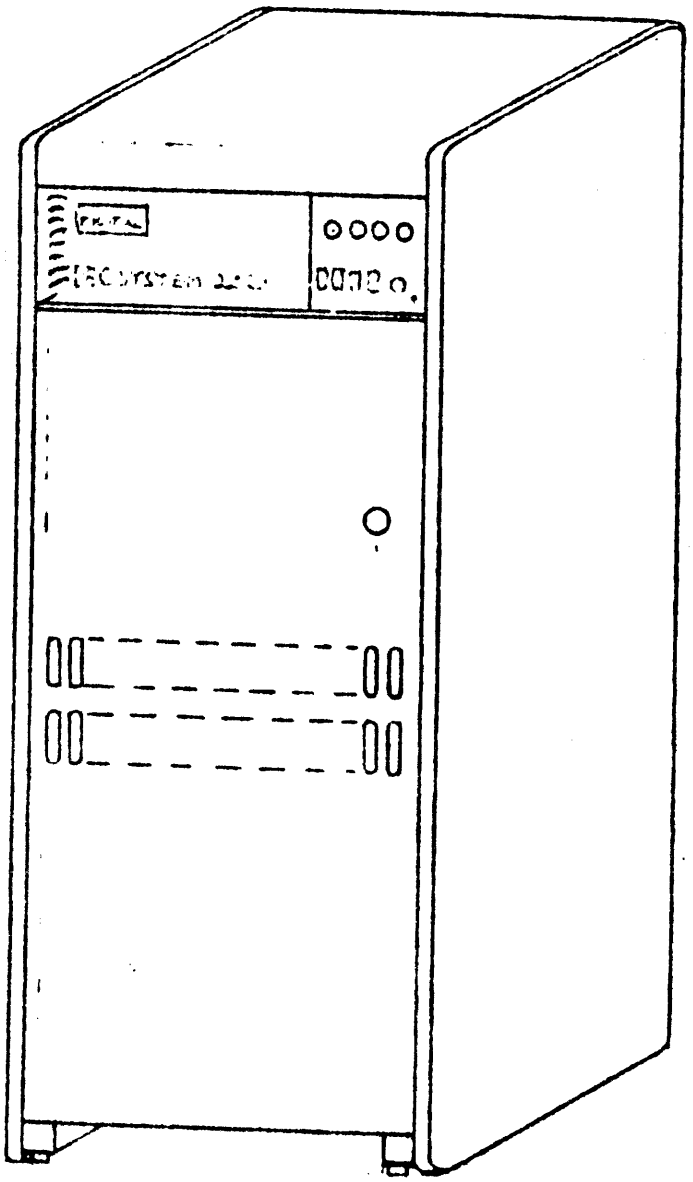


Figure 1-2 KS10 Cabinet

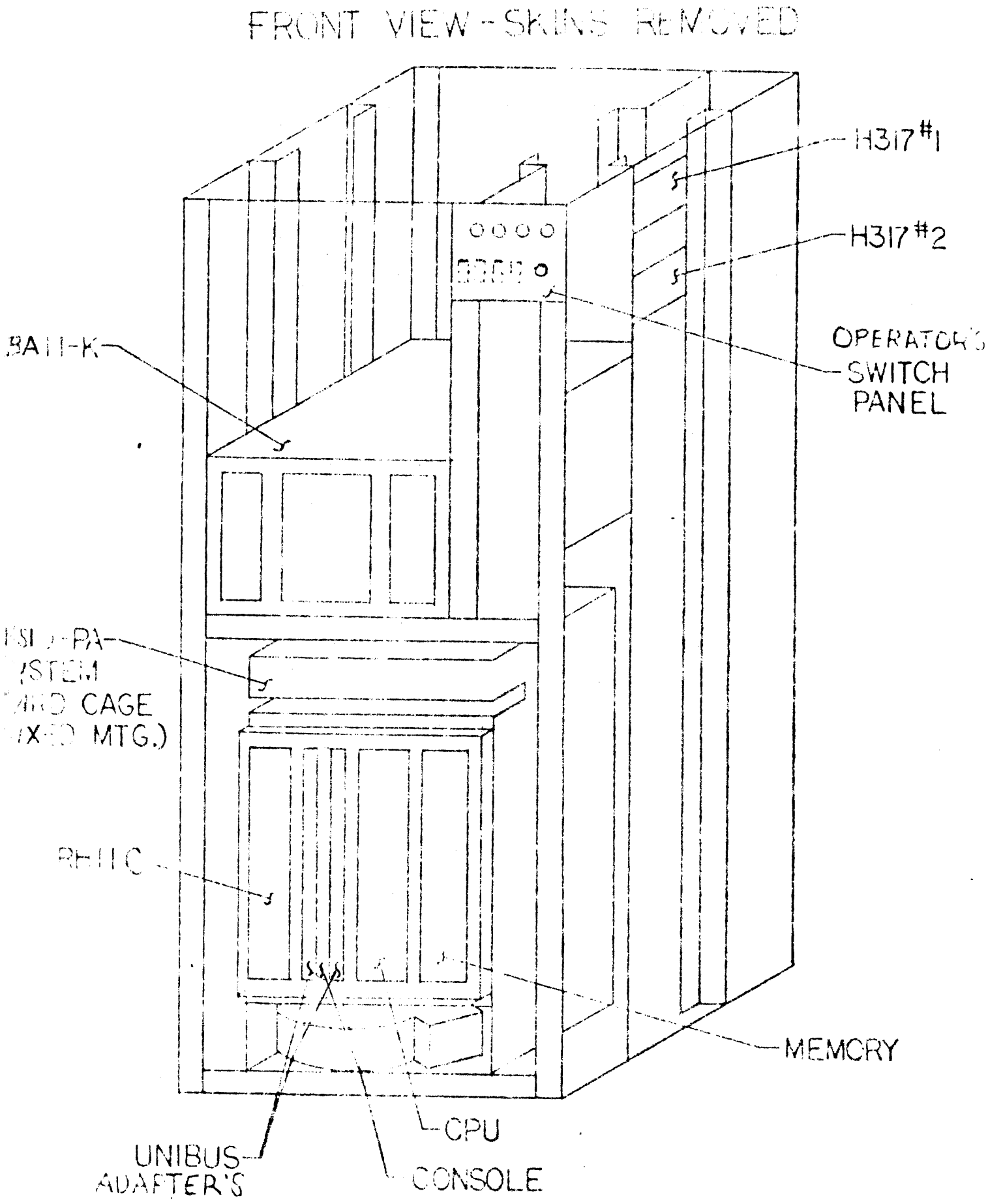


Figure 1-3 H310 Cabinet (Front View - Skins Removed)

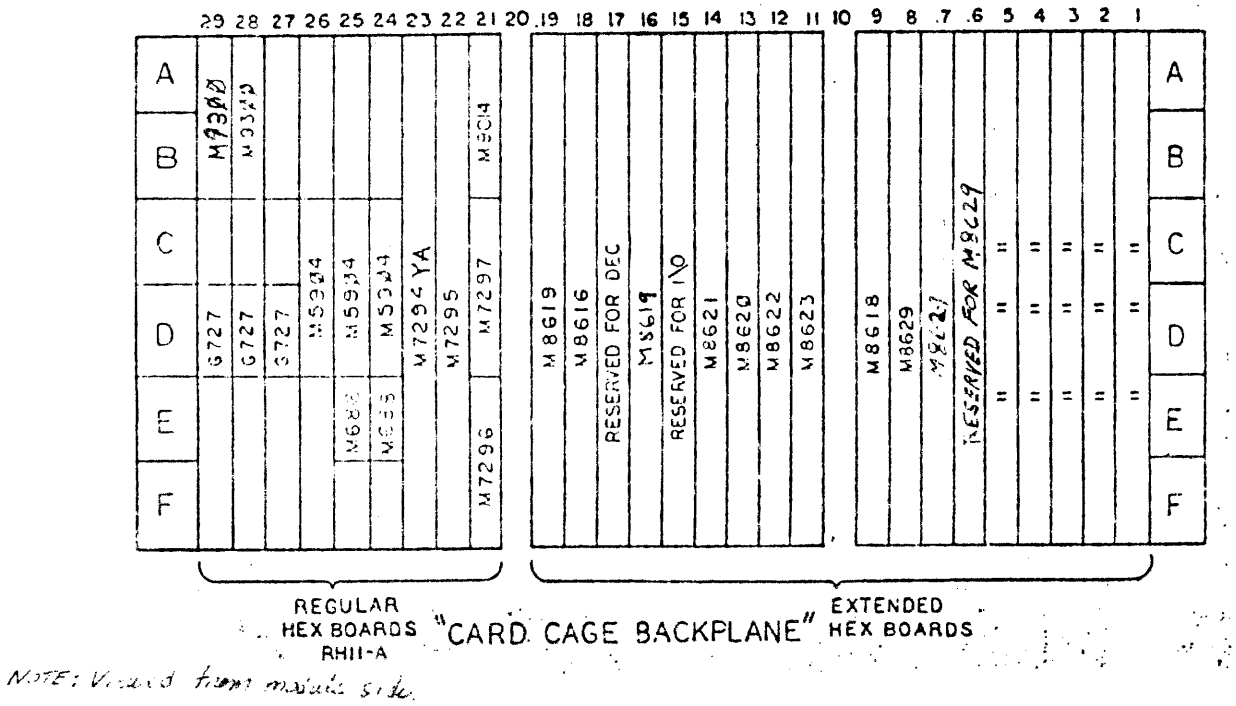


Figure 1-4 KS10PA Card Cage, MUL

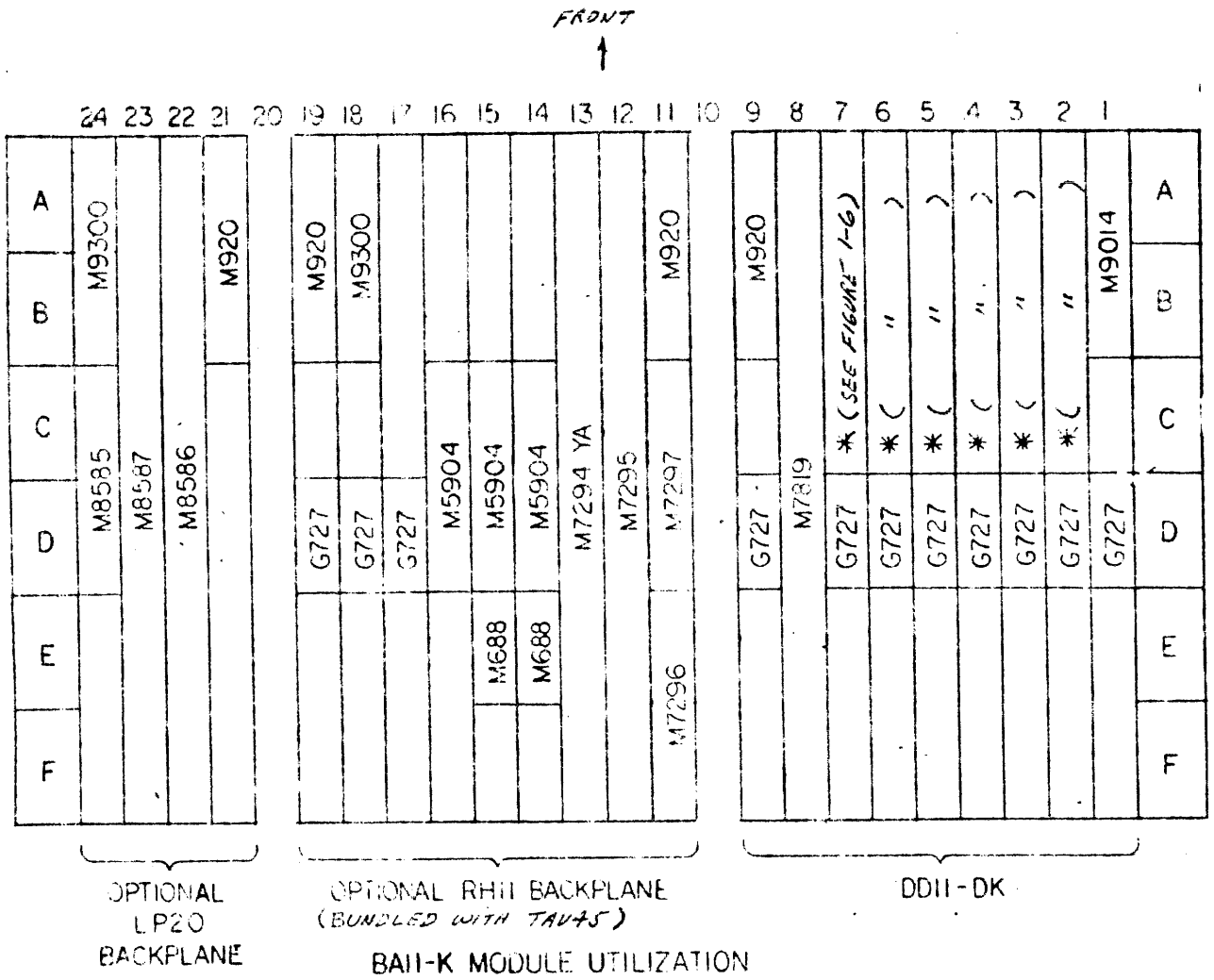
(8 lines), 4 maximum (32 lines).

2. DUP11/KMC11 synchronous communicationss controller: 0 minimum, 2 DUPils maximum (2 lines).

3. LP20 line printer controller: 0 minimum, 1 maximum.

4. RH11C magnetic tape system controller: 0 minimum, 1 maximum. (This option is bundled into the TAU45 tape system.)

BA11-K module utilization is shown in Figure 1-5. Option variations are shown in Figure 1-6.



NOTE: Viewed from module side.

Figure 1-5 BA11-K Drawer, MUL

* OPTION VARIATIONS					
SLOTS	ASYNC LINES 8-15	ASYNC LINES 16-23	ASYNC LINES 24-32	SYNC 1ST	SYNC 2ND
2				M8207 KMC11	
3					M7867 DUP11
4				M7867 DUP11	
5			M7819 DZ11		
** 6	(M7819 DZ11)	M7819 DZ11			
7	M7819 DZ11				

** NOTE: M7819 CAN ONLY BE INSTALLED IN SLOT 6 WHEN CONFIGURATION EQUALS 0-23 LINES.

Figure 1-6 BA11- K Option Variations

1.2.3 Power System

The major components in the KS10 power system are the 861 power control for AC power distribution, the L.H. switcher power supply for powering the KS10PA, and the H765 switcher power supply for powering the BA11-K. Component designations for 60 HZ and 50 HZ machines are as follows:

KS10AA (115V, 60 HZ)	KS10AB (230V, 50 HZ)
861C	861B
* L.H. P.S. (H7130C)	* L.H. P.S. (H7130D)
H765A (powers BA11K)	H765B (powers BA11K)

*** NOTE TO DEC IN-HOUSE FIELD SERVICE PERSONNEL**

The first of the in-house KS10 systems will contain H7130A (60 Hz) and H7130B (50 Hz) power supplies. Although input and output power specifications for these A and B (blue colored) models are the same as for the C and D (silver colored) models that are installed in all other machines, differences in power harness wiring prevent replacing one type with another. Thus, in the event of failure, replace a blue power supply with <u>only</u> a blue power supply, and replace a silver power supply with <u>only</u> a silver power supply.

1.2.4 MASSBUS Transition Plate

The MASSBUS transition plate is located at the top of the KS10 cabinet as shown in Figure 1-7. It is a connection plate that holds three MASSBUS connectors plus two 25 pin communications cable connectors. The MASSBUS connectors are allocated from right to left as follows:

1. Disk MASSBUS channel
2. Tape MASSBUS channel
3. Line printer channel

The two communication cable connectors are allocated as follows:

1. CTY (BC03L to BC03M)
2. KLINIK remote maintenance port (BC03L to BC05D)

1.2.5 Asynchronous Communications Panel (H317Es)

The KS10 is configured with a minimum of 1-H317E (up to 16 lines) and a maximum of 2-H317Es (32 lines). It will be configured with EIA communication only.

MINIMUM CONFIGURATION

Lines 0-7:

1 - DZ11 Module (8 line MUX)

1 - H317E distribution panel

1 - BC05W - 8 cable

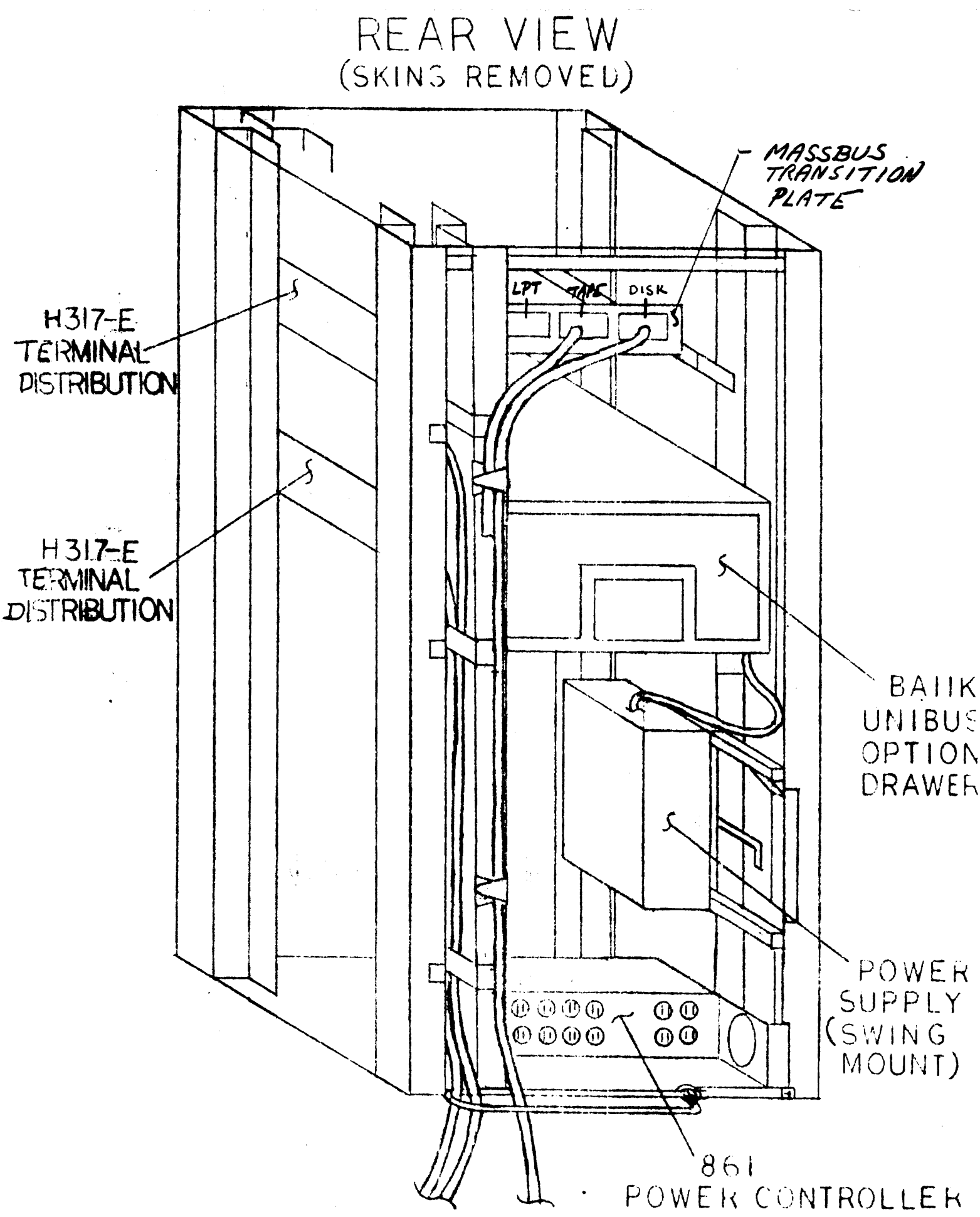


Figure 1-7 KS10 Cabinet (Rear View - Skins Removed)

OPTIONAL EXPANSION

Lines 8-15 (defined as a DZ11BA):

1 - DZ11 Module (8 line MUX)

1 - BC05W-8 cable

Line 16-23 (defined as a DZ11AA):

1 - DZ11 Module (8 line MUX)

1 - BC05W-8 cable

1 - H317E distribution panel

Line 24-32 (defined as a DZ11BA):

1 - DZ11 Module (8 line MUX)

1 - BC05W-8 cable

1.2.6 Operators Switch Panel

The operator's switch panel is located at the top-most front position in the KS10 cabinet (Figure 1-2). Switch and indicator functions are given in Section 4.

SECTION 2

SITE PREPARATION AND PLANNING

2.1 SITE PLANNING

Refer to Section 1 (Subsections 1.1 - 1.5) of the DECSYSTEM-20 Site Preparation Guide for the following information:

1. Schedule of site preparation prior to system delivery.

2. Summary of site preparation functions and responsibilities.

3. Site consideration and selection.

4. Building requirements.

2.2 ENVIRONMENTAL REQUIREMENTS

The recommended environmental specifications for DECSYSTEM-20 systems (including KS10 systems) are listed in Table 2-1. The environmental specifications for individual KS10 system components are given on data sheets in Subsection 2.5. Heat dissipation and air flow rate of internal fans are also given. To estimate cooling and other environmental requirements, refer to Subsection 1.6 of the DECSYSTEM-20 Site Preparation Guide.

Table 2-1 Recommended KS10 System Environmental Specifications

PARAMETER	SPECIFICATION
Temperature	18 C to 24 C (65 F to 75 F)
Humidity	40% to 60%
Temperature Rate of Change	2 degrees C/hr (3.6 degrees F/hr)
Humidity Rate of Change	2%/hr
Voltage Tolerance	120/208 V ± 10% for single phase/three phase (60HZ) 240/380 V ± 10% for single phase/three phase (50HZ)
Frequency Tolerance	60 HZ ± 1 HZ 50 HZ ± 1 HZ

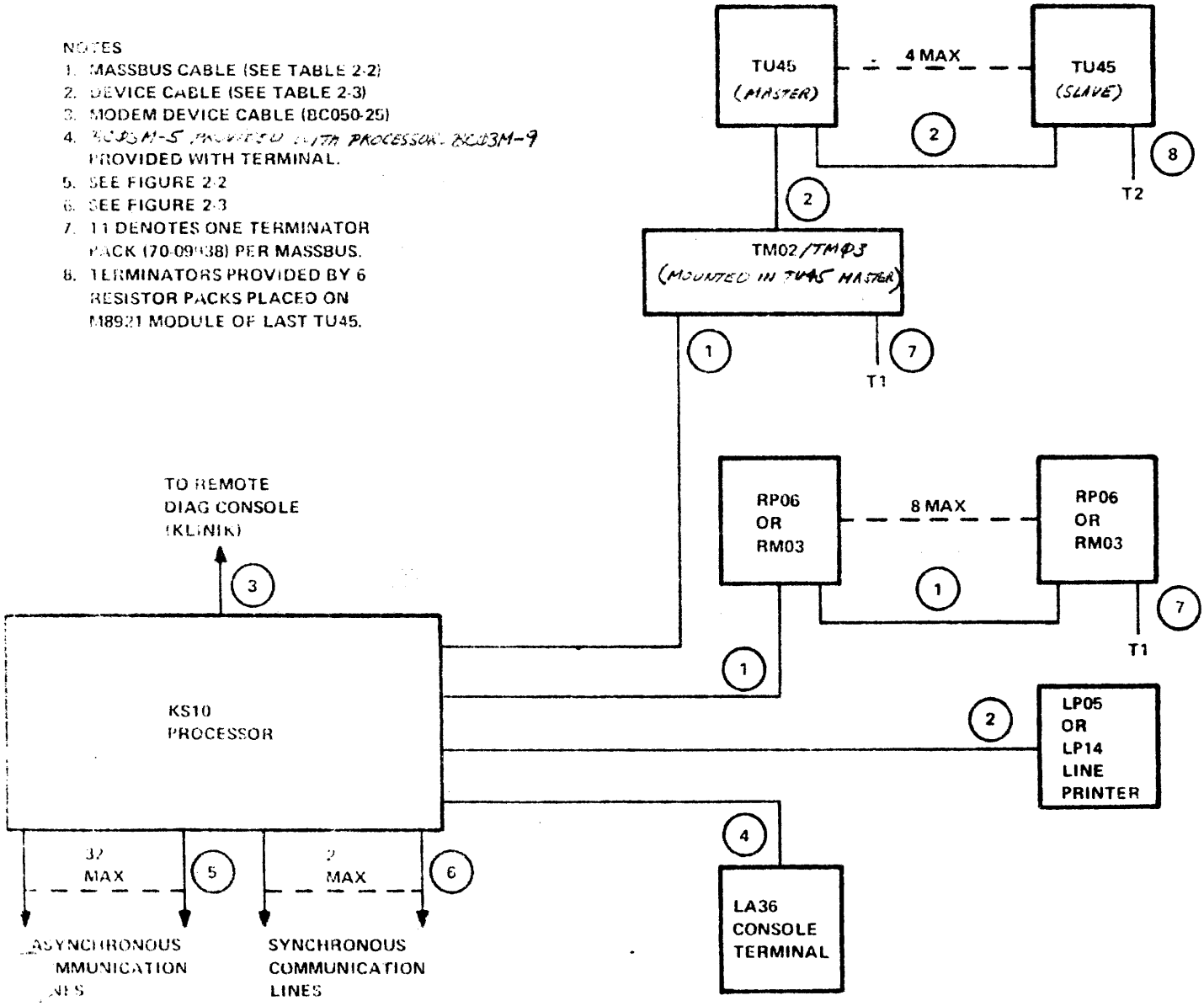
NOTE

Compliance to the environmental specifications above may be required if the system is under a DEC Maintenance Agreement.

2.3 SYSTEM CONFIGURATION

Figure 2-1 shows a typical KS10 system configuration. Reference is made on the figure to Tables 2-2 and 2-3, which provide MASSBUS and device cable data, and to Figure 2-2 and 2-3, which show interconnections of the asynchronous and synchronous communications lines.

NOTES
1. MASSBUS CABLE (SEE TABLE 2-2)
2. DEVICE CABLE (SEE TABLE 2-3)
3. MODEM DEVICE CABLE (BC050-25)
4. BC03M-5 PROVIDED WITH PROCESSOR. BC03M-9 PROVIDED WITH TERMINAL.
5. SEE FIGURE 2-2
6. SEE FIGURE 2-3
7. T1 DENOTES ONE TERMINATOR PACK (70-09038) PER MASSBUS.
8. TERMINATORS PROVIDED BY 6 RESISTOR PACKS PLACED ON M8921 MODULE OF LAST TU45.



MR 0850

Figure 2-1 Typical KS10 System Configuration

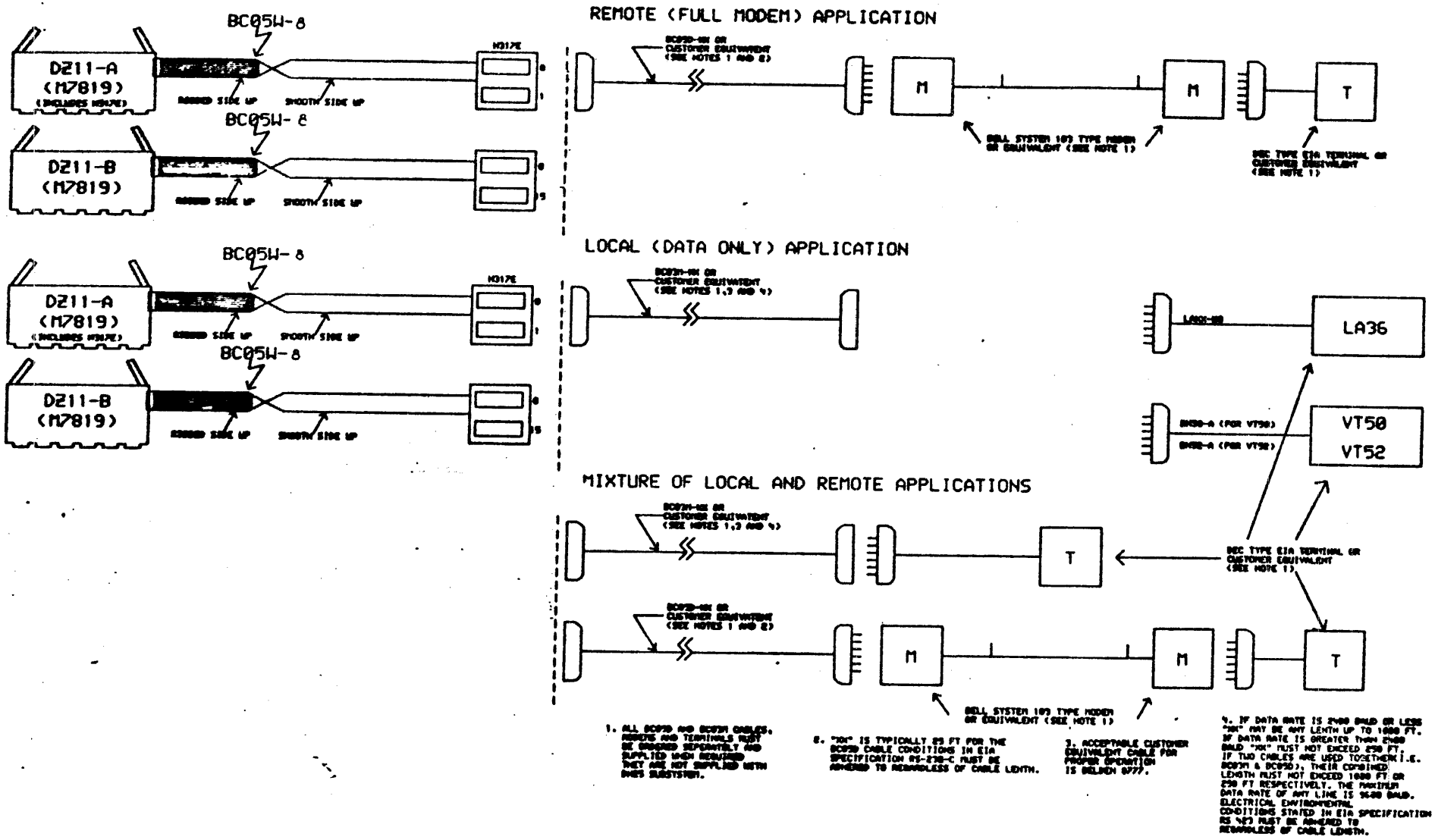


Figure 2-2 KS10 Asynchronous Communications Lines

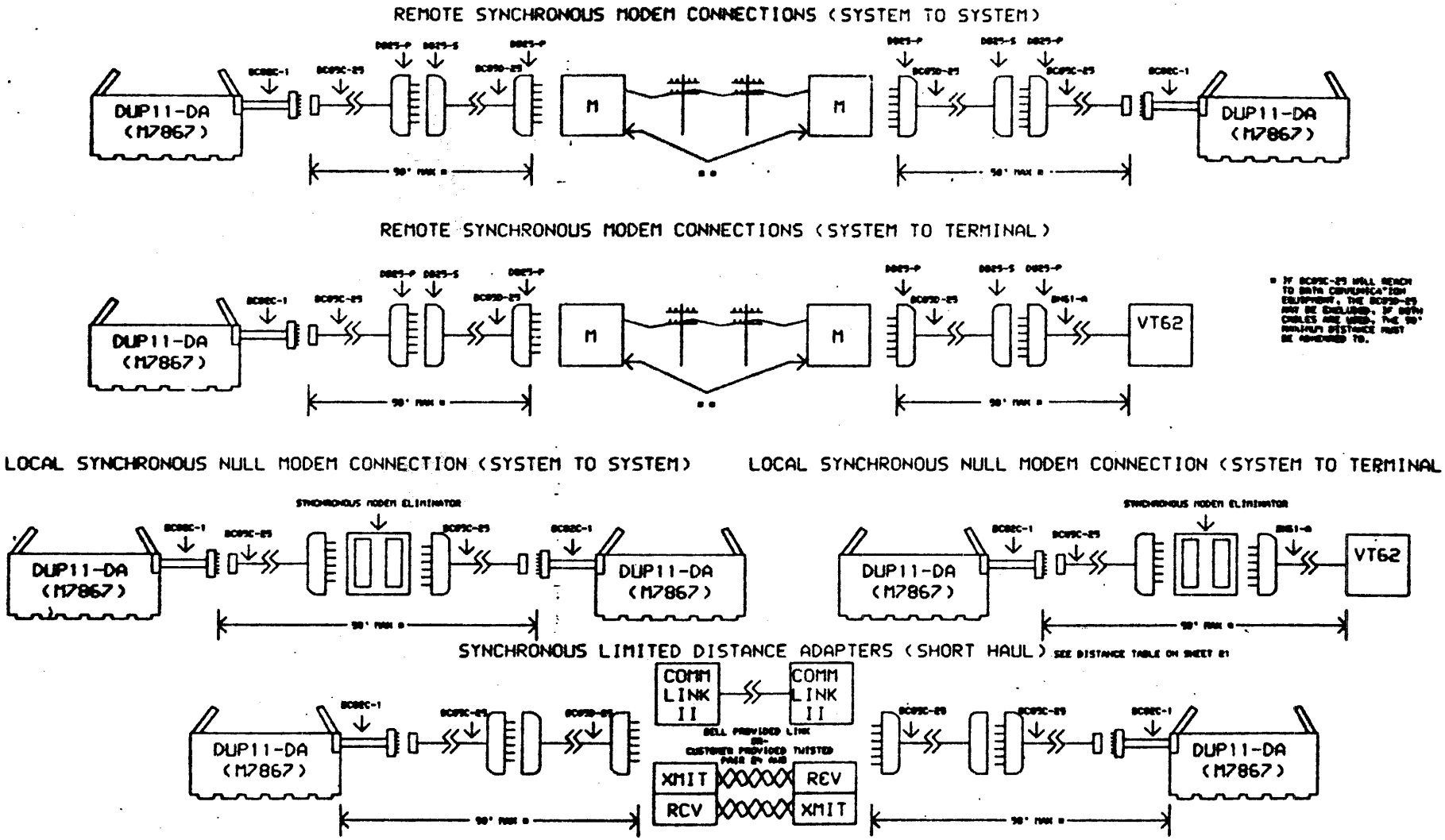


Figure 2-3 KS10 Synchronous Communications Lines

Table 2-2 MASSBUS Cabling

FROM	TO	CABLE	AVAILABLE LENGTH METERS	FEET
CPU	RP06	BC06S (AMP ZIF to AMP ZIF)	4.5	15
CPU	RM03	BC06S (AMP ZIF to AMP ZIF)	7.5	25
RP06	RP06	BC06S (AMP ZIF to AMP ZIF)	0.6/0.75	2/2.5
RM03	RM03	BC06S (AMP ZIF to AMP ZIF)	4.5	15
RP06	RM03	BC06S (AMP ZIF to AMP ZIF)	3	1φ
CPU	TM02/TM03	BC06S (AMP ZIF to AMP ZIF)	4.5	15

Table 2-3 Device Cabling

FROM	TO	CABLE	AVAILABLE LENGTH METERS	FEET
CPU	LP05/LP14	7011426 (AMP ZIF to Winchester)	7.5	25*
			30	100
TM02/TM03	TU45	BC06R (BERG to BERG-cabled internally)	3	10
TU45	TU45	BC06R (BERG to BERG)	1.8	6

NOTE

An asterisk (*) denotes the standard length that will be provided if no cable information is provided 60 days prior to scheduled shipment.

2.4 PRIMARY POWER (AC)

Primary power specifications for KS10 system components are provided on data sheets in Subsection 2-5. Refer to Section 1 (Subsections 1-7 - 1-8) of the DECSYSTEM-20 Site Preparation Guide for the following information:

1. Definition of data sheet parameters (surge current, leakage current, etc.)

2. Description of power regulation systems.

3. Phase balancing, grounding, and service outlet requirements.

4. Description of receptacles and plugs specified (on data sheets) for KS10 system components.

2.5 OPTION DATA SHEETS

This subsection contains option data sheets for the various KS10 system components. The data sheets are arranged in alphanumeric sequence by device designations as follows:

1. LA36
2. LP05
3. LP14
4. KS10-AA/AB Processor
5. RM03
6. RP06
7. TU45A (Master)
8. TU45A (Slave)

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
VE	46.4 kg 102 lbs	85 cm 33.5 in	70 cm 27.5 in	61 cm 24 in	VE	86.4 cm X 128.3 cm 34" X 50-1/2"

POWER (AC)

AC Voltage			Frequency Tolerance	Phase(s)	Steady State Current (RMS)	Surge Current	Surge Duration
Low	Nom	High					
90 180	115 230	132 264	60 Hz ± 1 50 Hz ± 1	1 1	3 Amps 1.5 Amps	60 Amps 30 Amps	1/2 cycle 1/2 cycle

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
	309 kg cal/hr 1230 Btu/hr	360	0.35	2.4 m 8 ft	NEMA = L5 30P	0.107 mA

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume Inlet
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	
~~15° to 32° C~~ ~~10°~~ to ~~40°~~ C ~~50°~~ to ~~104°~~ F 5° to 90° F	-40° to 66° C -40° to 151° F	~~10 - 90%~~ 20-80%	0 - 95%	7° C/hr 12° F/hr	2%/hr	100 CFM

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
15° to 32° C 59° to 90° F	15° to 32° C 59° to 90° F	20 - 80%	20 - 80%		

MAXIMUM CABLE LENGTH AND TYPE(S)

Memory	I/O Bus	Massbus	Device	Other
N/A	N/A	N/A	3 m 9 ft	N/A

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
VE	155 kg 340 lbs	112 cm 44.5 in	84 cm 33 in	66 cm 26 in	VE	33" X 40-1/2"

POWER (AC)

AC Voltage			Frequency Tolerance	Phase(s)	Steady State Current (RMS)	Surge Current	Surge Duration
Low	Nom	High					
90 180	115 230	137 265	60 Hz ± 1 50 Hz ± 1	1 1	4.5 Amps 2.3 Amps	10 Amps 5 Amps	1/2 cycle 1/2 cycle

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord - Conn Type	Leakage Current (Max)
	416 kg cal/hr 1800 Btu/hr	525	0.525	4.0 m 13 ft	NEMA = L5 15P	.551 mA

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume Inlet
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	
10° to 38° C 50° to 100° F	-18° to 60° C 0° to 150° F	30 - 90%	5 - 95%	7 C/hr 12° F/hr	2%/hr	300 CFM

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
24° C 75° F	24° C 75° F	45%	45%		

MAXIMUM CABLE LENGTH AND TYPE(S)

Memory	I/O Bus	Massbus	Device	Other
N/A	N/A	N/A	30 m 100 ft	N/A

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
VE	198 kg 435 lb	112 cm 45 in	84 cm 33 in	70 cm 23 in	VE	86.4 cm X 128.3 cm 34" X 50-1/2"

POWER (AC)

AC Voltage			Frequency Tolerance	Phase(s)	Steady State Current (RMS)	Surge Current	Surge Duration
Low	Nom	High					
100 200	115 230	125 240	60 Hz ± 1 50 Hz ± 1	1 1	7 A 3.5 A	140 A 70 A	1/2 cycle 1/2 cycle

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
5 ms	710 kg cal/hr ~~2830~~ 2815 Btu/hr	825	~~0.825~~ 0.81	3.7 m 12 ft	NEMA = L5-15P	0.394 mA

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume Inlet
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	
10° to 38° C 50° to 100° F	0° to 50° C -18° to 66° F	30-90%	0-95%	7° C/hr 12° F/hr	2%/hr	300 cfm

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
24° C 75° F	24° C 75° F	45%	45%		

MAXIMUM CABLE LENGTH AND TYPE(S)

Memory	I/O Bus	Massbus	Device	Other
N/A	N/A	N/A	30 m 100 ft	N/A

KS10-AA/AB

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
FS	262 kg 590 lbs	152 cm 60 in	69 cm 27 in	76 cm 30 in	H9502H-7	N/A

POWER (AC)

AC Voltage			Frequency Tolerance	Phase(s)	Steady State Current (RMS)	Surge Current	Surge Duration
Low	Nom	High					
104 207	115 230	126 253	60 Hz ± 1 50 Hz ± 1	1 1	9.90 Amps 4.95 Amps	25 Amps 12.5 Amps	6 cycles 6 cycles

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
16 ms	920 kg cal/hr 3648 BTU/hr	1070	1.14	4.5 m 15 ft	NEMA # L5-30P	4.93 ma

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume Inlet
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	
15 to 32C 59 to 90F	-40 to 66C -40 to 151F	20-80%	0-95%	7 C/hr 12 F/hr	2%/hr	1100 CFM

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
N/A	N/A	N/A	N/A	N/A	N/A

MAXIMUM CABLE LENGTH AND TYPE(S)

Memory	I/O Bus	Massbus	Device	Other
N/A	N/A	See Note #4	See Note #4	N/A (Internal) ~~Unibus-See Note~~

RM43

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
FS	193.5 kg 430 lbs	99 cm 39 in	75 cm 29.5 in	83.8 cm 33 in	H9691 (modified)	

POWER (AC)

AC Voltage			Frequency Tolerance	Phase(s)	Steady State Current (RMS)	Surge Current	Surge Duration
Low	Nom	High					
102 213	115 230	128 257	60 Hz ±.6 50 Hz ±.5	1 1	6.4 Amps 3.1 Amps	30 Amps 23 Amps	14 seconds 14 seconds

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
	504 kg cal/hr 2000 BTU/hr	650	0.73	3.7 m 12 ft	NEMA # L5-15P	1.218 ma

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume Inlet
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	
15 to 32C 59 to 90F	-40 to 70C -40 to 158F	20-80%	5-95%	7 C/hr 12 F/hr	2%/hr	

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.

MAXIMUM CABLE LENGTH AND TYPE(S)

Memory	I/O Bus	Massbus	Device	Other
N/A	N/A	48.8 m (total 160 ft system)	N/A	N/A

RPT-

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
[illegible]	275 kg 600 lbs	119 cm 47 in	84 cm 33 in	81 cm 32 in	VE	12-10568-02

POWER (AC)

AC Voltage			Frequency		Steady State	Surge	Surge
Low	Nom	High	Tolerance	Phase(s)	Current (@RMS)	Current	Duration
104 208	115 230	127 254	60 Hz ± 1 50 Hz ± 1	3-wye 3-wye	6 Amps 3 Amps	30 Amps 15 Amps	1/2 cycle 1/2 cycle

POWER (AC)

Interrupt [illegible] Max	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
[illegible]	[illegible]	[illegible]	2.16	[illegible]	NEMA 5-15P 120-240V	4.36 ma

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	Inlet
15 to 32C 59 to 90F	5 to 45 C 40 to 113F	20%-80%	10%-90%	7 C/hr 12 F/hr	2%/hr	550 CFM

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
1 to 32 C 59 to 90F	5 to 45 C 40 to 110F	20-80%	20%-80%	7 C/hr	

MAXIMUM CABLE LENGTH AND TYPE(S)

Ethernet	I/O Bus	Masbus	Device	Other
[illegible]	~~[illegible]~~ N/A	~~[illegible]~~ 4000 m (Optic 16404 ft system)	~~[illegible]~~ N/A	N/A

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
FS	290 kg 640 lb	152 cm 60 in	69 cm 27 in	76 cm 30 in	H9502	N/A

POWER (AC)

AC Voltage			Frequency Tolerance	Phase(s)	Steady State Current (RMS)	Surge Current	Surge Duration
Low	Nom	High					
103 207	115 230	126 263	60 Hz ± 1 50 Hz ± 1	1 1	6.8 Amps 3.4 Amps	14 Amps 7 Amps	1/2 cycle 1/2 cycle

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
5 ms	~~640~~ 2560 kg cal/hr ~~2560~~ 10240 Btu/hr	750	0.78	4.5 m 15 ft	NEMA = L5 30P	3.16 mA

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume Inlet
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	
16° to 32° C 60° to 90° F	-40° to 60° C -40° to 140° F	20-80%	5-95%	7° C/hr 12° F/hr	N/A	500 cfm

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
16° to 32° C 60° to 90° F	-40° to 60° C -40° to 140° F	20-80%	5-95%	N/A	N/A

MAXIMUM CABLE LENGTH AND TYPE(S)

Memory	I/O Bus	Massbus	Device	Other
N/A	N/A	30 m 100 ft	1.8 m 6 ft	N/A

(G) BC06S

MECHANICAL

Mounting Code	Weight	Height	Width	Depth	Cab Type If Used	Skid Type
FS	272 kg 600 lb	152 cm 60 in	69 cm 27 in	76 cm 30 in	H9502	N/A

POWER (AC)

AC Voltage			Frequency Tolerance	Phase(s)	Steady State Current (RMS)	Surge Current	Surge Duration
Low	Nom	High					
103 207	115 230	126 263	60 Hz ± 1 50 Hz ± 1	1 1	4.6 Amps 2.3 Amps	9 Amps 5 Amps	1/2 cycle 1/2 cycle

POWER (AC)

Interrupt Tolerance (Max)	Heat Dissipation	Watts	KVA	PWR Cord Length	PWR Cord Conn Type	Leakage Current (Max)
N/A	385 kg cal/hr 1540 Btu/hr	450	0.53	4.5 m 15 ft	NEMA = L5-30P	3.16 mA

ENVIRONMENTAL (DEVICE)

Temperature		Relative Humidity		Rate of Change		Air Volume Inlet
Operating	Storage	Operating	Storage	Temp	Rel. Humid.	
16° to 32° C 60° to 90° F	-40° to 60° C -40° to 140° F	20-80%	5-95%	7° C/hr 12° F/hr	N/A	500 cfm

ENVIRONMENTAL (MEDIA)

Temperature		Relative Humidity		Rate of Change	
Operating	Storage	Operating	Storage	Temp	Rel. Humid.
16° to 32° C 60° to 90° F	-40° to 60° C -40° to 140° F	20-80%	5-95%	N/A	N/A

MAXIMUM CABLE LENGTH AND TYPE(S)

Memory	I/O Bus	Massbus	Device	Other
N/A	N/A	30 m 100 ft	3 m 10 ft	N/A

(3) BC06R

To be supplied.

SECTION 4

4.1 CONTROLS AND INDICATORS

The KS10 switch and indicator panel is shown in Figure 4-1. There are five switches, three of which provide for powering-up, resetting, and bootstrapping the system. The fourth switch serves as an interlock to prevent an inadvertent reset or bootstrap by the operator once the system is in operation. The last switch controls the remote diagnosis link to the system. Switch functions are listed in Table 4-1.

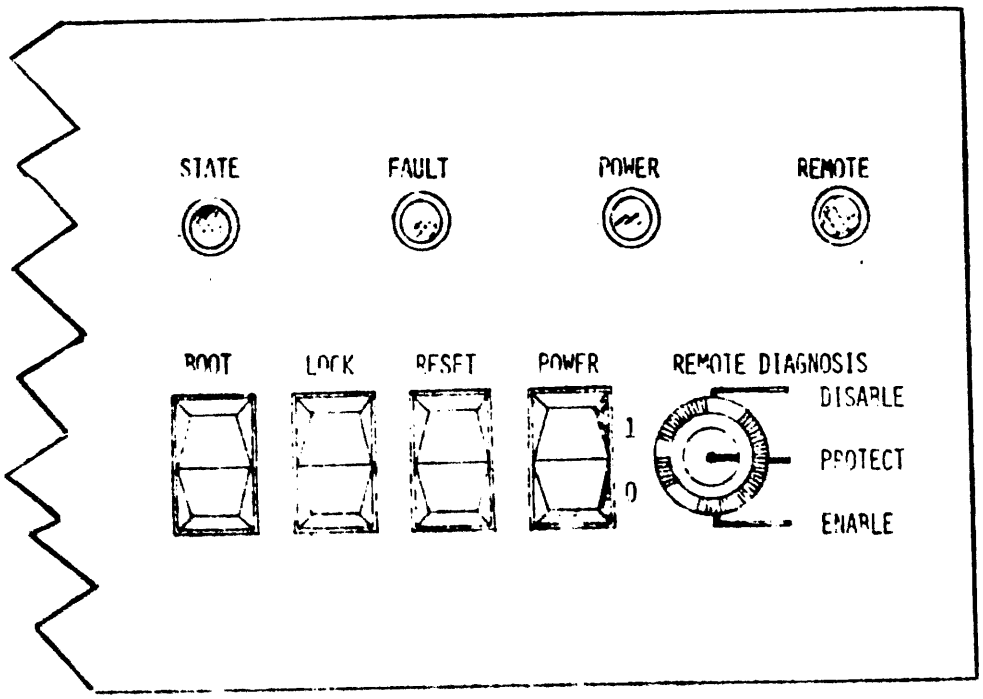


Figure 4-1 KS10 Switch and Indicator Panel

Table 4-1 KS10 Switch Functions

Switch	Function
POWER	Turns ac power on/off. (Causes 861 power control to apply/remove line power to the CPU and BA11K power supplies.)
RESET	Resets all KS10 System components (including the 8080 console hardware)
BOOT	Bootstraps the system. Performs same function as BT console command.
LOCK	Electrically interlocks the RESET and BOOT switches so that they have no effect. Also prevents the operator from switching the CTY from user mode to console mode (disables control-backslash command).
REMOTE DIAGNOSIS	Three position key operated switch that controls access by the remote diagnosis (KLINIK) line. DISABLE position - Prevents access to the system. PROTECT position - Allow access to the system with password. ENABLE position - Allows free access to the system without password protection.

The panel also has four indicators. One indicates power-on. The other three, which are under control of the 8080 console program, indicate the system's run state, when a system fault has been detected, and when the system's remote diagnosis line is enabled. Indicator functions are detailed in Table 4-2.

Table 4-2 KS10 Indicator Functions

Indicator	Function
POWER	Lights when dc power (-5 V and +12 V) is on.
REMOTE	Lights when KLINIK line is enabled with or without password. (REMOTE DIAGNOSIS switch in ENABLE or PROTECT position.)
FAULT	Lights for the following conditions: 1. KS10 bus parity error 2. UBA parity error 3. Memory parity error 4. Data path parity error 5. Console parity error 6. CRA parity error 7. CRM parity error 8. Memory refresh error 9. Boot command fails to start machine.
STATE	Lights when KS10 microcode is loaded and running. Indicator blinks (1 second on, 1 second off) when system monitor has been loaded and is maintaining "keep-alive" dialogue with console.

4.2 KS10 DIFFERENCES (KS10 vs KL10)

For the purpose of this document, some aspects of KS10 operation and programming are best described in relation to the KL10. There are only a few differences in the instruction set and the KS10 uses the same operating system as the KL10 (TOPS20) with minor modifications. The following differences between the KS10 and KL10 do exist, however.

4.2.1 Public Mode

Because TOPS20 does not support public mode, it has not been implemented in the KS10. All instructions behave as if the machine is in concealed mode.

4.2.2 Addressing

Only section 0 addressing is implemented in the KS10; that is, like the KL10-B (KL10-PA processor), virtual memory space is 256 K words. Extended addressing, 32 sections of 256 K words as implemented by the KL10-E (KL10-PV processor), is not currently supported by the KS10. It does support the model B instructions XJRSTF, XJEN, XPCW, and SFM.

4.2.3 Interrupt Handling

KS10 Priority interrupt operation is the same as the KL10-B (KL10-PA processor) except for the following:

1. Only the JSR or XPCW instruction is allowed as an interrupt instruction; that is, as the first instruction executed as a result of an interrupt. Any other instruction will halt the processor (Subsection 4.4).

2. The only interrupt function implemented for devices is the dispatch function (i.e., interrupt vector). Unlike the KL10-B, which dispatches to and executes the instrucion in the EPT location specified by the vector address, the KS10 first references an EPT location determined by the UBA number (EPT + 100 + CONTROLLER #). It then uses this word as an exec-virtual address of a table and executes the instruction at TABLE + VECTOR/4.

SHIFT RIGHT 2 BITS

3. The KS10 implements two levels of PIA for I/O (Unibus) devices, one PIA can have a higher priority than the other. The PI level (1-7) assigned to Unibus devices interrupting on BR levels 7 and 6 is set by loading a high level PIA (bits 30-32 of UBA status register). The PI level (1-7) for devices interrupting on BR levels 5 and 4 is set by loading a low level PIA (bits 33-35 of UBA status register).

Table 4-3 lists the hard-wired interrupt vectors and BR levels for the various KS10 I/O (Unibus) devices in a fully configured system. It also indicates which PIA, high or low level, is associated with each device.

Table 4-3 I/O (Unibus) Device Vectors and BR Levels

Device	UBA #	PIA	Interrupt Vector	BR
RH11 #1 (RP06)	1	HI	254	6
RH11 #3 (TU45)	3	HI	224	6
LP20 #1	3	LO	750	4
DZ11 #1	3	LO	340	5
DZ11 #2	3	LO	350	5
DZ11 #3	3	LO	360	5
DZ11 #4	3	LO	370	5
KMC11 #1	3	LO	540	5
DUP11 #1	3	LO	570	5
DUP11 #2	3	LO	600	5

4.2.4 Paging

Both TOPS10 and TOPS20 paging are implemented in the KS10. The paging mode is selected by bit 21 in the WREBR instruction (Subsection 4.3.1.9).

4.2.5 KS10 Instruction Set

The KS10 has the same instruction set as the KL10-B (i.e., Model PA Processor-section 0 addressing only) except for the following:

1. The single-precision (without rounding) floating point instructions that facilitate software double-precision operations are <u>not</u> supported on the KS10 and will trap as MUUOs. These are:

   a. UFA (Unnormalized Floating Add)
   b. DFN (Double Floating Negate)
   c. FADL (Floating Add Long)
   d. FSBL (Floating Subtract Long)
   e. FMPL (Floating Multiply Long)
   f. FDVL (Floating Divide Long)

2. The KS10 checks several MBZ (must be zero) fields in the extended instrucion set that are not checked by the KL10-B. Any non zero fields cause an MUUO trap.

3. In KI paging mode, if a MAP instruction is done to a page with A = 0 in the page table entry, the KS10 returns the address it was given. The KL10 returns zero as an address.

4. All KL10 I/O instructions have been replaced by a new I/O instruction set for the KS10. Because the KS10 I/O

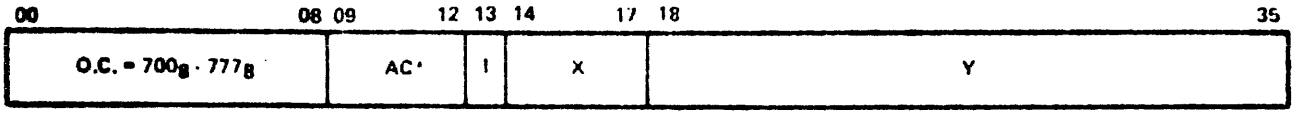
instructions do not specify a device code, instruction format has been changed to conform to the basic instruction format of opcode, AC, and effective address. The KS10 I/O instructions are described in Subsection 4.3.

NOTE

Appendix A shows KS10 word formats and lists all KS10 instructions alphabetically (by mnemonic) and numerically (by opcode). An algebraic representation of the function performed by each instruction is also given.

4.3 KS10 I/O INSTRUCTIONS

KS10 I/O instructions have the same basic format as the rest of the KS10 instruction set. (Format is shown in Figure 4-2.) I/O instruction op-codes are in the range 700 - 777 (octal). Op-code assignments are shown in Table 4-4



O.C. = OP-CODE (TABLE 4-4)
AC = ACCUMULATOR
I = INDIRECT ADDRESSING BIT
X = INDEX REGISTER
Y = ADDRESS

* NOTE: AC FIELD USED AS OP-CODE EXTENSION FOR APR I/O INSTRUCTIONS (TABLE 4-5).

MR 0228

Figure 4-2 I/O Instruction Format

Table 4-4 I/O Instruction Op Codes (Octal)

	0	1	2	3	4	5	6	7
700	APR0	APR1	APR2	-	UMOVE	UMOVEM		-
710	TIOE	TION	RDIO	WRIO	BSIO	BCIO	-	-
720	TIOEB	TIONB	RDIOB	WRIOB	BSIOB	BCIOB	-	
730	-	-	-	-	-	-	-	-
740	-	-	-	-	-	-	-	-
750	-	-	-	-	-	-	-	-
760	-	-	-	-	-	-	-	-
770	-	-	-	-	-	-	-	-

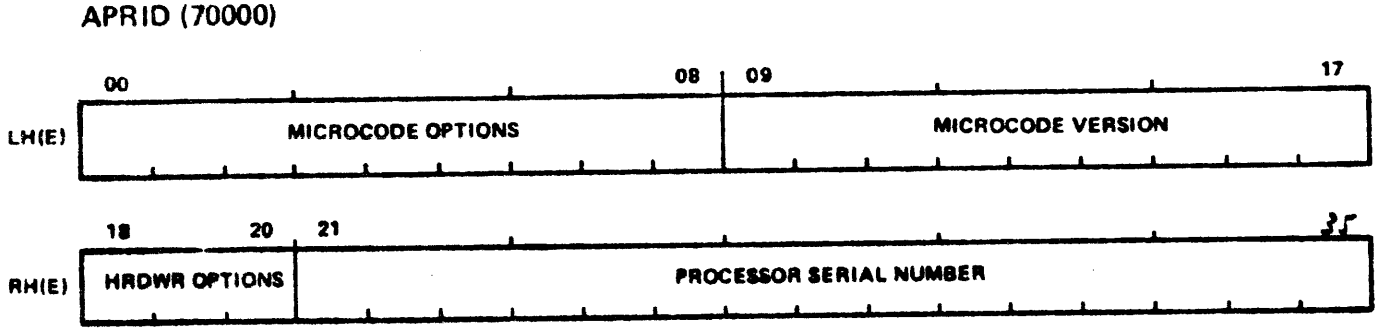
4.3.1 Internal (APR) I/O Instructions

The internal I/O instructions (APR0-2; op codes 700 - 702) use the AC field as an extension of the op code as indicated in Table 4-5. For example, the RDEBR instruction (an APR instruction with op code = 701) is specified by an AC value of 24 . Function and bit format for the various KS10 internal I/O instructions are given in Subsections 4.3.1.1 - 4.3.1.22. Any similarities to KL10 I/O instructions are noted.

Table 4-5 AC Field Assignments (Octal) for APR I/O Instructions

AC	700	701	702
00	APRID	-	RDSPB
04	-	RDUBR	RDCSB
10	-	CLRPT	RDPUR
14	-	WRUBR	RDCSTM
20	WRAPR	WREBR	RDTIME
24	RDAPR	RDEBR	RDINT
30	-	-	RDHSB
34	-	-	-
40	-	-	WRSPB
44	-	-	WRCSB
50	-	-	WRPUR
54	-	-	WRCSTM
60	WRPI	-	WRTIME
64	RDPI	-	WRINT
70	-	-	WRHSB
74	-	-	-

4.3.1.1 **APRID (70000)** - The APRID instruction, similar in function to the APRID instruction for the KL10, reads the KS10 microcode version number and CPU serial number. The information is stored in E. Bit format is shown in Figure 4-3.



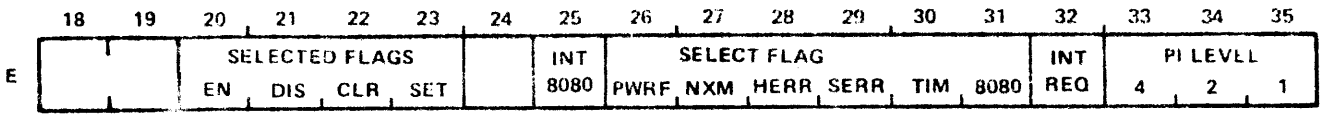
BIT(S)	FUNCTION
0-8	RESERVED FOR MICROCODE VERSION
9-17	MICROCODE VERSION NUMBER
18-20	HARDWARE OPTIONS (BITS CURRENTLY = 0)
21-35	PROCESSOR SERIAL NUMBER.

MR-0229

Figure 4-3 APRID Instruction

4.3.1.2 **WRAPR (70020)** - WRAPR is an immediate mode instruction used to control the processor. It is analogous to the CONO APR instruction used in the KL10. Bit format is shown in Figure 4-4.

WRAPR (70020)

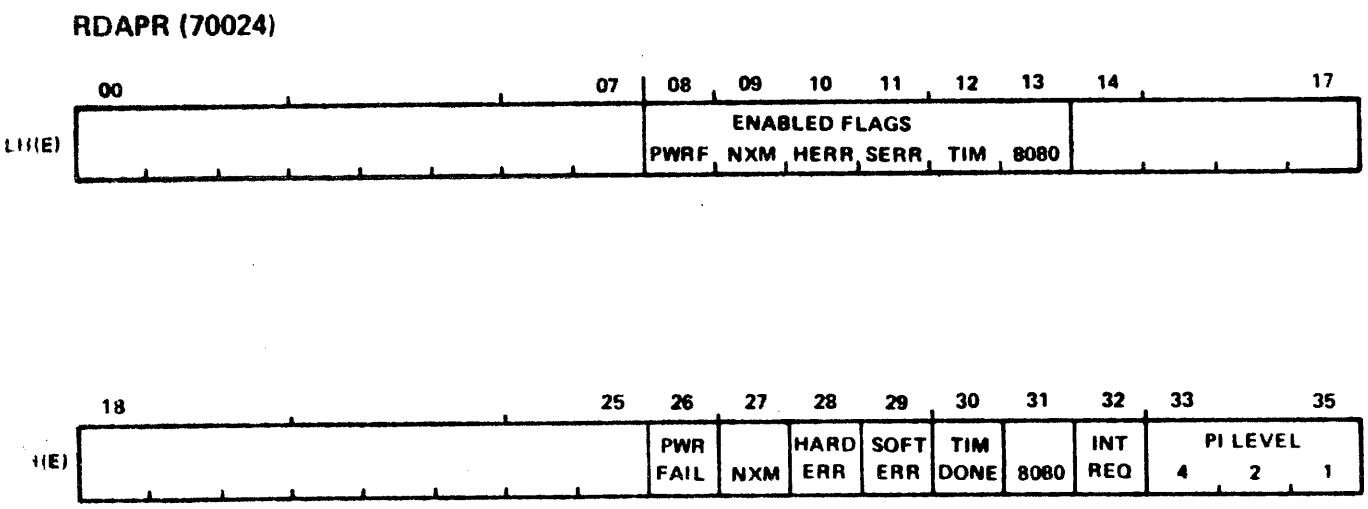


BIT(S)	FUNCTION
20	ENABLE CONDITIONS SELECTED BY BITS 26-31 TO CAUSE INTERRUPTS.
21	DISABLE INTERRUPTS FOR CONDITIONS SELECTED BY BITS 26-31.
22	CLEAR FLAGS INDICATED BY BITS 26-31.
23	SET FLAGS INDICATED BY BITS 26-31.
25	INTERRUPT 8080 CONSOLE
26	POWER FAIL
27	NON-EXISTENT MEMORY ERROR
28	HARD MEMORY ERROR (CANNOT BE CORRECTED BY ECC)
29	SOFT MEMORY ERROR (CORRECT DATA PLACED ON BUS)
30	INTERVAL TIMER
31	8080 CONSOLE
32	GENERATE INTERRUPT REQUEST
33-35	PIA

MR-0230

Figure 4-4 WRAPR Instruction

4.3.1.3 RDAPR (70024) - The RDAPR instruction stores APR status in E. It corresponds to the CONI APR instruction used in the KL10. Bit format is shown in Figure 4-5.



BIT(S)	FUNCTION
08	POWER FAIL ENABLED
09	NON-EXISTENT MEMORY ERROR ENABLED
10	HARD MEMORY ERROR INTERRUPT ENABLED
11	SOFT MEMORY ERROR INTERRUPT ENABLED
12	INTERVAL TIMER ENABLED
13	8080 CONSOLE INTERRUPT ENABLED
26	POWER FAIL ERROR
• 27	NON-EXISTENT MEMORY ERROR
• 28	HARD MEMORY ERROR (CANNOT BE CORRECTED BY ECC).
29	SOFT MEMORY ERR (CORRECT DATA PLACED ON BUS)
30	INTERVAL TIMER DONE
31	8080 CONSOLE INTERRUPT
32	INTERRUPT REQUESTED
33-35	PIA

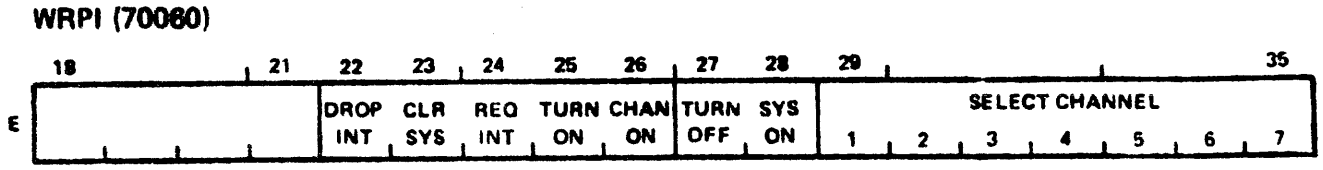
• NOTE:

PAGE FAIL OCCURS IF ERROR IS RESULT OF CPU MEMORY REQUEST. NXM FLAG ALSO SETS IN UNIBUS DEVICE IF ERROR IS RESULT OF UNIBUS NPR REQUEST.

MR-0231

Figure 4-5 RDAPR Instruction

4.3.1.4 WRPI (70060) - The WRPI instruction is identical to the KL10 CONO PI instruction except that bits 18-20 (write even parity) are not implemented. Bit format is shown in Figure 4-6.



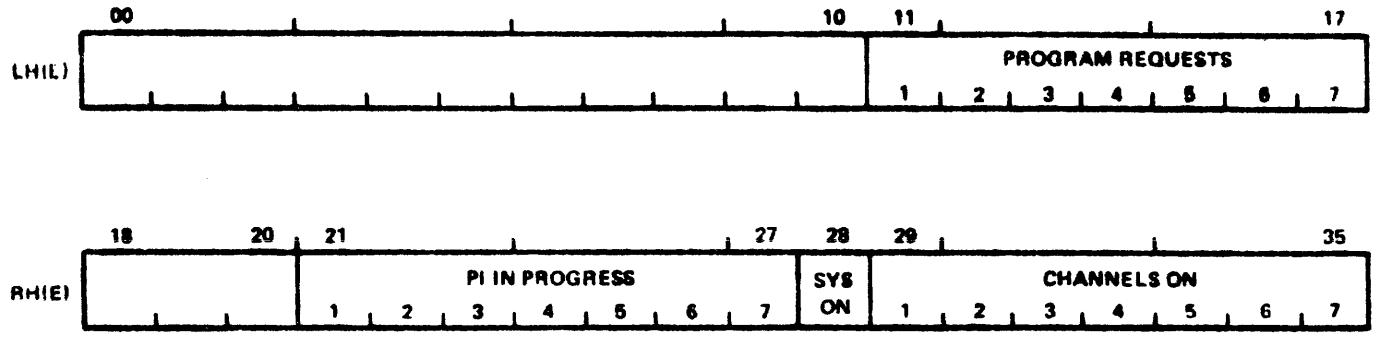
BIT(S)	FUNCTION
22	DROP PROGRAM REQUESTS ON SELECTED CHANNELS.
23	CLEAR PI SYSTEM
24	INITIATE INTERRUPTS ON THE SELECTED CHANNELS.
25	TURN ON THE SELECTED CHANNELS.
26	TURN OFF THE SELECTED CHANNELS.
27	DEACTIVATE THE PI SYSTEM.
28	ACTIVATE THE PI SYSTEM.
29-35	SELECT CHANNELS FOR BITS 22, 24, 25 AND 26.

MR 0232

Figure 4-6 WRPI Instruction

4.3.1.5 RDPI (70064) - The RDPI insstruction is identical to the KL10 CONI PI instruction except that bits 18-20 read no status (write even parity is not implemented). Bit format is shown in Figure 4-7.

RDPI (70064)



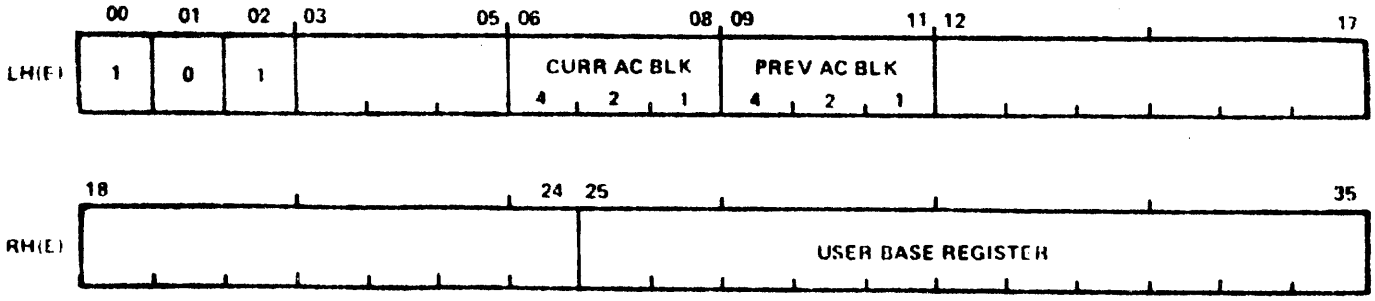
BIT(S)	FUNCTION
11-17	PROGRAM REQUESTS ON CHANNELS 1-7
21-27	INTERRUPTS HOLDING (IN PROGRESS) ON CHANNELS 1-7
28	PI SYSTEM ON
29-35	ACTIVE CHANNELS 1-7

MR-0233

Figure 4-7 RDPI Instruction

4.3.1.6 RDUBR (70104) - The RDUBR instruction, which is similar to the KL10 DATAI PAG instruction, reads the user base register (UBR) and stores the information in E. The word stored is in exactly the same format as used by the WRUBR instruction (Subsection 4.3.1.8). In order to allow the word to be used directly (by a WRUBR), bits 0 and 2 are set to ONE in the result. Bit format is shown in Figure 4-8.

RDUBR (70104)



BIT(S)	FUNCTION
0	1
1	0
2	1
6-8	CURRENT AC BLOCK
9-11	PREVIOUS AC BLOCK
25-35	USER BASE REGISTER

MR-0234

Figure 4-8 RDUBR Instruction

4.3.1.7 CLRPT (70110) - The CLRPT instruction is similar to the KL10 CLRPT instruction. It clears the hardware page table so that the next reference to the word at E will cause a refill cycle. There is only one entry in the page table for any virtual page. Clearing the mapping information for a page clears both the EXEC and USER mapping. Bit format is shown in Figure 4-9.

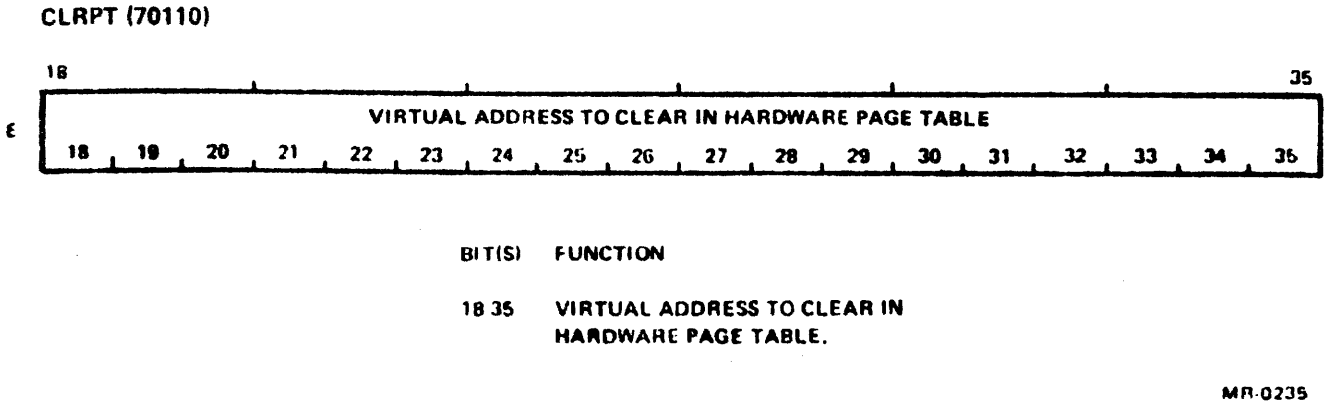
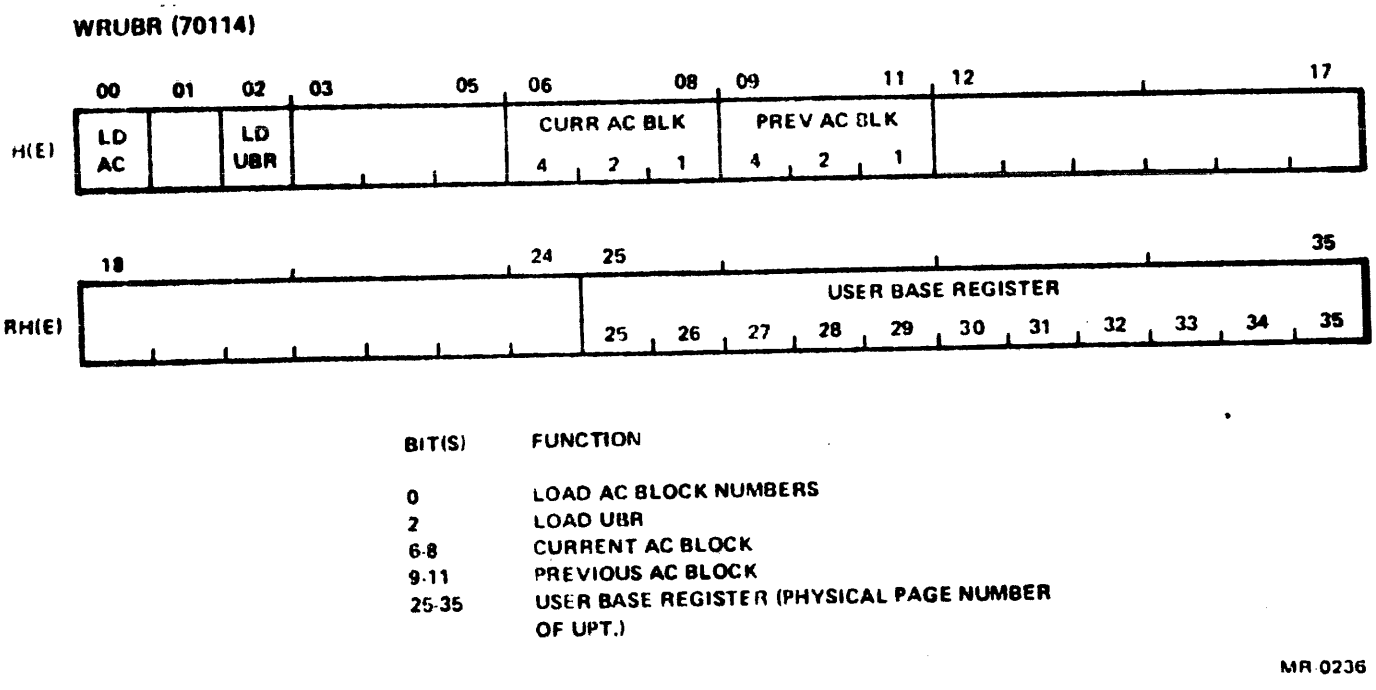


Figure 4-9 CLRPT Instruction

4.3.1.8 WRUBR (70114) - The WRUBR instruction, which is similar to the KL10 DATAO PAG instruction, loads the UBR with the word at E. Bit format is shown in Figure 4-10.

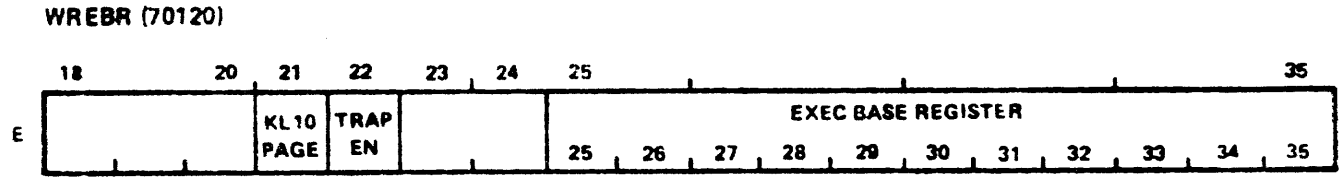


BIT(S)	FUNCTION
0	LOAD AC BLOCK NUMBERS
2	LOAD UBR
6-8	CURRENT AC BLOCK
9-11	PREVIOUS AC BLOCK
25-35	USER BASE REGISTER (PHYSICAL PAGE NUMBER OF UPT.)

MR-0236

Figure 4-10 WRUBR Instruction

4.3.1.9 **WREBR (70120)** - The WREBR instruction loads the executive base register (EBR) from E. It is similar in function to the KL10 CONO PAG instruction. Bit format is shown in Figure 4-11.



BIT(S)	FUNCTION
21	KL10 PAGING MODE
22	TRAP (AND PAGING) ENABLE
25-35	EXECUTIVE BASE REGISTER (PHYSICAL PAGE NUMBER OF EPT)

MR-0237

Figure 4-11 WREBR Instruction

4.3.1.10 RDEBR (70124) - The RDEBR instruction reads the EBR into the right half of E. It is comparable to the KL10 CONI PAG instruction. Bit format is shown in Figure 4-12.

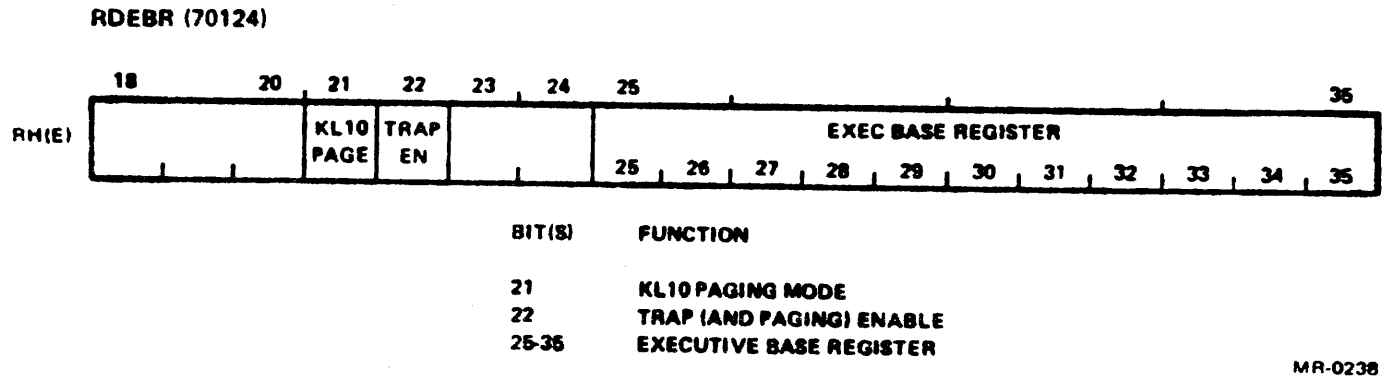


Figure 4-12 RDEBR Instruction

4.3.1.11 RDSPB (70200) - The RDSPB instruction reads the shared pointer table (SPT) base register and stores the value in E. Bit format is shown in Figure 4-13.

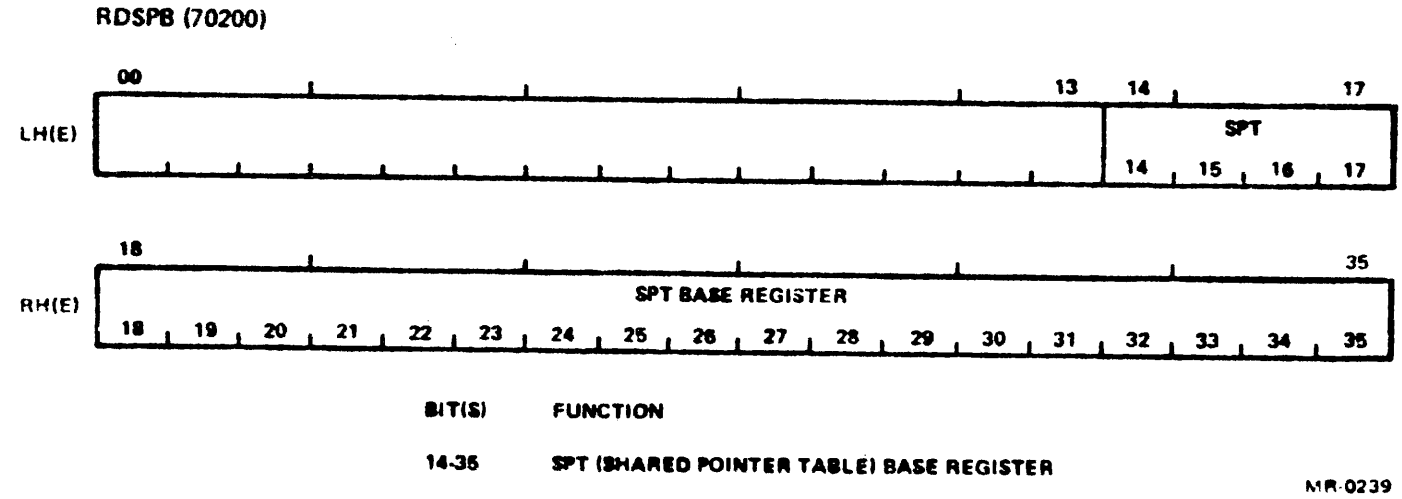


Figure 4-13 RDSPB Instruction

4.3.1.12 RDCSB (70204) - The RDCSB instruction reads the core status table (CST) base register and stores the value in E. Bit format is shown in Figure 4-14.

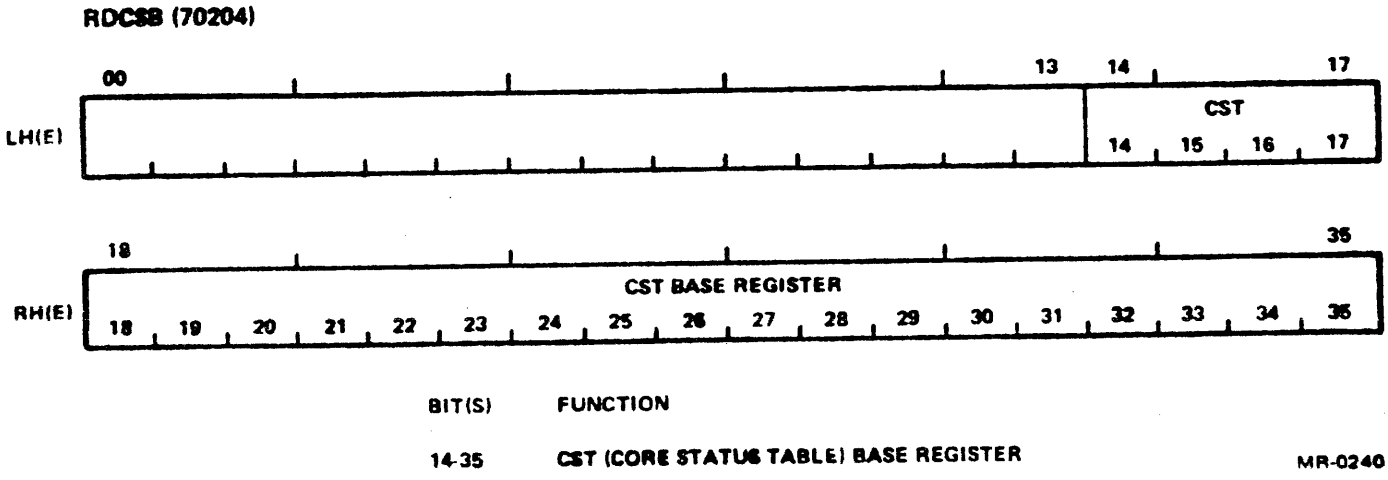


Figure 4-14 RDCSB Instruction

4.3.1.13 RDPUR (70210) - The RDPUR instruction reads the process use register (PUR) and stores the value in E. Bit format is shown in Figure 4-15.

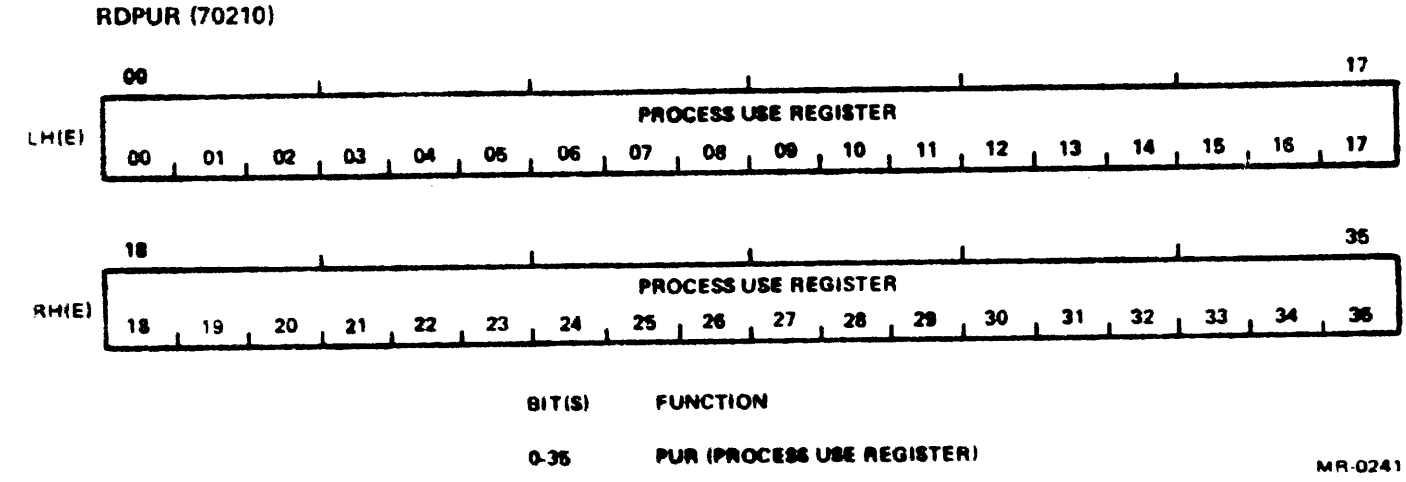


Figure 4-15 RDPUR Instruction

4.3.1.14 RDCSTM (70214) - The RDCSTM instruction reads the CST mask register and stores the value in E. Bit format is shown in Figure 4-16.

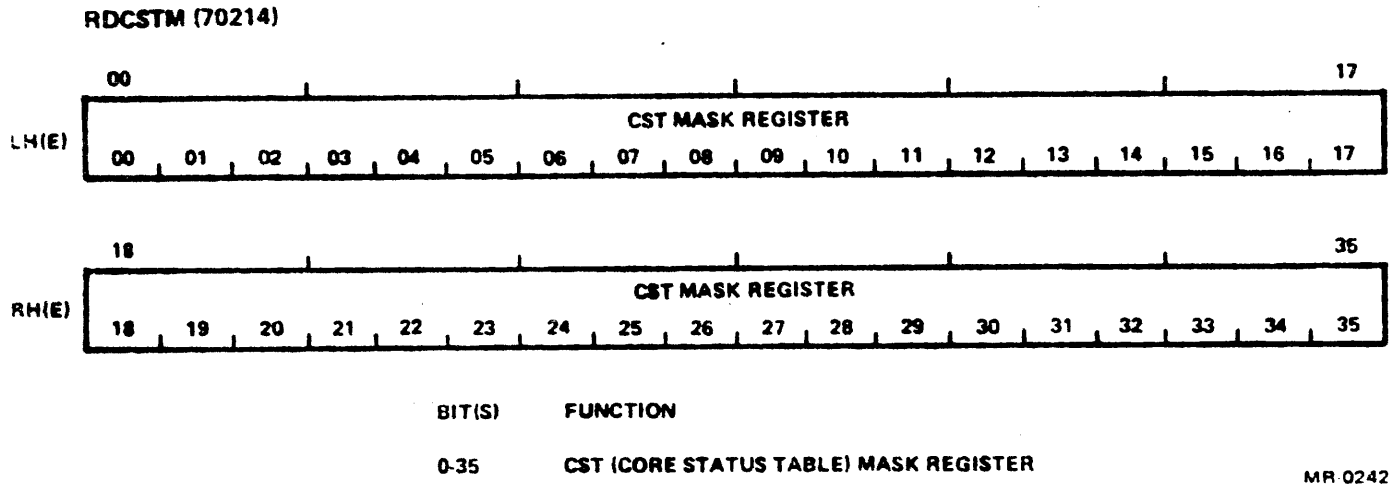
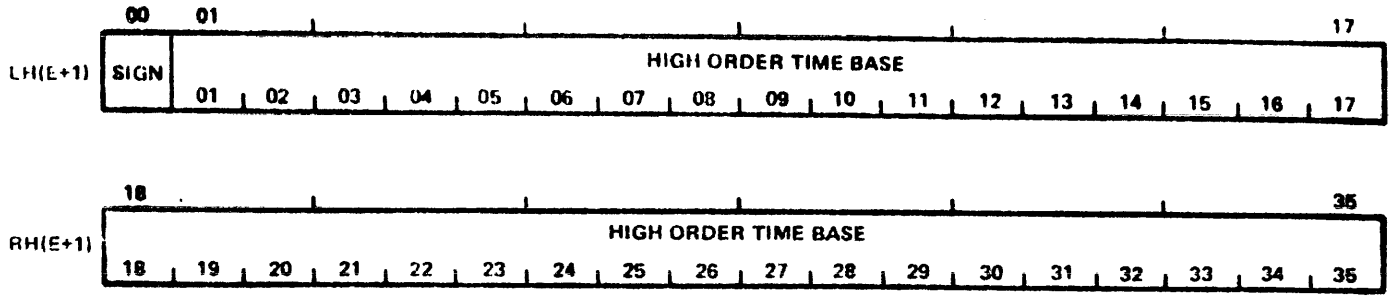


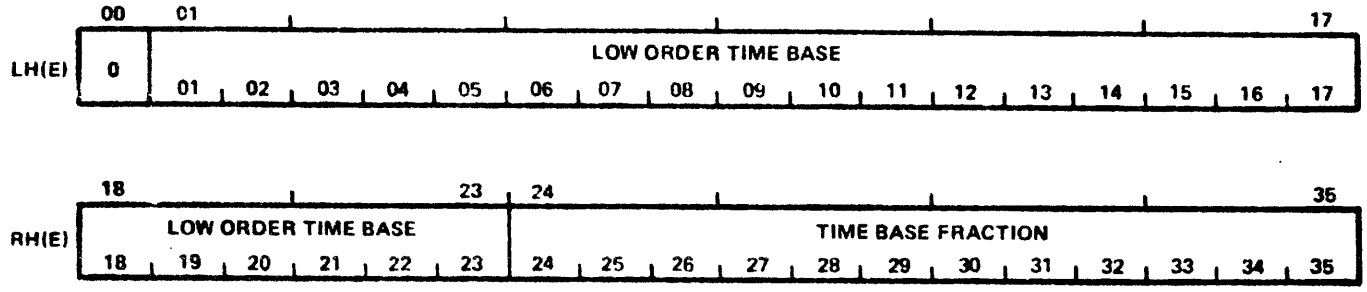
Figure 4-16 RDCSTM Instruction

4.3.1.15 RDTIME (70220) - THE RDTIME instruction is similar to the RDTIME instruction for the KL10. It reads the time base and stores the double-word value in E and E + 1. The time base up-counts at 4.096 mHz. Bit format is shown in Figure 4-17.

RDTIME (70220)



BIT(S)	FUNCTION
0	SIGN BIT: 0 (+), 1 (-)
1-35	HIGH ORDER TIME BASE (MILLISECONDS)



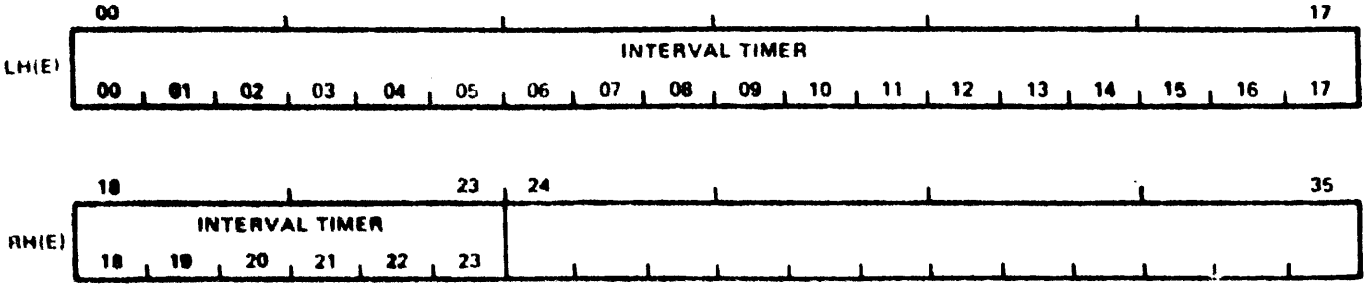
BIT(S)	FUNCTION
0	SIGN BIT: 0 (+), 1 (-)
1-23	LOW ORDER TIME BASE (MILLISECONDS)
24-35	TIME BASE FRACTION

MR-0243

Figure 4-17 RDTIME Instruction

4.3.1.16 RDINT (70224) - The RDINT instruction reads the current value of the interval timer period register and stores the value in E. Bit format is shown in Figure 4-18.

RDINT (70224)



BIT(S)	FUNCTION
0-23	INTERVAL TIMER PERIOD REGISTER (PERIOD = n MILLISECONDS)

MR 0244

Figure 4-18 RDINT Instruction

4.3.1.17 RDHSB (70230) - The RDHSB instruction stores the value of the halt status block address at E. Bit format is shown in Figure 4-19.

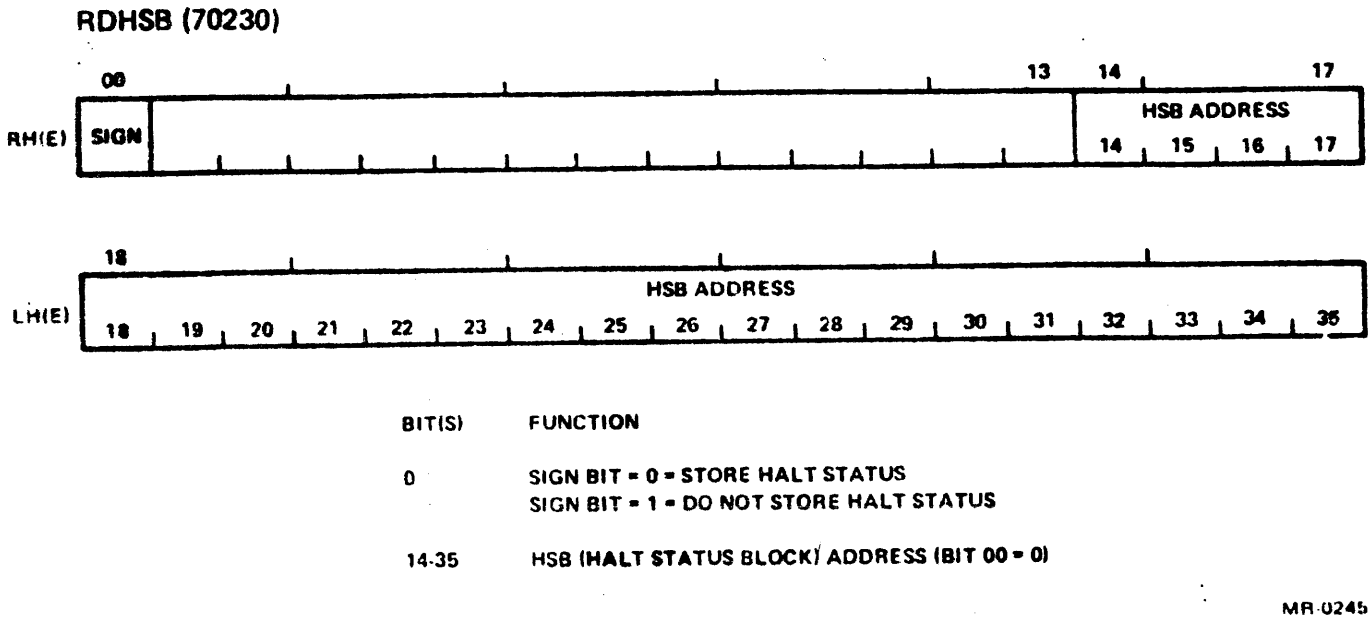


Figure 4-19 RDHSB Instruction

4.3.1.18 WRSPB (70240) - The WRSPB instruction loads the SPT base register from E. Bit format is shown in Figure 4-20.

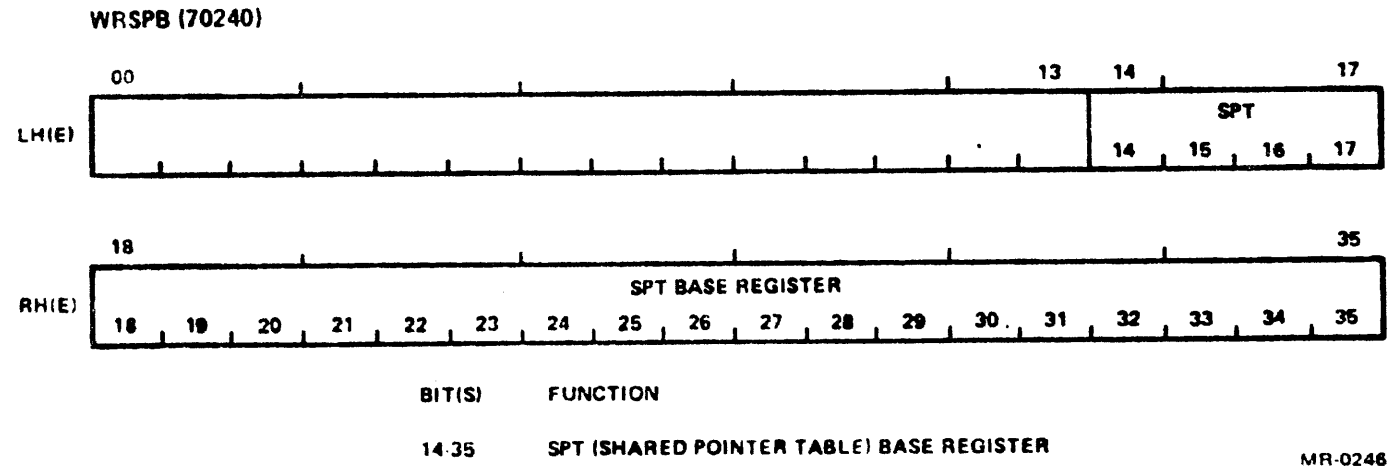


Figure 4-20 WRSPB Instruction

4.3.1.19 **WRCSB (70244)** - The WRCSB instruction loads the CST base register from E. Bit format is shown in Figure 4-21.

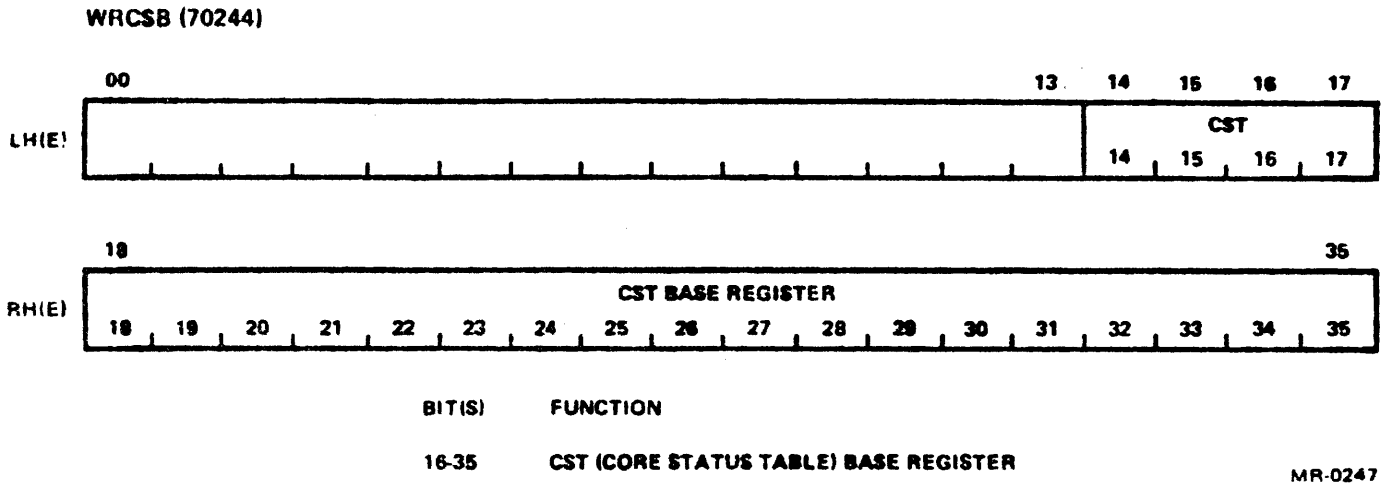


Figure 4-21 WRCSB Instruction

4.3.1.20 **WRPUR (70250)** - The WRPUR instruction loads the PUR from E. (Bit format is shown in Figure 4-22.) The PUR contains the AGER in the left-most bits. These bits are cleared by ANDing the CST entry with the CST mask; then the entire PUR is ORed with the CST entry.

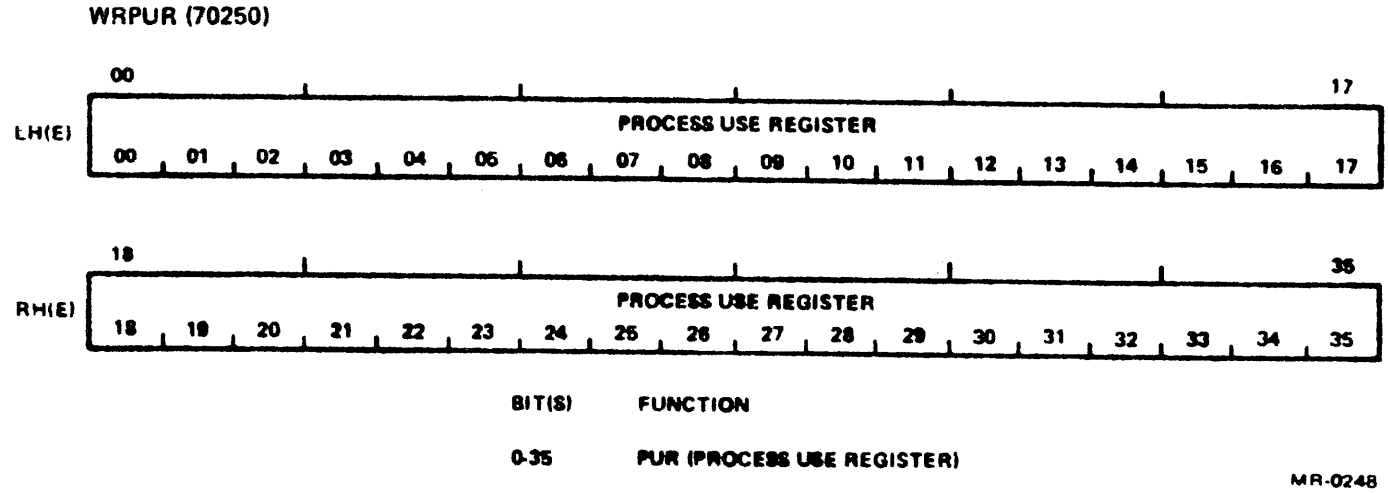


Figure 4-22 WRPUR Instruction

4.3.1.21 WRCSTM (70254) - The WRCSTM instruction loads the CST mask register from E. The CST mask register should contain a zero for every bit in the AGER and a one in all other bit positions. Bit format is shown in Figure 4-23.

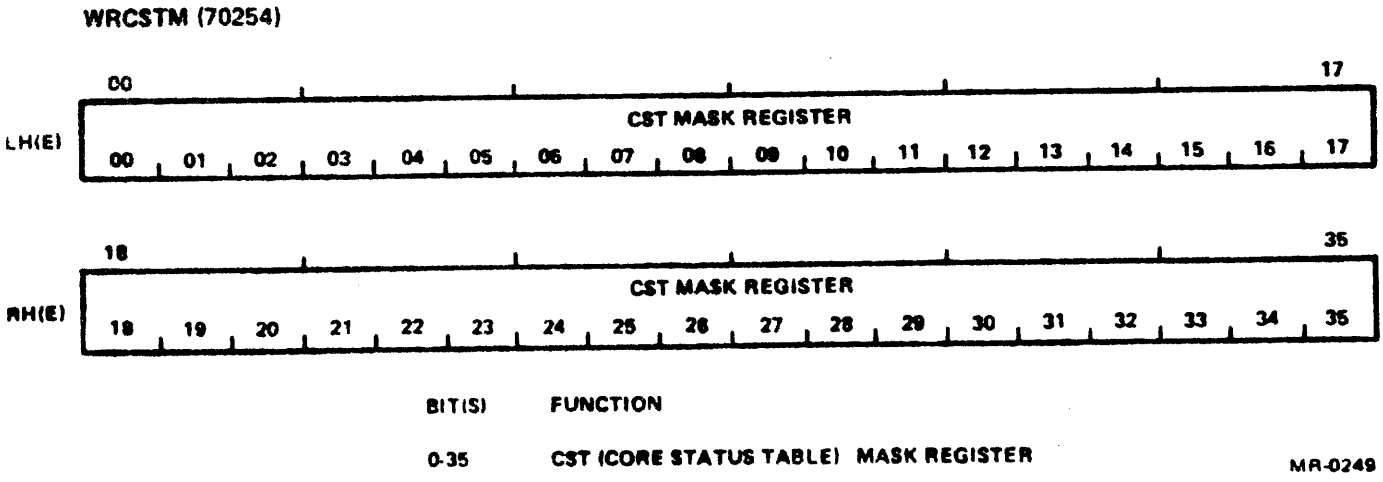


Figure 4-23 WRCSTM Instruction

4.3.1.22 WRTIME (70260) - The WRTIME instruction loads the double-word at E and E + 1 into the time base (Bit format is shown in Figure 4-24.) The Time Base up-counts at 4.096 mHz.

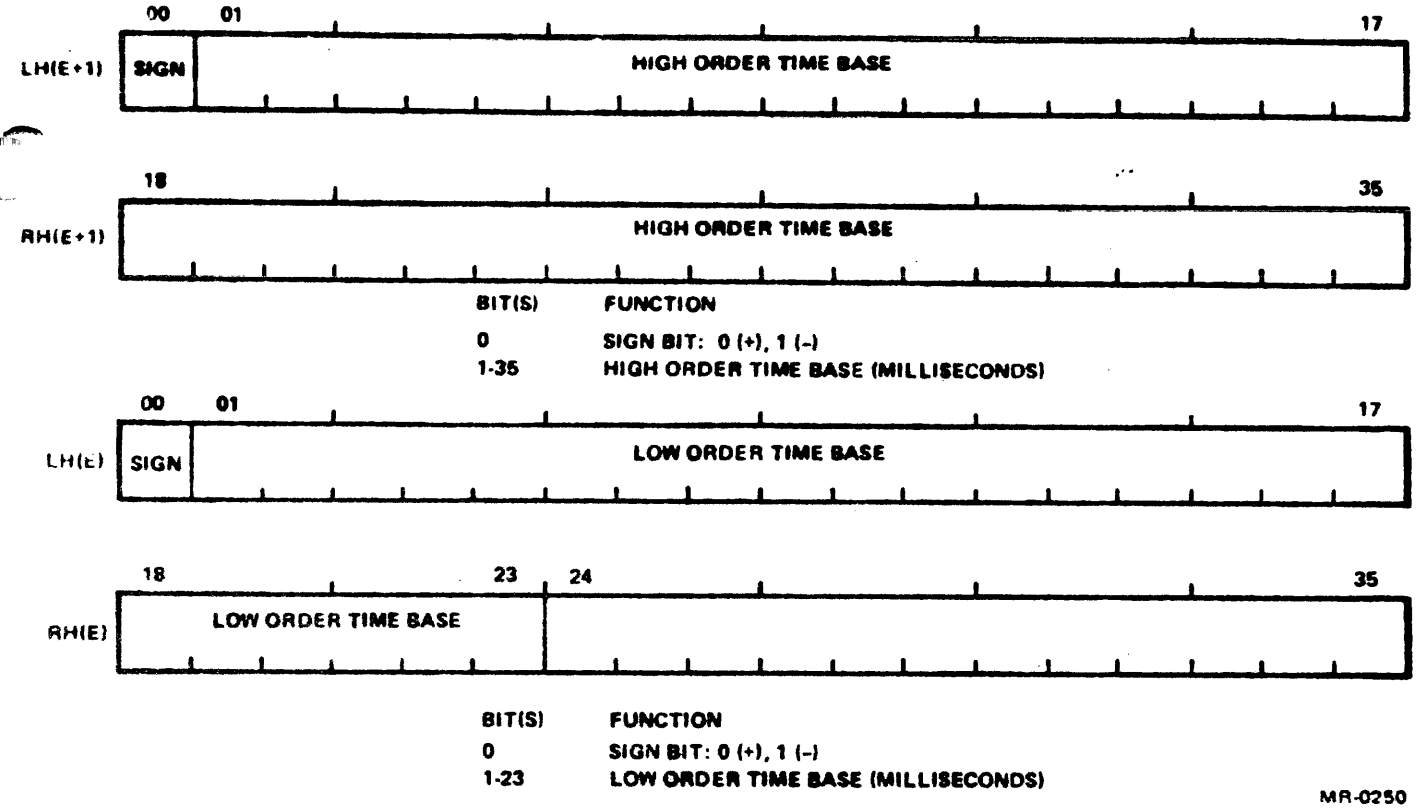


Figure 4-24 WRTIME Instruction

4.3.1.23 WRINT (70264) - The WRINT instruction loads the interval timer period register from E. The binary number (n) that is loaded determines the interval; that is, the interval (period) = n milliseconds. Bit format for the instruction is shown in Figure 4-25.

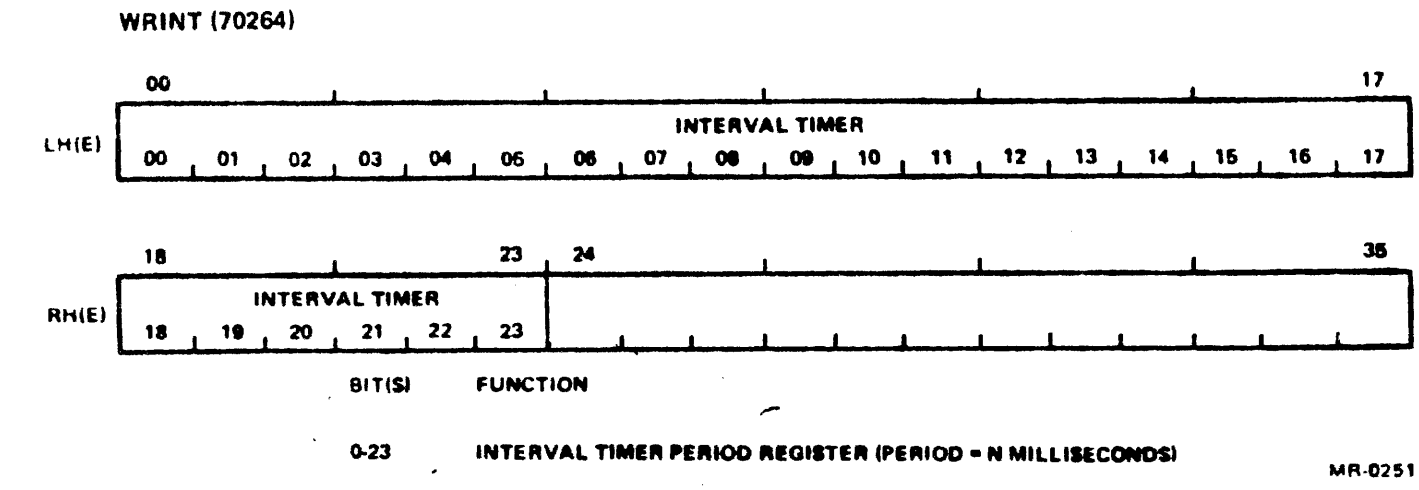


Figure 4-25 WRINT Instruction

4.3.1.24 **WRHSB (70270)** - The WRHSB instruction loads the signed word at E as the address of the halt status block (Subsection 4.7.2). If the word is negative or zero, no halt status will subsequently be stored. If the word is positive, the halt status block (20 words) will be stored starting at the specified address. Initially, when the microcode is loaded and started, the halt status block address is set to a value of (+) 376000. Bit format for the WRHSB instruction is shown in Figure 4-26.

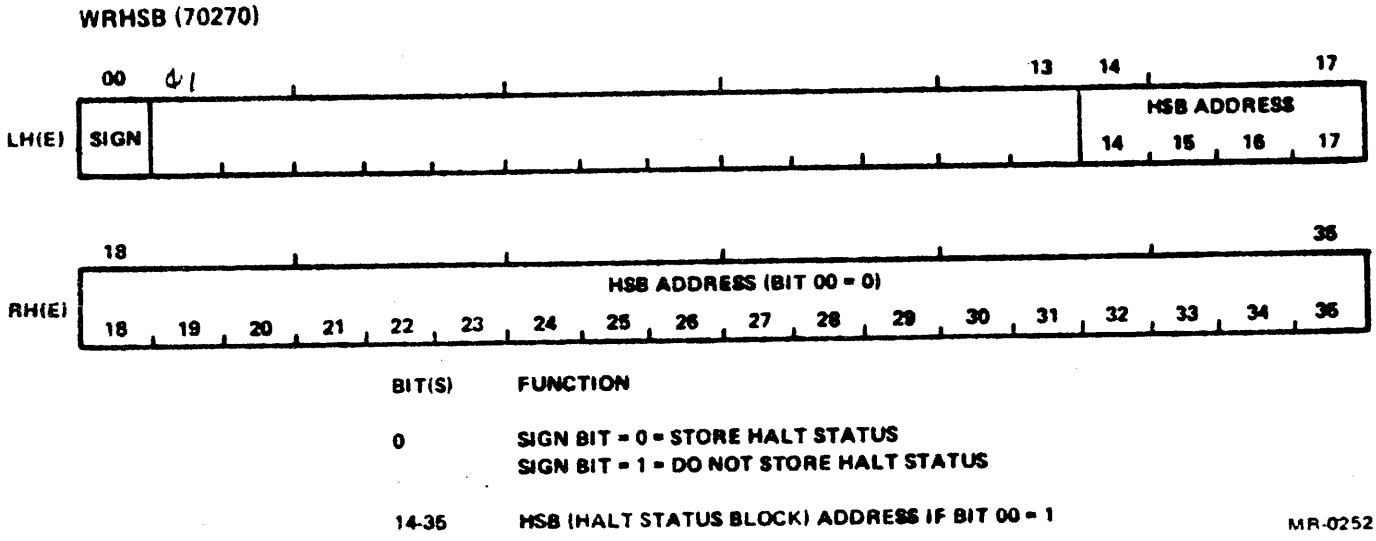
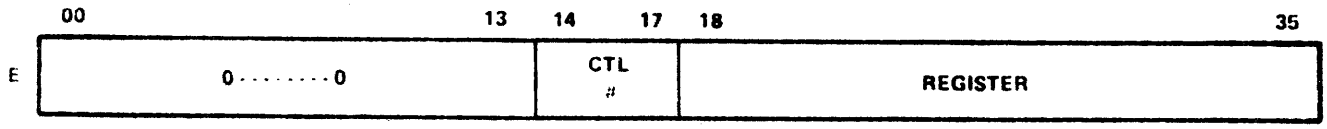


Figure 4-26 WRHSB Instruction

4.3.2 External I/O Instructions

The external I/O instructions read, write, modify, and test registers in KS10 devices external to the CPU. The effective address (E) for these instructions specify an I/O address; the specified AC holds register read/write data or mask data (for test or modification) depending on the instruction type. Both full-word (normal) instructions and byte instructions are implemented. The full-word instructions transfer 36 bits of data and use the full contents of an AC. The byte instructions, which are employed only when addressing Unibus device registers, transfer only eight bits of data and use only the eight right-most bits in an AC. The various external I/O instructions are described in Subsections 4.3.2.1 - 4.3.2.5.

The I/O address generated by the external I/O instructions' effective address consists of a controller number and a register address. Bit format and the general ranges of controller number and register address assignments are given in Figure 4-27. The specific I/O addresses for a fully comfigured KS10 are listed in Table 4-6.



CONTROLLER NUMBER	REGISTER ADDRESS	REGISTER(S)
0	0-077777	NOT USED
0	100000	MEMORY STATUS REGISTER
0	1000001-177777	NOT USED
0	200000	CONSOLE INSTRUCTION REGISTER
0	20000001-777777	NOT USED
1	0-377777	NOT USED
1	400000-777777	UNIBUS 1 (UBA AND DEVICE) REGISTERS
3	0-377777	NOT USED
3	400000-777777	UNIBUS 3 (UBA AND DEVICE) REGISTERS
2, 4-17		NOT USED

MR-0253

Figure 4-27 I/O Address Format

Table 4-6 External I/O Addresses

Register(s)	KS10 Bus Device	Unibus Device	CTL #	Register Address
Memory Status Register	Memory Cont.		0	100000
Console Instruction Register	Console		0	200000
UBA Paging RAM	UBA1		1	763000-77
UBA Status Register	UBA1		1	763100
UBA Maintenance Register	UBA1		1	763101
		Unibus 1		
	UBA1	RH11 # 1(RP06)	1	776700*
UBA Paging RAM	UBA3		3	763000-77
UBA Status Register	UBA3		3	763100
UBA Maintenance Register	UBA3		3	763101
		Unibus 3		
	UBA3	RH11 # 3(TU45)	3	772440*

UBA3	LP20 # 1	3	775400*
UBA3	DZ11 # 1	3	760010*
UBA3	DZ11 # 2	3	760020*
UBA3	DZ11 # 3	3	760030*
UBA3	DZ11 # 4	3	760040*
UBA3	KMC11 # 1	3	760540*
UBA3	DUP11 # 1	3	760300*
UBA3	DUP11 # 2	3	760310*

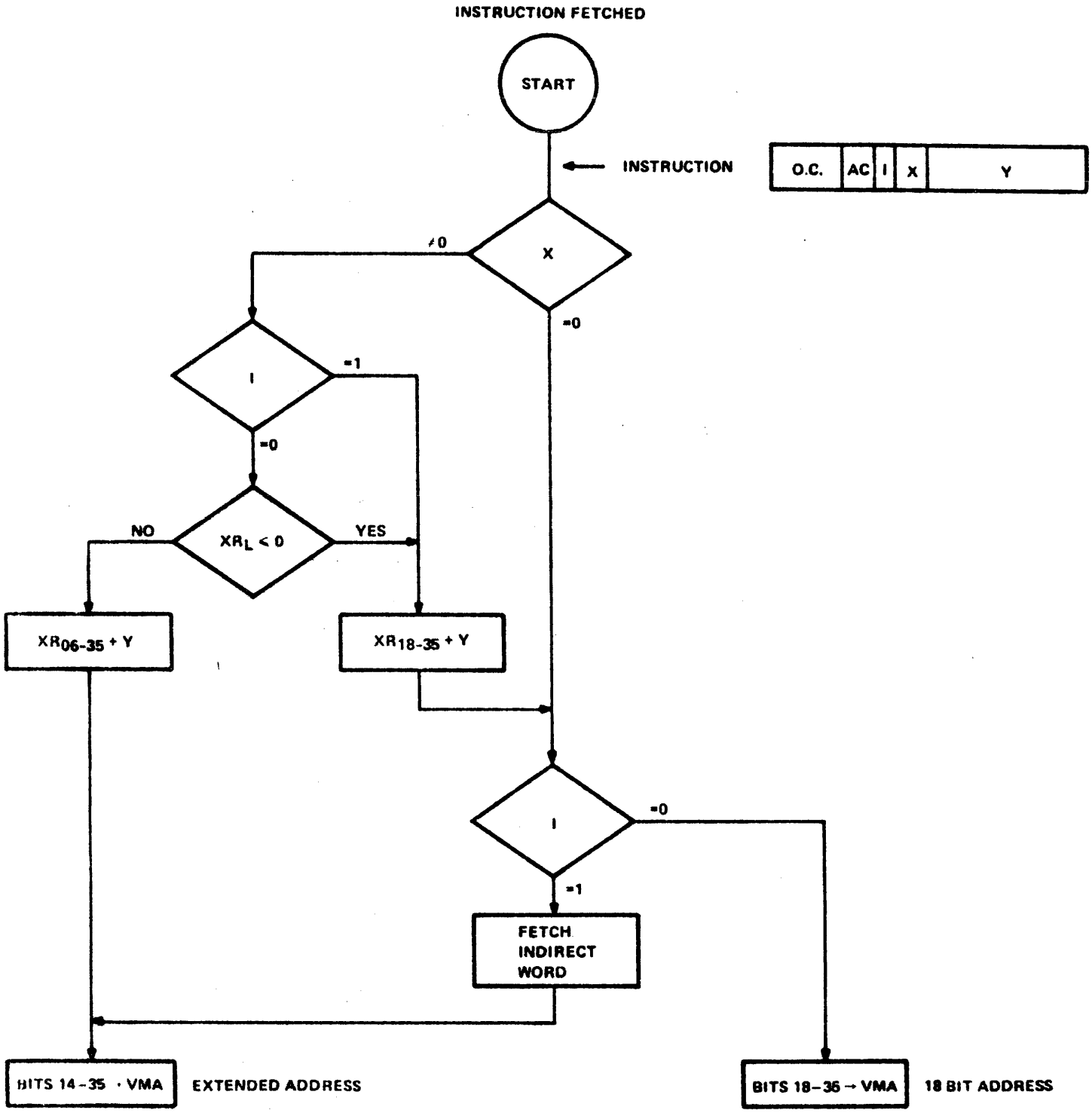
NOTE

An asterisk (*) indicates address is a base address. Refer to Appendix C for complete list of KS10 Unibus device registers and their addresses.

Note that an extended address (greater than 18 bits) is required to address controllers other than zero. The effective address calculation for external I/O instructions (diagramed in Figure 4-27A) is as follows:

a. If there is no indexing or indirection, Y is used as the effective address.

b. If there is indirection (or indirection and



NOTES:

I = INDIRECT BIT
X = INDEX REGISTER BIT
XR = INDEX REGISTER CONTENTS
Y = ADDRESS BITS
VMA = VIRTUAL MEMORY ADDRESS

MR-0841

Figure 4-27A Effective Address Calculation for External I/O Instructions

indexing), Y (or Y indexed by bits 18-35 of the index register) is used as an address for an indirect word fetch and the contents of the indirect word (bits 14-35) are used as the effective address.

c. If there is indexing (and no indirection) and the left half of the index register is less than or equal to zero, Y indexed by bits 18-35 of the index register is used as the effective address.

d. If there is indexing (and no indirection) and the left half of the index register is positive, Y indexed by bits 06-35 of the index register is used as the effective address.

The operations described above are summarized in Table 4-7. The result of the effective address calculation, which can be either an 18-bit or an extended I/O address is also indicated.

Table 4-7 Summary of Effective Address Calculation for External I/O Instructions

Indirection	Indexing	XRL	Effective Address	Comments
NO	NO	-	Y	18-bit Address
YES	NO	-	(Y)	Extended Address
YES	YES	-	(Y+XR18-35)	Extended Address

NO	YES	≤0	Y+XR18-35	18-bit Address
NO	YES	>0	Y+XR06-35	Extended Address

It can be seen that to address controllers other than zero, which require an extended address, instructions using indexing or indirect addressing must be used. Examples of each follow, both of which read the status register (register address = 763100) in UBA #1 (controller number = 1) into an AC with the RDIO external I/O instruction (Subsection 4.6.2.3).

Example 1 (using indexing):

RDIO AC,763100(X) where the contents of index register X = 1000000

The index register contents (the controller number) is added to Y (the register address) to give the extended I/O address 1736100.

Example 2 (using indirection):

RDIO AC, @ 100 where the contents of 100 = 1763100

An indirect word fetch of location 100 is made and the contents are used to generate the extended I/O address 1763100.

4.3.2.1 TIOE and TIOEB (710 and 720) - The TIOE (or TIOEB)

instruction fetches one word (or byte) from the I/O address specified by E, and ANDs the word (or byte) with the contents of the specified AC. The instruction skips if the result of the AND is zero. The contents of the AC are not modified.

4.3.2.2 **TION and TIONB (711 and 721)** - The TION (or TIONB) instruction performs the same function as the TIOE (or TIOEB) instruction except that the instruction skips if the result of the AND is not zero.

4.3.2.3 **RDIO and RDIOB (712 and 722)** - The RDIO (or RDIOB) instruction fetches the word (or byte) from the I/O address specified by E and stores the word (or byte) right-justified in the specified AC.

4.3.2.4 **WRIO and WRIOB (713 and 723)** - The WRIO (or WRIOB) instruction takes the word (or byte) contained in the specified AC and transfers the word (or byte) to the I/O address specified by E.

4.3.2.5 **BSIO and BSIOB (714 and 724)** - The BSIO (or BSIOB) instruction fetches the word (or byte) from the I/O address specified by E, ORs the word (or byte) with the contents of the specified AC, and then transfers the result back to the I/O address. The instruction(s) may be used to set selected bits in Unibus device registers. The contents of the AC are not modified.

4.3.2.6 **BCIO and BCIOB (715 and 725)** - The BCIO (or BCIOB) instruction is similar to the BSIO (or BSIOB) instruction except that the word (or byte) read from the I/O address is ANDed with the complement of the AC contents. The instruction(s) may be used to clear selected bits in Unibus device registers. The contents of the AC are not modified.

4.3.3 PXCT Extensions

The UMOVE and UMOVEM instructions have been originated for the KS10 to save time and space in the monitor.

4.3.3.1 **UMOVE (704)** - The UMOVE (move from previous context) instruction performs the same functions as:

```
PXCT 4, [MOVE AC, E]
```

4.3.3.2 **UMOVEM (705)** - The UMOVEM (move to previous context) instruction performs the same function as:

```
PXCT 4, [MOVEM AC, E]
```

4.4 KS10 PROCESSOR STATUS WORDS

Whenever the KS10 processor halts, it writes a halt status word, the PC, and (optionally) a halt status block (of 18 words) in memory.

The halt status word contains a code indicating the type of halt; the halt status block contains a read-out of several CPU registers (HR, VMA, etc.) at the time of the halt. Other KS10 status words include the page fail word, which is written into the UPT following a page failure; and the PC word (PC with flags), which is stored in an AC or memory location by certain system level instructions.

4.4.1 Halt Status Word

A halt status code is stored in physical memory location 0 (not AC 0) whenever the KS10 Processor halts. Codes in the range 0-77 (octal) indicate normal halts; codes in the range 100-177 (octal) indicate software failures; codes of 1000 (octal) or greater indicate microcode or software failures. Bit format and halt code definitions are given in Figure 4-28.

4.4.2 PC

A processor halt causes the PC to be stored right justified in physical memory location 1 (not AC1).

4.4.3 Halt Status Block

If the halt status block address is positive, a processor halt

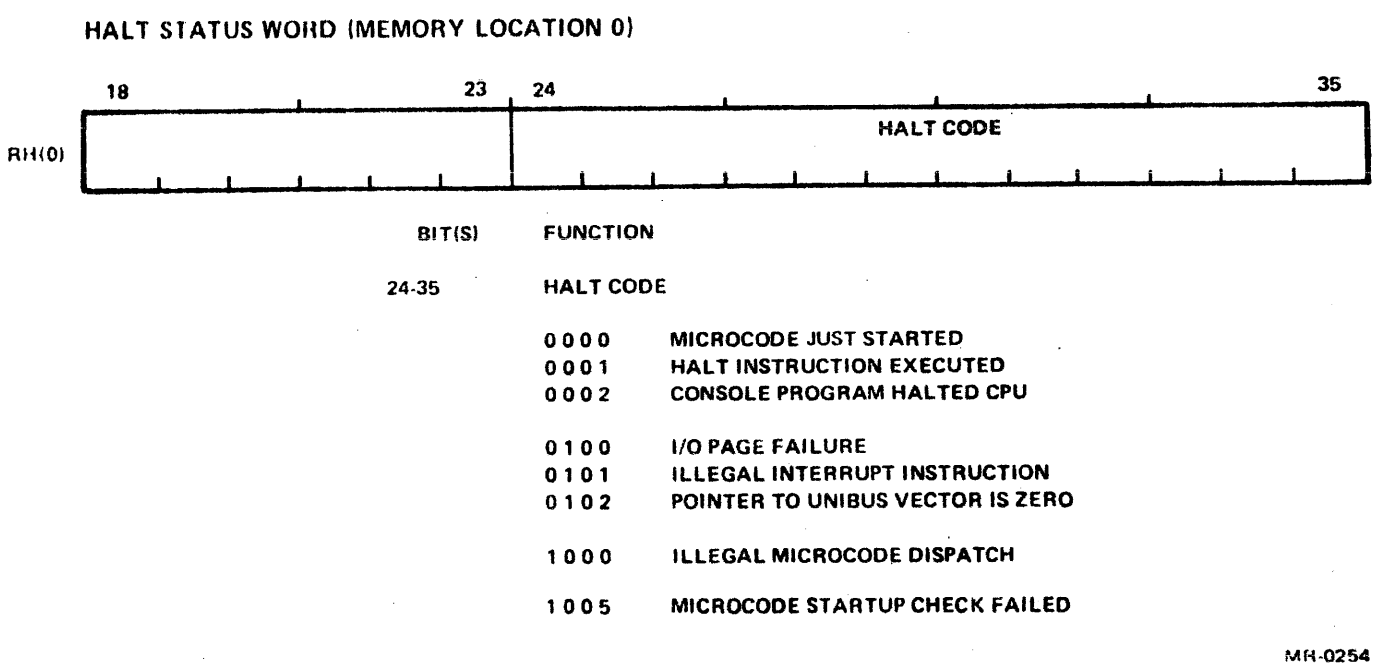
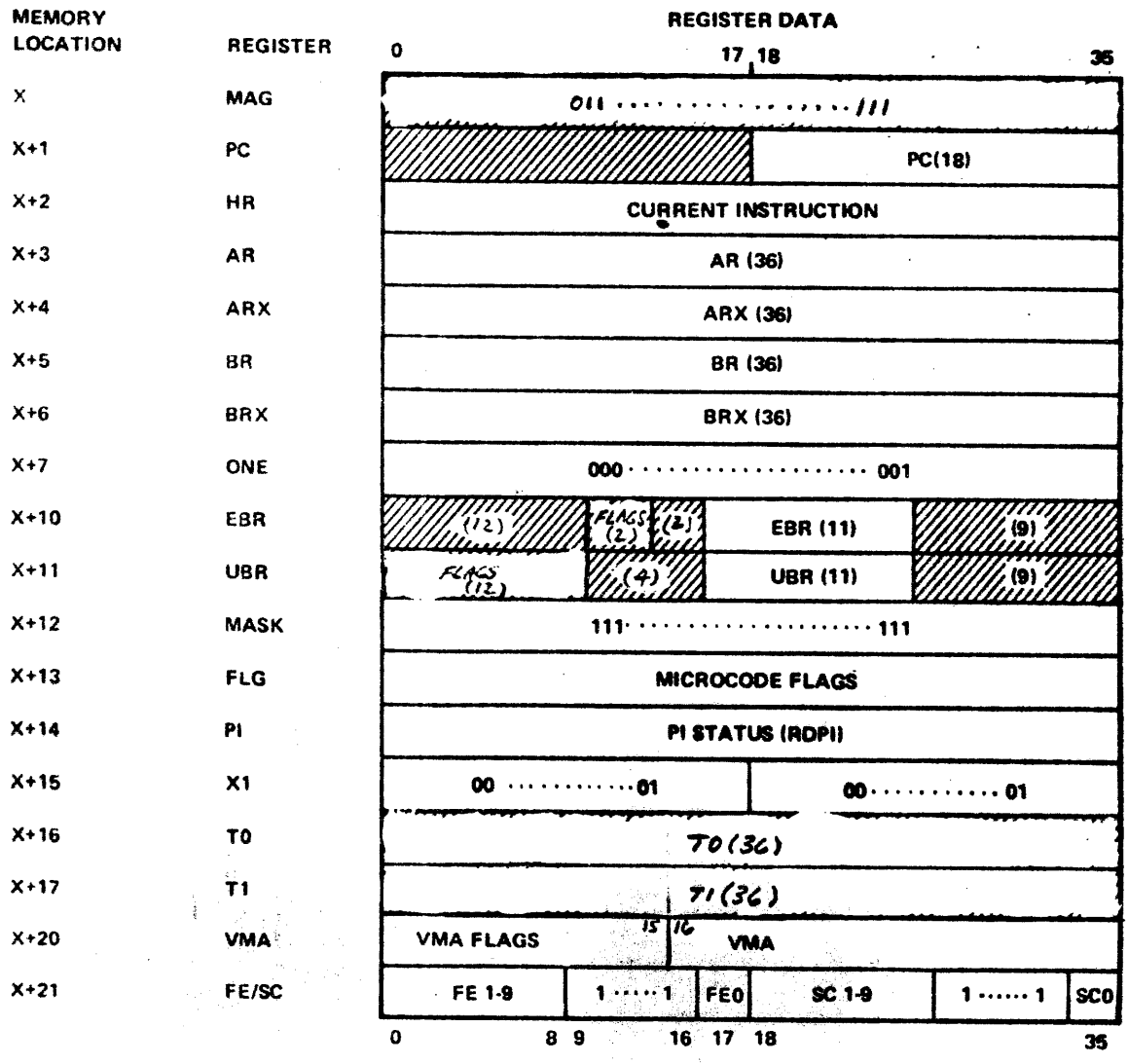


Figure 4-28 Halt Status Word

causes the contents of several processor registers to be stored in a block of KS10 memory starting at the specified address. Figure 4-29 shows the information stored in each location. if the halt status block address is negative, the halt status block is not stored. The WRHSB instruction (Subsection 4.3.1.24) allows the program to load any address value. Initially, when the microcode is started, the halt status block address is set to a value of (+) 376000 and the halt status block is stored.

The first 16 memory locations of the halt status block hold the register data read from the 16-word RAMs associated with the 2901 microprocessor circuits. Significant status information includes the PC (also stored in memory location 1), current instruction, EBR, UBR, microcode flags, and PI system status. The microcode flags are described in Subsection 4.4.3.1. PI system status is the same as that read by the RDPI instruction (Subsection 4.3.2.1).



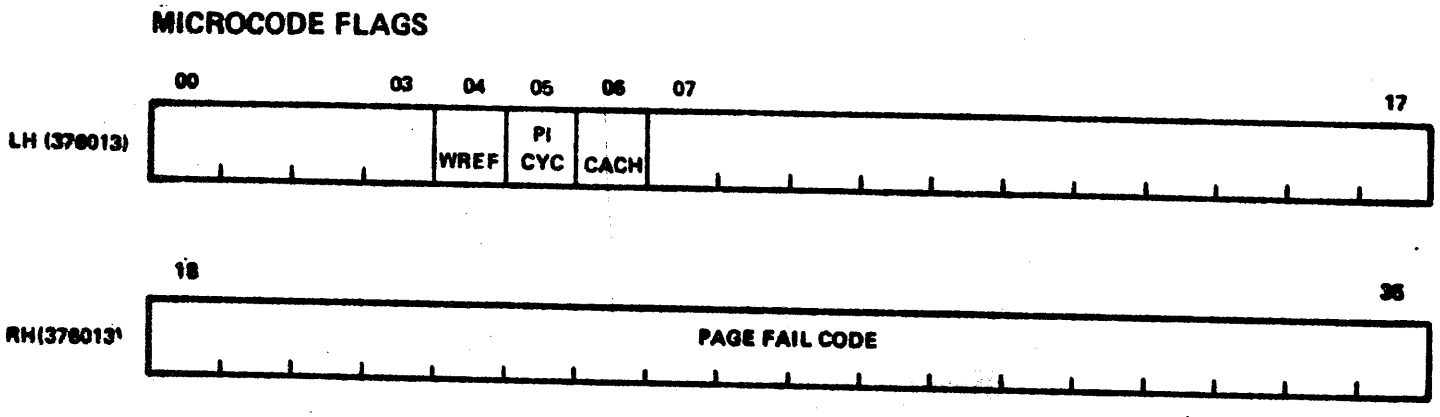
* NOTE. X = 376000 WHEN UCODE FIRST LOADED. ADDRESS MAY BE CHANGED BY WRHSB INSTRUCTION.

MR-0255

Figure 4-29 Halt Status Block

Another processor status word, the VMA contents (plus flags), is stored in the next to last location of the halt status block. Bit definitions for the VMA word are given in Subsection 4.4.3.2.

4.4.3.1 Microcode Flags - In the event of a page failure, three flags and a page fail code are stored as part of the halt status block in X+13. The page fail code, which is the contents of the microword's magic number field, specifies the operation for which the page failure occurred. Status word bit format and page fail code definitions are given in Figure 4-30.



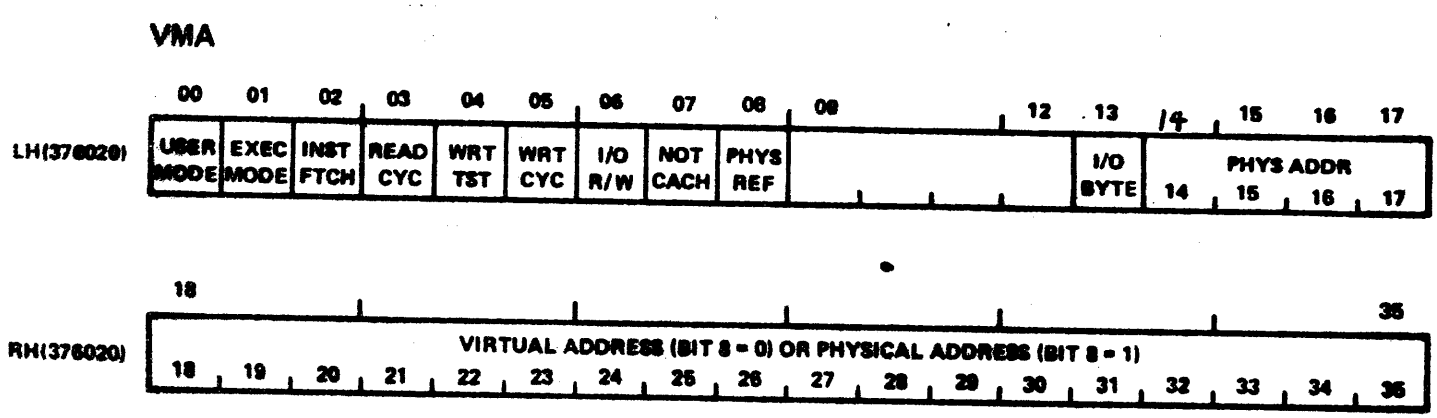
BIT(S)	FUNCTION
4	WRITE REFERENCE BIT FROM PAGE MAP
5	PI CYCLE
6	LOOK IN CACHE BIT FROM PAGE MAP
18-35	PAGE FAIL CODE

000000	SIMPLE INSTRUCTIONS
000001	BLT IN PROGRESS
400002	MAP IN PROGRESS
000003	MOVE STRING SOURCE IN PROGRESS
000004	MOVE STRING FILL IN PROGRESS
000005	MOVE STRING DESTINATION IN PROGRESS
000006	FILLING DESTINATION
000007	EDIT SOURCE
000010	EDIT DESTINATION
000011	CONVERTING DECIMAL TO BINARY
000012	COMPARING DESTINATION

MR-0256

Figure 4-30 Microcode Flags

4.4.3.2 **VMA** - The virtual memory address (VMA) and VMA flags are stored in location X + 20 of the halt status block. Bit format and definitions are given in Figure 4-31.

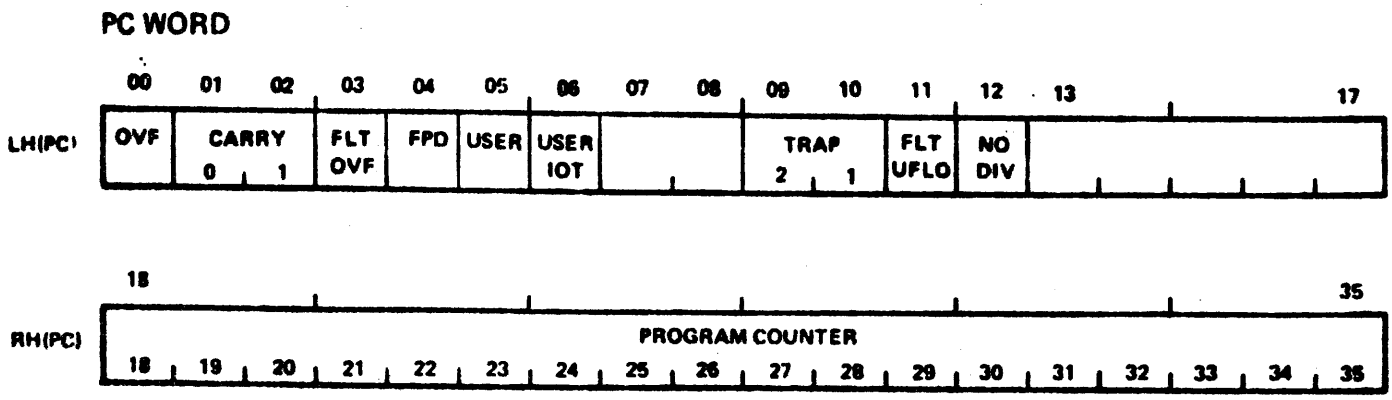


BIT(S)	FUNCTION
0	USER MODE
1	EXEC MODE
2	INSTRUCTION FETCH
3	READ CYCLE
4	WRITE TEST
5	WRITE CYCLE
6	I/O READ OR WRITE
7	DO NOT LOOK IN CACHE
8	PHYSICAL REFERENCE
13	I/O BYTE INSTRUCTION
14-17	BITS 14-17 OF PHYSICAL ADDRESS (OR 0s)
18-35	BITS 18-35 OF VIRTUAL ADDRESS (BIT 8 = 0) OR PHYSICAL ADDRESS (BIT 8 = 1)

MR-0267

Figure 4-31 VMA

4.4.4 **PC Word** - Several of the jump instructions (e.g., JSR) save the PC and various processor flags in a memory location or an AC. Bit format for this PC word is shown in figure 4-32.



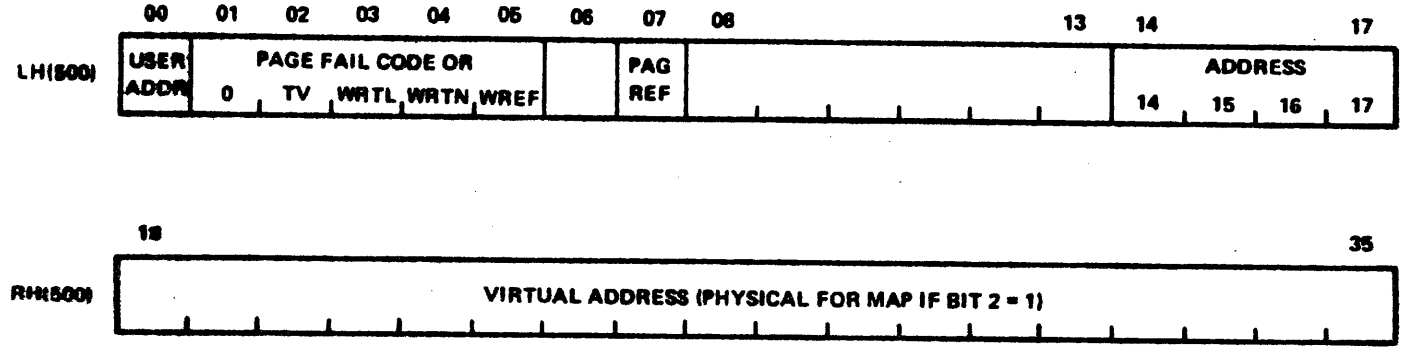
BIT(S)	FUNCTION
0	OVERFLOW
1	CARRY 0
2	CARRY 1
3	FLOATING OVERFLOW
4	FIRST PART DONE
5	USER MODE
6	USER IOT (ALSO PCU)
9	TRAP 2
10	TRAP 1
11	FLOATING UNDERFLOW
12	NO DIVIDE
18-35	PROGRAM COUNTER

MR-0258

Figure 4-32 PC Word

4.4.5 Page Fail Word - Following all page failures, except for in-out failures, the processor causes a page fail trap and stores a page fail word in location 500 (octal) of the UPT. Bit format is shown in Figure 4-33.

PAGE FAIL WORD (OR MAP AC)



BIT(S)	FUNCTION
0	USER ADDRESS
2-5 (BIT 1 = 0)	
2	TRANSLATION VALID
3	WRITABLE
4	WRITTEN
5	WRITE REFERENCE
2-5 (BIT 1 = 1)	PAGE FAIL CODE
	20 AN I/O INSTRUCTION SELECTED A NONEXISTENT DEVICE OR REGISTER. (BITS 14-35 = I/O ADDRESS)
	25 PAGE TABLE PARITY ERROR
	36 HARD MEMORY ERROR
	37 NXM
07	PAGED REFERENCE
18-35	VIRTUAL ADDRESS (PHYSICAL FOR MAP IF BIT 2 = 1)

MR-0259

Figure 4-33 Page Fail Word

4.5 KS10 EPT/UPT

Executive process table (EPT) and user process table (UPT) configurations for the KS10 are shown in Figures 4-34 and 4-35.

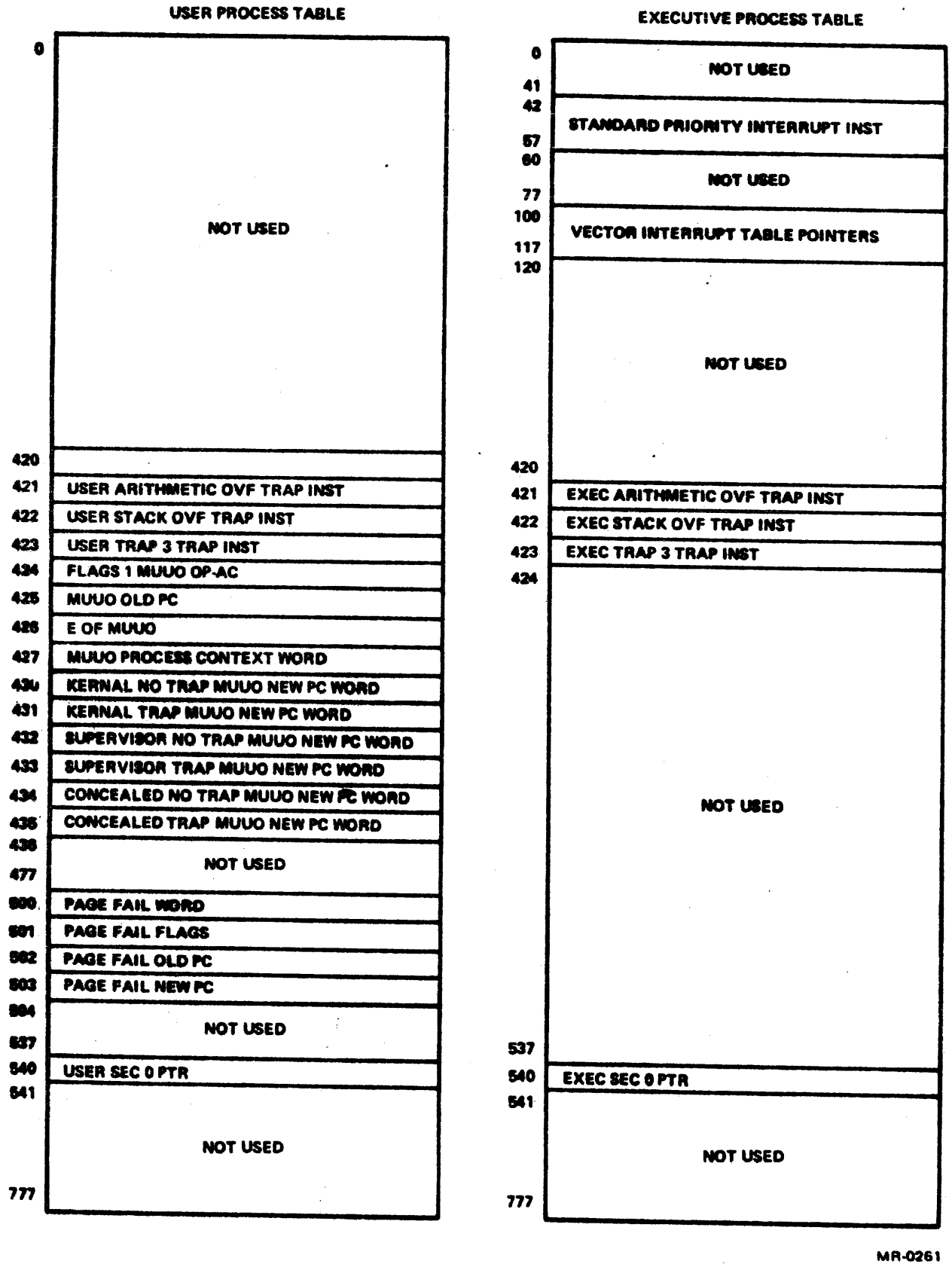


Figure 4-34 KS10 EPT/UPT (TOPS20 Paging)

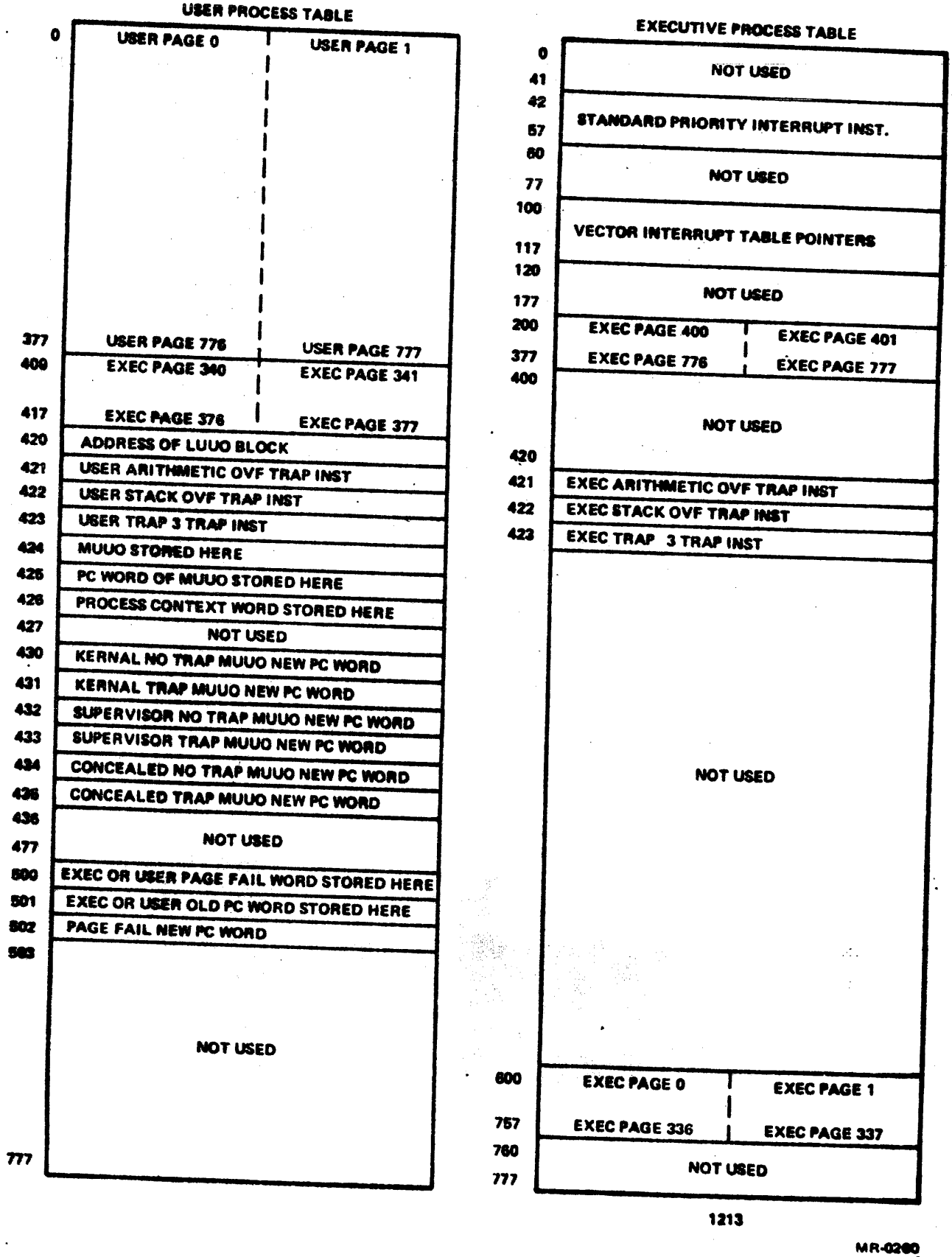


Figure 4-35 KS10 EPT/UPT (TOPS10 Paging)

4.6 OPERATOR CONSOLE

Local operator control of the KS10 is by a set of commands typed at the console terminal (CTY). The CTY connects directly to the 8080-based console hardware via a serial line. A second serial line, that operates in parallel with the first line, may also be connected to the console hardware to allow control of the KS10 by a remote diagnosis link. Other (user only) terminals connect to the KS10 via the Unibus (DZ11s).

The commands typed at the CTY, or entered from the remote diagnosis link, are implemented by the program running in the console module's 8080 microprocessor. The program is resident in PROM and valid at power-up.

The CTY and remote diagnosis link operate in either of two modes. The two modes are:

1. User mode

2. Console mode

NOTE

The CTY and remote diagnosis link may operate in the same mode or in different modes. Subsection 4.7, KLINIK OPERATION, describes how the remote link gains access to the system and enters either user mode or console mode. The

remainder of this subsection (Subsection 4.6) applies to CTY operation only. Commands are the same for the remote link but operation is restricted in some cases.

In user mode, the CTY is a user terminal and (with one exception) commands are passed to and from the KS10 CPU under control of the console program. The exception is a control backslash ("\"), which causes the console program to switch the CTY from user mode to console mode. All other commands performed in user mode are a function of the operating system (TOPS20) resident in KS10 memory.

In console mode, commands are directed to (and executed by) the 8080 console hardware. An operator may perform the following major functions:

1. Reset and bootstrap system.

2. Load and check microcode.

3. Deposit and examine memory.

4. Read and write I/O device registers.

5. Read and write KS10 bus.

6. Start and stop CPU clock.

7. Single-step the CPU clock.

8. Execute a given instruction.

9. Halt the machine.

10. Start the machine at a given location.

11. Single-instruct a program.

The console program initializes the CTY to console mode at power-up. When in console mode, starting or continuing program execution (ST or CO commands) or a control-Z switches the CTY to user mode. As stated previously, a control backslash ("\") in user mode causes a return to console mode. Also, an error which lights the FAULT indicator causes a return to console mode, as does any KS10 processor halt instruction.

4.6.1 Console Mode Commands

The console mode command prompt are the characters "KS10" followed by a greater than sign (KS10>). A command, or a string of commands separated by commas, may then be typed and followed by a carriage return (CR). The CR causes the command, or string of commands, to be executed. The various console mode commands are listed in Table 4-8. Error printouts are listed in Table 4-9.

Table 4-8 Console Mode Commands

Load Commands (Values loaded in 8080 RAM locations for subsequent use as arguments by associated deposit and examine commands.)	Function
LA xx	Set KS10 memory address to xx (0000000-1777777).
LC xx	Set CRAM address to xx (0000-3777).
LF xx	Load diagnostic write function xx (0-7). The function specifies a 12-bit group within a CRAM address.

LF	CRAM Bits
0	00-11
1	12-23
2	24-35
3	36-47
4	48-59
5	60-71
6	72-83

LI xx — Load I/O address. The address consists of a control number and a register address. I/O addresses accessable from the console are listed below. Note that the address of the console instruction register is not included. If the console attempts to access its own instruction register, no response occurs.

CTL	Register Address	Register(s)
0	100000	Memory status register
1,3	763000-77	UBA paging RAM
1,3	763100	UBA status register
1,3	763101	UBA maintenance register
1,3	7xxxxx	Unibus device registers (Appendix C)

LK xx — Set 8080 memory address to xx. (PROM address = 00000-17777; RAM address = 20000-21777).

Deposit Commands

DB xx — Deposit xx (36 bits) onto KS10 bus.

* DC xx — Deposit xx (96 bits) into CRAM. Address previously loaded by LC command.

*	DF	xx	Deposit xx (12-bit group) into CRAM. Address and diagnostic function previously loaded by LC and LF commands.
	DI	xx	Deposit xx (16, 18 or 36 bits) into an I/O register. Address previously loaded by LI command.
	DK	xx	Deposit xx (8 bits) into 8080 memory. Address previously loaded by LK command. (Data cannot be deposited in PROM addresses; only in RAM addresses.)
	DM	xx	Deposit xx (36 bits) into KS10 memory. Address previously loaded by LA command.
	DN	xx	Deposit xx into next (KS10, 8080, I/O) address.

Examine Commands

	EB		Examine KS10 Bus. Prints contents of console registers 100-103, 300, and 301 (Subsection 4.6.2).
*	EC		Examine current contents of CRAM control register.

*	EC	xx	Examine contents of CRAM address xx.
	EI		Examine contents of I/O register. Address previously loaded by LI command.
	EI	xx	Examine contents of I/O address xx.
*	EJ		Examine current CRAM address, next CRAM address, jump address, and subroutine return address.
	EK		Examine contents of 8080 memory. Address previously loaded by LK command.
	EK	xx	Examine contents of 8080 memory address xx.
	EM		Examine contents of KS10 memory. Address previously loaded by LA command.
	EM	xx	Examine contents of KS10 memory address xx.
	EN		Examine contents of next (KS10, 8080, I/O) address.

Start/Stop Clock Commands

	CH		Halt CPU clock.
*	CP		Pulse CPU clock.
*	CP	xx	Pulse CPU clock xx times.
*	CS		Start CPU clock.

Start/Stop Microcode Commands

*	PM		Pulse microcode. Performs a CP command to execute a microinstruction followed by an EJ command to print current CRAM address, next CRAM address, jump address, and subroutine return address.
*	SM		Reset and start microcode at CRAM address 0.
*	SM	xx	Reset and start microcode at CRAM address xx.
*	TR		Trace. Repeats PM command until any CTY key is depressed.
*	TR	xx	Trace. Repeats PM command until CRAM address xx is reached or until any CTY key is depressed.

Start/Stop Program Commands

HA		Halt KS10 program. Microcode enters halt loop.
CO		Continue KS10 program execution. Enter user mode.
SH		Shut down command. Deposits non-zero data into KS10 memory location 30 to allow orderly shut down of the monitor.
SI		Single instruct. Execute next KS10 instruction.
ST	xx	Start KS10 program at address xx. Enter user mode.

Select Device Commands

DS		Select disk for bootstrap. Console program asks for UBA number (default = 1), RH11 base address (default = 776700), and disk unit number (default = 0) as follows:

```
>>UBA? 1 <CR>
>>RHBASE? 776700 <CR>
>>UNIT? 0 <CR>
```

The default value for the RH11 base address is currently the only value permitted. Also, a carriage return in response to any question retains the current value.

MS

Select tape for bootstrap. Console program asks for UBA number (default = 3), RH11 base address (default = 772440), tape unit number (default = 0), tape density (default = 1600 BPI), and slave number (default = 0) as follows:

```
>>UBA? 3 <CR>
>>RHBASE? 772440 <CR>
>>UNIT? 0 <CR>
>>DENS? 1600 <CR>
>>SLV? 0 <CR>
```

The default value for the RH11 base address is currently the only value permitted. Also, a carriage return in response to any question retains the current value.

Boot Commands

BT

Bootstrap the KS10 from disk. Loads and starts microcode and monitor boot program from drive 0

on UBA1 (default address) or drive selected by last DS command; starts KS10 at memory address 1000.

BT 1 — Same as BT command except that diagnostic boot program (not monitor boot program) is loaded and started.

BC — Check the KS10 boot path.

LB — Load the monitor boot program from the disk selected last. Does not load microcode. Program must be started at 1000.

LB 1 — Same as LB command except that diagnostic boot program (not monitor boot program) is loaded. Program must be started at 1000.

MB — Load the monitor boot program from the tape selected last. Does not load microcode. Program must be started at 1000.

MT — Bootstrap the KS10 from tape. Loads and starts microcode and monitor boot program from tape unit 0, slave unit 0 on UBA3 (default address) or drive selected by last MS command.

Mark/Unmark Microcode Commands

*	MK	xx	Mark microcode word (set bit 95) at CRAM address xx.
*	UM	xx	Unmark microcode word (clear bit 95) at CRAM address xx.

Master Reset Command

	MR		Master reset. Issue bus reset.

Execute Command

	EX	xx	Execute the single KS10 systems-level instruction xx.

Enable/Disable Commands

	CE	xx	Enable (xx=1) or disable (xx=0) cache.
	PE	xx	Enable or disable parity detection as follows:

PE

0 Disable all parity detection.

4 Enable KS10 bus parity detection.

5 Enable DPE/DPM parity detection.

6 Enable CRA/CRM parity detection.

7 Enable all parity detection.

TE Enable (xx=1) or disable (xx=0) CPU interval timer interrupts.

TP Enable (xx=1) or disable (xx=0) CPU traps.

Following an enable/disable command with a carriage return gives the current value.

Read Cram Commands

* RC Read CRAM data. Performs diagnostic read functions 0-17 to read CRAM addresses and contents (of current address) as follows:

RC	Data
0	CRAM bits 00-11
1	Next CRAM address
2	CRAM subroutine return address

3	Current CRAM address
4	CRAM bits 12-23
5	CRAM bits 24-35 (Copy A)
6	CRAM bits 24-35 (Copy B)
7	0s
10	Parity bits A-F
11	KS10 Bus bits 24-35
12	CRAM bits 36-47 (Copy A)
13	CRAM bits 36-47 (Copy B)
14	CRAM bits 48-59
15	CRAM bits 60-71
16	CRAM bits 72-83
17	CRAM bits 84-95

Zero Memory Command

ZM	Zero memory. Deposit 0s into all KS10 memory locations.

Repeat Command

RP	Repeat last command, or last command string, until any CTY key is depressed.

Lamp Test Command

LT	Blink indicators. Momentarily lights (1-2

seconds) and turns off (1-2 seconds) STATE, FAULT, and REMOTE indicators. The indicators are then returned to their original state.

Password Command

PW xx	Set password xx (xx=maximum of 6 alpha-numeric characters).
	Following a PW command with a carriage return clears the password storage area.

KLINIK Command

KL xx	Enables remote link with access to system to operate in user mode but not in console mode (xx=0). Enables remote link with access to system to operate in console mode or in user mode (xx=1).
	Following a KL command with a carriage return gives the current value.

Special Control Characters

control-C	Abort current command. Console returns command prompt.
control-O	Inhibit CTY output (type-outs).
control-S	Inhibit CTY output and stop 8080 console program until control-Q is typed at CTY.
control-Q	Enable CTY output and continue 8080 console program.
control-U	Delete current line.
control-Z	Enter user mode.
RUB-OUT	Delete last character.

NOTES

1. **An asterisk (*) indicates that the CPU clock must be stopped in order to execute the command.**

2. **More than one command may be entered on a line (separated by commas) and executed as a command string.**

3. Commands (except for special control characters) and command strings are followed by a carriage return (CR) to cause command execution. (Special control characters are executed when typed.)

Table 4-7 Console Mode Error Messages

Message	Meaning
?A/B	A not equal to B (A and B copies of a microcode field did not match.)
?BFO	Buffer overflow. (Too many characters typed; console's 80 character input buffer is full.)
?BN	Bad number. (Character typed is not an octal number.)
?BT xx	BT command failed. (I/O ERRTEST at 8080 address xx failed.)
?C CYC	Command/address cycle failed. (KS10 bus data failure detected during DB command; good and bad data

printed.)

?D CYC — Data cycle failed. (KS10 bus data failure detected during DB command; good and bad data printed.)

?DNC — Did not complete. (HA or SM command did not cause microcode to enter halt loop.)

?DNF — Did not finish. (ST, CO, or EX command did not clear console's CONTINUE bit.)

?IA — Illegal address. (Address typed is out of range.)

?IL — Illegal command. (Command typed is not valid.)

?MEM REFRSH ERR — Memory refresh error. (Incomplete KS10 MOS memory cycle. Error occurs when memory must be refreshed in hung state.)

?NBR — No bus response. (Console did not receive GRANT after requesting KS10 Bus.)

?NDA — No data acknowledge. (Consle did not receive DATA CYCLE signal after a data request.)

?NXM — Non-existent memory. (Deposit or examine command

referenced non-existent KS10 MOS memory location.)

?PAR ERR xx System parity error. (CPU clock stopped due to system parity; xx=contents of the following console status registers in the order indicated:

100, 303, 103

Refer to Subsection 4.6.2 for status register bit format.

?RA Requires argument. (Command typed requires an argument.)

?RUNNING Clock running. (Command typed requires CPU clock to be stopped.)

?UI Unknown interrupt. (Console received interrupt but CTY has no character.)

4.6.2 Console Status Registers

The console program reads and prints (at the CTY) the contents of certain 8080 registers in response to the EB (examine bus) command and a system parity error (?PAR ERR). The EB command prints registers 100-103, 300, and 301. Registers 100, 303, and 103 are printed when the system parity error is detected. Register bit format is shown in Figure 4-36.

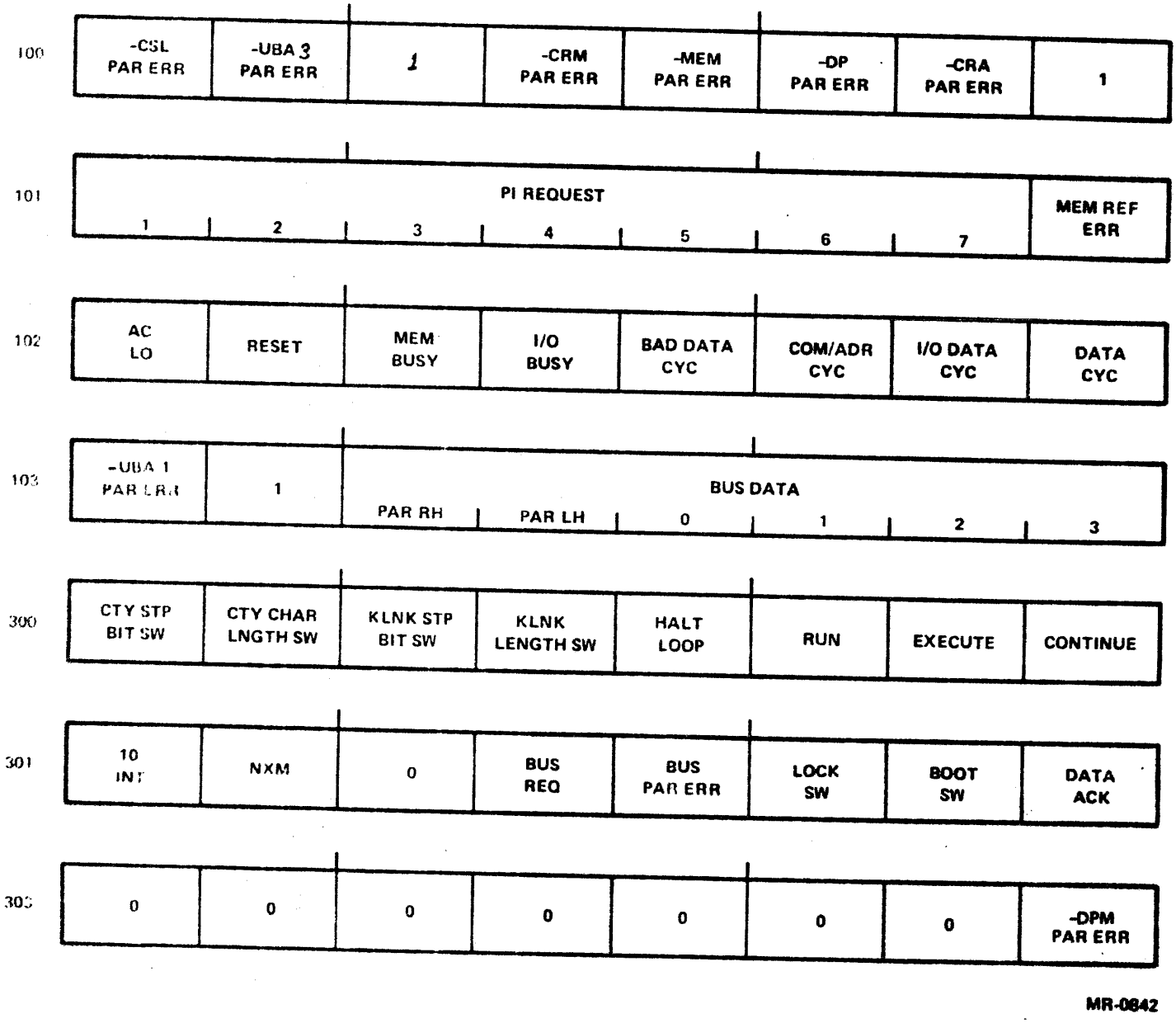


Figure 4-36 Console Status Registers

4.7 KLINIK OPERATION

TO BE SUPPLIED

SECTION 5

TECHNICAL DESCRIPTION

The organizational structure of the KS10 can be viewed as a hierarchy of buses, each of which is a data path shared by a number of different logical elements. The overall system comprises four or more major units or subsystems that communicate with one another over a backpanel bus. Each subsystem in turn is made up of secondary components grouped around one or more secondary buses or data paths. In almost all cases these buses are bidirectional. At the system level, the backpanel bus has a single set of data lines over which any subsystem can communicate with any other, although there are limitations on which subsystems can initiate transactions with which others and on which directions of data flow are available between a given pair of subsystems. In the memory or an in-out subsystem, data flow is bidirectional but occurs between a single controller and any one of a number of storage modules or peripheral devices. The processor, which is the most complex of the subsystems, and controls the normal operation of the entire system, is based on a unidirectional data path that is in several parts, and has several major loops, with many side loops and with logic elements lying in the various parts of the path.

The first section of this chapter discusses the overall flow of data in the system as a whole and in the processor in particular, and describes those characteristics common to

all subsystems, mainly the timing and the manner of communication over the backpanel bus. The remaining sections give detailed descriptions of the bus and the various subsystems, with several sections devoted to the major parts of the processor.

5.1 ORGANIZATION AND TIMING

Figure 5.1 shows the physical arrangement of the KS10 boards in the ~~rack~~ backplane. The slot numbers are at the tops of the columns representing the boards - just below the three-letter board designations. Since the boards correspond rather well to the logical elements that make up the system, the drawing also shows the elementary logical organization of a KS10-based DECSYSTEM-20. The system is made up of a number of subsystems, all of which are connected to a backpanel bus, which handles the movement of data among them. The minimal system has four subsystems: processor, memory, console, and an in-out subsystem based on a Unibus. The processor comprises the four boards in slots 11-14, and its connection to the bus is at the DPM board. The memory control board MMC and the memory array boards at its right make up the memory; essentially MMC is the interface between the backpanel bus and the memory array, as it is connected both to that bus and to a memory bus over which all transfers occur between control and memory. Similarly the Unibus adapter board UBA in slot 19 serves as

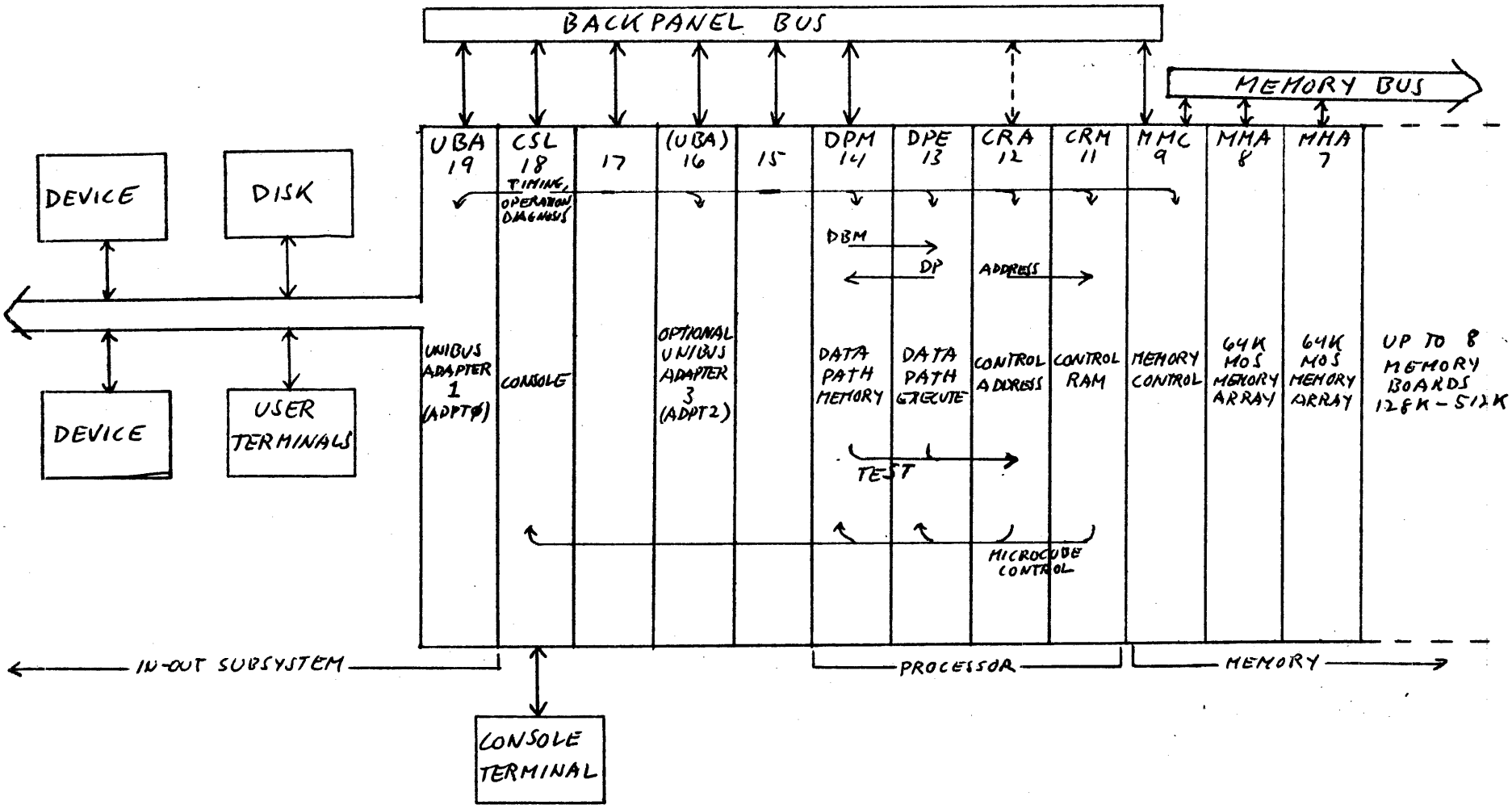


Figure 5.1 DECSYSTEM-2020 LAYOUT

the interface between the backpanel bus and the Unibus to the peripheral devices. A second adapter can be mounted in slot 16 for interfacing another Unibus; when this is done, adapter 1 handles the disks, and adapter 3 handles all of the other peripheral equipment.

There are of course many other interboard connections apart from the backpanel bus, especially among the four boards that make up the processor, and between the console and all other subsystems. The console board provides the connecting link to the operator via a terminal for all operating and diagnostic purposes. Moreover the console board contains the system clock and controls access to the backpanel bus: hence it supplies timing, arbitration, bootstrap and diagnostic signals to all other subsystems, including a myriad of signals to the individual processor boards (only the memory array boards lack direct connections to the console board, as they commmunicate solely via the memory bus with MMC).

The four boards that make up the processor are organized in two pairs. The data path boards, DPM and DPE, handle the execution of the instructions in the program under control of the microcontroller, which comprises the CRA and CRM boards. DPE contains the full word arithmetic logic, the program flags, the instruction register, and the RAM file containing the fast memory (AC blocks), the cache, and a

microcode workspace; DPM contains the 10-bit arithmetic logic for step counting and byte manipulation, the memory addressing logic, the cache directory, and the transceivers that connect to the backpanel bus. CRM is devoted solely to the microcode control RAM, about a third of which is on CRA, which also contains the microcode addressing logic. The DP boards supply many conditions that can be tested by CRA for sequencing the microcode; in particular each instruction code selects a location in a dispatch ROM that, although located on DPE, is part of the microcontroller. The selected dispatch word, together with the AC and index fields of the instruction word, provides CRA with information for dispatching to appropriate control RAM locations for calculating the effective address, fetching the operands, executing the instruction, and storing the result. These activities are carried out on DPM and DPE under control of the microinstruction bits supplied by CRA and CRM. Some bits are also supplied to CSL for handling console functions and to control the clock - its period can be lengthened whenever necessary for the data path operations.

The data path is in three parts. Contained entirely on DPE is the D bus, which supplies data to the arithmetic logic, the RAM file, and the instruction register. Via DP the output of the arithmetic logic is available to the D bus and to the various elements on the DPM board, including the

backpanel bus transceivers for transmission to memory or the I/O equipment. Data from various DPM elements, including words from memory and the DP data with its halves swapped, is available via DBM, which is an input to the mixer for the D bus.

The hardware on each board is shown in a set of circuit schematics, code CS. In every set, the final one, two or even three prints are devoted entirely to power and ground connections, capacitors, spare pins, and sometimes connector layouts and terminators. Each schematic has a number in the form W-X-YZ where W is the letter "M" followed by the board number, X is the revision number of the board, usually Ø, and Y is the three-letter mnemonic board designation, such as DPE or CRA. Z is a number or letter indicating the individual drawing in the series; the numbers are used first, followed by letters in alphabetical order when there are more than nine drawings. If a schematic is revised after being signed off by the engineer, a revision letter appears at the right of the drawing number. Throughout the text and in the block diagrams, individual prints are referenced simply by the YZ part of the drawing number.

The YZ designations are also used as prefixes in signal names to show the signal source. The board designations (Y) are used in the names of signals that are generated on the console and supplied to the other boards. Signals for

Unibus adapters 1 and 3 are designated by the mnemonics shown in parentheses in Figure 5.1, namely ADPTØ and ADPT2. The adapter numbers 1 and 3 are actually the subsystem numbers used for addressing the adapters over the backpanel bus.

5.1.1 Processor Logical Organization

The bold line across the center of Figure 5.1A divides the two major functional areas of the processor: the data path above, the microcontroller below. The dashed lines in the figure are the boundaries of the individual boards.

The basic data loop on the DPE board is made up of the D bus, the arithmetic logic, and the main data path DP. This is not a trivial loop, as the arithmetic logic, which is based on ten Am2901 microprocessor slices, contains not only an adder and a shifter, but an entire register file that holds the arithmetic registers, the program count PC, the current instruction, general purpose registers, and various constants and control words. Words on the D bus can also be saved in the cache or the AC's in the RAM file from which they are available for later use. The basic loop can be expanded by the inclusion of DBM and the logic associated with its mixer on the DPM board. To operations on full words, this extension of the path adds computations on exponents, manipulation of bytes within words, swapping the

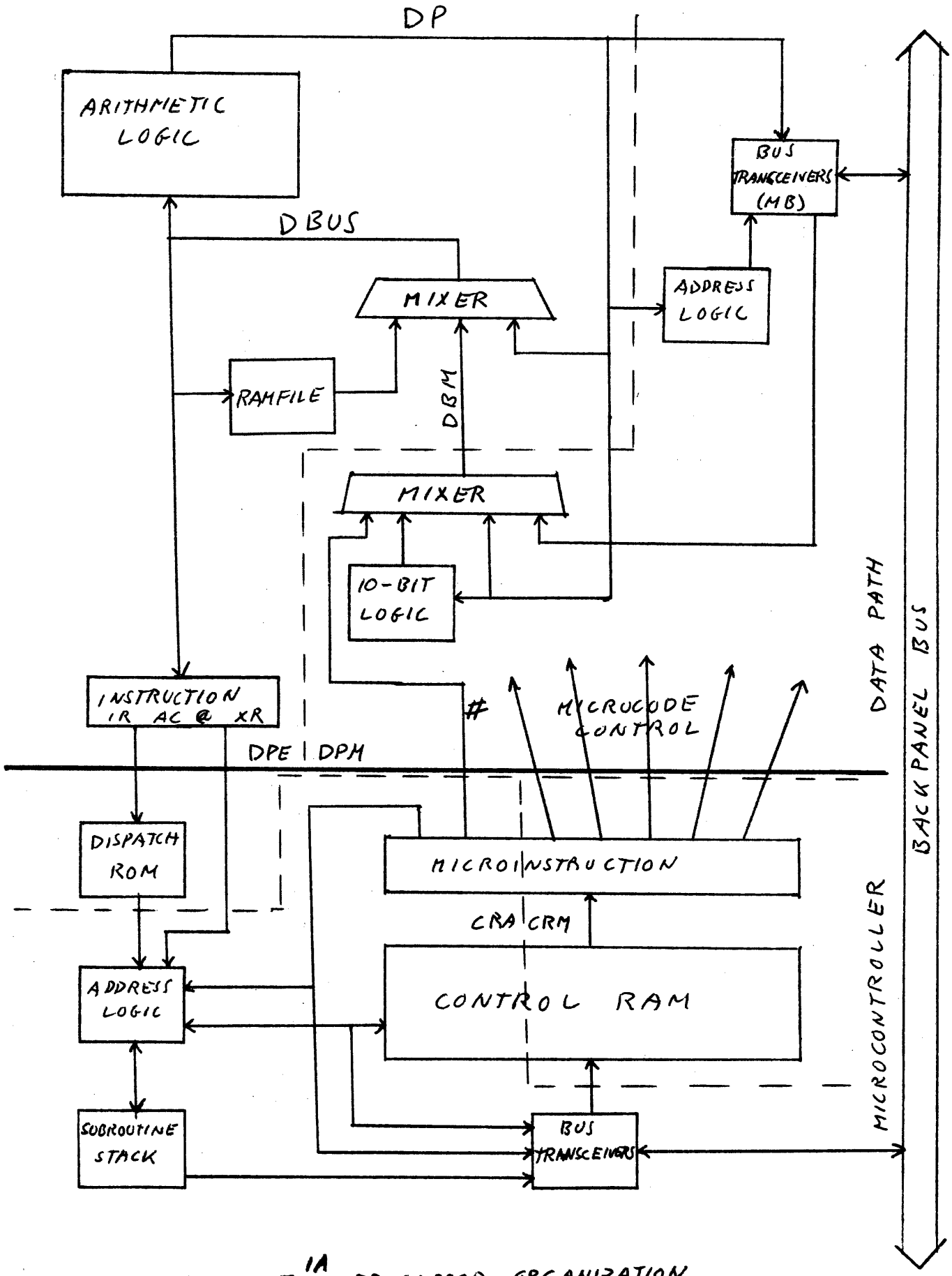


Figure 5.2[1A] PROCESSOR ORGANIZATION

two halves of a word, and operations using the number field of a microinstruction word. DP also supplies data and addresses to the bus transceivers for transmission over the backpanel bus to other subsystems. Data from other subsystems passes through the transceivers and is held in the memory buffer MB, from which it is available to the data path via DBM.

The microcontroller is also organized in a loop in that the addressing logic selects a location in the control RAM, and the contents of that location via the microinstruction register supply information back to the addressing logic for selecting the next location. This information includes not only an actual address but specification of conditions for skipping or dispatching and commands to call or return from a subroutine, for which the addressing logic employs a stack.

Together the two functional parts of the processor also form a control loop. The 96-bit microinstruction register not only supplies a number field as a quantity to the data path, but its individual bits feed a multitude of control lines that govern operations in every nook and cranny of the data path. In the opposite direction the data path supplies various conditions to the microcontroller address logic for skipping and dispatching, but principally it supplies the individual program instruction words, taken from memory for

execution by the microcontroller.

Following power turn-on, the console boots the system by using some of the backpanel bus data lines and transceivers on the CRA board to load the microcode into the control RAM in 12-bit segments. Following loading and starting by the console, the microcode initializes the machine, setting up various constants and tables that it stores in a workspace in the RAM file, and it then enters the halt loop in which it can respond to commands from the console. There are various general procedures that the microcode executes, procedures separate from actually executing program instructions, but in all cases these are associated in some way with the running of a program. To execute program instructions, microwords in the microinstruction register control the data path to determine the address for retreival of the instruction, save that address plus one as the program count in the PC location in the register file, send that address to memory, and upon receipt of the instruction word, place its left half in the instruction register. From this register the AC, index and indirect fields are available to the microcontroller address logic, and the instruction code selects a location in the dispatch ROM, which in turn supplies a dispatch word. Using these quantities the address logic sequences the microcode, through skipping and dispatching, to control the data path to carry out all of the operations necessary to execute the

instruction.

Figure 5.1B shows the processor data path in greater detail. The ten 2901s actually form a 40-bit arithmetic unit, with two extra bits at each end. In general these extra bits (in the register file and the adder) are disabled or ignored, but the processor does make use of the extra output bits at the left end of the adder for various sign manipulations. The adder operates on two words supplied by the A and B mixers, which in turn can receive (in various combinations) a word from the D bus, one or two words from the register file (selected by the A and B addresses), zero, or a word from the Q register. The adder output can go to any register in the file, direct or shifted one place left or right. The word in the Q register can also be shifted either way, or it can be replaced by the adder output.

The DPM address logic includes a virtual memory address register VMA, a page table and the cache directory. Addresses to VMA and mappings to the page table are both supplied via DP. For a virtual memory reference, a match of the address in VMA with the address in the directory indicates a cache hit, and the reference is made to the cache part of the RAM file rather than going to storage. When the reference is to storage, the page table supplies bits 16-26 of the physical memory address and VMA supplies the rest. Associated with VMA are various flags for

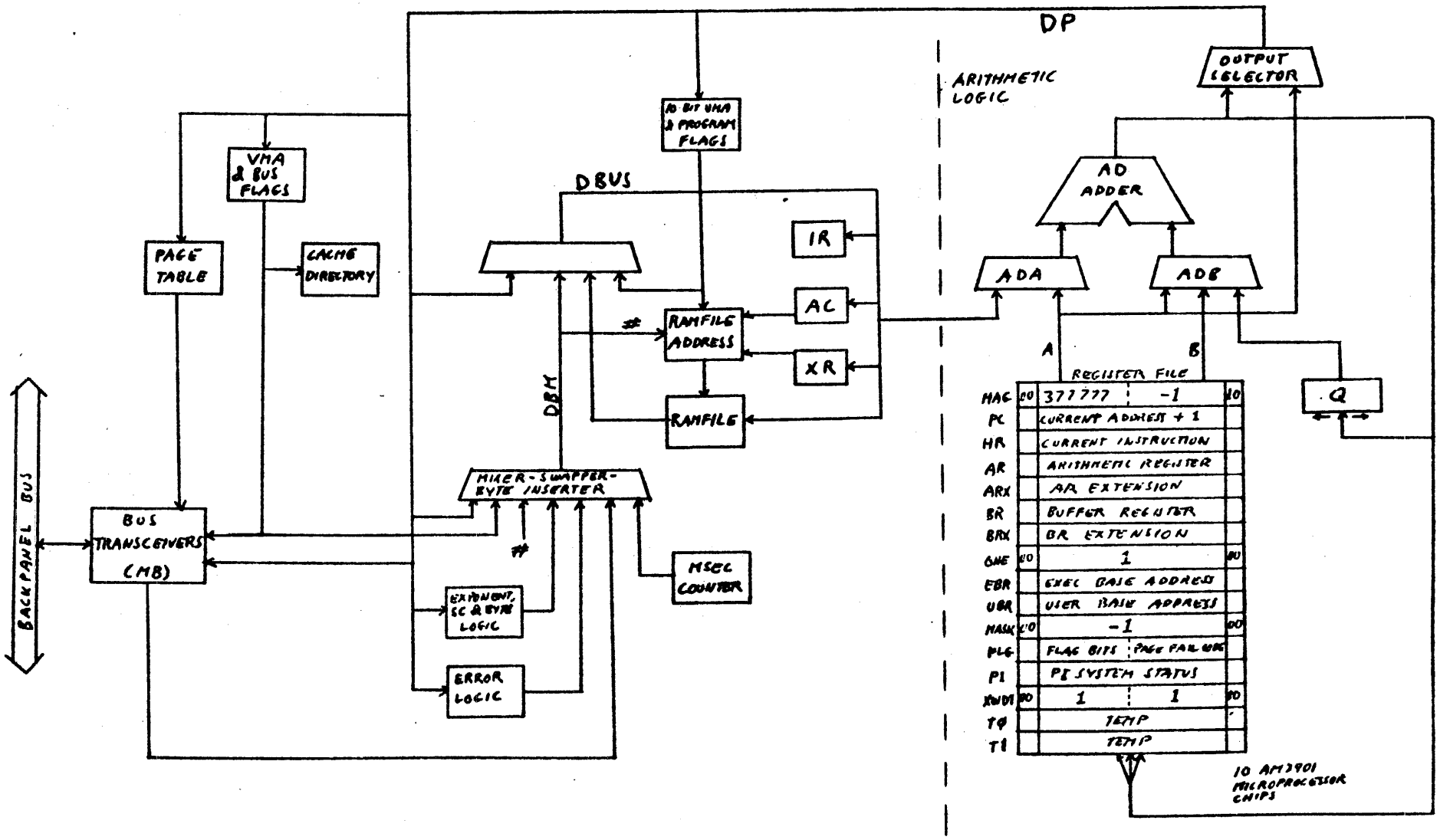


Figure 5.2[18] PROCESSOR DATA FLOW

specifying the type of transaction for which the backpanel bus will be used (VMA supplies device addresses for I/O instructions as well as memory addresses).Both VMA and the flags are available to the data path via DBM. Other logic associated with the DBM mixer provides not only the functions already mentioned, but also error and diagnostic information, and a reading of a 10-bit counter that does a 4096 count in each millisecond.

Whenever VMA is loaded from DP, a copy of the right ten bits is kept on the DPE board, for use along with AC, XR and the microword number field in addressing RAM file locations. This copy and associated program flags are directly available to the D bus.

5.1.2 System Timing

The CSL board has an oscillator and divider that generates two 13.33 MHz clock trains, the T and R clocks (transmit and receive), whose relationship is shown by the top two lines in Figure 5.12. These clocks are supplied via the backpanel to the other boards; all boards use the T clock, but the R clock is used only by those connected to the bus. From the basic clock supplied by CSL, each board derives its own clock signals, many of which are gated to produce ticks only when particular conditions are met. The clock is shown on print CSL1, and the clock circuitry for the various boards

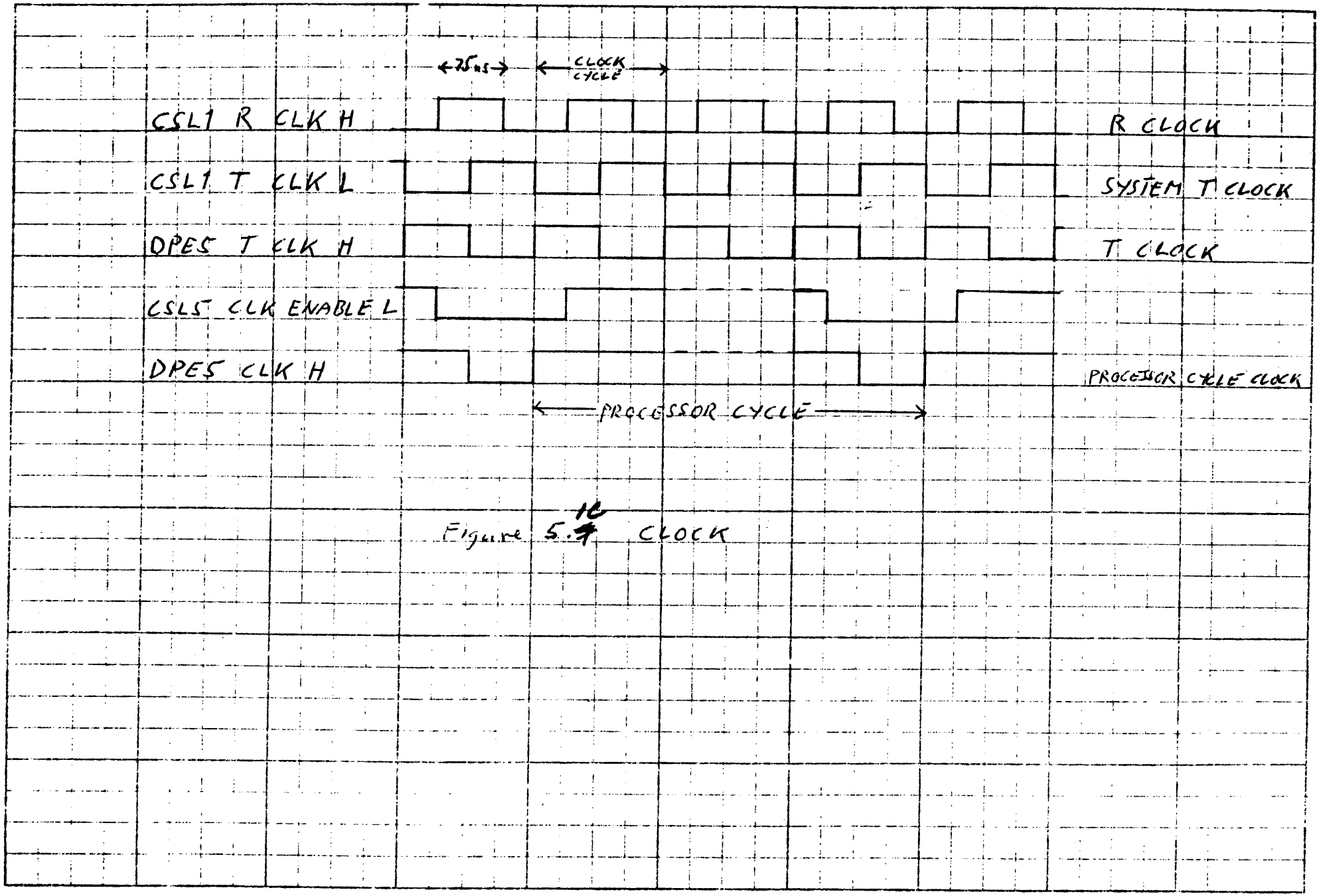


Figure 5.1C CLOCK

can be found on CRA2, CRM2, DPE5, DPMA, MMCC, CSL9 (T clock), CSLA (R clock), UBA1 (T clock) and UBA6 (R clock).

On the processor boards the clock is gated to define the processor cycle, which is a set of clock periods used for the execution of a single microinstruction. The cycle is defined by a CSL clock enable that passes a single tick of the T clock to terminate one cycle and begin the next. This is shown in the lower part of Figure 5.1C, where the nomenclature applicable to the DPE board is used as an example. The top gate in the clock circuitry at the left on DPE5 passes an entire T clock train, inverted but otherwise unchanged, which is available for operations apart from the actual microinstruction execution. The other gates generate seven DPE5 clock signals, each of which is limited to a single tick of 75 ns duration that occurs while the enable from CSL is low. Similar circuitry on the other boards acts in the same way so as to synchronize all operations on the various boards. Throughout the text the basic T clock train supplied from CSL1 will be referred to as the "system T clock", and the phrase "T clock" will refer to the inverted system clock such as exemplified by the signal DPE5 T CLK H or any of the other clock signals equivalent to it. Those particular T clock ticks that determine the processor cycle, namely DPE5 CLK H and equivalent signals, will be referred to as the "cycle clock". Almost all operations in the system, including transmission over the bus, are

synchronized to a rising edge of the T clock. The R clock, or its inversion, is used almost exclusively for latching information received from the bus.

The processor cycle always encompasses at least two periods of the T clock, but the microcode can increase this by one to three periods whenever additional time is needed, such as for setting up the adder. The cycle can also be extended by various hardware conditions such as waiting for memory, but in any event the cycle clock always occurs at the end of a processor cycle. In some cases, cycle clocks are gated by conditions besides the enable so that a particular clock signal may not occur at all in a given cycle. For example there are separate gated clocks for the left and right halves of the arithmetic logic, so that operations can be performed on one half of a word without affecting the other. Even the T clock for a board may be gated, as in the case of the CRA board where the first tick in each cycle manipulates the subroutine stack and sets up various skip and dispatch conditions for possible testing by the microinstruction.

5.1.3 Intrasystem Data Flow

Every subsystem maintains one connection to the backpanel bus. The memory connection is at MMC, the processor connection, at DPM. The dotted line between CRA and the bus in Figure 5.1 does not represent a true bus connection.

With the rest of the system quiescent, the console can boot the microcode, and for diagnosing the microcontroller it can read the microcode, various addresses, and the outputs of the CRM parity nets. for these operations the console "borrows" a dozen bus data lines, but no use is made of the arbitration logic or the bus control lines - CSL controls all events directly via other backpanel signals.

Any subsystem except memory can request access to or use of the bus. The memory is limited to responding to memory requests over the bus from other subsystems. A subsystem that gains access to the bus becomes the bus master, and the unit it addresses is the slave. To gain access the unit makes a bus request to the bus arbitrator, which is located on CSL (the logic is shown in the upper right of CSL1 and is discussed in detail in Section 5.7.2). At the completion of the current bus operation (if any), the arbitrator grants the bus to the requesting unit. If there is more than one request up at the same time, the grant goes to the unit of highest priority in the order console, adapter 1, adapter 3, processor. Upon receiving a grant, the master typically uses the bus for a command/address cycle, in which the master selects which unit is to be the slave (unless this is predetermined by other circumstances) and supplies the slave with information as to the type of data transfer to be made and the address of the memory location or peripheral device. When the master does execute a command/address cycle, the

arbitrator automatically grants a second cycle to the same unit for the data transfer. The easiest way to understand how the bus works is to consider the various kinds of transfers that can be made between different subsystems. Except where specifically indicated otherwise, bus access always involves two cycles, one for command/address and one for data.

The processor can communicate with memory over the bus for read or write access to the memory array. Upon receipt of the command/address cycle, MMC puts up a memory busy signal that holds up the bus until the data transfer is ready. The second bus cycle (for the data) then occurs when the read data has been retreived from the array or the microcode makes the write data available to the bus. In the write case the data cycle usually follows immediately after the command/address cycle.

An adapter can gain read or write access to memory as a nonprocessor request in the same way the processor makes a data access to memory (in the write case there is never a wait - the adapter always has the data ready). Moreover an adapter can also gain read-pause-write access, for inserting a byte into a memory word. In this case the memory holds the bus busy until two data cycles have occurred.

In response to a PI request, the processor does a command/address cycle that specifies an interrupt level and asks for the number of the subsystem (adapter) that is requesting an interrupt on that level. The appropriate adapter sends back its number during the following data cycle.

The processor can select a specific adapter either to get an interrupt vector (from one that has requested an interrupt) or to do an I/O instruction. For an output instruction the data from the processor follows immediately, freeing the bus, but an I/O busy signal from the adapter causes the processor (microcode) to wait until the data has been sent on to the addressed device. For an interrupt vector or an input instruction, the next consecutive bus cycle is thrown away, and the bus is then free for use by other subsystems while the I/O busy signal causes the processor to wait for the desired information.

An adapter can use the bus for a single I/O data cycle in which it sends an interrupt vector or data to the processor in response to a previous request.

The processor can do an I/O instruction to memory to read or write memory status. Although the data cycle always follows the command/address cycle immediately,

the memory asserts I/O busy on a status write and does the read status as an I/O data cycle because the procedure at the processor is handled by the same microcode that does I/O instructions to the adapters.

The processor can do an I/O instruction to read the console register. The data is returned immediately in an I/O data cycle.

Via the bus the console can communicate with any other subsystem: it can gain data access to memory for an examine or deposit; it can do an I/O instruction to memory or an adapter (which in the latter case may elicit a later I/O data cycle); and it can order the processor to get a word from the console register (via the bus) and execute it as an instruction.

Note that except for an I/O data cycle from adapter to processor or console, each bus transaction requires an initial command/address cycle. In some cases a data cycle may be thrown away, or the bus may be tied up for an extended period between the command/address cycle and completion of the transaction. If the arbitrator grants the bus to a subsystem that does not use it, the bus immediately becomes free again. Such an event may occur for example when the processor requests the bus for access to memory but then decides that it is an AC reference or the word is in

the cache.

The data connections to the bus for processor and memory, as shown in DPM8,9 and MMC1, are quite symmetrical in appearance: for each half word there are five 8646 4-bit transceiver latches for the 18 bits plus even parity. Equivalent logic for the other subsystems is on CSL6-8 and UBA8-C, where the latter group of prints also shows the Unibus transceivers and the interface between the two buses. The transceiver chip contains flip-flops for placing transmitted data on the bus and latches that serve as a buffer for received data. A low level at the TRN enable input of an 8646 allows a rising edge of the T clock at the TRN clock input to load the data at the TRN data inputs into the flip-flops that hold the data on the bus data lines. The data is held for one clock cycle of 150 ns, and a high level at TRN enable allows the next T clock to clear the flip-flops. During the transmission cycle, the receiving subsystem brings the REC latch inputs to its transceivers low to hold the data for its use. Latching occurs at the falling edge of the R clock or its inverse. Command/address information must be used during the transmission cycle; it may therefore need not be latched at all, or may be latched only by the low half of an R clock straddling the rising T edge that terminates the transmission cycle. On the other hand a subsystem usually requires that received data be available for an extended period. If it has no other use

for the bus during the interim, the receiving subsystem uses the latches as an ordinary register (e.g., they serve as the memory buffer MB in the processor) by having an extended signal hold the latch inputs low once latching has been accomplished by the R clock. Note that each bus transfer requires just one clock cycle, but each bus access requires at least two - one for handling the grant and one for making the transfer. However if the master gives a command/address cycle, a subsequent data cycle requires only one additional clock cycle unless there is a wait.

Each 8646 also detects odd parity for the 4 bits transmitted or received, and these signals are applied to standard parity nets to generate an even parity bit for the half word transmitted or to check the parity of the half word received. Note that a signal that is true when the parity of a group of bits is odd serves as an even parity bit for that same group. Besides data and parity the bus handles a number of control signals, not all of which are generated or used by all subsystems; there are for example more signals passed over the bus between console and adapter than between processor and memory. For most control signals transmission is permanently enabled (see DPMC or MMCB), and some received signals are not even latched - the REC latch input is held high permanently. ~~Table 5.1 explains the bus signals and lists the backpanel pins on which they appear (all bus signals are prefixed by the term "BUS").~~ Refer to Subsection 5.2 for a more detailed description of bus operation.

5.2 KS10 (BACKPLANE) Bus

The KS10 bus is a synchronous backplane bus internal to the KS10 Processor that provides a control and data path between the console, CPU, memory, and I/O controllers. (The only I/O controllers currently on the bus are the two UBAs.) The bus can accomodate another I/O controller (to allow for future expansion) and it performs the following major functions:

1. Memory Data Transfer - Transfers data to/from MOS memory via the memory controller under the control of the CPU, console, or a UBA (NPR data transfers).

2. I/O Register Data Transfer - Transfers data to/from I/O device registers under control of the CPU or console. An I/O device is considered any device external to the CPU. Thus, not only are the Unibus devices connected to a UBA considered to be I/O devices, but also the UBA itself as well as the memory (controller) and console.

3. PI Handling - Transmits PI requests generated by the UBAs and transfers the interrupting controller (UBA) numbers

and interrupt vectors from the UBAs to the CPU under control of the CPU.

4. System Synchronization - Provides a continuous clock train that is used by all bus devices to sequence logic and to synchronize operation with the rest of the system.

5. System Reset and Power Fail Indicator - Allows the console to reset the system and to signal ac power failure to the devices on the bus.

The KS10 bus data path is 36 bits wide. There are two parity bits associated with the data lines, one for data lines 0-17 and one for data lines 18-35. Each device checks for correct (even) parity when it receives information over the bus. If bad (odd) parity is detected, the CPU clock is stopped.

The number of control lines on the bus is minimized in that command/address information is transmitted over the data lines in addition to memory and I/O register data. For example, if a device is to write memory, it asserts command bits and the memory address on the data lines for one bus cycle. This cycle is called a command/address cycle. Then, during a following bus cycle, it transmits the 36-bit data word to be written in memory. This cycle is called a data cycle.

Before any device can transfer information over the KS10 bus, it

must first request and then be granted the bus. There is a bus request line and a corresponding grant line for each device. The bus arbitrator, located on the console module, monitors all requests, resolves request priority, and (whenever the bus is free) grants the bus by asserting the grant line for the highest priority device.

KS10 bus signals and information flow are shown in Figure 5-2. Bus signals are terminated at both ends of the wire run (Z=120 ohms). The majority of signals are terminated at the console module on one end and at the memory controller at the other end. Bus logic levels are as follows:

Logic Level	Voltage
0	+3.4 V
1	0V to +0.8 V

Table 5-1 summarizes the functions of the various signals on the KS10 bus.

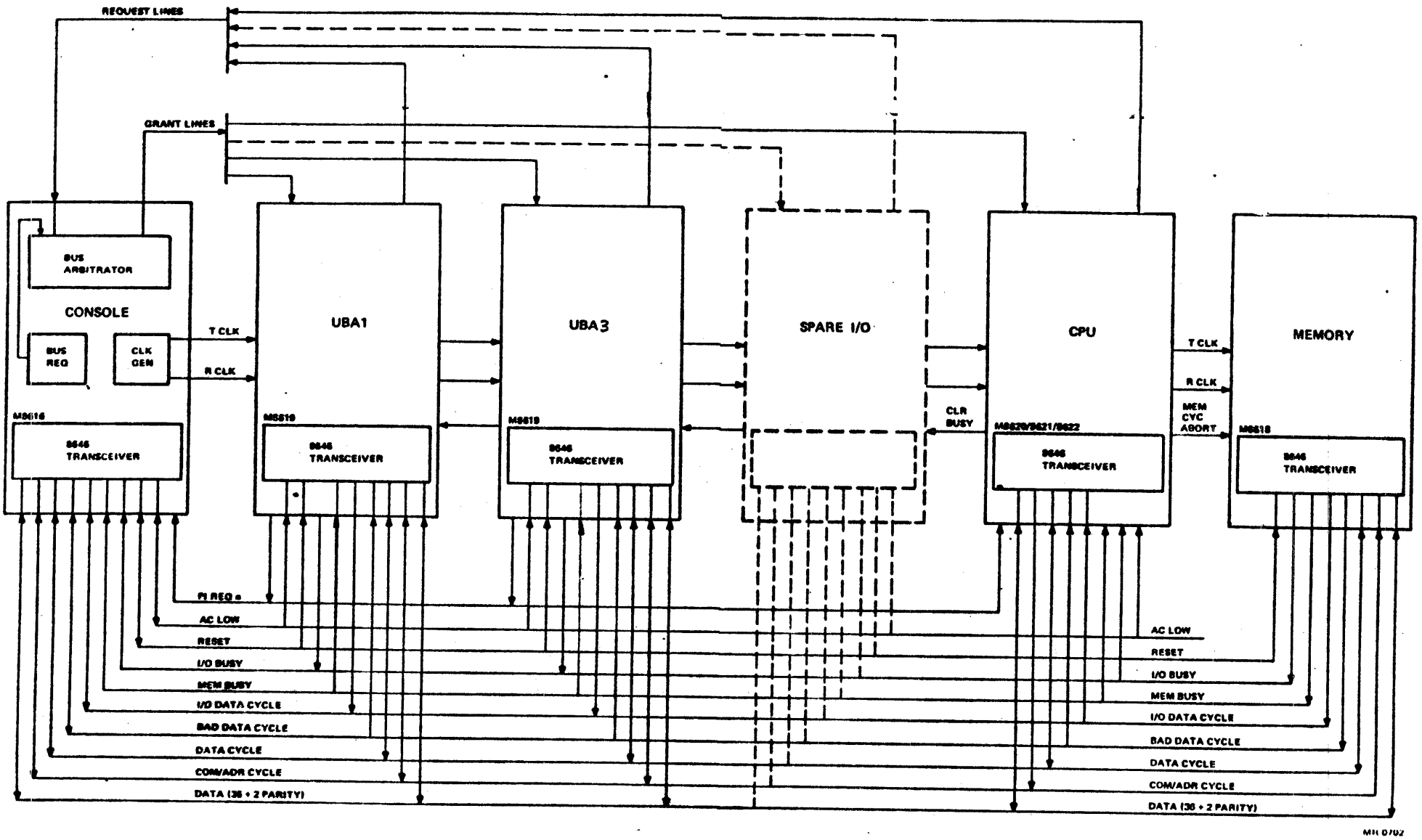


Figure 5-2 KS10 (Backplane) Bus

Table 5-1 KS10 Bus Signal Summary

Signal	Description
T CLK	6.66 MHz continuous clock generated on console module. Leading edge defines start of bus cycle. Used to clock data and control signals transmitted on bus.
R CLK	6.66 MHz continuous clock generated on console module. Used to latch data and control signals received on bus.
REQUEST (one per device)	Asserted by device requesting bus.
GRANT (one per device)	Asserted by bus arbitrator when device requesting bus has been granted the bus. (Device becomes bus master.)
COM/ADR CYCLE	Asserted by bus master when transmitting command/address on data lines. Asserted for one bus cycle.
DATA CYCLE	Asserted by bus master when transmitting memory write data or I/O register write data

on the data lines. Asserted by memory controller when transmitting memory read data on data lines. Asserted for one bus cycle.

BAD DATA CYCLE — Asserted by memory controller when transmitting uncorrectable memory read data on the data lines. Asserted for one bus cycle coincident with DATA CYCLE.

I/O DATA CYCLE — Asserted by bus device when transmitting I/O register read data or an interrupt vector on the data lines. Asserted for one bus cycle.

MEM BUSY — Asserted by the memory controller after receiving memory read, write, or read-pause-write command. Negated when memory is ready to accept another command. This signal disables bus arbitrator.

I/O BUSY — Always asserted by addressed bus device after receiving an I/O register write command. Also asserted by addressed bus device after receiving an I/O register read command (or a read interrupt vector command) when the device is NOT going to supply the register read data (or the vector) during the bus cycles alloted the bus master. (The bus device requests the

	bus and generates a data cycle to transfer the data at a later time.)
CLR BUSY	Asserted by CPU (after a time-out) to negate I/O BUSY in UBA after a nonexistent device register has been referenced.
MEM CYC ABORT	Asserted by CPU to terminate memory reference (cache hit or AC reference).
PI REQ n (n=1-7)	Asserted by UBAs to request CPU priority interrupt on channel n.
DATA 00-35	Bidirectional data lines used to transfer command/address and read/write data between devices on the bus.
PARITY LEFT	Transfers computed (even) parity for data lines 00-17.
PARITY RIGHT	Transfers computed (even) parity for data lines 18-35.

5.2.3.1 **Bus Timing** - T CLK and R CLK are system clocks generated on the console module and distributed to all devices on the bus. T CLK is used to transmit data and control signals on the bus; R CLK is used to receive data and control signals on the bus. Both clocks have a 150 NS period. Timing relationship is shown in Figure 5-3.

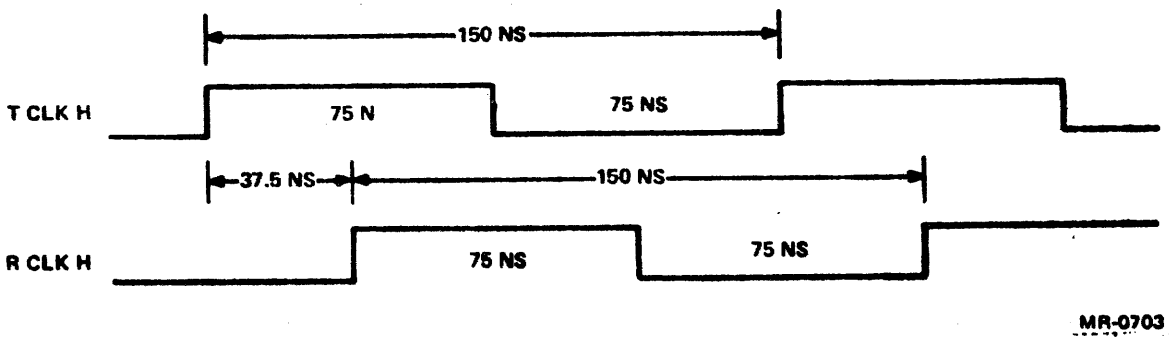


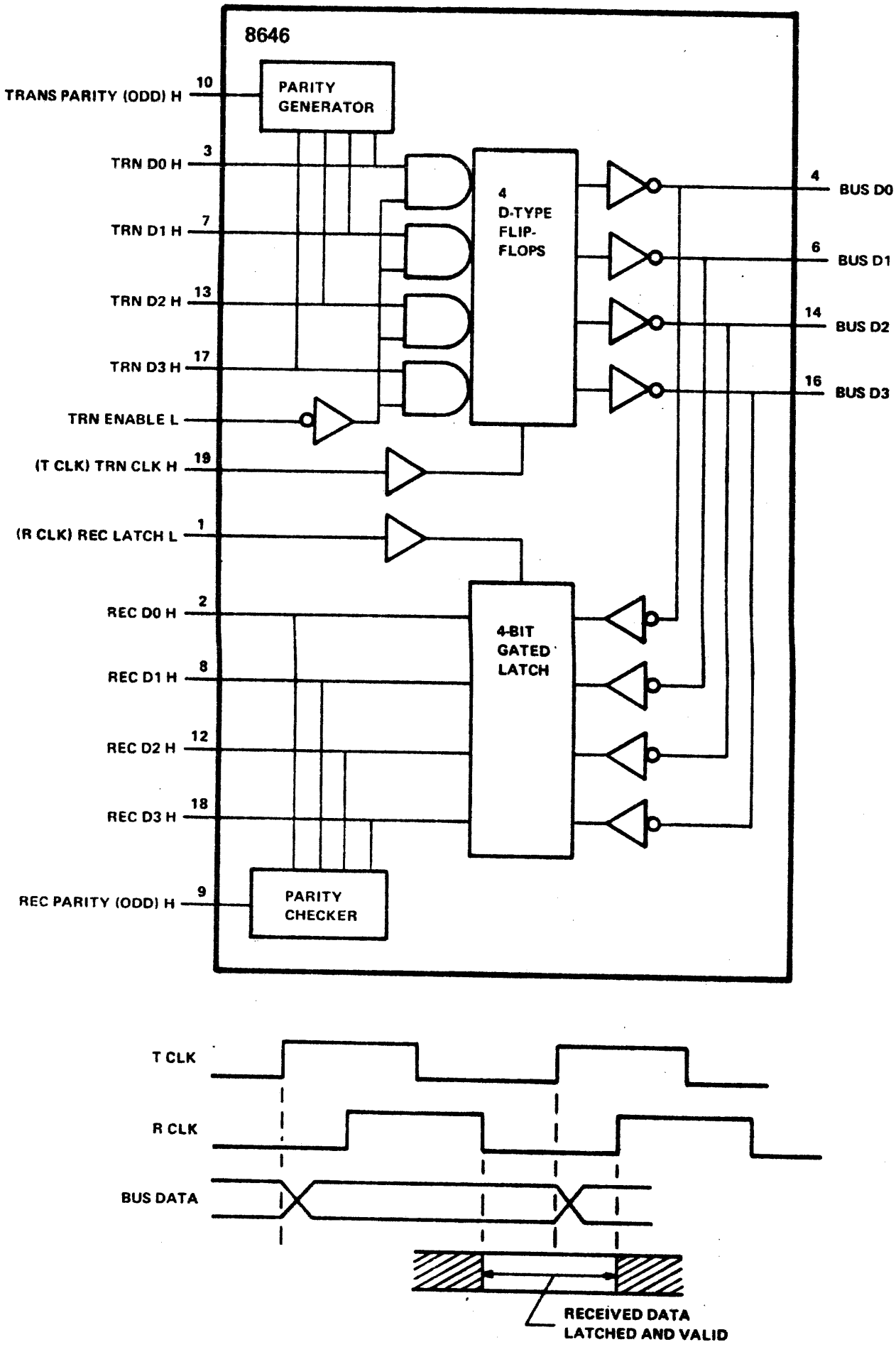
Figure 5-3 T CLK/R CLK Timing Diagram

5.2.3.2 **8646 Bus Transceiver** - System modules connecting to the KS10 bus use 8646 transceiver latch circuits to transmit and receive information on the data lines and a majority of the control lines. Each 8646 can transmit and receive four bus signals. In addition, the circuit determines parity for both input and output data.

A circuit schematic for the 8646 is shown in Figure 5-4. In KS10 devices, T CLK connects to the TRN CLK input, and R CLK connects to the REC LATCH input. If the TRN ENABLE input is true (low), input data is clocked by T CLK into D-type flip-flops and asserted on the bus. This data is asserted until the next T CLK (150 NS later) when the flip-flops are clocked again. To negate all four transmitted outputs, the TRN ENABLE input is made false (high) causing the next T CLK to load 0s in the flip-flops. Bus signals not transmitted by a device (only received) have the corresponding 8646 inputs permanently false (wired to ground).

The 8646 uses a 4-bit latch circuit to receive and buffer information on the bus. When the REC LATCH input (R CLK) is false (high), the latches remain open and the data currently on the bus at the latch inputs is asserted at the 8646 data output pins. When R CLK drives the REC LATCH input true (low), the bus data is latched and the data output pins will not change for the duration of the clock (75 NS). During this time, the latched data may be gated and clocked by T CLK and the next R CLK in the bus device, even though bus data is changing at the latch inputs.

Bus transceivers that connect to the KS10 bus data lines utilize the internal parity generator and parity checker. Little additional logic is required in a device to generate the two bus parity bits (PARITY LEFT and PARITY RIGHT) transmitted on the bus, or to check the parity of the entire 36 bits of data received on the bus.



MR-0704

Figure 5-4 8646 Bus Transceiver

5.2.3 Bus Arbitration - A device may request the bus at any time by asserting its REQUEST line at the leading edge of T CLK, the start of a bus cycle. The bus arbitrator on the console module will then grant the requesting device the bus by asserting the device GRANT line whenever the bus is free and if a higher priority device is not also requesting the bus. Bus priority, highest to lowest, is as follows:

1. Console
2. UBA1
3. UBA3
4. CPU

NOTE

The memory controller does not make bus requests.

Assuming there are no higher priority requests and the bus is not already being used by another device, the GRANT signal will be asserted by the arbitrator during the same bus cycle that REQUEST is asserted. When GRANT is received by a device, the device negates REQUEST (at leading edge of T CLK) and assumes control of the bus as bus master. With reference to Figure 5-5, the device then has the bus for the next two cycles, three cycles, or an unspecified number of cycles depending on what action it takes during the first cycle. A bus master may do the following, each affecting the arbitrator in a different way:

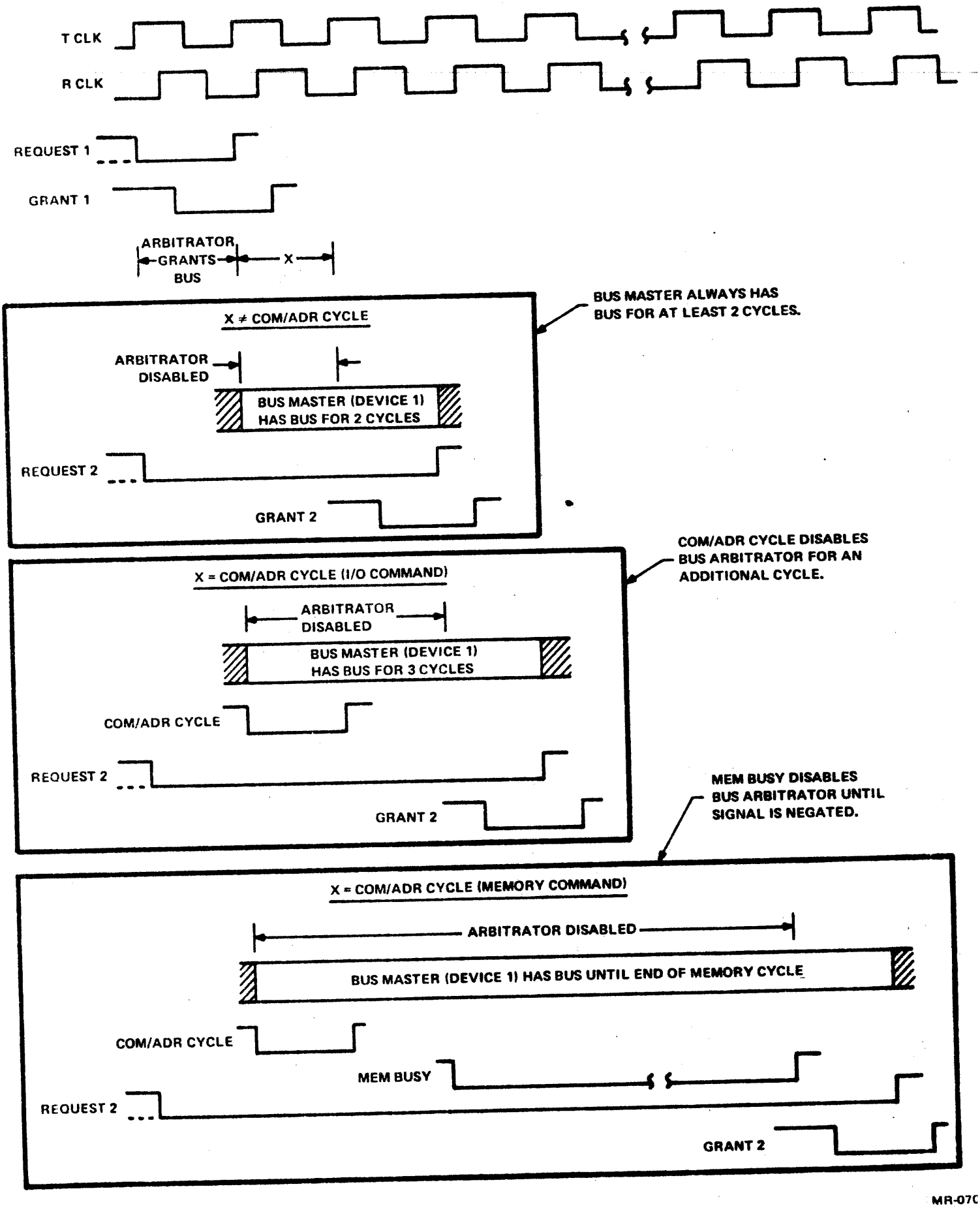


Figure 5-5 Request/Grant, Bus Timing Diagram

1. No Command/Address Cycle - Bus arbitrator grants bus for two cycles. Actions taken by the bus master for which the first cycle is not a command/address cycle are as follows:

   a. Data Cycle - I/O device requests and is granted the bus and then initiates a data cycle to send data to another device to complete an I/O register read operation or an interrupt vector read operation.

   b. No Action - Device requests and is granted the bus and then does not use the bus. Not using granted cycles is equivalent to giving up the bus; another bus request must be made to become bus master.

2. Command/Address Cycle (I/O Operation) - The arbitrator monitors the COM/ADR CYCLE bus signal and grants the bus master three bus cycles (an extra cycle) if the first cycle is a command/address cycle. When the command/address specifies an I/O operation, no additional bus signal will disable the arbitrator. All I/O operations, except those named above in 1(a) which require another bus request to transfer data, complete within the three allotted cycles.

3. Command/Address Cycle (Memory Operation) - As for a command/address cycle specifying an I/O operation, the

COM/ADR CYCLE signal causes the arbitrator to grant the bus master three cycles. However, when a memory operation is specified, the arbitrator is disabled during the third bus cycle by bus signal MEM BUSY. This signal is asserted by the memory controller in response to the command, and it is negated when the memory controller is ready to accept another command. Thus, the bus master has the bus for an unspecified number of cycles; that is, for the duration of the memory operation.

The operation of the bus arbitrator may be summarized as follows:

1. The bus master is always granted the bus at least two bus cycles.

2. If the first cycle is a command/address cycle, the bus master is granted the bus at least three cycles.

3. If the bus master initiates a memory operation, it is granted the bus until the operation completes.

5.2.4 Bus Usage - As discussed in Subsection 5.2.3, a device may do the following after it has been granted the bus:

1. Not use the bus.
2. Initiate a data cycle.
3. Initiate a command/address cycle.

For the current KS10 configuration and barring a malfunction, the only device that does not use the bus after a bus request is made is the CPU. To save time, the CPU always requests the bus for every memory reference. Then, if there is a cache hit or if the reference is to an AC, MOS memory need not be referenced and the CPU initiates no bus action.

NOTE

In some cases, a command/address cycle may be generated before a cache hit is detected. The CPU then asserts MEM CYC ABORT to terminate MOS memory operation.

The only device that does a data cycle after becoming bus master (without first performing a command/address cycle) is a UBA. The data cycle actually completes a previously initiated I/O register read operation by the CPU or console, or an interrupt vector read operation by the CPU. When first addressed, a UBA does not furnish register data or an interrupt vector within the three bus cycles allotted the device initiating the operation. Instead, the bus is requested again, this time by the UBA. When the bus is granted, a data cycle is generated to transfer the data.

A device normally uses the bus by first generating a command/address cycle. The command/address cycle, in turn, initiates one of the following eight bus operations:

1. Memory Write
2. Memory Read
3. Memory Read - Pause - Write
4. I/O Register Write
5. I/O Register Write (Byte)
6. I/O Register Read
7. Controller Number Read
8. Interrupt Vector Read

Table 5-2 lists the initiating and responding devices for each operation. For example, the first entry indicates that the console, CPU, and UBAs all write data into memory.

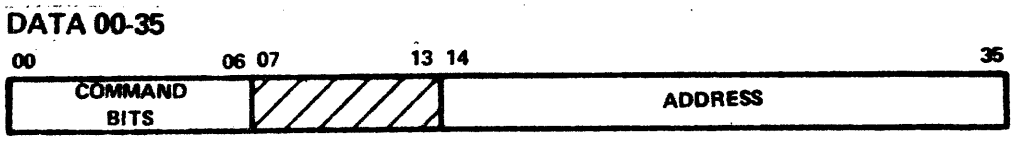
Table 5-2 Bus Operations

Operation	Initiated By	Directed To
Memory Write	CPU	Memory
	Console	Memory
	UBA	Memory
Memory Read	CPU	Memory
	Console	Memory
	UBA	Memory
Memory Read-Pause-Write	UBA	Memory
I/O Register Write	CPU	UBA
(byte operations directed	CPU	Memory
to UBA only)	Console	UBA
	Console	Memory
I/O Register Read	CPU	Console
	CPU	UBA
	CPU	Memory
	Console	UBA
	Console	Memory
Controller Number Read	CPU	UBA
Interrupt Vector Read	CPU	UBA

5.2.5 Command/Address Cycle - After being granted the bus, the bus master initiates a bus operation by transmitting a command/address on the data lines during the first allotted bus cycle. The bus master also asserts the COM/ADR CYCLE control line. COM/ADR CYCLE is monitored by the bus arbitrator to give the bus master an extra bus cycle (Subsection 5.2.4.3). Its principal function, however, is to cause the other devices on the bus to decode the transmitted command/address information. If addressed, a device will then respond to the specified command.

The basic command/address bit format on the data lines is shown in Figure 5-6. Data lines 0-6, the command bits, specify the bus operation to be performed; data lines 14-35 carry the address information specific to the command. The command/address bits are given in Figure 5-7.

The seven command bits (bit 3 is not used) specify the nine different bus operations in the following manner. Bit 0 determines whether the operation is a memory data transfer or an I/O data transfer; that is, bit 0 = 0 specifies a memory function and bit 0=1 specifies an I/O function. Bits 1 and 2, the read and write bits respectively, act in conjunction with bit 0 to further specify the type of operation. For example, if bit 0 = 0 and bit 1 (read) = 1, the operation is a memory read function. All three memory operations (read/write/read-pause-write) and the two I/O register operations (read/write) are specified by these first three command bits (0-2).



COMMAND BITS	FUNCTION
00	
1	I/O FUNCTION
0	MEMORY FUNCTION
01	READ (I/O OR MEMORY). BITS 14-35 SPECIFY ADDRESS.
02	WRITE (I/O OR MEMORY). BITS 14-35 SPECIFY ADDRESS.
03	NOT USED.
04	READ INTERRUPTING DEVICE NUMBER. BITS 15-17 SPECIFY PI CHANNEL.
05	READ INTERRUPT VECTOR. BITS 14-17 SPECIFY I/O CONTROLLER.
06	BYTE TRANSFER.

ADDRESS BITS	
14-17	I/O CONTROLLER ADDRESS
18-35	I/O REGISTER ADDRESS
14-35	MEMORY ADDRESS
15-17	PI CHANNEL NUMBER

MR-0706

Figure 5-6 Basic KS10 Command/Address Format

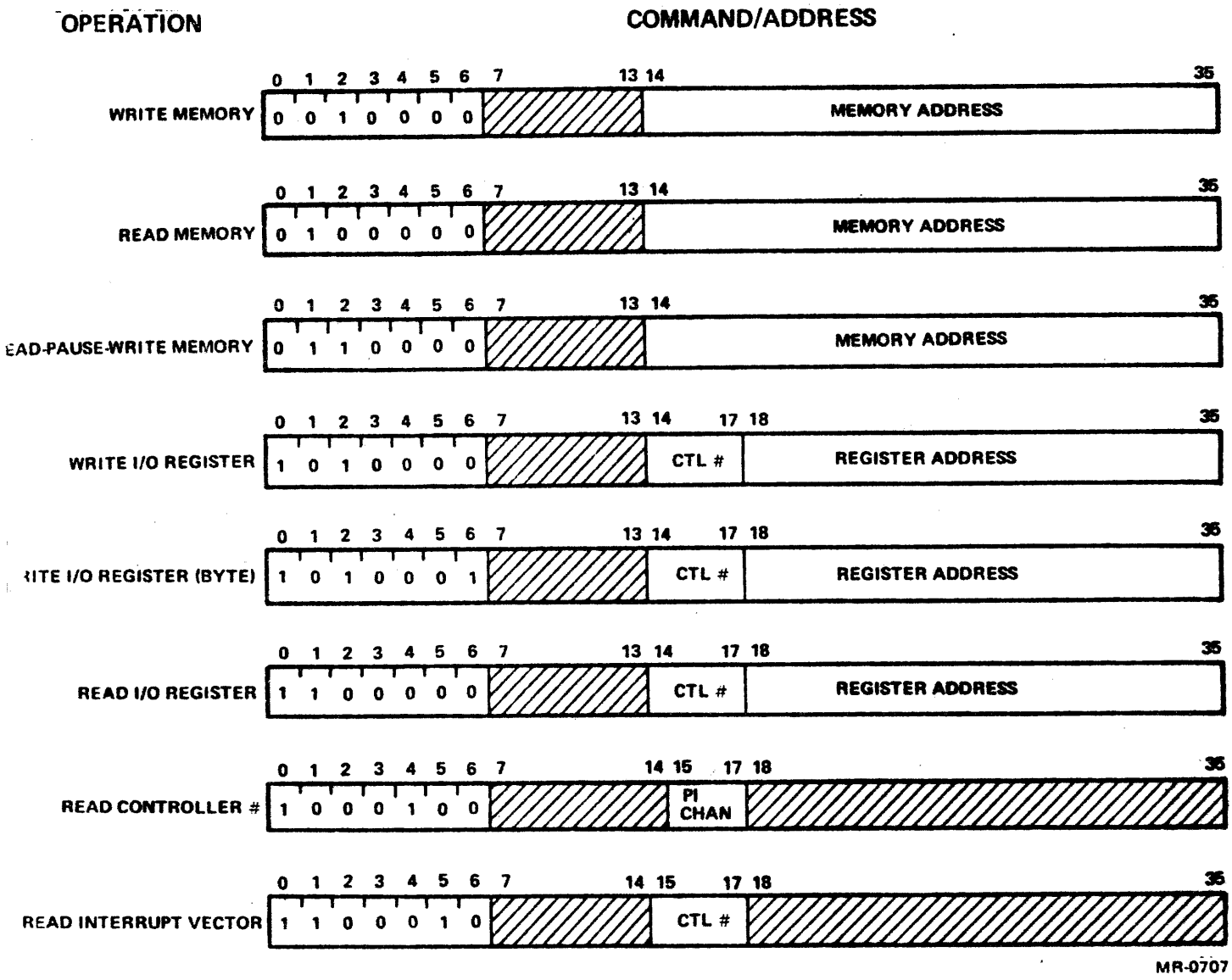


Figure 5-7 Command/Address Bits for KS10 Bus Operations

Bits 4 and 5 are used to specify the two PI operations performed over the KS10 bus. The CPU asserts one of the bits (bit 4 = 1) to read the interrupting controller number. It asserts the other bit (bit 5 = 1) to read the interrupt vector. Bit 0 = 1 for both PI operations. Bit 1 (read) = 1 for the vector read.

Bit 6, the byte transfer bit, has significance only for I/O register write operations that address Unibus device registers. Unibus devices allow full-word (16 bit) or byte (8 bit) transfers of register data and bit 6 is used to specify the transfer mode.

The 22 data lines (14-35) reserved for address information transfer either a memory address (bit 0 = 0) or an I/O address (bit 0 = 1). For memory functions, the least significant 20 bits of the address field are currently used (maximum memory configuration = 512K). For I/O register read/write functions, the I/O address consists of a controller number (bits 14-17) and a register address (bits 18-35). For the PI function that reads the interrupt vector (bit 5 = 1), only the controller number is significant. For the other PI function (bit 4 = 1), which reads the interrupting controller number, the I/O address consists of a 3-bit PI channel number (1-7) on data lines 15-17. The various KS10 I/O controller numbers and register addresses are given in Section 4 (Table 4-6) and Appendix C.

5.2.6 Bus Memory Operation - The CPU, console, and UBA all

reference MOS memory over the KS10 bus. Once granted the bus, the device making the reference first transmits a command/address on the data lines to specify a memory operation (bit 0 = 0), the memory address (bits 14-35), and the type of memory operation; that is, a read (bit 1 = 1), a write (bit 2 = 1), or a read-pause-write (bit 1 = 1, bit 2 = 1). When the memory controller receives the command address and the address is valid (in-bounds), it asserts MEM BUSY to freeze the bus arbitrator. The device making the reference then has the bus until the end of the memory operation, at which time, MEM BUSY is negated to unlock the arbitrator and allow the next bus operation to take place.

If the operation initiated by the command/address is a memory write operation, write data may be asserted on the data lines during any cycle following the command/address cycle (up to 7.5 usec maximum). The device making the reference initiates the data cycle by asserting the write data and the DATA CYCLE control signal. DATA CYCLE is used by the memory controller to strobe the write data from the bus and to start a memory write cycle. When the write cycle completes and the data has been stored in the MOS array, the memory controller negates MEM BUSY to end the operation. Bus timing for the memory write operation is shown in Figure 5-8.

If the operation is a memory read operation, the memory controller starts a memory read cycle after receiving the command/address. When data is read from the MOS array, it is transmitted on the

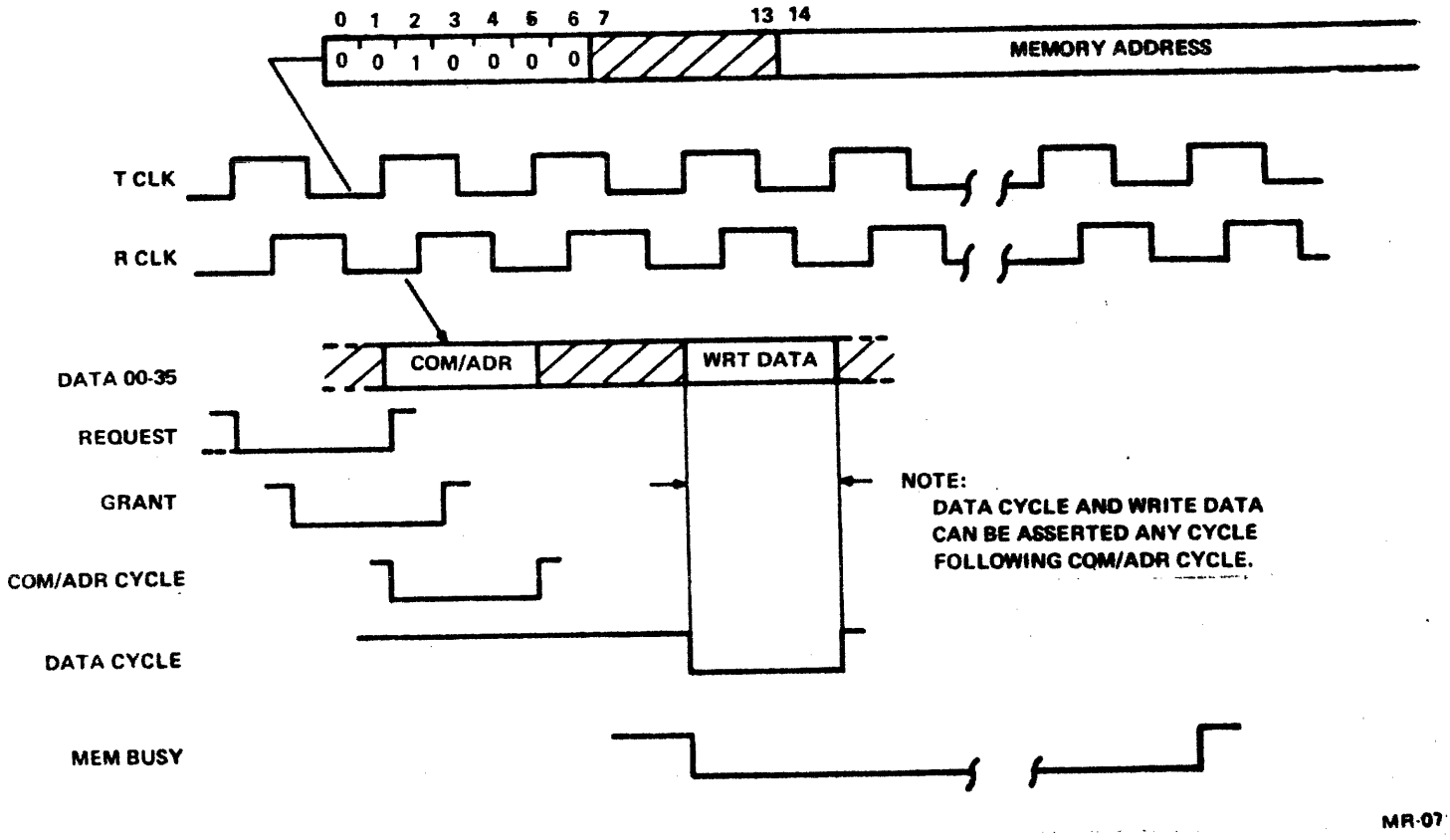


Figure 5-8 Write Memory, Bus Timing Diagram

data lines and the ECC (error correction code) is checked for error. If there is no error, the memory controller initiates a data cycle during the next bus cycle; that is, it continues to assert the data lines and it generates the DATA CYCLE control signal. DATA CYCLE acts as a data strobe (as for the write operation) and it is used by the device initiating the memory reference to gate the read data from the bus. When there is an ECC error, the memory controller attempts to correct the read data and delays the data cycle for one bus cycle; that is, the corrected or uncorrected data is transmitted for the next two cycles and DATA CYCLE is asserted during the second cycle. In either case, error or no error, the read data is on the data lines

for one full cycle before DATA CYCLE is generated. This is to allow extra propagation time before the data is gated and clocked in the CPU's 2901 microprocessor circuits. When the data read from the array is uncorrectable, the memory controller flags the data as invalid by asserting the BAD DATA CYCLE control line in addition to DATA CYCLE. Bus timing is shown in Figure 5-9.

During a memory read-pause-write operation, data is first read from the specified address with read data and DATA CYCLE asserted on the bus as previously described for the memory read operation. Then, following the read operation, the memory stays active (MEM BUSY = 1) and performs a memory write cycle when write data and DATA CYCLE are asserted on the bus as previously described for the memory write operation. The read-pause-write operation allows a device to read data from memory, modify it, and then write the modified data back into the same memory address all in one operation. Bus timing is shown in Figure 5-~~14~~.

5.2.7 Bus I/O Operation - The CPU and console read/write I/O registers over the KS10 bus. Both can access the I/O registers internal and external to the UBA as well as the memory status registers. In addition, the CPU can read the console instruction register.

NOTE

The console cannot read its own instruction register.

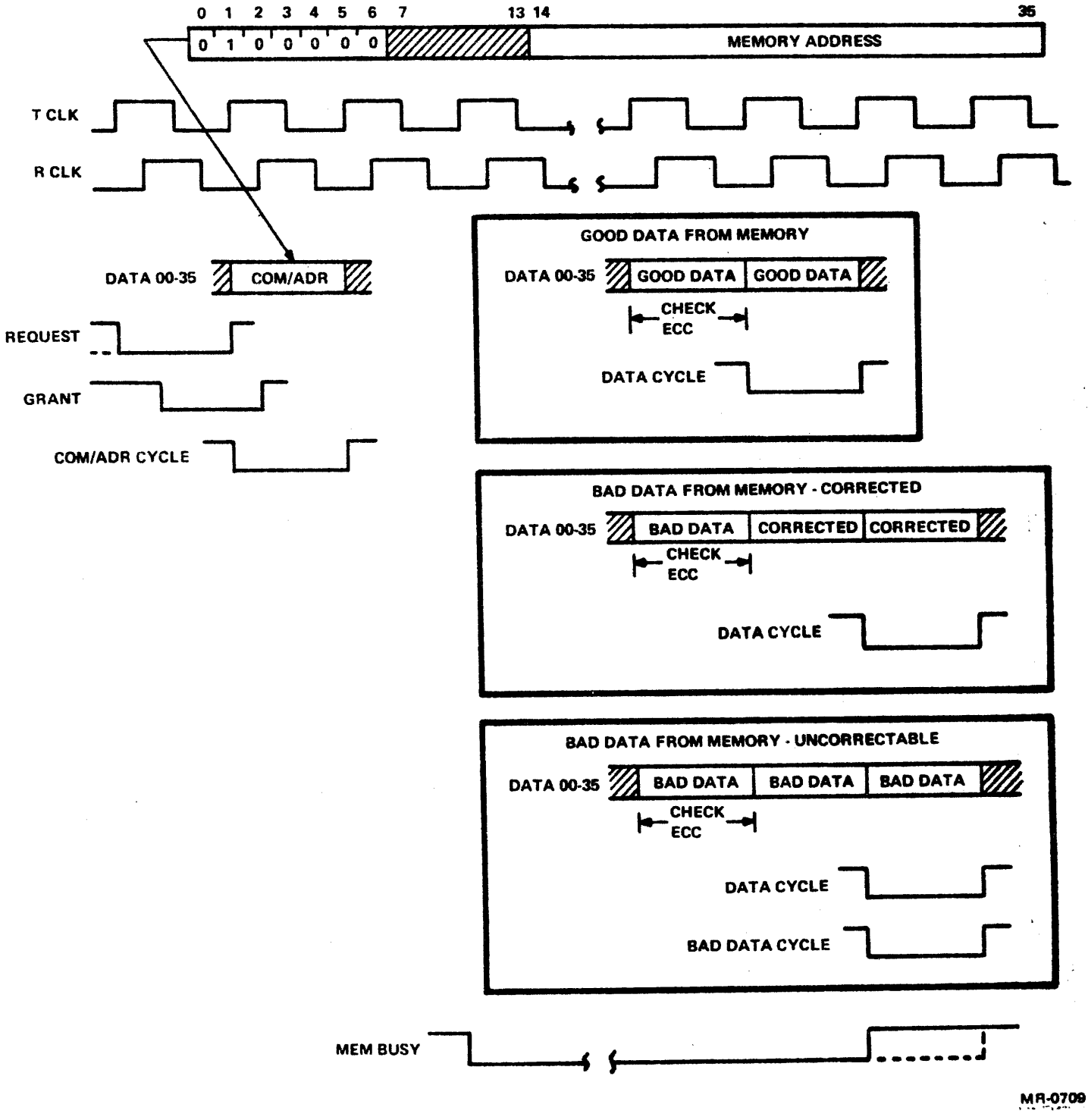


Figure 5-9 Read Memory, Bus Timing Diagram

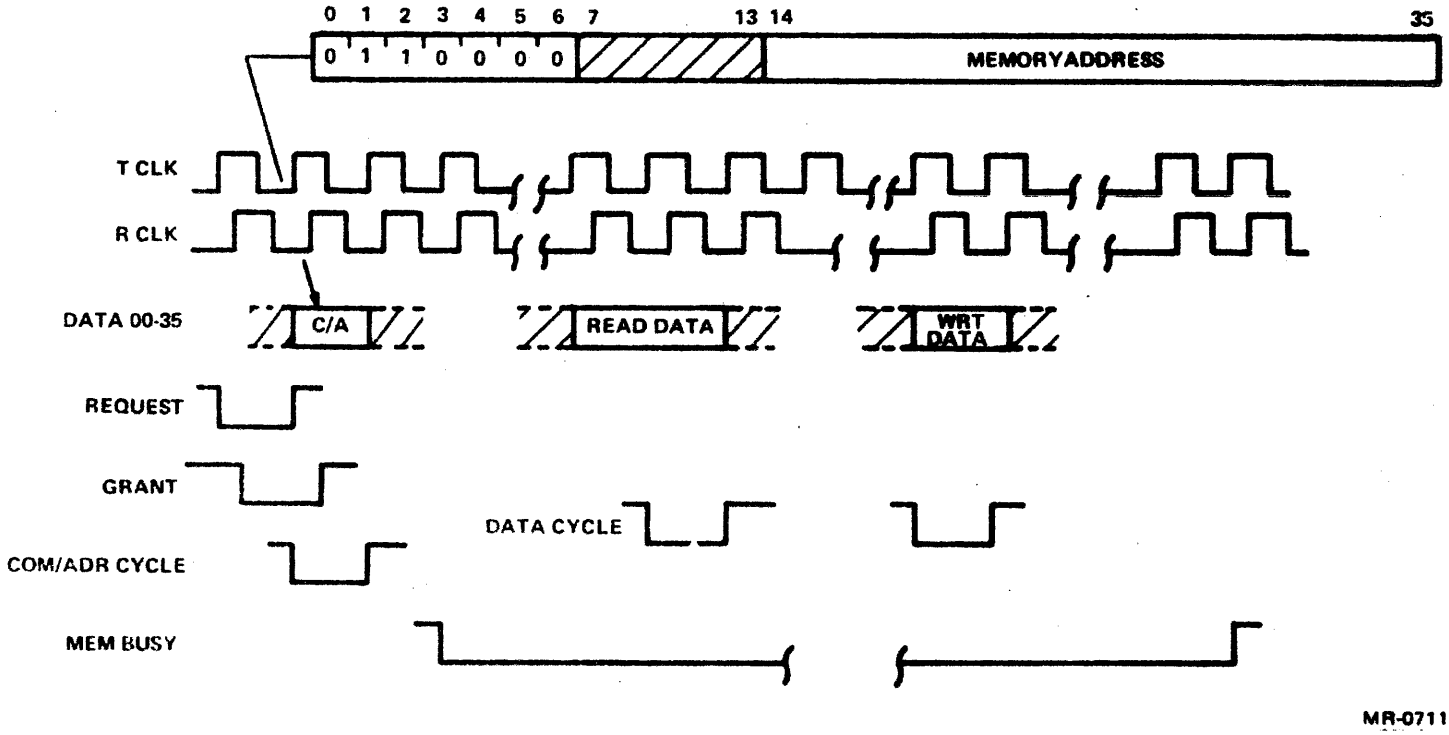


Figure 5-10 RPW, Bus Timing Diagram

After being granted the bus, the CPU or console accesses an I/O register by first transmitting a command/address on the data lines to specify an I/O operation (bit 0 = 1), an I/O address (bits 14-35), and the type of I/O operation; that is, a read (bit 1 = 1) or a write (bit 2 = 1). The command/address may also specify a byte transfer (bit 6 = 1) when a UBA external (Unibus) register has been addressed.

The I/O address consists of a controller number (bits 14-17) and a register address (bits 18-35). The memory and console both have a controller number = 0. UBA1 and UBA3 have controller numbers 1 and 3 respectively. Except for the memory status register (address = 100000) and console instruction register (address = 200000), all I/O registers are UBA internal or external registers. The internal registers include the 64 UBA paging RAM locations

(addresses = 763000-77), the UBA status register (address = 763100) and the UBA maintenance register (address = 763101). The external registers are the addressable registers in the Unibus devices connected to the UBA.

After the addressed bus controller receives the command/address, it always asserts the I/O BUSY control line whenever the operation is an I/O register write operation. If the operation is an I/O register read operation, only the UBA asserts I/O BUSY. (This is because bus controllers which do not supply read data during the requesting device's alloted bus cycles must assert I/O busy to flag the condition.) Unlike MEM BUSY, the I/O BUSY signal does not freeze the arbitrator. Consequently, devices initiating I/O register reads and writes have the bus for only two cycles after transmitting the command/address.

During an I/O register write, the device initiating the operation can assert write data on the data lines during either of the two following command/address cycles. As for the memory write operation, DATA CYCLE is also asserted when the write data is transmitted on the bus. DATA CYCLE is used by the addressed controller to strobe the write data from the bus and to store the information in the addressed register. Although the bus data cycle completes the bus operation, storing the write data may take additional time. For example, the UBA must initiate a DATO operation or DATOB operation (command bit 6 = 1) over the Unibus in order to transfer the information to an external register address. Bus timing for the I/O register write operation is shown in Figure 5-11.

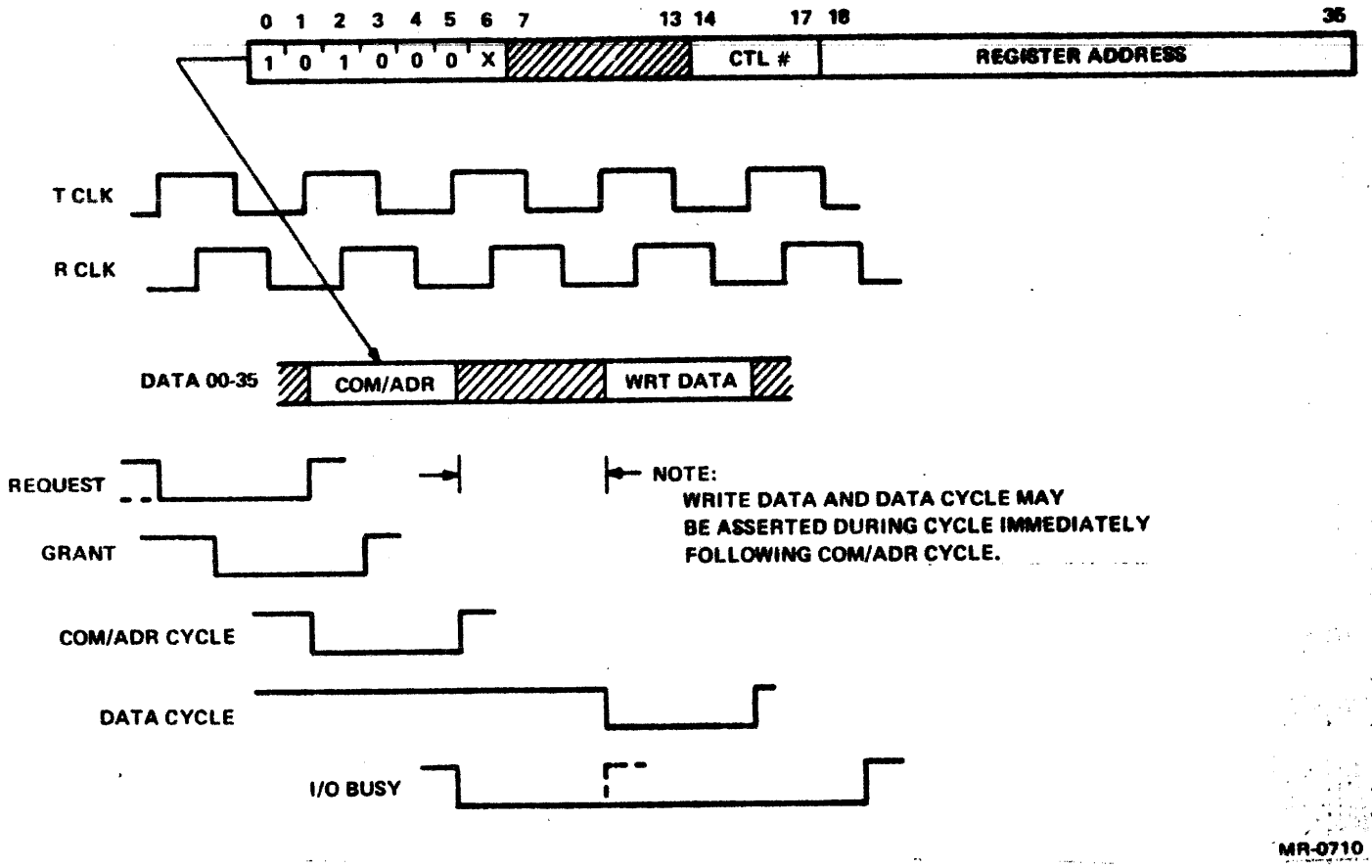


Figure 5-11 Write I/O Register, Bus Timing Diagram

During an I/O register read, bus operation differs depending on which controller register is addressed. If the memory or console I/O registers are addressed, read data is asserted on the data lines by the controller two cycles after the command/address is received. This leaves a free cycle between the command/address and data cycles as shown in the upper part of Figure 5-11. If a UBA register is addressed, read data is not asserted on the bus during the bus cycles allotted to the device initiating the operation. Instead, as shown in the lower part of Figure 5-11, the UBA requests the bus at some later time and transmits the read

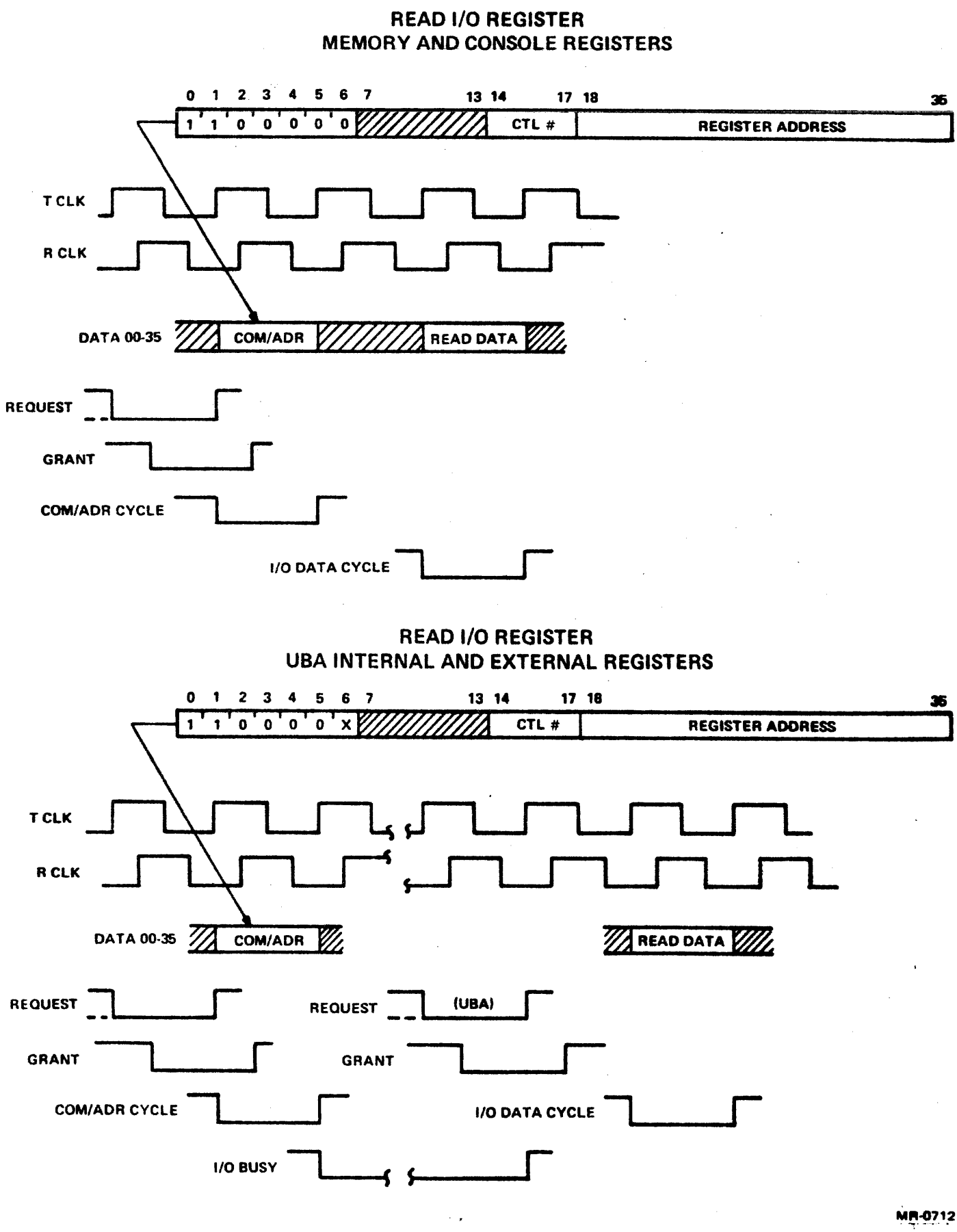


Figure 5-12 Read I/O Register, Bus Timing Diagram

data whenever the bus is granted. The reason for this is that the UBA must initiate a Unibus DATI operation to retrieve data from an external register, and the register data cannot possibly be supplied during the two bus cycles immediately following the command/address cycle. Although UBA internal registers could be read during these two bus cycles, the UBA control logic implements the same operation (bus request to transfer data) to simplify the design. For internal register addresses, the bus request is made during the second cycle following the command/address cycle.

During both types of register read operations, control line I/O DATA CYCLE is asserted on the bus coincident with the read data. Similar to the DATA CYCLE signal asserted during memory read operations, I/O DATA CYCLE serves as a data strobe so that the device initiating the operation may gate the data from the bus.

5.2.8 Bus PI Operation - Part of the control information stored in the UBA status register are 3-bit high level and low level priority interrupt channel numbers (PIAs). The high level PIA is associated with BR7 and BR6 on the Unibus; the low level PIA is associated with BR5 and BR4.

When conditions are met for initiating an interrupt, a Unibus device asserts its assigned BR level. The BR level, in turn, causes the UBA to assert one of seven PI REQ lines (1-7) on the KS10 bus. The PI REQ line that is asserted depends on the value of the stored PIA (1-7) corresponding to the BR level. For example, if BR7 is asserted on the Unibus and the channel number

stored in the high level PIA is 2, PI REQ 2 is asserted on the KS10 bus. As can be seen, with two levels of PIA, the UBA can assert more than one PI REQ at any one time. That is, in the preceding example, if BR5 was also asserted on the Unibus and the channel number stored in the low level PIA was equal to 4, the UBA would assert PI REQ 4 in addition to PI REQ 2. For the case when there are both high and low level interrupts and both PIAs are equal to the same PI channel number value, a single PI REQ would be asserted but as a result of two asserted BR levels.

When a high or low level PIA is set equal to 0, no PI REQ level is asserted on the KS10 bus even though the corresponding BR level is true. This provides a means for programmers to inhibit interrupt activity for a device.

The CPU monitors all PI REQ levels on the KS10 bus. More than one request line may be asserted at any one time (i.e., up to four with two UBAs in the system) and more than one UBA can assert the same request line. The CPU detects all interrupts and resolves interrupt request priority on a channel number basis (lowest channel has highest priority). When it is ready to serve the highest priority channel, it performs the first of two PI operations over the KS10 bus.

The first PI operation initiated by the CPU is to determine the UBA or UBAs interrupting on the PI channel that is to be served. Bus timing is shown in the upper part of Figure 5-13. After requesting and being granted the bus, the CPU asserts the command/address to specify that the operation is an I/O controller

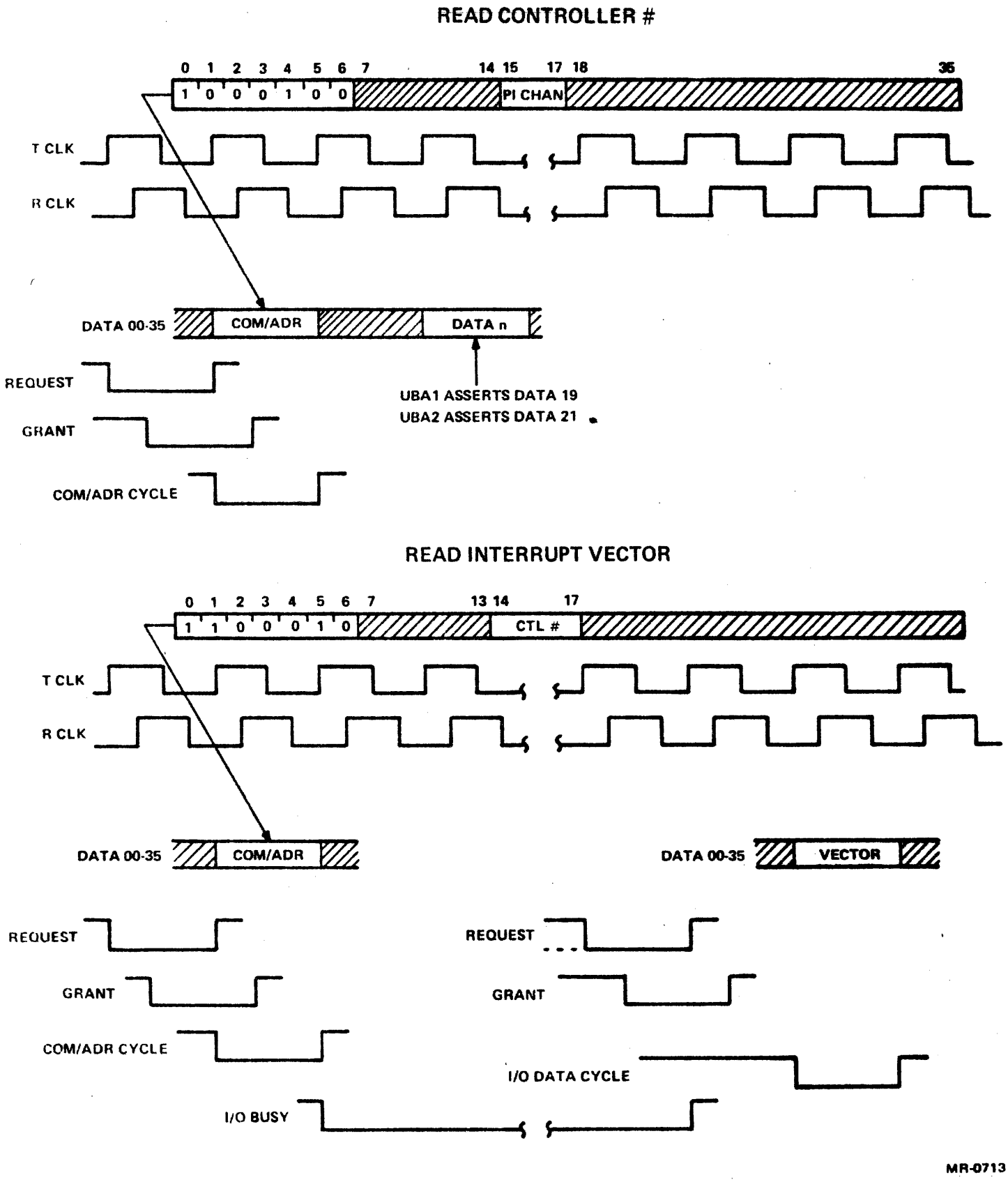


Figure 5-13 PI Operation, Bus Timing Diagram

number read (bit 0 = 1, bit 4 = 1) for controllers interrupting on PI channel n (bits 15-17). When a UBA receives the command/address, and if it is interrupting, it compares the channel number value received on the data lines with the stored PIA. If a match occurs, a UBA asserts one of the data lines to indicate its physical address; that is, UBA1 asserts data line 19 and UBA3 asserts data line 21. The CPU strobes the data lines, (during the second bus cycle following the command/address cycle), resolves controller number priority (UBA1 has highest priority), and then performs a second bus operation to read the interrupt vector from the highest priority UBA.

Bus timing for the second PI operation is shown in the lower part of Figure 5-13. The command/address specifies that an interrupt vector is to be read (bit 0 = 1, bit 1 = 1, bit 5 = 1) from controller n (bits 14-17). When the addressed UBA receives the command/address, it initiates a priority transfer control and interrupt sequence over the Unibus to read the vector from the interrupting device. The vector is read from the device interrupting on the highest level BR associated with the specified PI channel. For example, if both BR7 and BR6 are asserted and the high PIA is being served, the vector is read from the device interrupting on BR7. Because the vector cannot be read during the two bus cycles allotted the CPU after the command/address cycle, bus operation is similar to the I/O register read operation. The UBA requests the bus at some later time, when the Unibus priority transfer and interrupt operation completes, and then asserts the vector address on the KS10 bus data lines when it has been granted the bus. The UBA also asserts I/O DATA CYCLE, which the CPU uses to strobe the data lines to end PI operation on the KS10 bus.

5.2.8.9 Bus Parity Error - All devices connecting to the KS10 bus generate and check data line parity. Each device computes and transmits two parity bits whenever data is transmitted on the bus. PARITY LEFT is the computed parity (even) for the 18 least significant data lines (00-17). PARITY RIGHT is the computed parity for the 18 most significant data lines (18-35). Also, with one exception, each device checks parity when data is received on the bus. The exception is during the PI operation when the CPU reads the bus to determine the controller or controllers interrupting on a specified channel. Because more than one controller may assert a data line, the CPU ignores data line parity during this operation.

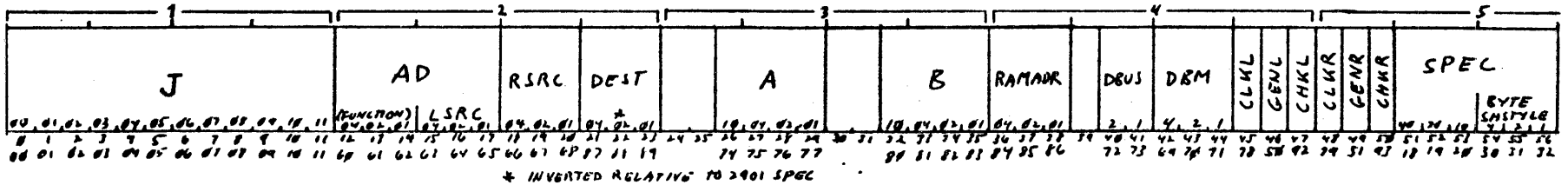
When a device detects bad (odd) parity for data received on the bus, it asserts a PARITY ERROR signal that causes the CPU clock to be stopped. The CPU clock is controlled by the console, and the PARITY ERROR signals from the various bus devices (including the console itself) are ORd together on the console module to set flip-flop CSL3 PE(1) when an error occurs. CSL3 PE, in turn, clears CSL5 ENABLE which negates CSL5 CRA/M CLK ENABLE and CSL5 DPE/M CLK ENABLE to stop the clock in all CPU modules. The parity error is also sensed by the 8080 program, which prints an error message at the CTY.

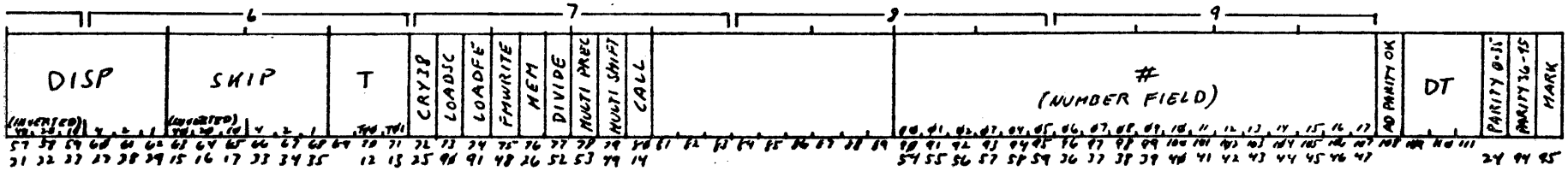
5.3 MICROCONTROLLER

The way the processor performs a program depends both on the processor hardware and on the microcode it executes. Most of the microcode is associated with the execution of the individual program instructions, and these are not treated here. The descriptive material in this and the next two sections is devoted almost entirely to the hardware, plus those microcode procedures of a more general nature, such as sequencing the microcode from one program-level operation to the next and handling priority interrupts and page failures. Associated with the microcode are two quantities, the microinstruction word itself, referred to as the "microword", and a dispatch word that supplies information for the execution of individual program instructions. The first two parts of this section discuss the structure of these words (Figure 5.14), and the rest of the section describes the hardware of the microcontroller (Figure 5.15).

5.3.1 Microword

The upper part of Figure 5.14 shows the format of the control RAM microword. Of the two rows of numbers below the boxes, the upper lists the numbers of the bits as determined by the microcode assembler, and the lower lists their physical numbers according to their positions in the control RAM. Bits lacking physical numbers are either simply not used or

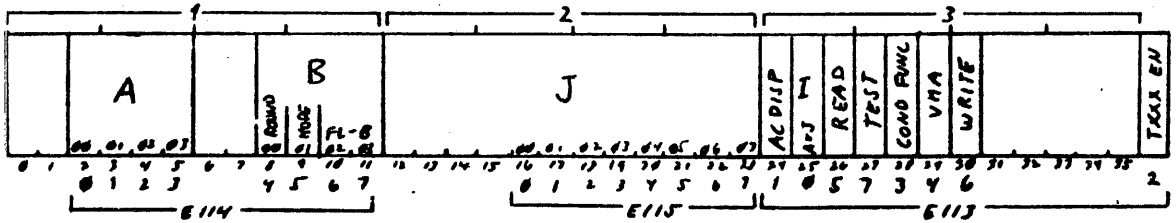




MICROWORD

NUMBER FIELD

ARBITRARY NUMBER
SCAD, SCADA, SCADB, S#
STATE REGISTER
WORKSPACE IN RAMFILE
PC FLAGS
MEMORY CYCLE
BUS FUNCTIONS
PI (2 SETS)
AC + # SELECTION
APRID DATA
HALT CODES
FLG BITS



DISPATCH WORD

Figure 5.14 MICRO FORMATS

are in special macro default fields, which are given in macro definitions but which do not appear in the assembly listing, as they are used only to default other fields that are in the actual microword. In the field definitions given in the microcode listing, the letter D means default to a constant, whereas F means default to a function, such as a macro default field. Bits that have only a physical number are created by the software that sets up the physical microwords. These include a mark bit for scope syncing and two even parity bits, one for that part of the RAM contained on the CRA board (bits Ø-35) and another for the rest of the RAM contained on the CRM board. The numbers above the boxes show the correlation between the parts of the microword as illustrated and the 4-digit groups that make up the words in the microcode listing. Preceding each word in the listing is the address of its location in the control RAM.

The labels used for the fields of the microword are those defined for the microcode assembly language. Multiple labels indicate bits used for more than one purpose depending on the circumstances - the number field is used for many purposes as listed at the lower right in the drawing. For bits labeled "inverted" a Ø rather than a 1 selects the defined function. The hardware signal names are very similar to the microcode labels and the reader should have no trouble identifying them; signal names for the bits on the CRA and CRM boards respectively are listed on CRA6

and CRM2, matched to the physical bit numbers. For those fields comprising more than a single bit, the signal names for the individual bits are numbered, which numbers are written on the bits inside the boxes in the format drawing.

The rest of this section explains the various groups of microword bits to serve as an introduction to the microcode listing. No attempt is made here to identify all the different quantities that can be selected by each field, as that information is given in complete detail at the beginning of the listing.

Ø-11 The address of the RAM location from which the next microword will be taken, perhaps modified by a skip or dispatch, or even supplanted altogether by a subroutine return, some other dispatch, or a page fail condition.

12-35 These fields govern the full word arithmetic unit. From the point of view of the microcode, there are 64 adder functions selected by the six AD bits, where the left three specify the function as defined by the 29Ø1 spec, and the right three specify the source operand. The 9-bit instruction specification is completed by the three destination bits. Note that the middle bit in the destination code is inverted before being applied to the 29Ø1s; hence the microcode

configurations 0-7 correspond respectively to the destination control codes 2,3,0,1,6,7,4,5 as given in the 2901 spec. Note also that the terms "right" and "left" as used in the spec are opposite from their KS10 meanings, and inputs and outputs are numbered in the opposite order.

Physically, the adder is controlled as two separate left and right halves insofar as the operand source is concerned, and the right three AD bits select only the left source. The RSRC field enables the programmer to select a different right source in order to perform operations in which one half of an operand is manipulated as desired while the other half is say simply cleared or left unchanged. If no right source selection is made however, the RSRC field defaults to the value given for LSRC.

The A and B fields supply the A and B addresses for the 2901 register file. Of course a specified address has no effect unless the selected 2901 instruction calls for its use. Only B can select a register for loading.

36-38 The source of the RAM file address. Note that in this field a VMA selection means a memory AC reference from the right four VMA bits, whereas a RAM selection means an absolute

reference to any RAM file location via the right ten VMA bits.

40-41 The source of data for the D bus via the D bus mixer.

42-44 If the D bus field selects DBM as the source, this field selects the source of data into the mixer that feeds the DBM input to the D bus mixer.

45-50 These are two sets of three bits that separately control certain operations in the left and right halves of the main data path. Bits 45 and 48 separately clock the two halves of the arithmetic unit, so that operations can be performed in one half while the other is unaffected. Note that this means there is no change at all in the other half: to load an arbitrary destination with a word half modified and half unchanged requires clocking both halves with separate left and right source selections.

For parity purposes there are an even parity bit and a valid bit associated with each half word in the register file. Whenever a location in the file is loaded, the two associated parity bits are set up from the parity signals generated for the two half words on the D bus, and the valid bits are

set up from bits 46 and 49 of the microword. Hence by means of the valid bits, the microprogrammer can label the parity bits according to whether they actually do represent true even parity for the stored half words. The parity signals generated for the D bus are correct for a word stored if the operation performed by the arithmetic unit is parity-conserving, as is the case for a simple transfer or anding with the mask, and the source of the data word is the RAM file or DBM. Of course parity storage is also valid if the operation is simply the transfer of the contents of register A to register B and the D bus mixer selects DP. For convenience in handling these control bits a macro definition can put a 1 in bit 108 to indicate that the macro operation conserves parity: in a microword containing the macro, the GENL and GENR fields default to the value given by bit 108 unless the programmer overrides it.

Bits 47 and 50 enable parity checking on the left and right halves of the D bus. A parity check should always be made when the source of the D bus data is the RAM file or the backpanel bus (MB). When the D bus mixer

selects DP, the parity check should be made only if the AU operation conserves the parity given by the bits associated with the file location selected by the B field (as that is the source of the parity indication against which the check is made). Even then the requested check for either half is overriden by the corresponding valid bit being off, indicating that the stored bit does not represent true parity for that register. No check should ever be made when the source is VMA or any DBM selection other than MB.

51-56 This field selects among a number of special functions such as loading IR, manipulating flags, and sweeping the cache. Additionally, if the AD function being performed involves shifting, the right three bits control the connections at the adder extremities for the type of shifting; and if the DBM field selects the data path input to the DBM mixer, the same three bits select the position (if any) for insertion of a 7-bit byte in the word. The right three bits are decoded together, and the left invidually select groups of eight functions. Hence functions can be combined. The same

right three configurations select loading IR and XR, so they can be loaded together (IR includes AC, and XR includes the indirect bit). Similarly the right code that selects arithmetic shifting also selects (via the 40 bit) the ASH overflow test, which is used for left shifting.

57-62 This field selects a quantity to be ored with J field bits 0-7 or 8-11 or both to select the location of the next microword to be executed. To jump to a specific location such as that given by the J field of the dispatch ROM or a return address from the subroutine stack, the microword J field must be zero.

63-68 This field selects a skip condition, which if satisfied, causes a 1 to be ored into bit 11 of the address for selecting the next control RAM location. Thus the microcode can jump to an even location with the possibility of skipping that word and going directly to the next odd location. The skip field can select one, two or three conditions from among three sets of six, where the skip occurs if any selected condition is satisfied.

70-71 The processor cycle is extended beyond two clock ticks by the number of ticks specified by this

field. For convenience the T field defaults to the value given by bits 109-111, which can be specified in a macro definition. Thus a macro can indicate when extra time is needed for the operation it produces, but the programmer can override this specification if the extra time is not needed because of the circumstances in which the macro is used.

72 Inserts a carry into the LSB of the adder.

73 Loads the step counter from SCAD as set up by the number field.

74 Loads FE from SCAD as set up by the number field.

75 Writes the contents of the D bus into the RAM file.

76 Starts or completes a memory cycle whose characteristics are specified by the number field.

77 This microinstruction is doing a divide.

78 This microinstruction is doing a multiprecision divide step.

79 Causes this microinstruction to be executed as a no-op if FE bit 0 is already 0, but otherwise causes it to be repeated until FE overflows (bit 0 becomes 0). This feature is used for multiple shifting.

80 Pushes the current location on the stack to effect a subroutine call.

90-107 This field supplies information for a variety of

functions selected by other fields. The kinds of information are listed in the format drawing.

5.3.2 Dispatch Word

The 9-bit instruction code in IR automatically selects one of the 512 locations in the dispatch ROM and makes its contents available to the skip and dispatch logic. Dispatching on the given information occurs mostly in the part of the microcode labeled "The Instruction Loop," which appears at the beginning of the listing just after the power-up sequence. The format of the words supplied by the dispatch ROM is shown in the lower part of Figure 5.14 using the same conventions as for the microword given above. However the dispatch ROM bits are not numbered physically, so chip locations and outputs are given instead.

2-5 This field specifies the kind of operand fetching to be done for the instruction, and for a simple read indicates whether the next instruction can then be fetched immediately.

4-7 This field specifies the test condition in all test instructions, the modification of the masked bits for logical testing, and in all other instructions it specifies the disposition of the results except that in floating point it also

indicates whether there is rounding and whether the operation is additive or multiplicative. The extra physical bit, TXXX EN, shown at the right end of the word, is a duplicate of bit 9 of the B field.

12-23 The address of the control RAM location at which execution of the instruction begins. This 8-bit field selects a location in the range 1400-1777.

24 Causes the AC field of the instruction word to replace the right four bits of the J field so that a jump to begin instruction execution will actually dispatch to one of 16 locations. This is used in JRST and the I/O instructions where the instruction code is expanded to 13 bits.

25 Causes an immediate dispatch on the J field when the microword calls for the standard AREAD dispatch on the A field. This is used for instructions that require no memory access or special setup (e.g. MOVEI, JFCL).

26 Starts a memory read when the microword does an AREAD dispatch.

27 Starts a write test (for page fail) when the microword does an AREAD dispatch.

28 Starts a memory write when the microword does a BWRITE dispatch on the B field.

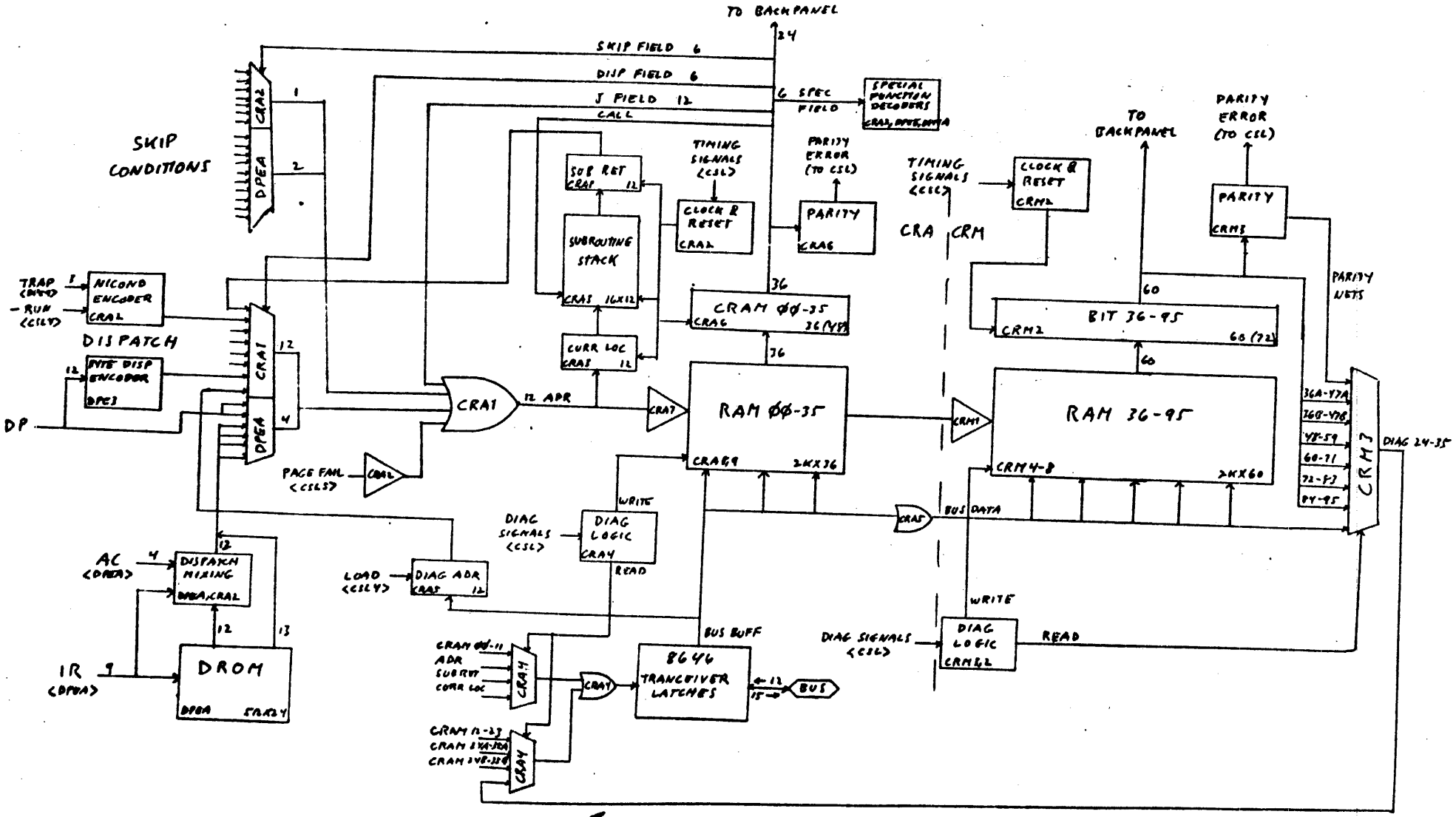
29 Loads VMA when the microword does an AREAD

dispatch.

30 Starts a memory write when the microword does an AREAD dispatch.

5.3.3 Control RAM

The 2048-word control RAM (Figure 5.6, upper right) is made up of a pair of 1K RAM chips for each microword bit. Bits 0-35 are on the CRA board and are shown on prints CRA8,9; bits 36-95 are on the CRM board and are shown on CRM4-8. An entire microword is selected by selecting a single bit from each pair of chips. Selection is made by an address supplied by the skip and dispatch logic (Section 5.3.4) and applied to the two parts of the RAM through the drivers on CRA7 and CRM1. Associated with the two parts of the RAM are two parts of a register that holds each microword while its bits are controlling the events that constitute its execution and are supplying an address for use in selecting the location of the next microword. These two parts are the CRAM register on CRA6 and the BIT register on CRM2.(In each there are duplicate flip-flops for the 12-bit segment that sustains the heaviest use.) At the end of each processor cycle the clock triggers the events for one microinstruction and loads the next into the register from the RAM. However if a 1 in the multishift bit disables the microcontroller clocks (on CSL) without affecting the data path clocks, the same microinstruction is repeated. The parity nets for



5.Ø MICROCONTROLLER

5-7Ø

checking the CRA part of a word are at the upper left on CRA6, and those for the CRM part are across the top and in the lower right corner of CRM3. The outputs of the nets go directly to CSL to stop the processor clock should an error occur.

The J field is used solely by the microcontroller address logic; all other CRAM and BIT signals are available via the backpanel to other boards, although most of the skip, dispatch and special function bits are used on CRA. Most of the bits that control the 29Ø1s are applied directly to those chips, although a few are also used elsewhere in the arithmetic logic. Most other multibit fields are applied to mixers to select among various sets of inputs, such as the data for DBM or the address for the RAM file. The right three special bits are applied to mixers for selecting shift inputs at the 29Ø1s, but are otherwise applied to decoders for generating specific functions, where the individual decoders are enabled by 1s in the 4Ø, 2Ø and 1Ø bits, or for byte insertion, by the appropriate configuration of the DBM field. In some cases, duplicate decoders are employed in order to get a function signal as close to the target logic as possible, and in a couple of cases individual function signals are duplicated for use in two different places. Decoders enabled by the 4Ø and 2Ø bits are at the upper left in DPE5; at the upper right in DPMA are a decoder for the 1Ø bit and a duplicate of that for the 2Ø bit; and a

duplicate for the 1Ø bit is at the right on CRA2.

5.3.4 Skip and Dispatch Logic

The logic that determines the location from which the next microword will be taken is shown in the left quarter of Figure 5.15 and appears mostly on prints DPEA and CRA1,2. Each address is supplied to the control RAM through the OR gates above the two rows of mixers on CRA1. From the 6-bit microword dispatch field, individual inputs to the mixers are selected by the right three bits and the different sets are enabled by single bits among the left three. The upper row on CRA1 has 4-bit mixers for the left eight address bits, and these are enabled by the 2Ø dispatch bit. The lower row, enabled by the 1Ø bit, contains 8-bit mixers for the right four address bits. A similar set of mixers for the right four bits but enabled by the 4Ø bit appears at the upper right on DPEA; the outputs of this set are applied directly to the lower row of OR gates on CRA1 as the DPEA DISP signals.

The OR gates on CRA1 combine the outputs of the several sets of mixers with the J field from the CRAM register. Hence there are two ways to address a single, arbitrary location in the control RAM: with the dispatch mixers disabled, the microcode can jump to the address given by the J field;

with J zero, dispatch mixers for all 12 bits can supply a specific number, such as a diagnostic or subroutine return address. But the microcontroller can dispatch within a range of four, eight or 16 locations, starting at that given by the J field, by oring a variable quantity into the right four address bits through the mixers enabled by the 10 or 40 bit. Note that for an individual mixer to have any affect the corresponding bit in the J field must be 0; a 1 in the J bit overrides any selection made by the mixer.

At the lower right on CRA1 the OR gates for address bit 11 also receive the outputs of the three skip mixers. This arrangement allows the microcode to give an even J with the possibility of going instead to the next odd location on the satisfaction of any of three independently specifiable skip conditions. The skip mixers function from the skip field of the microword in exactly the same way as the dispatch mixers. Those for the 40 and 20 bits (which handle mostly flag and arithmetic conditions) are in the upper left corner on DPEA, and the mixer for the 10 bit is at the upper right on CRA2. Note that the signals that can be selected for skipping are all inherently synchronized to processor operations except for conditions 4-7 in the CRA2 mixer. These four conditions are therefore synchronized to the cycle by means of the flip-flops in E115 (D3). One of these signals, I/O LATCH, is the OR function, by way of a flip-flop in E416 (A6), of the two bus signals that

represent response to an I/O instruction. The synchronization is handled via the bottom gate in the clock logic at the left and the top flip-flop in E416. The skip condition flip-flops are set up at the end of every processor cycle through assertion by the clock enable of the signal DISP & SKIP EN. The same T clock also sets the top flip-flop in E416 to generate FIRST CYCLE, which really means the first tick in the processor cycle. If microword bit TØ1 is 1, indicating a three-tick cycle, the asynchronous skip conditions are updated at E115 so they will be fresher when used at the end of the cycle.

Finally note that the page fail signal from the console is fed into all of the address OR gates. Hence it can override any selection made by the J field or the skip and dispatch mixers, and force selection of location 7777.

5.3.4.1 Dispatch ROM. The left half of each instruction is loaded from the D bus into the IR, AC, indirect and XR registers at the left on DPEA. Each instruction code from IR selects a location in the dispatch ROM, made up of the three 512X8-bit chips at right center. The outputs from the left chip are used as individual control signals or skip conditions, as in the net at C5 where TXXX EN is combined with AD=Ø to decide on a skip in the microcode to execute a skip or jump in an instruction. The microcode can dispatch on the A and B fields from the center chip by way of the

bottom two inputs to the dispatch mixers at the top of the print; the latter occurs in the "Store Answers" part of the instruction loop and elsewhere for specific instruction groups. Dispatching for instruction execution is on the J field from the right chip but this is somewhat roundabout. J bits 0-3 are input to address bits 4-7 through the upper mixer on CRA1, and the same dispatch function puts 1s in bits 2 and 3 so dispatching is in the range 1400-1777. In the normal situation J bits 4-7 go to address bits 8-11 through the dispatch mixer at the top of DPEA as the DPEA J signals available from mixer E118 (A3). But on an AC dispatch for JRST or an I/O instruction, the AC address is substituted for the DROM J bits.

The standard AREAD dispatch on the A field for the "Fetch Arguments" part of the instruction loop uses control RAM locations 40-57, but a 1 in the I bit of the dispatch word can cause an immediate dispatch on the J field. This is accomplished through two mixers, E119 at DPEA A2 and E420 at CRA2 C3. For the standard dispatch, AREAD bits 8-11 from E119 are equivalent to the DROM A field and are supplied to the address through input 3 of the mixers at the top of DPEA. E420 sets AREAD 04-07 to 0010, which selects the desired range through the upper mixers on CRA1. For an immediate dispatch, the I bit, which is the A=J signal, substitutes J 08-11 in AREAD 08-11, substitutes J 00-03 in AREAD 04-07, and inserts 1s directly into address bits 2 and

3 via mixer E517 (CRA1 C6) to make the range the same as that used for an ordinary J dispatch.

5.3.4.2 Other Dispatch Procedures. The next instruction condition or NICOND dispatch appears at the very beginning of the "Start Next Instruction" part of the microcode instruction loop. The dispatch is handled through the lower mixers (input 4) on CRA1 and the signals are generated, except for the most significant, through the priority encoder at E216 (CRA2 B3). Only five encoder inputs are used, and at the end of any program-level microcode operation they provide for dispatching to the next operation in the priority order trap 3, trap 2, trap 1, halt, and the ground at input 7 provides for going on to the next program instruction if none of the other conditions intervenes. The dispatch in the microcode actually has two sets of five locations distinguished by NICOND Ø8, which simply indicates whether a memory cycle is in progress. This condition has no effect on traps or a halt, as the microinstructions in each pair of dispatch locations distinguished only by the memory condition are identical. But it does affect the next normal instruction and indicates whether the instruction must still be fetched or is already being prefetched.

The D6 input of the mixers associated with the 1Ø bit provides for dispatching to every other location among 16 for the effective address calculation, which immediately

follows the NICOND dispatch in the microcode listing. Here again there are two categories of dispatching on whether or not there is indexing or indirection, one specifically for the instruction JRST 0, and one for all other instructions. The special case is the most frequently used instruction in the entire PDP-10 set, and the AND gate at A4 on DPEA saves a processor cycle by detecting JRST 0, directly from IR, obviating the J dispatch.

The most common byte size used is seven bits. The KS10 saves considerable time by having hardware for manipulation of 7-bit bytes with zero alignment built right in. Most of this hardware is associated with the DBM mixer and the 10-bit logic (Section 5.4) but the microcontroller has a mechanism for dispatching on byte position, i.e. on which byte in the word is being processed. The three byte dispatch signals available at the D5 inputs to the lower CRA1 dispatch mixers are provided by the decoder-mixer combination at the lower left on DPE3. When DP carries a byte pointer, the decoder is enabled by a size indication of 7, and the circuit translates the zero-alignment byte positions into byte dispatch configurations as follows.

Byte	Position	Dispatch Code
1	29	001
2	22	010

3	15	100
4	8	101
5	1	111

The single D7 input to the lower CRA1 mixers (at E122) provides for a skip of two locations (actually to the next even location) from the microword J field when SCAD is negative. Similarly the arithmetic condition at E121D2 provides a four skip that is used in multiplication. The remaining inputs to the mixers at the top of DPEA provide for dispatching on various sets of bits in a word on DP, in one case combined with arithmetic conditions.

5.3.5 Subroutine Stack

The binary counter and RAMs in the middle row on CRA3 provide a standard stack for microcode subroutine calling. Position in the stack is determined by the value in the counter, which goes up for pushing and down for popping. The top of the stack is defined as the location whose address is one greater than the number in the counter, and the stack therefore allows a depth of subroutine nesting of 15 levels. The address lines to the RAMs carry the current value in the counter unless SELECT NEXT enables the gates at the right of the counter, in which case they carry an address two greater. Following the initial reset of the

counter from the console, the top of the stack is at location 1.

At the end of every processor cycle, the cycle clock loads the address of the next control RAM location into the current location register at the bottom of the drawing. Halfway through the first tick the stack write signal through the top gate in the clock circuitry on CRA2 loads the current location into the RAM position one above the current top of the stack as selected by the select-next gates. This is done at the beginning of every microinstruction - if it turns out to be unnecessary, the stored address is just thrown away when the next current location is loaded in its place. However, if the microinstruction is a call or there is a page failure (the call or return signal from CRA2 A5 includes the page fail condition from the console), the counter is incremented so the temporary save location now becomes the top of the stack. Simultaneously the saved address is loaded into the register at the top of the drawing so it is available for a subsequent return. On the other hand, if the microinstruction is a return (and thus makes use of the SBR RET address), the select-next gates are disabled, and at the same time the cycle clock decrements the counter, the address from the top of the stack is moved to SBR RET for a subsequent return from the level in which the just-executed subroutine was nested.

5.3.6 Booting and Diagnosis

The logic through which the console directly manipulates the microcontroller is shown across the bottom of Figure 5.15. The reset signals are located on the same prints as the clock circuitry. Note in particular (CRA2 A3) that the reset for the stack is separate from that for the microinstruction register, so the console can clear the register, and then inspect locations or single step, without bothering the stack.

To bootstrap the microcode, control signals from the console bring in data 12 bits at a time from the backpanel bus via the transceivers on CRA5. Loading each location in the control RAM requires nine transfers, the first for an address, which is loaded into the register at the top of the drawing, and eight more for the 96-bit microword in 12-bit segments. With the microinstruction register clear, J is zero, the skip field selects no condition, and the dispatch field selects the diagnostic address through the mixers on CRA1. By means of the gates and decoders at the bottom of CRA4 the console can select and write three segments in the addressed location in the CRA part of the control RAM, and similar logic at the lower left of CRM3 handles the selection and writing of five segments in the CRM part.

The same selection signals, but with the write replaced but

a read (CRM2 B3), can read any 12-bit segment of the CRM part of the microinstruction register, the contents of the transceiver latches, or the output of the CRM parity nets through the mixers on CRM3. The same signal that enables the CRM read mixers, CSL4 DIAG 1Ø H, disables the upper mixers on CRA4 and selects the output of the CRM3 mixers as the input to the lower CRA4 set. When that signal has the opposite polarity however, the signals that select the sections of the CRA part of the control RAM select 12-bit CRA inputs to one or the other set of mixers on CRA4. The quantities selected can be any part of the CRAM, the contents of the current location or subroutine return register, or the address supplied to the control RAM by the skip and dispatch logic. The output of either mixer set is available through the OR gates at the top of the drawing to the TRN inputs to the transceivers on CRA5. Note that the parity signals generated for the transmitted data by the three transceivers that handle the 12 bits are themselves transmitted through a fourth transceiver on an additional three data lines to make the parity on the bus even.

When the console first starts the microcode, it executes the "Power-up Sequence", which is at the beginning of the listing. In the register file this sequence sets up the constants, clears control words, and also clears temporary registers to avoid parity errors. In the workspace it sets up a table of powers of 1Ø for binary-to-decimal conversion,

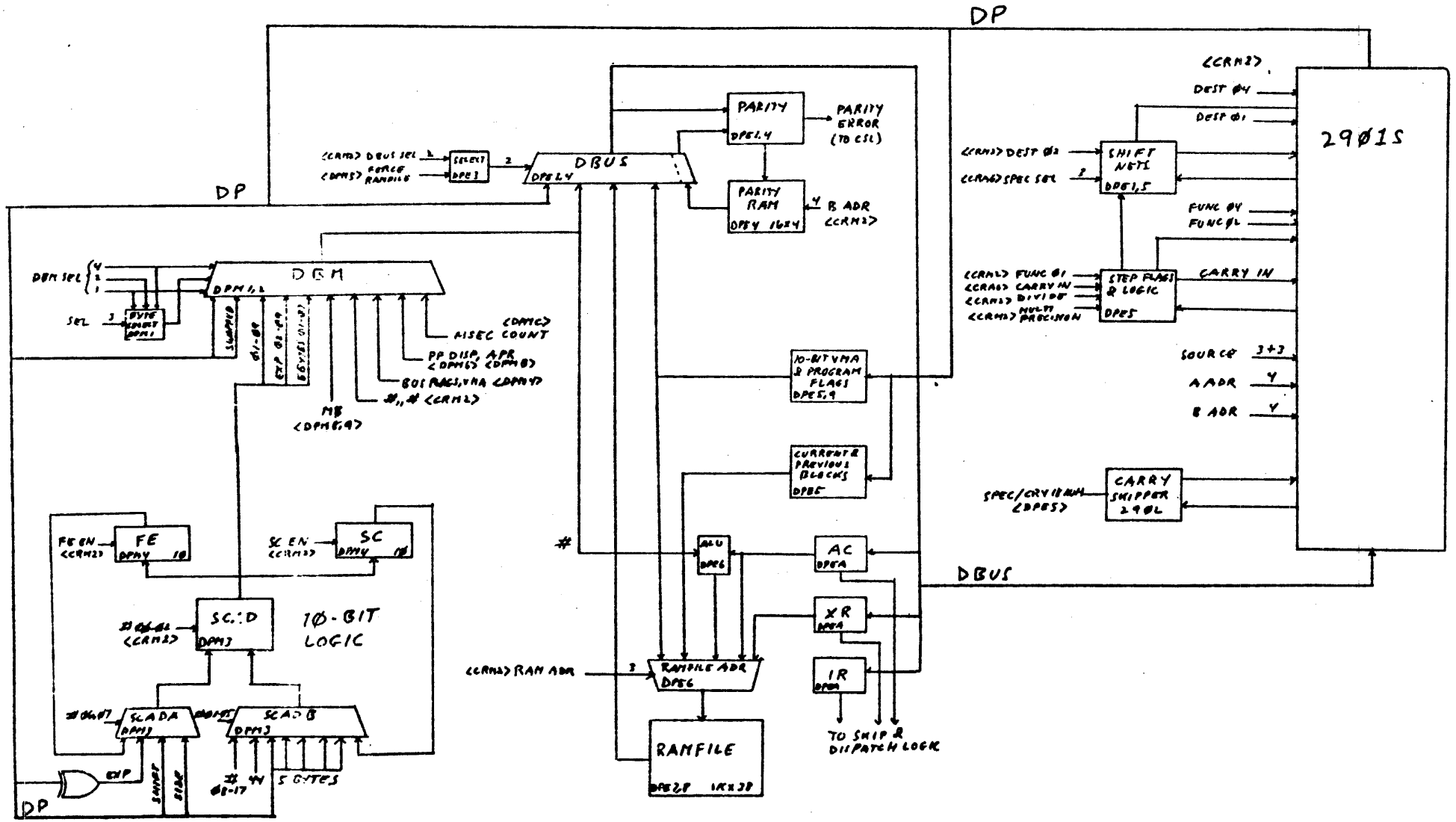
clears locations for the time base and flag enables, and saves the address of the halt status block. Finally it clears the flags, enters executive mode, and enters the halt loop.

5.4 DATA PATH EXECUTE

Although the activities of the two data path boards are very much intertwined, the logic can reasonably be divided into two parts: the execute data path, which handles all the internal operations for the execution of an instruction - arithmetic and logic operations, data manipulation; and the memory data path, which handles all aspects of communication over the backpanel bus for both memory and I/O instructions, including determining whether a memory access should be made instead to the RAM file (a cache or AC reference) and thus be turned over to the execute data path. Although the boards are labeled DPE and DPM, the logical and physical boundaries do not coincide, and both paths include elements on both boards. Here we deal with the execute part of the path; the memory part is discussed in the next section. Figure 5.14 is a block diagram of the execute path, but for the internal structure of the register file refer to Figure 5.18.

5.4.1 Arithmetic Unit

The heart of the main data path is the arithmetic unit, shown on prints DPE1,2. Most of the logic is the 2901s themselves, and they are described in the 2901 spec. Just as 36-bit words are centered in the 40-bit register file, the D bus inputs are centered in the ten slices, with the



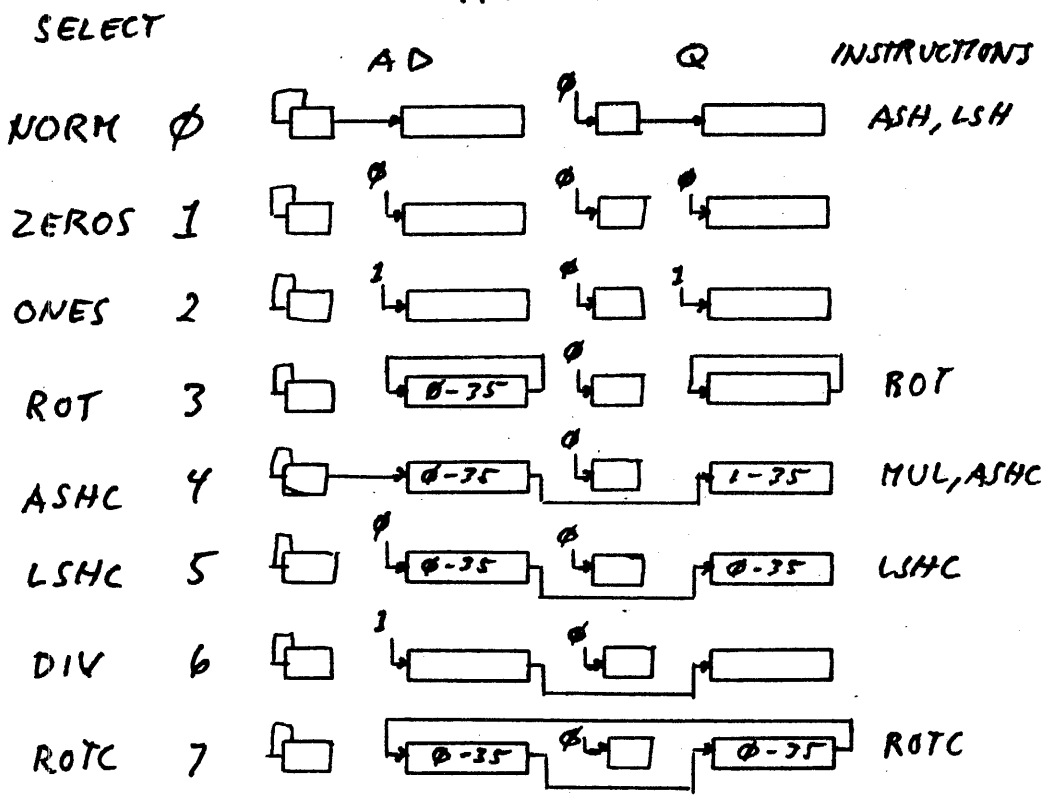
5.16 DATA PATH EXECUTE

extra pair of bits at the left receiving copies of bit Ø (the sign), and the extra pair at the right receiving Øs. The chips are interconnected for left and right shifting, but instead of direct carry connections the carry function is handled through look-ahead logic supplied by the 29Ø2s at the bottom on DPE2. Note that the carry from the right half to the left, i.e. into bit 17, is controlled by the microcode. Also under microcode control are the separate clocks for the two halves via the middle gates in the clock circuitry at the left on DPE5. The bits from the appropriate microword fields, with separate left and right source selections, are applied directly to the chips with two exceptions: the Ø2 destination bit, which must be inverted and distinguishes between left and right shifting in those functions that do shift; and the Ø1 function bit, which distinguishes between add and subtract when the Ø4 and Ø2 bits are both Ø.

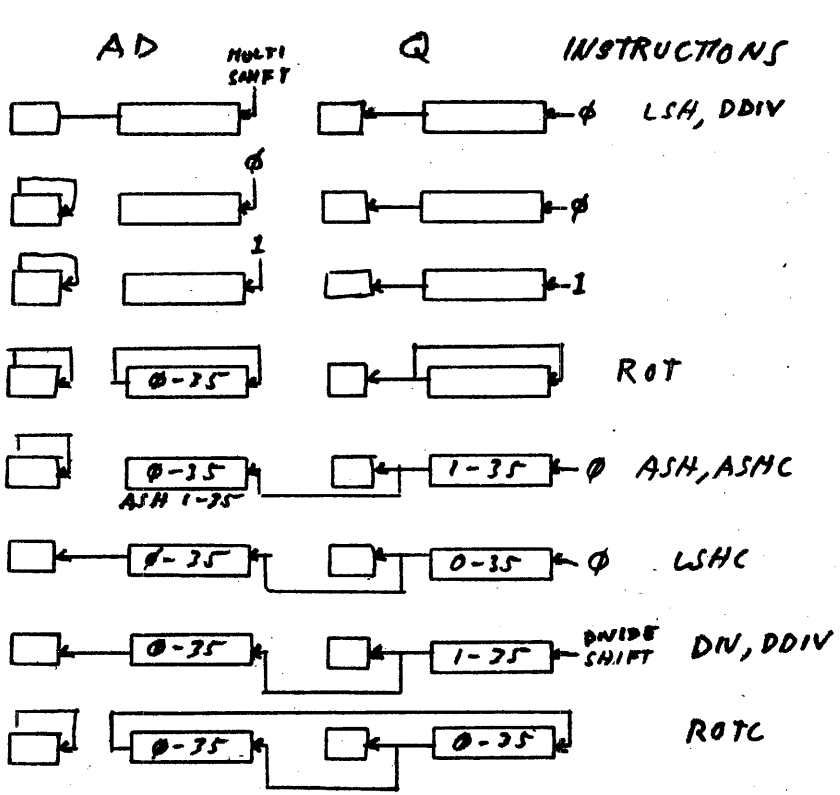
Having words centered in the 4Ø-bit adder is appropriate for some one-word shifts and for additive operations, as the sign is available at the left end or for LSH the extra bits can be masked out; and moreover the result of an arithmetic operation can never exceed 4Ø bits, so DP SIGN always has the correct sign even when DP ØØ is wrong because of overflow. On the other hand, for arithmetic shifting and multilength operations it is not suitable to have words centered, as there is then a hole in the middle of a double

length operand in AD and Q (which can be shifted together), and the connection between the sign and bit 1 is buried in the leftmost chip. Hence before performing such operations, the microcode must move the operands to the right, frequently placing them entirely in the right nine chips, which are then used as a 36-bit AU. That such action is expected is evidenced by the signals that serve as inputs to the shift logic at the bottom on DPE1 and by the fact that the carry out of bit 2 is an input to several logic nets and is available for testing by the microcode. The net in the lower right corner on DPE5 performs the necessary inversion of the Ø2 destination bit and also supplies the left and right shift signals to the shift logic on DPE1. When shifting is called for, the gates at the far left supply shift connections that are constant for a given direction, and the tristate mixers decode the right half of the special function field to set up those connections that vary depending on the type of shift. The tristate logic is necessary because the 29Ø1 pins that receive inputs for shifting in one direction supply outputs for the other, at which time the corresponding tristate circuits are disabled so their outputs neither drive nor load the signal lines significantly. Figure 5.17 shows the various kinds of shift arrangements for shift instructions and arithmetic subroutines, where the short boxes represent the left slice and the long boxes the other nine. The indicated use is for the main shift activity, and the numbers inside the boxes

RIGHT SHIFT



LEFT SHIFT



IF ~MULTI PRECISION, MULTI SHIFT IS 0
OTHERWISE MULTI SHIFT = FLAG PL 02

IF ~DIVIDE, DIVIDE SHIFT IS 0
OTHERWISE DIVIDE SHIFT = FLAG CARRY OUT
& ASSERTS CARRY IN & FUNC 01

indicate the initial position of the operands for the type of shift, if different from the normal. Before the main shifting activity, the microcode must of course move the operands from their normal positions to the ones given, making use of whatever shift arrangement is appropriate.

Note that two of the bit inputs to the shift logic do not come directly from the 29Ø1s. These two signals plus the carry into the right end of the adder and the above mentioned Ø1 function bit are supplied by the gates at the right on DPE5. Through the top two gates, 1s in the corresponding microword bits do assert the function and carry signals, but the rest of the logic is for assisting the microcode in division and certain multiprecision operations. The flip-flops save information from one step for use in the next or from operations on lower order words for use on higher order. In division for example, the carry out in one step means that in the next a 1 must be shifted into the partial quotient and the divisor must be subtracted from the dividend; hence FLAG CARRY OUT being set causes a divide step to assert DIVIDE SHIFT for input to the shift nets and implement a subtraction by generating a carry in and asserting the Ø1 function bit. This simple hardware feature saves a great deal of microcode time: instead of requiring the microcode to use a skip to decide whether to add or subtract in each divide step, it simply calls for an add in every step, and when the carry is present the add

changes automatically to a subtract accompanied by the carry in required for twos complement arithmetic. In a similar way the microword multiprecision bit carries over a subtraction from one step to the next, inserts a carry in a high order operation if there was a carry out of the low order (using the right nine slices), and bits shifted left out of the second position can be inserted at the right in the next normal shift step via MULTI SHIFT. This last signal, which has absolutely nothing whatever to do with the microword multishift bit, supplies Øs in LSH. FLAG QR 37 provides multiplier bits for dispatching in the multiply subroutine.

5.4.2 Main Path

The output of the arithmetic unit is available via DP to many processor elements, including the mixers for the D bus on prints DPE3,4. These mixers can also select the output of the RAM file, the output of the DBM mixer on the DPM board, or a word made up of the program flags, the number of the PI level on which a new request has been accepted, and the right 1Ø VMA bits that are kept on the DPE board for accessing the RAM file. Input selection is made according to the microword DBUS field through the gates at bottom center on DPE3. But note that a microword selection of DBM can be forced to a RAM file selection instead; this occurs when DBM is selected for MB and the memory request turns out

to be a cache hit or an AC reference. The selected word can be sent over the D bus to the arithmetic unit, the RAM file or the instruction register, which is at the lower left on DPEA. All of the instruction bits can be loaded together by two special functions, of which one handles both the IR and AC fields, and the other handles the XR and indirect fields as well as a bit that indicates the instruction is being executed by a PXCT and should do its indexing in the previous context. XR and AC are decoded for zero for use by the skip and dispatch logic.

The remaining D bus logic is for parity operations, and includes the standard nets for generating even parity bits and checking parity. It also includes, on DPE4, the E714 flip-flops and 16X4 RAM that implement the parity arrangement described in the discussion of microword bits 45-50 in Section 5.3.1. The RAM contains two validity bits and two parity bits for each location in the register file and is written according to the B field selection whenever the destination code loads a register. (Writing occurs at the leading edge of the cycle clock, 75 ns before the 2901s are clocked, through the bottom gate in the clock circuitry at the left on DPE5.) The E714 flip-flops allow generation of a left or right parity error signal for the console when the corresponding check indicates bad parity, but only if the microinstruction enables the parity check and the hardware provides the appropriate DBUS CHK EN signal. These

enable signals are supplied through an extra mixer for each half of the bus. The signals for both halves are always false on VMA selection and always true on RAM file or DBM selection (in the last case the microinstruction should enable parity checking only if MB is the source, because the parity bits that accompany the DBM selection are those supplied by the backpanel bus). When DP is the source, the parity bits are those supplied by the RAM location selected by the B field, and the D bus check enable signals stem from the corresponding valid bits.

The final part of the main path is the DBM mixer, which appears on DPM1,2. Inputs, as selected by the microword DBM field, can be any of those listed in the table at the lower right on DPM1. Selection of "bytes" provides Ø in bit 35 and five copies of SCAD Ø1-Ø7 in the other 35 bits. Reading the exponent puts a Ø in bit Ø, the exponent from SCAD Ø2-Ø9 in bits 2-9, and fills the rest of the left half from DP but reads the current value of the MSEC counter in the right half. The number field is duplicated on the two halves of DBM. As is to be expected the bits of the microword DBM field are applied to buffers for multiple drive lines for the mixers. But whereas the drive lines for the 4 and 1 bits typically each drive a third of the mixers, the 2 bit has five lines each corresponding to a 7-bit byte. These lines are further gated by the configurations of the right half of the special function field through an E412 decoder

that is enabled by DBM select code 1 or 3. When the code is 1 all of the drive lines for the select 2 bit are off as required. For code 3, the select 2 drive lines that are on cause selection of the DP input, but a function number from 1 to 5 turns off the corresponding select 2 line, causing one set of seven mixers to select the input for code 1 instead of code 3. This inserts a 7-bit byte from SCAD Ø1-Ø7 in the selected position with the rest of the word made up from DP. Byte 5 is handled as eight bits but the final mixer receives DP 35 for either code.

5.4.3 RAM File

The 38 RAMs on DPE7,8 provide storage for 1Ø24 words with an even parity bit for each half. The word contained in the location selected by the address inputs is available at the RAM outputs, and a falling edge at the write input replaces it with the contents of the D bus. The write signal, which occurs at the falling edge of the cycle clock, is produced through the gates at upper center on DPE5 when a microinstruction requests writing or a memory write occurs on a cache hit or an AC reference. Other DPE5 logic for the RAM file is the E3Ø8 flip-flops at lower center that hold the numbers of the current and previous fast memory blocks as specified by the program, and the upper right flip-flops that hold the DPE copy of the right 1Ø VMA bits. The loading of both VMA and its partial copy is produced through

the gate at A4 when the microcode gives a memory function that requests it or initiates a cache sweep.

The ALU at the left on DPE6 can generate numerous functions but is used principally to add the least significant four bits of the number field to the instruction AC field to generate addresses for the block of accumulators used in extend instructions. The rest of the logic is mixers for selecting the RAM file address according to the source specified by the microword RAMADR field as given by the table at the lower left. The generation of the address is logically in three parts corresponding to the three rows of mixers. The bottom row selects the obvious source for the least significant four bits directly according to the microword field. The middle row selects those three bits that for fast memory references correspond to the block designation. This requires an extra mixer at the left through which address bits Ø4 and Ø2 select other functions to make the address selection. The obvious selection is made for a cache hit, a VMA or number reference, or the current block for an accumulator; however an index reference may be to either the current or previous block, and substitution of an AC reference for memory may also be to either block. An address selection code less than four always means fast memory, so a Ø in the Ø4 microword bit disables the top row of mixers altogether. Codes 6 and 7 make the standard selection, but again the source for use of

the RAM file for a virtual reference depends on whether it is an AC reference or a cache hit: for the former the mixers put out all 0s, but for the latter they combine two VMA bits with a 1 in the most significant position, as the cache occupies the top half of the RAM file.

5.4.4 Ten-bit Logic

This logic is a small scale arithmetic unit controlled by the microword number field in the same way that the AD and other fields control the 2901s. Of course those other fields always control the AU, whereas the 10-bit logic is manipulated by the number field only when that field is not being used for something else. This smaller arithmetic unit performs computations on exponents, counts steps in shift and arithmetic operations, and manipulates 7-bit bytes with zero alignment, which can therefore be handled much more efficiently than other sizes.

The 10-bit logic comprises the two sets of mixers and adder on DPM3 and the SC and FE registers at the bottom on DPM4. The adder is made up of ALUs, but these are limited to the seven functions listed in the table at the upper left because selection is made by only three bits. The SCAD outputs are available to the two registers, which are themselves inputs to the adder via the mixers. SCAD also goes to the main data path in both byte and exponent

positions via DPM. Both rows of mixers on DPM3 handle 10-bit quantities, but the lower one requires eight inputs only for seven positions, and bits 0, 8 and 9 are handled by the 4X2 mixer at the left end. Most of the inputs to both sets are from DP, but they involve different parts of DP for different purposes. The upper set can receive FE, the exponent part of a word from DP always in positive form via the XOR gates at the left, the effective number of shifts in a shift or rotate instruction, and the size part of a byte pointer. The lower adder can receive SC, the right ten bits of the number field, octal 44 for generating an initial byte pointer, and a 7-bit byte from any position. Note that the inputs for 44 also receive DP 06 at the right mixer so as not to disrupt the size field when a position field is inserted in a byte pointer.

5.4.5 Program Flags

DPE9 shows the program flags and the multitude of gates through which they are set and cleared. There are essentially two ways in which the flags are manipulated: by conditions resulting from arithmetic and other operations in the hardware, and direct manipulations by the microcode for saving and restoring or, for example, setting the First Part Done flag for later control of its own activities. Direct microcode control is via the number field as listed at the upper left on the print. Hardware conditions come into play

on the selection of various tests by the microcode; the large number of such conditions is listed in detail with the discussion of the program flags in Section 2.9 of the System Reference Manual.

There are however a few special considerations that should be mentioned. Because AU is 40 bits, the net at the upper right corner detects overflow from a discrepancy between DP SIGN and DP 00, and determines the presence of carry 1 by overflow being opposite carry 0 (which is available as CARRY OUT); these signals are derived from DP signals and are therefore valid only if the adder is doing an arithmetic function and its output is on DP. Note that the gate that detects overflow in arithmetic shifting (C7) checks for opposing states of DP bits 1 and 2 but these are actually bits 0 and 1 of the word being shifted. Decoding of trap signals from the trap flags at the lower right requires trap enables from both the processor and the console.

5.9 KS10 POWER SYSTEM

A simplified block diagram of the KS10 power distribution system is shown in Figure 5-p1. Input power requirements and specifications are given below.

Device	Line Voltage	Freq.	RMS Current	Surge Current	Surge Duration	KVA
KS10-AA	104-126 VAC	60 Hz	9.90A	25 A	6 cycles	1.14 KVA
KS10-AB	207-253 VAC	50 Hz	4.95 A	12.5 A	6 cycles	1.14 KVA

The major power system components are as follows:

1. 861-C (60 Hz) or 861-B (50 Hz) power control.
2. H7130 power supply and 5413261 power distribution module.
3. H765A (60 Hz) or H765B (50 Hz) power supply.
4. Blower for CPU and memory.

The H7130 power supply is used to power the KS10PA card cage. The H765A power supply is used to power the BA11-KE drawer (115 VAC version); the H765B power supply is used to power the BA11-KF drawer (230 VAC version).

The location of the major power system components are shown in Section 2 and on the KS10 Unit Assembly drawing.

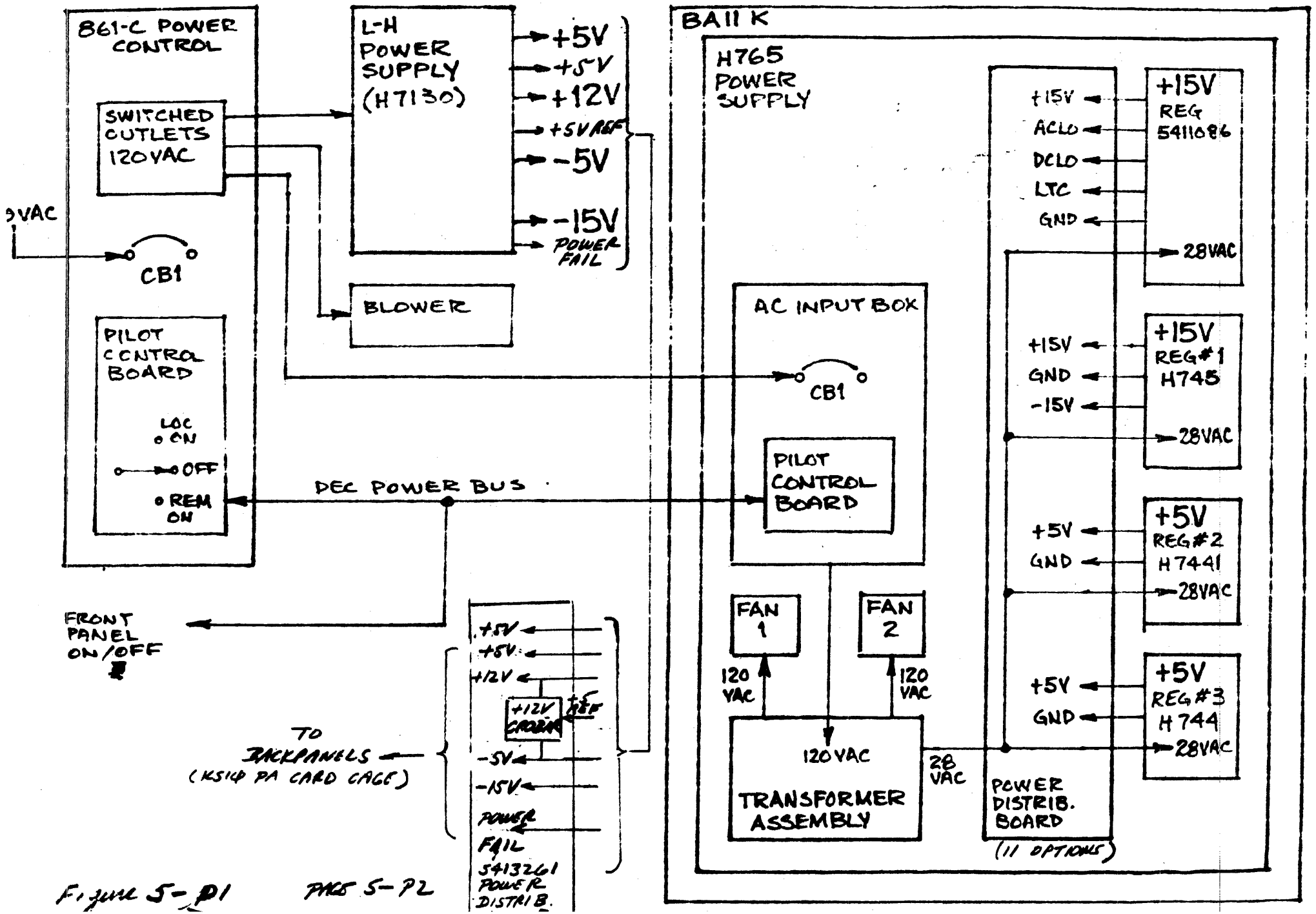

861-C POWER CONTROL
SWITCHED OUTLETS 120VAC
VAC
CB1
PILOT CONTROL BOARD
LOC ON
OFF
REM ON
L-H POWER SUPPLY (H7130)
+5V
+5V
+12V
+5V REF
-5V
-15V
POWER FAIL
BLOWER
DEC POWER BUS
FRONT PANEL ON/OFF
TO BACKPANELS (KS14P PA CARD CAGE)
+5V
+5V
+12V
+12V ERROR
+5V REF
-5V
-15V
POWER FAIL
5413261 POWER DISTRIB.
BA11 K
H765 POWER SUPPLY
AC INPUT BOX
CB1
PILOT CONTROL BOARD
FAN 1
FAN 2
120 VAC
120 VAC
120 VAC
TRANSFORMER ASSEMBLY
28 VAC
+15V
ACLO
DCLO
LTC
GND
28VAC
+15V REG 5411086
+15V
GND
-15V
28VAC
+15V REG#1 H745
+5V
GND
28VAC
+5V REG#2 H7441
+5V
GND
28VAC
+5V REG#3 H744
POWER DISTRIB. BOARD
(11 OPTIONS)

Figure 5-P1

5.9.1 861 Power Control

The 861 controls and distributes power in the KS10 cabinet. It performs the following functions:

1. Controls large amounts of power with low signal power.

2. Provides a convenient AC power distribution point.

3. Filters out electrical noise on the AC lines.

4. Disconnects power for servicing and in case of overload or overtemperature.

The 861 consists of four switched duplex outlets, two unswitched duplex outlets, a contactor with associated control circuitry, a circuit breaker, and a thermoswitch. All components are contained in a 19-in. rack-mounted box. The unit is supplied with 15 feet of power input cable with a suitable connector.

NOTE

Loads external to the KS10 cabinet are NOT to be plugged into the 861 power control.

Input power for the 861 is as follows:

Power Control	Voltage	Current (Maximum)	Phase

861-B	180 V - 264 V	16 A	1
861-C	90 v - 132 V	24 A	1

5.9.2 H7130 Power Supply and 5413261 Power Distribution Module

The H7130 is a multiple output power supply (+5 V, +5 VA, +12 V, -5 V, -15 V) which is used to power the CPU and MOS memory (KS10PA). The power supply is a off-line switching regulator that provides regulated AC to DC outputs under normal operating conditions. Input power is single phase line power.

Features of the power supply include overcurrent, overvoltage, power-fail, thermal shutdown protection, and power sequencing of +12 V with respect to the -5 V output.

Electrical specifications are as follows:

	Line Voltage	Freq.	Max. Run Current
H7130C	115 VAC +/-10%	60 Hz	3.75 Amps
H7130D	230 VAC +/-10%	50 Hz	1.87 Amps

Power sequencing for the H7130 is shown in Figure 5-p2. The +5 V (V2) and -5 V sequence up first. When the -5 V drops to -4 V, the other three voltages (+5 V (V1), +12 V, -15 V) sequence up. The -5 V is connected through a resistor on the power distribution

Power Sequencing timing diagram for H7136 power supply.

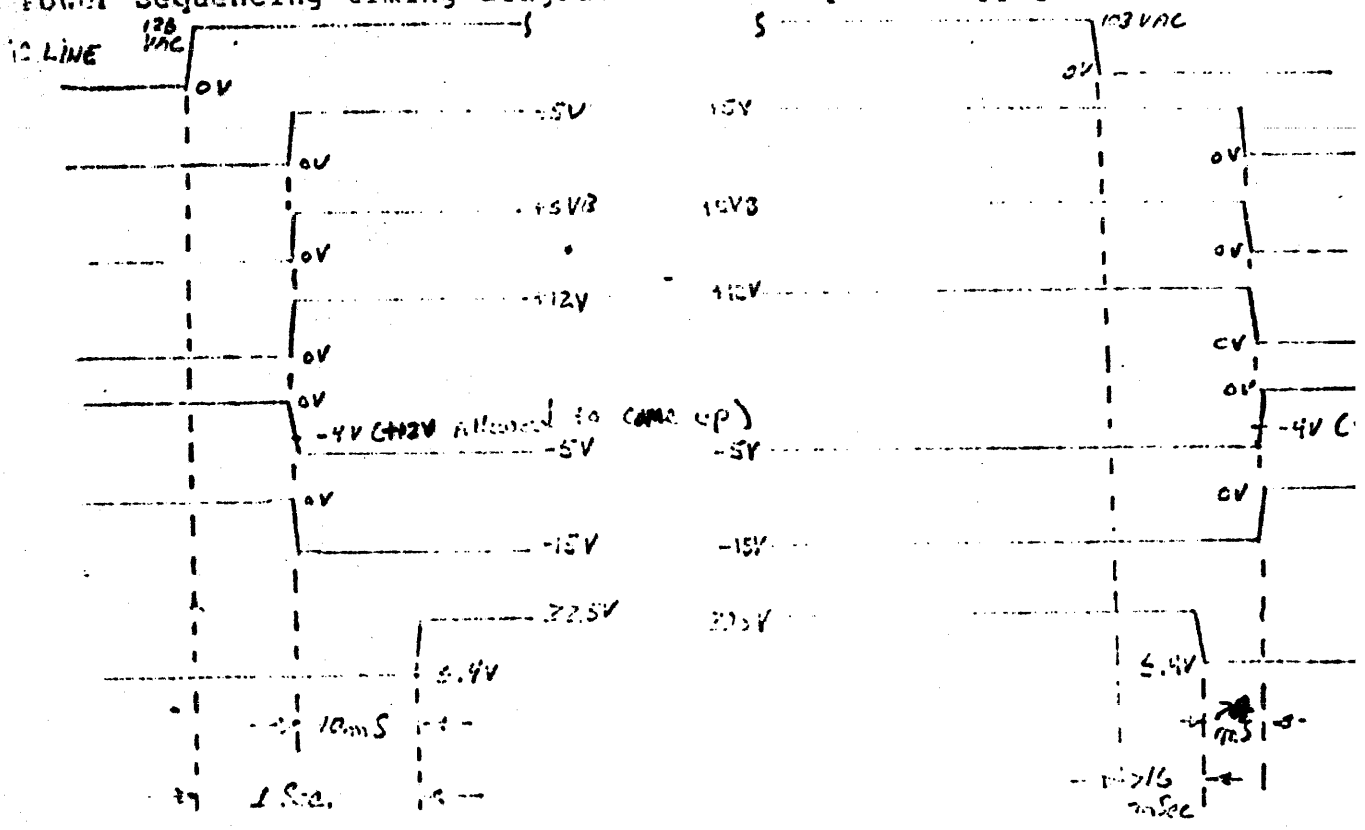
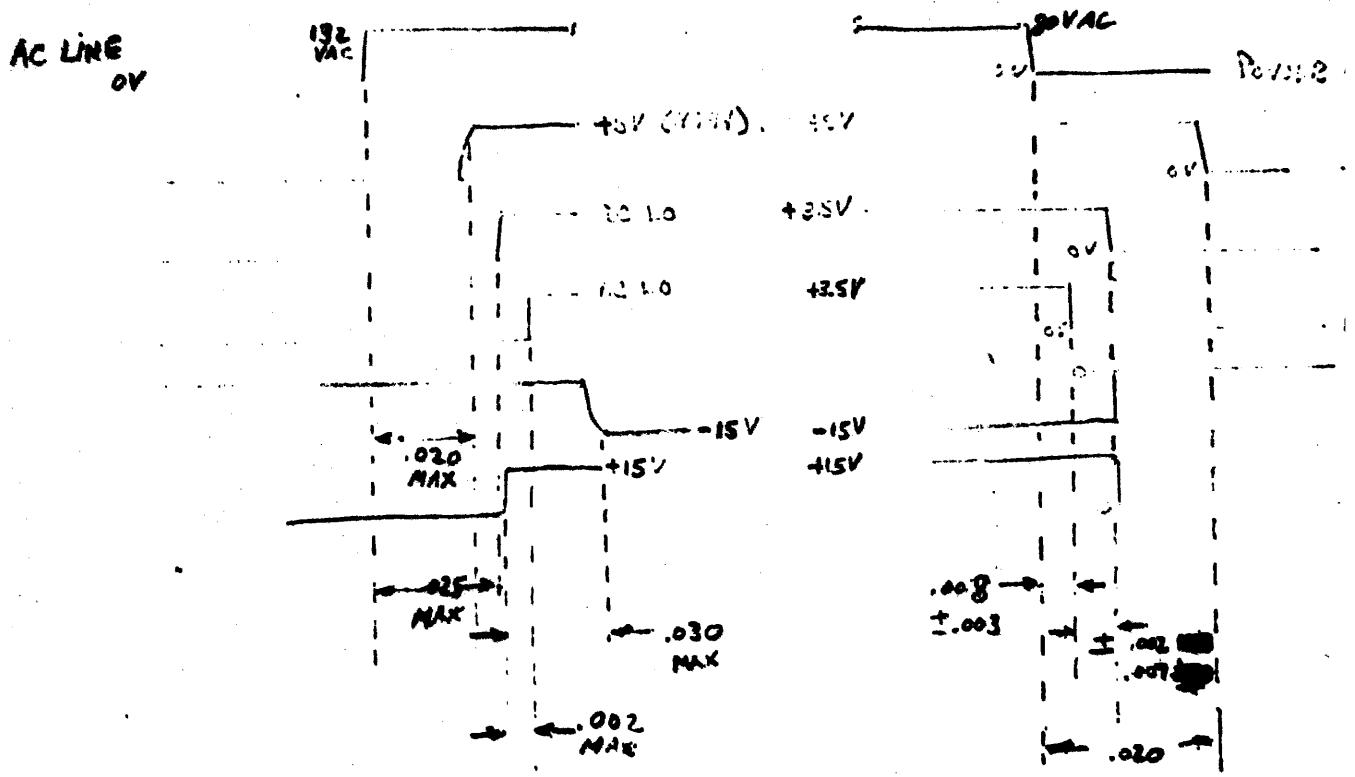


Figure 5-P2 H7134 Power Sequencing



* TIMES SHOWN ARE IN SECONDS

Figure 5-P3 H7135 Power Sequencing

module to the on-off terminal of the power supply to sequence the other three voltages.

In addition to the normal operating voltages, the H7130 supplies +5 V REF to power the 12 V crobar circuit on the power distribution module. Absence of this voltage will turn off all output regulators and short circuit the +12 V output.

As shown in Figure 5-p1, the H7130 output voltages connect to the 5413261 power distribution module. This module does the following:

1. Provides terminal blocks for interconnecting the power supply harness to the backplane harness.

2. Provides connectors to interconnect the front panel with the backplane.

3. Interconnects the thermo-sensors to the power control.

4. Has LEDs which indicate if the voltages are present. The voltages indicated are +5, +5 VA, +12 V, -15 V, -5 V. The LEDs do not indicate if the voltages are within spec.

5. Provides a circuit which senses the -5 volts, turns off the +12 V regulators, and shorts the +12 V to ground if the -5 V rises above -4 V.

5.9.3 H765 Power Supply (BA11-K)

The H765 power supply consists of five standard DEC regulators (2-H744's, 1-H745, 1-H754, and 1-5411086), a power control box (7009811), a power transformer, a power distribution board and two six-inch fans.

The H744 regulators each provide +5 V at 25 A. The H745 regulator provides -15 V at 10 A. The 5411086 regulator provides +15 V at 4 A. This board also generates the power fail signals AC LO and DC LO, and the line clock signal LTCL.

The 7009811 power control box contains a line cord circuit breaker, power relay, and relay control circuitry. Four three pin mate-n-lok connectors, two on the rear of the supply and two internal to the supply, allow low voltage control of the power on/off and emergency shutdown function. Two versions of the power control box are available, the 7009811-1 for 115 VAC operation, and the 7009811-2 for 230 VAC operation.

The power distribution board can provide DC power and control signals (AC LO, DC LO, and LTCL) to a maximum of five standard DEC system units.

Electrical Specifications for the H765 are as follows:

Line Voltage	Freq.	Max. Run Current

H765-A	90-132 VAC	47-63 Hz	3.03 Amps
H765-B	180-264 VAC	47-63 Hz	1.52 Amps

Power sequencing for the H765 power supply is shown in Figure 5-p3.

5.9.4 Blower for CPU and Memory

The blower used to provide cooling for both CPU and MOS memory operates at a voltage of 115 VAC and a current of 1.4 A.

5.9.5 Interconnection and Control

With reference to Figure 5-D1, the H765, H7130, and blower are connected to the 861 power control via the switched duplex outlets. The H765 provides a minimum of 5 ms ride-through power during a power outage condition. The H7130 power fail signal is connected to the M8616 module and provides power sequencing of the CPU. All H765 DC outputs have a minimum hold-up time of 20 ms.

The front panel ON/OFF switch interfaces to the 861 power control via the DEC power control bus. This allows all power supplies and the blower to be powered up simultaneously inside the KS10 cabinet.

To be supplied.

To be supplied.