```
                                                DEC Marlboro
                                                December
                                                1978


        KSREF.MEM for hardware REV 2
                        ***************************************************************************
                        ***               Hardware REV 3 changes included in stars              ***
                        ***************************************************************************
```

CONTENTS

Chapter	Subject	Page
8080		
	CONSOLE (8080) Overview	1-2
	CONSOLE (8080) Commands	1-3
	EB Command Truth Table	1-5
	CONSOLE Error Codes	1-7
	ERROR-mesages-during-BOOTING	1-10
	8080 <-> MONITOR Communication Area	1-11
	CONSOLE Register contents	1-12
	EC (8080) m-code breakdown	1-18
Processor		
	KS10 Processor Status Words	2-2
	UPT / EPT layout for TOP10 / TOPS20	2-5
	KS10 I/O Instruction format	2-6
	Internal register bit format	2-7
	External register bit format	2-14
	ECC (MOS memory error correction) bits	2-18
	UNIBUS Signal pin finder	2-19
Peripherals		
	Device adresses and Vectors for KS10	3-2
	KS10 option jumper and switch settings	3-3
	UART baud rate settings for CONSOLE	3-4
	Device command function codes	3-5
	RH11 RP04/5/6 register format	3-6
	RH11 RM03 register bit format	3-7
	RM03 / RP06 / RH11 register in english	3-8
	RH11 TM02/TM03 register bit format	3-9
	Magtape registers in english	3-10
	DZ-Control registers	3-12
	LP20/LP05/LP14 Bit Format	3-14
	CD11 Bit Format	3-16
	DUP11 Bit Format	3-17
	Unibus Adapter Mini Check	3-19
Micro machine		
	The micro-code HALT-status block	4-2
	m-code field listing format	4-3
	Dispatch ROM definitions	4-4
	KS10 m-word bits sorted by physical bit	4-5
	KS10 m-word bits sorted by function	4-8
Debugging		
	DDT bits and pieces	5-2
	UNIBUS adapter wrap around test	5-8
descriptions		
	descriptions of 8080 - sequences	6-2
UBA-RH11		
	DESCRIPTION OF UBA-RH11 READ-WRITE	7-1

CHAPTER 1

8080

The console (CSL) board, contains the main clock source, and arbitrates access to the backpanel bus, as well as the 8080 based console hardware. The operator controls the machine via console functions typed in via the console terminal to the 8080; he may load and check microcode; read and write memory; stop and start the clock; single step the clock; halt the machine; start at a given location; execute an instruction, etc. It does not have quite the power of the KL10 console, in that he does not have all the diagnostic features; he cannot, for example, read the PI status at any instant, nor can he at any given instant, read the VMA or PC registers or read the AC's. (Many registers, in fact, are internal to the 2901, and are not available even were the board on an extender module and an oscilloscope available.) [However, when entering the halt loop, either upon executing a halt instruction or when requested to stop by the 8080, the microcode deposits the Halt Status Block , to make these available to the console, as well as the reason for halting, before interrupting the 8080. Thus the operator does have this information available via the microcode, if not by diagnostic hardware.

The console has the same access to the backpanel as the CPU, and can therefore read or write any memory locations, and initiate any I/O that the CPU can (except I/O to the internal CPU registers PI and APR). These it may also do, if the microcode is loaded and the CPU is functional, by executing the required I/O instruction.

The console may also load microcode (2K x 96 bits), read it back to check it, and has access to the same diagnostic points as are available to the KL10 console. Thus the operator can read the current microcode location, next location, current top of stack, etc.

At power-up, the console may load microcode automatically from the disk, or may be programmed to wait for console commands to do so
.

On initial power-up , the PROM-program will wait 30 seconds before initiating an AUTO-BOOT (message "BT AUTO). A CONTROL-C or any character will abort the 30 second-wait and give the KS-prompt "KS10 " . If the PROM-program was doing a MEMORY- access at this time , the message " ?NXM " will appe on the CTY , this message can be ignored .

The AUTO-BOOT can also be inhibited by pushing the " BOOT button " , this will echoe as " BT SW " and will cause an immediate BOOT .

```
*******************************************************************************************************************************
*** REV 3 PROMS contain procedures for recovering from power-fail  ( in conjunction with battery back-up and the new power  ***
*** supply . Because there is no way to differentiate between a power-fail with battery backup involved , and the           ***
*** pushing of the front-panel " RESET switch " , the PROM program can be confused , when the " RESET switch " is pushed .  ***
*** If power is left on , and the " RESET switch " is pushed , the PROM program will wait 30 seconds , reload the MICRO-    ***
*** code , and cause the KS10 processor to begin executing instructions at location 70  ( BEWARE : MONITOR-patch required ) ***
*** Since this behaviour without MONITOR-patch and Battery back-up can be dangerous , it is recommended to abort the AUTO-  ***
*** BOOT by typing any character ( Control-C is easy to remember , but any character will do -- see above ) .              ***
*******************************************************************************************************************************
```

```
NOTE:   Several commands may be put on a single line , separated by commas

^\      ;enter CONSOLE mode
^U      ;rub out current line
^O      ;switch: first one stops cty-output , second one resumes cty-output
^S      ;stop tty-output and hangs 8080 waiting for CONTROL-Q ( see below )
^Q      ;resumes tty-output
^C      ;stops whatever the 8080 is doing
^Z      ;Enter USER-mode
RUB-OUT ;rub out previous character typed

********************************************************************************************************************************
*** BC  ;short for BOOTCHECK - floats 1's and 0's across KS10 BUS , checks page 1 of KS10 MOS-memory in an                  ***
***             ;addressing sequence , and also for ability to hold all 1's and 0's . Checks KS10 control-store via an      ***
***             ;addressing sequence for testing CRAM addressing and also all locations for ability to hold all 1's and     ***
***             ;all 0's.                                                                                                   ***
***             ;Beware: The BC-command will usually leave the entire KS10 control-store containing BAD PARITY . If the     ***
***             ;BC-command is allowed to run to completion , or should FAIL during the CRAM portion of its testing         ***
***             ;subsequent EXAMINES of control-store will yield PARITY ERRORS ! This is not necessarily because the        ***
***             ;KS10 has a problem , but because the control-store has intentionally been loaded with bad parity.          ***
***             ;                                                                                                           ***
***             ;The BC-command should be followed with a B2-command , which will attempt to load the control-store with    ***
***             ;the KS10 micro-code . This should eliminate the parity errors . If it does not ,You can always try         ***
***             ;clearing the control-store by typing a sequence of commands at the "KS10>" prompt :                        ***
***             ; Typein: MR,LC 0,DC 0 <Carriage Return>                                                                    ***
***             ;         DN 0,RP <Carriage Return> .... and then wait for 60 seconds                                       ***
***             ;         "Control C" to break the indefinite repeat                                                        ***
***             ; .... The "typical" execution time of BC-command is long ---- roughly 3 minutes ----                       ***
********************************************************************************************************************************
BT      ;Boot SYSTEM -- load CRAM from designated disk ( see DS ) via memory then
        ;load monitor boot from disk ( block/track/sector 0 ) and start at 1000
BT 1    ;same as BT , but loads SMMON and starts at 20000
********************************************************************************************************************************
*** B2  ; short for BOOTCHECK 2 . Loads a modified PRE-BOOT program , which loads a special sequence of modified             ***
***             ;functional diagnostics.                                                                                    ***
********************************************************************************************************************************
CE X    ;CACHE enable 0=OFF 1=ON <CR> current state?
CH      ;CPU clock halt
CO      ;continue(causes m-code to leave HALT-loop )
CP XX   ;CPU clock pulse (xx  NR of pulses -- default 1 pulse )
CS      ;CPU clock start
DB XX   ;deposit BUS , XX data
DC XX   ;deposit CRAM , XX is at least 32 octal characters
DF XX   ;write m-code bits according to last LF-command
DK XX   ;deposit 8080 loc ( only RAM locs stick )
DI XX   ;deposit I/O , XX data
DM XX   ;deposit MEMORY , XX data
DN XX   ;deposit next ( depending on last DK , DM or DI ) XX data
********************************************************************************************************************************
*** DR  ;deposit internal 8080 I/O register                                                                                 ***
********************************************************************************************************************************
DS      ;Disk select. Command to specify UNIT NUMBER ,RHBASE ( 1 or 3 ) and UNIBUS ADAPTER to
        ;load from when booting
EB      ;examine BUS and 8080 control registers (READS IO REGISTERS 100,303,103)
```

```
EC XX   ;examine CRAM at address XX
EC      ;examine CRAM ..curr. Control reg , no clocks .. current loc as addr.
EK      ;examine 8080 location
EK XX   ;examine 8080 address XX
EI      ;examine I/O ( last I/O address specified )
EI XX   ;exmaine I/O address XX
EJ      ;examine jumps -- prints CRAM address signals ( CURR , NXT , J , SUB )
EM      ;examine MEMORY ( last MEMORY location specified )
EM XX   ;examine MEMORY location XX
EN      ;examine next ( either from last EK , EM or EI )
*****************************************************************************************************************************
*** ER  ;examine internal 8080 I/O register                                                                              ***
*****************************************************************************************************************************
EX XX   ;execute KS10 instruction XX
FI      ;indirect FILE command ( not used yet , but may be !!)
HA      ;HALT KS10 ( execute HALT-instruction -causes m-code to write HSB and then to enter HALT-loop )
KL      ;Klinik-Command ,0=disable , 1=enable , nothing=state ( on/off )
LA XX   ;set memory address
LB      ;load bootstrap from designated disk ( see DS ) block/track/sector 0
LB 1    ;load diagnostic monitor SMMON
LC XX   ;set CRAM address
LF XX   ;selects a set ( 0-7 ) of 12 bits of m-code (see note at end ****)
LI XX   ;set I/O address
LK XX   ;set 8080 address
*****************************************************************************************************************************
*** LR  ;set internal 8080 I/O register                                                                                  ***
*****************************************************************************************************************************
LT      ;lamp test ,lights three lamps of front panel
MB      ;load only boots-trap of currently selected magtape
MK XX   ;marks micro-code location XX ( sets bit 95)
MM      ;set KLINIK state into APT-protocol (beware--not in the FIELD !!)
MR      ;MASTER RESET
MS      ;Magtape Select. Command to specify UNIT NUMBER , RH BASE  , UNIBUS ADAPTER,
        ;SLAVE NUMBER and DENSITY of magtape YOU would like to boot from
*****************************************************************************************************************************
*** The "UNIT?" message has been shortened to "TCU?" -- short for " Tape Control Unit ?".                                ***
*****************************************************************************************************************************
MT      ;Boot System from selected magtape
MT 1    ;BOOT diagnostic monitor SMMAG from magtape
PE X    ;enable parity ( 00=disable 7=enable all  1=DP-par 2=CRM-par
        ; 4=clock-par error stop
PM      ;pulse m-code ( issue single CP & EJ )
PW      ;clears KLINIK password , or sets it ( 6 char's max )
RC      ;read CRAM direct , functions 0-17 ( no resets, no load diag addr , no CPU clock ) ( see note at end ****)
RP      ;repeat - repeats last command , or line of commands which it delimits
        ;any character ( except CNTRL-O ) typed will stop repeat
        ;CNTRL-O acts as switch ( first one= stop output , next one=resume output)
        ;EXAMPLE: EM 0,EK 0 , EC 0,RP will repeat execution of this line
***************************************************************************************************************************
*** SC  x; command enables/disables SOFT CRAM ERROR recovery  ( x=0 off , x=1 on SC=state of switch )                  ***
***************************************************************************************************************************
```

```
SI      ;single instruction
SH      ;SHUTDOWN ( deposit non-zero data in memory location 30 )
        ;causing TOPS20 to shut down without warning
SM XX   ;start m-code at XX ( SM 1 causes dump of HALT-status block !!)
        ;Default is 0 -- Start m-code
ST XX   ;start KS10 at address XX
TE X    ;1 MSEC enable. 0= OFF , 1=ON , <CR> current state?
TP X    ;TRAPS enable 0=OFF , 1=ON (enables paging ) , <CR> current state?
TR XX   ;TRACE - repeats CP & EJ commands till any character typed
        ;XX ( if typed ) is desired CRAM stop-adress
TT      ;tranfer KLINIK-line to USER-MODE
UM XX   ;unmarks micro-code location XX
VD      ;verify CRAM ( using same load-path as previous ) from disk
VT      ;verify CRAM ( using same load-path as previous ) from magtape
ZM      ;zero KS10 MOS memory ( beware --- slow )


***********************************************************************
************            EB COMMAND TRUTH TABLE            *************
***********************************************************************
*                                                                     *
*      100       *      303       *      103       *    TRUE STATE    *
*                                                                     *
* 0X * XXX * XXX *                                      CSL REC PE    *
* X0 * XXX * XXX *                                      UBA3 PE       *
* XX * XX0 * XXX *                                      MEM PE        *
* XX * X0X * XXX *                                      CR/M PE       *
* XX * XXX * XX0 *                                      UBA 2 PE      *
* XX * XXX * X0X *                                      CRA PE        *
* XX * XXX * 0XX *                                      DP PE         *
*                * XX * XXX * XX0 *                     DPM PE        *
*                * XX * XXX * X0X *                     DPE/M CLK ENB *
*                * XX * XXX * 1XX *                     CRAM CLK ENB  *
*                                 * 0X * XXX * XXX *    UBA 1 PE      *
*                                 * X0 * XXX * XXX *    UBA 4 PE      *
***********************************************************************



                        0 = LOW
                        1 = HIGH
```

***** LF-command CRAM Bits

LF	CRAM bits
0	00-11
1	12-23
2	24-35
3	36-47
4	48-59
5	60-71
6	72-83
7	84-95

RC-Command CRAM Data

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

```
?A/B            A & B copies of CRAM bits did not match
?BC             BOOT check failed
?BFO            Input buffer overflow
?BN             received bad number on input
?BT             device error or timeout during BOOT operation
?BUS            BUS polluted on power up
?CHK            PROM checksum failed
?DNC            did not complete HALT
?DNF            did not finish instruction
?IA             illegal argument ( address out of range etc )
?IL             ILLEGAL Instruction
?KA             KEEP ALIVE failed
?MEM REFRSH ERR memory refresh error ( MEM BUSY stayed set too long , because it didn't
                release data on a write to memory
********************************************************************************************************************************
*** The message "?MEM REFRSH ERR" has been shortened to "?MRE" short for Memory Refresh Error ?                              ***
********************************************************************************************************************************
?NA             KLINIK Link Not Available
?NBR            Console was not granted BUS on a request
?NDA            received no data acknowledge on mem request
*******************************************************************************************************************************
***  ?NR-SCE            Non-Recoverable CRAM-Error detected by Soft Cram Error recovery                                     ***
***             - A CRAM Error is not recoverable , if it occurs while memory is busy , or causes a "MRE" , or if the       ***
***             disk is in a condition , which the PROM-program can not save and later restore                              ***
*******************************************************************************************************************************
?NXM            referenced nonexistent memory location

?PAR ERR        report clock-freeze due to parity error and type out READ IO of 100,303,103
*******************************************************************************************************************************
***             This message occurs on any non CRA/CRM parity errors , and also occurs if a CRA/CRM parity error happens    ***
***             at the same address twice in a row . ( i.ethe PROM program assumes , that , if the error happened twice in  ***
***             a row at the same location , it must be a hard bit failure and recovery could never be achieved             ***
*******************************************************************************************************************************
?PWL            Password length error
?RA             command requires argument ( YOU gotta type something )
?RUNNING        trying to do a command , that may screw up
?UI             unknown interrupt
```

OTHER 8080 CONSOLE MESSAGES

```
BUS 0-35                message header for EB command
BT AUTO                 beginning Auto Boot sequence ( 15 sec after power-up)
BT SW                   message says BOOTING , using BOOT switch
C CYC                   typed on DB-command if COM/ADR cycle blew
CYC                     cycle type for DB command
HLTD                    message "HALTED/XXXXXX " where xxxxxx is data
KS10>                   prompt message
OFF                     message , says this signal is off
ON                      message , says this signal is on
PC                      Program Counter ( what else )
RCVD                    data received on bus
*******************************************************************************************************************************
*** SCE xxxxxx          Soft Cram Error recovery performed and xxxxxx stands for the error-address .                        ***
*******************************************************************************************************************************
SENT                    data sent to bus
>>UBA?                  query for unibus adapter
>>UNIT?                 query for unit to use
>>RHBASE?               query for RH11 to use
>>DENS?                 query tape density
>>SLV?                  query tape slave
```

```
        On an error-condition , detected by the 8080 , the Fault-light will go on and a message of the form

                ?BT XXXYYY      will be printed on the CTY.

The following error-codes are only "rough" pointers , they can be caused by any of the following problems:

        Disk not a disk at all ,        wrong unit selected ( DS-command ! ) ,  Home blocks not readable or not there
        Home blocks not set by SMFILE for 8080 ,        8080 File-system garbage
********************************************************************************************************************************
*** Disk:                                                                                                                    ***
*** XXX=001     Disk error encountered while trying to read HOME-blocks                                                      ***
***                                                                                                                          ***
*** XXX=002     Disk error encountered while trying to read the page of                                                      ***
***             pointers , which make up the "8080-File-System"                                                              ***
***                                                                                                                          ***
*** xxx=003     Disk error encounterd while trying to read a page of                                                         ***
***             m-code                                                                                                       ***
***                                                                                                                          ***
*** xxx=004 Microcode did not successfully start up -- usually after BT- , MT- , MB- ,or LB-command                           ***
***       also after LB , before system-micro code has been loaded.                                                          ***
***                                                                                                                          ***
*** xxx=010     Disk error encountered while trying to read PRE-BOOT                                                         ***
***                                                                                                                          ***
*** yyy are the lower 8 bits of the 8080 adress of the failing "Channel Command List" operation. Normally it is here         ***
***     a good bet to do an "EI" to get the contents of the RH11 register that has the error-bits set !                      ***
***                                                                                                                          ***
*** Magtape:                                                                                                                 ***
***                                                                                                                          ***
*** The following ERROR-messages can point to the following problem-areas:                                                   ***
***                                                                                                                          ***
***     Magtape is no magtape at all ,wrong unit selected ( see MS-command ) ,Magtape is not bootable ( no m-code ,          ***
***     no PRE-BOOT )                                                                                                        ***
***                                                                                                                          ***
*** xxx=001     Error trying to read m-code first page                                                                       ***
***                                                                                                                          ***
*** xxx=003     Error trying to read additional pages of m-code                                                              ***
***                                                                                                                          ***
*** xxx=004  .... see disk-error description above ...                                                                       ***
***                                                                                                                          ***
*** xxx=010     Error trying to read in PRE-BOOT program                                                                     ***
***                                                                                                                          ***
*** yyy see above ( disk-section )                                                                                           ***
********************************************************************************************************************************
```

```
*********************************************************************************************************************************
***  BOOTCHECK ERROR-messages are of the form:  ?BC WWYYYY                                                                    ***
***                                                                                                                           ***
*** WW=00       Indicates a failure during the KS10-Bus-check . BOOTCHECK is floating a ONE and then a ZERO across the     ***
***             KS10-BUS. YYYY will be the failing BIT , in OCTAL --- Numbers will range from 0 to 43 , corresponding      ***
***             to DECIMAL 0-35.                                                                                           ***
***                                                                                                                           ***
*** WW=04       Indicates a failure during the MEMORY-check . BOOTCHECK is trying to verify page1 of MOS-memory and tests  ***
***             address 1000-1777 for ability to hold all ONES and all ZEROES , and to sequence thru that page of memory   ***
***             correctly . YYYY will be the failing memory address.                                                       ***
***                                                                                                                           ***
*** WW=10       Indicates a failure during the CRAM-check portion of BOOTCHECK . YYYY will be the failing CRAM-address     ***
*********************************************************************************************************************************
```

PRE-BOOT is loaded from Disk or Magtape (see 8080 commands DS , MS , BT , BT 1 , MT , MT 1)

PRE-BOOT is written onto the disk using "SMFILE.EXE" , it also is written on "standard" Diagnostic-tapes and onto the "MONITOR-DISTRIBUTION"-tapes.

PRE-BOOT is loaded by the 8080 into MEMORY-locations 1000 and up , and starts at 1000 . The ERROR-halts are:

1001	found "bad" core-transfer address (page 1 is illegal - cant overload PRE-BOOT)
1003	No RH11 Base Address
1004	Magtape Skip failure
1002	all other failures

At ERROR-halt time the following MEMORY-Locations contain the useful INFO :

	Disk-Booting	Magtape-Booting
100	"8080" disk-address	Not used
101	Memory transfer address	same
102	Index-pointer	same
103	RPCS1-register	MTCS1-register
104	RPCS2-register	MTCS2-register
105	RPDS - register	MTDS - register
106	RPER1-register	MTER1-register
107	RPER2-register (RP06 only)	Not used
110	RPER3-register	Not used
111	UBA Page RAM loc 0	same
112	UBA-status register	same
113	Version Nr. of PRE-BOOT	same

Note: The Version Nr. of PRE-BOOT will be the same as the Version Nr. of SMFILE. The "8080" disk-adress is in the form " CY SA TA "

THEREBY IT WILL BE POSSIBLE TO ASK A CUSTOMER WITH A PRE-BOOT FAILURE , TO DO AN :

```
        EM 77
        EN,RP
        ...... AND TYPE SOMETHING AFTER ADRESS 115
        .... AND THEN TELL US WHAT  HE SEE'S
```

The 8080 maintains and services an in-core communication area.
Currently used are words 31 to 40.

Word Nr.	Meaning
31	Keep Alive and Status word
32	KS-10 CTY input word (from 8080)
33	KS-10 CTY output word (to 8080)
34	KS-10 KLINIK user input word (from 8080)
35	KS-10 KLINIK user output word (to 8080)
36	BOOT RH-11 Base Address
37	BOOT Drive Number
40	Magtape Boot Format and Slave Number

Word 31	Keep Alive and Status word
Bit 4	Reload
Bit 5	Examine Keep Alive
Bit 6	KLINIK active
Bit 7	PARITY Error detect enabled
Bit 8	CRAM Parity Error detect enabled
Bit 9	DP Parity Error detect enabled
Bit 10	CACHE enabled
Bit 11	1 msec enabled
Bit 12	TRAPS enabled
Bit 20-27	Keep Alive
Bit 32	BOOT SWITCH BOOT
Bit 33	POWER FAIL
BIT 34	Forced RELOAD
BIT 35	Keep Alive failed to change

Word 32	KS-10 CTY input word (from 8080)
Bits 20-27	0 --- no action , 1 --- CTY character pending
Bits 28-35	CTY-character

Word 33	KS-10 CTY output word (to 8080)
Bits 20-27	0 --- no action , 1 --- CTY character pending
Bits 28-35	CTY-Character

Word 34	KS-10 KLINIK user input word (from 8080)
Bits 20-27	0 --- no action , 1 --- KLINIK character , 2 --- KLINIK active , 3 --- KLINIK carrier loss
Bits 28-35	KLINIK-Character

Word 35	KS-10 KLINIK user output word (to 8080)
Bits 20-27	0 --- no action , 1 --- KLINIK character , 2 --- Hangup request
Bits 28-35	KLINIK-Character

OUTPUT process KS10 ==> 8080

Load character and flag into 33 , set 8080-interrupt , 8080 examines
33 and gets character , clears interrupt , sends character to hardware ,
clears 33 and sets KS-10 interrupt.

INPUT process 8080 ==> KS10

8080 gets interrupted "TTY-char available" , 8080 gets character and
delivers into input-word (31) with flag(s) and sets KS-10 interrupt.

```
READ-I/O=0
*******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5      !    DBUS4      !   DBUS3       **   DBUS2       !   DBUS1       !   DBUS0       !
!           !              **                !               !               **               !               !               !
!           !              **                !               !               **               !               !               !
!R DATA 28  !   R DATA 29  **    R DATA 30   !   R DATA 31   !   R DATA 32   **   R DATA 33   !   R DATA 34   !   R DATA 35   !
!           !              **                !               !               **               !               !               !
!           !              **                !               !               **               !               !               !
*******************************************************************************************************************************

READ-I/O=1
*******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5      !    DBUS4      !   DBUS3       **   DBUS2       !   DBUS1       !   DBUS0       !
!           !              **                !               !               **               !               !               !
!           !              **                !               !               **               !               !               !
!R DATA 20  !   R DATA 21  **    R DATA 22   !   R DATA 23   !   R DATA 24   **   R DATA 25   !   R DATA 26   !   R DATA 27   !
!           !              **                !               !               **               !               !               !
!           !              **                !               !               **               !               !               !
*******************************************************************************************************************************


READ-I/O=2
*******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5      !    DBUS4      !   DBUS3       **   DBUS2       !   DBUS1       !   DBUS0       !
!           !              **                !               !               **               !               !               !
!           !              **                !               !               **               !               !               !
!R DATA 12  !   R DATA 13  **  R DATA 14     !   R DATA 15   !   R DATA 16   **   R DATA 17   !   R DATA 18   !   R DATA 19   !
!           !              **                !               !               **               !               !               !
!           !              **                !               !               **               !               !               !
*******************************************************************************************************************************


READ-I/O=3
*******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5      !    DBUS4      !   DBUS3       **   DBUS2       !   DBUS1       !   DBUS0       !
!           !              **                !               !               **               !               !               !
!           !              **                !               !               **               !               !               !
!R DATA  4  !   R DATA  5  **    R DATA  6   !   R DATA  7   !   R DATA  8   **   R DATA  9   !   R DATA 10   !   R DATA 11   !
!           !              **                !               !               **               !               !               !
!           !              **                !               !               **               !               !               !
*******************************************************************************************************************************


READ-I/O=100
*******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5      !    DBUS4      !   DBUS3       **   DBUS2       !   DBUS1       !   DBUS0       !
!    CSL    !              **                !               !               **               !               !               !
!  REC PE   !   UBA3 PE    ** SPARE          !      CR/M     !    MEM        **   DP PARITY   !   CRA PARITY  !     UBA2 PE   !
!           !              **                ! PARITY ERR    ! PARITY ERR    **  ERR L        !   ERR L       !               !
!       L   !       L      **       L        !      L        !      L        **               !               !       L       !
!           !              **                !               !               **               !               !               !
*******************************************************************************************************************************
```

READ-I/O=101

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
PI REQ 1	PI REQ 2	PI REQ 3	PI REQ 4	PI REQ 5	PI REQ 6	PI REQ 7	MMC REF ERR B H

READ-I/O=102

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
R AC LO	R RESET	R MEM BUSY	R I/O BUSY	R BAD DATA	R COM ADR	R I/O DATA	R DATA

READ-I/O=103

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
UBA1 PE L	UBA4 PE L	R PAR RIGHT	R PAR LEFT	R DATA 0	R DATA 1	R DATA 2	R DATA 3

READ-I/O=300

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
CTY STOP BIT # (SW)	CTY CHAR LENGTH (SW)	REMOTE DIAG STOP BIT # (SW)	REMOTE DIAG CHAR LENGTH (SW)	CSL4 HALT LOOP	RUN (1) H	EXECUTE H	CONTINUE H

READ-I/O=301

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
10 INTERRUPT H	NXM H	SPARE H	BUS REQ	PE (1) OCCURRED H	CONSOLE ENABLE H (SW)	BOOT H (SW)	DATA ACK H

READ-I/O=302

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
0	0	0	0	REMOTE PROTECT H	REMOTE EN H	TERMINAL CARRIER	REMOTE DIAG CARRIER

READ-I/O=303

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
0	0	0	0	R CLK ENB (0) H	CRAM CLK ENB H	DPE/M CLK ENB L	DPM PAR ERR ERR L

WRT-I/O=204

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
0	0	CRAM WRITE	CRM ADR LOAD	SS MODE	DP RESET H	STACK RESET H	CRAM RESET H

WRT-I/O=205

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
0	0		TRAP EN	diag 10	DIAG 4	DIAG 2	DIAG 1

WRT-I/O=206

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
0	0	0	0	0	0	SINGLE CLK H	CLK RUN H

WRT-I/O=210

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
CHECK NXM	CONSOLE REQ	T ENB FOR COM/ADR	T ENB FOR DATA CYCLE	CRA T CLK	CRA R CLK	LATCH DATA SENT	R CLK ENB L

WRT-I/O=212

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
0	0	0	0	0	RUN	EXECUTE	CONTINUE

WRT=100

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
RESET	PE DETECT ENABLE	CRM PE DETECT	DP PE DETECT	CACHE ENABLE	1 MSEC ENABLE H	0	0

WRT=101

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
0	0	0	0	MODEM DTR	STATE	REMOTE	FAULT

WRT=102/103 DATA/ADR

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
DATA 28	DATA 29	DATA 30	DATA 31	DATA 32	DATA 33	DATA 34	DATA 35

WRT=104/105 DATA/ADR

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
DATA 20	DATA 21	DATA 22	DATA 23	DATA 24	DATA 25	DATA 26	DATA 27

WRT=106/107 DATA/ADR

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
DATA 12	DATA 13	DATA 14	DATA 15	DATA 16	DATA 17	DATA 18	DATA 19

WRT=110/111 DATA/ADR

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
DATA 4	DATA 5	DATA 6	DATA 7	DATA 8	DATA 9	DATA 10	DATA 11

WRT=112/113 DATA/ADR

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
0	0	0	0	DATA 0	DATA 1	DATA 2	DATA 3

WRT=114/115 DATA/ADR

DBUS7	DBUS6	DBUS5	DBUS4	DBUS3	DBUS2	DBUS1	DBUS0
0	0	0	0	BAD DATA CYCLE	COM/ADR CYCLE	I/O DATA CYCLE	DATA CYCLE

```
WRT=116
******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5     !     DBUS4     !    DBUS3      **    DBUS2      !    DBUS1      !    DBUS0      !
!           !              **               !               !               **               !               !               !
!           !              **               !               !               **               !               ! CSL INTERRUPT !
!      0    !        0     **        0      !        0      !        0      **      0        !        0      !    THE 10     !
!           !              **               !               !               **               !               !               !
!           !              **               !               !               **               !               !               !
******************************************************************************************************************************


WRT-I/O=200 CTY UART WRITE STATUS REGISTER(DATA BUFFER IS I/O 201)
******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5     !     DBUS4     !    DBUS3      **    DBUS2      !    DBUS1      !    DBUS0      !
!           !              **               !               !               **               !               !               !
!           !              **               !               !               **               !               !               !
! HUNT MODE !     UART     **   REQUEST     !      RESET    !      SEND     **     RECEIVE   !    TERMINAL   !     TRANSMIT  !
!  ON(SYNC) !    RESET     **   TO SEND L   !      ERRORS   !  BREAK CHAR   **     ENABLE    !     READY     !     ENABLE    !
!           !              **               !               !       H       **               !      L        !               !
******************************************************************************************************************************


RD-I/O=200 CTY UART READ STATUS REGISTER(DATA BUFFER IS I/O 201)
******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5     !     DBUS4     !    DBUS3      **    DBUS2      !    DBUS1      !    DBUS0      !
!           !              **               !               !               **               !               !               !
!           !              **               !               !               **               !               !               !
!  DATA SET !     SYNC     **    FRAMING    !    OVERRUN    !     PARITY    **  TRANSMITTER  !   RECEIVER    !  TRANSMITTER  !
!   READY   !     DETECT   **    ERROR      !    ERROR      !     ERROR     **    EMPTY      !    READY      !    READY      !
!           !              **               !               !               **               !               !               !
******************************************************************************************************************************


WRT-I/O=202 REMOTE UART WRITE STATUS REGISTER(DATA BUFFER IS I/O 203)
******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5     !     DBUS4     !    DBUS3      **    DBUS2      !    DBUS1      !    DBUS0      !
!           !              **               !               !               **               !               !               !
!           !              **               !               !               **               !               !               !
! HUNT MODE !     UART     **   REQUEST     !      RESET    !      SEND     **     RECEIVE   !    TERMINAL   !     TRANSMIT  !
!  ON(SYNC) !    RESET     **   TO SEND L   !      ERRORS   !  BREAK CHAR   **     ENABLE    !     READY     !     ENABLE    !
!           !              **               !               !       H       **               !      L        !               !
******************************************************************************************************************************


RD-I/O=202 REMOTE UART READ STATUS REGISTER(DATA BUFFER IS I/O 203)
******************************************************************************************************************************
! DBUS7     !   DBUS6      **     DBUS5     !     DBUS4     !    DBUS3      **    DBUS2      !    DBUS1      !    DBUS0      !
!           !              **               !               !               **               !               !               !
!           !              **               !               !               **               !               !               !
!  DATA SET !     SYNC     **    FRAMING    !    OVERRUN    !     PARITY    **  TRANSMITTER  !   RECEIVER    !  TRANSMITTER  !
!   READY   !     DETECT   **    ERROR      !    ERROR      !     ERROR     **    EMPTY      !    READY      !    READY      !
!           !              **               !               !               **               !               !               !
******************************************************************************************************************************
```

The heading is the BIT-Nr , followed by the BIT-definitions . The Bottom part links RAM-chips --
(top for even RAM - addresses , lower part for odd ones)

```
*012!345!678!911*111!111!112!222*222!222!333!333*333!344!444!444*445!555!555!555*666!666!666!677*777!777!778!888*888!888!999!999*
              01*234!567!890!123*456!789!012!345*678!901!234!567*890!123!456!789*012!345!678!901*234!567!890!123*456!789!012!345*
---------------------------------------------------------------------------------------------------------------------------------
[    J FIELD   ]*[]'![ ]![ ]![ ]*^''![ ]![ ]![ ]*[ SPECIAL #   ]*'''!'''![SP # ]*[ ALU CON ]![ ]*[][!  ]!''[!  ]*[ ]![ ]!'''!'^'*
*   !   !   !   *   !---!   !---1   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *
*[NEXT U BASE  ]*TI !SKI!SPC!DIS*C  !DIS!SPC!SKI*[2 PARTS****  ]*   !   ![2PART]*FUN!LSC!RSC!DBM*DB[! A]!  [! B]*RAM!ALU!   ! C *
*   !   !   !   *ME !ENA!ENA!ENA*R  !SEL!SEL!SEL*[  !   !   !  ]*   !   ![  !  ]*   !   !   !SEL*MX[!AD]!  [!AD]*SRC!DST!   ! R *
*012!345!678!911*01C!421!421!421*ACM!421!421!421*678!911!111!111*RMP!PDM!012!345*421!421!421!421*211!421!DD1!421*421!421!SFP!PMM*
*   !   !   ! 01*  A!000!000!000* RE!   !   !   *   ! 01!234!567*AUA!AIU!   !   *   !   !   !   *  0!   !PP0!   *   !   !CEA!A A*
*   !   !   !   *  L!   !   !   *BYM!   !   !   *   !   !   !   *MLR!RVL!   !   *   !   !   !   *   !   !   !   *   !   !  R!RBR*
*   !   !   !   *  L!   !   !   *D3 !   !   !   *   !   !   !   * T ! IT!   !   *   !   !   !   *   !   !CC !   *   !   !EE ! DK*
*   !   !   !   *   !   !   !   * 8W!   !   !   *   !   !   !   *WIE!EDI!   !   *   !   !   !   *   !   !LL !   *   !   !NNC!C  *
*   !   !   !   *   !   !   !   *P R!   !   !   *   !   !   !   *R N!NE !   !   *   !   !   !   *   !   !KK !   *   !   !  H!HP *
*   !   !   !   *   !   !   !   *A T!   !   !   *   !   !   !   *TS !  P!   !   *   !   !   !   *   !   !   !   *   !   !  K!KA *
*   !   !   !   *   !   !   !   *R  !   !   !   *   !   !   !   * HL!R R!   !   *   !   !   !   *   !   !EE !   *   !   !   ! R *
*   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   * I !  E!   !   *   !   !   !   *   !   !NN !   *   !   !  L!R  *
*   !   !   !   *   !   !   !   *B  !   !   !   *   !   !   !   * F !  C!   !   *   !   !   !   *   !   !   !   *   !   !   ! B *
*   !   !   !   *   !   !   !   *I  !   !   !   *   !   !   !   * T !   !   !   *   !   !   !   *   !   !LR !   *   !   !   ! I *
*   !   !   !   *   !   !   !   *T  !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *   !   !   ! T *
*   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *   !   !   !   *
*<<<<<<<<< ON CRA BOARD (M8622)>>>>>>>>>>>>>>>>>*<<<<<<<<<<<<<<<<<<<<<<<<<ON CRM BOARD (M8623)>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>*
*      RC0      *     RC4       *  RC5 A-LATCH  *  RC12 A-LATCH *     RC14      *      RC15     *     RC16      *     RC17      *
*               *               *  RC6 B-LATCH  *  RC13 B-LATCH *               *               *               *               *
*     LF0       *      LF1      *      LF2      *     LF3       *      LF4      *     LF5       *      LF6      *      LF7      *
---------------------------------------------------------------------------------------------------------------------------------
*<<<<<<<<< ON CRA BOARD (M8622)>>>>>>>>>>>>>>>>>*<<<<<<<<<<<<<<<<<<<<<<<<<ON CRM BOARD (M8623)>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>*
*888!888!777!777*666!666!555!555*444!444!333!333*777!777!777!777*666!666!666!666*555!555!555!555*444!444!444!444*333!333!333!333*
*000!001!000!001*000!001!000!001*000!001!000!001*000!011!111!222*000!011!111!222*000!011!111!222*000!011!111!222*000!011!111!222*
*135!791!135!791*135!791!135!791*135!791!135!791*246!802!468!024*246!802!468!024*246!802!468!024*246!802!468!024*246!802!468!024*
*===!===!===!===*===!===!===!===*===!===!===!===*===!===!===!===*===!===!===!===*===!===!===!===*===!===!===!===*===!===!===!===*
*888!888!777!777*666!666!555!555*444!444!333!333*777!777!777!777*666!666!666!666*555!555!555!555*444!444!444!444*333!333!333!333*
*000!011!000!011*000!011!000!011*000!011!000!011*000!011!111!222*000!011!111!222*000!011!111!222*000!011!111!222*000!011!111!222*
*246!802!246!802*246!802!246!802*246!802!246!802*357!913!579!135*357!913!579!135*357!913!579!135*357!913!579!135*357!913!579!135*
---------------------------------------------------------------------------------------------------------------------------------
```

There are also functions:

01====Read output of CRAM J field
02====Read sbr ret. reg. (current location from last call) Stack is 16 locs deep
03====Read current m-code instruction location.

07== NOT USED BY YOU, AS IT IS THE SELECT FORCED UP TO SELECT THE CRM BOARD MIXER INTO THE CRA BOARD MIXER FOR FUNCTIONS 10-17.

10====C RAM PARITY CHECKERS A-F(YOU CAN SPOT WHICH FIELD HOSED THE PARITY FROM THESE. !   ,   P,ABC,DEF!
NOTE:::: THESE CHECKERS !!!!!!DO NOT!!!!!!! LOOK AT THE OUTPUTS OF THE CRA BOARD(BITS 0-35 OF THE C RAM),
NOR DO THEY LOOK AT THE 'B' BUFFER LATCHES ON THE CRM BOARD.
THE ONLY WAY YOU KNOW OF CRA BOARD ERRORS IS VIA THE 8080 GIVING YOU A STATUS BYTE., AND THERE IS NO EASY WAY OF BREAKING DOWN
THE FIELD WITHOUT SCOPING THE CHECKERS ON PAGE CRA6.

11=THIS ALLOWS THE 8080 TO LOOP THE CRAM DATA IN 24-35 BACK TO THE KS BUS FOR DIAGNOSTIC CHECKING.

THIS IS A BREAKDOWN OF THE C RAM PARITY CHECKERS, WHICH ARE
AVAILABLE VIA A DIAG READ FUNCTION 10...

```
PARITY A(BIT 30 OF FUNC 10)
        36A-37A-48-49-60-61-72-73-84-85

PARITY B(BIT 31 OF FUNC 10)
        38A-39A-50-51-62-63-74-75-86-87

PARITY C(BIT 32 OF FUNC 10)
        40A-41A-52-53-64-65-76-77-88-89

PARITY D(BIT 33 OF FUNC 10)
        42A-43A-54-55-66-67-78-79-90-91

PARITY E(BIT 34 OF FUNC 10)
        44A-45A-56-57-68-69-80-81-92-93

PARITY F(BIT 35 OF FUNC 10)
        46A-47A-58-59-70-71-82-83-94-95


BIT 29 OF FUNC 10
        CRM BOARD HAD A ERROR(END PRODUCT OF A-F)
```

CHAPTER 2

PROCESSOR

```
HALT-STATUS-WORD        STORED IN MEM LOC 0     PC STORED IN MEM LOC 1
________________

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                *----------NOT USED---------------->*<-----------------HALT STATUS CODE------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

********	24-35	HALT CODE	DEFINITION
********		0000	MICRCODE JUST STARTED
********		0001	HALT INSTRUCTION EXECUTED
********		0002	CONSOLE PROGRAM HALTED CPU
********		0100	I/O PAGE FAILURE
********		0101	ILLEGAL INTERRUPT INSTRUCTION
********		0102	POINTER TO UNIBUS VECTOR IS ZERO
********		1000	ILLEGAL MICROCODE DISPATCH
********		1005	MICROCODE STARTUP CHECK FAILED

```
MICROCODE-FLAGS                 STORED IN HALT STATUS BLOCK LOCATION X+13 IN EVENT OF PAGE FAILURE-
_______________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH              *<---NOT USED---------->!WREF !PICYC*CACHE!<------------------NOT USED------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH              *<---------------------PAGE FAIL CODE------------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

4	WREF	WRITE REFERENCE BIT FROM PAGE MAP
5	PICYC	PI CYCLE
6	CACHE	LOOK IN CACHE BIT FROM PAGE MAP
18-35	PAGE FAIL CODE-SPECIFIES THE OPERATION FOR WHICH THE PAGE FAIL CODE OCCURRED	
	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

VMA

VIRTUAL MEM ADDRESS AND FLAGS -STORED IN LOCATION X+20 OF HALT STATUS BLOCK

```
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH              *USER !     !FETCH*R CYC!W TST!W CYC*     !-CACH!PHYS *VMA P!VMA I!WRU C!VEC B!BYTE !<---PHYS ADDRESS------>*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH              *<--------VIRTUAL ADDRESS(BIT 8=0) OR PHYSICAL ADDRESS(BIT 8=1)-------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

0	USER	USER MODE
1	NOT USED	
2	FETCH	INSTRUCTION FETCH
3	R CYC	READ CYCLE
4	W TST	WRITE TEST
5	W CYC	WRITE CYCLE
6	NOT USED	
7	-CACH	DO NOT LOOK IN CACHE
8	PHYS	PHYSICAL REFERENCE
9	VMA PREVIOUS	
10	VMA I/O	
11	WRU CYCLE	
12	VECTOR BYTE	
13	BYTE	I/O BYTE INSTRUCTION
14-17	BITS	14-17 OF A PHYSICAL ADDRESS(0'S)
18-35	BITS	18-35 OF ADDRESS

(REFERENCE INDEX TO FIND MORE INFORMATION ON THE HALT STATUS BLOCK)

PC-WORD

STORED BY JSR AND OTHER INSTRUCTIONS IN MEM LOCATIONS OR AC'S

```
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH              *OVFL !    CARRY  *FLT  !FPD  !USER *USER !<--NOT USED*  TRAP     !FLT  *NO   !<----------NOT USED----------*
                *     !  0  !  1  *OVFL !     !     *IOT  !           *  2  !  1  !UFLO *DIV  !                             *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH              *<----------------------------------PROGRAM COUNTER------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

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

```
PAGE-FAIL-WORD(or-MAP-AC)       -STORED IN LOCATION 500 OF UPT ON A CPU PAGE PAIL TRAP
_________________________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH              *USER !<--PAGE FAIL CODE OR-------->*     ! PT  !PAGED<-----NOT USED--------------->!<----ADDRESS---------->*
                *ADDR !  0  !TV   *WRTN !WRTL !WREF *     !CASH !REF                                !                       *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH              *<----------VIRTUAL ADDRESS(PYSICAL ADDRESS FOR MAP IF BIT 2=1)-------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                        0       USER ADDRESS
                        2-5     MAP AC(BIT 1=0)
                        2       TRANSLATION VALID
                        3       WRITTEN (TOPS 20 ONLY )
                        4       WRITEABLE (TOPS 20 ONLY )
                        5       WRITE REFERENCE

********                2-5     PAGE FAIL CODE(BIT 1=1)
********                2       20=IO INSTRUCTION SELECTED A NON EXISTENT DEVICE OR REGISTER(BITS 14-35=THE IO ADDRESS)
********                        25=PAGE TABLE PARITY ERROR
********                        36=HARD MEMORY ERROR
********                        37=NXM
********                7       PAGE CACHE-ABLE
********                8       PAGED REFERENCE
```

****
KS10 UPT TOPS20 PAGING
****

0-420	NOT USED
421	USER ARITHMETIC OVFL TRAP INST
422	USER STACK OVF TRAP INST
423	USER TRAP 3 TRAP INST
424	FLAGS 1 MUUO OP-AC
425	MUUO OLD PC
426	E OF MUUO
427	MUUO PROCESS CONTEXT WORD
430	KERNAL NO TRAP MUUO NEW PC WORD
431	KERNAL TRAP MUUO NEW PC WORD
432	SUPERVISOR NO TRAP MUUO NEW PC WORD
433	SUPERVISOR TRAP MUUO NEW PC WORD
434	CONCEALED NO TRAP MUUO NEW PC WORD
435	CONCEALED TRAP MUUO NEW PC WORD
436-477	NOT USED
500	PAGE FAIL WORD
501	PAGE FAIL FLAGS
502	PAGE FAIL OLD PC
503	PAGE FAIL NEW PC
504-537	NOT USED
540	USER SEC 0 PTR
541-777	NOT USED

****
KS10 EPT TOPS20 PAGING
****

0-41	NOT USED
42-57	STANDARD PRIORITY INTERRUPT INST
60-77	NOT USED
100-117	VECTOR INTERRUPT TABLE POINTERS
120-420	NOT USED
421	EXEC ARITHMETIC OVF TRAP INST
422	EXEC STACK OVF TRAP INST
423	EXEC TRAP 3 TRAP INST
424-537	NOT USED
540	EXEC SEC 0 PTR
541-777	NOT USED

****
KS10 UPT TOPS10 PAGING
****

0-377L	USER PAGE 0-776
0-377R	USER PAGE 1-777
400-417L	EXEC PAGE 340-376
400-417R	EXEC PAGE 341-377
420	ADDRESS OF LUUO BLOCK
421	USER ARITHMETIC OVF
422	USER STACK OVF TRAP INST
423	USER TRAP 3 TRAP INST
424	MUUO STORED HERE
425	PC WORD OF MUUO STORED HERE
426	PROCESS CONTEXT WORD STORED HERE
427	NOT USED
430	KERNAL NO TRAP MUUO NEW PC WORD
431	KERNAL TRAP MUUO NEW PC WORD
432	SUPERVISOR NO TRAP MUUO NEW PC WORD
433	SUPERVISOR TRAP MUUO NEW PC WORD
434	CONCEALED NO TRAP MUUO NEW PC WORD
435	CONCEALED TRAP MUUO NEW PC WORD
436-477	NOT USED
500	EXEC OR USER PAGE FAIL WORD STORED HERE
501	EXEC OR USER OLD PC WORD STORED HERE
502	PAGE FAIL NEW PC WORD
503-777	NOT USED

****
KS10 EPT TOPS10 PAGING
****

0-41	NOT USED
42-57	STANDARD PRIORITY INTERRUPT INST
60-77	NOT USED
100-117	VECTOR INTERRUPT TABLE POINTERS
120-177	NOT USED
200-377L	EXEC PAGE 400-776
200-377R	EXEC PAGE 401-777
400-420	NOT USED
421	EXEC ARITHMETIC OVF TRAP INST
422	EXEC STACK OVF TRAP INST
423	EXEC TRAP 3 TRAP INST
424-577	NOT USED
600-757L	EXEC PAGE 0-336
600-757R	EXEC PAGE 1-337
760-777	NOT USED

KS10-I/O-INSTRUCTION-FORMAT

IN THE KS10 THE IO INSTRUCTION FORMAT IS THE SAME AS A TRADITIONAL PDP10 INSTRUCTION FORMAT...AS FOLLOWS

```
                 00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
               *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH             *<-------INSTRUCTION OP CODE(700-777 OCTAL)---------->*<--AC FIELD(00-17)---->!  I  !<---INDEX REGISTER---->*
               *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                 18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
               *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH             *<--------------------Y(ADDRESS)------------------------------------------------------------------------->*
               *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

               I=INDIRECT ADDRESSING BIT               INDEX REGISTER 00-17
               *Note: The AC field is used as an op-code extension for theAPR I/O instructions .
```

KS10-IO-INSTRUCTION-OP-CODES-IN-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								

KS10-APR-IO-INSTRUCTIONS-AC-FIELD-ASSIGNMENTS

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	-	-	-

APRID=700000

```
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<-------MICROCODE OPTIONS--------------------------->*<--------------------MICROCODE VERSION NUMBER------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<-HDWR OPTIONS-->*<--------------------PROCESSOR SERIAL NUMBER-------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

WRAPR(CONO-APR)=700200

```
                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
        E       *     !     !EN   *DIS  !CLR  !SET  *     !80INT!PWRF *NXM  !HERR !SERR *ITIM !KSINT!INTRQ*PI4  !PI2  !PI1  *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
        20      EN      ENABLE CONDITIONS SPECIFIED BY BITS 26-31 TO CAUSE INTERRUPTS
        21      DIS     Disable interrupts for conditions selected by bits 26-31
        22      CLR     Clear flags indicated by bits 26-31
        23      SET     SET FLAGS SPECIFIED BY 26-31
        25      80INT   INTERRUPT 8080 FROM KS
        26      PWRF    POWER FAIL
        27      NXM     NON-EXISTENT MEMORY ERROR
        28      HERR    HARD MEMORY ERROR(UNCORRECTABLE)
        29      SERR    SOFT MEM ERROR  (CORRECTABLE)
        30      ITIM    INTERVAL TIMER
        31      KSINT   INTERRUPT KS FROM 8080
        32      INTRQ   GENERATE INTERRUPT REQUEST
        33-35   PI      PRIORITY INTERRUPT ASSIGNMENT
```

RDAPR(CONI-APR)=70024 STORE THE APR STATUS IN THE WORD ADDRESSED BY E

```
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<---------------------NOT USED---------------->*PWRFE*NXME !HERRE!SERRE*ITIME!8080E!<----NOT USED--------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<--------NOT USED----------------------------->!PWRF *NXM  !HERR !SERR *TDONE!80INT!INTRQ*PI 4 !PI 2 !PI 1 *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

        BITS 8,9,10,11,12,13=SELECTED FLAGS ENABLED
        13      8080E   8080 INTERRUPT enabled
        26      PWRF    POWER FAIL
        *27     NXM     NON EXISTENT MEMORY
        *28     HERR    HARD MEM ERROR
        29      SERR    SOFT MEM ERROR(CORRECTED DATA RETURNED ON BUS)          *NOTE:  PAGE FAIL OCCURS IF ERROR IS RESULT
        30      TDONE   INTERVAL TIMER DONE                                            OF CPU MEMORY REQUEST
        31      80INT   8080 INTERRUPT                                                 NXM ALSO SETS IN UNIBUS DEVICE IF ERROR
        32      INTRQ   INTERRUPT REQUEST                                              results from UNIBUS NPR REQUEST
        33-35   PIL     PRIORITY INTERRUPT LEVEL
```

```
WRPI(CONO-PI)=700600
____________________
                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
E               *<-----NOT USED ON  KS->!DROP !CLEAR*REQ  !TURN !CHAN *TURN !SYS  !<-------SELECT CHANNEL----------------->*
                *                       !INT  !SYS  *INT  !ON   !OFF  *OFF  !ON   !  1  *  2  !  3  !  4  *  5  !  6  !  7  *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                22      DROP PROGRAM REQUESTS ON SELECTED CHANNELS
                23      CLEAR PI SYSTEM
                24      INITIATE INTERRUPTS ON SELECTED CHANNELS
                25/26    TURN SELECTED CHANNELS ON OR OFF
                27/28   TURN THE PI SYSTEM ON OR OFF
                29-35   SELECT CHANNELS FOR BITS 22,24,25,26

RDPI=700640
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<---------------------NOT USED---------------------------------->!<-----PROGRAM REQUESTS------------------>*
                *                                                                 !  1  *  2  !  3  !  4  *  5  !  6  !  7  *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<--NOT USED----->*PI IN PROGRESS(INTERRUPTS HOLDING-------->!SYS  !<---------ACTICE CHANNELS ON----------->*
                *                 *  1  !  2  !  3  *  4  !  5  !  6  *  7  !ON   !  1  *  2  !  3  !  4  *  5  !  6  !  7  *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

RDUBR=70104
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *  1  !  0  !  1  *<--NOT USED----->*CURR AC BLK      *PREV AC BLOCK    *<-------------NOT implemented------*
                *                 *                 *4    !2    !1    *  4  !  2  !  1  *                                   *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                *<-----------NOT implemented------------->!<---USER BASE REGISTER(PHYSICAL PAGE #(BITS 16-26))------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                BITS 0,2        set to a one so that word may be directly used by WRUBR
                UBR     contains the physical page number of the user process table

WRUBR=70114     LOAD THE UBR WITH THE WORD AT E
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *LOAD !  0  !LOAD *<---NOT USED---->*<--CURR AC BLK-->*<--PREV AC BLK-->*<--------NOT implemented---------->*
                *ACBLK!     !UBR  *                 *  4  !  2  !  1  *  4  !  2  !  1  *                                   *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<---NOT implemented--------------------->!<-------USER BASE REGISTER(PHYSICAL PAGE # OF UPT(BITS 16-26))-->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

```
CLRPT=70110
___________
                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
E               *<------------------VIRTUAL ADDRESS TO CLEAR IN HARDWARE PAGE TABLE---------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

        This instruction clears the hardware page table so that the next reference to word at E will cause page refill cycle.
        Note: There is one entry only for any virtual page. Clearing the map information clears both EXEC and USER mapping.

WREBR=70120
___________
                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
E               *                 *KL10 !TRAP !     *     !<----EXEC BASE REGISTER(PHYSICAL PAGE # OF EPT(BITS 16-26)------>*
                *     !     !     *PAGE !EN   !     *     !                                                                 *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

        21      KL PAGING MODE
        22      TRAP  ENABLE

RDEBR=70124     STORE THE VALUE GIVEN BY THE PREVIOUS WREBR INTO THE EFFECTIVE ADDRESS
___________
                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *     !     !     *KL10 !TRAP !     *     !<-------EXEC BASE REGISTER(PHYSICAL PAGE # OF EPT(BITS 16-26))-->*
                *     !     !     *PAGE !EN   !     *     !                                                                 *
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

WRSPB=70240             WRITE SPT(SHARED POINTER TABLE BASE REGISTER)-LOAD THE WORD AT E INTO THE SPT BASE REGISTER
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<--------------------------NOT IMPLEMENTED---------------------------------------->!<-----SPT--------------*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<--------------SPT BASE REGISTER-------------------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

RDSPB=70200             READ SPT BASE REGISTER STORE THE SPT BASE REGISTER AT E
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<------------------------NOT IMPLEMENTED------------------------------------------>!<-------SPT------------*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<------------SHARED POINTER BASE REGISTER----------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

```
WRCSB=70244                     WRITE=CORE STATUS TABLE BASE REGISTER   LOAD THE CST BASE REGISTER WITH THE WORD AT E
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<-------------------------------------NOT USED----------------------------------->!<-----CST--------------*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<---------CORE STATUS TABLE BASE REGISTER (BITS 16-35)---------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

RDCSB=70204             READ CORE STATUS TABLE BASE REGISTER
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<------------------------NOT USED------------------------------------------------>!<-------CST-------------*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<----------------CORE STATUS TABLE BASE REGISTER---------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

WRPUR=70250     WRITE PROCESS USE REGISTER
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                *<------------------------PROCESS USE REGISTER------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<--------------------------PROCESS USE REGISTER----------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                *Note: Load the process use register from E . The PUR contains the ager ( AGE register) in the left few bits.
                The bits containing the ager are cleared by anding the CST entry with the CST mask , the entire PUR is
                or'ed with the CST entry.

RDPUR=70210             STORE THE PUR AT E
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<--------------------------------PROCESS USE REGISTER----------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<---------------------------------PROCESS USE REGISTER---------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

```
WRCSTM=70254            WRITE THE CST MASK REGISTER
____________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<------------CORE STATUS TABLE MASK REGISTER-------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<---------------------------CORE STATUS TABLE MASK REGISTER----------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                Note: Load the CST mask register from E. The CST mask register should contain a 0 for every bit in
                the ager and a ONE BIT in all other bit positions.

RDCSTM=70214
____________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<-----------------------CST MASK REGISTER----------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<-------------------------CST MASK REGISTER--------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

WRTIME=70260            Write the TIME BASE
____________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E+1)         *SIGN !<------------HIGH ORDER TIME BASE------------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E+1)         *<--------------------HIGH ORDER TIME BASE----------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *SIGN !<-----------------LOW ORDER TIME BASE--------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<-----LOW ORDER TIME BASE--------->*<-----------------------------------NOT USED-------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                SIGN    SIGN BIT 0=positive 1=negative  TIME BASE in MILLESECONDS

                Note: Load the double word at E and E+1 into the TIME BASE. The TIME BASE counts up at 4.09 MHz.
```

RDTIME=70220

```
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E+1)         *SIGN !<--------------------------HIGH ORDER TIMBASE(MILLISECONDS0----------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E+1)         *<------------------HIGH ORDER TIMEBASE(MILLISECONDS)------------------------------------------------------>*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *< 0 >!<---------------------LOW ORDER TIMEBASE------------------------------------------------------------>*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<LOW ORDER TIMEBASE--------------->*<-----------TIME BASE FRACTION---------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

Note: Store the TIME BASE at E and E+1

WRINT=70264

```
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<------------INTERVAL TIMER------------------------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<--INTERVAL TIMER----------------->*>-------------------NOT USED------------------------------------------>*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

Note: Load the INTERVAL TIMER period register with the word at E (units are milliseconds).

RDINT=70224   READ THE INTERVAL REGISTER

```
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *<-----------------------------INTERVAL TIMER-------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<-----INTERVAL TIMER-------------->*<-------------------------NOT USED------------------------------------>*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

Note: Store the current value of the INTERVAL register at E. Returns the interval
loaded by WRINT , does not change with time.

```
WRHSB=70270             WRITE HALT STATUS BLOCK ADDRESS
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *SIGN !<---------------------------NOT USED--------------------------------------->!<---HSB ADDRESS------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<---------------HSB ADDRESS--------------------------------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

Note: Use the word at E as the HALT-Status block address. If the SIGN bit is negative or zero no Halt Status will be stored. If positive the internal processor status will be stored. The status consists of the 16 2901 registers , VMA , SC and FE. The programmer should allow 32 words for this block. The m-code initially sets the address to 376000. TOPS20 changes it to 400 , TOPS10 changes it to 424.

```
RDHSB=70230             READ THE HALT STATUS BLOCK ADDRESS
___________
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
LH(E)           *SIGN !<---------------NOT USED---------------------------------------------------->!<-------HSB ADDRESS--->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
RH(E)           *<------------------------------HALT STATUS BLOCK ADDRESS------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

Note: Store the current value of the HALT Status block ADDRESS at E.

```
        To access an external register , the external IO instructions generate
an extended 30 bit address , which is used as an IO address. Bits 14-17 are IO
controller numbers and bits 18-35 are used as the register address. An
extended address is required to address controllers other than zero.
i.e. the programmer must use indexing or indirection to yield the desired
effective address. For example , to write the first control status register
in the RH11 with a write function code of 61 , assuming the RH11 is on the
first controller , the programmer could either
        1/WRTIO AC,0(XR) where XR(index register)contains 1,,776700
                and AC contains 61 or
        2/WRTIO AC,@DRPCS1 where the location DRPCS1 contains 1,776700
                and AC contains 61 or
        3/WRTIO AC,776700(XR) where XR contains the CONT # in BITS 14-17
        an addressing scheme such as this could be useful for using common
        code amoung different IO controllers.
Byte instructions will transfer only 8 bits of data from the LSB's of AC.
```

Currently there are two slots allocated for IO controllers. These are now UNIBUS ADAPTER modules. The controller numbers are hardwired on the backplane.

SLOT	I/O CONTROLLER#
19	1
16	3

UBA-PAGING-RAM IO ADDRESS 763000-763077

WRIO(713)-AC,7630XX

```
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                *<-----------------------------------------------NOT USED-------------------------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                *F RPW!EN16 !F M E!VALID!<-----NOT USED->!<-------PAGE BITS---10ADR 16 THRU 26---------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

RDIO(712)-AC,7630XX

```
                  00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                *<----!-----!-----!-----!RAM P!F RPW*EN16 !F M E !VALID*R P V!-----!-----*-----!-----!-----*-----!<-----PAGE
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

                  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
                -RAM BITS 10 ADR 16-26------------------------------->*<----------NOT USED--------------------------------->*
                *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

RAM P	RAM PARITY BIT	F RPW	FORCE READ PAUSE WRITE FOR EVEN/ODD MEM WRITES
EN 16	DISABLE UPPER 2 BITS ON UNIBUS TRANSFERS	F M E	Fast Mode Enable for 36 BIT FAST MODE TRANSFERS
VALID	PAGE IS VALID	R P V	PAGING RAM PARITY valid

UBA-STATUS-REGISTER   IO ADDRESS 763100

WRIO(713)

```
  18    19    20  * 21    22    23  * 24    25    26  * 27  ! 28    29  * 30    31    32  * 33    34    35
*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
*C TIM!C BAD!C PAR*C NED!<-------------NOT USED ON WRT----->!D XFR!UINIT*<----PIH-------->*<------PIL------>*
*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

RDIO(712)

```
  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
*  TIM!  BAD!  PAR*  NED!           *INT H!INT L!PWR L*     !DXFR !     *<----PIH-------->*<---------PIL--->*
*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

18	R/W	TIM	UNIBUS ARBITRATOR TIMEOUT-NO SACK IN 5 MICRO SEC-CP ADDRESSED NON EXISTENT DEV REG OR DEV ADDRESSED NON EXISTENT MEMORY 1 CLEARS THE BIT
19	R/W	BAD	BAD MEM DATA ON NPR TRANSFER-MASTER WILL TIME OUT ON BAD MEM DATA IF 28 SET-WRT CLEARS THE BIT
20	R/W	PAR	KS10 BUS PARITY ERROR-WRITE CLEARS THE BIT
21	R/W	NED	CPU ADDRESSED NON EXISTENT DEVICE-WRT CLEARS THE BIT
24	RONLY	INT H	INTERRUPT REQUEST ON BR6 OR BR7
25	RONLY	INT L	INTERRUPT REQUEST ON BR5 OR BR4
26	RONLY	PWR L	AC LOW OR DC LOW-CLEARED ON REGISTER WRITE
28	R/W	DXFR	DISABLE TRANSFER ON UNCORRECTABLE DATA
29	WONLY	UINIT	ISSUE UNIBUS INIT
30-32	R/W	PIH	PI LEVEL OF BR6,BR7
33-35	R/W	PIL	PI LEVEL OF BR4,BR5

UBA-MAINTENANCE-REG   IO ADDRESS 763101

```
  18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
*<------------------------------------NOT USED------------------------------------------------->!S M B!C N A*
*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

WONLY	BIT 34	S M B	Spare Maintenance Bit
WONLY	BIT 35	C N A	Change NPR address

MEMORY-STATUS-REGISTER IO ADDRESS 100000

THE MEMORY WILL HOLD THE ERROR ADDRESS,THE ECC CODE AND THE TYPE OF ERROR FOR THE FIRST ERROR DETECTED.
BIT00,THE ERROR HOLD BIT, MUST BE CLEARED TO ALLOW THE MEMORY TO CONTINUE AND THE CAPTURE THE
NEXT ERROR IN THE STATUS REGISTER.

```
   00    01    02  * 03    04    05  * 06    07    08  * 09    10    11  * 12    13    14  * 15    16    17
 *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
 *HOLD !BAD D!R ERR*P ERR!ECCON! CP  * C40 ! C20 ! C10 * C04 ! C02 ! C01 *PFAIL!  0  !ADR14*ADR15!ADR16!ADR17*
 *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*

   18    19    20  * 21    22    23  * 24    25    26  * 27    28    29  * 30    31    32  * 33    34    35
 *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
 *ADR18!ADR19!ADR20*ADR21!ADR22!ADR23*ADR24!ADR25!ADR26*ADR27!ADR28!ADR29*ADR30!ADR31!ADR32*ADR33!ADR34!ADR35*
 *-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*-----+-----+-----*
```

```
00      HOLD    MMC5 ERR HOLD           SET ON ERROR CONDITION DETECTED BY CONTROLLER-CLEARED WHEN WRITTEN WITH A 1
01      BAD D   MMC4 UNCOR ERR HOLD     SET SAYS ERROR IN REG WAS UNCORRECTABLE
02      R ERR   MMC9 REF ERR            REFRESH ERR SET 5.5 USEC AFTER COM/ADR IF NO DATA RECIEVED.CLEARS ANY
                                        CYCLE IN PROGRESS AND REFRESHES THE MOS ARRAYS.CLEARED WHEN WRITTEN WITH A 1 OR MR.
03      P ERR   MMC7 PARITY ERR         MMC RECIEVED  A PARITY ERR ON THE KS10 BUS. SET
                                        BY WRITING 1 TO BIT 3;CLEARED BY WRITING 0 TO BIT 3 OR MR.
04      ECCON   MMC3 ECC ON             THE ERROR CORRECTION LOGIC IS ENABLED WHEN ON.WHEN OFF ERRORS WILL STILL BE DETECTED
                                        BUT NO CORRECTIONS MADE.TURNED ON BY MR ,turned off by IO WRITE TO BIT 35 OF THE
                                        STATUS REG
05      CP      MMC4 ERR CP             PARITY BIT OF THE CORRECTION CODE
06      C40     MMC4 ERR C40            CORR CODE BIT C40
07      C20     MMC4 ERR C20            CORRCTION CODE BIT C20
08      C10     MMC4 ERR C10            CORR CODE BIT C10
09      C04     MMC4 ERR C04            CORR CODE C04
10      C02     MMC4    ERR C02         CORR CODE BIT C02
11      C01     MMC4 ERR CO1            CORR CODE BIT C01
12      P FAIL  MMC5 POWER FAILED       SET IF MEM HAS LOST POWER OR BATTERY BACKUP FAILED.CLEARED WHEN WRITTEN WITH A 1.
13      0       UNUSED                  ALWAYS UNASSERTED
14-35   ADRXX   MMC8 ERR ADR14-35       BITS 14-35 OF THE ERROR ADDRESS REGISTER.

        IADDRESS BITS
        -------------
        14-16   SELECT CONTROLLER
        17-19   SELECT ARRAY BOARD
        20-21   SELECTS ONE OF FOUR WORDS ON THE SELECTED BOARD
        22-28   7 ROW ADDRESS LINES
        29-35   7 COLUMN ADDRESS LINES
```

I/O-INSTRUCTION-SET

TIOE(AC0)AND TIOEB(720)-TEST IO-NO MODIFICATION,SKIP IF EQUAL-FETCH WORD OR BYTE FROM WORD SPECIFIED BY E,AND WITH(AC),SKIP IF ZERO
TION(711) AND TIONB(721)-TEST IO,NO MODIFICATION,SKIP IF NOT EQUAL
RDIO(712) AND RDIOB(722)-READ IO-FETCH WORD OR BYTE SPECIFIED BY E,STORE RIGHT JUSTIFIED IN SPECIFIED AC
WRIO(713) AND WRIOB(723)-WRITE IO-STORE(AC) TO THE IO ADDRESS SPECIFIED BY E
BSIO(714) AND BSIOB(724)-BIT SET IO-FETCH(E),OR WITH WORD OR BYTE IN(AC),STORE BACK IN THE IO ADDRESS E.NO MODIFICATION TO AC.
BCIO(715) AND BCIOB(725)-BIT CLEAR IO-FETCH(E),AND WITH COMPLIMENT(AC),STORE BACK IN THE IO ADDRESS E.NO MODIFICATION TO AC.
UMOVE(704)-MOVE FROM PREVIOUS CONTEXT-EQUAL TO PXCT 4,[MOVE AC,E]
UMOVEM(705)-MOVE TO PREVIOUS CONTEXT-EQUAL TO PXCT 4,[MOVEM AC,E]

C-Field is "hidden" as (Memory status register):

MOS DATA BIT	Meaning
36	Check Bit CP
37	Check Bit C40
38	Check Bit C20
39	Check Bit C10
40	Check Bit C4
41	Check Bit C2
42	Check Bit C1

C-field		Meaning	C-field		Meaning
0	:	Unknown ECC-CODE 0	40	:	ECC Bit C40 failed
1	:	ECC-Bit C1 failed	41	:	Bit 18 failed
2	:	ECC-Bit C2 failed	42	:	Bit 19 failed
3	:	Unknown ECC-CODE 3	43	:	Bit 20 failed
4	:	ECC-BIT C4 failed	44	:	Bit 21 failed
5	:	Unknown ECC-CODE 5	45	:	Bit 22 failed
6	:	Unknown ECC-CODE 6	46	:	Bit 23 failed
7	:	Unknown ECC-CODE 7	47	:	Unknown ECC-CODE 47
10	:	ECC-Bit C10 failed	50	:	Unknown ECC-CODE 50
11	:	Bit 0 failed	51	:	Bit 24 failed
12	:	Bit 1 failed	52	:	Bit 25 failed
13	:	Bit 2 failed	53	:	Bit 26 failed
14	:	Bit 3 failed	54	:	Bit 27 failed
15	:	Bit 4 failed	55	:	Bit 28 failed
16	:	Bit 5 failed	56	:	Bit 29 failed
17	:	Unknown ECC-CODE 17	57	:	Unknown ECC-CODE 57
20	:	ECC-Bit C20 failed	60	:	Unknown ECC-CODE 60
21	:	Bit 6 failed	61	:	Bit 30 failed
22	:	Bit 7 failed	62	:	Bit 31 failed
23	:	Bit 8 failed	63	:	Bit 32 failed
24	:	Bit 9 failed	64	:	Bit 33 failed
25	:	Bit 10 failed	65	:	Bit 34 failed
26	:	Bit 11 failed	66	:	Bit 35 failed
27	:	Unknown ECC-CODE 27	67	:	Unknown ECC-CODE 67
30	:	Unknown ECC-CODE 30	70	:	Unknown ECC-CODE 70
31	:	Bit 12 failed	71	:	Unknown ECC-CODE 71
32	:	Bit 13 failed	72	:	Unknown ECC-CODE 72
33	:	Bit 14 failed	73	:	Unknown ECC-CODE 73
34	:	Bit 15 failed	74	:	Unknown ECC-CODE 74
35	:	Bit 16 failed	75	:	Unknown ECC-CODE 75
36	:	Bit 17 failed	76	:	Unknown ECC-CODE 76
37	:	Unknown ECC-CODE 37	77	:	Unknown ECC-CODE 77

UNIBUS SIGNAL NAME	BACKPANEL PIN ASSIGNMENT	M9014 CONNECTOR PIN ASSIGNMENT JACK	PIN
A00L	BH2	JI	D
A01L	BH1	J1	B
A02L	BJ2	J1	J
A03L	BJ1	J1	F
A04L	BK2	J1	N
A05L	BK1	J1	L
A06L	BL2	J1	T
A07L	BL1	J1	R
A08L	BM2	J1	X
A09L	BM1	J1	V
A10L	BN2	J1	BB
A11L	BN1	J1	Z
A12L	BP2	J1	FF
A13L	BP1	J1	DD
A14L	BR2	J1	LL
A15L	BR1	J1	JJ
A16L	BS2	J1	RR
A17L	BS1	J1	NN
ACLOL	BF1	J2	VV
BBSY L	AP2	J3	BB
BG4 H	BE2	J3	X
BG5 H	BB1	J3	T
BG6 H	BA1	J3	N
BG7 H	AV1	J3	J
BR4 L	BD2	J3	V
BR5 L	BC1	J3	R
BR6 L	AU2	J3	L
BR7 L	AT2	J3	F
C0 L	BU2	J3	NN
C1 L	BT2	J1	TT
D00 L	AC1	J2	B
D01 L	AD2	J2	F
D02 L	AD1	J2	D
D03 L	AE2	J2	L
D04 L	AE1	J2	J
D05 L	AF2	J2	R
D06 L	AF1	J2	N
D07 L	AH2	J2	V
D08 L	AH1	J2	T
D09 L	AJ2	J2	Z
D10 L	AJ1	J2	X
D11 L	AK2	J2	DD
D12 L	AK1	J2	BB
D13 L	AL2	J2	JJ
D14 L	AL1	J2	FF
D15 L	AM2	J2	NN

GND	AB2	THE FOLLOWING PINS		
GND	AC2	ON J1 J2 J3 ARE		
GND	AN1	GROUNDED		
GND	AP1	A	SS	HH
GND	AS1	E	W	MM
GND	AT1	H	Y	PP
GND	AR1	C	U	KK
GND	AV2	K	AA	SS
GND	BB2	M	CC	UU
GND	BC2	P	EE	

GND	BD1	J1	VV
GND	BE1	J2	TT
GND	BT1	J3	DD
GND	BV2	J3	RR
INIT L	AA1	J3	Z
INTR L	AB1	J3	FF
MSYN L	BV1	J3	LL
NPG H	AU1	J3	D
NPR L	AS2	J3	B
PA L	AM1	J2	LL
PB L	AN2	J2	RR
+5V	AA2	-	-
+5V	BA2	-	-
SACK L	AR2	J3	TT
DCLO L	BF2	J3	VV
SSYN L	BU1	J3	JJ

CHAPTER 3

PERIPHERALS

DEVICE	#	ADDRESS	VECTOR	UNIT	BR LEVEL	JUMPERS/SWITCHES	
RH11	1	776700	254	1	6		
RH11	2	772440	224	3	6		
LP20	1	775400	754	3	4		
LP20	2	775420	750	3	4		
DZ11	1	760010	340	3	5	A/1 ON	V/4,5,6,OFF
	2	760020	350	DITTO	DITTO	A/2 ON	V/2,4,5,6 OFF
	3	760030	360	FOR THE	REST	A/1,2 ON	V/3,4,5,6 OFF
	4	760040	370			A/3 ON	V/2,3,4,5,6 OFF
		+10 ETC	+10 ETC				
KMC11	1	760540	540			A/3,4,6 OFF	V/1,2,5 OFF
	2	760550	550			A/1,3,4,6 OFF	V/2,5 OFF
		+10 ETC	+10 ETC				

Note: Lowest acceptable CS Rev D.

Each backplane slot containing an M8204 must have wire CA1 to CB1 removed. Should the module be removed for any reason , the wires should be reinstalled to insure that the NPG signal will be passed along the UNIBUS.

DUP11	1	760300	570			A/4,5 OFF	V/2 OFF
	2	760310	600			A/1,4,5 OFF	V/1,2 ON
		+10 ETC	+10 ETC				

NOTE:LOWEST ACCEPTABLE CS REV F.STANDARD FACTORY SET JUMPERS(FOR BELL 201 COMPATABILITY)ARE AS FOLLOWS:

W1 IN	W5 OUT
W2 OUT	W6 IN
W3 IN	W7 IN
W4 IN	

```
LP20-LINE PRINTER CONTROLLER
----------------------------
M8586 CONTROL LOCATION ABCDEF02
JUMPER          STATE           FUNCTION
------          ------          ----------
W1              IN(#1),OUT(#2)  ADR BIT 4
W2              IN              ADR BIT 5
---------------------------------------
W3              IN              ADR BIT 6
W4              IN              ADR BIT 7       IF REV E M8586 WITH SWITCH PACKS INSTEAD OF JUMPERS
W5              OUT             ADR BIT 8       775400 ADDRESS=5,6,8,9,10 OFF
---------------------------------------         754 VECTOR=3 OFF
W6              OUT             ADR BIT 9
W7              IN              ADR BIT 10
W8              OUT             ADR BIT 11
-----------------------------------------
W9              OUT             ADR BIT 12


W10             IN(#1),OUT(#2)  VEC BIT 2
-----------------------------------------
W11             IN              VEC BIT 3
W12             OUT             VEC BIT 4
W13             IN              VEC BIT 5
-----------------------------------------
W14             IN              VEC BIT 6
W15             IN              VEC BIT 7
W16             IN              VEC BIT 8
------------------------------------------
W1-9 CORRESPOND TO BASE ADR 775400(UNIT #1) AND 775420(UNIT #2)
W10-W16 CORRESPOND TO VECTOR 754(#1) AND 750 (#2)
DIP SITE E6 CONTAINS PRIORITY PLUG BR4 54-08776


M8587 DATA PATHS
----------------
W1      OUT     INSTALL TO ENABLE PARITY
W2      IN      INSTALL FOR DAVFU


CABLE
-----
CABLE IS BC06R.CONNECT FROM BERG CONNECTOR (J1) ON THE M8587 MODULE,RIB SIDE
TOWARDS MODULE(RED LINE AWAY FROM HANDLE),TO CENTER SLOT OF RECEPTICLE HOUSING
MOUNTED IN THE CONNECTOR TRAY(RED LINE TOWARDS THE CONNECTOR FASTENER.
```

CTY--BOTTOM BERG CONNECTOR--SWITCH PACK E508 FOR BAUD RATE SELECTION
KLINIK-TOP BERG CONNECTOR--SWITCH PACK E509 FOR BAUD RATE SELECTION

BAUD	SWITCHES					
RATE	1	2	3	4	5*	6
110	0	0	0	0	0	0
300	0	0	1	0	1	0
600	1	0	0	0	1	0
1200	0	1	0	0	1	0
1800	0	1	0	1	1	0
2400	0	0	1	1	1	0
4800	0	1	1	0	1	0
9600	0	1	1	1	1	0

*NOTE:SWITCH 5=STOP BIT SELECTION 0=2 STOP BITS 1=1 STOP BITS

REGISTER DESIGNATIONS

AE	ADDRESS EXTENSION
AS	ATTENTION SUMMARY
BA	BUFFER ADDRESS
BAE	BUFFER ADDRESS EXTENSION
CK	CHECK CHARACTER
CS	CONTROL
DA	DESIRED SECTOR/TRACK ADDRESS
DB	DATA BUFFER
DC	DESIRED CYLINDER
DS	STATUS
DT	DRIVE TYPE
EC	ERROR CORRECTION
ER	ERROR REGISTER
FC	FRAME COUNT
HR	HOLDING REGISTER
LA	LOOK AHEAD
MR	MAINTENANCE REGISTER
OF	OFFSET
SN	SERIAL NUMBER
TC	TAPE CONTROL
WC	WORD COUNT

The register designations are preceded by MT,RM, or RP, depending on the register being looked at.

COMMAND CODE(OCTAL)	FIXED HEAD DSK	MOVING HEAD DSK	MAGNETIC TAPE
01	NOOP	NOOP	NOOP
03		UNLOAD	REWIND,OFF-LINE
05		SEEK	
07		RECALIBRATE	REWIND
11	DRIVE CLEAR	DRIVE CLEAR	DRIVE CLEAR
13		RELEASE	
15		OFFSET	
17		RETURN TO CENTERLINE	
21	READ IN PRESET	READIN PRESET	READIN PRESET
23		PACK ACKNOWLEDGE	
25			ERASE
27			WRITE FILE MARK
31	SEARCH	SEARCH	SPACE FORWARD
33			BACKSPACE
51	WRITE CHECK DATA	WRITE CHECK DATA	WRITE CHECK FORWARD
53		WRITE CHECK HEADER/DATA	
57			WRITE CHECK REVERSE
61	WRITE DATA	WRITE DATA	WRITE FORWARD
63		WRITE HEADER AND DATA	
71	READ DATA	READ DATA	READ FORWARD
73		READ HEADER AND DAT	
77			READ REVERSE

```
MBUS!DCL   !RH11 !     *      !    !     *     !     !     *     !     !     *     !     !     *      !    !     *UNIBUS!
ADR !PRINT !NAME !     *      !    !     *     !     !     *     !     !     *     !     !     *     !     !     *ADR   !NOTE
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
10  !      !     !20   *21   !22   !23   *24   !25   !26   *27   !28   !29   *30   !31   !32   *33   !34   !35   *      !
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
MBUS!      !     !  C15*  C14!  C13!  C12*  C11!  C10!  C09*  C08!  C07!  C06*  C05!  C04!  C03*  C02!  C01!  C00*      !
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
    !      !     !     *     !     !     *     !     !     *     !     !     *     !     !     *     !     !     *      !
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
00  !RG3RG6!RPCS1!SC   *TRE  !MCPE !0    *DVA  !PSEL !A17  *A16  !RDY  !IE   *F4   !F3   !F2   *F1   !F0   !GO   *776700!1W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !      !RPWC !WC15 *WC14 !WC13 !WC12 *WC11 !WC10 !WC09 *WC08 !WC07 !WC06 *WC05 !WC04 !WC03 *WC02 !WC01 !WC00 *776702!
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !      !RPBA !BA15 *BA14 !BA13 !BA12 *BA11 !BA10 !BA09 *BA08 !BA07 !BA06 *BA05 !BA04 !BA03 *BA02 !BA01 !BA00 *776704!
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
05  !SS3SS6!RPDA !0    *0    !0    !TA16 *TA08 !TA04 !TA02 *TA01 !0    !0    *0    !SA16 !SA08 *SA04 !SA02 !SA01 *776706!W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !      !RPCS2!DLT  *WCE  !UPE  !NED  *NEM  !PGE  !MXF  *MDPE !OR   !IR   *CLR  !PAT  !BAI  *U2   !U1   !U0   *776710!
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
01* !RG6   !RPDS !ATA  *ERR  !PIP  !MOL  *WRL  !LST  !PGM  *DPR  !DRY  !VV   *DE1  !DL64 !GRV  *DIGB !DF20 !DF5  *776712!
    !      !     !     *     !     !     *     !     !     *     !     !     *0    !0    !0    *0    !0    !0    *      !
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
02  !RG0   !RPER1!DCK  *UNS  !OPI  !DTE  *WLE  !IAE  !AOE  *HCRC !HCE  !ECH  *WCF  !FER  !PAR  *RMR  !ILR  !ILF  *776714!W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
04  !DP    !RPAS !0    *0    !0    !0    *0    !0    !0     *0    !ATA7 !ATA6 *ATA5 !ATA4 !ATA3 *ATA2 !ATA1 !ATA0 *776716!W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
07  !DP6   !RPLA !0    *0    !0    !0    *0    !SC4  !SC3  *SC2  !SC1  !SC0  *EXT1 !EXT0 !0    *0    !0    !0    *776720!
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !      !RPDB !DB15 *DB14 !DB13 !DB12 *DB11 !DB10 !DB09 *DB08 !DB07 !DB06 *DB05 !DB04 !DB03 *DB02 !DB01 !DB00 *776722!
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
03  !EC1RG3!RPMR !0    *0    !0    !0    *0    !HCT  !SBD  *ZDT  !DEN  !ECCE *MWR  !MRD  !MSCLK*MIND !MCLK !DMD  *776724!W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
06  !EC8   !RPDT !NBA  *TAP  !MOH  !0    *DRQ  !0    !0    *DT8  !DT7  !DT6  *DT5  !DT4  !DT3  *DT2  !DT1  !DT0  *776726!
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
10  !EC8   !RPSN !SN38 *SN34 !SN32 !SN31 *SN28 !SN24 !SN22 *SN21 !SN18 !SN14 *SN12 !SN11 !SN08 *SN04 !SN02 !SN01 *776730!
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
11  !RG1   !RPOF !SGCH *0    !0    !FMT22*ECCI !HCI  !0    *0    !OFS7 !OFS6 *OFS5 !OFS4 !OFS3 *OFS2 !OFS1 !OFS0 *776732!W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
12* !SS1   !RPDC !0    *0    !0    !0    *0    !0    !0    *DC09 !DC08 !DC07 *DC06 !DC05 !DC04 *DC03 !DC02 !DC01 *776734!W
    !      !     !     *     !     !     *     !     !DC10 *     !     !     *     !     !     *     !     !     *
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
13* !SS1   !RPCC !0    *0    !0    !0    *0    !0    !0    *CC09 !CC08 !CC07 *CC06 !CC05 !CC04 *CC03 !CC02 !CC01 *776736!
    !      !     !     *     !     !     *     !     !CC10 *     !     !     *     !     !     *     !     !     *      !
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
14* !EC6   !RPER2!ACUNS*0    !PLU  !30VU *IXE  !NHS  !MHS  *WRU  !FEN  !TUF  *TDF  !MSE  !CSU  *WSU  !CSF  !WCU  *776740!W
    !      !     !NC   *     !     !0    *     !     !     *     !ABS  !     *     !RAW  !     *     !     !     *      !
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
15* !EC7   !RPER3!OCYL *SKI  !0    !0    *0    !0    !0    *0    !0    !ACL  *DCL  !DISE !UWR  *0    !VUF  !PSU  *776742!W
    !      !     !     *     !OPE  !     *     !     !     *     !     !     *     !35VF !0    *     !WAO  !DCU  *      !
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
16  !EC2   !RPEC1!0    *0    !0    !4096 *2048 !1024 !512  *256  !128  !64   *32   !16   !8    *4    !2    !1    *776744!
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
17  !EC1   !RPEC2!0    *0    !0    !0    *0    !BIT10!BIT9 *BIT8 !BIT7 !BIT6 *BIT5 !BIT4 !BIT3 *BIT2 !BIT1 !BIT0 *776746!
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
1.SHARED REGISTER  C15-C13,C10-C06 IN RH11  *WHERE TWO ENTRIES  TOP=RP04  BOTTOM=RP06 W=READ/WRITE --=RH11 REGISTER     REST STORED IN DRIVE
```

```
MBUS*RH11  ! RM03!     *     !     !     *     !     !     *     !     !     *     !     !     *     !     !     *UNIBUS!
ADR !NAME  !BOARD!     *     !     !     *     !     !     *     !     !     *     !     !     *     !     !     *ADR   !NOTE
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
10  !      !     !20   *21   !22   !23   *24   !25   !26   *27   !28   !29   *30   !31   !32   *33   !34   !35   *      !
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
MBUS!      !     !  C15*  C14!  C13!  C12*  C11!  C10!  C09*  C08!  C07!  C06*  C05!  C04!  C03*  C02!  C01!  C00*      !
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
00  !RMCS1 !IF   !SC   *TRE  !MCPE !0    *DVA  !PSEL !A17  *A16  !RDY  !IE   *F4   !F3   !F2   *F1   !F0   !GO   *776700!1,R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !RMWC  !     !WC15 *WC14 !WC13 !WC12 *WC11 !WC10 !WC09 *WC08 !WC07 !WC06 *WC05 !WC04 !WC03 *WC02 !WC01 !WC00 *776702!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !RMBA  !     !BA15 *BA14 !BA13 !BA12 *BA11 !BA10 !BA09 *BA08 !BA07 !BA06 *BA05 !BA04 !BA03 *BA02 !BA01 !BA00 *776704!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
05  !RMDA  ! DS  !0    *0    !0    !0    *0    !TA04 !TA02 *TA01 !0    !0    *0    !SA16 !SA08 *SA04 !SA02 !SA01 *776706!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !RMCS2 !     !DLT  *WCE  !UPE  !NED  *NEM  !PGE  !MXF  *MDPE !OR   !IR   *CLR  !PAT  !BAI  *U2   !U1   !U0   *776710!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
01  !RMDS  !IF   !ATA  *ERR  !PIP  !MOL  *WRL  !LBT  !PGM  *DPR  !DRY  !VV   *0    !0    !0    *0    !0    !OM   *776712!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
02  !RMER1 !IF   !DCK  *UNS  !OPI  !DTE  *WLE  !IAE  !AOE  *HCRC !HCE  !ECH  *WCF  !FER  !PAR  *RMR  !ILR  !ILF  *776714!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
04  !RMAS  !IF   !0    *0    !0    !0    *0    !0    !0     *0   !ATA7 !ATA6 *ATA5 !ATA4 !ATA3 *ATA2 !ATA1 !ATA0 *776716!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
07  !RMLA  !CS   !0    *0    !0    !0    *0    !SC16 !SC8  *SC4  !SC2  !SC1  *0    !0    !0    *0    !0    !0    *776720!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !RMDB  !     !DB15 *DB14 !DB13 !DB12 *DB11 !DB10 !DB09 *DB08 !DB07 !DB06 *DB05 !DB04 !DB03 *DB02 !DB01 !DB00 *776722!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
03  !RMMR1 !CS   !OCC  *R&G  !EBL  !REXC *ESRC !PLFS !ECRC *PDA  !PHA  !CONT *PS   !EECC !WD   *LS   !LS&T !DMD  *776724!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
    !      !     !DBCLK*DBEN !DEBL !MSTOP*MCLK !MRD  !MURDY*MOCYL!MSERR!MOF  *MSCLK!NC   !MWP  *MIND !MSC  !DMD  *776724!W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
06  !RMDT  !IF   !0    *0    !MOH  !0    *DRQ  !0    !0    *DT8  !DT7  !DT6  *DT5  !DT4  !DT3  *DT2  !DT1  !DT0  *776726!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
10  !RMSN  !CS   !S8000*S4000!S2000!S1000*S800 !S400 !S200 *S100 !S80  !S40  *S20 !S10  !SN08 *SN04 !SN02 !SN01 *776730!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
11  !RMOF  !IF   !0    *0    !0    !FMT16*ECI  !HCI  !0    *0    !OFF D!0    *0    !0    !0    *0    !0    !0    *776732!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
12  !RMDC  !DS   !0    *0    !0    !0    *0    !0    !0    *DC512!DC256!DC128*DC64 !DC16 !DC08 *DC04 !DC02 !DC01 *776734!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
13  !RMHR  !IF   !0    *0    !0    !0    *0    !0    !0    *   0 !   0 !   0 *   0 !   0 !   0 *   0 !   0 !   0 *776736!UNUSED
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
14  !RMMR2 !CS   !REQA *REQB !TAG  !TEST *CCTAG!CHTAG!BB9  *BB8  !BB7  !BB6  *BB5  !BB4  !BB3  *BB2  !BB1  !BB0  *776740!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
15  !RMER2 !IF   !0    *SKI  !OPE  !IVC  *LSC  !LBC  !0    *0    !DEVCK!0    *0    !0    !DPE  *0    !0    !0    *776742!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
16  !RMEC1 !DS   !0    *0    !0    !P4096*P2048!P1024!P512 *P256 !P128 !P64  *P32  !P16  !P8   *P4   !P2   !P1   *776744!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
17  !RMEC2 !DS   !0    *0    !0    !0    *0    !PAT11!PAT10*PAT9 !PAT8 !PAT7 *PAT6 !PAT5 !PAT4 *PAT3 !PAT2 !PAT1 *776746!R
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !RMBAE !     !0    *0    !0    !0    *0    !0    !0    *0    !0    !0    *A21  !A20  !A19  *A18  !A17  !A16  *776750!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
--  !RMCS3 !     !APE  *DPEHI!DPELO!WCEHI*WCELO!DBL  !0    *0    !0    !IE   *0    !0    !IPCK3*IPCK2!IPCK1!IPCK0*776752!R/W
-----------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------------
SHARED REGISTER  C15-C13,C10-C06 IN RH11 REST STORED IN DRIVE  *   --=RH11 REGISTER             DS--DATA SEQUENCER
(M7685)  IF--CONTROL INTERFACE (M7686)    CS--COMMAND SEQUENCER (M7684)
```

RM03/RP06/RH11-REGISTER-BITS-IN-PLAIN-ENGLISH REFER TO MANUALS.

30VU	30 VOLTS UNSAFE	F	FUNCTION	PAR	PARITY ERROR
35VF	35 VOLT FAILURE	FEN	FAILSAFE ENABLED	PAT	PARITY TEST
A	UNIBUS ADDRESS EXTENTION BITS	FER	FORMAT ERROR	PATXX	PATTERN
ABS	ABNORMAL STOP	FMT16	16 BITS/WORD FORMAT(same as FMT22)	PDA	PACK DATA AREA
ACL	AC LOW	FMT22	FORMAT 22 SECTORS	PDA	PACK DATA AREA
ACU	AC UNSAFE	GO	PERFORM OPERATION	PGE	PROGRAM ERROR
AOE	ADDRESS OVERFLOW ERROR	GRV	GO REVERSE	PGE	PROGRAM ERROR
APE	ADDRESS PARITY ERROR	HCE	HEADER COMPARE ERROR	PGM	PROGRAMMABLE
ATA	ATTENTION ACTIVE	HCI	HEADER COMPARE INHIBIT	PHA	PACK HEADER AREA
BA	BUS ADDRESS	HCRC	HEADER CRC ERROR	PIP	POSITIONING IN PROGRESS
BAI	UNIBUS ADDR. INCR. INHIBIT	IAE	INVALID ADDRESS ERROR	PLFS	PACK LOOKING FOR SYNC
BBX	BUS IN LINES-dep. on TAG LINES	IE	INTERRUPT ENABLE	PLU	PLO UNSAFE
BIT	BURST	ILF	ILLEGAL FUNCTION	PS	PROM STROBE
BLC	BURST LOCATION COUNT	ILR	ILLEGAL REGISTER	PSEL	PORT SELECT
CC	CURRENT CYLINDER	IPCKX	INVERTED PARITY CHECK BITS	PSU	PACK SPEED UNSAFE
CCTAG	CONTR. SEL. OR CYL. SELECT TAG	IR	INPUT READY	PXXXXX	ECC POSITION REGISTER BITS
CHTAG	CONTR. SELECT or HEAD SEL. TAG	IVC	INVALID COMMAND	R&G	RUN AND GO
CLR	CONTROLLER CLEAR	IXE	INDEX ERROR	RAW	READ AND WRITE
CONT	CONTINUE	LBC	LOSS OF BIT CHECK	RDY	READY
CSF	CURRENT SINK FAILURE	LBT	LAST BLOCK TRANSFERRED	REQA	REQUEST ON PORTA
CSU	CURRENT SWITCH UNSAFE	LS	LAST SECTOR	REQB	REQUEST ON PORTB
DB	DATA BUFFER	LS&T	LAST SECTOR AND TRACK	REXC	RECIEVED EXCEPTION
DBCLK	DEBUG CLOCK	LSC	LOSS OF SYSTEM CLOCK	RMR	REGISTER MODIFICATION REFUSED
DBEN	DEBUG CLOCK ENABLE	LST	LAST SECTOR TRANSFERRED	SA	SECTOR ADDRESS
DBL	DOUBLE WORD OPERATION	MCLK	MAINTENANCE CLOCK	SBD	SYNC BYTE DETECTED
DC	DESIRED CYLINDER	MCPE	MASSBUS CONTROL BUS PARITY ERROR	SC	SPECIAL CONDITION
DCK	DATA CHECK	MDF	MAINT DRIVE FAULT	SCG	SIGN CHANGE
DCL	DC LOW	MDPE	MASSBUS DATA BUS PARITY ERROR	SKI	SEEK INCOMPLETE
DCU	DC UNSAFE	MHS	MULTIPLE HEAD SELECT	SN	SERIAL NUMBER
DE1	DIFFERENCE EQUALS ONE	MIND	MAINTENANCE INDEX	SXXX	SERIAL NUMBER
DEBL	DIAGNOSTIC EBL	MOCYL	MAINT ON CYL	TA	TRACK ADDRESS
DEN	DATA ENVELOPE	MOH	MOVING HEAD	TAG	CONTROL SELECT TAG
DEVCK	DC OR HEAD SEL. FAULT	MOL	MEDIUM ON LINE	TAP	TAPE DRIVE
DF20	DRIVE FORWARD 20 INCHES PER SEC	MRD	MAINTENANCE READ	TDF	TRANSITIONS DETECTOR FAILURE
DF5	DRIVE FORWARD 5 INCHES PER SEC	MS	MAINT SECTOR PULSE	TEST	COMMAND SEQUENCER IS BRANCHING
DIGB	DRIVE TO INNER GUARD BAND	MSC	MAINTENANCE SECTOR COMPARE	TRE	TRANSFER ERROR
DL64	DIFFERENCE LESS THAN 64	MSCLK	MAINTENANCE SECTOR CLK	TUF	TRANSITIONS UNSAFE
DLT	DATA LATE	MSE	MOTOR SEQUENCE ERROR	U	UNIT SELECT
DMD	DIAGNOSTIC MODE	MSERR	MAINT SEEK ERR	UNS	UNSAFE
DPE	DATA PARITY ERROR RECIEVED	MSTOD	MAINT SEARCH TIME OUT DISABLE	UPE	UNIBUS PARITY ERROR
DPEHI	DATA PARITY ERROR-ODD WORD	MURDY	MAINT UNIT READY	UWR	ANY UNSAFE EXCEPT READ/WRITE
DPELO	DATA PARITY ERROR EVEN WORD	MWP	MAINT WRITE PROTECT	VUF	VELOCITY UNSAFE
DPR	DRIVE PRESENT	MWR	MAINTENANCE WRITE	VV	VALID VOLUME
DRQ	DRIVE REQUEST REQUIRED	MXF	MISSED TRANSFER	WAO	WRITE AND OFFSET
DRY	DRIVE READY	NBA	NOT BLOCK ADDRESSED	WC	WORD COUNT
DT	DRIVE TYPE	NC	NOT CONNECTED	WCE	WRITE CHECK ERROR
DTE	DRIVE TIMING ERROR	NED	NONEXISTENT MEMORY	WCEHI	WRITE CHECK ERROR-ODD WORD
DVA	DRIVE AVAILABLE	NEM	NON EXISTENT MEMORY	WCELO	WRITE CHECK ERROR-EVEN WORD
EBL	END OF BLOCK LEVEL	NHS	NO HEAD SELECTION	WCF	WRITE CLOCK FAIL
ECCE	ECC ENVELOPE	OCYL	OFF CYLINDER	WCU	WRITE CURRENT UNSAFE
ECH	ECC HARD ERROR	OES	OFFSET INFORMATION	WD	WRITE DATA
ECI	ECC INHIBIT	OFF D	OFFSET DIRECTION	WLE	WRITE LOCK ERROR
ECRC	ENABLE CRC OUT	OM	OFFSET MODE	WRL	WRITE LOCK
EECC	ENABLE ECC OUT	OPE	OPERATOR PLUG ERROR	WRU	WRIRE READY UNSAFE
ERR	ERROR	OPI	OPERATION INCOMPLETE	WSU	WRITE SELECT UNSAFE
ESRC	ENABLE SEARCH	OR	OUTPUT READY	ZDT	ZERO DETECT
EXT	EXTENTION				

```
MBUS!     !      !     !     *     !     !     *     !     !     *     !     !     *     !     !     *     !     !     *      !
ADR !NAME !      !     !     *     !     !     *     !     !     *     !     !     *     !     !     *     !     !     *      !NOTE
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
10  !     !      !     !20   *21   !22   !23   *24   !25   !26   *27   !28   !29   *30   !31   !32   *33   !34   !35   *      !
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
MBUS!     !      !     !  C15*  C14!  C13!  C12*  C11!  C10!  C09*  C08!  C07!  C06*  C05!  C04!  C03*  C02!  C01!  C00*      !
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
    !     !      !     !     *     !     !     *     !     !     *     !     !     *     !     !     *     !     !     *      !
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
00  !MTCS1!772440!     !SC   *TRE  !MCPE !0    *DVA! *PSEL !A17  *A16  !RDY  !IE   *F5   !F4   !F3   *F2   !F1   !F0/GO*1**   !R/W
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
--  !MTWC !772442!     !WC15 *WC14 !WC13 !WC12 *WC11 !WC10 !WC09 *WC08 !WC07 !WC06 *WC05 !WC04 !WC03 *WC02 !WC 01!WC00 *      !R/W
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
--  !MTBA !772444!     !BA15 *BA14 !BA13 !BA12 *BA11 !BA10 !BA09 *BA08 !BA07 !BA06 *BA05 !BA04 !BA03 *BA02 !BA01 !BA00 *      !R/W
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
05  !MTFC !772446!     !FC15 *FC14 !FC13 !FC12 *FC11 !FC10 !FC09 *FC08 !FC07 !FC06 *FC05 !FC04 !FC03 *FC02 !FC01 !FC00 *      !R/W
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
--  !MTCS2!772450!     !DLT  *WCE  !UPE  !NED  *NEM  !PGE  !MXF  *MDPE !OR   !IR   *CLR  !PAT  !BAI  *U2   !U1   !U0   *      !
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
01  !MTDS !772452!     !ATA  *ERR  !PIP  !MOL  *WRL  !EOT  !NC   *DPR  !DRY  !SSC  *PES  !SDWN !IDB  *TM   !BOT  !SLA  *      !R
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
02  !MTER !772454!     !COR/ *UNS  !OPI  !DTE  *NEF  !CS/  !FCE  *NSG  !PEF/ !INC/ *DPAR !FMT  !CPAR *RMR  !ILR  !ILF  *      !R
    !     !      !     !CRC  *     !     !     *     !ITM  !     *     !LRC  !VPE  *     !     !     *     !     !     *      !
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*------------
04  !MTAS !772456!     !NC   *NC   !NC   !NC   *NC   !NC   !NC   *NC   !ATA7 !ATA6 *ATA5 !ATA4 !ATA3 *ATA2 !ATA1 !ATA0 *      !R/W
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*-----------
07  !MTCK !772460!     !NC   *NC   !NC   !NC   *NC   !NC   !NC   *CRCP !CRC7 !CRC6 *CRC5 !CRC4 !CRC3 *CRC2 !CRC1 !CRC0 *NRZ   !R
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*-----------
07  !MTCK !772460!     !NC   *NC   !NC   !NC   *NC   !NC   !NC   *DT4 P!DT7  !DT6  *DT5  !DT3  !DT9  *DT1  !DT8  !DT2  * PE   !R
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*-----------
--  !MTDB !772462!     !DB15 *DB14 !DB13 !DB12 *DB11 !DB10 !DB09 *DB08 !DB07 !DB06 *DB05 !DB04 !DB03 *DB02 !DB01 !DB00 *      !R/W
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*-----------
03  !MTMR !772464!     !MDF8 *MDF7 !MDF6 !MDF5 *MDF4 !MDF3 !MDF2 *MDF1 !MDF0 !SWC  *MC   !MOP3 !MOP2 *MOP1 !MOP0 !MM   *      !R/W
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*-----------
06  !MTDT !772466!     !NSA  *TAP  !MOH  !7 CH *DRQ  !SPR  !NC   *<---------------DRIVE TYPE 00-08----------------------------!R
-----------------------------------------------------------------------------------------------------------------------------------
10  !MTSN *772470!     !SN15 *SN14 !SN13 !SN12 *SN11 !SN10 !SN09 *SN08 !SN07 !SN06 *SN05 !SN04 !SN03 *SN02 !SN01 !SN00 *      !R
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------*-----------
11  !MTTC !772472!     !ACCL *FCS  !TCW/ !EAO  *NC   !DEN  !DEN  *DEN  !FMT  !FMT  *FMT  !FMT  !EV   *SS2  !SS1  !SS0  *      !R/W
    !     !      -     !     *     -  SAC !DTE  *     !2    !1    *0    !SEL3 !SEL2 *SEL1 !SEL0 !PAR  *     -     !     *      !
-----------------------------*-----------------*-----------------*-----------------*-----------------*-----------------------------
```

```
*HARDWIRED SET  **1=SHARED REGISTER BITS 15-13,10-6 IN RH11,REST IN DRIVE
--=RH11 REGISTER
TCW- TMO2       SAC- TMO3
```

NAME	BIT#	ORIGIN	DESCRIPTION
DBXX	15-00		DATA BUFFER-R/W-READ OBUF ON READ-LOAD IBUF ON WRITE AND SEQUENCE SILO
WCXX	15-00		WORD COUNT REGISTER BITS-R/W-IN RH-INCREMENTED DURING DATA TRANSFERS
BAXX	15-00		UNIBUS ADDRESS REGISTER BITS-R/W-STARTING MEM ADDRESS OF TRANSFER-INCREMENTED BY 2 AFTER EACH XFER
DLT	15		DATA LATE-DATA OVERRUN-READ ONLY
WCE	14		WRITE CHECK ERROR-READ ONLY-TAPE WORD DOES NOT MATCH MEM WORD-MEM ADR IS MTBA-1-DATA IN MTDB
PE	13		PARITY ERROR-R/W-DISABLED FOR 18 BIT TRANSFERS
NED	12		NON EXISTENT DRIVE-READ ONLY-DRIVE FAILS TO ASSERT TRA IN 1.5 USEC AFTER DEM
NEM	11		NON-EXISTENT MEMORY-READ ONLY-MEM DID NOT RESPOND TO MSYNC IN 10 US-ADR=MTBA+2
PGE	10		PROGRAM ERROR-READ ONLY-PROGARM TRIED A DATA TRANSFER WHILE ONE WAS IN PROGRESS
MXF	09		MISSED TRANSFER-R/W-DRIVE DID NOT RESPOND TO DATA TRANSFER WITHIN 250 MSEC
MDPE	08		MASSBUS DATA BUS PARITY ERROR-READ ONLY-IF IT WAS A WRITE DRIVE WILL DETECT AND SET PAR ALSO
OR	07		OUTPUT READY -READ ONLY-WORD PRESENT AND READY IN MTDB
IR	06		INPUT READY-READ ONLY-WORD MAY BE WRITTEN INTO MTDB
CLR	05		CONTROLLER CLEAR-WRITE ONLY-INIT CONTROLLER AND DRIVES
PAT	04		PARITY TEST-R/W-GENERATE EVEN PARITY ON CONTROL AND DATA BUS
BAI	03		BUS ADDRESS INCREMENT INHIBIT-R/W-DO NOT INCREMENT MTBA ON DATA TRANSFERS
U2-0	2-0		UNIT SELECT-TM02/03 SELECT-R/W
SC	15		SPECIAL CONDITION-READ ONLY-SET BY TRE,ATTN OR CONTROL BUS PARITY ERR-CLEARED BY INIT
TRE	14		TRANSFER ERROR-SET BY DLT,WCE,UPE,NED,NEM, PGE,MXF,MDPE OR DRIVE ERROR-CLEARED BY INIT OR GO
MCPE	13		MASSBUS CONTROL BUS PARITY ERROR-READ ONLY-CLEARED BY INIT,CONTROLLER CLEAR OR GO BIT
PSEL	10		PORT SELECT-R/W-0=UNIBUS A TRANSFER 1=UNIBUS B TRANSFER
A17	9		MTBA EXTENSION BIT-R/W
A16	8		MTBA EXTENTION BIT-R/W
RDY	7		READY-NORMALLY 1-ZERO DURING DATA TRANSFERS
IE	6		INTERRUPT ENABLE-R/W
F5-F0	5-0		MASSBUS FUNCTION BITS AND GO BIT
7CH	12		7 CHANNEL UNIT -NEGATED ON 9CH DRIVE OR POWER LOSS
ACCL	15		ACCELERATION-SET WHEN TRANSPORT NOT ACTIVELY READING OR WRITING
ATA	15	M	ATTENTION ACTIVE-ATTN L INTERFACE SIGNAL ASSERTED
ATA0-7			ATTENTION SUMMARY PSEUDO REGISTER(1 BIT PER TM02/03) BITS
BOT	01	SS	BEGINNING OF TAPE
COR/CRC	15		COORECTABLE DATA ERROR(PE)OR CRCC READ DOES NOT MATCH COMPUTED CRCC(NRZI)
CPAR	03		CONTROL BUS PARITY-INCORRECT PARITY DETECTED
CRC0-7			CRCC BITS FOR LAST TRANSFER IN NRZI MODE
CS/ITM	10		CORRECTABLE SKEW(PE) OR ILLEGAL TAPE MARK(NRZI)
DEN0-2	08-10		SPECIFIES TAPE CHARACTER DENSITY
DPAR	05		DATA BUS PARITY ERROR
DPR	08	M	DRIVE PRESENT-HARDWIRED SET
DRQ	11		DRIVE REQUEST REQUIRED-NEGATED TO INDICATE SINGLE PORT UNIT
DRY	07	M	DRIVE READY-TM02/03 AND SELECTED TRANSPORT READY FOR COMMAND
DT0-7			DEAD TRACKS-TRACK MAY HAVE DROPPED ONE OR MORE BITS DURING THE TRANSFER
DT00-08	00-08		DRIVE TYPE 011=TU16/TM02 012=TU45/TM02 052=TU45/TM03 054=TU77/TM03
DTE	12		DRIVE TIMING ERROR-CLASS B
DVA	11		DRIVE AVAILABLE HARDWIRED SET
EAODTE	12		ENABLE ABORT ON DATA TRANSFER OPERATIONS-ENABLES ABORT ON ERROR REG BITS15,7,6,5
EOT	10	SS	END OF TAPE
ERR	14	M	COMPOSITE ERROR-ANY BIT IN ERROR REG IS SET
EVPAR	03		EVEN PARITY -EVEN PARITY READ/WRITTEN DURING NRZI OPERATION-IGNORED IN PE MODE
FC00-15			FRAME COUNT REGISTER BITS-COUNTS TAPE EVENTS
FCE	09		FRAME COUNT ERROR-CLASS A
FCS	13		FRAME COUNT STATUS-SET AT END OF WRITE TO FRAME COUNT REG-RESET ON FC REG OVERFLOW
FMT	04		FORMAT-DATA TRANSFER WITH INCORRECT FORMAT CODE DETECTED
FMT SEL0-3			FORMAT SELECT CODE--SEE BELOW

FX	0-5		FUNCTION CODE BITS
IDB	03	M	IDENTIFICATION BURST(PE) DETECTED-ASSERTED TILL NEXT TAPE MOTION COMMAND
ILF	00		ILLEGAL FUNCTION-CLASS B
ILR	01		ILLEGAL REGISTER-CLASS A
INC/VPE	06		INCOORECTABLE DATA ERROR OR VERTICAL PARITY ERROR-VPE ON NRZI READ OR DEAD TRACK ERRORS AND /OR SKEW OVERFLOW ON PE READ
MC	05		MAINTENANCE CLOCK
MDF0-8	07-15		MAINTENANCE DATA FIELD-BUFFERS DATA ON WRAP OPERATION-LRC AT END OF NRZI TRANSFERS
MM	OO		MAINTENANCE MODE
MOH	13		NEGATED TO INDICATE NON -MOVING HEAD UNIT
MOL	12	SS	MEDIUM ON LINE-TAPE LOADED AND ON-LINE
MOP0-3	01-04		MAINT OPERATION CODE
NC			NOT CONNECTED-UNUSED
NEF	11		NONEXECUTABLE FUNCTION-CLASS B
NSA	15		NOT SECTOR ADDRESSED-ALWAYS ASSERTED
NSG	08		NONSTANDARD GAP TAPE CHAR DETECTED DURING FIRST HALF OF EOR
OPI	13		OPERATION INCOMPLETE-CLASS B
PEF/LRC	07		PE FORMAT ERROR OR NRZI CHECK CHAR ERROR
PES	05	SS	PHASE ENCODED STATUS-DRIVE CONFIGURED FOR PE OPERATION
PIP	13	M/SS	POSITIONING IN PROGRESS-ASSERTED BY TM02/03 ON SPACE OR DRIVE ON REWIND
RMR	02		REGISTER MODIFICATION REFUSED-REG WRITE ATTEMPTED DURING TAPE OPERATION
SAC	14		STATUS ADDRESS CHANGE-TM03 ONLY
SDWN	04	SS	SETTLE DOWN--TAPE MOTION IS STOPPING
SLA	00	SS	SLAVE ATTENTION-SEL SLAVE HAS COME ON LINE
SN00-15			SERIAL NUMBER OF TRANSPORT-BCD RERPRESENTATION(4 BITS PER DIGIT)
SPR	10		SLAVE PRESENT
SS0-2	00-02		SLAVE SELECT BITS-SPECIFIES UNIT # OF TRANSPORT TO BE USED
SSC	06	S	SLAVE STAUS CHANGE
SWC	06		SELECTED SLAVE CLOCK-WRT CLOCK SIGNAL GENERATED BY SEL SLAVE
TAP	14		ASSERTED TO INDICATE TAPE TRANSPORT
TCW	14		TAPE CONTROL WRITE-SET ON TC REG WRITE-RESET ON MOTION COMMAND-TM02 ONLY
TM	02	M	TAPE MARK DETECTED
UNS	14		UNSAFE-CLASS B
WRL	11	SS	WRITE LOCK-WRITE LOCK RING INSTALLED

SS=SELECTED TRANSPORT
S=ANY TRANSPORT
M=TM02 LOGIC

FORMAT SELECT CODES

SYSTEM	CODE	DESC
PDP10	0000	CORE DUMP
PDP10	0001	7TRK-NOT USED
PDP10	0010	ASCII
PDP10	0011	COMPATIBLE
PDP11	1100	NORMAL
PDP11	1101	CORE DUMP
PDP11	1110	15 NORMAL

```
7600xx=CSR-WRT (xx = 10 , 20 , 30 , 40 )
______________

. 20/15 * 21/14 . 22/13 . 23/12 * 24/11 . 25/10 . 26/9  * 27/8  . 28/7  . 29/6  * 30/5  . 31/4  . 32/3  * 33/2  . 34/1  . 35/0  *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
.       *XMTR   .       .Enable *       .       .       *       .       .RCVR   *Turn   .Clean  .Maint  *       .       .       *
.       *INTER  .       .SILO   *       .       .       *       .       .INTER  *on     .UARTS, .loop   *       .       .       *
.NO     *enable .NO     .ALARM  *N.U.   .NO     .NO     *NO     .NO     .enable *LINE   .SILO,  .XMTR   *N.U.   .N.U.   .N.U.   *
.       *       .       .INTER. *       .       .       *       .       .       *SCAN   .CSR    .to     *       .       .       *
.       *       .       .       *       .       .       *       .       .       *       .       .RCVR   *       .       .       *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*

7600xx=CSR-READ ( xx = 10 , 20 , 30 , 40 )
_______________

. 20/15 * 21/14 . 22/13 . 23/12 * 24/11 . 25/10 . 26/9  * 27/8  . 28/7  . 29/6  * 30/5  . 31/4  . 32/3  * 33/2  . 34/1  . 35/0  *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
.XMTR   *XMTR   .SILO   .SILO   *       .XMTR   .XMTR   *XMTR   .RCVR   .RCVR   *Turn   .Clear  .MAINT  *       .       .       *
.ready  *INTER. .has    .Alarm  *       .Line   .Line   *Line   .DONE   .INTER. *on     .UARTS  .XMTR   *       .       .       *
.       *enable .16     .INTER  *N.U.   .Bit    .Bit    *Bit    .       .       *Line   .Silo   .looped *N.U.   .N.U.   .N.U.   *
.       *       .words  .when   *       .4      .2      *1      .       .       *SCAN   .CSR    .to     *       .       .       *
.       *       .       .Bit 13 *       .       .       *       .       .       *       .15 ms  .RCVR   *       .       .       *
.       *       .       .is on  *       .       .       *       .       .       *       .long   .       *       .       .       *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*

7600xx=LPR-write ( xx= 12 , 22 , 32 , 42 )
________________

. 20/15 * 21/14 . 22/13 . 23/12 * 24/11 . 25/10 . 26/9  * 27/8  . 28/7  . 29/6  * 30/5  . 31/4  . 32/3  * 33/2  . 34/1  . 35/0  *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
.       *       .       .Turn   *RCVR  - BAUD  - RATE           .Use    .PAR    *Stop   .Char. - Length *Line  - Nr. to  load   *
.       *       .       .on     *8       4       2       1      .odd    .Error  *bit    .00=5    stop   *bit 12  allows  RCVR   *
.N.U.   *N.U.   .N.U.   .RCVR   *50=00   75=01   110=02  134=03 .PAR    .enable *0=1    .01=6                                   *
.       *       .       .Clock  *150=04  300=05  600=06  1200=07.when   .       *1=1.5  .10=7                                   *
.       *       .       .       *1800=10 2000=11 2400=12 3600=13.set    .       *or 2   .11=8                                   *
.       *       .       .       *4800=14 7200=15 9600=16 192k=17.       .       *       .                                       *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*

7600xx=RBUF-read ( xx= 12 , 22 , 32 , 42 )
________________

. 20/15 * 21/14 . 22/13 . 23/12 * 24/11 . 25/10 . 26/9  * 27/8  . 28/7  . 29/6  * 30/5  . 31/4  . 32/3  * 33/2  . 34/1  . 35/0  *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
.RCVR   *OVER   .Frame  .Parity *       .RCVR  - Line  - Number .R Data .R Data *R Data .R Data .R Data *R Data .R Data .R Data *
.Data   *RUN    .Error  .Error  *       .4       2       1      .Bit 7  .Bit 6  *Bit 5  .Bit 4  .Bit 3  *Bit 2  .Bit 1  .Bit 0  *
.valid  *Error  .       .       *N.U.   .       .       *       .       .       *       .       .       *       .       .       *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
```

7600xx=TCR-write (xx= 14 , 24 , 34 , 44)

```
. 20/15 * 21/14 . 22/13 . 23/12 * 24/11 . 25/10 . 26/9  * 27/8  . 28/7  . 29/6  * 30/5  . 31/4  . 32/3  * 33/2  . 34/1  . 35/0  *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
.DTR    *DTR    .DTR    .DTR    *DTR    .DTR    .DTR    *DTR    .XMTR   .XMTR   *XMTR   .XMTR   .XMTR   *XMTR   .XMTR   .XMTR   *
.to     *to     .to     .to     *to     .to     .to     *to     .enable .enable *enable .enable .enable *enable .enable .enable *
.Modem  *Modem  .Modem  .Modem  *Modem  .Modem  .Modem  *Modem  .       .       *       .       .       *       .       .       *
.Line 7 *Line 6 .Line 5 .Line 4 *Line 3 .Line 2 .Line 1 *Line 0 .Line 7 .Line 6 *Line 5 .Line 4 .Line 3 *Line 2 .Line 1 .Line 0 *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
```

7600xx=TCR-read (xx= 14 , 24 , 34 , 44)

```
. 20/15 * 21/14 . 22/13 . 23/12 * 24/11 . 25/10 . 26/9  * 27/8  . 28/7  . 29/6  * 30/5  . 31/4  . 32/3  * 33/2  . 34/1  . 35/0  *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
.DTR    *DTR    .DTR    .DTR    *DTR    .DTR    .DTR    *DTR    .XMTR   .XMTR   *XMTR   .XMTR   .XMTR   *XMTR   .XMTR   .XMTR   *
.to     *to     .to     .to     *to     .to     .to     *to     .enable .enable *enable .enable .enable *enable .enable .enable *
.Modem  *Modem  .Modem  .Modem  *Modem  .Modem  .Modem  *Modem  .       .       *       .       .       *       .       .       *
.Line 7 *Line 6 .Line 5 .Line 4 *Line 3 .Line 2 .Line 1 *Line 0 .Line 7 .Line 6 *Line 5 .Line 4 .Line 3 *Line 2 .Line 1 .Line 0 *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
```

7600xx=TDR-write (xx = 16 , 26 , 36 , 46)

```
. 20/15 * 21/14 . 22/13 . 23/12 * 24/11 . 25/10 . 26/9  * 27/8  . 28/7  . 29/6  * 30/5  . 31/4  . 32/3  * 33/2  . 34/1  . 35/0  *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
.BREAK  *BREAK  .BREAK  .BREAK  *BREAK  .BREAK  .BREAK  *BREAK  .XMT    .XMT    *XMT    .XMT    .XMT    *XMT    .XMT    .XMT    *
.       *       .       .       *       .       .       *       .Data   .Data   *Data   .Data   .Data   *Data   .Data   .Data   *
.Line 7 *Line 6 .Line 5 .Line 4 *Line 3 .Line 2 .Line 1 *Line 0 .Bit 7  .Bit 6  *Bit 5  .Bit 4  .Bit 3  *Bit 2  .Bit 1  .Bit 0  *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
```

Caution: "BREAK" transmits a stream of 0 bits   (NULL-chars) , not 0-characters !!

7600xx=MSR-read (xx= 16 , 26 , 36 , 46)

```
. 20/15 * 21/14 . 22/13 . 23/12 * 24/11 . 25/10 . 26/9  * 27/8  . 28/7  . 29/6  * 30/5  . 31/4  . 32/3  * 33/2  . 34/1  . 35/0  *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
.Carrier*Carrier.Carrier.Carrier*Carrier.Carrier.Carrier*Carrier.Ring   .Ring   *Ring   .Ring   .Ring   *Ring   .Ring   .Ring   *
.detect *detect .detect .detect *detect .detect .detect *detect .Ind.   .Ind.   *Ind.   .Ind.   .Ind.   *Ind.   .Ind.   .Ind.   *
.Line 7 *Line 6 .Line 5 .Line 4 *Line 3 .Line 2 .Line 1 *Line 0 .Line 7 .Line 6 *Line 5 .Line 4 .Line 3 *Line 2 .Line 1 .Line 0 *
.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*-------.-------.-------*
```

BUSADR	NAME	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
		C15	C14	C13	C12	C11	C10	C09	C08	C07	C06	C05	C04	C03	C02	C01	C00	
775400	LPCSRA	ERR	PAG 0	ILCHR	VFURY	ONLIN	DEL	INIT	RESET	DONE	INTEN	A17	A16	MOD00	MOD01	PAREN	GO	R/W
775402	LPCSRB	VDATA	LA180	NRDY	PAR	OPVFU	TST02	TST01	TST00	OFLIN	VFUER	PARER	MEMER	RAMER	SYNTO	DEMTO	GOERR	R/W
775404	LPBSAD	ADR15	ADR14	ADR13	ADR12	ADR11	ADR10	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02	ADR01	ADR00	R/W
775406	LPBCTR	SP	SP	SP	SP	BC11	BC10	BC09	BC08	BC07	BC06	BC05	BC04	BC03	BC02	BC01	BC00	R/W
775510	LPPCTR	SP	SP	SP	SP	PC11	PC10	PC09	PC08	PC07	PC06	PC05	PC04	PC03	PC02	PC01	PC00	R/W
775412	LPRAMD	SP	SP	SP	R PAR	R INT	R DEL	R TRN	R PI	RDAT7	RDAT6	RDAT5	RDAT4	RDAT3	RDAT2	RDAT1	RDAT0	R/W
775414	LPCCTR/LPCBUF	CLCT7	CLCT6	CLCT5	CLCT4	CLCT3	CLCT2	CLCT1	CLCT0	CBUF7	CBUF6	CBUF5	CBUF4	CBUF3	CBUF2	CBUF1	CBUF0	R/W
775416	LPTDAT/LPCKSM	CHK15	CHK14	CHK13	CHK12	CHK11	CHK10	CHK 09	CHK08	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	RONLY

LPCSRA	CONTROL AND STATUS REGISTER A
LPCSRB	CONTROL AND STATUS REGISTER B
LPBSAD	DMA BUS ADDRESS REGISTER
LPBCTR	DMA BYTE COUNT REGISTER
LPPCTR	PAGE COUNT REGISTER
LPRAMD	RAM DATA REGISTER
LPCCTR/LPCBUF	COLUMN COUNT REGISTER(HIGH BYTE),CHARACTER BUFFER REGISTER(LOW)
LPCKSM/LPTDAT	CHECKSUM REGISTER ,PRINTER DATA REGISTER

```
A16     BUS ADDRESS BIT 16
A17     BUS ADDRESS BIT 17
ADRXX   ADDRESS OF NEXT BYTE OF PRINTER DATA BUFFER-R/W
BCXX    TWO'S COMPLIMENT # OF BYTES REMAINING TO BE TRANSFERED,0=DONE&INTERUPT
CBUFX   LAST DATA BYTE FROM MEM IN NORMAL OPERATION-R/W
CHKXX   RUNNING VALUE OF CHARACTERS IN CURRENT TRANSFER BLOCK,CHECKSUM-RONLY
CLCTX   COLUMN FOR NEXT PRINTABLE CHARACTER-RESET TO ZERO AFTER PRINTING-R/W
DATAX   DATA SENT TO PRINTER SAMPLED AT TRANSMITTERS-RONLY
DEL     DELHLD-LAST CHARACTER RECIEVED WAS A DELIMITER(RAM BIT 10 SET)-R/W
DEMTO   DEMAND TIMEOUT,CONTROLLER TIMEOUT WAITING FOR PRINTER TO ACK DATA,RONLY
ERR     ERROR-RONLY-LOGICAL "OR"OF ALL ERRORS
GO      GO START DMA TRANSFERS=1,R/W,RESET BY LPBCTR OVERFLOW,ERR,PAG0,ILCHR
GOERR   GO SET AND ERROR UP ,OR DEMAND NOT UP WHEN GO SET,WRITTEN FOR DIAGS
ILCHR   ILLEGAL OR UNDEFINED CHARACTER CAUSED CONTROLLER TO STOP-RONLY
INIT    LOCAL INITIALIZE-RESETS FLAGS,SETS DONE,LIKE A UNIBUS INITWONLY
INTEN   INTERUPT ENABLE-ERROR,PAG 0,ILCHR,DONE ALL CAUSE INTERUPTS
LA180   SET IF LA180 TYPE PRINTER
MEMER   MEMORY PARITY ERROR DETECTED DURING TRANSFER ,RONLY
MODXX   MODE BITS
        00=NORMAL
        01=TSTMOD=TEST CONTROLLER BY INHIBIT PRINTING BUT CONTROLLER ACTS NORMAL
        10=VFULOD=DMA LOAD TO THE VFU,RAM DISABLED UNLESS TSTMOD SET
        11=RAMLOD=LOAD THE TRANSLATION RAM,DMA MODE ONLY
NRDY    SET IF PRINTER NOT READY DUE TO A FAULT OTHER THAN TAPE FAULT
OFLIN   PRINTER IS OFF LINE
ONLIN   PRINTER IS READY AND ON LINE-RONLY
OPVFU   OPTICAL VFU ,HARDWIRED ,ZERO IF DAVFU AS IN LP05
PAG 0   PAGE COUNTER INCREMENTED TO ZERO-RONLY
PAR     THE PARITY BIT AS SENT TO THE PRINTER
PAREN   PARITY ENABLE,R/W,ENABLES RAM AND MEM PARITY CHECKING
PARER   PRINTER RECIEVED BAD DATA FROM CONTROLLER,LP05 DOESN'T USE IT,RONLY
PCXX    PAGE COUNT,DECREMENTED AT TOP OF PAGE LOADED WITH PAGE COUNT LIMIT-R/W
R DEL   DELIMITER BIT-TAKE DATA FROM RAM INSTEAD OF CBUF-R/W
R INT   INTERRUPT BIT-GENERATE A BR,DO NOT STROBE DATA TO PRINTER-R/W
R PAR   RAM PARITY BIT-RONLY
R PI    PAPER INSTRUCTION BIT-RDAT SENT TO PRINTER DAVFU-R/W-REFER TO MANUAL
R TRN   TRANSLATION BIT-CAUSES RDAT TO BE SENT TO PRINTER-R/W-REFER TO MANUAL
RAMER   RAM PARITY ERROR DURING DMA TRANSFER,RONLY
RDAT    RAM DATA-ADR IS IN LPCBUF-BIT 13 IS RONLY-R/W
RESET   RESET ERRORS,SET DONE,RESET GO-WONLY
RESET   RESET ERRORS,SET DONE,RESET GO-WONLY
SYNTO   MASTER SYNC TIMEOUT DURING MEM REFERENCE,SSYNC NOT RECEIVED,RONLY
TSTXX   TEST BITS DECODED BY CONTROLLER FOR VARIOUS TESTS R/W BITS
VDATA   VALID DATA FLAG,TOGGLES WITH DATA STROBE IN PRINTER INTERFACE
VFUER   VERTICAL FORMAT UNIT ERROR,DAVFU ERROR,RONLY
VFURY   DAVFU IS LOADED AND READY-FORCED TRUE IF OPTICAL VFU-RONLY
```

```
===================================================================================================================================
!BUSADR!NAME  !  RD/WRT  !20   *21   !22   !23   *24   !25   !26   *27   !28   !29   *30   !31   !32   *33   !34   !35   *
-----------------------------------------------------------------------------------------------------------------------------------
!      !      !          !C15  *C14  !C13  !C12  *C11  !C10  !C09  *C08  !C07  !C06  *C05  !C04  !C03  *C02  !C01  !C00  *
===================================================================================================================================
!777164! CDBA!          !     * BA8 ! BA7 ! BA6 *     !     !     * BA5 ! BA4 ! BA3 *     !     !     * BA2 ! BA1 ! BA0 *
-----------------------------------------------------------------------------------------------------------------------------------
!777162! CDCC !          !     * CC8 ! CC7 ! CC6 *     !     !     * CC5 ! CC4 ! CC3 *     !     !     * CC2 ! CC1 ! CC0 *
-----------------------------------------------------------------------------------------------------------------------------------
!777166! CDDB !NONPACKING!     *     !     !     *ZONE !ZONE !ZONE *ZONE !ZONE !ZONE *ZONE !ZONE !ZONE *ZONE !ZONE !ZONE *
!      !      ! MODE     !     *     !     !     * 12  ! 11  ! 0   * 1   ! 2   ! 3   * 4   ! 5   ! 6   * 7   ! 8   ! 9   *
-----------------------------------------------------------------------------------------------------------------------------------
!777166! CDDB ! PACKING  !     *     !     !     *     !     !     *     !ZONE !ZONE *ZONE !ZONE !ZONE *    OCTAL CODE   *
!      !      ! MODE     !     *     !     !     *     !     !     *     !  12 ! 11  * 0   ! 9   ! 8   *    ZONES  1-7   *
-----------------------------------------------------------------------------------------------------------------------------------
!777160! CDST !          !ERR  *READR! END ! OFF *DATA ! DATA! NO  *POWER!READY!INTER*     !     !TRANS*HOPER!DATA !READ *
!      !      !          !     *CHECK! FILE! LINE*ERROR! LATE! MEM *CLEAR!     ! ENB *     !     ! LINE*CHECK! PACK!     *
-----------------------------------------------------------------------------------------------------------------------------------

        14      READER CHECK                    03      HEADER TRANSITION TO ON-LINE
        13      END OF FILE                     02      HOPPER CHECK
        09      NONEXISTENT MEMORY              01      DATA PACKING
        06      INTERRUPT ENABLE


-----------------------------------------------------------------------------------------------------------------------------------

CDBA -  CURRENT ADDRESS REGISTER- MEMORY ADDRESS INTO WHICH THE NEXT COLUMN OF DATA IS TO BE STORED
CDCC -  CURRENT COUNT REGISTER- COLUMN COUNT 2'S COMPLEMENT OF THE NUMBER OF COLUMNS TO BE TRANSFERRED TO MEMORY
CDDB -  DATA BUFFER REGISTER- NONPACKING MODE
                              PACKING MODE
CDST -  STATUS AND CONTROL REGISTER
```

```
=================================================================================================================================
!BUSADR!NAME  !  RD/WRT  !20   *21   !22   !23   *24   !25   !26   *27   !28   !29   *30   !31   !32   *33   !34   !35   *
---------------------------------------------------------------------------------------------------------------------------------
!      !      !          !C15  *C14  !C13  !C12  *C11  !C10  !C09  *C08  !C07  !C06  *C05  !C04  !C03  *C02  !C01  !C00  *
=================================================================================================================================
!76XXX2!PARCSR!   WRT    !DEC  *     !     !SEC  *     !     ! CRC *     !    SECONDARY STATION ADDRESS RECEIVER SYNC    *
!      !      !          !MODE *     !     !MODE *     !     !INHIB*     !                                    *
---------------------------------------------------------------------------------------------------------------------------------
NOTE: XXX = 030(DUP0),031(DUP1)

        12      SECONDARY MODE SELECT                   09      CRC INHIBIT


=================================================================================================================================
!76XXX0!RXCSR !          !DATA *RING !CLEAR!CAR- * RX  !SECRX!DATA *STRIP! RX  ! RX  *DATA ! RX  !SECTX* REQ !TERM !DATA *
!      !      !          !SET A*     !SEND !RIER *ACTIV!DATA !READY*SYNC !DONE !INTEN*INTEN! ENB !DATA *SEND ! RDY !SET B*
---------------------------------------------------------------------------------------------------------------------------------
                         !  R  *  R  !  R  !  R  *  R  !  R  !  R  * R/W !  R  ! R/W * R/W ! R/W ! R/W * R/W ! R/W !  R  *

        NOTE: XXX = 030(DUP0),031(DUP1)

        15      DATA SET CHANGE A                       06      RECEIVER INTERRUPT ENABLE
        13      CLEAR TO SEND                           05      DATA SET INTERRUPT ENABLE
        11      RECEIVER ACTIVER                        04      RECEIVER ENABLE
        10      SECONDARY RECEIVED DATA                 03      SECONDARY TRANSMIT DATA
        09      DATA SET READY                          02      REQUEST TO SEND
        07      RECEIVER DONE                           01      DATA TERMINAL READY
                                                        00      DATA SET CHANGE B
=================================================================================================================================
!76XXX2!RXDBUF!          ! RX  * RX  !     !RXCRC*     ! RX  !RXEND*START!                 RECEIVE DATA                  *
!      !      !          ! ERR *OVRUN!     ! ERR *     !ABORT! MSG * MSG !                                               *
---------------------------------------------------------------------------------------------------------------------------------

        NOTE: XXX = 030(DUP 0), 031(DUP1)

        15      RECEIVER ERROR                          10      RECEIVER ABORT
        14      RECEIVER OVERRUN                        09      END OF RECEIVED MESSAGE
        12      RECEIVER CRC ERROR                      08      START OF RECEIVED MESSAGE


=================================================================================================================================
!76XXX4!TXCSR            !DATA *MAINT!MAINT!MAINT*MAINT!MAINT! TX  *DEVIC! TX  ! TX  *     !     !HALF *     !     !     *
!      !                 !LATE * OUT ! CLK !MODEA*MODES! IN  ! ACT *RESET!DONE !INTEN*     !SEND ! DUP *     !     !     *
---------------------------------------------------------------------------------------------------------------------------------
                         !  R  *  R  ! R/W ! R/W * R/W ! R/W !  R  *  W  !  R  ! R/W *     ! R/W * R/W *

        NOTE: XXX = 030(DUP0), 031(DUP1)

        15      TRANSMITTER DATA LATE ERROR                     10      MAINTENANCE INPUT DATA
        14      MAINTENANCE TRANSMIT DATA OUT                   09      TRANSMITTER ACTIVE
        13      MAINTENANCE CLOCK                               08      DEVICE RESET
        12      MAINTENANCE MODE SELECT A                       07      TRANSMITTER DONE
        11      MAINTENANCE MODE SELECT B                       06      TRANSMITTER INTERRUPT ENABLE
                                                                03      HALF DUPLEX FULL DUPLEX
```

```
==================================================================================================================================
!76XXX6!TXDBUF          !RXCRC*     !TXCRC!MAINT* TX  ! END !START*                                                          *
!      !                ! IN  *     ! IN  !TIMER*ABORT! MSG ! MSG *                        DATA                              *
----------------------------------------------------------------------------------------------------------------------------------
                        !     *  R  !     !  R  *  R  ! R/W ! R/W * R/W !                  R/W                               *



        14      RECEIVER CRC INTERNAL                   10      TRANSMIT ABORT
        12      TRANSMITTER CRC INTERNAL                09      END OF TRANSMITTED MESSAGE
        11      MAINTENANCE TIMER                       08      TRANSMIT START OF MESSAGE


PARCSR  -       PARAMETER CONTROL AND STATUS REGISTER
RXCSR   -       RECEIVER CONTROL AND STATUS REGISTER
RXDBUF  -       RECEIVER DATA BUFFER REGISTER
TXCSR   -       TRANSMITTER CONTROL AND STATUS REGISTER
TXDBUF  -       TRANSMITTER DATA BUFFER REGISTER
```

UNIBUS ADAPTER MINI CHECK

IF YOU ARE HAVING PROBLEMS BOOTING FROM EITHER YOUR DISK OR MAGTAPE , TRY THIS UNIBUS WRAP AROUND TEST. THIS TEST WILL CHECK OUT APPROXIMATELY 95% OF THE UBA MODULE(M8619).

```
****************************************************************************

     EI UBA#,763000           ;ADDRESS OF PAGING RAM FOR LOCATION ZERO
     DI 40000 (18 BIT,VALID),140000 (36 BIT,VALID),240000 (16 BIT,VALID)
     EI UBA#,763101           ;ADDRESS OF UBA MAINTENANCE REGISTER
     DI 1                     ;BIT 35 FOR WRAP AROUND
     EI UBA#,ADR              ;LOAD MEMORY ADDRESS TO BE WRITTEN
             ;LOW ORDER UNIBUS ADDRESS BITS ARE DISCARDED
             ;C00 SAYS 8 BIT UNIBUS BYTE MODE OR NOT
             ;C01 SAYS RIGHT HALF OR LEFT HALF

                              3,,0=LH MEMORY LOCATION 0
                              3,,2=RH MEMORY LOCATION 0
                              3,,4=LH MEMORY LOCATION 1
                              3,,6=RH MEMORY LOCATION 1
DI XXXXXX(DATA PATTERN)       ;DATA WILL LOOP THROUGH UBA TO MEMORY LOCATION
EM MEMORY LOCATION            ;CHECK TO SEE IF IT GOT THERE


EXAMPLE 1:       EI 1763000
                 DI 140000
                 EI 1763101
                 DI 1
                 EI 1000100
                 DI 555333
                 EI 1000102
                 DI 121212
                 EM 20
      CORRECT DATA SHOULD BE     000000,,000020/555333,,121212


EXAMPLE 2:       EI 1763000
                 DI 40000
                 EI 1763101
                 DI 1
                 EI 1000100
                 DI 777777
                 EM 20
     CORRECT DATA SHOULD BE      000000,,000020/777777,,000020

EXAMPLE 3:       EI 1763000
                 DI 240000
                 EI 1763101
                 DI 1
                 EI 1000100
                 DI 743216
                 EM 20
     CORRECT DATA SHOULD BE      000000,,000020/143216,,000020
```

CHAPTER 4

MICRO MACHINE

The "magic" Halt-status address is at "power-up" set to 376000 , changed by
TOPS-20 BOOT to 400 and by TOPS-10 BOOT to 424.

376000	400	424:	MAG (ONES IN BITS 1-36, REST ZERO)
376001	401	425:	PC (ADDRESS OF CURRENT INSTRUCTION + 1)
376002	402	426:	HR (CURRENT INSTRUCTION)
376003	403	427:	AR (TEMP -- MEM OP AT INST START)
376004	404	430:	ARX (TEMP -- LOW ORDER HALF OF DOUBLE PREC)
376005	405	431:	BR (TEMP)
376006	406	432:	BRX (TEMP -- LOW ORDER HALF OF DOUBLE PREC BR!BRX)
376007	407	433:	ONE (THE CONSTANT 1)
376010	410	434:	EBR (EXEC BASE REGISTER)
376011	411	435:	UBR (USER BASE REGISTER)
376012	412	436:	MASK (ONES IN BITS 0-35, ZERO IN -1, -2, 36 AND 37)
376013	413	437:	FLG (FLAG BITS) ! PAGE FAIL CODE
376014	414	440:	PI (PI SYSTEM STATUS REGISTER [RDPI])
376015	415	441:	XWD1 (1 IN EACH HALF WORD)
376016	416	442:	T0 (TEMP)
376017	417	443:	T1 (TEMP)
376020	420	444:	VMA and flags

This block of information is stored in memory by micro-code at any of the following events:

1. Entering the HALT-loop

2. Requested stop by 8080 [SM 1-command etc.]

3. Executing a HALT-instruction

4. Capture of an uncorrectable KS10 or m-code error

Default start-address of HALT-status block is 376000 ,
TOPS20 resets this at startup to 400
TOPS10 resets this at startup to 424

```
;                                                               ; 3633  1561:
;                                                               ; 3634  SUB:    [AR]-AC-[AR],
;                                                               ; 3635          AD FLAGS, 3T,
;       U 1561, 1500,2551,0303,0274,4463,7701,4200,0001,0001    ; 3636          EXIT
;         [--]  [--] !*!! [][] !!![-][][-][]! !!!   [------]
;         !     !    !!!! ! !  !!!!  ! !  ! ! !!!        !
;         !     !    !!!! ! !  !!!!  ! !  ! ! !!!        +---- # (MAGIC NUMBER)
;         !     !    !!!! ! !  !!!!  ! !  ! ! !!!
;         !     !    !!!! ! !  !!!!  ! !  ! ! !!+------------- MULTI PREC, MULTI SHIFT, CALL (4S, 2S, 1S)
;         !     !    !!!! ! !  !!!!  ! !  ! ! !!
;         !     !    !!!! ! !  !!!!  ! !  ! ! !+-------------- FM WRITE, MEM, DIVIDE (4S, 2S, 1S)
;         !     !    !!!! ! !  !!!!  ! !  ! ! !
;         !     !    !!!! ! !  !!!!  ! !  ! ! +--------------- CRY38, LOAD SC, LOAD FE (4S, 2S, 1S)
;         !     !    !!!! ! !  !!!!  ! !  ! !
;         !     !    !!!! ! !  !!!!  ! !  ! +----------------- T
;         !     !    !!!! ! !  !!!!  ! !  !
;         !     !    !!!! ! !  !!!!  ! !  +------------------- SKIP
;         !     !    !!!! ! !  !!!!  ! !
;         !     !    !!!! ! !  !!!!  ! +---------------------- DISP
;         !     !    !!!! ! !  !!!!  !
;         !     !    !!!! ! !  !!!!  +------------------------ SPEC
;         !     !    !!!! ! !  !!!!
;         !     !    !!!! ! !  !!!+--------------------------- CLOCKS & PARITY (CLKR, GENR, CHKR, CLKL, GENL, CHKL)
;         !     !    !!!! ! !  !!!
;         !     !    !!!! ! !  !!+---------------------------- DBM
;         !     !    !!!! ! !  !!
;         !     !    !!!! ! !  !+----------------------------- DBUS
;         !     !    !!!! ! !  !
;         !     !    !!!! ! !  +------------------------------ RAM ADDRESS
;         !     !    !!!! ! !
;         !     !    !!!! ! +--------------------------------- B
;         !     !    !!!! !
;         !     !    !!!! +----------------------------------- A
;         !     !    !!!!
;         !     !    !!!+------------------------------------- DEST
;         !     !    !!!
;         !     !    !!+-------------------------------------- RSRC
;         !     !    !!
;         !     !    !+--------------------------------------- LSRC   ]
;         !     !    !                                                ] - AD
;         !     !    +---------------------------------------- ALU    ]
;         !     !
;         !     +--------------------------------------------- J
;         !
;         LOCATION OF THIS MICRO WORD
;       DISPATCH ROM DEFINITIONS

;ALL ON DPEA

        A/=<2:5>                ;OPERAND FETCH MODE
                READ=0          ;READ
                WRITE=1         ;WRITE
                DREAD=2         ;DOUBLE READ
                DBLAC=3         ;DOUBLE AC
                SHIFT=4         ;SIMPLE SHIFT
                DSHIFT=5        ;DOUBLE SHIFT
                FPI=6           ;FLOATING POINT IMMEDIATE
                FP=7            ;FLOATING POINT
                RD-PF=10        ;READ, THEN START PREFETCH
                DFP=11          ;DOUBLE FLOATING POINT
                IOT=12          ;CHECK FOR IO LEGAL THEN SAME AS I

        B/=<8:11>               ;STORE RESULTS AS
                SELF=4          ;SELF
                DBLAC=5         ;DOUBLE AC
                DBLB=6          ;DOUBLE BOTH
                AC=15           ;AC
                MEM=16          ;MEMORY
                BOTH=17         ;BOTH

        ;B-FIELD WHEN USED IN FLOATING POINT OPERATIONS
        ROUND/=<8>              ;ROUND THE RESULT
        MODE/=<9>               ;SEPARATE ADD/SUB & MUL/DIV ETC.
        FL-B/=<10:11>           ;STORE RESULTS AS
                AC=1            ;AC
                MEM=2           ;MEMORY
                BOTH=3          ;BOTH

        J/=<12:23>              ;DISPATCH ADDRESS (MUST BE 1400 TO 1777)

        ACDISP/=<24>            ;DISPATCH ON AC FIELD
        I/=<25>                 ;IMMEDIATE DISPATCH. DISP/AREAD DOES A DISP/DROM
                                ; IF THIS BIT IS SET.
        READ/=<26>              ;START A READ AT AREAD
        TEST/=<27>              ;START A WRITE TEST  AT AREAD
        COND FUNC/=<28>         ;START A MEMORY CYCLE ON BWRITE
        VMA/=<29>D,1            ;LOAD THE VMA ON AREAD
        WRITE/=<30>             ;START A WRITE ON AREAD
```

CRA6 CRAM 00 H	CRA6 J 00 H	NC
CRA6 CRAM 01 H	CRA6 J 01 H	NC
CRA6 CRAM 02 H	CRA6 J 02 H	NC
CRA6 CRAM 03 H	CRA6 J 03 H	NC
CRA6 CRAM 04 H	CRA6 J 04 H	NC
CRA6 CRAM 05 H	CRA6 J 05 H	NC
CRA6 CRAM 06 H	CRA6 J 06 H	NC
CRA6 CRAM 07 H	CRA6 J 07 H	NC
CRA6 CRAM 08 H	CRA6 J 08 H	NC
CRA6 CRAM 09 H	CRA6 J 09 H	NC
CRA6 CRAM 10 H	CRA6 J 10 H	NC
CRA6 CRAM 11 H	CRA6 J 11 H	NC
CRA6 CRAM 12 H	CRA6 T00 H	A12E2
CRA6 CRAM 13 H	CRA6 T01 H	A12M2
CRA6 CRAM 14 H	CRA6 CALL H	B12E2
CRA6 CRAM 15 H	CRA6 SKIP EN 40 L	B12M2
CRA6 CRAM 16 H	CRA6 SKIP EN 20 L	C12E2
CRA6 CRAM 17 H	CRA6 SKIP EN 10 L	C12M2
CRA6 CRAM 18 H	CRA6 SPEC EN 40 H	A12F1
CRA6 CRAM 19 H	CRA6 SPEC EN 20 H	A12N1
CRA6 CRAM 20 H	CRA6 SPEC EN 10 H	B12F1
CRA6 CRAM 21 H	CRA6 DISP EN 40 L	B12N1
CRA6 CRAM 22 H	CRA6 DISP EN 20 L	C12F1
CRA6 CRAM 23 H	CRA6 DISP EN 10 L	C12N1
CRA6 CRAM 24A H	CRA6 CRAM PARITY BIT A H	A12F2
CRA6 CRAM 24B H	CRA6 CRAM PARITY BIT B H	A12J2
CRA6 CRAM 25A H	CRA6 CARRY IN A H	A12N2
CRA6 CRAM 25B H	CRA6 CARRY IN B H	A12R2
CRA6 CRAM 26A H	CRA6 MEM FUNCTION A H	B12F2
CRA6 CRAM 26B H	CRA6 MEM FUNCTION B H	B12J2
CRA6 CRAM 27A H	CRA6 DISP SEL 4A H	B12N2
CRA6 CRAM 27B H	CRA6 DISP SEL 4B H	B12R2
CRA6 CRAM 28A H	CRA6 DISP SEL 2A H	C12F2
CRA6 CRAM 28B H	CRA6 DISP SEL 2B H	C12J2
CRA6 CRAM 29A H	CRA6 DISP SEL 1A H	C12N2
CRA6 CRAM 29B H	CRA6 DISP SEL 1B H	C12R2
CRA6 CRAM 30A H	CRA6 SPEC SEL 4A H	A12K1
CRA6 CRAM 30B H	CRA6 SPEC SEL 4B H	A12K2
CRA6 CRAM 31A H	CRA6 SPEC SEL 2A H	A12S1
CRA6 CRAM 31B H	CRA6 SPEC SEL 2B H	A12S2
CRA6 CRAM 32A H	CRA6 SPEC SEL 1A H	B12K1
CRA6 CRAM 32B H	CRA6 SPEC SEL 1B H	B12K2
CRA6 CRAM 33A H	CRA6 SKIP SEL 4A H	B12S1
CRA6 CRAM 33B H	CRA6 SKIP SEL 4B H	B12S2
CRA6 CRAM 34A H	CRA6 SKIP SEL 2A H	C12K1
CRA6 CRAM 34B H	CRA6 SKIP SEL 2B H	C12K2
CRA6 CRAM 35A H	CRA6 SKIP SEL 1A H	C12S1
CRA6 CRAM 35B H	CRA6 SKIP SEL 1B H	C12S2

CRM2 BIT 36A H	CRM2 # 06A H	A11D1
CRM2 BIT 36B H	CRM2 # 06B H	A11D2
CRM2 BIT 37A H	CRM2 # 07A H	A11M1
CRM2 BIT 37B H	CRM2 # 07B H	A11M2
CRM2 BIT 38A H	CRM2 # 08A H	B11D1
CRM2 BIT 38B H	CRM2 # 08B H	B11D2
CRM2 BIT 39A H	CRM2 # 09A H	B11M1
CRM2 BIT 39B H	CRM2 # 09B H	B11M2
CRM2 BIT 40A H	CRM2 # 10A H	C11D1
CRM2 BIT 40B H	CRM2 # 10B H	C11D2
CRM2 BIT 41A H	CRM2 # 11A H	C11M1
CRM2 BIT 41B H	CRM2 # 11B H	C11M2
CRM2 BIT 42A H	CRM2 # 12A H	D11D1
CRM2 BIT 42B H	CRM2 # 12B H	D11D2
CRM2 BIT 43A H	CRM2 # 13A H	D11M1
CRM2 BIT 43B H	CRM2 # 13B H	D11M2
CRM2 BIT 44A H	CRM2 # 14A H	E11D1
CRM2 BIT 44B H	CRM2 # 14B H	E11D2
CRM2 BIT 45A H	CRM2 # 15A H	E11M1
CRM2 BIT 45B H	CRM2 # 15B H	E11M2
CRM2 BIT 46A H	CRM2 # 16A H	F11D1
CRM2 BIT 46B H	CRM2 # 16B H	F11D2
CRM2 BIT 47A H	CRM2 # 17A H	F11M1
CRM2 BIT 47B H	CRM2 # 17B H	F11M2
CRM2 BIT 48H	CRM2 RAMFILE WRITE H	A11E1
CRM2 BIT 49 H	CRM2 SPARE 49 H	A11N1
CRM2 BIT 50 H	CRM2 PARITY EN LEFT H	B11E1
CRM2 BIT 51 H	CRM2 PARITY EN RIGHT H	B11N1
CRM2 BIT 52 H	CRM2 DIVIDE H	C11E1
CRM2 BIT 53 H	CRM2 MULTI PRECISION H	C11N1
CRM2 BIT 54 H	CRM2 # 00 H	D11E1
CRM2 BIT 55 H	CRM2 # 01 H	D11N1
CRM2 BIT 56 H	CRM2 # 02 H	E11E1
CRM2 BIT 57 H	CRM2 # 03 H	E11N1
CRM2 BIT 58 H	CRM2 # 04 H	F11E1
CRM2 BIT 59 H	CRM2 # 05 H	F11N1
CRM2 BIT 60 H	CRM2 FUNC 04 H	A11H2
CRM2 BIT 61 H	CRM2 FUNC 02 H	A11R2
CRM2 BIT 62 H	CRM2 FUNC 01 H	B11H2
CRM2 BIT 63 H	CRM2 SOURCE L 04 H	B11R2
CRM2 BIT 64 H	CRM2 SOURCE L 02 H	C11H2
CRM2 BIT 65 H	CRM2 SOURCE L 01 H	C11R2
CRM2 BIT 66 H	CRM2 SOURCE R 04 H	D11H2
CRM2 BIT 67 H	CRM2 SOURCE R 02 H	D11R2
CRM2 BIT 68 H	CRM2 SOURCE R 01 H	E11M2
CRM2 BIT 69 H	CRM2 DBM SEL 4 H	E11R2
CRM2 BIT 70 H	CRM2 DBM SEL 2 H	F11H2
CRM2 BIT 71 H	CRM2 DBM SEL 1 H	F11R2
CRM2 BIT 72 H	CRM2 DBUS SEL 2 H	A11J1
CRM2 BIT 73 H	CRM2 DBUS SEL 1 H	A11S1
CRM2 BIT 74 H	CRM2 DP B ADR 10 H	B11J1
CRM2 BIT 75 H	CRM2 DP B ADR 04 H	B11S1
CRM2 BIT 76 H	CRM2 DP B ADR 02 H	C11J1
CRM2 BIT 77 H	CRM2 DP B ADR 01 H	C11S1

CRM2 BIT 78 H	CRM2 CLK EN LEFT H	D11J1
CRM2 BIT 79 H	CRM2 CLK EN RIGHT H	D11S1
CRM2 BIT 80 H	CRM2 DP A ADR 10 H	E11J1
CRM2 BIT 81 H	CRM2 DP A ADR 04 H	E11S1
CRM2 BIT 82 H	CRM2 DP A ADR 02 H	F11J1
CRM2 BIT 83 H	CRM2 DP A ADR 01 H	F11S1
CRM2 BIT 84 H	CRM2 RAM ADR 04 H	A11J
CRM2 BIT 85 H	CRM2 RAM ADR 02 H	A11S2
CRM2 BIT 86 H	CRM2 RAM ADR 01 H	B11J2
CRM2 BIT 87 H	CRM2 DEST 04 H	B11S2
CRM2 BIT 88 H	CRM2 DEST 02 H	C11J2
CRM2 BIT 89 H	CRM2 DEST 01 H	C11S2
CRM2 BIT 90 H	CRM2 SC EN H	D11J2
CRM2 BIT 91 H	CRM2 FE EN H	D11S2
CRM2 BIT 92 H	CRM2 PARITY CHECK LEFT H	E11J2
CRM2 BIT 93 H	CRM2 PARITY CHECK RIGHT H	E11S2
CRM2 BIT 94 H	CRM2 CRAM PARITY BIT H	F11J2
CRM2 BIT 95 H	CRM2 MARK BIT H	F11S2

CRM2 BIT 54 H	CRM2 # 00 H	D11E1
CRM2 BIT 55 H	CRM2 # 01 H	D11N1
CRM2 BIT 56 H	CRM2 # 02 H	E11E1
CRM2 BIT 57 H	CRM2 # 03 H	E11N1
CRM2 BIT 58 H	CRM2 # 04 H	F11E1
CRM2 BIT 59 H	CRM2 # 05 H	F11N1
CRM2 BIT 36A H	CRM2 # 06A H	A11D1
CRM2 BIT 36B H	CRM2 # 06B H	A11D2
CRM2 BIT 37A H	CRM2 # 07A H	A11M1
CRM2 BIT 37B H	CRM2 # 07B H	A11M2
CRM2 BIT 38A H	CRM2 # 08A H	B11D1
CRM2 BIT 38B H	CRM2 # 08B H	B11D2
CRM2 BIT 39A H	CRM2 # 09A H	B11M1
CRM2 BIT 39B H	CRM2 # 09B H	B11M2
CRM2 BIT 40A H	CRM2 # 10A H	C11D1
CRM2 BIT 40B H	CRM2 # 10B H	C11D2
CRM2 BIT 41A H	CRM2 # 11A H	C11M1
CRM2 BIT 41B H	CRM2 # 11B H	C11M2
CRM2 BIT 42A H	CRM2 # 12A H	D11D1
CRM2 BIT 42B H	CRM2 # 12B H	D11D2
CRM2 BIT 43A H	CRM2 # 13A H	D11M1
CRM2 BIT 43B H	CRM2 # 13B H	D11M2
CRM2 BIT 44A H	CRM2 # 14A H	E11D1
CRM2 BIT 44B H	CRM2 # 14B H	E11D2
CRM2 BIT 45A H	CRM2 # 15A H	E11M1
CRM2 BIT 45B H	CRM2 # 15B H	E11M2
CRM2 BIT 46A H	CRM2 # 16A H	F11D1
CRM2 BIT 46B H	CRM2 # 16B H	F11D2
CRM2 BIT 47A H	CRM2 # 17A H	F11M1
CRM2 BIT 47B H	CRM2 # 17B H	F11M2
CRA6 CRAM 14 H	CRA6 CALL H	B12E2
CRA6 CRAM 25A H	CRA6 CARRY IN A H	A12N2
CRA6 CRAM 25B H	CRA6 CARRY IN B H	A12R2
CRM2 BIT 78 H	CRM2 CLK EN LEFT H	D11J1
CRM2 BIT 79 H	CRM2 CLK EN RIGHT H	D11S1
CRA6 CRAM 24A H	CRA6 CRAM PARITY BIT A H	A12F2
CRA6 CRAM 24B H	CRA6 CRAM PARITY BIT B H	A12J2
CRM2 BIT 94 H	CRM2 CRAM PARITY BIT H	F11J2
CRM2 BIT 71 H	CRM2 DBM SEL 1 H	F11R2
CRM2 BIT 70 H	CRM2 DBM SEL 2 H	F11H2
CRM2 BIT 69 H	CRM2 DBM SEL 4 H	E11R2
CRM2 BIT 73 H	CRM2 DBUS SEL 1 H	A11S1
CRM2 BIT 72 H	CRM2 DBUS SEL 2 H	A11J1
CRM2 BIT 89 H	CRM2 DEST 01 H	C11S2
CRM2 BIT 88 H	CRM2 DEST 02 H	C11J2
CRM2 BIT 87 H	CRM2 DEST 04 H	B11S2
CRA6 CRAM 23 H	CRA6 DISP EN 10 L	C12N1
CRA6 CRAM 22 H	CRA6 DISP EN 20 L	C12F1
CRA6 CRAM 21 H	CRA6 DISP EN 40 L	B12N1
CRA6 CRAM 29A H	CRA6 DISP SEL 1A H	C12N2
CRA6 CRAM 29B H	CRA6 DISP SEL 1B H	C12R2
CRA6 CRAM 28A H	CRA6 DISP SEL 2A H	C12F2
CRA6 CRAM 28B H	CRA6 DISP SEL 2B H	C12J2
CRA6 CRAM 27A H	CRA6 DISP SEL 4A H	B12N2

CRA6 CRAM 27B H	CRA6 DISP SEL 4B H	B12R2
CRM2 BIT 52 H	CRM2 DIVIDE H	C11E1
CRM2 BIT 83 H	CRM2 DP A ADR 01 H	F11S1
CRM2 BIT 82 H	CRM2 DP A ADR 02 H	F11J1
CRM2 BIT 81 H	CRM2 DP A ADR 04 H	E11S1
CRM2 BIT 80 H	CRM2 DP A ADR 10 H	E11J1
CRM2 BIT 77 H	CRM2 DP B ADR 01 H	C11S1
CRM2 BIT 76 H	CRM2 DP B ADR 02 H	C11J1
CRM2 BIT 75 H	CRM2 DP B ADR 04 H	B11S1
CRM2 BIT 74 H	CRM2 DP B ADR 10 H	B11J1
CRM2 BIT 91 H	CRM2 FE EN H	D11S2
CRM2 BIT 62 H	CRM2 FUNC 01 H	B11H2
CRM2 BIT 61 H	CRM2 FUNC 02 H	A11R2
CRM2 BIT 60 H	CRM2 FUNC 04 H	A11H2
CRA6 CRAM 00 H	CRA6 J 00 H	NC
CRA6 CRAM 01 H	CRA6 J 01 H	NC
CRA6 CRAM 02 H	CRA6 J 02 H	NC
CRA6 CRAM 03 H	CRA6 J 03 H	NC
CRA6 CRAM 04 H	CRA6 J 04 H	NC
CRA6 CRAM 05 H	CRA6 J 05 H	NC
CRA6 CRAM 06 H	CRA6 J 06 H	NC
CRA6 CRAM 07 H	CRA6 J 07 H	NC
CRA6 CRAM 08 H	CRA6 J 08 H	NC
CRA6 CRAM 09 H	CRA6 J 09 H	NC
CRA6 CRAM 10 H	CRA6 J 10 H	NC
CRA6 CRAM 11 H	CRA6 J 11 H	NC
CRM2 BIT 95 H	CRM2 MARK BIT H	F11S2
CRA6 CRAM 26A H	CRA6 MEM FUNCTION A H	B12F2
CRA6 CRAM 26B H	CRA6 MEM FUNCTION B H	B12J2
CRM2 BIT 53 H	CRM2 MULTI PRECISION H	C11N1
CRM2 BIT 92 H	CRM2 PARITY CHECK LEFT H	E11J2
CRM2 BIT 93 H	CRM2 PARITY CHECK RIGHT H	E11S2
CRM2 BIT 50 H	CRM2 PARITY INH LEFT H	B11E1
CRM2 BIT 51 H	CRM2 PARITY INH RIGHT H	B11N1
CRM2 BIT 86 H	CRM2 RAM ADR 01 H	B11J2
CRM2 BIT 85 H	CRM2 RAM ADR 02 H	A11S2
CRM2 BIT 84 H	CRM2 RAM ADR 04 H	A11J
CRM2 BIT 48 H	CRM2 RAMFILE WRITE H	A11E1
CRM2 BIT 90 H	CRM2 SC EN H	D11J2
CRA6 CRAM 17 H	CRA6 SKIP EN 10 L	C12M2
CRA6 CRAM 16 H	CRA6 SKIP EN 20 L	C12E2
CRA6 CRAM 15 H	CRA6 SKIP EN 40 L	B12M2
CRA6 CRAM 35A H	CRA6 SKIP SEL 1A H	C12S1
CRA6 CRAM 35B H	CRA6 SKIP SEL 1B H	C12S2
CRA6 CRAM 34A H	CRA6 SKIP SEL 2A H	C12K1
CRA6 CRAM 34B H	CRA6 SKIP SEL 2B H	C12K2
CRA6 CRAM 33A H	CRA6 SKIP SEL 4A H	B12S1
CRA6 CRAM 33B H	CRA6 SKIP SEL 4B H	B12S2
CRM2 BIT 65 H	CRM2 SOURCE L 01 H	C11R2
CRM2 BIT 64 H	CRM2 SOURCE L 02 H	C11H2
CRM2 BIT 63 H	CRM2 SOURCE L 04 H	B11R2
CRM2 BIT 68 H	CRM2 SOURCE R 01 H	E11M2
CRM2 BIT 67 H	CRM2 SOURCE R 02 H	D11R2
CRM2 BIT 66 H	CRM2 SOURCE R 04 H	D11H2
CRM2 BIT 49 H	CRM2 SPARE 49 H	A11N1

```
CRA6 CRAM  20 H         CRA6 SPEC EN 10 H                B12F1
CRA6 CRAM  19 H         CRA6 SPEC EN 20 H                A12N1
CRA6 CRAM  18 H         CRA6 SPEC EN 40 H                A12F1
CRA6 CRAM 32A H         CRA6 SPEC SEL 1A H               B12K1
CRA6 CRAM 32B H         CRA6 SPEC SEL 1B H               B12K2
CRA6 CRAM 31A H         CRA6 SPEC SEL 2A H               A12S1
CRA6 CRAM 31B H         CRA6 SPEC SEL 2B H               A12S2
CRA6 CRAM 30A H         CRA6 SPEC SEL 4A H               A12K1
CRA6 CRAM 30B H         CRA6 SPEC SEL 4B H               A12K2
CRA6 CRAM  12 H         CRA6 T00 H                       A12E2
CRA6 CRAM  13 H         CRA6 T01 H                       A12M2
```

CHAPTER 5

DEBUGGING

1. TYPE-OUT MODES

$S	SYMBOLIC INSTRUCTIONS
$C	NUMERIC
$F	FLOATING POINT
$T	ASCII TEXT
$6T	SIXBIT TEXT
$5T	RADIX50
$H	HALFWORDS
$NO	BYTES

2. ADDRESS MODES

$R	RELATIVE TO SYMBOLIC ADDRESS
$A	ABSOLUTE NUMERIC ADDRESS

3. RADIX CHANGE

$NR	N=RADIX

4. EXAMINING STORAGE WORDS

ADR/	OPEN AND EXAMINE
ADR!	OPEN BUT INHIBIT TYPEOUT
ADR[	OPEN AND EXAMINE AS A NUMBER
ADR]	OPEN AND EXAMINE AS SYMBOLIC INSTRUCTION
;	RETYPE LAST QUANTITY

5. RELATED STORAGE WORDS

^	EXAMINE ADR-1
TAB	EXAMINE LOCATION SPECIFIED BY ADDRESS
\	EXAMINE LOCATION SPECIFIED BY ADDRESS BUT DON'T CHANGE POINTER
CR	CLOSE CURRENTLY OPEN LOCATION

6. RETYPING IN MODES OTHER THAN PREVAILING OR TEMPORARY

=	REPEAT LAST TYPEOUT AS A NUMBER
-	REPEAT LAST TYPEOUT AS A SYMBOLIC INSTRUCTION
/	TYPEOUT LOCATION POINTED TO BUT DON'T CHANGE POINTER
[	TYPEOUT LOCATION POINTED TO AS A NUMBER
]	TYPEOUT LOCATION POINTED TO AS A SYMBOLIC INSTRUCTION
LF	EXAMINE ADR+1

7. TYPING IN

INST	TYPE IN SYMBOLIC INSTRUCTION
#,,#	TYPE IN HALF WORDS
#	TYPE IN OCTAL
#.	TYPE IN DECIMAL
#.#	TYPE IN FLOATING POINT
"/A/	TYPE IN ASCII
"A$	TYPE IN ONE ASCII CHAR
$"/A/	TYPE IN SIXBIT
$"A$	TYPE IN ONE SIXBIT CHAR

8. SYMBOLS

NAME$:	OPEN PROGRAM SYMBOL TABLE
N<SYM:	INSERT SYMBOL WITH VALUE N
SYM:	INSERT SYMBOL WITH VALUE OF LOCATION POINTER
SYM$$K	DELETE A SYMBOL FROM SYMBOL TABLE
SYM$K	KILL A SYMBOL FOR TYPEOUT
$D	PERFORM $K ON LAST SYMBOL TYPED OUT
SYM#	DECLARE A SYMBOL WHOSE VALUE IS TO BE DEFINED LATER
?	TYPE OUT A LIST OF UNDEFINED SYMBOLS

9. SPECIAL DDT SYMBOLS

.	REPRESENTS THE ADDRESS OF THE LOCATION POINTER
$Q	REPRESENTS THE LAST QUANTITY TYPED
$$Q	REPRESENTS THE LAST QUANTITY TYPED, HALVES REVERSED
@	THE INDIRECT BIT
$M	THE ADDRESS OF THE SEARCH MASK REGISTER
$I	THE ADDRESS OF THE SAVED FLAGS, ETC.
$NB	THE POINTERS ASSOCIATED WITH THE NTH BREAKPOINT

10. ARITHMETIC OPERATORS

+	ADDITION
-	SUBTRACTION
*	MULTIPLICATION
'	DIVISION

```
11.     FIELD DELIMITERS IN SYMBOLIC TYPE IN

SPACE   DELIMITS OP-CODE FIELDS
,       DELIMITS ACCUMULATOR FIELD
L,,R    DELIMIT HALF WORDS
()      DELIMIT INDEX REGISTER
@       INDICATE INDIRECT ADDRESSING

12.     BREAKPOINTS

ADR$NB  SET A SPECIFIC BREAKPOINT
ADR$B   SET THE NEXT BREAKPOINT
ADR$$B  SET A BREAKPOINT WITH AUTO PROCEED
X,,ADR$B SET A BREAKPOINT WHICH WILL PRINT ADDRESS X
0$NB    REMOVE A SPECIFIC BREAKPOINT
$B      REMOVE ALL BREAKPOINTS
$NB/    CHECK THE STATUS OF BREAKPOINT N
$P      PROCEED FROM A BREAKPOINT
N$P     SET THE PROCEED COUNTER AND PROCEED
$$P     PROCEED ALWAYS

13.     CONDITIONAL BREAKPOINTS

$NB+1/INST  INSERT A CONDITIONAL BREAKPOINT

14.     STARTING THE PROGRAM

$G      START A STARTING ADDRESS IN JOBSA
ADR$G   START AT SPECIFIED ADDRESS
INST$X  EXECUTE AN INSTRUCTION

15.     SEARCHING

A<B>C$W  SET LOWER (A), SET UPPER (B), SEARCH FOR WORD (C)
A<B>C$N  SEARCH FOR A NOT-WORD
A<B>C$E  SEARCH FOR AN EFFECTIVE ADDRESS
$M/     EXAMINE MASK USED IN SEARCHES
N$M     INSERT ANOTHER QUANTITY IN MASK

16.     ZEROING MEMORY

FIRST<LAST$$Z  ZERO MEMORY FIRST THRU LAST
```

```
17.     SINGLE STEP EXECUTE

$X      EXECUTE A SINGLE INSTRUCTION, THEN INCREMENT THE PC
        THE OPERANDS ARE PRINTED AFTER EXECUTION

N$X     REPEAT THE $X CYCLE N TIMES

N$$X    SAME AS N$X EXCEPT PRINTING OCCURS ONLY FOR LAST CYCLE

$$X     PERFORM A NON-PRINTING $X CYCLE UNTIL THE PC REACHES EITHER
        .+1 OR .+2  ;USED FOR TREATING SUBROUTINE CALLS AS A SINGLE
        INSTRUCTION.

18.     PATCHING A PROGRAM

ADR/CONTENTS    $<
PATCH/  NEW INST
PATCH+1/ NEW INST$>
PATCH+2/ CONTENTS
PATCH+3/ JUMPA 1,ADR+1
PATCH+4/ JUMPA 2,ADR+2
```

A PATCH IS MADE BY OPENING AN ADDRESS, TYPING (ALTMODE)(ANGLE-BRACKET), THIS SAVES THE CURRENT CONTENTS OF THE ADDRESS, OPENS THE PATCH AREA FOR NEW INSTRUCTIONS, AFTER THE NEW INSTRUCTIONS ARE ENTERED THE PATCHING IS CLOSED BY TYPING (ALTMODE)(ANGLE-BRACKET). THE ORIGINAL CONTENTS ARE THEN PLACED IN THE PATCH AREA AND TWO JUMP INSTRUCTIONS PLACED FOLLOWING WHICH WILL RETURN TO THE ORIGINAL ADDRESS +1 OR +2 DEPENDING ON WHETHER THE LAST INSTRUCTION IN THE PATCH SKIPS OR NOT.

```
19.     SPECIAL EDITING CHARACTERS

RUBOUT  DELETE LAST TYPED CHARACTER
^U      (CONTROL U) DELETE LINE
^W      (CONTROL W) DELETE LAST WORD, BACK TO DELIMITER
^R      (CONTROL R) RETYPE LAST LINE
```

```
Locations useful to know:

10000   Start Adress of SMDDT  ;  20000 SMMON/SMMAG Start Address
30000   Start Address of all Diagnostics
1000    Start Address of PRE-Boot

Highest Core-Adress - 1 page ==> Start Address of TS-Boots ( BOOT> )

Monitor "Hide-outs":

BOOT>/L         ; to only load monitor
BOOT>/G140      ; to start loaded monitor at EDDT
BOOT>/D         ; to dump core-image of crashed monitor

EDDTF/0         ;normal         1 to keep EDDT resident for debugging
DBUGSW/0        ;normal         1 to stop at BUGHLT     2 to use SYJOB.DEBUG
                ;                                         "no timesharing"
                ;                                        and to write-enable
                ;                                        swappable monitor
DCHKSW/0        ;normal         1 to stop at BUGCHK

GOTSWM$B        ;to set DDT-breakpoint just after swappable MONITOR came in
                ; 80% of all hardware used ( context switching and paging)
CALL SWPMLK$X   ; to lock swappable MONITOR in core ( more than 128 K MOS !!)
SYSGO1$G        ; Normal startup of MONITOR via EDDT
MONCOR          ; Nr. of pages of resident MONITOR ( excluding Data-Base )
SWPCP0          ; first page used for swapping
MMAP            ; Page map , if bit 14-17==0 , RH is Page-NR in core
FORKX           ; Nr. of fork last ( currently ) running
                ; -1 if in scheduler
JOBNO ( in PSB) ; JOB Nr. to which current fork belongs
FKJOB + Fork Nr.; JOB NR. ,, SPT Index into JSB
JOBDIR+ JOB Nr. ; Conn Dir,, Logged in Dir Nr.
JOBPT + JOB Nr. ; Controling TTY Nr ,, Top Fork Nr.
FKSTAT+ Fork NR.; Fork-Nr ,, Address of Fork-Wait Routine
FKPGS + Fork Nr.; SPT Index of Page table ,, SPT Index of PSB

BUGHLT          ; Adress where BUGHLT-Routine was called
BUGCHK          ; same for BUGCHK
BUGACS          ; AC's are stored till BUGACS + 17 here
SEBQOU          ; Start of incore syserr-buffer
SEBDAT+BG%ACS   ; offset from SEBQOU to saved AC's ( only first entry ! )
TRAPPC          ; MONITOR PC at TRAP

$M/ -1  -1,,0
FKJOB<FKJOB+NFKS>x,,0$W ; finds all forks for job x

                ; PSB-goodies
UAC             ; Users AC's at last JSYS-time
PAC             ; Monitor's AC's ....
PPC             ; users PC
UPDL            ; users push down stack
```

```
^EQUI
MX>/            ; gets into MDDT ( normally not resident )

^P or ^Z        ; gets out of it  ( back to MX> )
                ; S or E brings YOU back to EXEC out of which YOU "QUITTED"
                ; by the way -- from there on ^P acts as a switch , the next
                ; one brings YOU back to MDDT ( sometimes good to know if all
                ; else fails !! ( valid till YOU log off )

n$U             ; sets SPT Index n to be used for virtual addressing ( FILDDT )

CALL SWPMLK$X   ; locks swappable MONITOR in core for debugging ( => 128 K MOS )
CALL SWPMWE$X   ; write-enables swappable MONITOR for patching
CALL SWPMWP$X   ; write-protects swappable MONITOR ( normal case )
FFF             ; Start of Patch-area for resident MONITOR ( used and updated
                ;                                               by hand )
PAT.. or SWPF   ; Start of Patch-area for swappabel MONITOR ( auto - patching )
143             ; MONITOR start to redefine Structures , rewrite BAD-blocks etc.
146             ; RELOAD and RESTART
141             ; RESET and go to EDDT
145             ; Soft RESTART ( reset I/O devices and "try" to resume jobs )??
```

CHAPTER 6

DESCRIPTIONS

Arbitration of the KS10 bus occurs on print CSL1. A BUS MASTER always has control of the bus for at least two T-clock cycles. The arbitrator will grant the bus to the requesting device of the highest priority unless

1. MEMORY BUSY is asserted , in which case the arbitrator is disabled until the operation is completed.

2. The current cycle is a COMMAND ADDRESS cycle

3. The cycle immediately following a COMMAND ADDRESS CYCLE.

CONSOLE-DATA-REG

A CPU command address IO READ 200000 will read the contents of CSL IO registers 102, 104, 106, 110 and 112 . These registers (8 bits each) contain the last instruction executed from the console or data for a DATA CYCLE. This feature is used , when starting up the OPERATING SYSTEM . The 8080 code will write a JRST ADDR in the DATA register and issue EXECUTE and CONTINUE. The KS10 executes the JRST to the start of the bOOT program and continues to boot .

The decoding of the command address data lines occurs on CSL4 and sets
CSL4 IO READ which generates CSL$ DATA CYCLE. This signal opens the read
gate on the 4x4 register memories which precede the 8646 transceivers. At the
next T-clock the contents of 102 , 103 , 104 , 106 , 110 , 112 will appear
on the BUS with the IO DATA CYCLE.

INTERRUPTS

KS10 can interrupt the CONSOLE by asserting DPMB CSL INTERRUPT L
Pin=F18L1

The CONSOLE can interrupt the KS10 by asserting CSL4 INT 10 L
Pin=F18T2

KS10 BUS
ARITHMETIC DATA PATH
MICRO STORE
MEMORY
RAMFILE (CACHE , AC'S , WORKSPACE)
CACHE DIRECTORY
PAGE TABLE
UBA PAGING RAM

1. Parity is generated for bus data both left and right halves by the 8646 transceivers , whenever data is sent out on the bus . Parity is checked by the system unit receiving the data.

2. Ramfile (Cache , AC's , workspace) parity is generated at the output of the D-bus mixers on DPE3 , DPE4.

3. Arithmetic data path parity is checked at the output of the D-bus mixers on DPE3 and DPE4.

4. The arithmetic parity bit (for destination registers in the 2901) is stored in the 16x4 K RAM on DPE4. The m-code has the option of storing and checking the even data path parity.

5. Even directory parity is generated at the input to the cache directory from the VMA bits on DPM7.

6. Cache directory parity is checked on DPM7 -- if it is even the error will go to CSL to stop the clock.

7. Even page table parity is generated at the input to the page table whenever the page table is written from DP.

8. Page table parity is checked at the output on DPM6 -- if it is even , it will go to CSL to stop the clock.

9. Microcode parity is written into the control-RAMs , when they are loaded. It is checked whenever a C-RAM word is read. Even parity is stored into the C-RAMs. If it is odd on the read for either the CRA or CRM boards , a parity error will result in stopping the CPU clock.

10. UBA paging ram parity (odd) is generated on UBA5 whenever the ram is written.

11. UBA paging ram parity is checked , whenever the paging ram is accessed such as a UNIBUS NPR transfer -- the parity MUST be odd and the VALID bit true BEFORE a KS10 bus transfer is allowed.

A cache hit can be generated on a read from memory. A fetch from memory will occur , when the CPU wants to fetch an instruction , fetch an argument needed for an instruction execution or on a page table refill (occurs when page is invalid). The memory read cycle is initiated by the micro-code instruction which contains either the "start read" or "fetch" macro. the VMA should be loaded at this time or has been loaded previously. When the macro is used , the micro-code will have CRA6 memory function and CRM2 #16 set within the instruction. The other way of starting the cycle is by a "AREAD DISP". At AREAD DISP time , which occurs during EA CALC the signal DPEA DROM COND FUNC out of the DROM must be generated by the OP code of the instruction being executed. CRA6 MEM FUNC and CRM2 #17 will also be set within the m-code instruction doing the AREAD DISP. Referring to DPM5 , the result of starting the cycle in either fashion is that DPM5 READ DELAY EN is produced. The last 75 ns of the m-instruction that generated DPM5 MEM EN , will , with DPM5 SYNC E L , set DPM5 START CYCLE. On the next T-clock , because of DPM5 START CYCLE -- DPM5 READ DELAY EN , DPM6 MEM CYCLE and DPM5 REQU CPU will be set. DPM5 BUS REQU goes to the arbitration logic on the console board (CSL2). If the requested data is in CACHE , a cache hit will occur at this time.

Assuming the desired argument is in cache , the location being addressed has an entry in the page-table consisting of the physical page number , VALID bit and CACHEABLE bit both set. The hardware sees the page is valid and checks the cache directory (addressed by the virtual line #) to see , if the line is in cache. The cache directory caontains VALID , PARITY and USER bits in addition to the virtual page number for that line. If the outputs of the cache directory match the virtual page address , the entry is valid andif the parity is good (on the virtual page address bits) DPM7 CACHE HIT EN will be true.

!*NOTE:SINCE A HIT DID OCCUR WE KNOW THAT THE DATA IN CACHE HAS BEEN FETCHED BEFORE.WE ALSO KNOW THAT IT IS THE LINE THAT WE NEED(I.E.THE SAME LINE NUMBER FROM DIFFERENT PAGES CANNOT COEXIST IN CACHE).THIS IS SO BECAUSE WE ADDRESS THE CACHE DIRECTORY WITH LINE # ADDRESS BITS.THIS IS A DEPARTURE FROM THE KL10 CACHE WHICH IS A FOUR PART STRUCTURE.

DPM6: VMA18-26 ADDRESS THE PAGE TABLE YIELDING OUTPUTS PAGE VALID,WRITEABLE, CACHEABLE,USER,PARITY, AND PHYSICAL PAGE ADDRESS BITS 16-26.

DPM7: VMA 27-35 ADDRESS THE CACHE DIRECTORY YIELDING OUTPUTS CACHE VALID, USER,PARITY, AND VIRTUAL PAGE ADDRESS BITS(AS CACHE 18-26). NAND GATE AT E513 WILL GENERATE CACHE HIT EN IF THE FOLLOWING COND- ITIONS ARE MET:THERE ARE NO PAGE FAILS,PAGING IS ENABLED,WE ARE DOING A READ CYCLE,THE CACHE IN NOT INHIBITED,THE ADDRESS IS CACHEABLE, AND THE PAGE NUMBER IN THE CACHE DIRECTORY MATCHES THE VIRTUAL ADDRESS WE WANT TO REFERENCE. AS LONG AS THE VMA IS NOT BEING LOADED THE OUTPUT DPM7 CACHE HIT WILL BE TRUE.

DPM5: THE MEM CYCLE ALREADY BEGUN MUST BE ABORTED.CACHE HIT EN CLEARS THE KS10 BUS REQUEST.CACHE HIT GENERATES DPM5 STOP MAIN MEMORY.THESE GO TO DPMC TO ABORT THE CYCLE AS SOON AS POSSIBLE AND GENERATE MEM CYCLE ABORT TO THE MEMORY CONTROLLER.

PREVIOUSLY, WHEN DPM4 VMA WAS LOADED FROM DP, ANOTHER COPY WAS LOADED ON DPE5 VIA DP 26-35.THESE ADDRESS THE CACHE DIRECTORY. THE MICROCODE IS SELECTING VMA 26-35 AS THE INPUTS TO THE RAMFILE ADDRESS MIXER AT THIS TIME, ANTICIPATING A CACHE HIT.RAMFILE ADR 0-11 WILL ADDRESS THE CACHE ON DPE7 AND DPE8, YIELDING RAMFILE 00-35. DPM7 CACHE HIT EN GENERATED DPM5 FORCE RAMFILE WHICH FORCES THE D-BUS SELECT SIGNALS ON DPE3 TO GATE RAMFILE 00-35 ,RATHER THAN MEMORY DATA (DBM), ON TO THE D-BUS.THE D-BUS DATA 00-35 IS INPUT TO THE 2901'S ON DPE1 & DPE2 AND STROBED INTO THE 2901 REGISTER OR AC SELECTED BY THE MICROCODE INSTRUCTION DOING THE "MEM READ".

```
COMMAND:DB XX (36 BIT DATA ARGUMENT) DEPOSIT DATA PATTERN TO KS10 BUS
DESC:   THIS COMMAND PERFORMS AN INTERNAL LOOPBACK FUNCTION TO CHECK
        THE ABILITY OF THE CONSOLE BOARD TO WRITE DATA THRU THE KS10
        BUS TRANSCIEVERS AND READ IT BACK WITHOUT ERRORS.THE COMMAND
        IS SPLIT INTO TWO PARTS WHICH WRITE AND READ THE ODD AND EVEN
        IO REGISTERS LOCATED ON CSL 6,7,8.THE FIRST PART WRITES THE ODD
        REGISTERS NORMALLY USED FOR COMMAND ADDRESS CYCLES,READ THE DATA
        BACK AND COMPARES IT TO THE DATA SENT.THE SECOND PART WRITES THE EVEN
        REGISTERS NORMALLY USED FOR DATA ARGUMENTS(AND FOR A JRST INSTRUCTION
        TO START KS10) ,DOES THE READ AND COMPARES.THE READ OF BUS DATA
        OCCURS THRU THE D BUS MIXERS ON CSLA PRINT.THESE RETURN INVERTED
        DATA TO THE DBUS. A FAILURE IN THE FIRST PART WILL YIELD A ?C CYC
        ERROR CODE.A FAILURE IN THE SECOND PART WILL YIELD A ?D CYC
        ERROR CODE.
```

```
COMMAND:DC XX(96 BITS) INTO ADR SPECIFIED BY PREVIOUS EC,LC COMMAND
        DF XX(12 BITS) INTO ADR SPEC BY PREVIOUS EC,LC,LF
DESC:   CRA CONNECTS TO LEAST SIGNIFICANT 12 BITS OF KS10 BUS.THE CONSOLE
        CAN DEP/EXAMINE THE CRAM REGISTER,CURRENT LOCATION,THE TOP
        OF THE SUBRTN STACK AND THE CRAM ADDRESS LINES.THE SELECTION
        OF THE 12 BIT SEGMENT TO BE READ/WRITTEN IS DONE BY
        CSL4 DIAG 4,2,1 H.  CSL4 DIAG 10 H DETERMINES WHETHER CRA OR
        CRM IS INVOLVED.36 BITS MICRO STORE IS ON CRA,60 BITS ON CRM.
        A BUFFERED COPY OF THE 12 BITS IS AVAILABLE TO CRM FROM CRA5.
        BUS DATA 21,22,23 ARE DRIVEN BY THE PARITY OF THE 12 BITS TO
        MAINTAIN EVEN PARITY ON THE KS10 BUS.THE DF COMMAND IS
        PROBABLY THE BETTER TROUBLESHOOTING AID BECAUSE THE PREVIOUS
        LF COMMAND WILL DEFINE KNOWN DIAG BITS AND 12 BIT DESTINATION
        IN MICRO STORE.
```

```
COMMAND: DI XX(16 BITS OF DATA) INTO A REGISTER ADDRESS ALREADY SPECIFIED
        BY A PREVIOUS LI OR EI COMMAND
DESCRIPTION:THIS COMMAND WILL LOAD DATA TO A PLACE SPECIFIED BY ADDRESS
        GIVEN IN A PREVIOUS LI OR EI COMMAND.
        BITS 14-17 WHICH INDICATE A CONTROLLER #,AND BITS 18-35 WHICH
        INDICATE A DEVICE ADDRESS.LOCATION 100000 IS THE MEMORY STATUS
        REGISTER.ALL OTHER ADDRESSES
```

```
CONDITIONS:THE DESIRED ADDRESS TO BE WRITTEN MUST HAVE BEEN LOADED
        BY A PREVIOUS LA OR EM COMMAND
DESC:STORE THE ADDRESS AND DATA INTO 8080 IO REGISTERS.SET THE
        APPROPRIATE BITS FOR THE COMMAND/ADDRESS CYCLE AND DATA CYCLE
        AND DO IT BY ENABLING T CLKS.
```

```
DESC:   THE CONSOLE MUST ISSUE A COMMAND ADDRESS CYCLE ON THE KS10
        BUS WITH THE MEMORY ADDRESS.IT THEN WAITS FOR MEMORY TO
        RESPOND WITH DATA AND THE SIGNAL DATA CYCLE WHICH LATCHES
        THE DATA IN THE RECIEVERS.
```

```
COMMAND:EX XX(36 BITS)
DESC:THE COMMAND EXECUTES A 10 INSTRUCTION.THE MICROCODE MUST BE STARTED
        AND SITTING IN THE HALT LOOP.THE CPU ISSUES AN IO READ TO THE
        CONSOLE IN RESPONSE TO EXECUTE AND CONTINUE.THE INSTRUCTION IS
        PASSED TO THE CPU OVER THE KS10 BUS AND THE MICROCODE EXECUTES IT.
```

```
COMMAND:EC XXXX(0-3777) OR EC
DESC:PRINTS THE CONTENTS OF LOCATION XXXX IN MICRO STORE
        IF NO ARGUMENT,LAST CRAM ADDRESS EXAMINED OR DEPOSITED
        OR ADDRESS OF LAST LC COMMAND
```

CHAPTER 7

UBA-RH11 DESCRIPTION OF READ/WRITE

UBA PAGER

THE KS10 SETS UP THE PAGER BY ADDRESSING 7630000-77 AND THE DATA WILL BE THE PAGE NUMBER (ADDRESS BITS 16-26). WHEN THE RH11-C SENDS AN ADDRESS ON UNIBUS BITS 17-0,BITS 16-11 WILL ADDRESS THE PAGING RAM(VIA THE MIXER TO BECOME 30-35). ITS OUTPUT SHALL BE 16-26 OR PAGE # AND UNIBUS BITS 10-2 (10 DEFINE HALF WORD AND BYTE) DEFINE THE LINE #. BITS 10-2 FEED BITS 27-35 OF THE 10 ADDRESS.THIS WILL ALLOCATE 777 OR 512.(DECIMAL) WORDS.

THE UBA REGISTER READ

THE KS10 WILL FETCH AN I/O INSTRUCTION IN THE FOLLOWING FORMAT 712760,001000,WHERE THE AC 17 WILL HOLD THE REGISTER DATA TO BE READ AND INDIRECT ADDRESS 1000 WILL CONTAIN 1,,763000-763100.IF THE ADDRESS IS 763000-763077 THE PAGING RAM WILL HAVE ONE OF ITS 64.LOCATIONS SELECTED.IF THE ADDRESS IS 763100 THE STATUS REGISTER OF THE UBA WILL BE SELECTED. AFTER THE PROCESSOR HAS FETCHED THE EFFECTIVE ADDRESS IT WILL ISSUE A BUS REQUEST TO THE CONSOLE.IF THE CONSOLE IS NOT ARBITRATING AN OTHER REQUEST IT WILL ISSUE BUS GRANT TO THE KS10.THE KS10 WILL NOW ISSUE A COMMAND/ADDRESS CYCLE STROBING D0-35 TO THE UBA IN THE FOLLOWING FORMAT:

D0=1 DEFINING I/O
D1=1 DEFINING READ
D14-17 DEFINING THE UBA#1
D18-35 DEFINING THE REGISTER SELECTED(763000-763100)

THE UBA UPON RECEIVING THE C/A CYCLE DECODE D1 DETERMINE THAT IT IS NOT AN EXTERNAL REGISTER AND TAKE THE DATA AND LOAD IT INTO THE TRANCEIVERS.THE UBA WILL NOW ISSUE A BUS REQUEST TO THE CONSOLE (THIS IS DONE BECAUSE THE KS10 DOES NOT KNOW HOW LONG IT WILL TAKE TO ACCESS THE DATA).THE CONSOLE WILL RESPOND WITH A BUS GRANT AND THE UBA WILL NOW HAVE CONTROL OF THE BUS.THE UBA WILL ISSUE AN I/O DATA CYCLE STROBING D0-35 TO THE KS10.THE KS10 WILL NOW LOAD THE CONTENTS OF D0-35 INTO AC 17.***IT SHOULD BE REALIZED THAT THE ONLY TIME I/O DATA CYCLE IS TRUE IS WHEN THE KS10 IS READING A REGISTER.

UBA REGISTER WRITE

THE KS10 WILL FETCH AN INSTRUCTION OF 713760,001000 WHERE THE AC 17 WILL CONTAIN IN BITS 20-35 THE REGISTER DATA TO BE WRITTEN AND INDIRECT ADDRESS 1000 WILL CONTAIN 1,,763000-763100 THE ADDRESS TO SELECT THE UBA REGISTER.THE KS10 WILL NOW ISSUE A BUS REQUEST TO THE CONSOLE.IF THE CONSOLE IS NOT SERVICING ANOTHER DEVICE IT WILL RETURN BUS GRANT AND THE KS WILL NOW HAVE CONTROL OF THE BUS.A COMMAND/ADDRESS CYCLE WILL STROBE D0-35 TO THE UBA IN THE FOLLOWING FORMAT:

D0=1 I/O
D2=1 WRITE DATA
D14-17 UBA#1
D18-35 ADDRESS OF THE REGISTER(763000-763100)

THE UBA WILL RESPOND TO THE C/A CYCLE BY OPENING THE ADDRESSED REGISTER. THE KS10 WILL NOW ISSUE A DATA CYCLE WHERE D0-35 WILL CONTAIN THE CONTENTS OF AC17(REGISTER DATA).THE UBA WILL TAKE DATA CYCLE AND STROBE THE DATA INTO THE SELECTED REGISTER. RH11-C REGISTER WRITE

THE KS10 WILL FETCH AN INSTRUCTION IN THE FOLLOWING FORMAT 713760001000. WHERE 713 WILL DEFINE AN I/O WRITE AC 17 WILL CONTAIN THE CONTENTS OF THE REGISTER DATA TO BE WRITTEN AND THE INDIRECT ADDRESS OF 1000 WILL CONTAIN 1,,7767XX.THE KS10 WILL REQUEST THE BUS FROM THE CONSOLE AND WITH THE RECEIPT OF BUS GRANT ISSUE A COMMAND/ADDRESS CYCLE TO THE BUS STROBING D0-35+2P.D0-35+2P WILL BE IN THE FOLLOWING FORMAT:

D0=1 I/O
D2=1 WRITE
D14-17=UBA#1
D18-35=RH11-C ADDRESS(7767XX)

THE UBA UPON RECEIPT OF THE C/A CYCLE WILL DETERMINE IF THE REGISTER IS INTERNAL OR EXTERNAL TO THE UBA AND GATE THE ADDRESS TO THE BUS ADDRESS TRANSMITTERS OF THE UNIBUS.D2 WILL GATE THROUGH THE UBA AND BECOME C1 ON THE UNIBUS. THE KS10 WILL NOW ISSUE A DATA CYCLE STROBING D0-35+2P TO THE UBA WITH D20-35 CONTAINING THE DATA TO BE WRITTEN INTO THE REGISTER.THE UBA WILL GATE THE BITS 18-35 TO THE TRANSCEIVERS AND THROUGH THE 4X4 MEMORY(LOC.1)TO THE UNIBUS.THE UBA ARBITRATOR WILL NOW GENERATE ADDR TO BUS AND MSYNC WILL NOW STROBE ADDRESS LINES 17-0(7767XX),C1,AND DATA 15-0(REGISTER DATA)TO THE RH11-C.THE MSYNC SIGNAL WILL ALSO PULL BBSY LOW.THE RH11-C WILL TAKE THE ADDRESS LINES AND DECODE THE BITS 17-6 TO DEFINE THIS RH11-C AND DECODE BITS 5-1 TO DETERMINE IF THE REGISTER IS LOC/REMOTE (INTERNAL OR EXTERNAL) AND WHICH ONE.WITH THE REGISTER DECODED C1 AND THE REGISTER # WILL ENABLE THE D15-0 DATA BITS TO BE LOADED INTO THE THE SELECTED REGISTER.THE RH11-C WILL NOW ISSUE SSYNC BACK TO THE UBA AND TERMINATE THE OPERATION.

RH11-C REGISTER READ

IN ORDER TO SELECT A REGISTER IN THE RH11-C THE ADDRESS MUST BE A 7767XX FOR THE RH CONNECTED TO THE RM03 7724XX FOR THE RH

CONNECTED TO THE TMO2/TU45.

THE KS10 PROCESSOR WILL FETCH AN INSTRUCTION IN THE FOLLOWING FORMAT 712760,001000 WHERE 712 WILL DEFINE AN I/O READ,AC17 WILL BE WHERE REGISTER DATA WILL BE STORED,AND INDIRECT ADDRESS 1000 WILL HOLD THE FOLLOWING 1,,776700.THE KS10 WILL ISSUE A BUS REQUEST TO THE CONSOLE BOARD WHICH WILL RESPOND WITH BUS GRANT.THE KS10 WILL NOW ISSUE A COMMAND/ADDRESS CYCLE STROBING D0-35 IN THE FOLLOWING FORMAT:

D0=1 I/O READ
D1=1 READ
D14-17 UBA#1
D18-35 RH11-C CONTROL REG

THE UBA WILL RECEIVE THE C/A CYCLE AND GATE THE D18-35 UP TO THE UNIBUS ADDRESS TRANSMITTERS AND CAUSE D1 TO RESET C1 OFTHE UNIBUS. THE UBA WILL NOW CAUSE ADDRESS TO BUS TO GO TRUE AND PULL BBSY LOW AND ISSUE MSYNC STROBING ADDR 17-0(776700) AND C1=0 TO THE RH11-C. THE RH11-C WILL DECODE ADDR. BITS 17-6 TO DEFINE THE RH11-C AND BITS 5-1 TO FEED A ROM TO SELECT THE REGISTER AT MSYNC TIME.THE RH11-C WILL PLACE THE CONTENTS OF THE CS1 REG ON D15-0 AND STROBE IT TO THE UBA WITH SSYNC.THE UBA WITH SSYNC WILL CAUSE A BUS REQUEST TO THE CONSOLE. WITH BUS GRANT RECEIVED THE UBA WILL ISSUE I/O DATA CYCLE AND STROBE BACK THE CONTENTS OF CS1 TO AC 17.

RH11-C REGISTER WRITE

THE KS10 WILL FETCH AN INSTRUCTION IN THE FOLLOWING FORMAT 713760,,001000,WHERE 713 WILL DEFINE I/O WRITE AND AC 17 WILL CONTAIN THE REGISTER DATA.INDIRECT ADDRESS 1000 WILL CONTAIN 1,,7767XX SPECIFYING UBA#1 AND RH11-C REGISTER XX. THE KS10 WILL REQUEST THE BUS FROM THE CONSOLE AND WAIT FOR BUS GRANT.IT IS AT THIS TIME THAT THE KS10 WILL ISSUE A COMMAND/ ADDRESS CYCLE WITH D0-35+2P IN THE FOLLOWING FORMAT:

D0=1 I/O
D2=1 WRITE
D14-17= UBA#1
D18-35=RH11-C REGISTER XX(7767XX)

THE UBA WILL TAKE THE DATA OFF THE KS10 BUS AND DECODE THAT IT IS AN EXTERNAL REGISTER AND GATE THE ADDRESS 7767XX TO ADDR 17-0 OF THE UNIBUS TRANSCEIVERS.D2 WILL BECOME C1 OF THE UNIBUS.THE KS10 WILL NOW ISSUE A DATA CYCLE AND STROBE D0-35+2P OVER TO THE UBA WITH D20- 35 CONTAINING THE DATA TO BE WRITTEN INTO THE RH11-C REGISTER.THE UBA WILL GATE THE 4X4 MEM(LOC1) AND THE SIGNAL ADDR-BUS WILL COME TRUE AND CAUSE BBSY TO GO LO.THE UBA WILL GATE THE ADDRESS,C1,AND THE DATA TO THE UNIBUS AND STROBE IT TO THE RH11-C WITH MSYNC. THE RH11-C WILL DECODE THE ADDRESS LINES 17-0 IN THE FOLLOWING ORDER ADDR 17-6 WILL DECODE A 7767 AS"DEV SEL",ADDR5-1 WILL CAUSE THE PROM TO DECODE LOC/REMOTE (INTERNAL/EXTERNAL REG) AND THE REGISTER SELSECT.C1 WILL DEFINE A WRITE REG FUNCTION AND WITH THE SELECTED REGISTER LOAD D15-0 INTO THR RH11-C REGISTER.THE RH11-C WILL THEN ISSUE A SSYNC TO THE UNIBUS AND TERMINATE THE OPERATION.

NPR DATA WRITE (FAST MODE)

THE KS10 PROCESSOR WILL PERFORM A SERIES OF REGISTER WRITES TO SET UP THE RM03,RH11-C,AND THE UBA.THE RM03 WILL HAVE THE OFFSET,DESIRED CYLINDER, AND DESIRED TRACK/SECTOR SET AND THE RH11-C WILL SET UP THE WORD COUNT REGISTERS (512 WORDS 2'S COMPLIMENT),CONTROL AND STATUS REGISTER #2 (SET UNIT SELECT) AND THE BUFFER ADDRESS REGISTER WITH THE STARTING LOCATION OF OUR DATA (IT SHOULD BE NOTED THAT THIS ADDRESS IS UNPAGED AND SHIFTED LEFT TWO BITS TO COMPENSATE FOR BYTE AND HALFWORD ADDRESSING). THE UBA WILL HAVE ITS PAGER SET WITH A LOCATION DEFINING THE VALID BIT,36 BIT ENABLE AND PAGE #. THE KS10 WILL THEN FETCH AN I/O INSTRUCTION IN THE FOLLOWING FORMAT:

713760,001000 WITH 713 DEFINING I/O WRITE,AC 17 CONTAING A FUNCTION CODE OF A 61 AND INDIRECT ADDRESS 1000 CONTAINING 1,776700 DEFINING UBA 1 AND THE BUS HOG RH11-C.THE PROCESSOR WILL GENERATE BUS REQUEST TO THE CONSOLE.IF THE BUS IS FREE THE ARBITRATOR WILL ISSUE BUS GRANT PASSING CONTROL TO THE KS10.THE PROCESSOR WILL ISSUE A COM/ADDR CYCLE STROBING D0-35+2P IN THE FOLLOWING FORMAT:

D0=1 I/O
D2=1 WRITE
D14-17= UBA#1
D18-35= RH11-C&REG 00

THE UBA UPON RECEIPT OF THE COMMAND/ADDRESS CYCLE WILL GATE D2 TO THE UNIBUS TRANSMITTER FOR C1 AND D18-35 TO THE UNIBUS ADDRESS TRANSMITTERS DEFINING 776700(RH11-C&REG 00). THE KS10 WILL NOW ISSUE A DATA CYCLE STROBING D0-35+2P TO THE UBA CARRYING A FUNCTION CODE OF A 61 IN BITS D30-35. THE UBA WILL NOW CAUSE WITH DATA CYCLE BBSY TO GO LOW AND THE FUNCTION CODE 61 TO BE GATED THROUGH THE 4X4 MEM TO THE UNIBUS DATA LINES D15-0. MSYNC WILL BE ASSERTED TO THE UBA AND STROBE ADDRESS 17-0,C1&C0,AND D15-0 TO THE RH11-C. THE RH11-C UPON THE RECEIPT OF MSYNC WILL GATE THE ADDRESS 17-0 LINES INTO ITS DECODER.BITS 17-6 WILL CAUSE DEVICE SELECT AND BITS 5-1 WILL DECODE TO REGISTER SELECT 4-0.CS1 WILL DECODE A WRITE FROM 61 AND INITIATE TWO BUS OPERATIONS. ON THE MASSBUS THE RH11-C WILL SEND OUT DR.SEL.2-0 FROM CS2,REG.SEL. 4-0 WITH 00,C1=1 AND C15-0+P WILL CONTAIN A FUNCTION CODE OF 61.IT WILL BE STROBED OVER WITH DEMAND AND CAUSE THE RM03 TO DO THE FOLLOWING. THE MBA WILL DECODE THE DRIVE SELECT LINES TO GENERATE MBA SELECT AND ENABLE A GREY CODE COUNTER TO CAUSE CTOD TO BECOME ASY WRT.THE REG.SEL LINES WILL BE DECODED AND CAUSE RMCNTR REG TO COME TRUE.WITH ASY WRT AND THE CONTROL REG SELECTED THE MBA WILL GATE C15-0+P INTO THE CONTROL REG. THE CONTROL REG WILL NOW WAIT FOR RUN AND THE CONTROL SEQUENCER OR THE MBA WILL ENTER AN F ENABLE LOOP.THE GREY CODE COUNTER WILL SEND OUT TRANSFER AND OCCUPIED TO THE MASSBUS AND NOW THE MBA WILL WAIT FOR RUN. THE RH11-C WILL INITIATE THE SECOND BUS OPERATION ON THE UNIBUS.IN ORDER TO SEND RUN THE RH11-C SILO MUST BE FULL,SO NPR'S MUST TAKE PLACE. THE RH11-C WILL SEND NPR TO THE UBA AND RECEIVE NPG BACK.THE RH11-C WILL RESPOND WITH SACK AND HOLD BBSY LOW UNTIL 16 WORDS ARE LOADED (BUS HOG.THE RH11-C WILL TAKE THE UNPAGED ADDRESS(FROM BUFF.ADDR) AND BROADCAST IT WITH C1=0 TO THE UBA WITH MSYNC.THE UBA UPON RECIEPT OF MSYNC WILL REQUEST THE KS10 BUS VIA THE CONSOLEFEED THE ADDRESS TO THE PAGING

RAM.WITH THE RECEIPT OF BUS GRANT THE UBA WILL BROADCAST A COMM/ADDR CYCLE WITH D0-35+2P IN THE FOLLOWING FORMAT:

D0=0 MEM
D1=1 READ
D14-35 PAGED
ADDRESS FROM THE BUF & PAGER.

BECAUSE D0 IS RESET THE MEMORY WILL TAKE THE PAGED ADDRESS LINES 14-35 AND DECODE THEM IN THE FOLLOWING FORMAT:

D14-16 SELECTS MMC BOARD
D17-19 SELECTS MMA BOARD
D20-21 SELECTS ONE OF 4 POSSIBLE 16K CHIPS ON MMA
D22-28 ROW ADDRESS
D29-35 COLUMN ADDRESS

WITH D1 SET THE MOS MEMORY WILL LOAD A ROW/COLUMN ADDRESS STROBE AND READ 43 DATA BITS AND DECODE THE ECC FOR PROPER PARITY. THE MOS MEMORY WILL ISSUE A DATA CYCLE BACK TO THE UBA CARRYING D0-35+2P TO THE UBA.UPON RECEIPT OF THE DATA CYCLE THE UBA WILL GATE D0-17 INTO THE 4X4 MEMORY LOC 0 AND D18-35 INTO LOC 2.THE UBA WILL GATE D0-17 TO THE UNIBUS AND WITH SSYNC STROBE THE FIRST 18 BIT HALFWORD TO THE RH11-C ON D17-0 OF THE UNIBUS. THE RH11-C WILL GATE THE WORD INTO THE IBUFF AND ALLOW IT TO BUBBLE UP THROUGH THE SILO TO THE OBUFF.THE RH11-C WILL ALSO ISSUE ANOTHER MSYNC WITH THE NOW INCREMENTED BUFF. ADDR. AND C1 REST TO FETCH THE SECOND HALFWORD.THE UBA UPON RECEIPT OF EVERY TWO MSYNC'S WILL REQUEST A WORD FROM MEMORY. THIS PROCESS WILL CONTINUE UNTIL THE SILO IS FULL AND THE SIGNAL START COMES TRUE IN THE RH11-C.WITH START THE RH11-C WILL BROADCAST RUN ON THE MASSBUS. WITH RUN RECEIVED THE RM03 WILL INITIATE A DATA WRITE BY FIRST SEARCHING FOR THE HEADER.WITH HEADER FOUND THE RM03 WILL SEND AN SCLK WHICH WILL BE WRAPPED AROUND AND SENT BACK AS WCLK.WCLK WILL STROBE THE OBUFF TO THE MBA ON D17-0+P LOADING THE DATA BUFFER IN THE MBA.AS THE WORDS ARE

SENT TO THE SHIFT REGISTER AND THE DB BECOMES EMPTY THE RH11-C WILL CONTINUE TO SEND DATA TO THE RM03 WITH WCLKS.THE RH11-C WILL ALSO CONTINUE TO REQUEST THE BUS FOR WORDS FROM THE UBA.WHEN THE WORD COUNT REG. IS FINALLY EMPTY (TRANSFERRED 521 X 18 BIT WORDS) RUN WILL BE RESET.WITH RUN RESET AT EBL TIME THE RM03 WILL TERMINATE AND RETURN TO AN F ENABLE LOOP. EBL RECEIVED IN THE RH11-C WILL CAUSE DONE TO SET AND BROADCAST A PRIORTY INTERRUPT TO THE UBA KS10.

NPR WRITE (SLOW MODE)

THE KS10 PROCESSOR WILL PERFORM A SERIES OF REGISTER WRITES TO THE UBA, RH11 AND TM02/03. THE TM02/03 WILL HAVE THE FRAME COUNT AND TAPE CONTROL REGISTER LOADED WITH THE NUMBER OF FRAMES TO WRITE, SLAVE SELECT, DENSITY AND FORMAT BITS, RESPECTIVELY. THE RH11 WILL HAVE THE BUFFER ADDRESS, WORD COUNT AND CS2 LOADED. UBA #3 WILL HAVE THE PAGE SET WITHOUT 36 BIT ENABLE ON AND A VALID PAGE INSERTED. THE KS10 WILL FETCH AN I/O WRITE INSTRUCTION IN THE FOLLOWING FORMAT: 713760, 1000 WHERE AC17 WILL CONTAIN A FUNCTION CODE OF 61 AND INDIRECT ADDRESS 1000 WILL CONTAIN 3,,772440. THE PROCESSOR WILL GENERATE A BUS REQUEST TO THE CONSOLE BOARD. IF NOT OCCUPIED THE CONSOLE WILL RESPOND WITH BUS GRANT. THE KS10 NOW IN CONTROL OF THE BUS, WILL ISSUE A COM/ADR CYCLE STROBING D0-35 + 2P TO THE UBA IN THE FOLLOWING FORMAT:

D0=1 I/O
D2=1 WRITE
D14-17 UBA#3
D18-35 RH11 (772440)

UBA #3 UPON RECEIPT OF THE COM/ADR CYCLE WILL LOAD THE UNIBUS ADDRESS TRANSMITTER WITH D18-35 AFTER THE UBA DETERMINES THAT AN EXTERNAL REGISTER IS SELECTED. D2 WILL DIRECTLY FEED C1 OF THE UNIBUS DEFINING A DATAO.

THE KS10 WILL NOW GENERATE A DATA CYCLE AND STROBE D0-35+2P TO THE UBA CARRYING A FUNCTION CODE OF 61 IN BIT D30-35. THE UBA UPON RECEIPT OF THE DATA CYCLE WILL GATE D18-35 TO THE 4X4 MEM AND CAUSE A UNIBUS CYCLE. THE SIGNAL ADDR BUS WILL COME TRUE AND CAUSE BBSY TO GO LOW. MSYNC WILL NOW STROBE THE ADDRESS, C1 AND C0, AND DATA BITS OVER TO THE RH11. WITH ADDRESS BITS 17-0 CONTAINING 772440 THE RH11 WILL DECODE DEVICE SELECT AND WITH C1 CAUSE CS1 TO COME TRUE. THE D15-0 LINES WILL CONTAIN THE FUNCTION CODE OF 61 AND BE LOADED INTO THE CONTROL REGISTER SETTING THE FUNCTION WRITE.

TWO OPERATIONS WILL NOW BE PERFORMED BY THE RH11. ON THE MASSBUS DRIVE SELECT 2-0 (FROM CS2), REGISTER SELECT 4-0 (FROM THE PROM) CTOD SET TO A ONE (FROM C1) AND C15-0 (FROM D15-0) CARRYING A FUNCTION OF 61 WILL BE STROBED TO THE TM02/03 WITH DEMAND. THE TM02/03 WILL COMPARE THE DRIVE SELECT LINES WITH THE UNIT SELECT PLUG. WITH A MATCH THE TM02/03 WILL ALLOW THE REGISTER SELECT LINES TO BE DECODED AND WITH CTOD LOAD D15-0+P INTO THE CONTROL REGISTER (R0). TRANSFER WILL NOW BE SENT BACK TO THE RH11 WITH OCCUPIED AND CAUSE SSYNC TO BE SENT ON THE UNIBUS.

AT THE SAME TIME THE RH11 WILL INITIATE NON-PROCESSOR REQUESTS (NPR) TO THE UBA TO FILL UP THE SILO SO RUN CAN BE ISSUED. IT SHOULD BE NOTED THAT THE RH11 HAS BUS HOG BUT ONLY TO TRANSFER 2-18 BIT WORDS BEFORE ANOTHER NPR MUST BE GENERATED.THE RH11 WILL GENERATE NPR AND THE UBA WILL ISSUE NPG AND CAUSE SACK TO BE ASSERTED PULLING DOWN BBSY. THE RH11 WILL NOW GATE THE UNPAGED ADDRESS BITS TO THE UNIBUS WITH C1 RESET. MSYNC WILL BE GATED TO THE UNIBUS AND STROBE IT INTO THE UBA.

UPON RECEIPT OF MSYNC THE UBA WILL GATE THE ADDRESS LINES TO THE PAGER AND ALSO GENERATE BUS REQUEST TO THE CONSOLE BOARD. THE CONSOLE WILL ARBITRATE THE REQUEST AND GENERATE BUS GRANT BACK TO THE UBA. THE UBA WILL NOW INITIATE A COM/ADR CYCLE STROBING D0-35+2P TO THE MEMORY IN THE FOLLOWING FORMAT:

D0=0 MEM
D1=1 READ
D14-35 PAGED ADDRESS

THE MOS MEMORY WILL DECODE THE ADDRESS LINES SELECTING THE APPROPRIATE LOCATION AND CHECKING THE ECC CODE. WITH PROPER PARITY THE MOS WILL ISSUE DATA CYCLE STROBING D0-35+2P TO THE UBA.

THE UBA WILL GATE D0-17 INTO THE 4X4 MEMORY (LOC 1) AND PLACE IT ON THE UNIBUS WITH SSYNC STROBING D17-0 TO THE RH11. THE RH11 UPON RECEIPT OF SSYNC WILL GATE D17-0 INTO THE 1BUFF AND UP THE SILO TO THE 0BUFF. THE RH11 WILL NOW SEND ANOTHER MSYNC TO UBA WITH C1

RESET AND THE NOW UPDATED UNPAGED ADDRESS.

THE UBA WILL NOW REQUEST THE KS10 BUS (BUS REQUEST) FROM THE CONSOLE AND WAIT FOR BUS GRANT. THE UBA WILL INITIATE A READ FROM MEMORY OF THE SAME LOCATION AS PREVIOUS ONLY GATING D18-35 INTO THE 4X4 (LOC 1) MEMORY AND OUT TO THE UNIBUS ON D17-0 WITH SSYNC.

THE RH11 WILL AGAIN GATE DATA INTO THE SILO. THIS PROCESS OF LOADING THE SILO WILL CONTINUE UNTIL IT IS FULL. IT SHOULD BE NOTED THAT EACH TIME THE RH11 REQUESTED THE BUS THE BUFFER ADDRESS AND WORD COUNT REGISTE WERE INCREMENTED.

ONCE THE SILO IF FULL THE RH WILL CAUSE RUN TO BE SENT ON THE MASSBUS AND CAUSE THE TM02/03 TO START WRITING DATA. THE BIT FIDDLER WILL REQUEST TWO WORDS BY SENDING SCLK WRAPPING IT AROUND AND STROBING THE 0BUFF TO THE TM02/03 WITH WCLK ON D17-0+P. WITH THE SECOND SCLK/WCLK SIGNAL THE SECOND 18 BIT WORD WILL BE LAODED, INCREMENTING THE DRIVE WORD COUNT REGISTER AND CAUSING THE RH11 TO AGAIN REQUEST WORDS FROM MEMORY.

THE PROCESS OF REQUESTING WORDS WILL CONTINUE UNTIL THE WORD COUNT REGISTER IS EMPTY. THE RH11 WILL THEN RESET RUN AND WAIT FOR EBL.

THE TM02/03 WILL UPON FRAME COUNT OVERFLOW GENERATE EBL AND CAUSE THE TM02/03 TO TERMINATE AND THE RH11 TO SET DONE AND PERFORM A PI (PRIORITY INTERRUPT).

NPR READ (FAST MODE)

THE KS10 WILL PERFORM A SERIES OF REGISTER WRITES TOTHE UBA, RH11-C, AND RM03 TO SET UP THE PAGER, CS2, WC, BA, DCA, DA OFFSET

REGISTERS. THE PAGER MUST HAVE THE VALID BIT AND 36 BIT ENABLE BIT SET. THE KS10 WILL FETCH AN INSTRUCTION IN THE FOLLOWING FORMAT. 713760,1000, WHERE AC 17 WILL DEFINE A 71 READ AND INDIRECT ADDRESS 1000 WILL CONTAIN 1,,776700 SELECTING UBA #1 AND THE RH11-C.

THE KS10 WILL NOW ISSUE BR TO THE CONSOLE AND WAIT FOR BG. Upon receipt of bus grant the KS10 will assert D0-35+2P and strobe it with command/address cycle. D0-35+2P will be in the following format:

D0=1 I/O
D2=1 Write
D14-17 = UBA #1
D18-35 = RH11-C (776700)

The UBA will gate D2 to the C1 line transmitter and D18-35 to the unibus address transmitters.

The KS10 will noW issue a data cycle strobing the function code of 71 on D30-35. The UBA will gate 18-35 to the unibus D15-0 lines, pull down bus busy and assert MSYNC strobing ADDR17-0 (776700), C1&C0, and D15-0 (71) to the RH11-C.

The RH11-C upon receipt of MSYNC will gate the adder linES into a comparitor to generate DEV SEL. The register select PROM will cause, with C1, CS1 in and load D15-0 in CS1 setting the function read. The RH11-C will now cause a massbus cycle AND send drive select 2-0 (from CS2), register select 4-0 (from PROM decode), CTOD=1 (from C1) and C15-0+P (71) strrobing it to the RM03 with demand. The RH11-C wil also gate SSYNC back to the UBA.

The selected RM03 will decode the drive select bits to cause MBA select and gate the read function code into the control sequencer. The RM03 will send TRA and OCC back to THE RH11-C and in return the RH11-C will send RUN AND SSYNC.

The RM03 will now search for the proper header and proceed to read the data field into the SR and data buffer. The RM03 wilL issue a SCLK to the RH11-C strobing D17-0+P into the IBuff of the RH11-C. The data word will bubble up through the silo and enter the OBuff causing an NPR.

The UBA will respond with NPG and the RH11-C will send SACK holding down BBSY. The drive word count register will be incremented and the word count register also. As MSYNC is asserted the RH11-C will gate C1, addr 17-0 (from the unpaged BA) and D17-0 (OBuff) to the unibus.

The UBA will now load D17-0 into the fast transfer register and wait for the next MSYNC. (It should be noted that the fast transfer register only holds D17-0 and never D18-35).

The RM03 will now generate another SCLK strobing D17-0+P to the UBA. After bubbling up through the silo the RH11 will issue another MSYNC strobing the second 18 bit word on D17-0 with the now updated buffer address. The UBA upon receipt of MSYNC will gate D17-0 into D18-35 of the 2x4 memory. The UBA will now issue a BR to the console and wait for a BG. A com/adr cycle will occur strobing D0-35+2P to the MOS memory in the following FORMAT:

D0=0 MEM
D2=1 Write
D14-35 = Address (Paged)

The UBA will take the address lines of the unibus, gate them to the pager and out to the KS10 bus. The address will be decoded by the MOS memory and open up the location.

The UBA will now gate D0-17 and 18-35 to the KS bus transmitters adding parity and being strobed by data cycle. The data will be loaded into MOS memory.

The process of loading words and incrementing the word count and buffer address registers will continue until word count overflow occurs IN the drive word count register. Run will be negated by the RH11 and at EBL time in the RM03 cause the transfer to stop. The RH11 will stop transferring data when the silo is empty and cause DONE to set and issue a PI to the UBA.

NPR Read (Slow Mode)

The KS10 will perform a series of register writes to the UBA, RH11, and TM02/03 to set up the pager, CS2, WC, BA and tape control register. The pager must have the valid bit set and 36 bit enable shut off. The KS10 will fetch an instruction in the following format 713 760,1000, where AC 17 will contain a function code of 71 and indirect address 1000 will contain 3,,772440 selecting UBA #3 and the RH11 attached to the TM02/03. The KS10 will issue a bus request (BR) to the console and wait for bus grant (BG) to return. The KS10 will then issue a comm/addr cycle clocking D0-35+2P in the following format:

D0=1 I/O
D2=1 Write
D14-17 UBA #3
D18-35 = RH11 (772440)

The UBA upon receipt of the C/A cycle will gate D2 up to the C1 transceiver and D18-35 to the address transceiver of the Unibus.

The KS10 will now issue a data cycle strobing D0-35+2P carrying the function code of 71. The UBA will gate the function code to the 4x4 memory and pull down BBSY and strobe C1 and C0, ADDR17-0 and D15-0 with MSYNC to the RH11.

The RH11 will decode the address lines and generate device select and CS1 will come true. With C1 set D15-0 will gate the 71 function code into CS1 and generate read. The RH11 will now initiate a massbus cycle sending drive select 2-0 (from CS2), Register Select 4-0 from address lines 6-1, CTOD=1 (from C1) and D15-0+P (from D15-0 of Unibus). Demand will come true and strobe the data to the TM02/03. SSYNC will also be gated back to the UBA when transfer and occupied are received. The TM02/03 will now load the function code of 71 into its control register and now wait for run.

The RH11 will now issue RUN and wait for the TM02/03 to gate data with an SCLK. SCLK will strobe D17-0+P to the RH11 loading the IBUFF.

After bubbling up the RH11 will issue an NPR and wait for NPG. Sack will now be asserted holding BBSY low. The RH11 will gate the unpaged left shifted address to the Unibus with C1 and the 0BUFF contents in D17-0 with MSYNC. The UBA upon receipt of MSYNC will gate the address lines to the pager and load the D17-0 lines into the 2x4 memory bits 0-17.

The UBA will now issue BR and wait for BG. It will then cause a C/A cycle with D0-35+2P in the following format:

D0=0 Memory
D2=1 Write
D14-35 = Paged Address

The memory will gate the address into the MOS memory lines and open up the location.

The UBA will now issue data cycle and strobe D0-17 over to the memory. (Note D18-35 will go but data is not valid). The RH11 will again issue MSYNC to the UBA loading the same unpaged address (address bit 1 now equals 1) and gate the D17-0 to the 2x4 memory bits 18-35.

The UBA will now issue another BR and wait for BG. A C/A cycle will again strobe D0-35+2P in the following format:

D0=0 Memory
D1 D2=1 Read/Pause Write
   _
D14-35=Paged Address

The memory will open the location, read the data sent, cause a data cycle and strobe D0-35+2P to teh RH11 (D0-17 is valid).

With receipt of D0-17 the UBA will gate D18-35 to the KS10 bus and cause another data cycle (two data cycles for one C/A cycle).

This process will continue until the RH11 gets word count overflow and negates RUN.

The TM02/03 will issue EBL and check RUN. With RUN !!!!! the TM02/03 will shut down. The RH11 will cause a PI when the silo is empty.

PI

The RH11 upon receipt of an exception, attention or DONE will initiate an interrupt (provided its enabled). The RH11 will send a BUS REQUEST (BR) on the Unibus.

The UBA upon receipt of the BR(1-7) will gate it to the encoder to generate a PI on the KS10 bus. The KS10 upon receipt of the interrupt will enter an interrupt routine. The KS10 will issue a Bus Request to the console and wait for Bus Grant back. The KS10 will thens end D0-35+2P with a C/A cycle. D0-35 will be in the following format:

D0=1 I/O
D4=1 Device Read
D15-17=PI Level Received

Upon receipt of the C/A cycle all devices having a PI level equal to D15-17 and having an interrupt pending will now pull down D19-23 (each bit defining physical device, # example UBA #1 pulls D19.) The KS10 will examine the bus to determine what and how many devices have a PI pending. Once the KS10 has determined what device it will service first it will again request the KS10 bus and wait for grant in return. The KS10 will issue another C/A cycle with D0-35+2P in the following format:

D0=1 I/O
D1=1 Read
D5-1 Vector Address
D14-17 Physical Device #

The UBA upon receipt of the C/A cycle will gate BG back to the Unibus and RH11.

The RH11 upon receipt of BG will issue the signal interrupt and gate the interrupt vector address back on D15-0 with C1 set. D15-0 received by the UBA will cause SSYNC to return to the RH11.

The UBA will now request the KS10 bus and with grant received generate I/O data cycle. D0-35+2P will be in the following format:

D20-35 will contain vector address

The KS10 micorcode will now fetch to the EPT the location specified by the microcode and perform an interrogation routine to determine the cause of the interrupt.

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG        C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS AC LO              -H  1F14P2          1-01 !  1  DPMB S M8621        1         4     8              N  1                      4
BUS AC LO              -H  1F15S2          1-02 *  O  UBAC S M8619 O      2         0  -800              N  1     >1 SOURCE        4
BUS AC LO              -H  1F16S2          1-03 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE        4
BUS AC LO              -H  1F17S2          1-04 *  O  UBAC S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE        4
BUS AC LO              -H  1F19S2          1-05 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE        4
BUS AC LO              -H  1F18S2          1-06 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM       4
BUS AC LO              -H                  1                                     -516 -3749     0     0     5-4/8                  4
BUS BAD DATA CYCLE     -H  1B09M2          1-01 *  T  MMCA S M8618 T      2      -520  -557              N  3     EXTRA TERM       5
BUS BAD DATA CYCLE     -H  1B14M2          1-02 *  O  DPMC S M8621 O      1         0  -800              N  1     >1 SOURCE        5
BUS BAD DATA CYCLE     -H  1B15M2          1-03 *  O  UBA5 S M8619 O      2         0  -800              N  1     >1 SOURCE        5
BUS BAD DATA CYCLE     -H  1B16M2          1-04 *  O  UBA5 S M8619 O      1         0  -800              N  1     >1 SOURCE        5
BUS BAD DATA CYCLE     -H  1B17M2          1-05 *  O  UBA5 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE        5
BUS BAD DATA CYCLE     -H  1B19M2          1-06 *  O  UBA5 S M8619 O      1         0  -800              N  1     >1 SOURCE        5
BUS BAD DATA CYCLE     -H  1B18M2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM       5
BUS BAD DATA CYCLE     -H                  1                                    -1040 -5114     0     0     8-4/8                  5
BUS COM/ADR CYCLE      -H  1B09J2          1-01 *  T  MMCA S M8618 T      2      -520  -557              N  3     EXTRA TERM       6
BUS COM/ADR CYCLE      -H  1B14J2          1-02 *  O  DPMC S M8621 O      1         0  -800              N  1     >1 SOURCE        6
BUS COM/ADR CYCLE      -H  1B15J2          1-03 *  O  UBA5 S M8619 O      2         0  -800              N  1     >1 SOURCE        6
BUS COM/ADR CYCLE      -H  1B16J2          1-04 *  O  UBA5 S M8619 O      1         0  -800              N  1     >1 SOURCE        6
BUS COM/ADR CYCLE      -H  1B17J2          1-05 *  O  UBA5 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE        6
BUS COM/ADR CYCLE      -H  1B19J2          1-06 *  O  UBA5 S M8619 O      1         0  -800              N  1     >1 SOURCE        6
BUS COM/ADR CYCLE      -H  1B18J2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM       6
BUS COM/ADR CYCLE      -H                  1                                    -1040 -5114     0     0     8-4/8                  6
BUS DATA 00            -H  1C09D2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM       7
BUS DATA 00            -H  1C14D2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE        7
BUS DATA 00            -H  1C15D2          1-03 *  O  UBAC S M8619 O      2         0  -800              N  1     >1 SOURCE        7
BUS DATA 00            -H  1C16D2          1-04 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE        7
BUS DATA 00            -H  1C17D2          1-05 *  O  UBAC S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE        7
BUS DATA 00            -H  1C19D2          1-06 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE        7
BUS DATA 00            -H  1C18D2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM       7
BUS DATA 00            -H                  1                                    -1040 -5114     0     0     8-4/8                  7
BUS DATA 01            -H  1C09E2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM       8
BUS DATA 01            -H  1C14E2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE        8
BUS DATA 01            -H  1C15E2          1-03 *  O  UBAC S M8619 O      2         0  -800              N  1     >1 SOURCE        8
BUS DATA 01            -H  1C16E2          1-04 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE        8
BUS DATA 01            -H  1C17E2          1-05 *  O  UBAC S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE        8
BUS DATA 01            -H  1C19E2          1-06 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE        8
BUS DATA 01            -H  1C18E2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM       8
BUS DATA 01            -H                  1                                    -1040 -5114     0     0     8-4/8                  8
```

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG        C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS DATA 02            -H  1C09F2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM     9
BUS DATA 02            -H  1C14F2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE      9
BUS DATA 02            -H  1C15F2          1-03 *  O  UBAB S M8619 O      2         0  -800              N  1     >1 SOURCE      9
BUS DATA 02            -H  1C16F2          1-04 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE      9
BUS DATA 02            -H  1C17F2          1-05 *  O  UBAB S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE      9
BUS DATA 02            -H  1C19F2          1-06 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE      9
BUS DATA 02            -H  1C18F2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM     9
BUS DATA 02            -H                  1                                    -1040 -5114     0     0     8-4/8                9
BUS DATA 03            -H  1C09H2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM    10
BUS DATA 03            -H  1C14H2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE     10
BUS DATA 03            -H  1C15H2          1-03 *  O  UBAB S M8619 O      2         0  -800              N  1     >1 SOURCE     10
BUS DATA 03            -H  1C16H2          1-04 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE     10
BUS DATA 03            -H  1C17H2          1-05 *  O  UBAB S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE     10
BUS DATA 03            -H  1C19H2          1-06 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE     10
BUS DATA 03            -H  1C18H2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM    10
BUS DATA 03            -H                  1                                    -1040 -5114     0     0     8-4/8               10
BUS DATA 04            -H  1C09J2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM    11
BUS DATA 04            -H  1C14J2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE     11
BUS DATA 04            -H  1C15J2          1-03 *  O  UBAB S M8619 O      2         0  -800              N  1     >1 SOURCE     11
BUS DATA 04            -H  1C16J2          1-04 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE     11
BUS DATA 04            -H  1C17J2          1-05 *  O  UBAB S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE     11
BUS DATA 04            -H  1C19J2          1-06 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE     11
BUS DATA 04            -H  1C18J2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM    11
BUS DATA 04            -H                  1                                    -1040 -5114     0     0     8-4/8               11
BUS DATA 05            -H  1C09K2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM    12
BUS DATA 05            -H  1C14K2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE     12
BUS DATA 05            -H  1C15K2          1-03 *  O  UBAB S M8619 O      2         0  -800              N  1     >1 SOURCE     12
BUS DATA 05            -H  1C16K2          1-04 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE     12
BUS DATA 05            -H  1C17K2          1-05 *  O  UBAB S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE     12
BUS DATA 05            -H  1C19K2          1-06 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE     12
BUS DATA 05            -H  1C18K2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM    12
BUS DATA 05            -H                  1                                    -1040 -5114     0     0     8-4/8               12
BUS DATA 06            -H  1C09L2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM    13
BUS DATA 06            -H  1C14L2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE     13
BUS DATA 06            -H  1C15L2          1-03 *  O  UBAA S M8619 O      2         0  -800              N  1     >1 SOURCE     13
BUS DATA 06            -H  1C16L2          1-04 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE     13
BUS DATA 06            -H  1C17L2          1-05 *  O  UBAA S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE     13
BUS DATA 06            -H  1C19L2          1-06 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE     13
BUS DATA 06            -H  1C18L2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM    13
BUS DATA 06            -H                  1                                    -1040 -5114     0     0     8-4/8               13
BUS DATA 07            -H  1C09M2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM    14
BUS DATA 07            -H  1C14M2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE     14
BUS DATA 07            -H  1C15M2          1-03 *  O  UBAA S M8619 O      2         0  -800              N  1     >1 SOURCE     14
BUS DATA 07            -H  1C16M2          1-04 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE     14
BUS DATA 07            -H  1C17M2          1-05 *  O  UBAA S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE     14
BUS DATA 07            -H  1C19M2          1-06 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE     14
BUS DATA 07            -H  1C18M2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM    14
BUS DATA 07            -H                  1                                    -1040 -5114     0     0     8-4/8               14
```

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG        C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS DATA 08            -H  1C09N2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      15
BUS DATA 08            -H  1C14N2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       15
BUS DATA 08            -H  1C15N2          1-03 *  O  UBAA S M8619 O      2         0  -800              N  1     >1 SOURCE       15
BUS DATA 08            -H  1C16N2          1-04 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       15
BUS DATA 08            -H  1C17N2          1-05 *  O  UBAA S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       15
BUS DATA 08            -H  1C19N2          1-06 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       15
BUS DATA 08            -H  1C18N2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      15
BUS DATA 08            -H                  1                                    -1040 -5114      0     0    8-4/8                 15
BUS DATA 09            -H  1C09P2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      16
BUS DATA 09            -H  1C14P2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       16
BUS DATA 09            -H  1C15P2          1-03 *  O  UBAA S M8619 O      2         0  -800              N  1     >1 SOURCE       16
BUS DATA 09            -H  1C16P2          1-04 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       16
BUS DATA 09            -H  1C17P2          1-05 *  O  UBAA S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       16
BUS DATA 09            -H  1C19P2          1-06 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       16
BUS DATA 09            -H  1C18P2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      16
BUS DATA 09            -H                  1                                    -1040 -5114      0     0    8-4/8                 16
BUS DATA 10            -H  1C09R2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      17
BUS DATA 10            -H  1C14R2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       17
BUS DATA 10            -H  1C15R2          1-03 *  O  UBA9 S M8619 O      2         0  -800              N  1     >1 SOURCE       17
BUS DATA 10            -H  1C16R2          1-04 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       17
BUS DATA 10            -H  1C17R2          1-05 *  O  UBA9 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       17
BUS DATA 10            -H  1C19R2          1-06 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       17
BUS DATA 10            -H  1C18R2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      17
BUS DATA 10            -H                  1                                    -1040 -5114      0     0    8-4/8                 17
BUS DATA 11            -H  1C09S2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      18
BUS DATA 11            -H  1C14S2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       18
BUS DATA 11            -H  1C15S2          1-03 *  O  UBA9 S M8619 O      2         0  -800              N  1     >1 SOURCE       18
BUS DATA 11            -H  1C16S2          1-04 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       18
BUS DATA 11            -H  1C17S2          1-05 *  O  UBA9 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       18
BUS DATA 11            -H  1C19S2          1-06 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       18
BUS DATA 11            -H  1C18S2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      18
BUS DATA 11            -H                  1                                    -1040 -5114      0     0    8-4/8                 18
BUS DATA 12            -H  1D09D2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      19
BUS DATA 12            -H  1D14D2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       19
BUS DATA 12            -H  1D15D2          1-03 *  O  UBA9 S M8619 O      2         0  -800              N  1     >1 SOURCE       19
BUS DATA 12            -H  1D16D2          1-04 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       19
BUS DATA 12            -H  1D17D2          1-05 *  O  UBA9 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       19
BUS DATA 12            -H  1D19D2          1-06 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       19
BUS DATA 12            -H  1D18D2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      19
BUS DATA 12            -H                  1                                    -1040 -5114      0     0    8-4/8                 19
BUS DATA 13            -H  1D09E2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      20
BUS DATA 13            -H  1D14E2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       20
BUS DATA 13            -H  1D15E2          1-03 *  O  UBA9 S M8619 O      2         0  -800              N  1     >1 SOURCE       20
BUS DATA 13            -H  1D16E2          1-04 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       20
BUS DATA 13            -H  1D17E2          1-05 *  O  UBA9 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       20
BUS DATA 13            -H  1D19E2          1-06 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       20
BUS DATA 13            -H  1D18E2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      20
BUS DATA 13            -H                  1                                    -1040 -5114      0     0    8-4/8                 20
```

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG          C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS DATA 14            -H  1D09F2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2-4/8 EXTRA TERM      21
BUS DATA 14            -H  1D13F2          1-02 *  O  DPEB S M8620 O      2         0  -800              N  1     >1 SOURCE       21
BUS DATA 14            -H  1D14F2          1-03 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       21
BUS DATA 14            -H  1D15F2          1-04 *  O  UBA8 S M8619 O      2         0  -800              N  1     >1 SOURCE       21
BUS DATA 14            -H  1D16F2          1-05 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       21
BUS DATA 14            -H  1D17F2          1-06 *  O  UBA8 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       21
BUS DATA 14            -H  1D19F2          1-07 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       21
BUS DATA 14            -H  1D18F2          1-08 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      21
BUS DATA 14            -H                  1                                    -1040 -5914     0     0     9-0/8                 21
BUS DATA 15            -H  1D09H2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2-4/8 EXTRA TERM      22
BUS DATA 15            -H  1D13H2          1-02 *  O  DPEB S M8620 O      2         0  -800              N  1     >1 SOURCE       22
BUS DATA 15            -H  1D14H2          1-03 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       22
BUS DATA 15            -H  1D15H2          1-04 *  O  UBA8 S M8619 O      2         0  -800              N  1     >1 SOURCE       22
BUS DATA 15            -H  1D16H2          1-05 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       22
BUS DATA 15            -H  1D17H2          1-06 *  O  UBA8 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       22
BUS DATA 15            -H  1D19H2          1-07 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       22
BUS DATA 15            -H  1D18H2          1-08 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      22
BUS DATA 15            -H                  1                                    -1040 -5914     0     0     9-0/8                 22
BUS DATA 16            -H  1D09J2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2-4/8 EXTRA TERM      23
BUS DATA 16            -H  1D13J2          1-02 *  O  DPEB S M8620 O      2         0  -800              N  1     >1 SOURCE       23
BUS DATA 16            -H  1D14J2          1-03 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       23
BUS DATA 16            -H  1D15J2          1-04 *  O  UBA8 S M8619 O      2         0  -800              N  1     >1 SOURCE       23
BUS DATA 16            -H  1D16J2          1-05 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       23
BUS DATA 16            -H  1D17J2          1-06 *  O  UBA8 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       23
BUS DATA 16            -H  1D19J2          1-07 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       23
BUS DATA 16            -H  1D18J2          1-08 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      23
BUS DATA 16            -H                  1                                    -1040 -5914     0     0     9-0/8                 23
BUS DATA 17            -H  1D09K2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2-4/8 EXTRA TERM      24
BUS DATA 17            -H  1D13K2          1-02 *  O  DPEB S M8620 O      2         0  -800              N  1     >1 SOURCE       24
BUS DATA 17            -H  1D14K2          1-03 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       24
BUS DATA 17            -H  1D15K2          1-04 *  O  UBA8 S M8619 O      2         0  -800              N  1     >1 SOURCE       24
BUS DATA 17            -H  1D16K2          1-05 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       24
BUS DATA 17            -H  1D17K2          1-06 *  O  UBA8 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       24
BUS DATA 17            -H  1D19K2          1-07 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       24
BUS DATA 17            -H  1D18K2          1-08 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      24
BUS DATA 17            -H                  1                                    -1040 -5914     0     0     9-0/8                 24
BUS DATA 18            -H  1D09L2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      25
BUS DATA 18            -H  1D14L2          1-02 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       25
BUS DATA 18            -H  1D15L2          1-03 *  O  UBAC S M8619 O      2         0  -800              N  1     >1 SOURCE       25
BUS DATA 18            -H  1D16L2          1-04 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       25
BUS DATA 18            -H  1D17L2          1-05 *  O  UBAC S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       25
BUS DATA 18            -H  1D19L2          1-06 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       25
BUS DATA 18            -H  1D18L2          1-07 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      25
BUS DATA 18            -H                  1                                    -1040 -5114     0     0     8-4/8                 25
```

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG        C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS DATA 19            -H  1D09M2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      26
BUS DATA 19            -H  1D14M2          1-02 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       26
BUS DATA 19            -H  1D15M2          1-03 *  O  UBAC S M8619 O      2         0  -800              N  1     >1 SOURCE       26
BUS DATA 19            -H  1D16M2          1-04 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       26
BUS DATA 19            -H  1D17M2          1-05 *  O  UBAC S M8619 O      2         0  -800              N  3-3/8 >1 SOURCE       26
BUS DATA 19            -H  1E19J1          1-06 *  O  UBA7 S M8619 O      3         0  -700              N  2-5/8 >1 SOURCE       26
BUS DATA 19            -H  1D19M2          1-07 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       26
BUS DATA 19            -H  1D18M2          1-08 *  T  CSL7 S M8616 T             -520  -557                       EXTRA TERM      26
BUS DATA 19            -H                  1                                    -1040 -5814     0     0    13-0/8                 26
BUS DATA 20            -H  1D09N2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      27
BUS DATA 20            -H  1D14N2          1-02 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       27
BUS DATA 20            -H  1D15N2          1-03 *  O  UBAB S M8619 O      2         0  -800              N  1     >1 SOURCE       27
BUS DATA 20            -H  1D16N2          1-04 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE       27
BUS DATA 20            -H  1D17N2          1-05 *  O  UBAB S M8619 O      2         0  -800              N  2-5/8 >1 SOURCE       27
BUS DATA 20            -H  1E17J1          1-06 *  O  UBA7 S M8619 O      3         0  -700              N  3-4/8 >1 SOURCE       27
BUS DATA 20            -H  1D19N2          1-07 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE       27
BUS DATA 20            -H  1D18N2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      27
BUS DATA 20            -H                  1                                    -1040 -5814     0     0    13-1/8                 27
BUS DATA 21            -H  1D09P2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      28
BUS DATA 21            -H  1D12P2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       28
BUS DATA 21            -H  1D14P2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       28
BUS DATA 21            -H  1D15P2          1-04 *  O  UBAB S M8619 O      2         0  -800              N  2-5/8 >1 SOURCE       28
BUS DATA 21            -H  1E16J1          1-05 *  O  UBA7 S M8619 O      3         0  -700              N  2-3/8 >1 SOURCE       28
BUS DATA 21            -H  1D16P2          1-06 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE       28
BUS DATA 21            -H  1D17P2          1-07 *  O  UBAB S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       28
BUS DATA 21            -H  1D19P2          1-08 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE       28
BUS DATA 21            -H  1D18P2          1-09 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      28
BUS DATA 21            -H                  1                                    -1040 -6614     0     0    13-0/8                 28
BUS DATA 22            -H  1D09R2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      29
BUS DATA 22            -H  1D12R2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       29
BUS DATA 22            -H  1D14R2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       29
BUS DATA 22            -H  1D15R2          1-04 *  O  UBAB S M8619 O      2         0  -800              N  2-3/8 >1 SOURCE       29
BUS DATA 22            -H  1E15J1          1-05 *  O  UBA7 S M8619 O      3         0  -700              N  2-7/8 >1 SOURCE       29
BUS DATA 22            -H  1D16R2          1-06 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE       29
BUS DATA 22            -H  1D17R2          1-07 *  O  UBAB S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       29
BUS DATA 22            -H  1D19R2          1-08 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE       29
BUS DATA 22            -H  1D18R2          1-09 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      29
BUS DATA 22            -H                  1                                    -1040 -6614     0     0    13-2/8                 29
BUS DATA 23            -H  1D09S2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      30
BUS DATA 23            -H  1D12S2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       30
BUS DATA 23            -H  1D14S2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       30
BUS DATA 23            -H  1D15S2          1-04 *  O  UBAB S M8619 O      2         0  -800              N  1     >1 SOURCE       30
BUS DATA 23            -H  1D16S2          1-05 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE       30
BUS DATA 23            -H  1D17S2          1-06 *  O  UBAB S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       30
BUS DATA 23            -H  1D19S2          1-07 *  O  UBAB S M8619 O      1         0  -800              N  1     >1 SOURCE       30
BUS DATA 23            -H  1D18S2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      30
BUS DATA 23            -H                  1                                    -1040 -5914     0     0     9-0/8                 30
```

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG        C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS DATA 24            -H  1E09D2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      31
BUS DATA 24            -H  1E12D2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       31
BUS DATA 24            -H  1E14D2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       31
BUS DATA 24            -H  1E15D2          1-04 *  O  UBAA S M8619 O      2         0  -800              N  1     >1 SOURCE       31
BUS DATA 24            -H  1E16D2          1-05 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       31
BUS DATA 24            -H  1E17D2          1-06 *  O  UBAA S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       31
BUS DATA 24            -H  1E19D2          1-07 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       31
BUS DATA 24            -H  1E18D2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      31
BUS DATA 24            -H                  1                                    -1040 -5914     0     0     9-0/8                 31
BUS DATA 25            -H  1E09E2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      32
BUS DATA 25            -H  1E12E2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       32
BUS DATA 25            -H  1E14E2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       32
BUS DATA 25            -H  1E15E2          1-04 *  O  UBAA S M8619 O      2         0  -800              N  1     >1 SOURCE       32
BUS DATA 25            -H  1E16E2          1-05 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       32
BUS DATA 25            -H  1E17E2          1-06 *  O  UBAA S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       32
BUS DATA 25            -H  1E19E2          1-07 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       32
BUS DATA 25            -H  1E18E2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      32
BUS DATA 25            -H                  1                                    -1040 -5914     0     0     9-0/8                 32
BUS DATA 26            -H  1E09F2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      33
BUS DATA 26            -H  1E12F2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       33
BUS DATA 26            -H  1E14F2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       33
BUS DATA 26            -H  1E15F2          1-04 *  O  UBAA S M8619 O      2         0  -800              N  1     >1 SOURCE       33
BUS DATA 26            -H  1E16F2          1-05 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       33
BUS DATA 26            -H  1E17F2          1-06 *  O  UBAA S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       33
BUS DATA 26            -H  1E19F2          1-07 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       33
BUS DATA 26            -H  1E18F2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      33
BUS DATA 26            -H                  1                                    -1040 -5914     0     0     9-0/8                 33
BUS DATA 27            -H  1E09H2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      34
BUS DATA 27            -H  1E12H2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       34
BUS DATA 27            -H  1E14H2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       34
BUS DATA 27            -H  1E15H2          1-04 *  O  UBAA S M8619 O      2         0  -800              N  1     >1 SOURCE       34
BUS DATA 27            -H  1E16H2          1-05 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       34
BUS DATA 27            -H  1E17H2          1-06 *  O  UBAA S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       34
BUS DATA 27            -H  1E19H2          1-07 *  O  UBAA S M8619 O      1         0  -800              N  1     >1 SOURCE       34
BUS DATA 27            -H  1E18H2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      34
BUS DATA 27            -H                  1                                    -1040 -5914     0     0     9-0/8                 34
BUS DATA 28            -H  1E09J2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      35
BUS DATA 28            -H  1E12J2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       35
BUS DATA 28            -H  1E14J2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       35
BUS DATA 28            -H  1E15J2          1-04 *  O  UBA9 S M8619 O      2         0  -800              N  1     >1 SOURCE       35
BUS DATA 28            -H  1E16J2          1-05 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       35
BUS DATA 28            -H  1E17J2          1-06 *  O  UBA9 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       35
BUS DATA 28            -H  1E19J2          1-07 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       35
BUS DATA 28            -H  1E18J2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      35
BUS DATA 28            -H                  1                                    -1040 -5914     0     0     9-0/8                 35
```

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG        C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS DATA 29            -H  1E09K2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      36
BUS DATA 29            -H  1E12K2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       36
BUS DATA 29            -H  1E14K2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       36
BUS DATA 29            -H  1E15K2          1-04 *  O  UBA9 S M8619 O      2         0  -800              N  1     >1 SOURCE       36
BUS DATA 29            -H  1E16K2          1-05 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       36
BUS DATA 29            -H  1E17K2          1-06 *  O  UBA9 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       36
BUS DATA 29            -H  1E19K2          1-07 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       36
BUS DATA 29            -H  1E18K2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      36
BUS DATA 29            -H                  1                                    -1040 -5914     0     0     9-0/8                 36
BUS DATA 30            -H  1E09L2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      37
BUS DATA 30            -H  1E12L2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       37
BUS DATA 30            -H  1E14L2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       37
BUS DATA 30            -H  1E15L2          1-04 *  O  UBA9 S M8619 O      2         0  -800              N  1     >1 SOURCE       37
BUS DATA 30            -H  1E16L2          1-05 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       37
BUS DATA 30            -H  1E17L2          1-06 *  O  UBA9 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       37
BUS DATA 30            -H  1E19L2          1-07 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       37
BUS DATA 30            -H  1E18L2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      37
BUS DATA 30            -H                  1                                    -1040 -5914     0     0     9-0/8                 37
BUS DATA 31            -H  1E09M2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      38
BUS DATA 31            -H  1E12M2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       38
BUS DATA 31            -H  1E14M2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       38
BUS DATA 31            -H  1E15M2          1-04 *  O  UBA9 S M8619 O      2         0  -800              N  1     >1 SOURCE       38
BUS DATA 31            -H  1E16M2          1-05 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       38
BUS DATA 31            -H  1E17M2          1-06 *  O  UBA9 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       38
BUS DATA 31            -H  1E19M2          1-07 *  O  UBA9 S M8619 O      1         0  -800              N  1     >1 SOURCE       38
BUS DATA 31            -H  1E18M2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      38
BUS DATA 31            -H                  1                                    -1040 -5914     0     0     9-0/8                 38
BUS DATA 32            -H  1E09N2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      39
BUS DATA 32            -H  1E12N2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       39
BUS DATA 32            -H  1E14N2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       39
BUS DATA 32            -H  1E15N2          1-04 *  O  UBA8 S M8619 O      2         0  -800              N  1     >1 SOURCE       39
BUS DATA 32            -H  1E16N2          1-05 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       39
BUS DATA 32            -H  1E17N2          1-06 *  O  UBA8 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       39
BUS DATA 32            -H  1E19N2          1-07 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       39
BUS DATA 32            -H  1E18N2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      39
BUS DATA 32            -H                  1                                    -1040 -5914     0     0     9-0/8                 39
BUS DATA 33            -H  1E09P2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      40
BUS DATA 33            -H  1E12P2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       40
BUS DATA 33            -H  1E14P2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       40
BUS DATA 33            -H  1E15P2          1-04 *  O  UBA8 S M8619 O      2         0  -800              N  1     >1 SOURCE       40
BUS DATA 33            -H  1E16P2          1-05 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       40
BUS DATA 33            -H  1E17P2          1-06 *  O  UBA8 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       40
BUS DATA 33            -H  1E19P2          1-07 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       40
BUS DATA 33            -H  1E18P2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      40
BUS DATA 33            -H                  1                                    -1040 -5914     0     0     9-0/8                 40
```

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG        C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS DATA 34            -H  1E09R2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      41
BUS DATA 34            -H  1E12R2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       41
BUS DATA 34            -H  1E14R2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       41
BUS DATA 34            -H  1E15R2          1-04 *  O  UBA8 S M8619 O      2         0  -800              N  1     >1 SOURCE       41
BUS DATA 34            -H  1E16R2          1-05 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       41
BUS DATA 34            -H  1E17R2          1-06 *  O  UBA8 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       41
BUS DATA 34            -H  1E19R2          1-07 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       41
BUS DATA 34            -H  1E18R2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      41
BUS DATA 34            -H                  1                                    -1040 -5914     0     0     9-0/8                 41
BUS DATA 35            -H  1E09S2          1-01 *  T  MMC1 S M8618 T      1      -520  -557              N  2     EXTRA TERM      42
BUS DATA 35            -H  1E12S2          1-02 *  O  CRA5 S M8622 O      2         0  -800              N  1-4/8 >1 SOURCE       42
BUS DATA 35            -H  1E14S2          1-03 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       42
BUS DATA 35            -H  1E15S2          1-04 *  O  UBA8 S M8619 O      2         0  -800              N  1     >1 SOURCE       42
BUS DATA 35            -H  1E16S2          1-05 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       42
BUS DATA 35            -H  1E17S2          1-06 *  O  UBA8 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       42
BUS DATA 35            -H  1E19S2          1-07 *  O  UBA8 S M8619 O      1         0  -800              N  1     >1 SOURCE       42
BUS DATA 35            -H  1E18S2          1-08 *  T  CSL6 S M8616 T             -520  -557                       EXTRA TERM      42
BUS DATA 35            -H                  1                                    -1040 -5914     0     0     9-0/8                 42
BUS DATA CYCLE         -H  1B09L2          1-01 *  T  MMCA S M8618 T      2      -520  -557              N  3     EXTRA TERM      43
BUS DATA CYCLE         -H  1B14L2          1-02 *  O  DPMC S M8621 O      1         0  -800              N  1     >1 SOURCE       43
BUS DATA CYCLE         -H  1B15L2          1-03 *  O  UBA5 S M8619 O      2         0  -800              N  1     >1 SOURCE       43
BUS DATA CYCLE         -H  1B16L2          1-04 *  O  UBA5 S M8619 O      1         0  -800              N  1     >1 SOURCE       43
BUS DATA CYCLE         -H  1B17L2          1-05 *  O  UBA5 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       43
BUS DATA CYCLE         -H  1B19L2          1-06 *  O  UBA5 S M8619 O      1         0  -800              N  1     >1 SOURCE       43
BUS DATA CYCLE         -H  1B18L2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM      43
BUS DATA CYCLE         -H                  1                                    -1040 -5114     0     0     8-4/8                 43
BUS I/O BUSY           -H  1B09P2          1-01 *  T  MMCB S M8618 T      2      -520  -557              N  6-5/8 0001024000      44
BUS I/O BUSY           -H  1D12K2          1-02 *     CRA2 S M8622        1         2     3              N  7-5/8                 44
BUS I/O BUSY           -H  1F15N2          1-03 *  O  UBA7 S M8619 O      2         0  -700              N  1     >1 SOURCE       44
BUS I/O BUSY           -H  1F16N2          1-04 *  O  UBA7 S M8619 O      1         0  -700              N  1     >1 SOURCE       44
BUS I/O BUSY           -H  1F17N2          1-05 *  O  UBA7 S M8619 O      2         0  -700              N  1-4/8 >1 SOURCE       44
BUS I/O BUSY           -H  1F19N2          1-06 *  O  UBA7 S M8619 O      1         0  -700              N  1     >1 SOURCE       44
BUS I/O BUSY           -H  1F18N2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM      44
BUS I/O BUSY           -H                  1                                    -1038 -3911     0     0    18-6/8                 44
BUS I/O DATA CYCLE     -H  1B09N2          1-01 *  T  MMCB S M8618 T      1      -520  -557              N  1-5/8 EXTRA TERM      45
BUS I/O DATA CYCLE     -H  1B12M1          1-02 *     CRA2 S M8622        2         2     3              N  1-7/8                 45
BUS I/O DATA CYCLE     -H  1B14N2          1-03 *  O  DPMC S M8621 O      1         0  -800              N  1     >1 SOURCE       45
BUS I/O DATA CYCLE     -H  1B15N2          1-04 *  O  UBA5 S M8619 O      2         0  -800              N  1     >1 SOURCE       45
BUS I/O DATA CYCLE     -H  1B16N2          1-05 *  O  UBA5 S M8619 O      1         0  -800              N  1     >1 SOURCE       45
BUS I/O DATA CYCLE     -H  1B17N2          1-06 *  O  UBA5 S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       45
BUS I/O DATA CYCLE     -H  1B19N2          1-07 *  O  UBA5 S M8619 O      1         0  -800              N  1     >1 SOURCE       45
BUS I/O DATA CYCLE     -H  1B18N2          1-08 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM      45
BUS I/O DATA CYCLE     -H                  1                                    -1038 -5111     0     0     9-0/8                 45
BUS MEM BUSY           -H  1B09F2          1-01 *  T  MMCB S M8618 T      2      -520  -557              N  3     EXTRA TERM      46
BUS MEM BUSY           -H  1B14F2          1-02 *     DPMC S M8621        1         2     3              N  1                     46
BUS MEM BUSY           -H  1B15F2          1-03 *  O  UBAC S M8619 O      2         0  -800              N  1     >1 SOURCE       46
BUS MEM BUSY           -H  1B16F2          1-04 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       46
BUS MEM BUSY           -H  1B17F2          1-05 *  O  UBAC S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       46
BUS MEM BUSY           -H  1B19F2          1-06 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       46
BUS MEM BUSY           -H  1B18F2          1-07 *  T  CSL8 S M8616 T             -504  -556                       EXTRA TERM      46
BUS MEM BUSY           -H                  1                                    -1022 -4310     0     0     8-4/8                 46
```

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG        C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS PARITY LEFT        -H  1B09S2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      47
BUS PARITY LEFT        -H  1B14S2          1-02 *  O  DPM8 S M8621 O      1         0  -800              N  1     >1 SOURCE       47
BUS PARITY LEFT        -H  1B15S2          1-03 *  O  UBAC S M8619 O      2         0  -800              N  1     >1 SOURCE       47
BUS PARITY LEFT        -H  1B16S2          1-04 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       47
BUS PARITY LEFT        -H  1B17S2          1-05 *  O  UBAC S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       47
BUS PARITY LEFT        -H  1B19S2          1-06 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       47
BUS PARITY LEFT        -H  1B18S2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM      47
BUS PARITY LEFT        -H                  1                                    -1040 -5114     0     0     8-4/8                 47
BUS PARITY RIGHT       -H  1F09D2          1-01 *  T  MMC1 S M8618 T      2      -520  -557              N  3     EXTRA TERM      48
BUS PARITY RIGHT       -H  1F14D2          1-02 *  O  DPM9 S M8621 O      1         0  -800              N  1     >1 SOURCE       48
BUS PARITY RIGHT       -H  1F15D2          1-03 *  O  UBAC S M8619 O      2         0  -800              N  1     >1 SOURCE       48
BUS PARITY RIGHT       -H  1F16D2          1-04 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       48
BUS PARITY RIGHT       -H  1F17D2          1-05 *  O  UBAC S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       48
BUS PARITY RIGHT       -H  1F19D2          1-06 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       48
BUS PARITY RIGHT       -H  1F18D2          1-07 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM      48
BUS PARITY RIGHT       -H                  1                                    -1040 -5114     0     0     8-4/8                 48
BUS PI REQ 1           -H  1F13E1          1-01 *  O  DPEB S M8620 O      1         2   -76              N  1-6/8 >1 SOURCE       49
BUS PI REQ 1           -H  1F15E2          1-02 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       49
BUS PI REQ 1           -H  1F16E2          1-03 *  O  UBA3 S M8619 O      1         0   -80              N  1     >1 SOURCE       49
BUS PI REQ 1           -H  1F17E2          1-04 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       49
BUS PI REQ 1           -H  1F18E2          1-05 *     CSLA S M8616        1         2     3              N  1                     49
BUS PI REQ 1           -H  1F19E2          1-06 *  O  UBA3 S M8619 O                0   -80                       >1 SOURCE       49
BUS PI REQ 1           -H                  1                                        4  -393     0     0     5-6/8 NO HI DRV       49
BUS PI REQ 2           -H  1F13F1          1-01 *  O  DPEB S M8620 O      1         2   -76              N  1-6/8 >1 SOURCE       50
BUS PI REQ 2           -H  1F15F2          1-02 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       50
BUS PI REQ 2           -H  1F16F2          1-03 *  O  UBA3 S M8619 O      1         0   -80              N  1     >1 SOURCE       50
BUS PI REQ 2           -H  1F17F2          1-04 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       50
BUS PI REQ 2           -H  1F18F2          1-05 *     CSLA S M8616        1         2     3              N  1                     50
BUS PI REQ 2           -H  1F19F2          1-06 *  O  UBA3 S M8619 O                0   -80                       >1 SOURCE       50
BUS PI REQ 2           -H                  1                                        4  -393     0     0     5-6/8 NO HI DRV       50
BUS PI REQ 3           -H  1F13H1          1-01 *  O  DPEB S M8620 O      1         2   -76              N  1-6/8 >1 SOURCE       51
BUS PI REQ 3           -H  1F15H2          1-02 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       51
BUS PI REQ 3           -H  1F16H2          1-03 *  O  UBA3 S M8619 O      1         0   -80              N  1     >1 SOURCE       51
BUS PI REQ 3           -H  1F17H2          1-04 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       51
BUS PI REQ 3           -H  1F18H2          1-05 *     CSLA S M8616        1         2     3              N  1                     51
BUS PI REQ 3           -H  1F19H2          1-06 *  O  UBA3 S M8619 O                0   -80                       >1 SOURCE       51
BUS PI REQ 3           -H                  1                                        4  -393     0     0     5-6/8 NO HI DRV       51
BUS PI REQ 4           -H  1F13J1          1-01 *  O  DPEB S M8620 O      1         2   -76              N  1-6/8 >1 SOURCE       52
BUS PI REQ 4           -H  1F15J2          1-02 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       52
BUS PI REQ 4           -H  1F16J2          1-03 *  O  UBA3 S M8619 O      1         0   -80              N  1     >1 SOURCE       52
BUS PI REQ 4           -H  1F17J2          1-04 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       52
BUS PI REQ 4           -H  1F18J2          1-05 *     CSLA S M8616        1         2     3              N  1                     52
BUS PI REQ 4           -H  1F19J2          1-06 *  O  UBA3 S M8619 O                0   -80                       >1 SOURCE       52
BUS PI REQ 4           -H                  1                                        4  -393     0     0     5-6/8 NO HI DRV       52
```

```
KS10 PROTOTYPE REV D/.2  WRP288.V34(62)-1 31-Jul-75                                   18-Jan-78      12:30
RUN NAME              A/P  PIN     ORDER   BAY -   Q  DRAW   MODULE TYPE  Z  C  LOADING                 NC LENGTH EXCEPTIONS     RUN
                           NAME     PIN    ORDER          OPT    FLAG        C   HIGH   LOW    AC EXTRA FLAG                  NUMBER
BUS PI REQ 5           -H  1F13K1          1-01 *  O  DPEB S M8620 O      1         2   -76              N  1-6/8 >1 SOURCE       53
BUS PI REQ 5           -H  1F15K2          1-02 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       53
BUS PI REQ 5           -H  1F16K2          1-03 *  O  UBA3 S M8619 O      1         0   -80              N  1     >1 SOURCE       53
BUS PI REQ 5           -H  1F17K2          1-04 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       53
BUS PI REQ 5           -H  1F18K2          1-05 *     CSLA S M8616        1         2     3              N  1                     53
BUS PI REQ 5           -H  1F19K2          1-06 *  O  UBA3 S M8619 O                0   -80                       >1 SOURCE       53
BUS PI REQ 5           -H                  1                                        4  -393     0     0     5-6/8 NO HI DRV       53
BUS PI REQ 6           -H  1F13L1          1-01 *  O  DPEB S M8620 O      1         2   -76              N  1-6/8 >1 SOURCE       54
BUS PI REQ 6           -H  1F15L2          1-02 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       54
BUS PI REQ 6           -H  1F16L2          1-03 *  O  UBA3 S M8619 O      1         0   -80              N  1     >1 SOURCE       54
BUS PI REQ 6           -H  1F17L2          1-04 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       54
BUS PI REQ 6           -H  1F18L2          1-05 *     CSLA S M8616        1         2     3              N  1                     54
BUS PI REQ 6           -H  1F19L2          1-06 *  O  UBA3 S M8619 O                0   -80                       >1 SOURCE       54
BUS PI REQ 6           -H                  1                                        4  -393     0     0     5-6/8 NO HI DRV       54
BUS PI REQ 7           -H  1F13M1          1-01 *  O  DPEB S M8620 O      1         2   -76              N  1-6/8 >1 SOURCE       55
BUS PI REQ 7           -H  1F15M2          1-02 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       55
BUS PI REQ 7           -H  1F16M2          1-03 *  O  UBA3 S M8619 O      1         0   -80              N  1     >1 SOURCE       55
BUS PI REQ 7           -H  1F17M2          1-04 *  O  UBA3 S M8619 O      2         0   -80              N  1     >1 SOURCE       55
BUS PI REQ 7           -H  1F18M2          1-05 *     CSLA S M8616        1         2     3              N  1                     55
BUS PI REQ 7           -H  1F19M2          1-06 *  O  UBA3 S M8619 O                0   -80                       >1 SOURCE       55
BUS PI REQ 7           -H                  1                                        4  -393     0     0     5-6/8 NO HI DRV       55
BUS RESET              -H  1F09P2          1-01 *  T  MMCB S M8618 T      1      -520  -557              N  3-4/8 EXTRA TERM      56
BUS RESET              -H  1F15P2          1-02 *  O  UBAC S M8619 O      2         0  -800              N  1     >1 SOURCE       56
BUS RESET              -H  1F16P2          1-03 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       56
BUS RESET              -H  1F17P2          1-04 *  O  UBAC S M8619 O      2         0  -800              N  1-4/8 >1 SOURCE       56
BUS RESET              -H  1F19P2          1-05 *  O  UBAC S M8619 O      1         0  -800              N  1     >1 SOURCE       56
BUS RESET              -H  1F18P2          1-06 *  T  CSL8 S M8616 T             -520  -557                       EXTRA TERM      56
BUS RESET              -H                  1                                    -1040 -4314     0     0     8-0/8                 56
```

INDEX

```
7600xx=csr-read  . . . . . . . . . . . . . . . . . . . . . . . . . 3-12
7600xx=csr-wrt . . . . . . . . . . . . . . . . . . . . . . . . . . 3-12
7600xx=lpr-write . . . . . . . . . . . . . . . . . . . . . . . . . 3-12
7600xx=msr-read  . . . . . . . . . . . . . . . . . . . . . . . . . 3-13
7600xx=rbuf-read . . . . . . . . . . . . . . . . . . . . . . . . . 3-12
7600xx=tcr-read  . . . . . . . . . . . . . . . . . . . . . . . . . 3-13
7600xx=tcr-write . . . . . . . . . . . . . . . . . . . . . . . . . 3-13
7600xx=tdr-write . . . . . . . . . . . . . . . . . . . . . . . . . 3-13

8080-communication-area  . . . . . . . . . . . . . . . . . . . . . 1-11
8080-console-error-codes . . . . . . . . . . . . . . . . . . . . . 1-6
8080-error-messages-during-bootcheck . . . . . . . . . . . . . . . 1-9
8080-error-messages-during-booting . . . . . . . . . . . . . . . . 1-8

Aprid=700000 . . . . . . . . . . . . . . . . . . . . . . . . . . . 2-7

Cd11-bit-format  . . . . . . . . . . . . . . . . . . . . . . . . . 3-15
Clrpt=70110  . . . . . . . . . . . . . . . . . . . . . . . . . . . 2-9
Console  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1-1
Console-board-uarts-switch-configurations  . . . . . . . . . . . . 3-3
Console-data-reg . . . . . . . . . . . . . . . . . . . . . . . . . 6-2
Console-register-contents  . . . . . . . . . . . . . . . . . . . . 1-12
Contents . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
Csl-commands . . . . . . . . . . . . . . . . . . . . . . . . . . . 1-2

Ddt-command's  . . . . . . . . . . . . . . . . . . . . . . . . . . 5-1
Description-deposit-cram-from-console  . . . . . . . . . . . . . . 6-6
Description-deposit-data-into-an-i/o-register-from-the-console . . 6-7
Description-deposit-memory-from-console  . . . . . . . . . . . . . 6-8
Description-deposit-the-bus-from-the-console . . . . . . . . . . . 6-5
Description-examine-cram-from-console  . . . . . . . . . . . . . . 6-11
Description-examine-memory-from-console  . . . . . . . . . . . . . 6-9
Description-execute-ks10-instruction-from-the-console  . . . . . . 6-10
Description-ks10-bus-arbitrator  . . . . . . . . . . . . . . . . . 6-1
Description-ks10-cache-hit . . . . . . . . . . . . . . . . . . . . 6-4
Description-parity-generation-and-checking-networks-in-the-ks10  . 6-3
Device-addresses-and-vector-assignments-for-ks10 . . . . . . . . . 3-1
Dispatch-rom-definitions . . . . . . . . . . . . . . . . . . . . . 4-3
Drive-command-function-codes . . . . . . . . . . . . . . . . . . . 3-4
Dup11-bit-format . . . . . . . . . . . . . . . . . . . . . . . . . 3-16
Dz-control-register-bits . . . . . . . . . . . . . . . . . . . . . 3-11

Ec-command-breakdown . . . . . . . . . . . . . . . . . . . . . . . 1-18
Ecc(mos-memory-error-correction)bits . . . . . . . . . . . . . . . 2-17
Error-messages-out-of-pre-boot . . . . . . . . . . . . . . . . . . 1-10
External-registers . . . . . . . . . . . . . . . . . . . . . . . . 2-13
External-registers-bit-formats . . . . . . . . . . . . . . . . . . 2-14

Halt-status-block  . . . . . . . . . . . . . . . . . . . . . . . . 4-1
Halt-status-word . . . . . . . . . . . . . . . . . . . . . . . . . 2-2
```

I/o-instruction-set 2-17
Internal-register-bit-formats 2-6

Ks10-apr-io-instructions-ac-field-assignments 2-6
Ks10-i/o-instruction-format 2-5
Ks10-icroword-bits-sorted-by-physical-bit 4-4
Ks10-io-instruction-op-codes-in-octal 2-6
Ks10-micro-word-bits-sorted-by-function 4-7
Ks10-options-jumper-and-switch-configurations 3-2
Ks10-processor-status-words 2-1

Lp20/lp05/lp14-bit-format 3-13

Magtape-register-bits-in-plain-english 3-9
Memory-status-register 2-17
Microcode-fields-listing-format 4-2
Microcode-flags 2-2

Page-fail-word(or-map-ac) 2-4
Paging 2-4
Pc-word 2-3

Rd-i/o=200 1-18
Rd-i/o=202 1-18
Rdapr(coni-apr)=70024 2-7
Rdcsb=70204 2-10
Rdcstm=70214 2-11
Rdebr=70124 2-9
Rdhsb=70230 2-13
Rdint=70224 2-12
Rdio(712) 2-16
Rdio(712)-ac,7630xx 2-15
Rdpi=700640 2-8
Rdpur=70210 2-10
Rdspb=70200 2-9
Rdtime=70220 2-12
Rdubr=70104 2-8
Read-i/o=0 1-13
Read-i/o=1 1-13
Read-i/o=100 1-13
Read-i/o=101 1-14
Read-i/o=102 1-14
Read-i/o=103 1-14
Read-i/o=2 1-13
Read-i/o=3 1-13
Read-i/o=300 1-14
Read-i/o=301 1-14
Read-i/o=302 1-15
Read-i/o=303 1-15
Rh11-rm03-register-bit-format 3-7
Rh11-rp04/5/6-register-bit-format 3-5
Rh11/tm02/tm03-register-bit-format 3-9
Rm03/rp06/rh11-register-bits-in-plain-english 3-8

Uba-maintenance-reg 2-16

Uba-paging-ram 2-15
Uba-status-register 2-16
Unibus-adapter-mini-check 3-18
Unibus-signal-pin-finder 2-18

Vma 2-3

Wire-wrap-list-backplane 7-10
Wrapr(cono-apr)=700200 2-7
Wrcsb=70244 2-10
Wrcstm=70254 2-11
Wrebr=70120 2-9
Wrhsb=70270 2-13
Wrint=70264 2-12
Wrio(713) 2-16
Wrio(713)-ac,7630xx 2-15
Wrpi(cono-pi)=700600 2-8
Wrpur=70250 2-10
Wrspb=70240 2-9
Wrt-i/o=200 1-18
Wrt-i/o=202 1-18
Wrt-i/o=204 1-15
Wrt-i/o=205 1-15
Wrt-i/o=206 1-15
Wrt-i/o=210 1-16
Wrt-i/o=212 1-16
Wrt=100 1-16
Wrt=101 1-16
Wrt=102/103 1-16
Wrt=104/105 1-17
Wrt=106/107 1-17
Wrt=110/111 1-17
Wrt=112/113 1-17
Wrt=114/115 1-17
Wrt=116 1-18
Wrtime=70260 2-11
Wrubr=70114 2-8