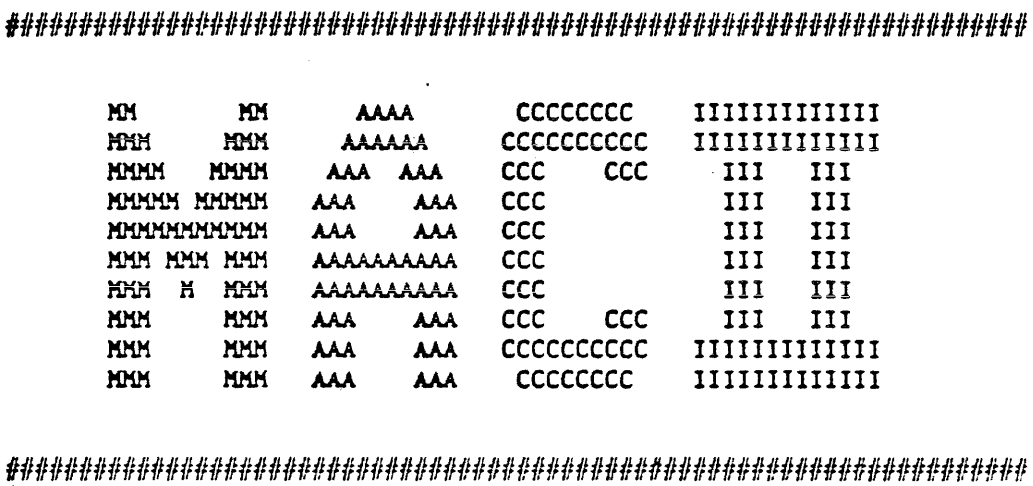
B1990 UPDATE SEMINAR



B1990 UPDATE SEMINAR

INTRODUCTION

April 85

B1990 UPDATE SEMINAR

REASON FOR CHANGE

CASSETTE IS OBSOLETE AND GIVES MANY PROBLEMS

CASSETTE IS MORE EXPENSIVE THAN FLEXIBLE DISK

AUTOMATIC RESTART CAPABILITY WITH FLEXIBLE DISK

MANY MTR'S ON ONE FLEXIBLE DISK

MAIN CHANGES

CABINET : NEW INTERCONNECT BOARD

NEW PANEL CASSETTE SUPPORT

BOX DRIVE (TO HOLD FLEXIBLE DISK DRIVE)

CARD H10: 2 PROMS DELETED - 1K ROM left from 3

64K RAM ADDED (70ns)

CLOCK CIRCUIT HAS BEEN MODIFIED

8080 HAS BEEN REPLACED BY Z80

2meg 4meg - left from

SELF-TEST: ACCORDING TO CARD REPLACEMENT PHILOSOPHY

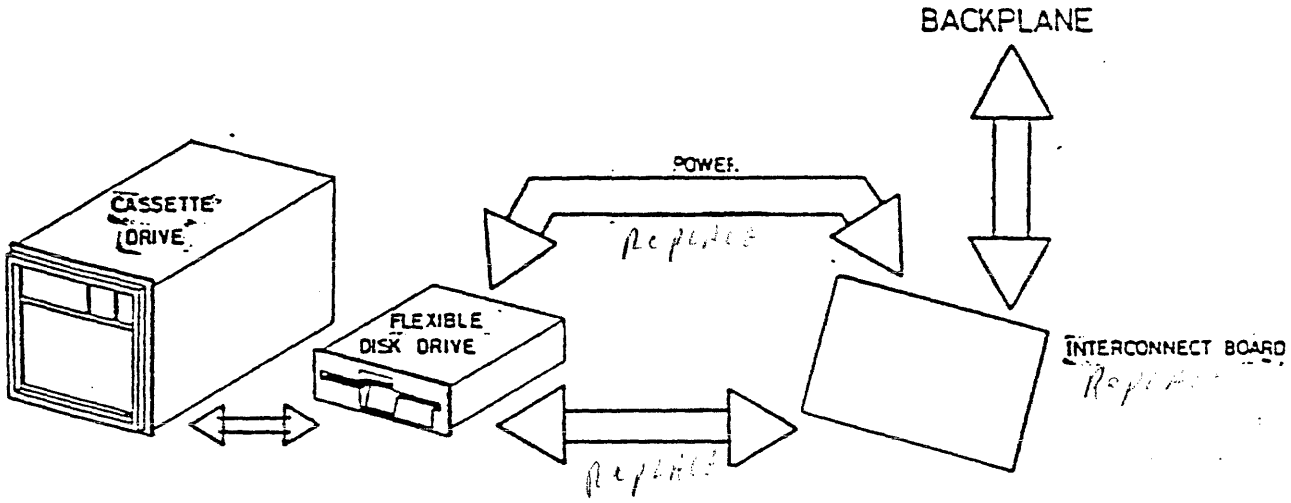
MOST OF THE FEATURES FOR MANUAL TESTING

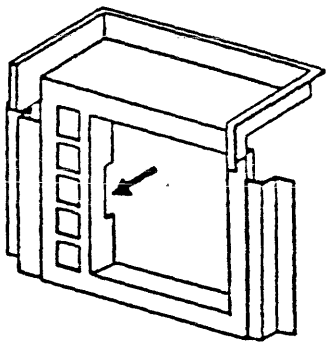
HAVE BEEN DELETED.

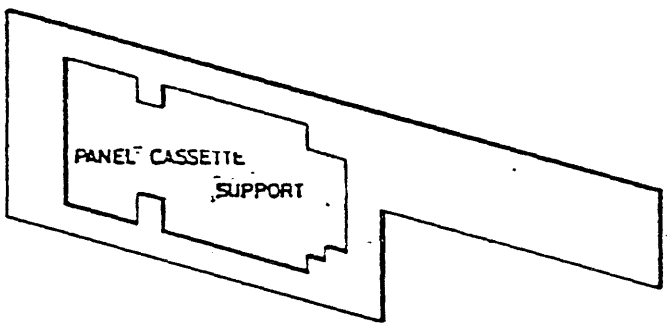
SOFT-CON.: CHANGE TO EXISTING COMMANDS (TAPE, MTR,REWIND)

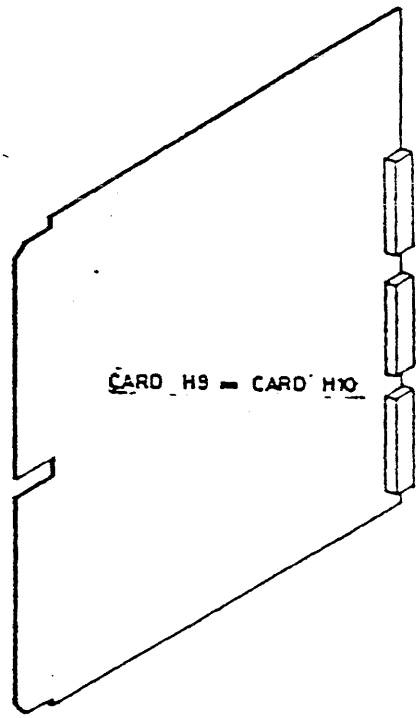
NEW ODT COMMANDS (DIR, AUTO)

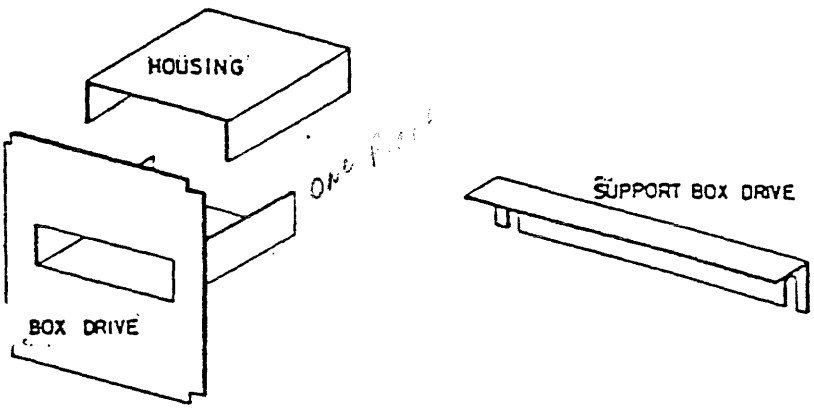
B1990 UPDATE SEMINAR





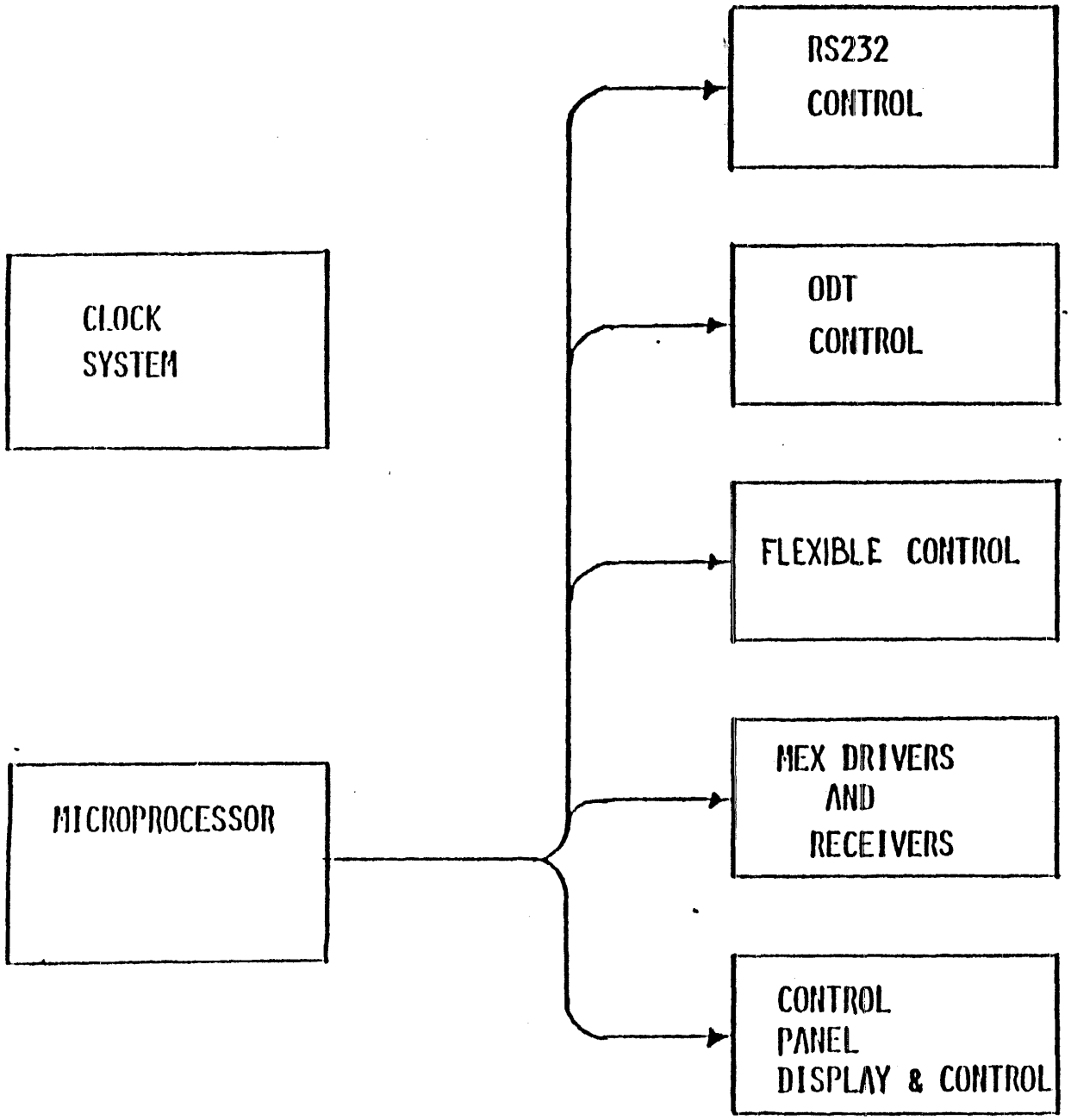






MAINTENANCE ACCESS CARD MODEL II (MAC II)

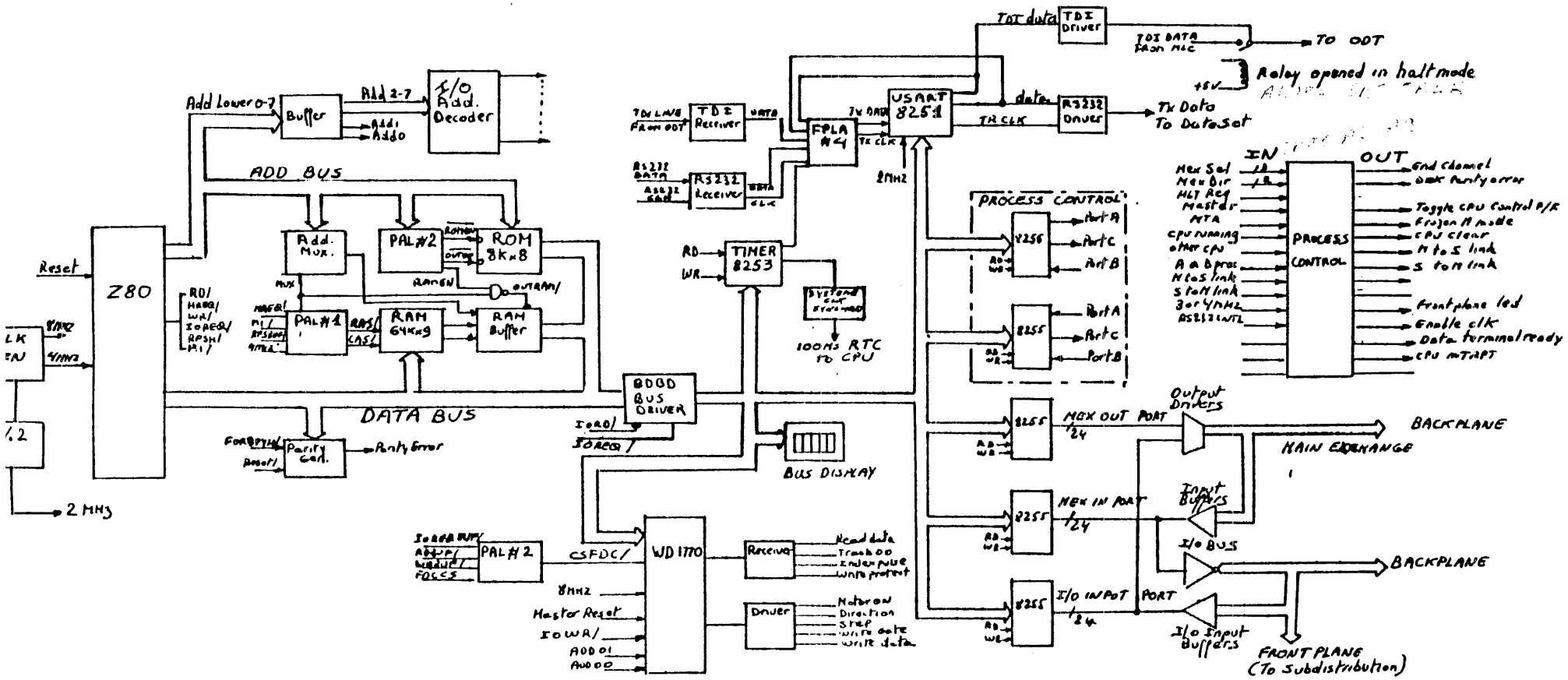
CASSETTE DRIVE REPLACEMENT BYE A FLEXIBLE DISK DRIVE



H.10

GENERAL BLOCK DIAGRAM

MACII BLOCK DIAGRAM



B1990 UPDATE SEMINAR

THEORY OF OPERATION

April 85

USE OF 5 1/4 INCH DISKS ON B1965/95 MAC.

The diskettes to be used on the new maintenance access card will be 800 Kbytes 5 1/4 inch TP400 disks (as formatted on an ET2000 processor), and will be driven by a WD1770 LSI.

The media characteristics will be as follows :

Sector size	512 bytes
Total size.	1600 sectors
Number of cylinders . .	80
Number of surfaces. . .	2
Sectors per track . . .	10

The software formatting may be summarized as follows :

Sector 0	: Reserved area.
Sector 1	: First copy of file allocation table (FAT), (341 entries of 12 bits).
Sector 2	: Second copy of file allocation table.
Sector 3 - 11	: 144 directory entries (32 bytes/entry), each entry contains (among other items) : - file identifier (8 bytes), - file id. extension (3 bytes), - first cluster address (see below).
Sector 12 - 1599	: Data area, divided into several "CLUSTERS" - allocation units of 8 consecutive physical sectors.

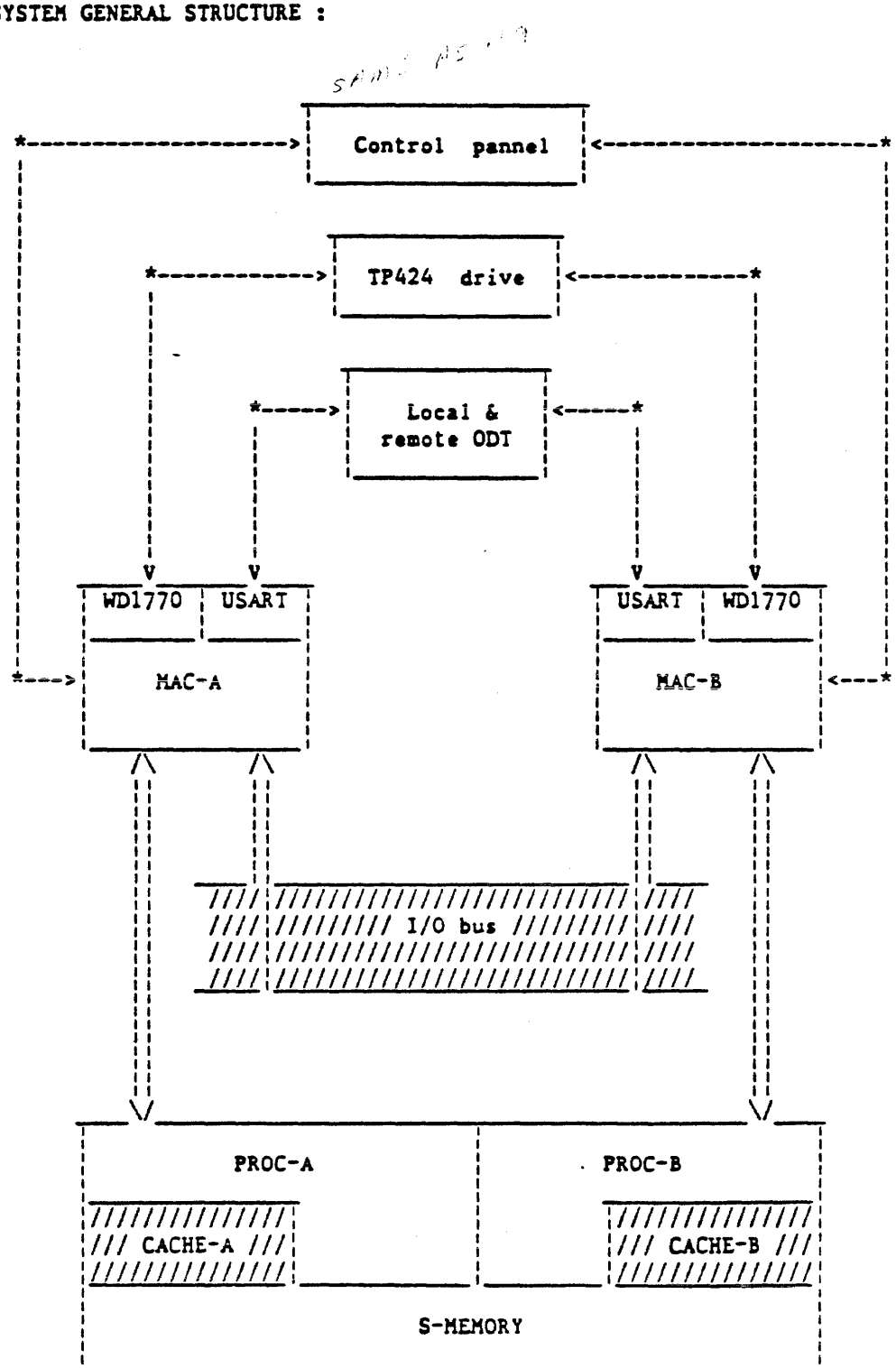
Each cluster belonging to a file is linked to the next one (if any) via the file allocation table (FAT).

There will be 3 types of files present on the diskette :

- xxxxxxxx.SYS files : containing H firmware.
- xxxxxxxx.MTR files : data files to be transferred to the host via the "U" register.
- xxxxxxxx.EXE files : programs executable by the H card (via the "EX" command).

B1990 UPDATE SEMINAR

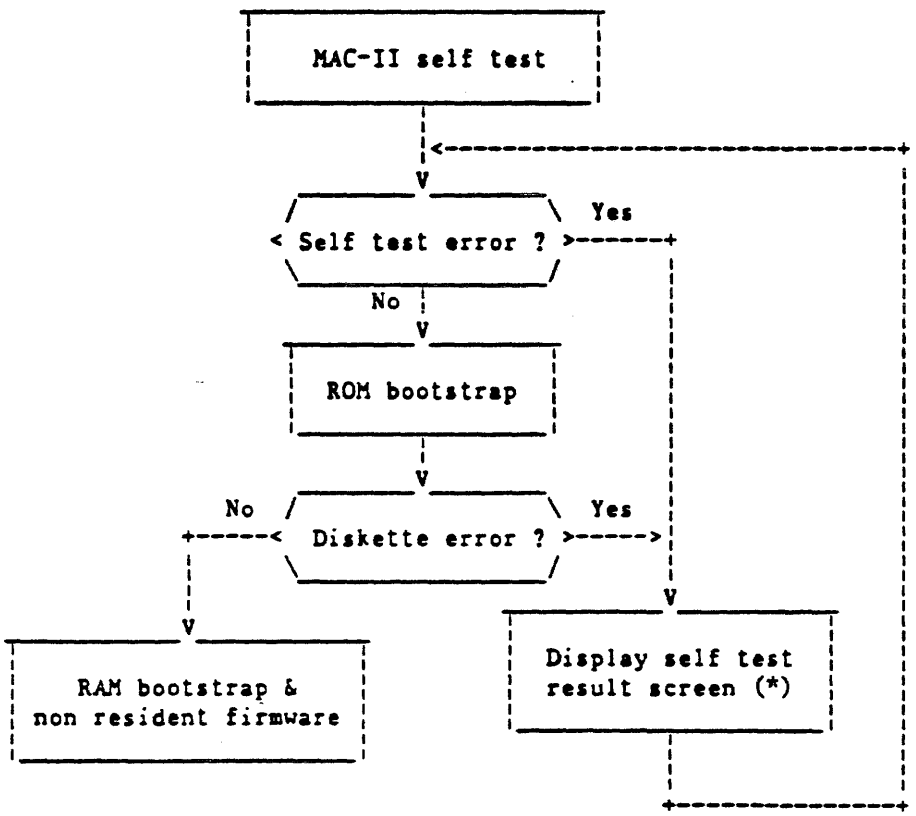
SYSTEM GENERAL STRUCTURE :



MAC-II FIRMWARE STARTUP :

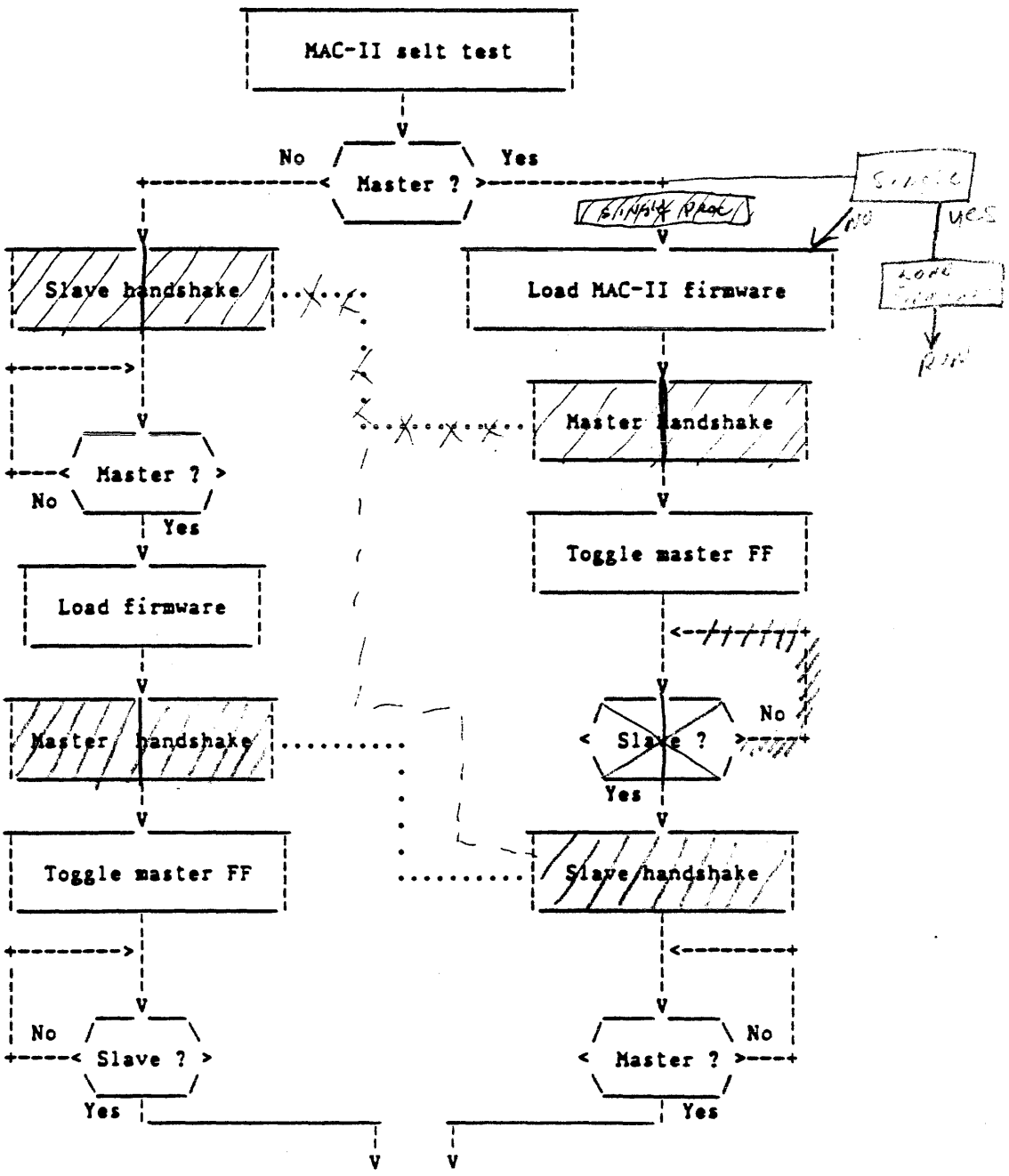
As mentionned above, the whole MAC firmware will no longer be resident in ROM, but will rather be bootstrap'ed from disk after completion of the self test, provided a suitable disk was inserted in the drive when the firmware is started up.
Since the ODT handling module is overlayed on disk, there needs to be an appropriate procedure to display any exeception condition that occurs while loading the RAM firmware.

These functions will be performed as follows :

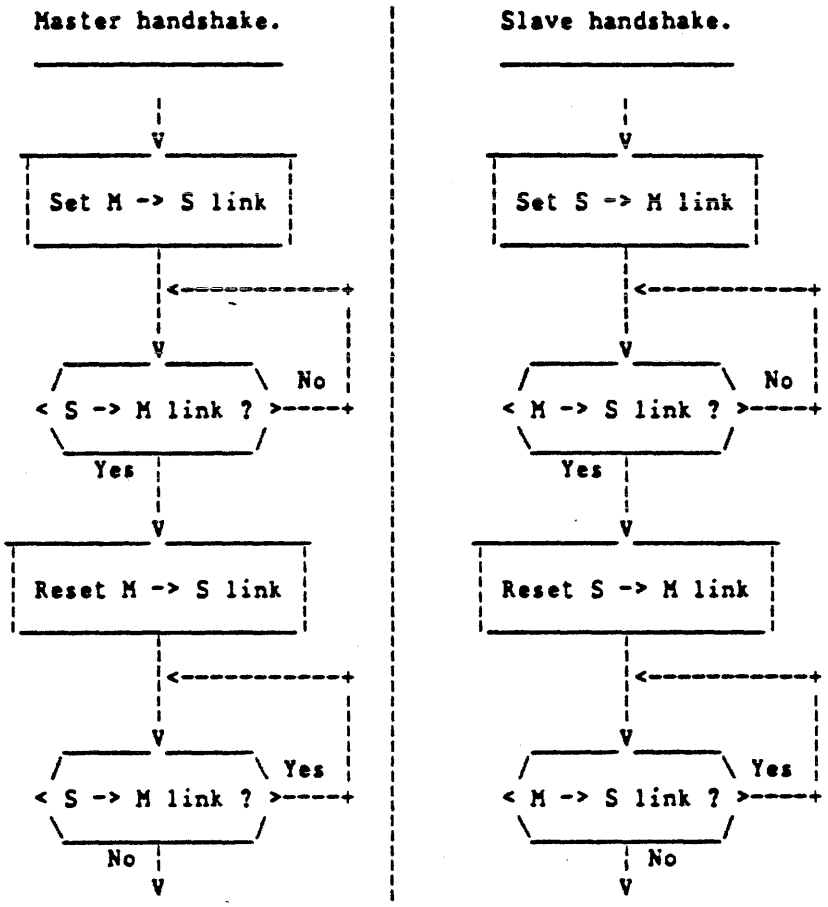


(*) : Contains information on the diskette availability.

Since only the master processor may access the disk driver chip, the above procedure has to be executed 2 times sequentially by each one of the A/B processor, according to the following flow :



B1990 UPDATE SEMINAR



B1990 UPDATE SEMINAR

MAC-II I/O DEVICES :

The firmware will no longer access peripherals using an I/O memory map as on the previous H9 card, exercisable via moves to/from M micro instructions, but rather issue IN/OUT micros (@DBxx / @D3xx) on corresponding I/O ports (xx = address).

The following conversion table applies to this change :

Peripheral	Type	MAC-I Memory map addresses	MAC-II Port addresses
PPI1	8255	@7580 - @7583	@90 - @93
PPI2	8255	@7600 - @7603	@94 - @97
IPPI	8255	@7680 - @7683	@88 - @8B
HPPI	8255	@7700 - @7703	@84 - @87
MPPI	8255	@7780 - @7783	@80 - @83
USART	8251	@7500 - @7503	@B0 - @B3
PTMR	8253	@7480 - @7483	@A0 - @A3

In addition, the diskette replacing the old-fashion cassette drive will no longer be accessible via the USART, according to "MODE SELECT" in PPI1c, but via a specific disk controller, named WD1770, to which the I/O port addresses @C0 - @C3 will be assigned, while the diskette side select function will be execised by an OUT micro on port address @D0.

The new MAC-II error display LED's will be driven by the firmware using the I/O port address @E0.

CHANGES TO EXISTING ODT COMMANDS :

(1) TAPE -----> LOAD "< file name >"

(2) MTR -----> MTR "< file name >"

(3) REWIND -----> UNLOAD

Where < file name > is an 8 characters alpha string.

- (1) Will place the processor in MTR mode, search for "< file name >.MTR" in the directory of the diskette already loaded in the RAM of the H card.
Any subsequent "GO" command will cause the first cluster of the file to be read (if it can be found), and the "U" register of the host to be filled with the data read from the RAM buffers.

- (2) In addition to known features of the MTR command, this command will cause FAT and directory to be reloaded from the diskette to RAM, "< file name >.MTR" to be searched. Afterwards, by means of a "GO" command, data from that file will be provided to the "U" register of the host.

Both commands (1) and (2) need an additional error message to handle unsuccessful search conditions.

If no < file name > is provided to commands (1) and (2), the file identifier stored in RAM will simply not be modified, and the previously existing one will be re-used for subsequent commands, after completion of the self test, the file identifier in RAM will be filled with the "CLRSTART" pattern.

- (3) Will cause stepping the head mechanism of the diskette to track zero position in order to allow removing the floppy disk safely.

NEW ODT COMMANDS :

(1) EX "<file name>" [for hypothetical future use]

(2) DIR

(3) AUTO ---+---> "<file name>" ---+-----+---+---> ON ---+---+---> OFF ---+

(1) will cause "< file name >.EXE" to be searched in the directory of the diskette, loaded into MACII memory, and executed as a firmware overlay.

(2) will cause the 128 first directory file identifiers of the diskette to be displayed on the ODT screen, free entries in the directory will be represented by a dot (".") pattern, all loadable files (.MTR files) will be highlighted.

(3) will cause the specified file name or the currently used file name (if not specified in the command) to be searched on the diskette, and, if found, this file identifier will be written at a specified location in the first directory entry, followed by a flag indicating the ON/OFF variant of the command (default is ON), according to the following format :

Start of directory.

Msb Lsb

0 ----------> 1 ----------> 2 ----------> 3 -->

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

Diskette pack id	Auto start identifier	SF	

SF = start flag, equal to @FF for "ON" variant,
@00 for "OFF" variant.

Any subsequent power-on/reset of the card will retrieve this file name, make it the current file name, and, if the "ON" variant was specified, initiate its execution.

A display of the file identifier being loaded/executed should be inserted in the menu section of the screen display, instead of the cassette status.

FIRMWARE IMPLEMENTATION :

The following firmware modules are required to assemble the "SYSO.SYS" aggregate file :

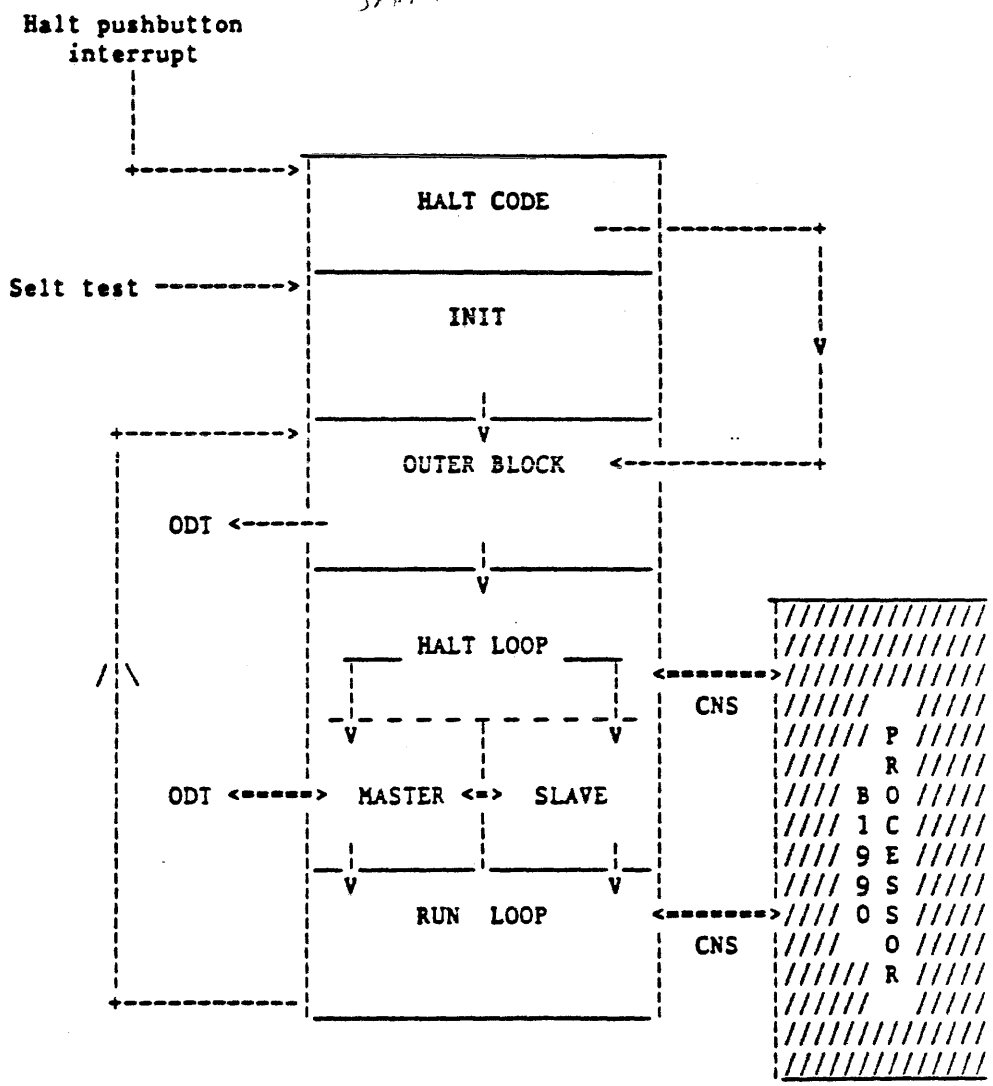
Name :	PLM	ASM	Function :
DISK	/////	*	WD1770 disk driver
MAIN	*	/////	Firmware loop executive
SCANNER. . . .	*	/////	ODT input analyzer
SCREEN	*	/////	ODT message formatter
SPO.	/////	*	Data comm driver for ODT's & slave <=> master
CPU.	/////	*	Handles communications between MAC and B1990
TAPE	/////	*	TP424 disk file handler
SCAN	/////	*	Subroutines of scanner

The "DISK" module will be loaded at memory byte address @2000, and will contain links to & from self test routines.

B1990 UPDATE SEMINAR

MAIN MODULE STRUCTURE :

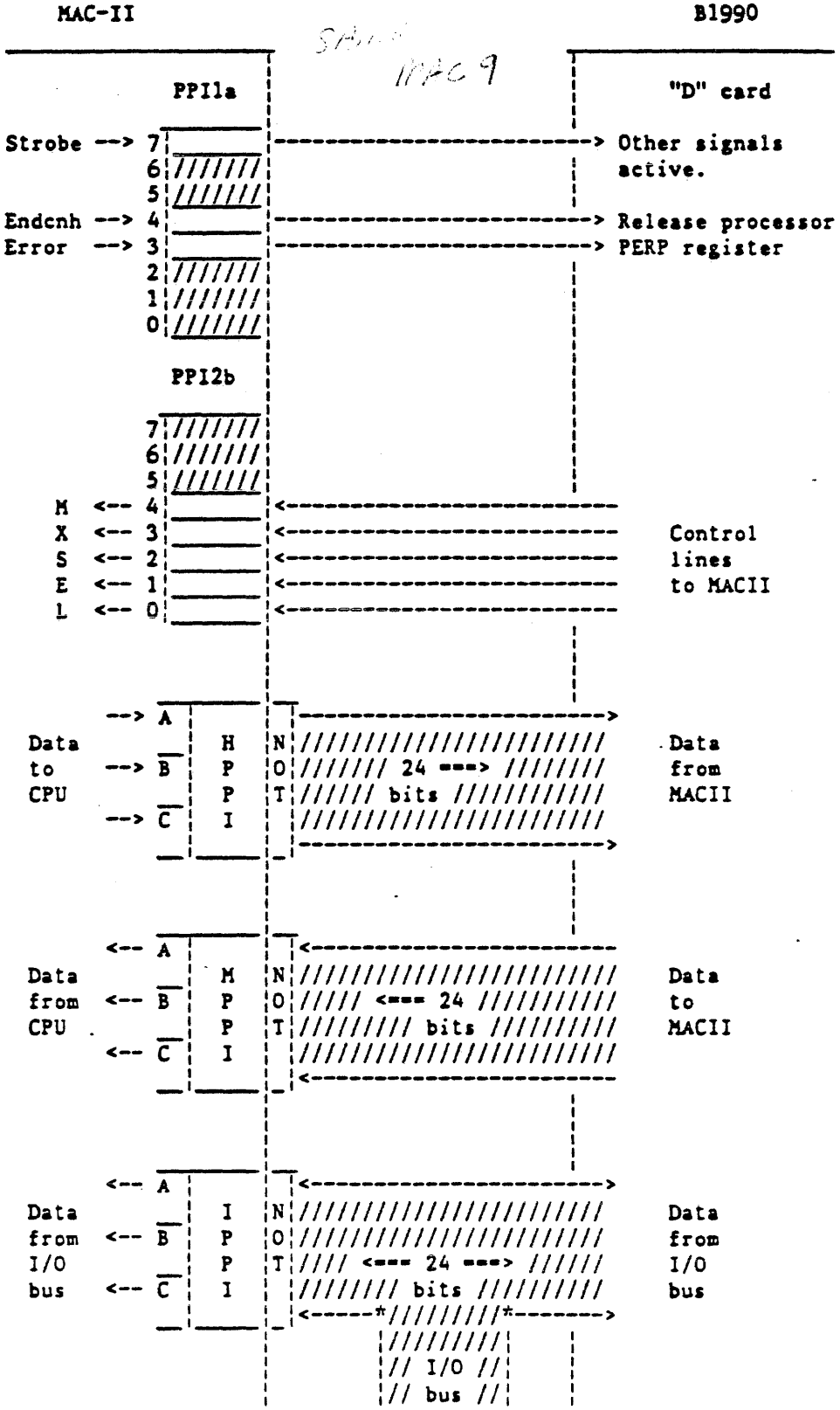
The "MAIN" module has 2 entry points, one from the self test after power-on or reset ("BEGPLM" label), the other from the interrupt @38 mechanism ("HALTON" label). See following block diagram :



The "INIT" block will initialize PPIs and default system options, branch to "OUTER BLOCK" which displays the selected screen and enter the halt loop which handles the ODT input.
The "HALT LOOP" will be exited whenever either the "GO" command is entered, the "RUN" pushbutton depressed.
The "RUN LOOP" will poll the MEX select lines waiting for a CNS command, and will exit to "OUTER BLOCK" when PPI1b run bit is off.
The drawing on next page shows the interface (data + control lines) between MAC-II and B1990 cpu.

B1990 UPDATE SEMINAR

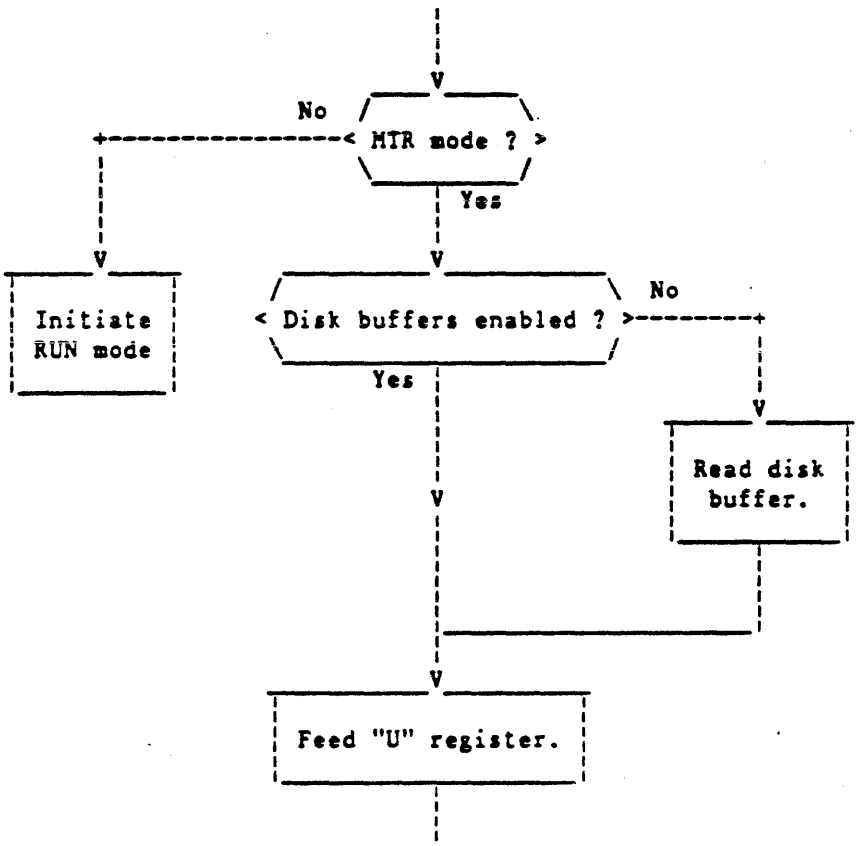
MAC-II TO B1990 CPU COMMUNICATION



SEMANTICS OF THE "GO" COMMAND :

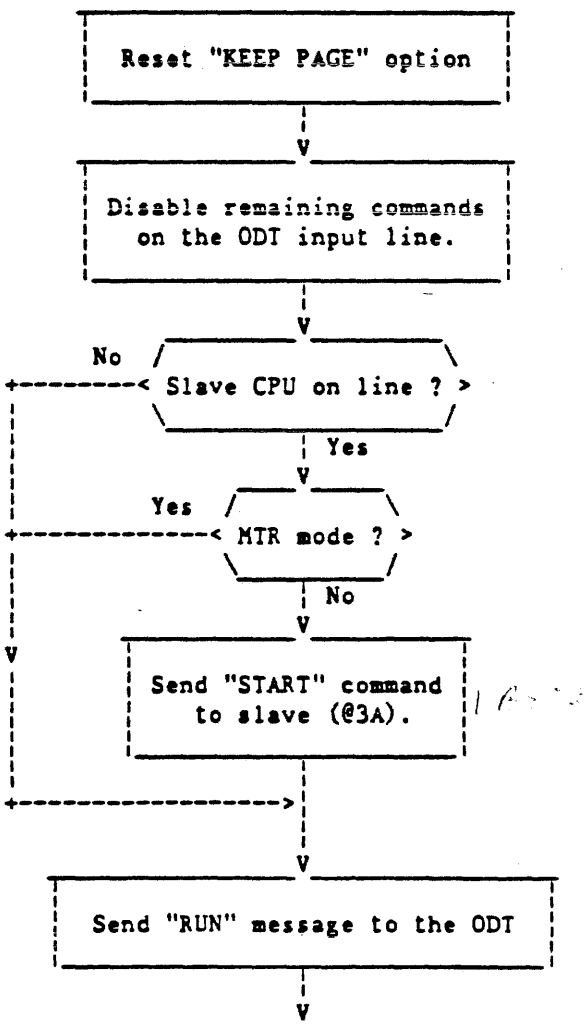
Any such command issued when the system is in MTR mode will cause the firmware to check whether the diskette read buffers have already been filled with valid data from a previously selected ".MTR" file, if not, data will be read from that file, starting from a sector address which logically follows the last used one (mainly the first one), afterwards the transfer to the "U" register will be enabled.

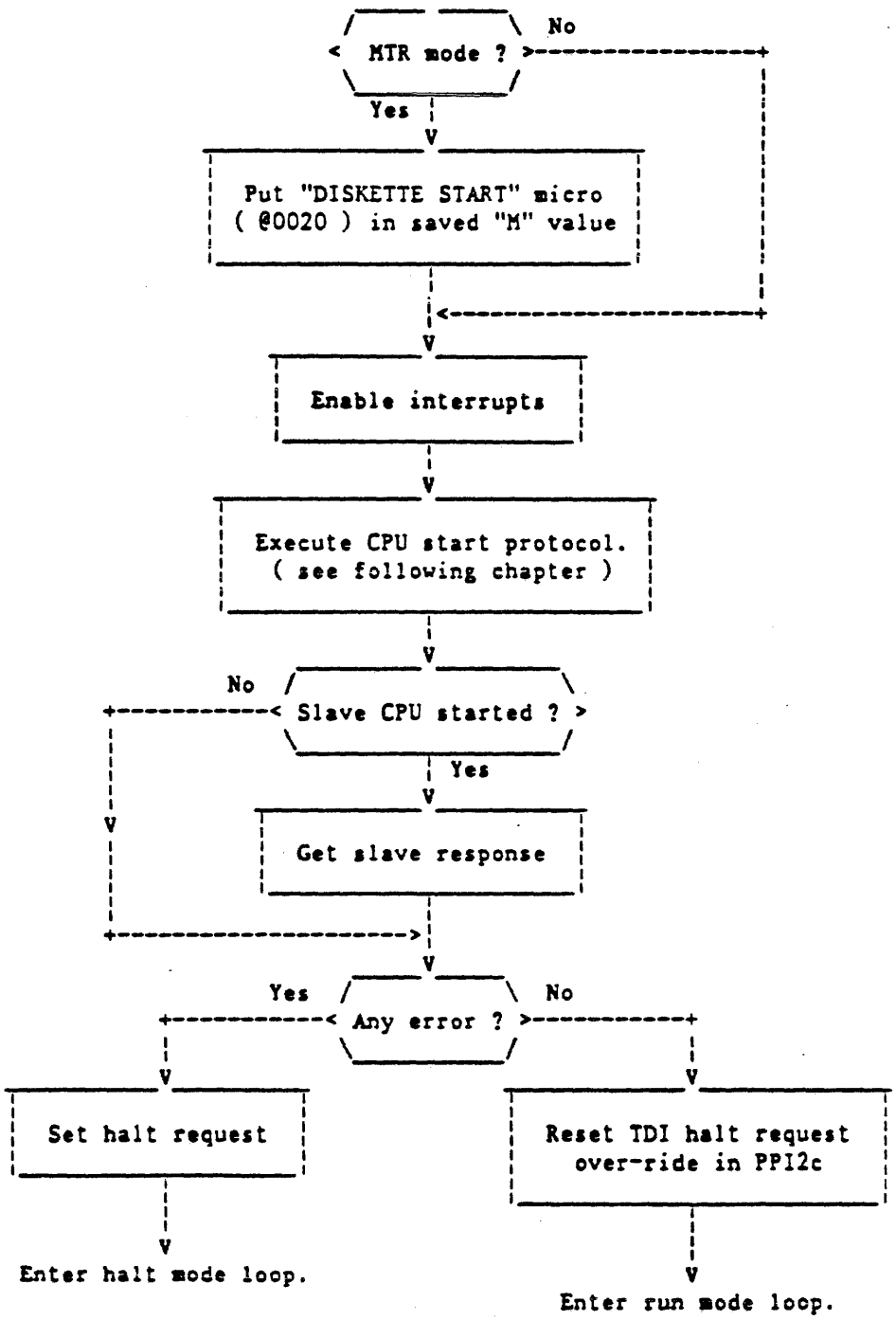
The following flow applies to the process described above :



The following graph describes more precisely the sequence of operations required to put both master/slave CPU's in run mode, when either

- a "GO" command is entered on the ODT,
- the "HALT/RUN" pushbutton is depressed in halted mode,
- a "HALT RESTART" request is made by the CPU.




MTR mode ?
No
Yes
Put "DISKETTE START" micro (@0020) in saved "M" value
Enable interrupts
Execute CPU start protocol. (see following chapter)
Slave CPU started ?
No
Yes
Get slave response
Any error ?
Yes
No
Set halt request
Reset TDI halt request over-ride in PPI2c
Enter halt mode loop.
Enter run mode loop.

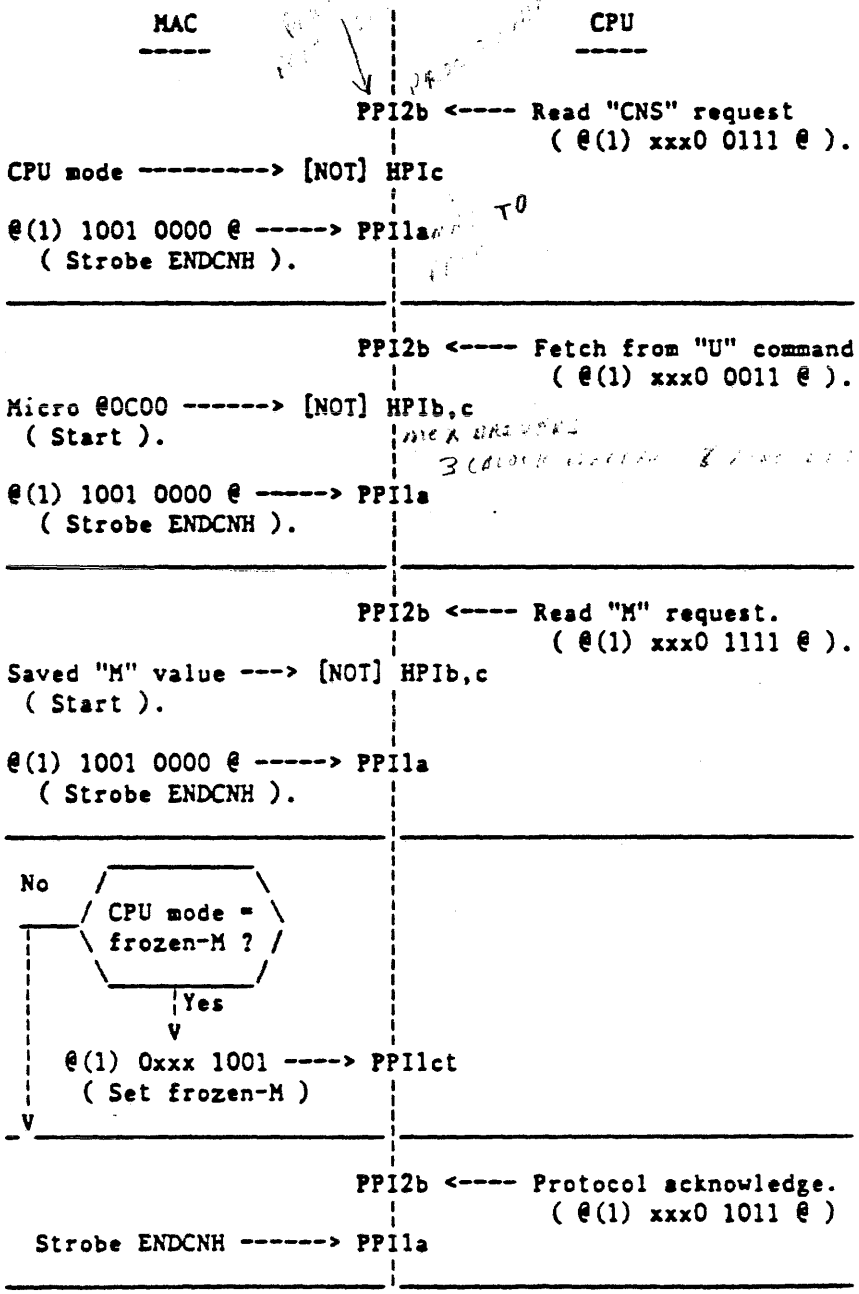
Same as #9

CPU START PROTOCOL :

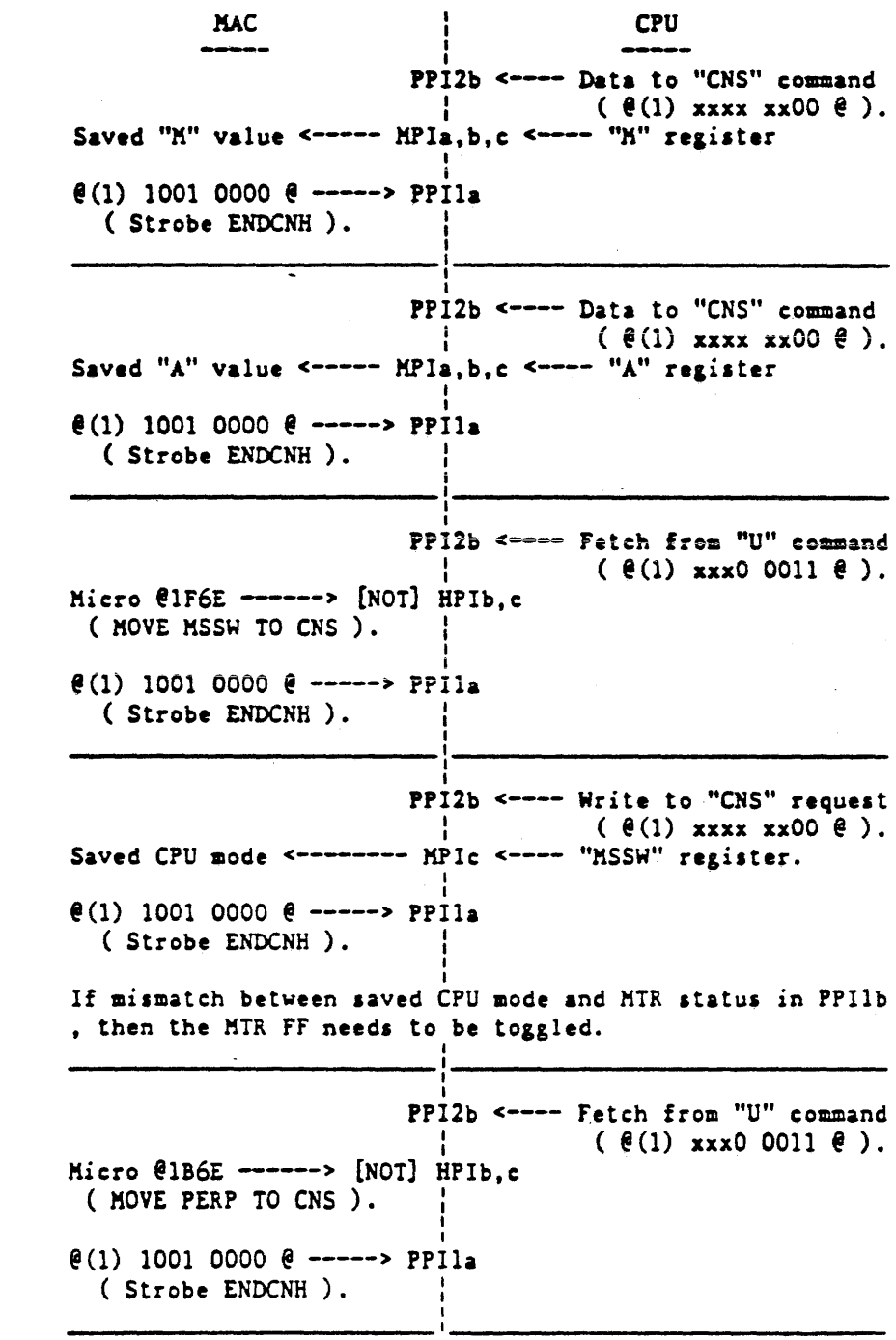
MAC		CPU
	PPI2b <----	Fetch from "U" command (@(1) xxx0 0011 @).
Micro @1FDB ------> [NOT] HPIb,c (Clear PERP).		
@(1) 1001 0000 @ -----> PPI1a (Strobe ENDCNH).		
	PPI2b <----	Fetch from "U" command (@(1) xxx0 0011 @).
Micro @1FDA ------> [NOT] HPIb,c (Clear PERM).		
@(1) 1001 0000 @ -----> PPI1a (Strobe ENDCNH).		
	PPI2b <----	Fetch from "U" command (@(1) xxx0 0011 @).
Micro @3F14 ------> [NOT] HPIb,c (Clear CD(0-1-3)).		
@(1) 1001 0000 @ -----> PPI1a (Strobe ENDCNH).		
Clear deferred command flag.		
@(1) 0xxx 1000 @ -----> PPI1ct (Reset frozen-M bit)		
	PPI2b <----	Fetch from "U" command (@(1) xxx0 0011 @).
Micro @1E9F ------> [NOT] HPIb,c (Move CNS to MSSW).		
@(1) 1001 0000 @ -----> PPI1a (Strobe ENDCNH).		

(continued next page)

CPU START PROTOCOL (continued) :

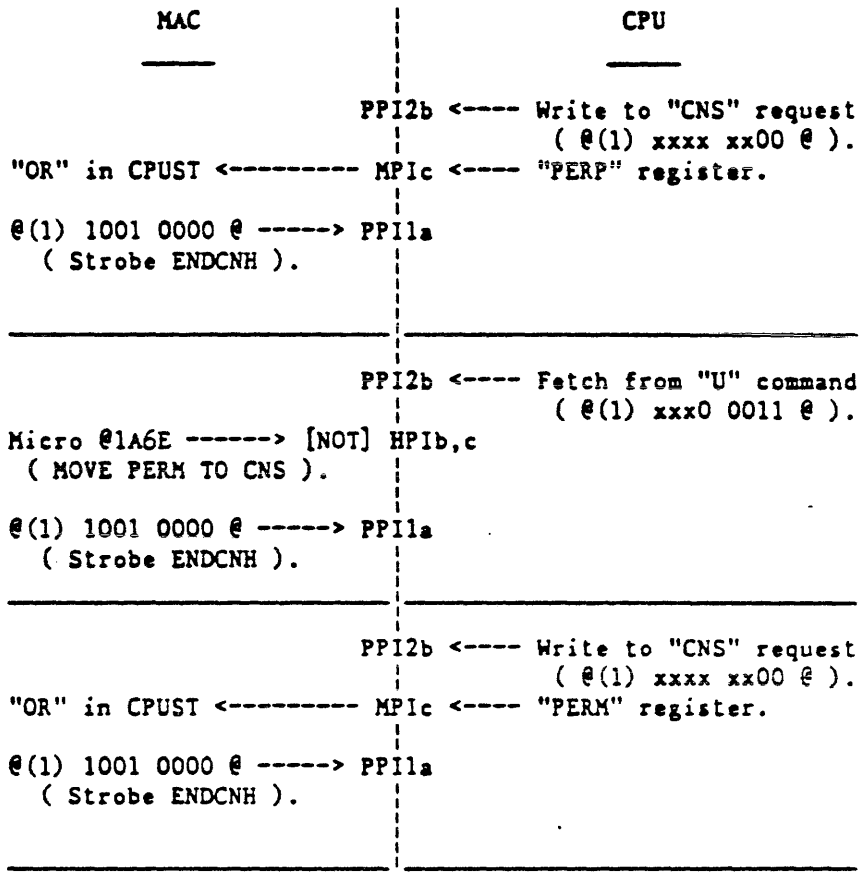


CPU HALT PROTOCOL :



(continued next page)

CPU HALT PROTOCOL (continued) :



The above procedure applies if either the CPU halts, or if the MAC-II is sending a "SYSTEM-CLEAR" command to the CPU in bit# 3 of PPI1c.

SEMANTICS OF THE "CLEAR" COMMAND :

The following sequence of operations will be performed by the MAC-II firmware of the master CPU upon receipt of a "CLEAR" command in halt mode :

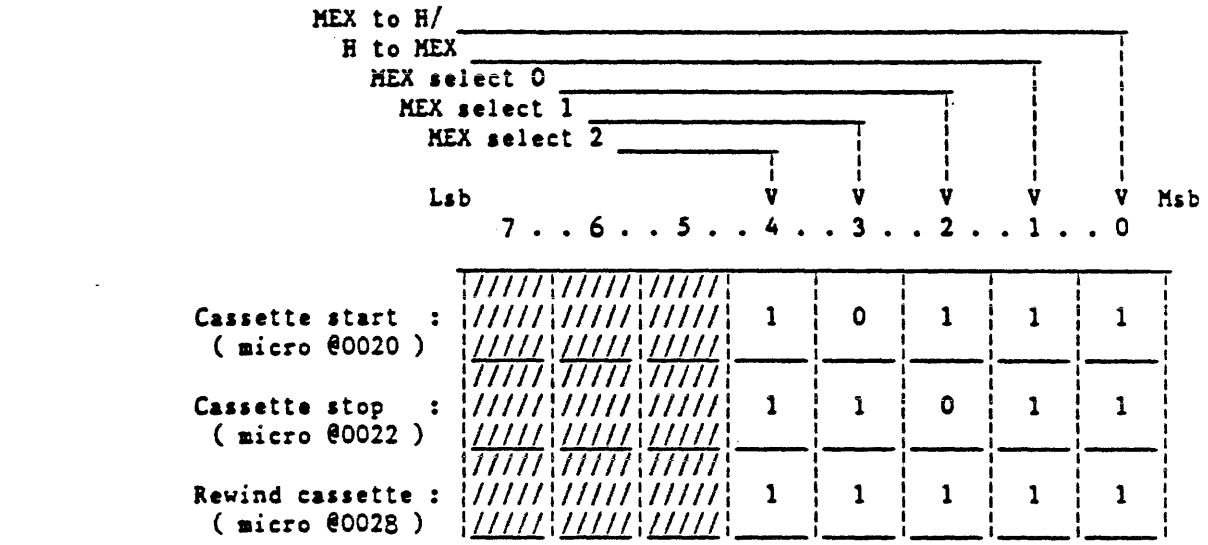
- Send a "SYSTEM-CLEAR" signal in PPI1CNTL (port @93) which will force the CPU to send registers "A" & "M" to the MAC card (see halt protocol).
- Send "MOVE NULL TO A" & "MOVE NULL TO BR" micros to the CPU (@1FE4 & @1FE6 respectively).
- Fill with zeroes the RAM area whose contents will be sent to the "M" register when executing the CPU start protocol.
- Send a "READ-CLEAR-ELOG" micro (@0B0B) micro the the CPU, read the results from the "Y" register (send "MOVE Y TO CNS" micro - @11AE) and store them in RAM.
- If not in cache-only source mode (CPUMD=2), send a clear cache micro (@0005) micro to the CPU.

Up to here, the "CLEAR" protocol is very similar to the one of the previous MAC, but in addition, the MAC-II will have to reload the FAT & directory from the system diskette, in order to allow the firmware to take into account a new diskette setting.

If the slave CPU is present and online, it will perform the same operations (except for the diskette handling), while if it is present but off-line, a RESET operation will just be executed.

SEMANTICS OF THE CASSETTE MANIPULATE MICROS :

These micros generate a request to the MAC-II processor on PPI2b (I/O address @95) with the following format :



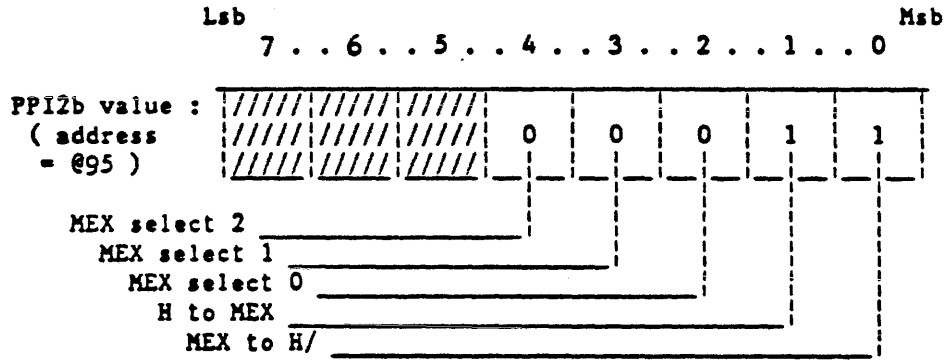
CASSETTE START : will enable data to be transferred to the "U" register from the diskette read buffers, if valid data is available in those buffers, if not, a read operation will take place, a sector pointer in the ".MTR" file will be incremented, and a byte pointer in the read buffers will be reset.

CASSETTE STOP : will invalidate the data in the read buffers of the diskette so that a subsequent CASSETTE START micro will cause next data to be read from the diskette, the sector pointer to be incremented, and the byte pointer to be reset.

CASSETTE REWIND : in addition of the known features of the cassette stop micro (invalidation of the buffers, buffer pointers reset), this instruction will cause the restore mechanism of the diskette to be initiated (the effect is then similar to the "UNLOAD" command of the ODT).

SEMANTICS OF THE "MOVE U TO register" MICRO :

Generates a request to the MAC-II processor on PPI2b according to the following format :



This request will cause 16 bits of data from the read buffers of the diskette to be moved to the "U" register of the CPU, provided those buffers were enabled by either a cassette start micro (run mode) or by a "GO" command from the ODT when the system is in MTR mode (see semantics of the "GO" command).

The transfer to "U" consists of 3 "OUT" micros to the HPPI port of the 2's complement of the data to be sent over, then strobe ENDCNH in PPI1a to release the host CPU.

After this transfer to the "U" register, the byte pointer in the disk read buffers will be incremented by 2, and when it reaches the end address of the buffers, a read operation of the next logical sector within the file will be initiated, the logical sector pointer will be incremented, and the byte pointer will be reset to the start address of the disk read buffers.

Crossings over TP400 cluster boundaries will be accomplished using the copy of the File Allocation Table in the RAM of the MAC-II card created by the execution of a previous "MTR" command from the ODT.

Disk read errors will cause the firmware to strobe the adequate error flag in PPI1a, in addition, the same procedure will be followed if the CPU tries to get data beyond the EOF cluster.

CNS REGISTER COMMANDS FORMAT :

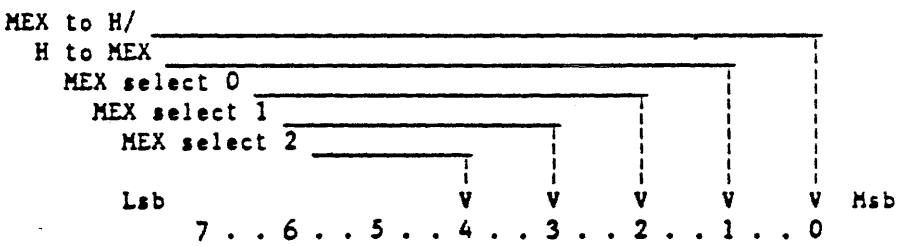
All the commands that can be directed to the CNS register may be described by the following table :

	Msb 2222 3210	1111 9876	1111 5432	1100 1098	0000 7654	0000 3210 Lsb
CNS value :	xxxx	xxxx	xxxx	xxxx	CDMR	ssss

CDMR = 1xxx : Control commands,
CDMR = 01xx : Deferred commands (master only),
CDMR = 001x : MAC-II typical commands,
CDMR = 0001 : Read/write MAC-II memory commands,
CDMR = 0000 : Immediate action commands,
ssss : Command sub-type.
All xxxx's must be reset for control, deferred, and immediate action commands, and are considered as data for MAC-II commands (will be described in the following paragraph).

Deferred, MAC-II, and immediate action commands are valid only if they were enable by the receipt of a specific control command (CNS = @000081).

All those commands are executed by the MAC processor when a request is made by the host to PPI2b (I/O address @95) with the following format :



	7	6	5	4	3	2	1	0
Data to CNS :	/////	/////	/////	/////	/////	/////	0	0
Register to CNS :	/////	/////	/////	1	0	0	1	0
Read CNS :	/////	/////	/////	0	0	1	1	1

CNS COMMAND FORMAT (CONTINUED) :

Whenever the MAC receives a command which implies an output to CNS from the host, the contents of that data will be available on MPIa,b,c (I/O address @80 - @83) and will be stored in the RAM of the MAC card.

If such a command has a "D" flag on (deferred command), the MAC firmware will memorize that the corresponding action has to be taken when the host processor halts, then, in halt mode the firmware main loop will check that there is a command pending, analyze the CNS, and execute the specified request. Note that any control command has to clear the flag which eventually indicates that a deferred command must be executed in halt mode, and any execution of the "START" procedure also clears that flag.

A "READ CNS" request from the host will cause the MAC firmware to read each byte of the CNS from the RAM, complement it, and send it sequentially over to HPIa (I/O address @84), HPIb (address @85), and HPIc (address @86) before releasing the host processor (send the "ENDCNH" strobe in PPI1a).

NEW CNS REGISTER COMMANDS :

A set of new CNS commands has to be introduced in order to allow the host processor :

- to select the name of the ".MTR" file he wants to read through the "U" register between diskette start/stop.
- to cause MAC-II firmware to read/write any particular 512 bytes disk sector to/from its RAM memory.
- to access (on input & output) the MAC-II RAM memory.

These commands may be summarized as follows :

Command name :	Value :
Reload FAT & directory.	00 00 21
Specify filename bytes 1 & 2.	xx xx 22
" " " 3 & 4.	xx xx 23
" " " 5 & 6.	xx xx 24
" " " 7 & 8.	xx xx 25
Read absolute sector.	xx xx 26
Write absolute sector	xx xx 27
Read RAM.	xx xx 11
Write RAM address	xx xx 12
Write RAM data.	xx xx 13
Write ODT mode.	xx xx 14

Where all xx's represent 8 bits command parameters which will be described later on.

Those commands will only be valid if enabled by the execution of the CNS @000081 command from the host, and will be nak'ed if the read buffers of the diskette are enabled (namely between a CASSETTE START and a CASSETTE STOP micro).

RELOAD DISK CNS COMMAND : [00 00 21]

The following command issued by the host to the CNS register will cause the firmware to read the FAT (file allocation table) sector and the 9 sectors of directory from the diskette currently inserted in the drive.
The RAM address of the buffer which contains this information will be available from the CNS response, so that it can be read later on using the "READ RAM" CNS command.

Command format :

Msb Lsb

2222 : 1111	1111 : 1100	0000 : 0000
3210 : 9876	5432 : 1098	7654 : 3210

0000	0000	0000	0000	0010 CDMR	0001 ssss

Response format :

Msb Lsb

2222 : 1111	1111 : 1100	0000 : 0000
3210 : 9876	5432 : 1098	7654 : 3210

Lower byte of buffer address	Upper byte of buffer address	////////////////

Upper and lower bytes of the 16 bits directory buffer address are inverted in the CNS response - refer to 8080 microprocessor type of addressing.

SPECIFY FILE NAME BYTES 1 & 2 COMMAND : [xx xx 22]

The following command issued by the host to the CNS register will cause the firmware to update the first and the second bytes of the MTR filename to be searched later on with the first & second bytes of the command, the CNS response is meaningless.

Command format :

Msb 2222 : 1111 : 3210	Msb 1111 : 1100 : 9876	Msb 0000 : 5432 : 1098	Lsb 0000 : 7654 : 3210
1 st byte of MTR filename.	2 nd byte of MTR filename.	0010 CDMR	0010 ssss

SPECIFY FILE NAME BYTES 3 & 4 COMMAND : [xx xx 23]

The following command issued by the host to the CNS register will cause the firmware to update the third and the fourth bytes of the MTR filename to be searched later on with the first & second bytes of the command, the CNS response is meaningless.

Command format :

Msb 2222 : 1111 : 3210	Msb 1111 : 1100 : 9876	Msb 0000 : 5432 : 1098	Lsb 0000 : 7654 : 3210
3 rd byte of MTR filename.	4 th byte of MTR filename.	0010 CDMR	0011 ssss

SPECIFY FILE NAME BYTES 5 & 6 COMMAND : [xx xx 24]

The following command issued by the host to the CNS register will cause the firmware to update the fifth and the sixth bytes of the MTR filename to be searched later on with the first & second bytes of the command, the CNS response is meaningless.

Command format :

```
Msb                                                    Lsb
  2222 : 1111  :  1111 : 1100  :  0000 : 0000
  3210 : 9876  :  5432 : 1098  :  7654 : 3210
```

5 th byte of MTR filename.	6 th byte of MTR filename.	0010 CDMR	0100 ssss

SPECIFY FILE NAME BYTES 7-8 & SEARCH COMMAND : [xx xx 25]

The following command issued by the host to the CNS register will cause the firmware to update the seventh and eighth bytes of the MTR filename and to search for that file name pattern within the directory buffer in RAM, which should have been read from disk by a previous command ; the address of the directory buffer will be returned in the CNS response.

Command format :

```
Msb                                                    Lsb
  2222 : 1111  :  1111 : 1100  :  0000 : 0000
  3210 : 9876  :  5432 : 1098  :  7654 : 3210
```

7 th byte of MTR filename.	8 th byte of MTR filename.	0010 CDMR	0101 ssss

Response format :

```
Msb                                                    Lsb
    2222 : 1111  :  1111 : 1100  :  0000 : 0000
    3210 : 9876  :  5432 : 1098  :  7654 : 3210
```

Lower byte of buffer address	Upper byte of buffer address	Search status

@(1) 0000 0110 @ (ACK) : successful search <--
@(1) 0001 0101 @ (NAK) : file not found <------

READ DISKETTE CNS COMMAND : [xx xx 26]

The following command issued by the host to the CNS register will cause the firmware to initiate a read disk operation on the disk sector number specified by the command; if successful, the data read will be available from the RAM locations specified in the CNS response (see "READ RAM" CNS command).

Command format :

```
Msb                                                    Lsb
  2222 : 1111  :  1111 : 1100  :  0000 : 0000
  3210 : 9876  :  5432 : 1098  :  7654 : 3210
 ┌──────────────┬──────────────┬──────┬──────┐
 │ Lower byte   │ Upper byte   │ 0010 │ 0110 │
 │ of disk addr.│ of disk addr.│ CDMR │ ssss │
 └──────────────┴──────────────┴──────┴──────┘
```

Upper and lower bytes of the 16 bits disk address need to be inverted in the CNS command - refer to 8080 microprocessor type of addressing.

Response format :

```
Msb                                                    Lsb
    2222 : 1111  :  1111 : 1100  :  0000 : 0000
    3210 : 9876  :  5432 : 1098  :  7654 : 3210
 ┌──────────────┬──────────────┬──────────────┐
 │ Lower byte of│ Upper byte of│ Read status  │
 │ buffer address│ buffer address│             │
 └──────────────┴──────────────┴──────┬───────┘
                                      │
@(1) 0000 0110 @ (ACK) : successful read <----┤
@(1) 0001 0101 @ (NAK) : read failure <-------+
```

WRITE DISKETTE CNS COMMAND : [xx xx 27]

The following command issued by the host to the CNS register will cause the firmware to initiate a write disk operation on the disk sector number specified by the command.
The 512 bytes buffer used to perform this operation will have the same address as the one used to perform the read operation (see command above) and may eventually have been filled from the host using the "WRITE RAM ADDRESS" and "WRITE RAM DATA" CNS commands (see following paragraphs).

Command format :

```
Msb                                                    Lsb
  2222 : 1111  :  1111 : 1100  :  0000 : 0000
  3210 : 9876  :  5432 : 1098  :  7654 : 3210
```

Lower byte of disk addr.	Upper byte of disk addr.	0010 CDMR	0111 ssss

Upper and lower bytes of the 16 bits disk address need to be inverted in the CNS command - refer to 8080 microprocessor type of addressing.

Response format :

```
Msb                                                    Lsb
    2222 : 1111  :  1111 : 1100  :  0000 : 0000
    3210 : 9876  :  5432 : 1098  :  7654 : 3210
```

Lower byte of buffer address	Upper byte of buffer address	Write status

@(1) 0000 0110 @ (ACK) : successful write <---
@(1) 0001 0101 @ (NAK) : write failure <-------

READ MACII RAM CNS COMMAND : [xx xx 11]

The following command issued by the host to the CNS register will cause the firmware to read 24 bits of RAM at the address specified by the command, which will be presented to the MEX lines, and available to the host by a "MOVE CNS TO register".

Command format :

```
Msb                                                    Lsb
  2222 : 1111  :  1111 : 1100  :  0000 : 0000
  3210 : 9876  :  5432 : 1098  :  7654 : 3210
```

Lower byte of RAM addr	Upper byte of RAM addr	0001 CDMR	0001 ssss

Upper and lower bytes of the 16 bits RAM address need to be inverted in the CNS command - refer to 8080 microprocessor type of addressing.

Response format :

```
Msb                                                    Lsb
  2222 : 1111  :  1111 : 1100  :  0000 : 0000
  3210 : 9876  :  5432 : 1098  :  7654 : 3210
```

RAM contents at [address]	RAM contents at [address]+1	RAM contents at [address]+2

This command might be useful for programs who need to access data which are resident in the MAC memory, such as diskette directory, disk data, pack identifier, or current MTR filename.

WRITE MACII RAM ADDRESS COMMAND : [xx xx 12]

The following command issued by the host to the CNS register will cause the firmware to memorize the first 2 bytes of the MEX lines as the address where any subsequent "WRITE RAM DATA" command will update then RAM memory.

Command format :

Msb 2222 : 1111 / 3210 : 9876	1111 : 1100 / 5432 : 1098	0000 / 7654	0000 Lsb / 3210
Lower byte of RAM addr	Upper byte of RAM addr	0001 CDMR	0010 ssss

Upper and lower bytes of the 16 bits RAM address need to be inverted in the CNS command - refer to 8080 microprocessor type of addressing.

The response format is meaningless.

WRITE MACII RAM DATA CNS COMMAND : [xx xx 13]

The following command issued by the host to the CNS register will cause the firmware to write 16 bits of data at the RAM address specified by a previous "WRITE RAM ADDRESS" command.

Command format :

Msb 2222 : 1111 3210 : 9876	1111 : 1100 5432 : 1098	0000 : 0000 7654 : 3210 Lsb
Lower byte of data	Upper byte of data	0001 CDMR / 0011 ssss

Upper and lower bytes of the 16 bits RAM address need to be inverted in the CNS command - refer to 8080 microprocessor type of addressing.

After completion of this update, the RAM address where the following such command should take place will automaticly be incremented by 2 (bytes), so that only one "WRITE ADDRESS" command is required to fill a buffer of contiguous memory locations.

Response format :

Msb 2222 : 1111 3210 : 9876	1111 : 1100 5432 : 1098	0000 : 0000 7654 : 3210 Lsb
Lower byte of next address.	Upper byte of next address.	////////////////

WRITE ODT MODE CNS COMMAND : [xx xx 14]

The following command issued by the host to the CNS register will cause the firmware store the medium byte of the command in the memory location used by the screen formatter to specify the kind of menu that will be displayed on the screen in halt mode.

Command format :

Msb 2222 : 1111 3210 : 9876	1111 : 1100 5432 : 1098	0000 : 0000 7654 : 3210	Lsb
//////////////	Code for ODT screen mode	0001 CDMR	0100 ssss

OPR page	0000 : 0000
FE page.	0000 : 0001
MAC page	0000 : 0010
REG page	0000 : 0011
STACK page	0000 : 0100
CKEY page.	0000 : 0101
S-MEM 16 page. . .	0000 : 0110
S-MEM 24 page. . .	0000 : 0111
S-MEM 39 page. . .	0000 : 1000
DIR page	0000 : 1001
TEXT page.	0000 : 1010

The response code is meaningless.

CHANGES TO EXISTING CNS COMMANDS :

The response to the "GET STATUS VECTOR" command should be modified in order to reflect the presence of a 5 1/4 disk as a bootstrap device instead of a cassette, so that any program wishing to use the "U" register will be able to know whether or not he has to use the file name selection commands (namely "SYSTEM/LOAD.CAS").

The response to a "GET STATUS VECTOR" CNS command (value = @000002) will be formatted as follows :

```
Msb                                                        Lsb
    2222 : 1111  :  1111 : 1100  :  0000 : 0000
    3210 : 9876  :  5432 : 1098  :  7654 : 3210
```

0000	00SR	Firmware level number	Contents of PPI1b

R = 1 : remote switch on,
S = 1 : synchronous link on (if R=1),

H card firmware level numbers are :
- @(1) 0001 0000 @ for MAC-I,
- @(1) 0010 0000 @ for MAC-II,

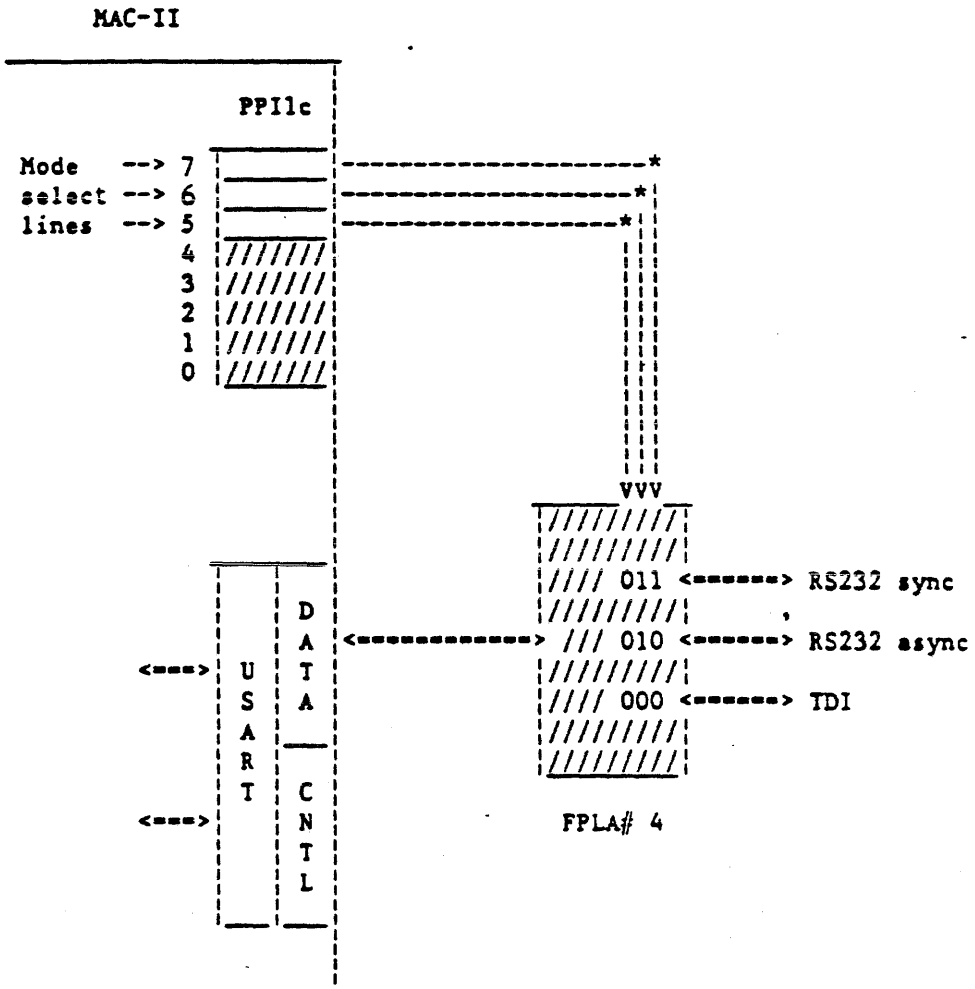
See appendix for a complete description of the bit mapping of PPI1b, found in the least significant byte of the CNS response.

In addition, the users of the "MTR restart" command, i.e. move @000041 to CNS, must be aware that, before they issue such a command, the file identifier field in MAC RAM must have been filled with a valid pattern, otherwise a NAK response will be provided by the firmware.

In case of parity detection in the RAM of the MAC card, a Non Maskable Interrupt will force the firmware to enter a loop which will give a NAK response to all the requests issued by the host to the CNS register.

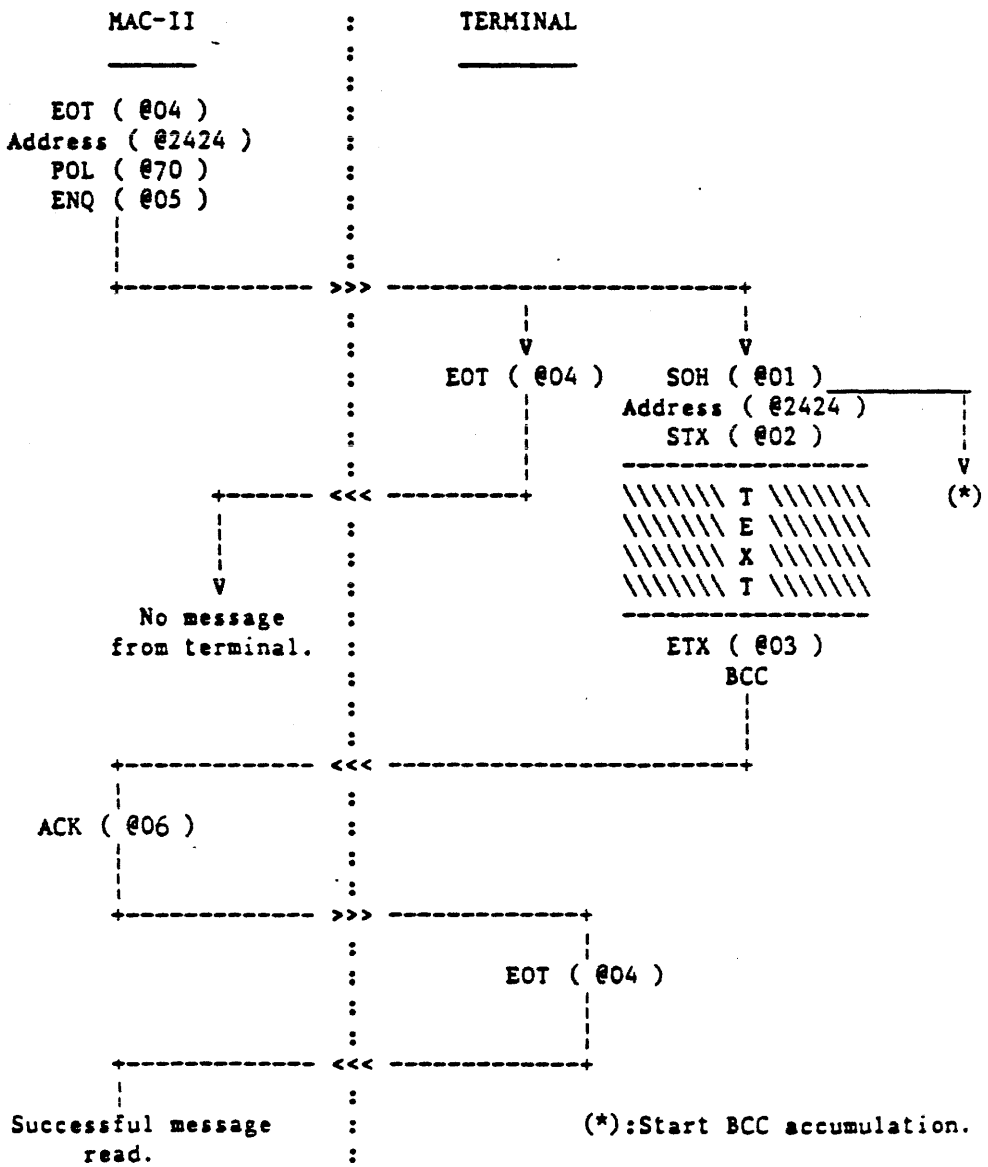
MAC-II ODT COMMUNICATION

The messages of the MAC card can be directed through the USART to either a local ODT (using a TDI line) or a remote ODT (using an RS-232 sync or a-sync line) by an appropriate setting of the mode select lines of PPI1c - see drawing below :

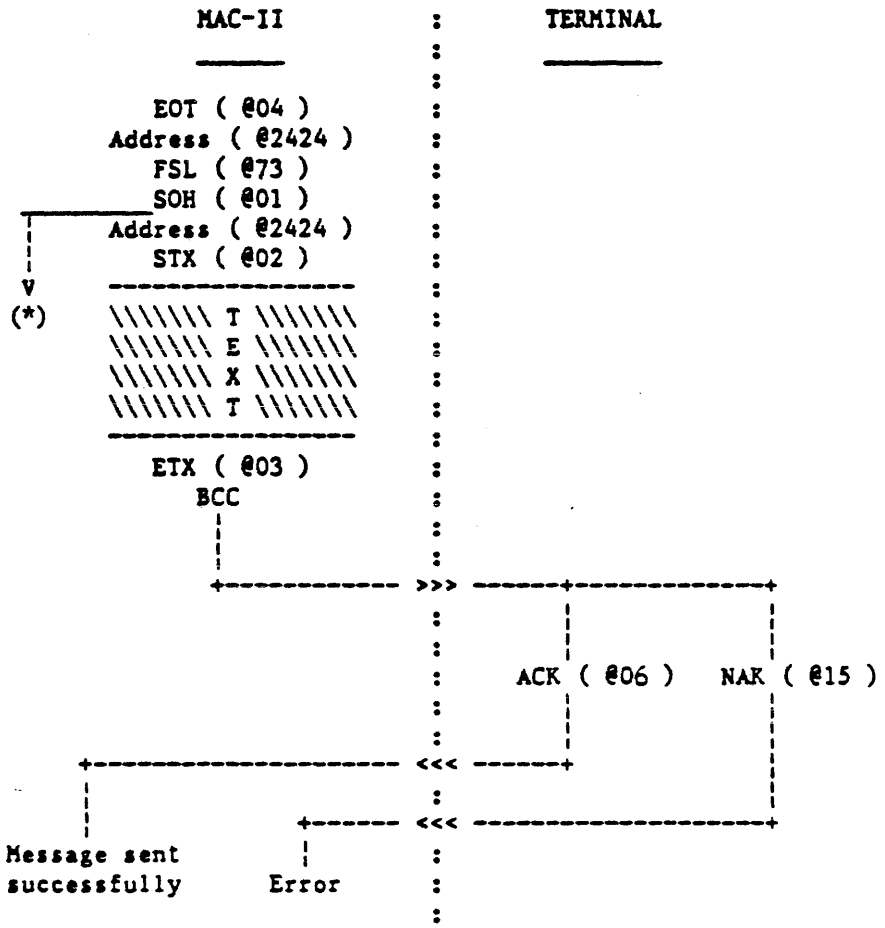


In halt mode, the MAC-II firmware communicates to a local ODT using a Two-wire Direct Interface line at 9600 baud, with a terminal whose physical address is "$$" (@2424).
The protocols used are "FAST SELECT" (write operations) and "POLL" (read operations), without transmission number.
The following pages summarize the data interchange made to perform this :

"POLL" procedure :



"FAST SELECT" PROCEDURE :



(*) : Start BCC accumulation.

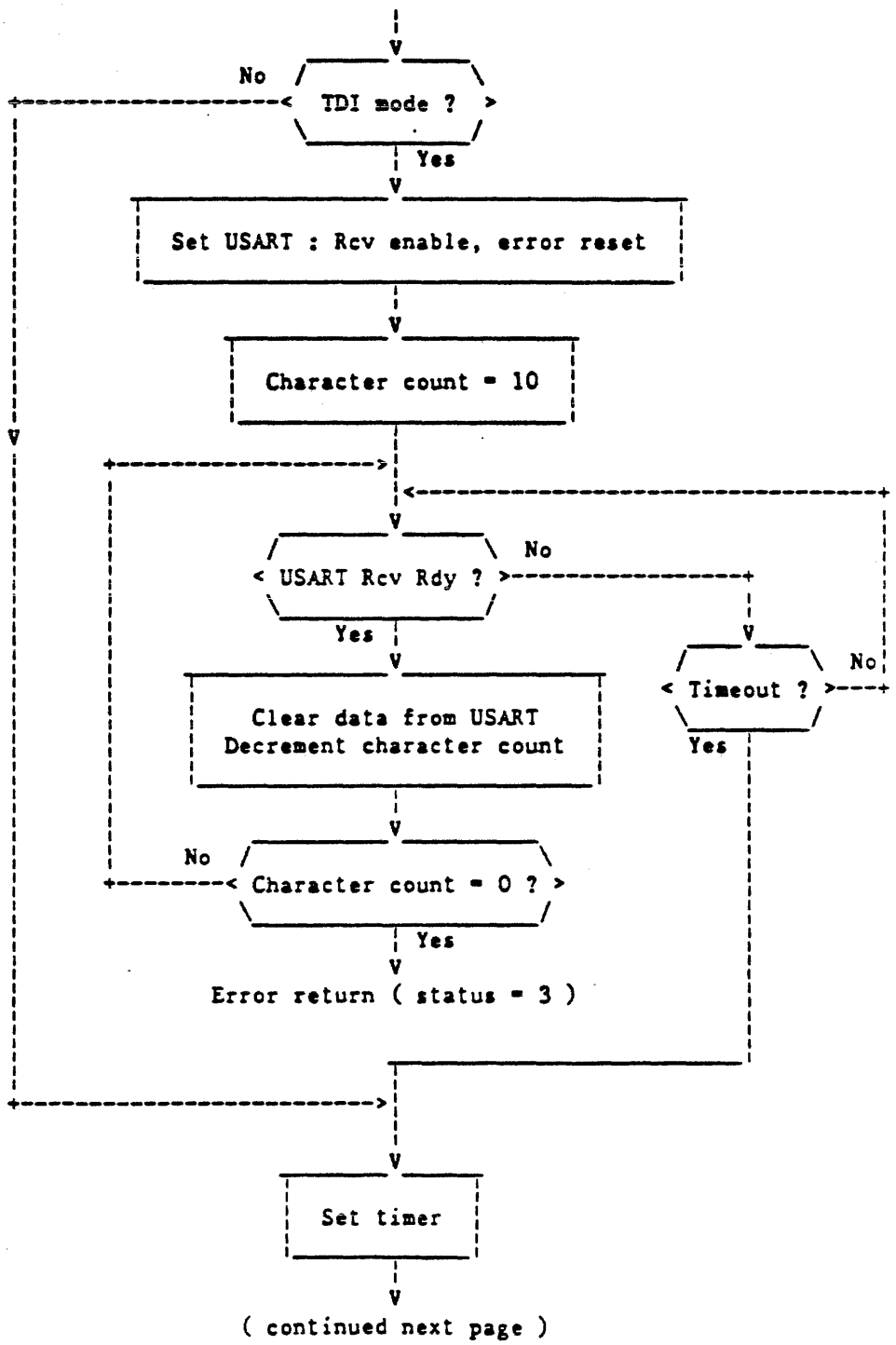
The Block Check Code (BCC) will in this case be a simple LRC (Longitudinal Redundancy Check - exclusive OR of all characters following SOH, up to and including ETX).

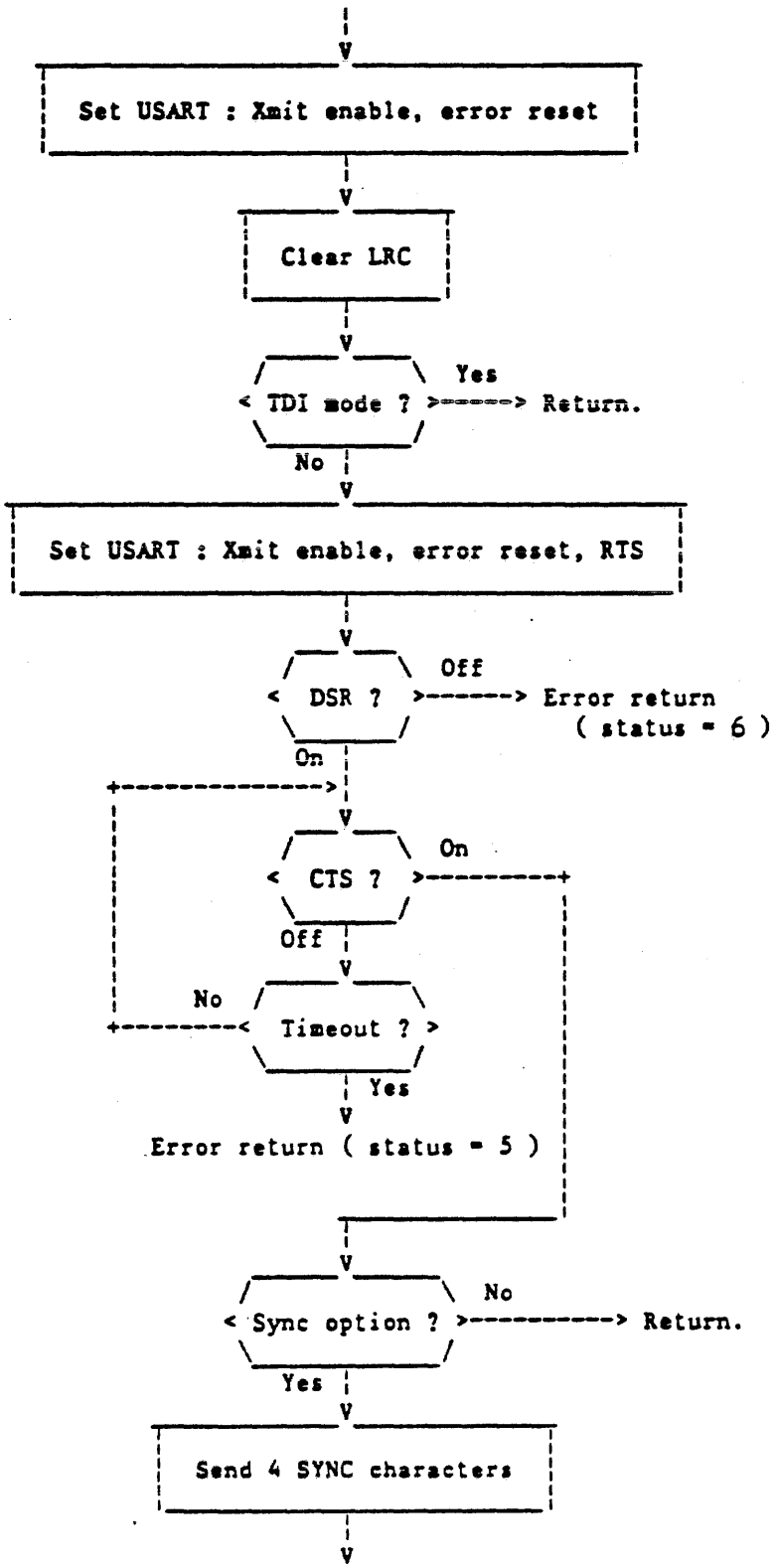
The USART mode word for the transmission from/to the local ODT will be equal to @7A, meaning 1 stop bit, even parity, 7 bits per character, Baud factor = 16 x (see appendix for further details).

The same procedures also apply to a remote ODT, through a RS-232 line accessible via the USART, provided an appropriate command had turned the remote option on. The remote link, as selected by the "MODE" bits in PPI1c, may be either SYNC or ASYNC (the latter with 3 possible values of the Baud rate 300, 1200 or 1800 Bps).

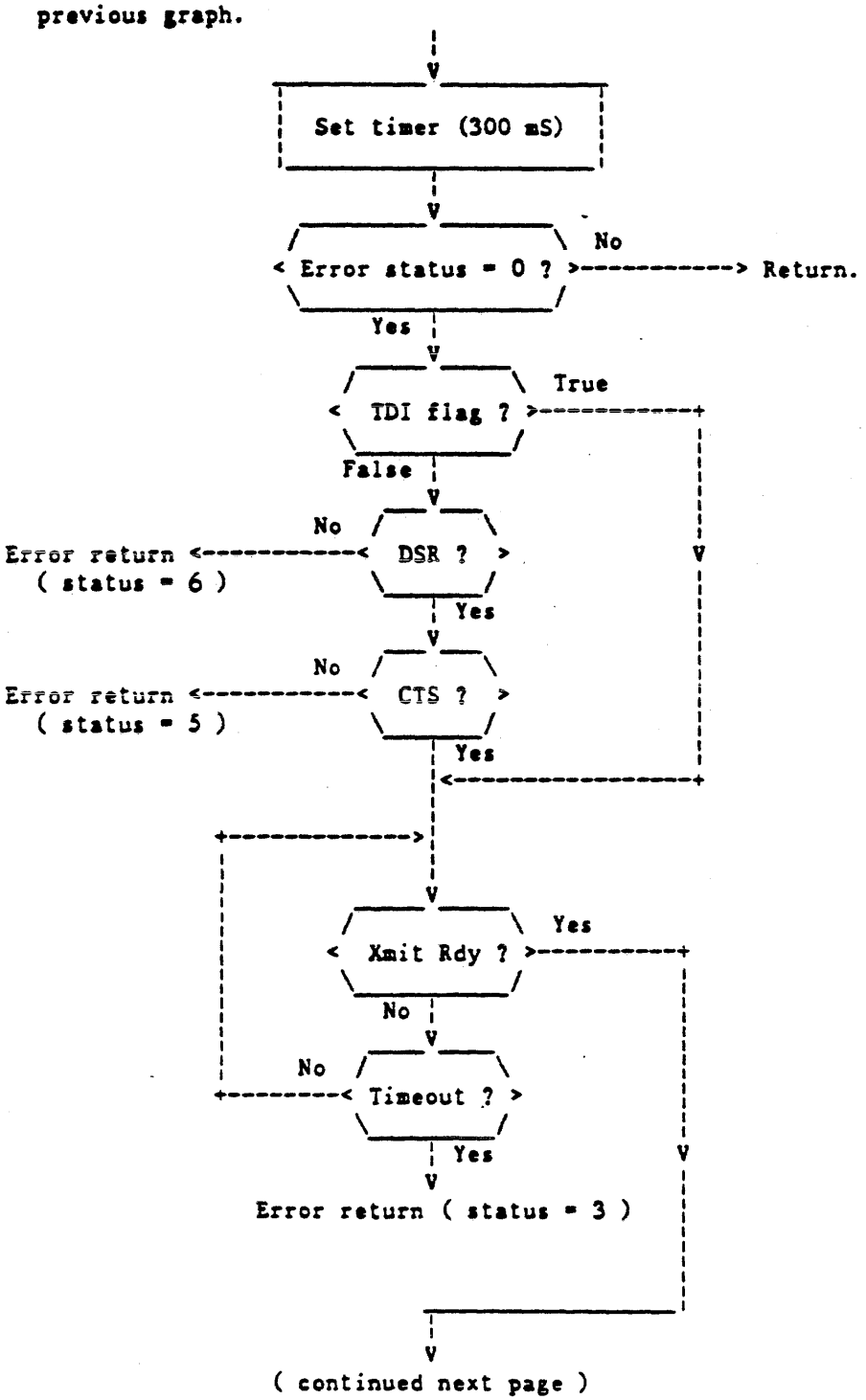
USART TRANSMIT SEQUENCE :

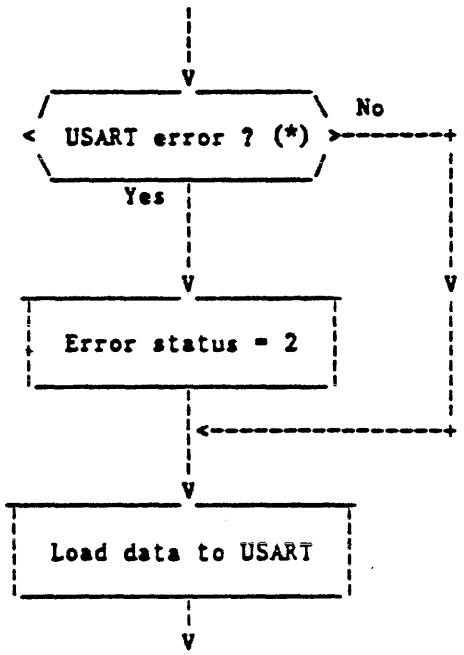
The following flow describes the procedure followed by the MAC-II firmware to initiate a transmit sequence through the USART :




Set USART : Xmit enable, error reset
Clear LRC
TDI mode ?
Yes
Return.
No
Set USART : Xmit enable, error reset, RTS
DSR ?
Off
Error return
(status = 6)
On
CTS ?
On
Off
Timeout ?
No
Yes
Error return (status = 5)
Sync option ?
No
Return.
Yes
Send 4 SYNC characters

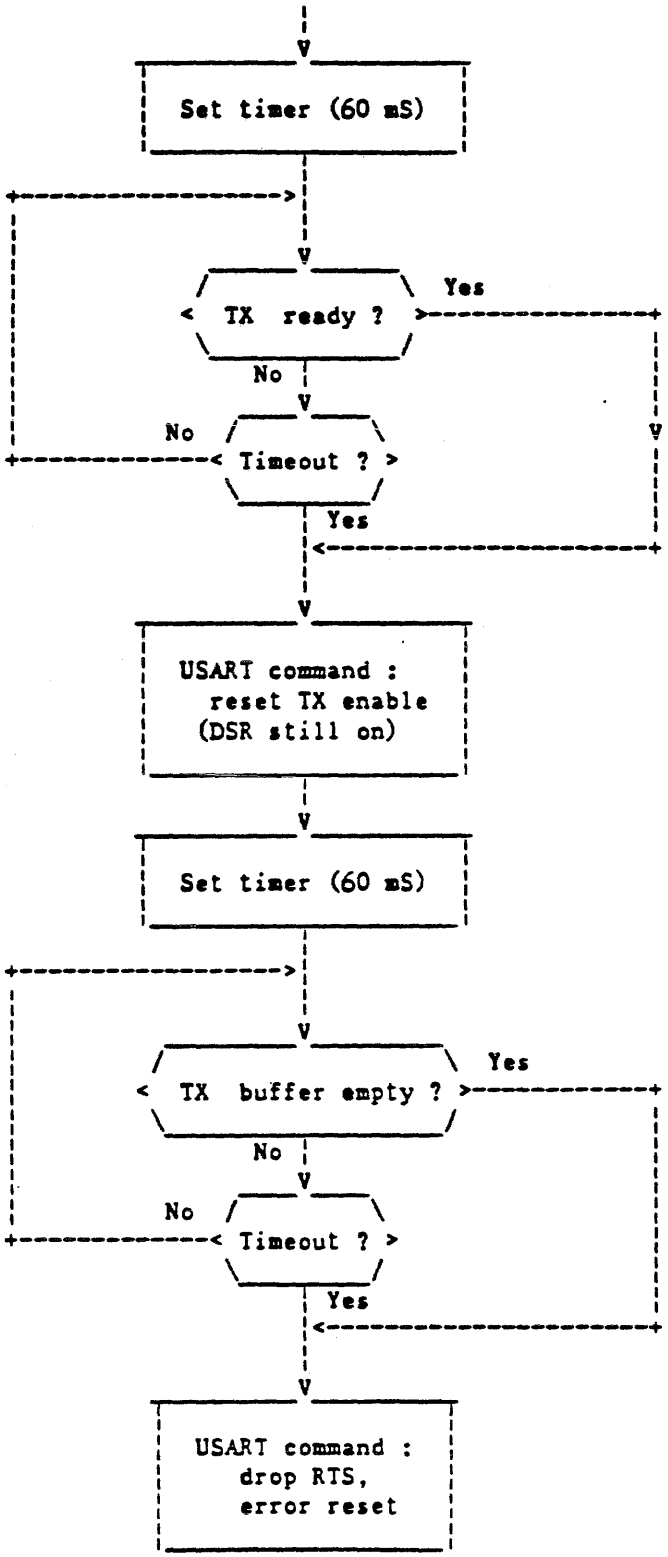
The following sequence is used to send a character through the USART, once the transmit procedure has been initiated by the previous graph.





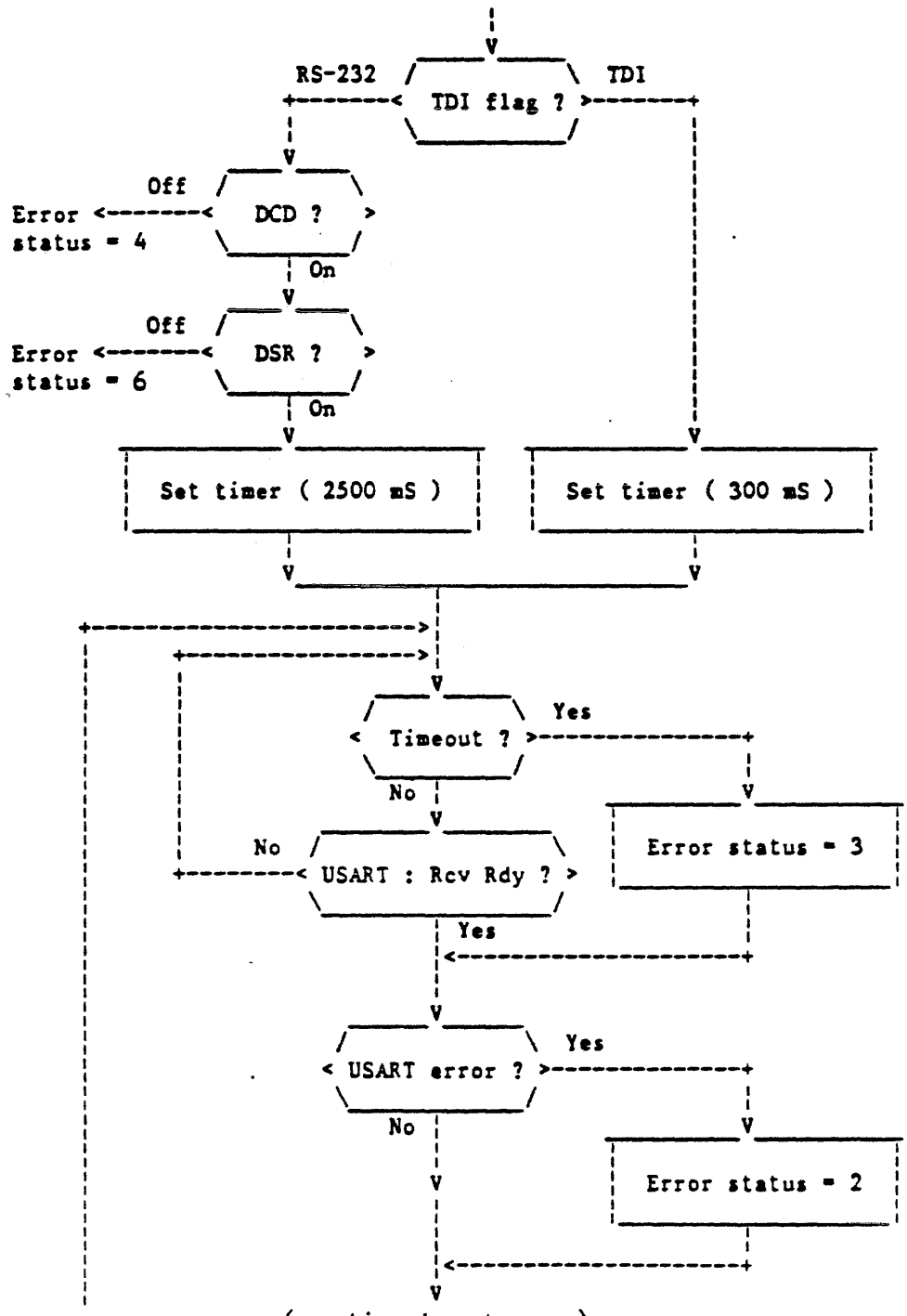
(*) USART errors - frame error,
- over-run,
- parity error.

All data transmit sequences through the USART end up according to the procedure described by the graph on following page.
Note that transmit enable (bit# 0 of USART command word) is reset as soon as the datacomm inteface chip is ready to accept one more character after the end of the message, while the Request To Send signal (RTS-bit# 5 of USART command word) is lowered whenever all characters have left the transmit buffer.

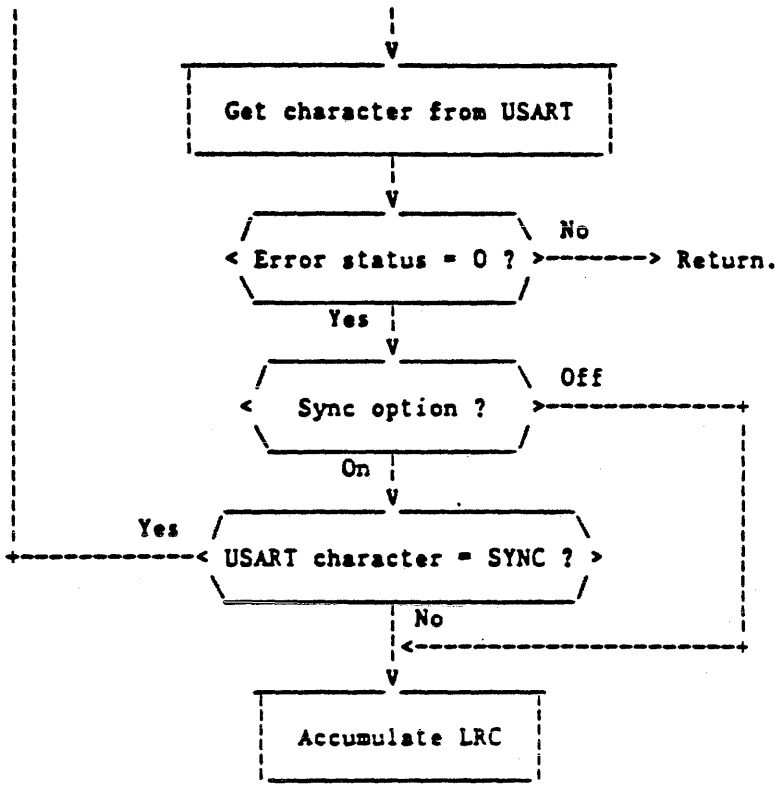

Set timer (60 mS)
TX ready ?
Yes
No
Timeout ?
No
Yes
USART command :
reset TX enable
(DSR still on)
Set timer (60 mS)
TX buffer empty ?
Yes
No
Timeout ?
No
Yes
USART command :
drop RTS,
error reset

USART RECEIVE SEQUENCE :

The following flow applies when the MAC-II firmware has to fetch a character from the USART (control port @B1, data port @B0)



(continued next page)



NOTES : USART errors include parity error, overrun & frame error

- A special handling has to be done in SYNC mode in order not to disregard SYN (@16) characters when receiving the LRC of a poll sequence.

- Prior to enter this flow, the Rcv enable, and eventually SYN detect flags must have been set in the USART command as well as mode select in PPI2c & control byte in PTMRC.

- The async ODT transactions will use PTMR1 in mode 3 (square wave rate generator) according to the terminal speed i.e. :

```
TDI ..................... @0D00 ( 9600 Baud )
Rs 232 async ............ @6800 ( 1200 Baud )
  "     "    ............ @A001 (  300 Baud )
  "     "    ............ @4500 ( 1800 Baud )
```

  while PTMR2 will be used in mode 0 (timeouts will be reported in PPI2a, bit# 7).

According to the flows above, a one byte result status will be returned by the ODT handler on each local (SPOST) or remote (RMPST) terminal transaction whose signification can be interpreted as follows :

SPOST/RMPST value	POLL sequence	SELECT sequence	
0	Valid message	Message ACK'ed	
1	Message NAK'ed	Transmission error	(*)
2	Parity / frame / overrun error		
3	<- Timeout error ->		
4	<- DCD error ->		
5	<- CTS error ->		
6	<- DSR error ->		
255	EOT received	////////////////////	

(*) : Ten retries will be performed before reporting select transmission errors.

B1990 UPDATE SEMINAR

MASTER -> SLAVE MAC-II COMMUNICATION :

Performed by a 1 byte communication over a TDI line, with the following signification :

Request :	Value :
Stop / reset request	@30
Clear request	@31
Clear cache request	@32
Slave on-line request	@33
Slave off-line request	@34
Single step on request	@35
Single step off request . . .	@36
Interrupt on request	@37
Interrupt off request	@38
Step request	@39
Start request	@3A
Remote on request	@3B
Remote off request	@3C
Baud rate select request . . .	@3D - (1200)
" " " " . . .	@3E - (300)
" " " " . . .	@3F - (1800)
Synchronous remote request . .	@40
Start self test request . . .	@41

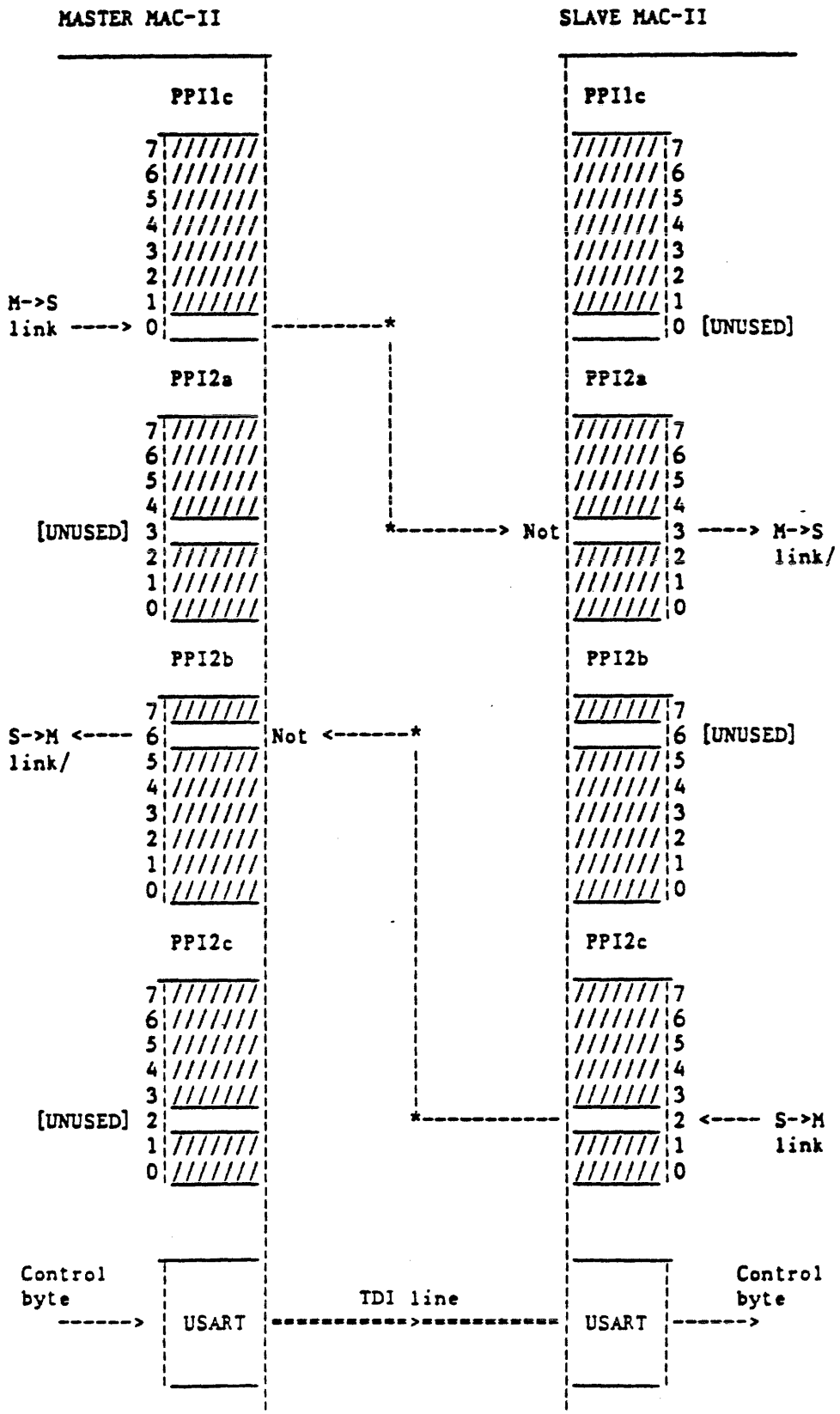
Prior to establish this communication, the master will check if the SLAVE-TO-MASTER link (PPI2b) was already on, if not, the USART will be initialized for TDI transmit (see flow above), the MASTER-TO-SLAVE link (PPI1c) will be raised, and the master MAC will wait for a 500 mS timeout period for the slave (which is supposed to poll constantly the master link in PPI2a) to set the SLAVE-TO-MASTER link (PPI2c). The specified character may then be transmitted through the USART using the above flows.

If any error occurs during this process, the master MAC will drop the MASTER-TO-SLAVE link in PPI1c.

If either the command character could not be received correctly, or the command character was decoded as invalid, or any error occurred during the execution of the command character, the slave has to send a nak response to the master.

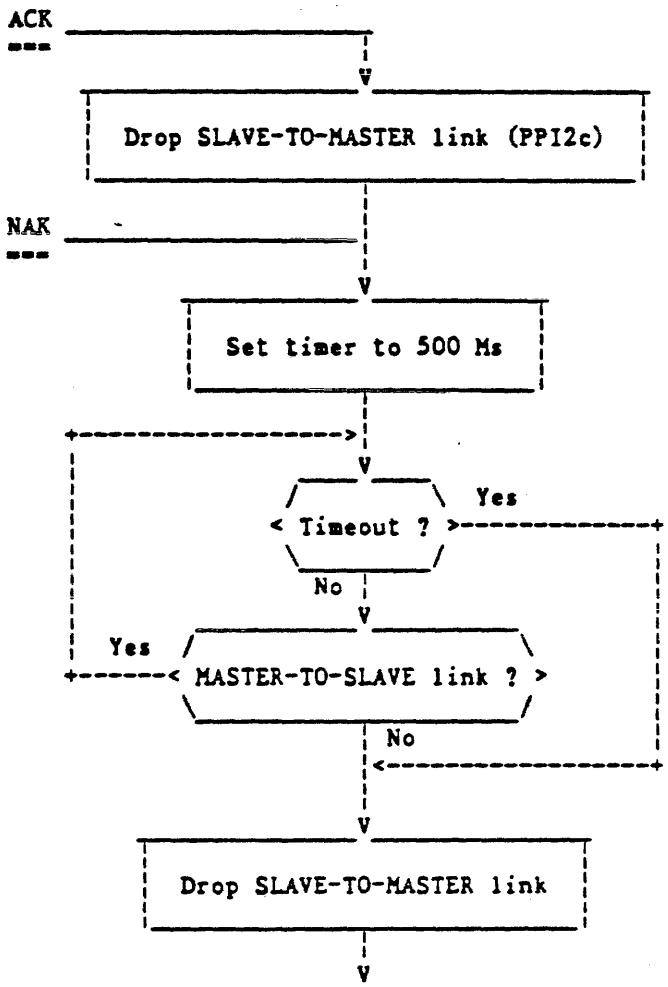
The drawing on the following page describes the communication lines between master and slave Maintenance Access Cards :

B1990 UPDATE SEMINAR



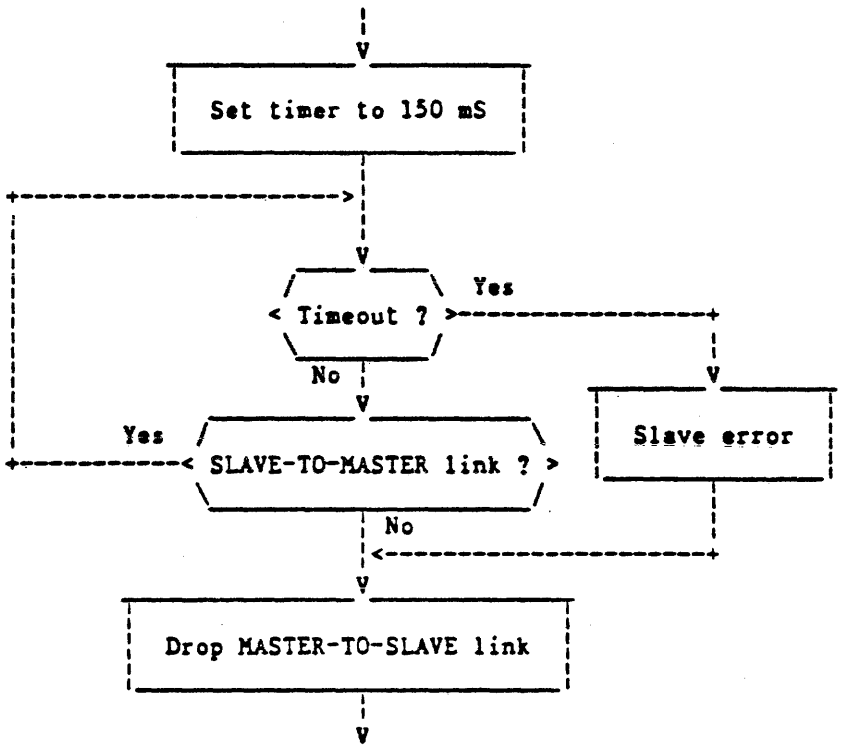
SLAVE RESPONSE TO A MASTER COMMAND :

The following flows apply to the handshake protocol between master and slave to evaluate the execution of a master command :



The ACK protocol will lower the SLAVE-TO-MASTER link immediately, while a NAK response will do it only when either a timer period of 500 mS has elapsed, or when the master has dropped his link with the slave.

EVALUATION OF THE SLAVE RESPONSE :



If the SLAVE-TO-MASTER link was still on after a time period of 150 mS, a slave error is assumed, otherwise the master command may be considered as successful, in both cases the MASTER-TO-SLAVE link needs to be lowered.

ODT MESSAGES HANDLING :

Consists of 2 main procedures : "GETTOKEN" and "SCANNER".

"GETTOKEN" is used to isolate the next keyword from the ODT input line and needs as an output the character pointer and the length of the previous token (respectively CHARP & TOKLEN).
The outputs will be the CHARP and TOKLEN of the new token.
This routine will disregard leading blanks, and convert any lower case alpha character of the new token to upper case.

GETTOKEN recognises 3 types of keywords (indicated by TOKENTYPE at the output of the procedure) i.e. :

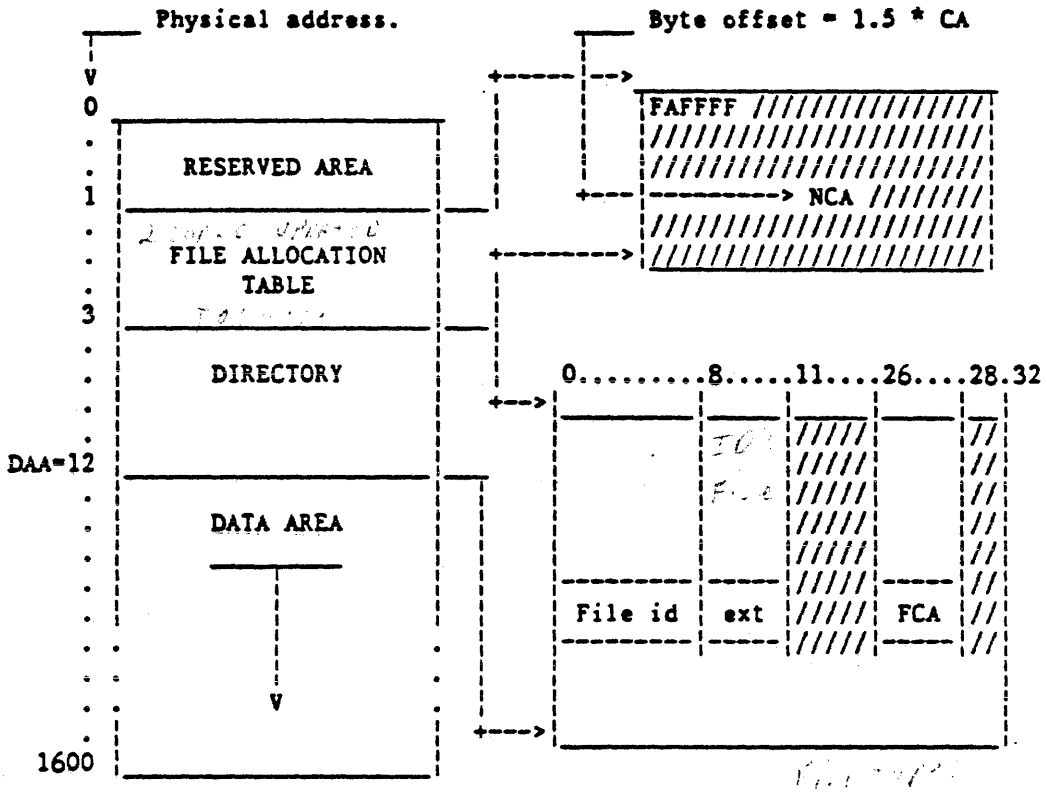
- End of line pseudo token (value 1) whenever the scan of the first character reaches the size of the ODT input line (indicated by OPMSLT).
- Alphanumeric token (value 2), character range : A->Z, 0->9
- Special token (value 3), TOKLEN always equal to 1.

"SCANNER" compares the symbol isolated by the GETTOKEN procedure with lists of allowed keywords and gives an associated symbol type and a symbol parameter as output.

The symbol type may be interpreted as follows :
- Value 0 : invalid symbol.
- Values 1 - 51 : primary command word.
- Values 101 - 105 : modifier symbol.

The "INTERACT" routine called by the halt mode main loop will perform a case statement on the symbol type in order to branch to the appropriate piece of code which will make use of the symbol parameter to execute the selected ODT function.

APPENDIX : TP400 DISK STRUCTURE.



UA = unit of allocation (8).
DAA = data area address (12).
FCA = first cluster address.
CA = cluster address.
NCA = next cluster address.

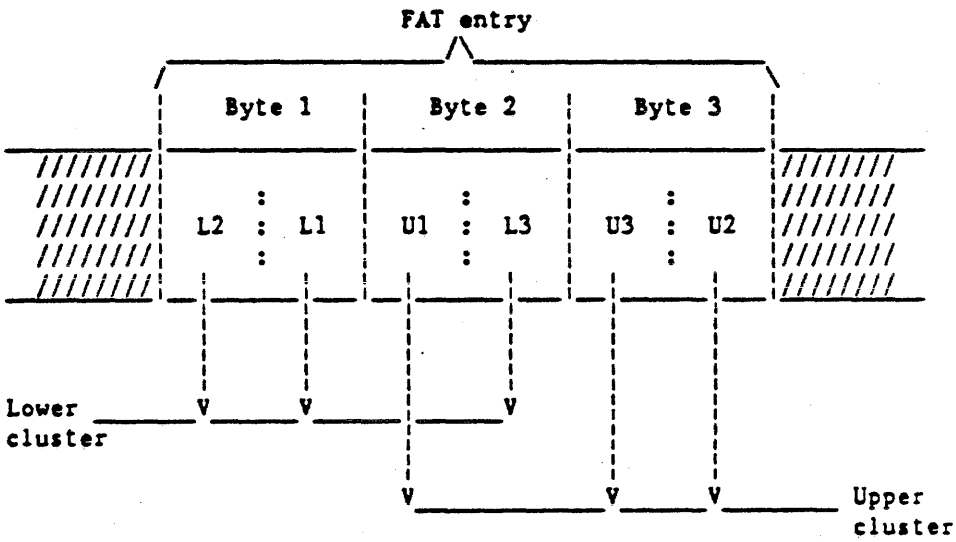
CA = @000 : Cluster available for file allocation.
CA = @FF7 : Hardware error within cluster.
CA = @FF8 - FFF : End of file cluster.
Else : Physical sector address := (CA - 2) * UA + DAA

(continued next page)

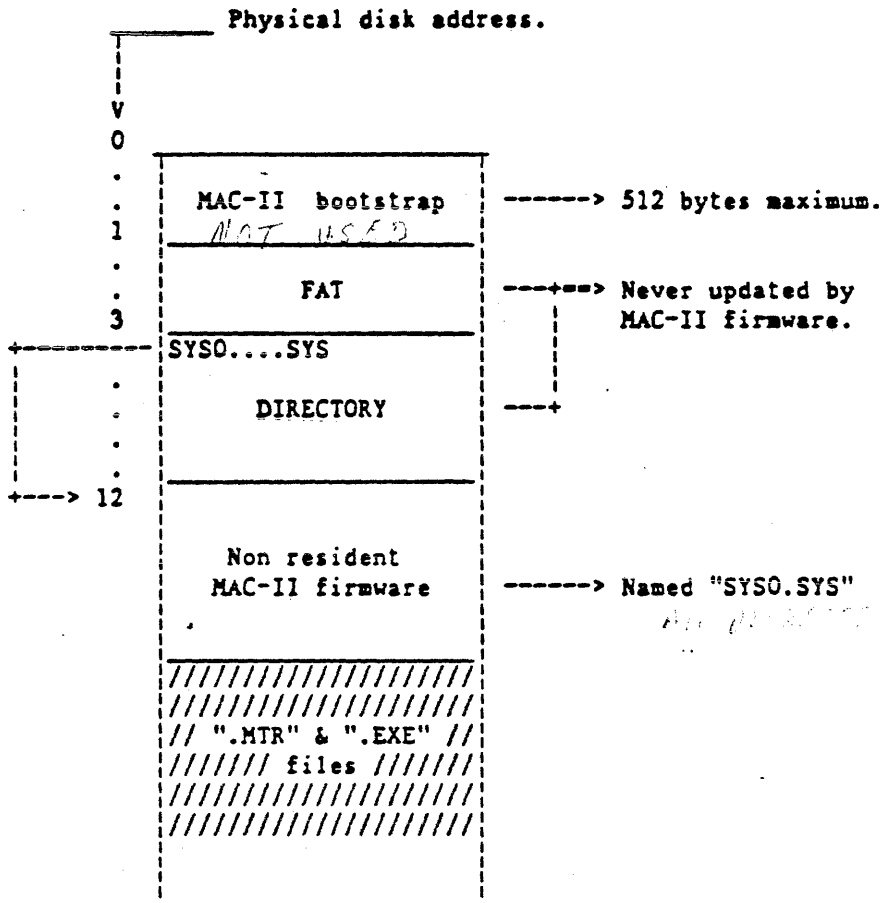
TP400 FAT FORMAT :

The addresses of the clusters pertaining to any particular file are linked in the disk FAT (file allocation table).
Each 24 bits FAT entry contains 2 x 12 bits cluster addresses.
Let L1-L2-L3 & U1-U2-U3 be 4 bit digits, and let the two 16 bit cluster addresses of a 24 bit FAT entry be @0L1L2L3 (lower) & @0U1U2U3 (upper).

Those addresses will be represented in the 24 bits FAT entry according to the following format :



APPENDIX : CONTENTS OF MAC-II DISK.



GENERATION OF MAC-II BOOTSTRAP DISKS :

The creation of MAC-II bootstrap code on the first sector of a TP400 disk can be done by means of the "BOOTSTRP" program on the ET2000.

The purpose of this program is to copy the first 512 bytes of a specified standard ET2000 file to the physical sector number zero of another specified disk.

The program will first ask the input parameters as follows :

1) ENTER INPUT UNIT :

The answer to this question is a 1 character input which specifies the unit where the input file is located; the input format will be A,B,C,D or 0,1,2,3.

2) ENTER FILE IDENTIFIER :

The answer is an ET2000 file name (without extension) ended either when the 8th character is entered, or when the return key (@0D) is depressed. The blank characters will be disregarded, and the rub-out key may be used to correct typing errors.

The program will then search for the specified file with a file identifier extension of ".BOT", if the file cannot be located, an appropriate message will be sent, and the whole process will be restarted, if the file is present, its first sector will be read and displayed on the ODT screen line by line (skip to next line by depressing any key).

The operator will then be asked to specify the output disk drive where the bootstrap code will be written, using the following message : "ENTER OUTPUT UNIT : ".
The format of the answer to this question will be similar to the one used to specify the input drive.

After a successful output to the requested drive, the program will be restarted from the beginning.

The whole process can be resumed/aborted by means of the CTRL-C mechanism.

APPENDIX : WD1770 DISK CONTROLLER FORMAT.

I/O port address	Input	Output
@C0	Status register	Command register
@C1	Track register	
@C2	Sector register	
@C3	Data register	

COMMAND REGISTER FORMAT :

+---- Type I (head motion) commands.

	Msb 7	6	5	4	3	2	1	Lsb 0
RESTORE :	0	0	0	0	M	V	Step rate	
SEEK :	0	0	0	1	M	V	Step rate	
STEP :	0	0	1	U	M	V	Step rate	
STEP IN :	0	1	0	U	M	V	Step rate	
STEP OUT :	0	1	1	U	M	V	Step rate	

U : update LSI's internal track register (U=0, no update)

M : disable "MOTOR ON" flag (M=0, enabled).

V : verify on destination track (V=0, no verify).

Step rate : 00 = 6 milli sec.
01 = 12 milli sec.
10 = 20 milli sec.
11 = 30 milli sec.

WD1770 FORMAT (continued) :

+---- Type II (input / output) commands.

	Msb 7	6	5	4	3	2	1	Lsb 0
READ :	1	0	0	S/M	M	D	0	0
WRITE :	1	0	1	S/M	M	D	P	W/D

S/M : single/multiple sector (0 = single).

M : same as type I commands.

D : add 30 milli sec. delay (0 = no delay).

P : disable write precomp flag (0 = enable).

W/D : write/delete flag (1 = delete)

STATUS REGISTER FORMAT :

Bit#	Meaning :	Type I	Type II
Msb 7	Motor on	X	X
6	Write protect.	-	X
5	Spin up.	X	-
	Record type.	-	X
4	Record not found	-	X
3	CRC error.	X	X
2	Lost data.	-	X
	Track 00 pin	X	-
1	Index pin.	X	-
	Data request	-	X
Lsb 0	Busy	X	X

B1990 UPDATE SEMINAR

SELF TEST MODIFICATIONS

April 85

SELF TEST FLOW CONTROL

On POWER ON or H Card Reset, the "microprocessor test" is automatically executed. The "microprocessor test" tests the Z80A microprocessor instructions used on H10 card, the 50 bytes of high-order RAM required for self test and stack operations and write good parity into all MACII RAM locations.
The microprocessor test, when executed in the system environment causes the control panel incandescent lamps to light. A probable failure in the microprocessor or related logic exists if the lamps fail to come on or never go off. Refer to Section 3 if either condition exists.

Also, the microprocessor can control the incandescent lamps despite failures which prevent certain instructions from executing. If such a failure exists, an attempt to freeze display error in test 1F will be made and then both the control panel lamps and the front plane LED on the H10 card will flash continuously. Go to Section 3 if this occurs.
Further self tests cannot be invoked until the microprocessor test is made to complete successfully.

The I/O ports are also initialized at the beginning of the self test.
To initialize an I/O port, the microprocessor writes a control word to the I/O address of the control of that I/O port. The control word will then define the function of the I/O port. Table 2 shows the function of every port and the required control word.

Following successful completion of the "Microprocessor Test" the tests which will execute depend on the test switch settings (see Table 2.1). Test ID (Test Switch Setting) 00, 13 and 1F are pseudo tests in the sense that they do not specifically refer to a particular test.
Test ID 00 refers to a comprehensive package of thirteen different "core" tests required for successful H card operation; namely, tests 01 to 0D. As part of Test ID 00 selection, an attempt to display the "H10 Self Test Summary" on the ODT is made if any of the thirteen tests were in error.

Test ID 13 is exactly the same as Test ID 00 except that the "H10 Self Test Summary" is always displayed on the ODT even if there are no failures.

Test ID 1F causes a "No Self Test" selection. This switch setting might cause H-card errors to go unnoticed and hence it is advisable not to use it unless confidence has already been established regarding the proper operation of the H card. This setting is primarily meant for Engineering Design purposes.

TEST ID	SELECTION
a) 00000 (00)	Comprehensive test (thirteen tests - 01 to 0D) followed by attempt to write Self Test Summary on ODT Screen ONLY on error.
b) 00001 (01)	ROM Test
c) 00010 (02)	RAM Test
d) 00011 (03)	Clock Test - This test forces execution of tests 05,07 & 08 prior to starting itself.
e) 00100 (04)	MPPI Test
f) 00101 (05)	USART Test
g) 00110 (06)	Disk Test
h) 00111 (07)	Logic A Test
i) 01000 (08)	Logic B Test
j) 01001 (09)	Logic C Test
k) 01010 (0A)	CPU Clear Test (CLRB)
i) 01011 (0B)	Mex Echo Test - This test causes execution of Test 0A if D card-H card control is not in 'FETCH FROM MAC' state. If Subtest 3 of Test 0A fails, Mex Echo Test is skipped.
m) 01100 (0C)	IO Echo Test - This test causes execution of Test 0A if D card-H card control is not in 'FETCH FROM MAC' state. If subtest 3 of Test 0A fails, IO Echo Test is skipped.
n) 01101 (0D)	Interrupt Test (uP)
o) 01110 (0E)	Spare : Assume Comprehensive Test 00
p) 01111 (0F)	Loop Test (Address Switch)
q) 10000 (10)	Spare : Assume Comprehensive Test 00
r) 10001 (11)	Loop Test (TDI)
s) 10010 (12)	Spare : Assume Comprehensive Test 00
t) 10011 (13)	Comprehensive Test (13 Tests - 01 to 0D) followed by attempt to write self test summary on ODT screen.
u) 11111 (1F)	Skip all self tests (uP test is always executed.)

NOTE: All other test ID settings cause default selection of Test ID 00.

THE CARD EXTENDED ENVIRONMENT

The H10 card contains the following items for test purposes:

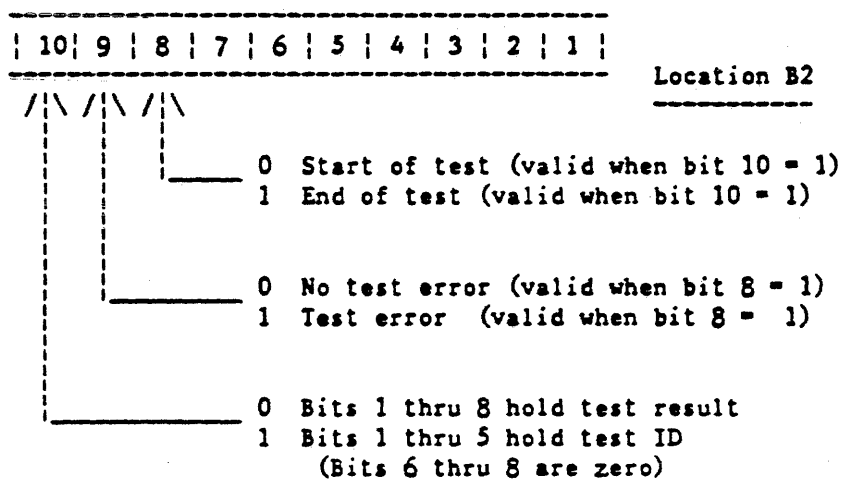
a. One DIP Switch (D2)

b. One 10 segment bar LED chip (B2)

c. One frontplane mounted momentary action pushbutton switch

d. One individual LED located in the middle frontplane

THE 10 SEGMENT BAR LED CHIP

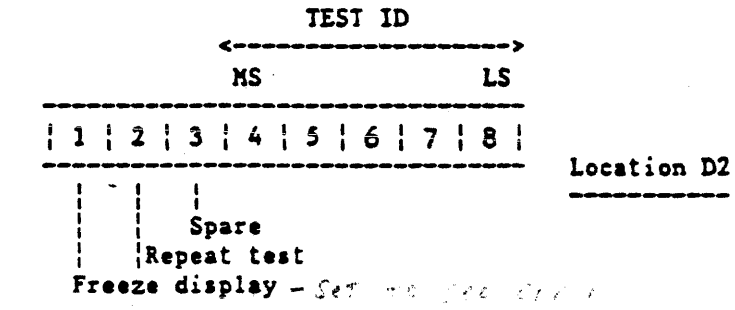


Bits 1 to 8 of the 10 segment bar LED chip represent via the display port the contents of the 8 bit microprocessor data bus.
These segments can thus be set and reset by the microprocessor program

Bits 9 and 10 of the bar LED represent the contents of the LED89 port loaded by program from bit 0 and 1 of microprocessor bus. These segments can thus be also set and reset by the microprocessor program.

The bar LED displays valid information depending of value DIP switch functions outlined after.

THE DIP SWITCH



DIP/SWITCH	SIGNIFICANCE

D2-1 Freeze Display

OFF Depression of the frontplane momentary action push button switch will cause the card to be reset after which the test selected by the TEST ID switches will execute.
If error occurs LED will display

1	1	1	0	0	T	T	T	T	T

With TTTTT = TEST ID of test in error

ON Same as above but the if the specified test is in error the LED display will showns :

1 -

1	1	1	0	0	T	T	T	T	T

T = TEST ID

2 -

0	1	R	R	R	R	R	R	R	R

R = RESULT WORD

3 -

0	0	R	R	R	R	R	R	R	R

R = RESULT WORD

4 - Keep display "3" except for RAM TEST, MEX ECHO TEST IO & ECHO TEST which loop on sequences 1-2-3.

If specified test is not in error, the bar LED will display at the end of test :

| 1 | 0 | 1 | 0 | 0 | T | T | T | T | T |

D2-2 Repeat test

ON -- Causes the test specified by TEST ID switches to repeated until repeat switch is turned off

D2-4 to D2-8 Test ID.

Test IDs range from 00000(00) to 11111(1F) and are used to select individual tests.

ON Represents 1
OFF Represents 0

MANUAL TEST SEQUENCING PROCEDURE

(Card Extended Environment)

All tests listed can be sequenced manually. The freeze feature is used in this procedure to allow information to be displayed on the 10-segment LED.

At least two displays will occur with each test. These display the test identification at the start and end of the test
An additional display will follow if failures were detected during the test. The error word(s) associated with the test will be displayed at that time.

PROCEDURE:

1. Extend Card.
2. Place Dip Switch D2-1 ON (Freeze Display)
3. Place Dip Swtich D2-2 on (Repeat)
4. Select the test using Dip Switches D2-4 (MSB) through D2-8 (LSB).
5. Depress the push button switch. The test will be repeated until repeat switch is OFF .

   At the end of test,
   - If the selected test is in error, display will shows first Test ID and then Test Error and Test Result
   - If the selected test is not in error, the display will show Test End and Test ID.

The following tests have multiple error words. For these tests a delay of about one second separates the multiple error words displays.

Test ID	Error Words
02 (RAM test)	4 (1 if part of comprehensive)
0B (MEX echo)	3 (1 if part of comprehensive)
0C (IO echo)	3 (1 if part of comprehensive)

AUTOMATIC TEST REPEAT PROCEDURE

(CARD EXTENDED ENVIRONMENT)

All tests can be made to repeat indefinitely. This is a useful tool for waveform observation and signal tracing.When used for this purpose, the Freeze Display switch have to be OFF except as noted in following procedure

PROCEDURE:

1. Extend the card
2. Place D2-1 ON (Freeze Display)
3. Place D2-2 on (Repeat)
4. Select Test using Dip Switches D2-4 through D2-8
5. Depress push button switch: the LED will display
   Start of Test,
   Test Error and Test Result (if test is in error)
   Test End
6. Place D2-1 OFF : Display disable
   The selected test will now repeat indefinitely.

THE DISK TEST - (Test ID = 06)

Test 1

In this test data is written out to the FDC (F1) output ports and then read back and compared. The data patterns used are hex 55, 66, AA and 99.

Test 2

This test send a "Restore Command" to the Floppy Drive and check for a good result in Status Register of FDC.

Error Word (LED, ODT)

8	7	6	5	4	3	2	1

Bit	Significance
1	Error in test 1: TRACK Register of FDC at F1.
2	Error in test 1: SECTOR Register of FDC at F1.
3	Error in test 1: DATA Register of FDC at F1.
4-7	Unused
8	Error in test 2: Status Register of FDC in error after sending a Restore Command to Floppy Drive. May be a "Not Ready" if Drive is not ready.

Schematic Reference	Pages
Floppy Controller(FDC) F1	17
Interface Floppy D0,D1,F0,E1,E0	17

B1990 UPDATE SEMINAR

SOFT CONSOLE MODIFICATIONS

April 85

SCREEN DISPLAY - MENU SECTION

The screen display consists of a MENU SECTION followed by a DATA SECTION. Formats for the DATA SECTION are shown in Figures 1 thru 9, beginning on page 18. The format of the MENU SECTION is as follows:

```
COMMAND >                                                    <
 Mode. .Switches. .Display ..Reads.. .....Writes..... ......Actions.....
NORMAL MASTER: A  REG S16  SRnn:addr SWnn:addr=values CLEAR RESET UNLOAD
SONLY   SLAVE:OFF STK S24  CR:addr    CW:addr=values  MTR CLRELOG NOTEXT
CONLY  REMOTE:OFF CK  S39  BACK or -  register=data   CCLR  STEP  RUN GO
DISK   SINGLE:OFF CSE MAC  NEXT or +  ALLREGS=data    RC   SCREEN HTEST
FROZEN INTRPT:OFF OPR DIR  RDTEXT     TEXT characters LOAD AUTO [ON,OFF]
DISK: pack-name /file-name   SLAVE:ABSENT     MASTER:HALTED     ERROR
```

The COMMAND line is followed by a set of reserved words which may be entered on the COMMAND line to effect a MODE change, a SWITCH change, a DISPLAY change, a REGISTER-MEMORY READ-WRITE or an ACTION. The last line of the menu indicates SYSTEM STATUS as follows:

- o DISK: File-name indicates the reference file which is associated by default in a MAC to HOST xfer This name is used by default in the MTR, LOAD & AUTO command.
- o SLAVE: Status can be ABSENT or OFFLINE or HALTED or RUNNING.
- o MASTER: Status can be HALTED or RUNNING.
- o ERROR: Indicated if any bit of the master processor's PERP or PERM registers was true when the processor halted or if any ERROR condition arose during the last interaction of the processor with the maintenance control or whenever the MASTER processor is RUNNING despite having received a HALT request.

ACTION COMMANDS

The SCREEN command or the SPCFY key will cause a redisplay of the most recently displayed screen. This command is useful to restore a screen which had been altered or cleared by the operator.

The CLEAR command will clear the master processor. It will cause the slave processor to CLEAR only if online or to RESET if offline (See RESET below). CLEAR will cause a running slave processor to halt. CLEAR will clear the following registers : PERP, PERM, MSSW, CC, CD, INCN, A, BR, "M", ELOG and also the CACHE if mode is not CONLY.

The CCLR command will cause all VALIDITY bits in the CACHE Memory of the master processor to be cleared.

The RESET command will cause a CLRB signal to be issued to the master processor (also slave), to memory and to the I/O. RESET will clear the following registers : PERP, PERM, MSSW, CC, CD and INCN. CLRB will also cause a running slave processor to halt.

The CLRELOG command will cause the ELOG in the memory subsystem to be cleared.

The UNLOAD command will cause the head of the disk drive to restore , the file pointer is positioning on the first record and will cause subsequent commands, if dependent upon the completion of the unload, to be delayed until the restore is completed (e.g., RUN in DISK mode).

The DIR command loads directory from the floppy disk to MAC and DISPLAYS directory.

The LOAD "<file-name>" command sets DISK mode, search the directory for the file-name and if successfull search, read the first file sector. LOAD will not cause RUN to be executed automatically.

The MTR "<file-name>" will cause the directory to be reloaded in memory, CLEAR the system ,and execute a LOAD "<file name>" command.

The AUTO "<file-name>" ON/OFF modifies on the disk the reference file name associated with the MTR, LOAD command, and/or the variant status.
If variant is ON or not specified, then after each POWER-ON or HTEST, the MTR "<file-name>" GO commands will be executed without operator intervention .
If file name is not specified then no change occurs regarding the name of the reference file.

MODE COMMANDS

NORMAL is a mode, which if on when RUN is executed, causes the processor to obtain its string of M-instructions in a normal manner, i.e. from Cache. Instructions not in Cache are automatically loaded to Cache from S-Memory.

CONLY is a mode, which if on when RUN is executed, causes the processor to obtain its string of M-instructions from Cache only. Instructions not in Cache will cause a halt.

SONLY is a mode, which if on when RUN is executed, causes the processor to obtain its string of M-instructions from S-Memory only.

DISK is a mode, which if on when RUN is executed, causes floppy data to be transferred to the master processor. Floppy data may be M-instructions or M-instructions followed by a data field.

FROZEN is a mode, which if on when RUN or STEP is executed, causes the master processor to retain the micro-instruction contained in the M register after execution of the command. For RUN with SINGLE micro OFF, the micro-instruction is executed repeatedly until halted via the Halt switch on the cabinet. For STEP or for RUN with SINGLE micro ON, the micro-instruction is executed once. The A register is incremented for each execution.

CNS REGISTER

This register is used by the processor as a source or a destination. When used as a destination, hex values may be passed to the maintenance control to effect an action as noted below. Responses to the requests are returned in CNS immediately or after completion of the requested action as indicated below.

REQUEST	CNS VALUE	CNS RESPONSE	ACTION
Enable Commands	000081	000081	Subsequent values of 000001-000006 and 000040-000042 moved to CNS will be interpreted as commands. Commands 000040-000042 are valid for master only.
Enable U Parity	000082	000082	Subsequent reads of floppy data will have a parity bit included, suitable for writing to cache.
Disable Commands	000083	000083	CNS command interpretation is disabled. Also disabled are any pending deferred responses. Disabled is the normal or default state.
Disable U Parity	000084	000084	Floppy data transferred to the processor will not include a parity. This is the normal or default state.
Set "keep-page"	000085	000085	This command sets the "keep-page" toggle. If this toggle is true when the processor halts, a message will be displayed on the 25th line of the ODT, indicating that the status information is available on page 1. The keep-page toggle is set false each time the processor is started.
Set Interrupt	000001	000000	Set interrupt toggle ON. Effective only for processor performing the move to CNS.
Get Status	000002		Return a status vector (see below).
Halt Slave	000003	000000	Set slave's halt request.
Halt Processors	000004	000000	Set both master's & slave's halt request.
Reset Interrupt	000005	000000	Reset interrupt toggle. Effective only for the processor performing the move to CNS.

Get Baud Rate	000006	Baud Rate	Returns the presently set Remote Link Async baud rate. (see notes below)
Reload Disk	000021	lluu00	Read FAT(file allocation table) and 9 sectors of directory ll = lower byte of buffer address uu = upper byte
Specify file name			
byte 1 & 2	aabb22	------	update first and second bytes of MTR file name. aa = first byte,bb = second
byte 3 & 4	ccdd23	------	update third and fourth bytes of MTR file name. cc = third byte,dd = fourth
byte 5 & 6	eeff24	------	update fifth and sixth bytes of MTR file name. ee = fifth byte,ff = sixth
byte 7 - 8 and search	gghh25	lluuss	update seventh and eighth bytes of MTR file name and search for file name pattern gg = seventh bytes,hh = eighth ll = lower byte of buffer address uu = upper byte,ss = search status
Read Diskette	ldud26	lluurs	read disk. ld = lower disk address, ud = upper address,rs = read status.
Write Diskette	ldud27	lluuws	write disk.ld,ud = same read. ws = write status.
Read Ram Data	lluull	d1d2d3	Read RAM. ll = lower address, uu = upper address, d1,d2,d3 = first,second,thirth data.
Write Ram Address	lluul2	------	Write ram address
data	d1d213	lnun--	write data on ram d1 = lower byte data d2 = upper byte data,ln = lower byte of next address,un = upper byte.
Halt Restart	000040	000040	Master is re-started after a halt. CNS is set to 000000.
MTR Restart	000041	000041	A CLEAR, set DISK mode, UNLOAD and RUN (restart) is executed after the processor halts.
ALLREGS Restart	000042	000042	All processor's registers (including STK) are set to tthe value of the X register, except CNS, which is set to 000000. Action is after processor halts.
Invalid	others		Treated as Disable commands, Disable U parity, and Reset keep-page. Any pending deferred command is lost.

Commands 000001-000042 are recognized only if an Enable Command (000081) has been executed.

Commands 000081 to 000085 are completely independent.

Commands 000040-000042 are valid only for master processor. They are ignored by a slave.

Only one CNS command can be retained; therefore the deferred action of only the last received command will be done. This is especially important in using the restart commands (000040-000042).

The Baud Rate returned is not defined if link is set for Synchronous operation. The format of the Baud Rate is BCD. Presently allowable baud rates are 300, 1200 and 1800.

B1990 UPDATE SEMINAR

ANNEXES

MACII PPI'S ASSIGNMENT

	7(MSB)	6	5	4	3	2	1	0(LSB)
PPI1A @90 OUT	STROBE	REMOTE H/R TOGGLE	ECHO	END CHANNEL	DISK PARITY ERROR	LOCAL H/R TOGGLE	TOGGLE SELECT 2	TOGGLE SELECT 1
PPI1B @91 IN	SLAVE OFF LINE/	RUN/	A CPU/	HALT REQ	OTHER CPU RUN/	MTR/	A CPU MASTER	B CPU IN/
PPI1C @92 OUT	MODE SEL.3	MODE SEL.2	MODE SEL.1	FROZEN M.	SYSTEM CLEAR	DTR/	SLAVE OFF LINE	MASTER TO SLAVE LINK
PPI2A @94 IN	TIMER2 (EVENT TIMER)	TIMER1 (RATE TIMER)	REAL TIMER		MASTER TO SLAVE LINK/			CLOCK 0=3Mhz 1=4Mhz
PPI2B @95 IN	MUXED TEST RSLT	SLAVE TO MASTER LINK/		MXS2	MXS1	MXS0	H>MX	H<MX/
PPI2C @96 OUT	INTRPT	ENABLE CLK 2	32US CLOCK SOURCE	FRONT PLANE LED	HALT REQ OVER RIDE	SLAVE TO MASTER LINK	ENABLE CLK 1	ENABLE CLK 0

MAC-II PPI'S BIT ASSIGNMENT (CONTINUED) :

Significance of toggle select :

Toggle sel.1	Toggle sel.2	Strobe	Meaning :
x	x	0	No action
0	0	1	No action
0	1	1	Toggle master FF
1	0	1	Toggle MTR FF
1	1	1	Force memory bad parity

Significance of mode select :
Select the source of data & clocks into the USART.

Mode sel.3	Mode sel.2	Mode sel.1	Meaning :
0	0	0	TDI
0	0	1	Clear memory parity error
0	1	0	Rs232 asynchronous
0	1	1	Rs232 synchronous
1	0	0	Loop test, card in system
1	0	1	Loop test, card not in system
1	1	0	Send 0 (MAC test summary)
1	1	1	Send 1 (MAC test summary)

Significance of MEX select function :

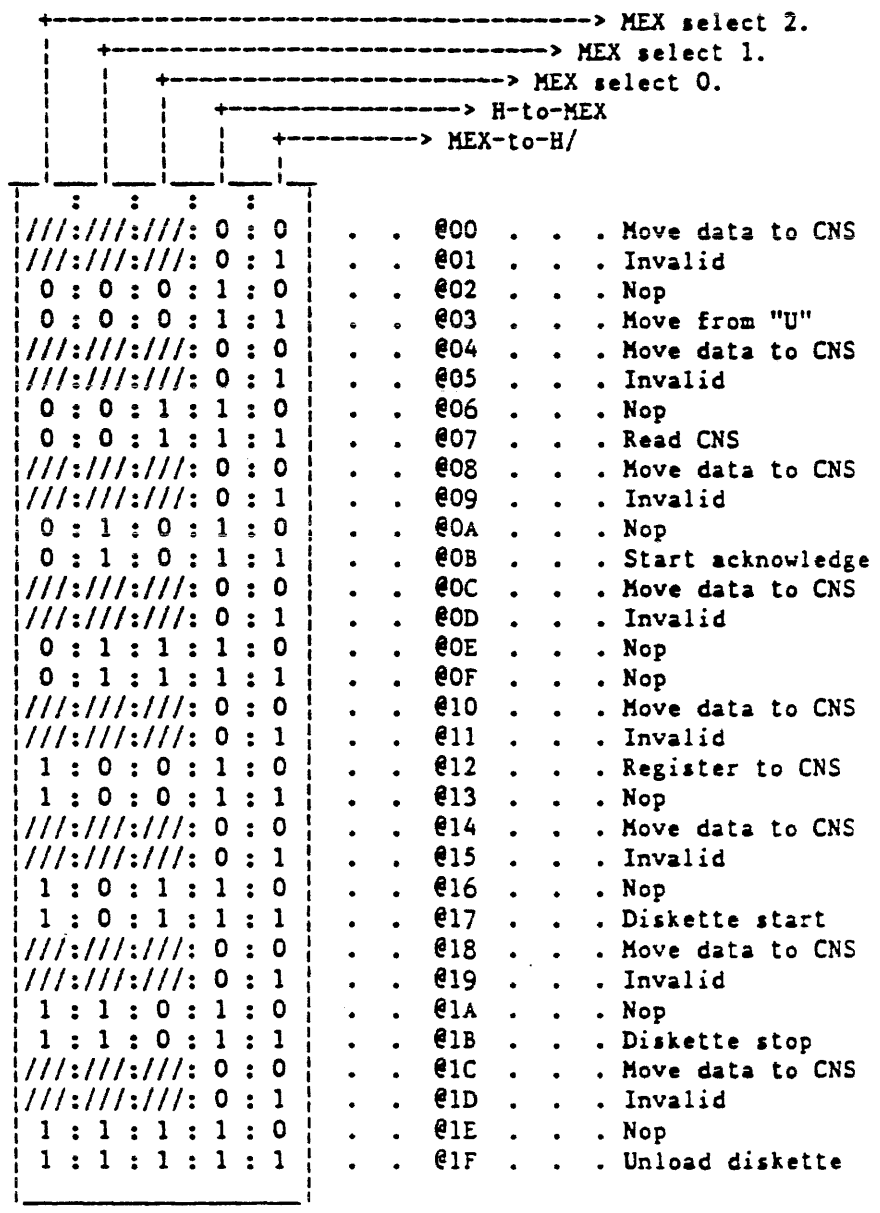
MXs2	MXs1	MXs0	H->MX	H<-MX/	Meaning :
x	x	x	0	0	Write to CNS
x	x	x	0	1	No action
0	0	0	1	1	Read U register
1	0	0	1	0	Register to CNS
1	0	0	1	1	Read I/O bus
0	0	1	1	1	Read CNS
1	0	1	1	1	Diskette start
1	1	0	1	1	Diskette stop
1	1	1	1	1	Unload diskette
0	1	1	1	1	Read M register
0	1	0	1	1	Acknowledge

Initialization of I/O Ports of the H10 Card

PORT NAME	FUNCTION	I/O ADDRESS	CONTROL WORD	RESULT
PI1CT	PROCESS CONTROL PORT 1	93H	82H	A PORT = OUTPUT B PORT = INPUT C PORT = OUTPUT
PI2CT	PROCESS CONTROL PORT 2	97H	92H	A PORT = INPUT B PORT = INPUT C PORT = OUTPUT
IPICT	I/O EXCHANGE PORT	8BH	9BH	A PORT = INPUT B PORT = INPUT C PORT = INPUT
HPICT	MEX PORT, MAC TO PROCESSOR	87H	80H	A PORT = OUTPUT B PORT = OUTPUT C PORT = OUTPUT
HPICT	MAC TO PROCESSOR HEX PORT	83H	9BH	A PORT = INPUT B PORT = INPUT C PORT = INPUT
PTMRCT	TIMER CONTROL	A3H	36H	TIMER 0 = MODE 3 (RATE GENERATOR MODE)
		A3H	B0H	TIMER 2 = MODE 0 (INTERRUPT ON TERMINAL COUNT)
HSCIC	USART CONTROL	B1H	94H	USART = SYN MODE
		B1H	14H	USART = ASYN MODE

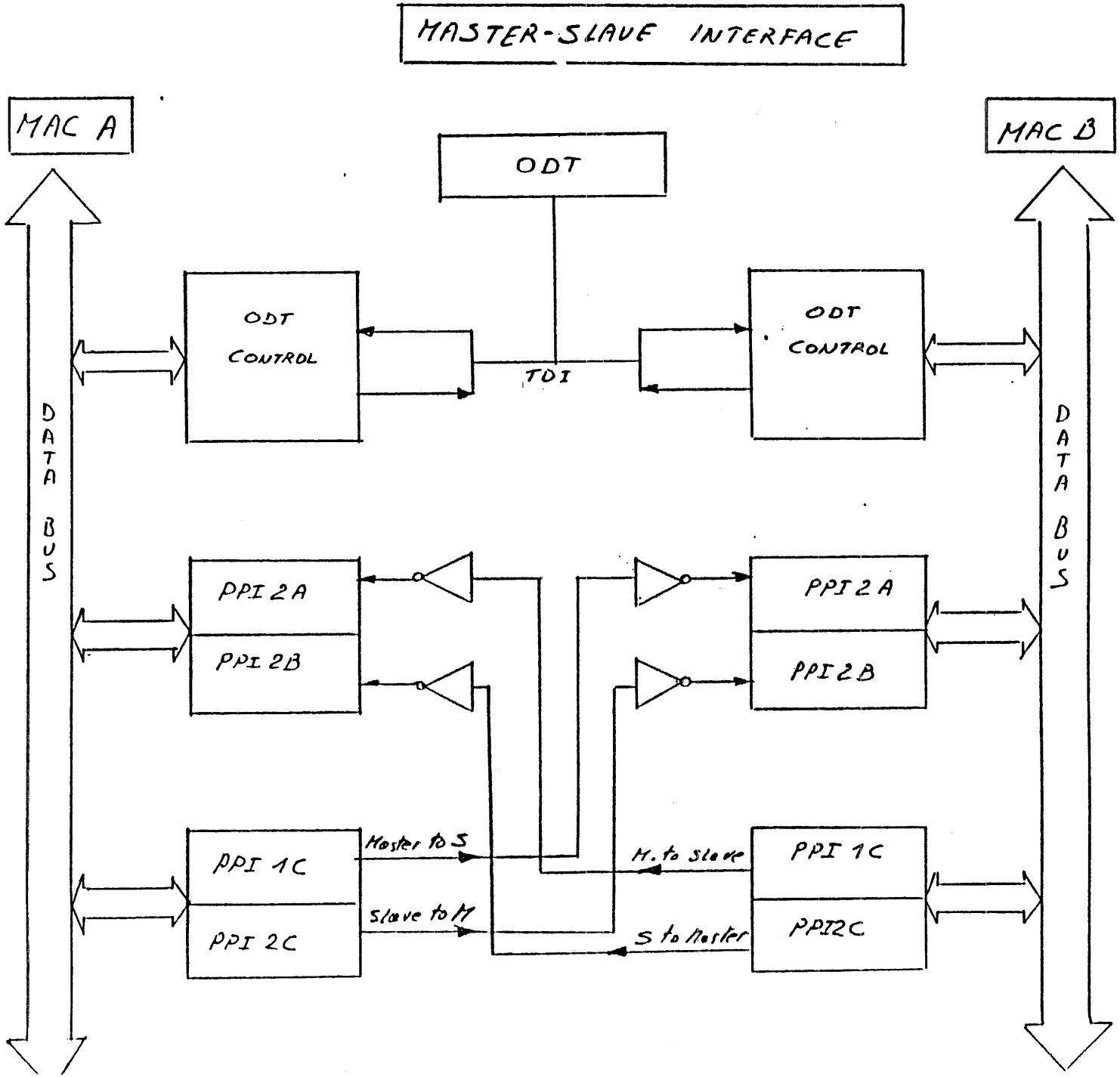
MEX SELECT SUMMARY.

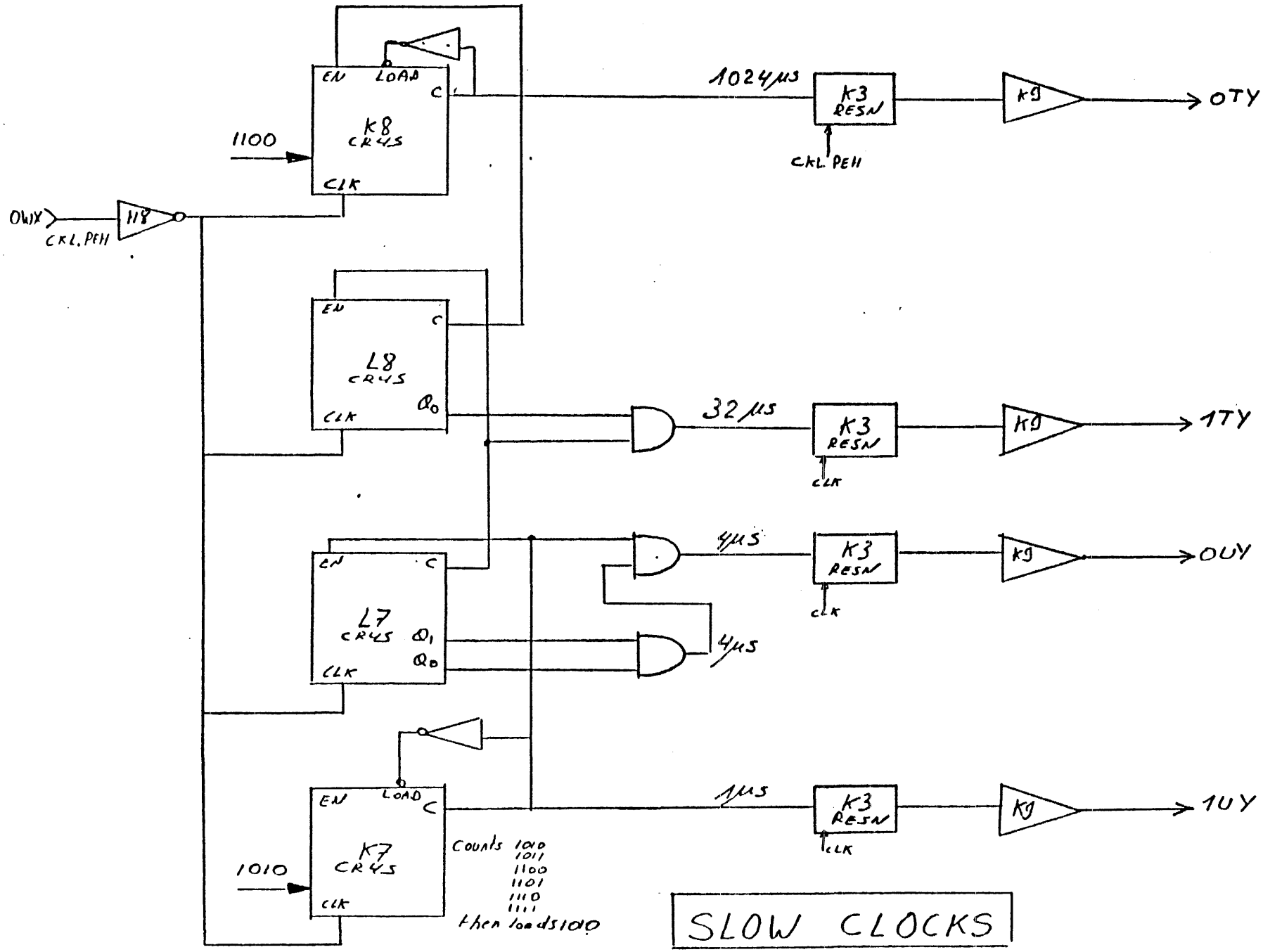
The following table may be useful whenever the MEX select lines (5 lsb's of PPI2b) are monitored to determine what functions are requested to the MAC.

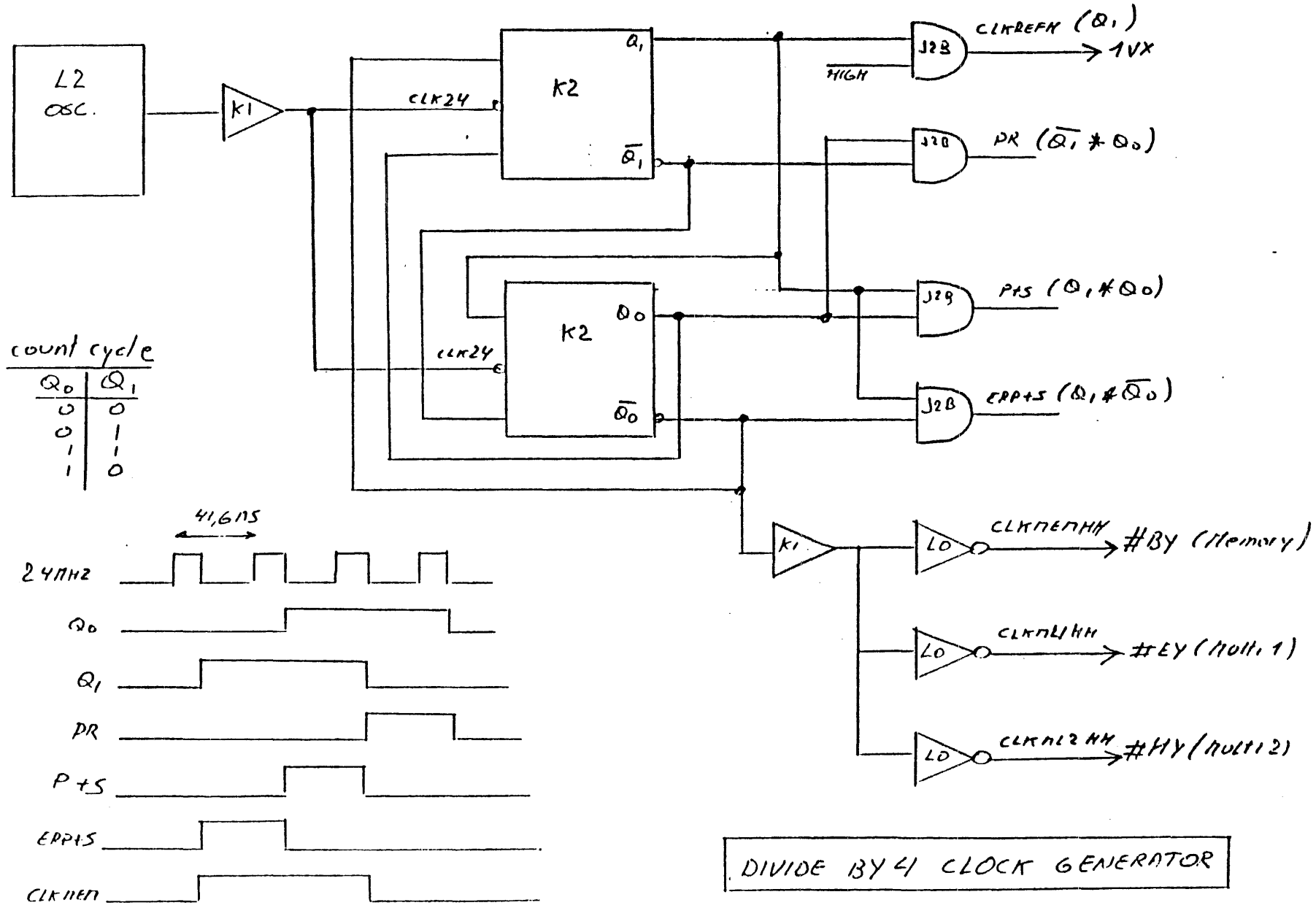


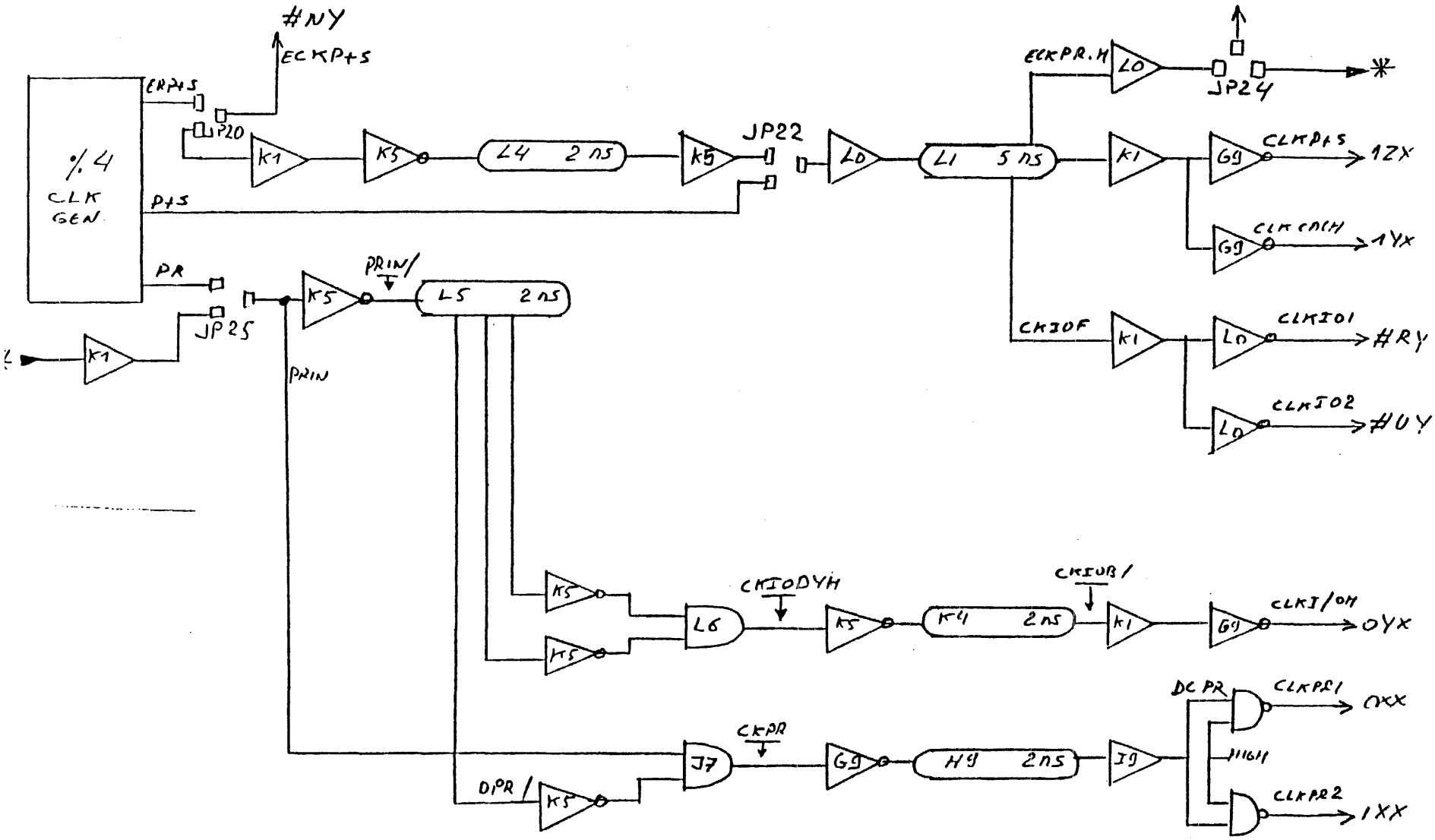
MEX select 2	MEX select 1	MEX select 0	H-to-MEX	MEX-to-H/		
///	///	///	0	0	@00	Move data to CNS
///	///	///	0	1	@01	Invalid
0	0	0	1	0	@02	Nop
0	0	0	1	1	@03	Move from "U"
///	///	///	0	0	@04	Move data to CNS
///	///	///	0	1	@05	Invalid
0	0	1	1	0	@06	Nop
0	0	1	1	1	@07	Read CNS
///	///	///	0	0	@08	Move data to CNS
///	///	///	0	1	@09	Invalid
0	1	0	1	0	@0A	Nop
0	1	0	1	1	@0B	Start acknowledge
///	///	///	0	0	@0C	Move data to CNS
///	///	///	0	1	@0D	Invalid
0	1	1	1	0	@0E	Nop
0	1	1	1	1	@0F	Nop
///	///	///	0	0	@10	Move data to CNS
///	///	///	0	1	@11	Invalid
1	0	0	1	0	@12	Register to CNS
1	0	0	1	1	@13	Nop
///	///	///	0	0	@14	Move data to CNS
///	///	///	0	1	@15	Invalid
1	0	1	1	0	@16	Nop
1	0	1	1	1	@17	Diskette start
///	///	///	0	0	@18	Move data to CNS
///	///	///	0	1	@19	Invalid
1	1	0	1	0	@1A	Nop
1	1	0	1	1	@1B	Diskette stop
///	///	///	0	0	@1C	Move data to CNS
///	///	///	0	1	@1D	Invalid
1	1	1	1	0	@1E	Nop
1	1	1	1	1	@1F	Unload diskette

No-op functions will cause the firmware to strobe the "ENDCNH" signal to the host CPU , while invalid commands will not.


MASTER-SLAVE INTERFACE
MAC A
MAC B
ODT
ODT CONTROL
TDI
ODT CONTROL
DATA BUS
DATA BUS
PPI 2A
PPI 2B
PPI 2A
PPI 2B
PPI 1C
PPI 2C
Master to S
Slave to M
M. to slave
S to Master
PPI 1C
PPI2C


EN
LOAD
C
K8
CRYS
CLK
1100
OWX
CKL. PEH
H8
1024μs
K3
RESN
K9
OTY
CKL PEH
L8
CRYS
Q0
32μs
1TY
CLK
L7
CRYS
Q1
4μs
OUY
K7
CRYS
1010
Counts 1010
1011
1100
1101
1110
1111
then loads 1100
1μs
1UY
SLOW CLOCKS


L2
OSC.
K1
CLK24
K2
Q1
Q1
K2
Q0
Q0
CLK24
HIGH
J2B
CLKREFH (Q1)
1VX
J2B
PR (Q1 * Q0)
J2B
P+S (Q1 * Q0)
J2B
EPP+S (Q1 * Q0)
count cycle
Q0 Q1
0 0
0 1
1 1
1 0
41,6NS
24MHZ
Q0
Q1
PR
P+S
EPP+S
CLKMEM
K1
LO
CLKMEMHH
#BY (Memory)
LO
CLKMLIHH
#EY (Mem 1)
LO
CLKML2HH
#HY (Mem 2)
DIVIDE BY 4 CLOCK GENERATOR



CLOCK GENERATION