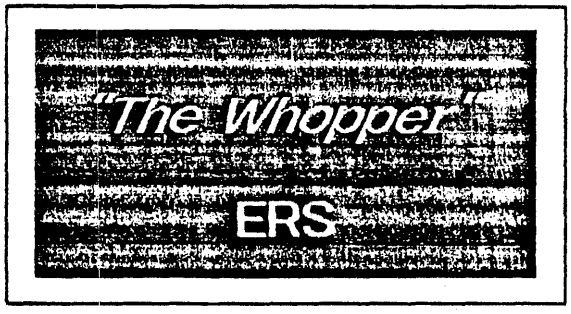


Final Version 2.0

July 14, 1983

Please direct your comments in writing to:

Gary Marten MS 2R

Paul Baker MS 2R

Ann Nunziata MS 2R

Approvals

_____

Engineering

_____

Hardware Engineering

_____

Software Engineering

_____

Marketing

_____

Manufacturing

_____

Mass Storage Division

Approval of "The Whopper" project by the above releases "The Whopper" as defined by this document. All future changes, if any, must be agreed upon by above.

The Concept

The Lisa has evolved to be a fairly complex and expensive system to build. Our initial idea of what the hardware would be has grown significantly due to software requirements and has greatly increased the cost of the Lisa system to Apple and the customer.

The Lisa system has numerous boards in it. The boards/subsystems included in the Lisa assembly are:

- o 1 - CPU
- o 1 - I/O
- o 2 - 1/2 MB Memory cards
- o 1 - Power Supply
- o 1 - Video
- o 2 - Floppy Disk Drives
- o 1 - Profile Hard disk assembly

The goal of the Whopper is to repackage the above hardware in the existing Lisa package in such a way that the packaging design is minimized while the same functionality is provided to the user for lower cost. In specific, the Card Cage, power supply, video board, and the Disk assemblies would be changed while much of the package and the keyboard would remain the same. The Lisa card cage contains four plug-in cards; the Whopper logic will be contained on two cards. In addition a hard disk can be installed in place of one of the floppy drives, thus eliminating the cost of one floppy and the Profile packaging and power supply.

The final Whopper System would thus have the following configuration:

- o 10MHz 68010 processor
- o MMU with 512 byte pages and 4 MB real address space
- o 720 x 544 video (with 720 x 364 mode)
- o 128K ROM (maximum)
- o 896K RAM (maximum without expansion card)
- o Built-in Sony Floppy Disk drive
- o Built-in Widget Hard Disk drive
- o 2 Built-in Serial ports
- o Keyboard/mouse interface with TOD clock

To enhance the compatibility between Whopper and Mac, a "square dots" video mode is provided, the Sony Floppy Disk drive is used and a "Mac-like sound" generator is provided. Considerable software effort will still be required to make Mac applications run on the Whopper.

In addition to the electrical design the Whopper will require mechanical design to provide new sheet metal for mounting the disk drives and different card cage components. A new power cable harness will also have to be designed. Cost reduction is an important part of the Whopper project and these parts will be redesigned in the most cost effective manner possible.

Differences in the Whopper

A significant space reduction in the Whopper comes from the floppy disk controller, which in the Lisa 1 is about 30 IC's and occupies about 1/2 the I/O board. This is reduced to the single IWM chip and several I/O bits in 1.75. This will require the 68010 to control the data transfers to and from the disk directly rather than relying on the 6504 as is done in the Lisa. Careful driver design will be required to prevent interference between the floppy disk, Applebus and other serial communication. To ease this design effort a general purpose timer chip will be included in the Whopper. This chip will provide three 16 bit timers under software control. These timers will make it possible to sequence the disk head stepper motors under interrupt control rather than using wait loops. In addition, the timers will make it possible to read a sector, determine the distance to the desired sector, and set the timer to interrupt just before that sector passes, avoiding the need for the driver to wait until the desired sector passes.

A built in hard disk will be provided in the Whopper. In order to make room for the hard disk mechanism the upper floppy disk drive will be eliminated, saving the cost of the second floppy.

Since the hard disk is built-in and all printers will be serial, there is no longer a need for a built-in parallel port. Removing this frees up both board space and I/O pins thus making the two board Whopper possible.

Parameter memory has been eliminated. With a built-in hard disk this is no longer necessary. Seven bits of parameter memory will be provided in the COPS chip to store the default boot device (this will be stored as 7 data bits and a parity bit). Battery backup of the time and parameter memory will also be eliminated, saving board space and cost. The +5 Standby supply will be retained so the clock will only have to be reset when there is a power failure or the system is unplugged.

Total RAM is decreased from 1MB to 896KB, this is due to board space limitations. Also, main system RAM can be installed in increments of 128K bytes, thus making system memory sizes of 256K, 384K, 512K, 640K and 768K possible. Programs that work on the Lisa may not work on the Whopper however the intention is that the applications and high level parts of the OS should work on both systems; drivers will have to change but may sense the operating environment and adjust their actions accordingly.

Additions and Enhancements

The 68010 has been chosen because it supports instruction restart and fast move loops.

A 10 MHz CPU clock has been selected because 10MHz parts are now available, it's faster, and 10MHz is 1/3 the 30MHz clock used in the video circuit.

The MMU in the Whopper will do both the relocation and check in parallel rather than serially, as is done in Lisa. In addition, separate video memory will be provided so that accesses to main memory will not have to wait for video accesses. The result of these two changes is that the memory cycle time will go from 800 nS per access to 500 nS in the Whopper. Note that this is slower than twice as fast as Lisa; even with the removal of the video access one wait state is introduced to perform the memory management function.

In addition to main memory, ROM accesses and I/O accesses will not be synchronized to video. By doing this, and by using faster ROMs, the ROM cycle time can be decreased from 800ns to 500ns. By using improved I/O interfaces I/O accesses can be reduced from 1600ns to as little as 400ns.

The built-in 10MB hard disk provides more storage with faster access times at lower cost compared to the Profile hard disk. The Whopper will make use of the HSD designed controller that will support 2:1 interleave. Cooperation between POSD and HSD will be required to achieve 2:1 interleave; if a data transfer rate higher than 1 MB/sec is required the result will be 3:1 interleave. This decision was made to reduce the risk associated with designing another disk controller.

More ROM has been added to provide space for commonly used code and I/O device drivers.

Block Diagrams

Attached to this description are two block diagrams "Whopper CPU" and "Whopper MMU and Memory". These block diagrams show the division of components to the two logic boards in the Whopper system.

ROM

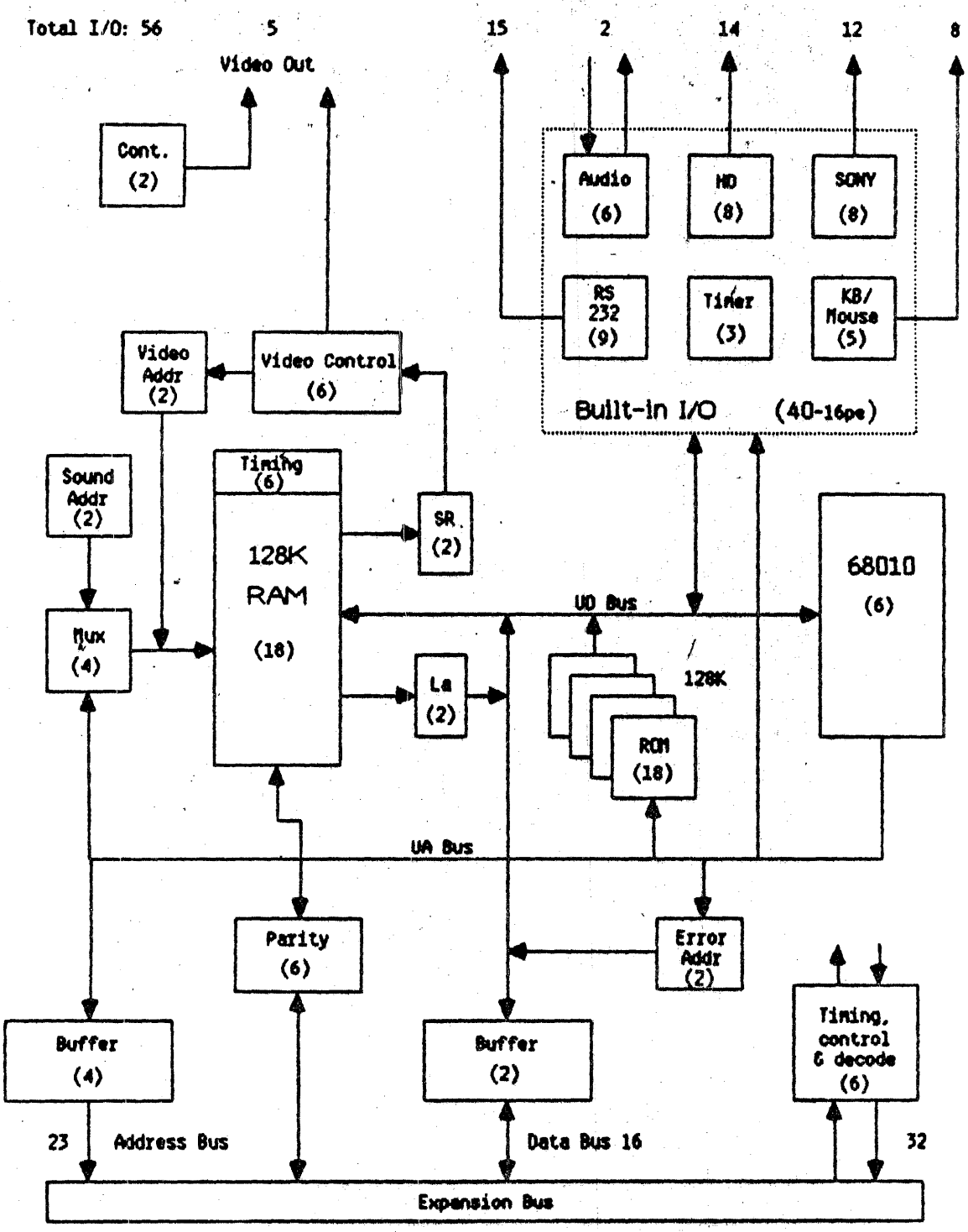
Up to 128 Kbytes of EPROM can be installed in the Whopper CPU board. This translates to four pairs of ROMs, 32 KB per pair, 8 ROMs total. Four pairs of ROMs are provided to permit the ROM code to be broken up into sections that can be ECO'd individually, making code maintenance easier. A possible division is:

ROM	Segment #	Address Range	Contents
ROM 0	0,32,64,96	$000000-$007FFF	Diagnostics and Boot Code
ROM 1	1,33,65,97	$020000-$027FFF	Quickdraw
ROM 2	2,34,66,98	$040000-$047FFF	Operating System - Drivers
ROM 3	3,35,67,99	$060000-$067FFF	Lisabug

ROM's can be accessed in the indicated address range if the SETUP bit is high, or if the MMU is set to map the ROM to the address. The two high order address bits are don't cares when decoding the ROM so the high nibble of the ROM addresses can be $0, $4, $8, or $C. If the MMU is used to map ROM addresses, the segment number used to map the ROM must one of the numbers indicated in the segment # column. Note that 32 KB is the maximum memory size that can be accomodated by the ROM sockets, smaller ROMs can be used if less code is required.

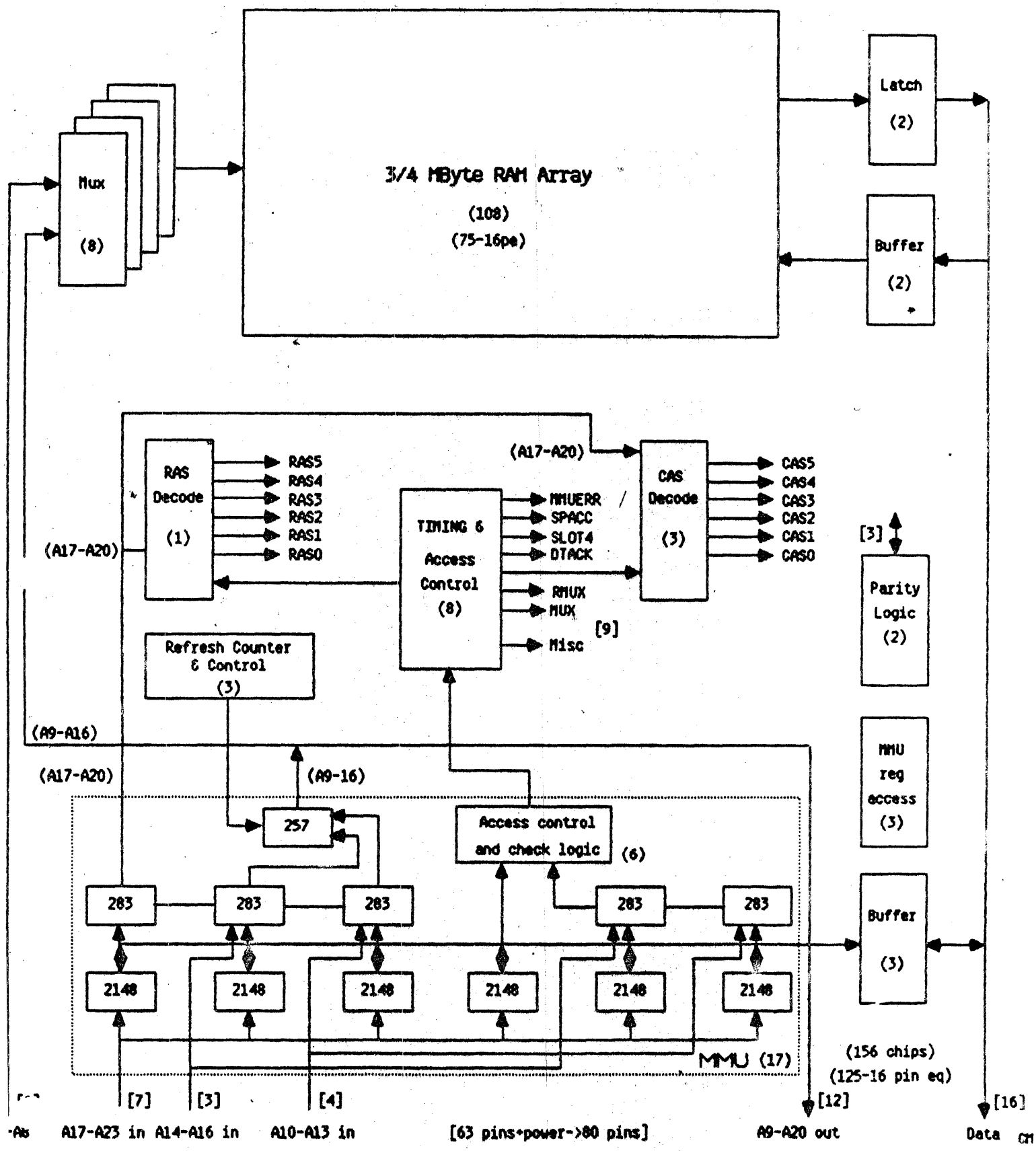
The system serial number will be stored in a PROM. There are three possible locations for this PROM: The video state machine PROM (as in the Lisa), a small PROM installed specifically to store the serial number, or in one of the ROM's that store the system code. The order of preference is the same as the order presented here; every attempt will be made to use option 1 or 2 rather than option 3.

The Whopper CPU Board



16 pin eq. ICs: 126

Bus:	71
I/O:	56
Power:	16
Total:	143


Mux
(8)
3/4 MByte RAM Array
(108)
(75-16pe)
Latch
(2)
Buffer
(2)
RAS
Decode
(1)
RAS5
RAS4
RAS3
RAS2
RAS1
RAS0
(A17-A20)
TIMING &
Access
Control
(8)
MMUERR
SPACC
SLOT4
DTACK
RMUX
MUX
Misc
[9]
(A17-A20)
CAS
Decode
(3)
CAS5
CAS4
CAS3
CAS2
CAS1
CAS0
[3]
Parity
Logic
(2)
Refresh Counter
& Control
(3)
(A9-A16)
(A17-A20)
(A9-16)
257
Access control
and check logic
(6)
MMU
reg
access
(3)
283
283
283
283
283
Buffer
(3)
2148
2148
2148
2148
2148
2148
MMU (17)
(156 chips)
(125-16 pin eq)
[7]
[3]
[4]
[12]
[16]
A17-A23 in
A14-A16 in
A10-A13 in
[63 pins+power->80 pins]
A9-A20 out
Data

Video RAM

128 KB of RAM reside on the CPU board. The first 48K of this RAM contain the video bitmap that is displayed on the screen.

The Video in the Whopper provides resolution of 720x544 in its normal mode. A Lisa compatability mode is provided to provide a 720x364 screen size so the Lisa programs will run on the Whopper.

The state of the video mode bit control the mcurrent mode of the video display. The following addresses in video RAM are used for the display:

Video Mode	VM bit	Screen Size	Address (in Segment # 30,62,94,or 126)
Normal	0	720x544	$000000 - $008F40
Lisa	1	720x364	$001FA4 - $009F9C

The state of the video mode bit can be read in the system Status Register 1, bit 0. The address of Status Register 1 is $003003 in Segment # 12, 44, 76, or 108.

Note that the segment # used can only be one of the four numbers indicated, mapping thru the MMU can only relocate the bits to one of the four segments and control access. Also note that when the SETUP bit is set to a 1, the main memory is not accessible, so the video RAM is the only RAM available. Thus ROM programs that set the MMU will store their data structures in a quadrant of the video RAM. As will be mentioned later a part of the video memory is set aside for use by the sound generation system.

Like the Lisa system, video and CPU accesses to video memory are interleaved. However, in the Whopper, during the horizontal and vertical retrace intervals the CPU has access to video memory and the video circuit never makes any accesses, permitting a higher CPU access rate. In the SERVANT application, where there is no video circuit, the video circuit can be programmed to access the video RAM only fast enough to refresh the RAM, making the video RAM have the same performance as main memory. Note that accesses to main memory (RAM other than video RAM) are not interleaved and will be performed in 500 nS.

The video circuit can be programmed to generate a level 1 interrupt at the start of vertical retrace. The state of the interrupt flag can be checked by reading system status register 0, bit 2. The addresses are:

Vertical Retrace Interrupt Enable: Read $003000
Disable: Read $003000

All accesses in Segment # 12, 44, 76, or 108

Hard disk Interface

The hard disk interface provides a parallel port that can talk to an interface that is similar to the Profile interface. The built-in hard disk will have a Profile+ controller which will be installed with the drive in the disk cage. The Profile+ controller can accomodate either a 10 or a 20 Mbyte disk drive. The interface consists of an 8 bit bidirectional data bus, a strobe and data direction control line , two handshake lines and the reset line. In the Whopper implementation, the strobe is generated by accessing the address of the data port and the data direction control line is just the state of the system Read line. Thus to transfer data to or from the disk controller a program simply writes to or reads from the data port and the control signals to perform the transfer are generated. The two handshake lines BSY and CMD are controlled by reading or writing bits in the status and control registers. BSY can be read in bit 7 of Status register 1 at location $003001 in Segment # 12,44,76, or 108. CMD is controlled as follows:

CMD	Set to low:	Read $003060 All accesses in Segment # 12,44,76, or 108
	Set to high:	Read $003070

In addition to reading the state of the BSY signal, a negative edge on the BSY signal will generate a level 1 interrupt; the state of the hard disk interface interrupt can be read in bit 3 of Status register 0 at location $003001 in Segment # 12,44,76, or 108. The hard disk interrupt may be disabled or reset by accessing the Hard Disk Interrupt Mask in the control register. To reset the interrupt, disable then enable the interrupt. The bit is controlled as follows:

HD Int Mask	Enable:	Read $003007 All accesses in Segment # 12,44,76, or 108
	Disable:	Read $003006

When data is transferred to or from the hard disk the parity of the data is monitored. When a parity error is detected a flip flop is set. The state of the flip flop can be read in bit 6 of Status register 0 at location $003001 in Segment # 12,44,76, or 108. The parity flip flop is reset or disabled by accessing a location in the control register. To reset the parity flip flop parity is disabled then enabled. The following controls the parity enable:

HD Parity Mask	Enable:	$00300F All accesses in Segment # 12,44,76, or 108
	Disable:	$00300E

The system RESET signal is connected to the hard disk interface so the hard disk controller will be reset when the Whopper is reset.

Floppy Disk Controller

The floppy disk controller is implemented using the IWM controller chip. In addition to the IWM chip, Timer 1 in the system timer is designed to be used in conjunction with floppy disk control, and two programmable bits are provided to control the floppy disk. Since the Whopper requires the 68010 to control the floppy disk several functions must be provided by the 68010. These include:

1 - When the 10 ms timer interrupt occurs, the interrupt handler should poll the drive to see if a disk has been inserted or if the disk eject button has been pushed.

2 - When the carriage is to be moved (e.g. when a seek occurs) the 68010 should program Timer 1 to interrupt when the phases must be rotated to the next position.

3 - When data is transferred to or from the disk the 68010 will perform the nibbleization and denibbleization and the data transfer via the IWM chip.

The reader is referred to the IWM chip documentation for programming details on the IWM chip. The 8 IWM registers are addressed as follows:

All accesses in Segment # 12, 44,76, or 108

CA0	High	$002003
	Low	$002001
CA1	High	$002007
	Low	$002005
CA2	High	$002008
	Low	$002009
LSTRB	High	$00200F
	Low	$002000
L-Motor-On		$002011
Drive-Sel		$002015
L6		$002019
L7		$00201D

Floppy Disk Head Select	Side 0:	Read $003040
	Side 1:	Read $003050
Floppy Disk Mptor	On:	Read $003005
	Off:	Read $003004

Note that the system A1 (second from LSB) is connected to the A0 input of the IWM chip.

Since the Whopper uses the Mac Sony Floppy Drive, the motor speed must be controlled as a part of the disk interface. The speed information is sent to the floppy drive as a pulse width modulated signal which can be filtered (inside the drive) to produce a reference voltage to control the motor speed. Attached to this document is the document "3.5 Inch Sony Disk Interface" which describes the motor speed adjustment procedure. Basically the PWM signal is sent to the drive and the tachometer in the drive is monitored to determine the speed. If the speed is not correct, the PWM signal is adjusted and the tach monitored again. Since there is no motor speed adjustment in the disk drive, the absolute value of the speed number may vary from drive to drive (unlike the Twiggy which does have motor speed adjustment) and the driver will have to search to find the exact PWM value that gives the correct speed.

The data value that is supplied to the PWM circuit is stored along with the sound data value in a series of bytes in the video memory. For details of accessing this memory see the section on the audio circuit below.

Audio and Contrast Circuit

Software controlled contrast and speaker volume is provided in the Whopper. The contrast level and the speaker volume is stored in an 8 bit latch. The latch is written 8 bits at a time; three of the bits control the speaker volume, the remaining 5 bits control the contrast level. The latch can be written by writing to address $007001 in Segment # 12, 44, 76, or 108. The format of the latch is as follows:

Data Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Contents	Unused							Contrast					Volume			
								MSB				LSB	MSB			LSB

Note that the contrast level and speaker volume are both written at the same time; when one must be changed, the other must be rewritten. As in Lisa, when the 5 bit binary number in the Contrast field is all 0's the contrast is at its maximum, when the bits are all 1's the contrast is at its lowest. When the 3 bit binary number in the Volume field is all 0's the speaker volume is at its minimum, when the bits are all 1's the volume is maximum.

The TONE signal used to drive the speaker is derived from Timer 2 in the 8253 System Timer chip, using the Timer 2 output pin and setting the timer up in Mode 3 (square wave output). Unlike the Lisa, the 8253 timer can be programmed to produce a square wave whose period can be adjusted by simply changing a 16 bit timer value. Addresses to access the System Timer are given below in the section on the Timer.

In addition to the ability to generate square waves the Whopper has the ability to generate sounds from a sequence of bytes stored in the video memory. The bytes in the table are fetched, one word (two bytes) at a time at the end of every other horizontal scan line, or about once every 60 uS. One of the two bytes (the high byte) is used to produce the sound signal, the other byte (the low byte) is used to generate the PWM signal for the floppy disk. Since this table of bytes is sequenced synchronously with the video data, the vertical retrace interrupt will provide a signal that the end of the table has nearly been reached. The vertical retrace interrupt occurs 12 bytes (or about 720 uS) before the first byte in the table will be read out. This permits a program to change the data in the table to produce a time varying waveform that does not repeat at a multiple of 60 Hz (the video refresh rate). The location of the table in video memory is:

SOUND/PWM byte table: 00C000 - 00C238 in Segment # 30, 62, 94, or 126

The byte table is sequenced from 00C000 to 00C238 and then repeats.

As in Lisa the contrast output voltage will reset to maximum contrast, however the speaker volume will reset to minimum volume.

Parity Circuitry

the Whopper provides parity checking on accesses to system main memory and the video RAM. The parity checking circuitry is on the CPU board. When an error is detected, the address of the failing location is latched in the Error Address Latch, the Parity Error bit in the system status register is set, and a NMI is

generated. Parity checking can be disabled by setting the Parity Enable bit in the system control register to a 0. Setting the Parity Enable bit to a 1 enables parity checking. The Parity Enable bit is set to a 0 when the system is reset. The access address for the Parity Enable bit is:

Parity Enable	set to 0:	Read $003020	All accesses in Segment # 12, 44, 76, or 108
	set to 1:	Read $003030	

When a parity error is detected the NMI handler can determine that the source of the NMI is a parity error by reading the PE bit in the system status Register 0, bit 5. If a parity error has occured the bit will be a 1 and the Error Address Latch will contain the address of the first parity error to occur since the latch was last reset. If a parity error occurs during a DMA transfer the NMI will not be detected by the 68010 until the DMA transfer is complete. If several parity errors occur only the address of the first error since the latch was reset will be stored. The Error Address Latch and the PE status bit are reset by reading the Error Address Latch. The address of the Error Address Latch is:

Access in Segment # 12, 44, 66, or 108	
Error Address Latch:	Read a word from $007000

The Whopper only checks parity on CPU accesses, never on video accesses. Since the Error Address Latch is located on the CPU board and the physical address bus doesn't appear on the CPU board, the Error Address Latch latches the 13 most significant bits of the logical address.In addition to the 15 address bits, the Error Address Latch stores the state of the BGACK signal at the time of the error. The parity error NMI routine must read the Seg bits to determine the context and perform the mapping operation to determine the physical address of the failing memory location. If the BGACK bit is high the parity error occurred during a DMA transfer and Context 7 was used to map the access. This may be used by diagnostic programs to determine the failing word. The register must be read using a word read operation as two byte reads will result in one of the bytes containing the address after the latch has been partly cleared. If the PE status bit is not set, the contents of the Error Address Latch is not defined. The contents of the Error Address Latch are as follows:

Data Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Contents:	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	BGACK

System Timer

The system timer is provided by a 8253 type timer chip. This chip contains three timers whose use is a described below:

Timer 0 - This timer is to be used as the 10 ms interrupt timer. It will be clocked at 1.25 MHz and will produce interrupts at interrupt level 1. Using the Latching Count mode this timer may be read on the fly for accurate microsecond timing measurements. The state of the Timer 0 interrupt flag can be read in the system status Register 0, bit 0.

Timer 1 - This timer is to be used with the Floppy Disk controller. It will be clocked at 1.25 MHz and will produce interrupts at interrupt level 6. The state of the Timer 1 interrupt flag can be read in the system status Register 0, bit 1.

Timer 2 - This timer is to be used to produce the TONE output for driving the speaker. It will be clocked at 1.25 MHz. It does not produce interrupts.

The reader is directed to the Intel 8253 data sheet for programming details on the 8253 timer. The timer may be accessed at the following addresses:

```
All addresses in Segment # 12, 44,76, or 108

$004001  Read/Load Counter 0
$004003  Read/Load Counter 1
$004005  Read/Load Counter 2
$004007  Write Mode Register/Read is a NOP
```

COPS Keyboard, Mouse, Power, Clock and Timer Functions

The COPS in 1.75 is provides the same functions as in Lisa and operates in much the same manner. The COPS receives data from the keyboard, mouse and on/off switch, processes the information, and relays it on to the 68010 in a more convenient form. The COPS also provides a time-of-day clock and a timer/alarm for the system.

Sending Commands to the COPS

The COPS is designed to communicate with the 68010 via interrupts, however it can be treated as a polled device, although this requires more software. Commands are sent to the COPS by writing the command to the COPS directly; however, a command cannot be sent if the last command has not been processed or if the COPS is in the process of sending data to the 68010.

If the COPS interrupt is enabled there is no need to check for incoming data from the COPS since this will generate an interrupt. If the COPS interrupt is disabled then the interrupt line from the COPS must be checked before sending a command to the COPS. All interrupts must be disabled between the check of the interrupt line and the time that the command is written into the COPS. In addition, the time between these two events must not exceed 7us or the command may be lost occasionally. The flow chart on the sheet entitled "Talking to the 1.75 COPS" details this.

If there is no incoming data, then the COPS must be checked for command

processing. This is done by reading the COPS, if the last command has not been accepted by the COPS it will still be in the COPS register. If the last command was accepted by the COPS the register will have been set to an 'FF' by the COPS.

Reading Data from the COPS

Under interrupt operation reading data from the COPS is very simple. Once it is determined that the COPS is the interrupting device the 68010 reads the COPS register, this is the data. This register contains the data until it is written into by the 68010, which is the way the 68010 tells the COPS that it has read the data and no longer needs to read it from the COPS.

Note that with this protocol the data from the COPS can be read any number of times without losing it. Also note that the data in the COPS register may not become valid until up to 30us after the interrupt line becomes active. However, this time may tend to be masked by the normal interrupt handling functions performed by the 68010.

System Reset, NMI and Soft Power

In the Whopper the COPS controls the SYSTEM RESET function, as on Lisa, SYSTEM RESET is generated upon the following two conditions. The first is power-on, the COPS generates a SYSTEM RESET when the system is turned on either by pressing the power button or the power-on alarm turning the system on. The second is the reset button located on the back of the Lisa. In addition, the software can send a SYSTEM RESET command to the COPS to generate a SYSTEM RESET.

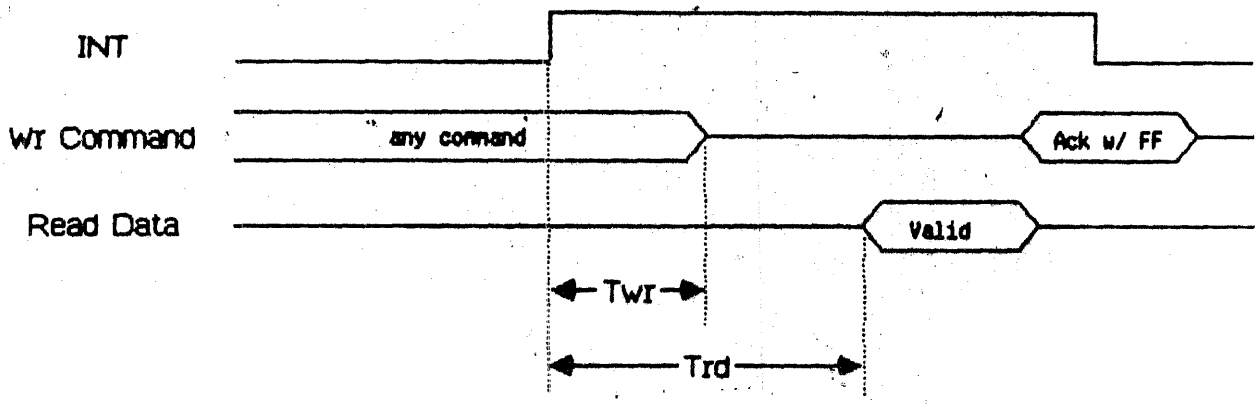
Note that as on Lisa the RESET button may be pressed at anytime, thus this *"use at own risk"* function can cause undesired results when pressed by the user. The reset switch will be located on the CPU board and will be activated by an button that will attach to the back cover through the vent slots. This button may be removed by the user or OEM without disassembling the Lisa, if desired.

Lisa provides a method of generating a NMI from the keyboard using a key on the keyboard. The Whopper will delete this function and provide instead a pushbutton switch on the CPU board that can be activated by an optional button that will attach to the back cover through the vent slots.

As on Lisa, the COPS controls the system on-off function. When the ENABLE PORT command is executed, the *soft-power* function is also enabled. After the *soft-power* function has been enabled after a SYSTEM RESET the only way to turn the system off (short of pressing the RESET button to restart the system and try again, or pulling the power cord) is for the software to send the COPS a TURN-OFF command.

The Clock and Timer/Alarm

The clock and timer/alarm in the COPS are the same as in Lisa. The clock is a 32 bit binary counter with resolution to the second. The timer/alarm is a 24-bit binary countdown timer that decrements every second, when it goes negative an 80 FC is set to the 68010. The timer can also be programmed to turn on the power, if desired.



Parm	Definition	min	max
Twr	Time to Write command after INT	---	7us
Trd	Time until Read Data valid	22us	30us

Note - If polling is used to determine if INT is true, all interrupts must be disabled during the polling for Command Write operation to the COPS. If the COPS interrupt is enabled then no polling of the INT line is required since the interrupt itself will prevent the command from being executed until after the interrupt is serviced.

Keyboard/Mouse COPS Commands

Command	Description
0000 0000	Turn Output Port on, enable soft-off
0000 0010	Read Clock Data
0001 nnnn	Write nnnn to clock
0010 spnn	Set Clock Mode s - enable clock set mode p - power on nn - 00 Clock/timer disabled 01 Timer disabled 10 Timer underflow interrupt 11 Timer underflow power on
0011 nnnn	Write nnnn to low keyboard indicator
0100 nnnn	Write nnnn to high keyboard indicator
0101 nnnn *	Set high digit boot device
0110 nnnn *	Set low digit boot device
0111 nnnn	AutoMouse timer value
10rr rrrr *	Read register rr rrrr
1100 nnnn *	Write nnnn to last register read (diagnostic only, see cops listing for register functions, WARNING: this instruction can cause unpredictable results)
1101 0001 *	Enable register write mode for next command (diagnostic only)
1101 0010 *	Reset Keyboard
1101 0100 *	Perform SYSTEM RESET Function
1111 1111 *	no operation
anything else	undefined, unpredictable results may occur

* new command

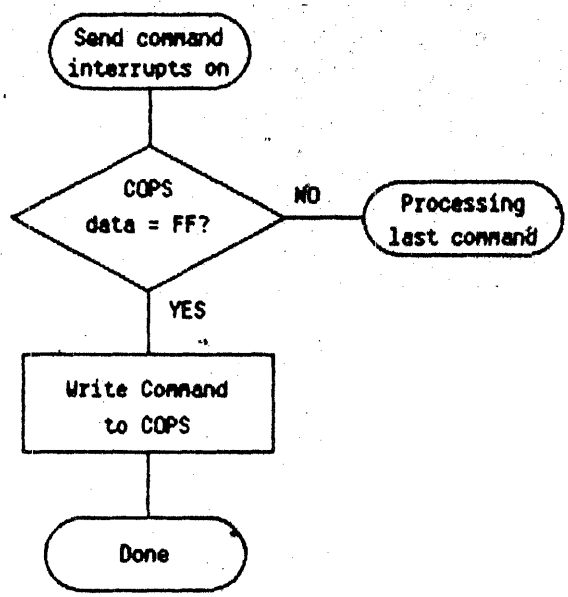
RESET CODES

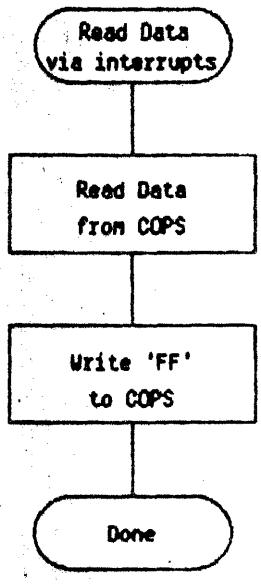
Reset Code	Definition
00 - BF	Reserved Keyboard ID codes
C0 - CF	Reserved for future use
Dx	Data x read from COPS RAM
Ey	Clock data follows (y=year)
F0 - FA	Reserved for future use
FB	Soft on/off switch pressed
FC	Timer Interrupt
FD	Keyboard disconnected
FE	CPU board COPS hardware error detected
FF	Keyboard hardware error detected

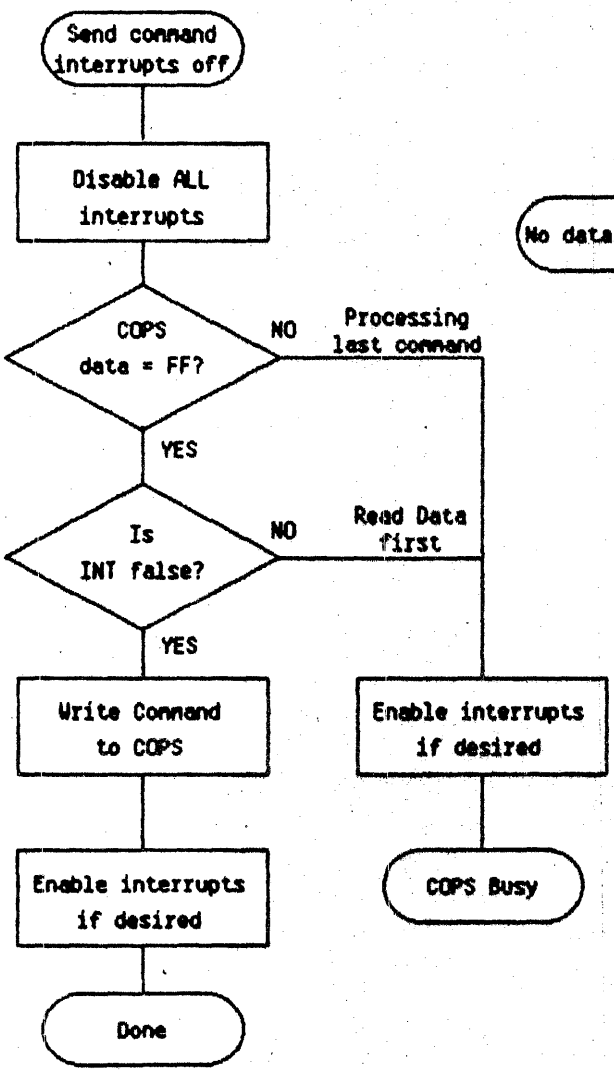
SYSTEM RESET FUNCTIONS

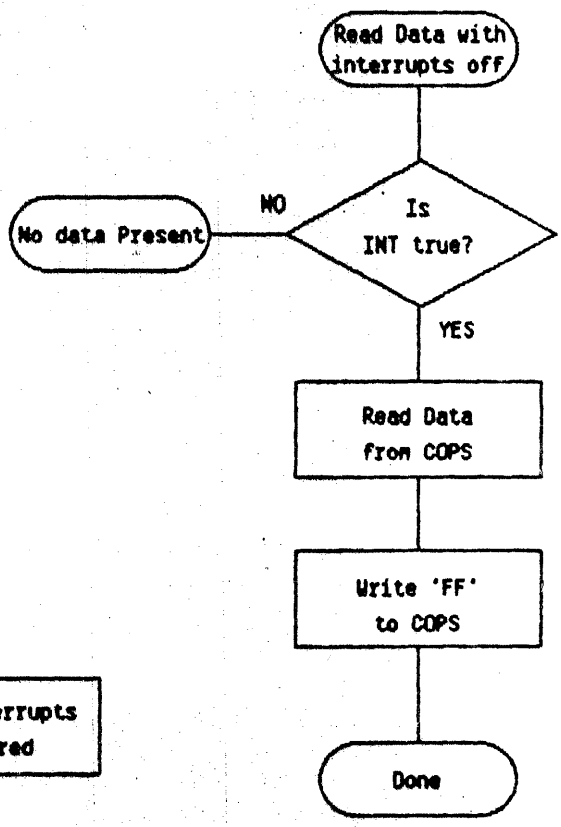
The following events occur following the release of the RESET button (or the on/off switch if the system was off) and are part of the SYSTEM RESET function:

Event	Data -> 68010	Comments
Reset 68010, I/O	---	Reset system
Disable port	---	Don't send data to 68010 until its ready to receive data
Disable Mouse	---	Don't read mouse until 68010 is ready
Reset Mouse switches	---	Reset mouse switch status to all UP position
Reset Keyboard	80 FD	If Keyboard unplugged
Keyboard ID	80 id	If Keyboard sends its ID


Send command interrupts on
COPS data = FF?
NO
Processing last command
YES
Write Command to COPS
Done


Read Data via interrupts
Read Data from COPS
Write 'FF' to COPS
Done


Send command interrupts off
Disable ALL interrupts
COPS data = FF?
NO
Processing last command
YES
Is INT false?
NO
Read Data first
YES
Write Command to COPS
Enable interrupts if desired
Done
Enable interrupts if desired
COPS Busy


Read Data with interrupts off
No data Present
NO
Is INT true?
YES
Read Data from COPS
Write 'FF' to COPS
Done

Serial Communications

the Whopper uses the Zilog SCC to provide two serial communication channels. One of the channels is configured with modem control lines, the other channel has minimal handshake lines but supports the Applebus drivers and receivers. Both channels are clocked at the same frequencies as the Lisa: Channel A is clocked at 4 MHz and Channel B is clocked at 3.6864 MHz. The signals supported by the two channels are:

Signal	Channel A	Channel B
TxD	yes	yes
RxD	yes	yes
CTS	yes	no
RTS	yes	no
DCD	yes	no
DSR	yes*	yes
DTR	yes	yes
Tx clock out	yes	no
Tx clock in	yes	yes
Rx clock in	yes	yes

* DSR on Channel A is read by reading the state of the DCD input on Channel B, or by reading the state of the SYNCA pin (asynch mode only), no interrupt may be generated by edges on DSR.

* The reader is directed to the Zilog SCC manual for programming details for the SCC chip. The addresses of the SCC registers are as follows:

All accesses in Segment # 12, 44,76, or 108

Channel A	Data:	$006007
	Control:	$006003
Channel B	Data:	$006005
	Control:	$006001

To permit self testing, a control bit that can disable the TxD A output has been provided. It is accessed as follows:

All accesses in Segment # 12, 44,76, or 108

TxD A/LED	Disable/On:	$003000
	Enable/Off:	$00300C

In addition to enabling the TxD A output, the bit controls a LED mounted on the CPU board. This LED is provided to permit diagnostics that run during board burnin to communicate results of tests. The LED will be off during normal system operation.

Built-In System Devices Addresses (SPECIAL address space)

Device Address (SetUp=1 or MMU Set here)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	Device Segment					0	64K byte device address block															

0 32/64K ROM (ROM0)
1 32/64K ROM (ROM1)
2 32/64K ROM (ROM2)*
3 32/64K ROM (ROM3)*

4-8 Undefined
C Internal I/O
D Exp Slot 1
E Exp Slot 2

F Exp Slot 3
10-1D Undefined
1E-1F Video RAM

* Optional, may not be installed

CPU Board Devices (Subject to change) (SetUp=1 or mapped via MMU)

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	0	1	1	0	0	0	0	Dev Select			4 KByte Device Address Space											

	READ	WRITE		READ	WRITE
0	Hard Disk	Hard Disk	4	Timer	Timer
1	(not defined)	(not defined)	5	COPS	COPS
2	Floppy Disk	Floppy Disk	6	SCC	SCC
3	Status/Control	(not defined)	7	Error Addr Latch	Video/Audio Levels

Write Control ad Location...

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	0	1	1	0	0	0	0	0	1	1				D				D				D

(Memory Board)		(CPU Board)		(CPU Board)	
0 No Change	4 ---	0 No Change	4 ---	0 No Change	4 ---
1 Seg 0	5 ---	1 Parity Enable	5 Video Mode	1 Wr Wr Parity	5 ---
2 Seg 1	6 ---	2 FD Head Select	6 Vertical Retrace	2 FD Motor On	6 LED
3 Seg 2	7 SetUp	3 HD $\overline{CMD}$	7 Disable ALL int	3 HD Int Mask	7 HD Parity Mask

Read Status

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	0	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1/$\overline{0}$	X
		1		8				7				0				0						0 or 2	

SU SetUp
SG2 Seg 2
SG1 Seg 1
SG0 Seg 0

HDB HD Busy
HDP HD Parity
PE Parity Error
COP COPS Int

HD Hard disk Int
VRT Vertical Int
T0 Timer 0 LP Int
T1 Timer 1 HP Int

Status Register 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	SU	SG2	SG1	SG0	HDB	HDP	PE	COP	HD	VRT	T1	T0

SL4 Status Slot 4
SL5 Status Slot 5
MMU MMU Error

VID Video bits
CS CSync
VM Video Mode

Status Register 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	SL4	SL5	MMU	X	X	X	X	X	VID	CS	VM	X

Status Register

As indicated in sections above there are various bits in the system that can be read and written. A summary of these bits is contained in the figure "Built-In System Devices Addresses". This figure shows the layout of the status registers and the locations that can be read to set the various control lines in the system. Note that the status registers are word wide with the low byte reporting the state of bits on the CPU board and the high byte reporting the state of bits on the Memory board.

The "new" Slot 4

The space that the Lisa memory cards occupied is now free for one additional card. A separate 2MB address space has been defined for this card. It is anticipated that the slot will be used for memory expansion and to provide a place for slave processors.

DMA

the Whopper supports DMA by using the same mechanism that is used in the Lisa. The 68010 bus master control lines are connected to the expansion slots and bus arbitration is performed on the CPU board. In addition to the three expansion slots, slot 4 can control the bus and are thus provided access to the bus master control lines. The arbitration scheme used is that the Bus Grant (BG) signal is daisy chained from slot to slot to establish the DMA priority. In the Whopper the priority is:

Highest	Slot 4
	Expansion Slot 3
	Expansion Slot 2
Lowest	Expansion Slot 1

Once the bus master has been established and the BGACK signal asserted, the new bus master can access memory. In the Whopper DMA controllers generate logical addresses which are mapped through the MMU using context 7. To generate a logical address requires 24 address bits which are generated as follows for Expansion Slots 1-3:

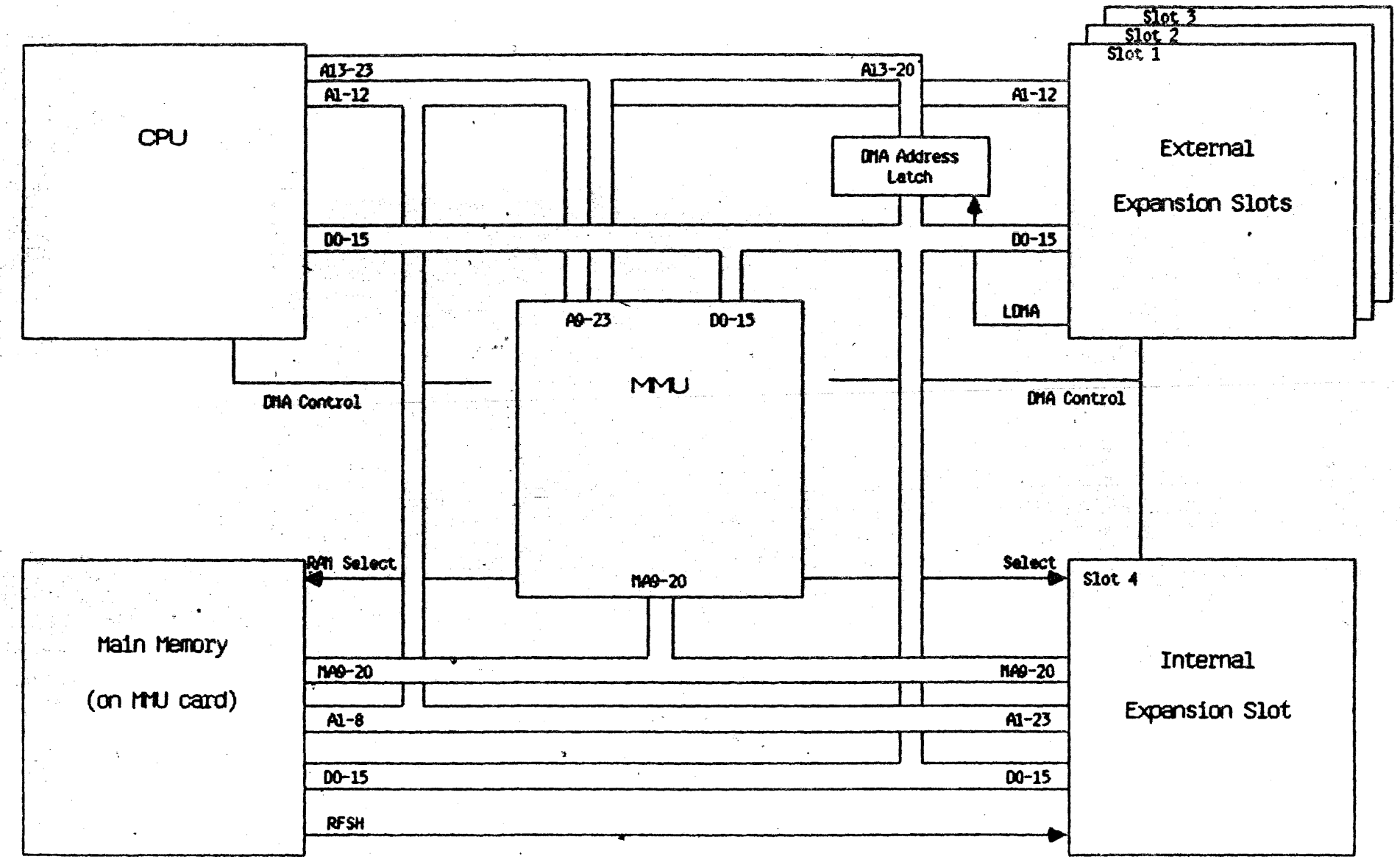
A23-A21 - Forced to 1 by DMA logic
A20-A13 - Generated by DMA latch
A12-A1 - Generated by DMA controller, connected to bus

For Slot 4 all address bit (A23-A1) are generated by the card in Slot 4.

The DMA latch is loaded by a rising edge on the LDMA signal as in Lisa. Note that since A23-A21 are forced to a 1 for Expansion Slots 1-3, DMA can be performed to video memory but not to other I/O or ROM and that since DMA is performed using Logical addresses the MMU can be used to map accesses to memory located in slot 4. Note that since all address bit are provided for DMA to Slot 4, any memory/IO can be accessed. Care is therefore required so that slot 4 devices don't inadvertantly change system status registers.

In addition, the MMU can control access to memory when DMA is performed. If an access violation occurs during a DMA transfer the data will not be transferred to memory and the MMU error bit, Status register 1, bit 9 will be set and can be checked at the end of the DMA transfer to determine if the transfer was not permitted.

WHOPPER BUS STRUCTURE



MMU Access -- Register Addressing Definitions

To access the MMU registers access this location with SetUp=1:

MMU Register Access Address:

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Segment								1	B/L	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(SetUp must be true)

1 = Base
0 = Limit

The 16-bit base address register is defined as follows:

MMU Base Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	STK	SP	EXP	Base Address											

(Subj to change)

0 Men RO	4 Unmapped	8 Men R/W	C Men Stack
1 Exp RO*	5 Undefined	9 Exp R/W	D Exp Stack*
2 Undefined	6 Undefined	A Undefined	E Undefined
3 Undefined	7 Undefined	B Spec R/W	F Undefined

* May not be implemented on all devices
Note: Do not set MMU to Undefined states

The 8-bit Segment limit register is defined as follows:

MMU Limit Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
not used								Limit							

Since slot 4 can contain either a bus master, which operates on logical addresses or memory which operates on physical addresses, both address buses must be present at the slot connector. Figure "Whopper Block Diagram" shows the buses in the Whopper. As can be seen, the physical address bus MA9-MA20 and the logical address bus A1-A23 are connected to slot 4 (duplicating A9-A20), which permits access to both physical and logical addresses. Note that it is possible to perform DMA from a card in slot 4 to memory located on that card.

MMU Address Spaces, control and protection

The Whopper MMU is very similiar to the Lisa MMU. There are three 2 MB address spaces. The three address spaces are:

Address Space	Function	Relocation	Limit Check	Read Only	Stack
MAIN MEMORY	Main Storage	yes	yes	yes	yes
EXPANSION	Additional RAM	yes	yes	yes	yes
	Additional I/O	yes	yes	"	" *
	Additional CPU	yes	yes	"	"
SPECIAL	Built-In I/O	limited	no	no	no

*Depends on device implenentation of slot access control and devices

The MAIN MEMORY address space and EXPANSION address space are each divided into 128 128KB segments with an incremental resolution of 512 bytes. These segments may be treated as READ ONLY or READ WRITE and as STACK or DATA/CODE Segments. Limit checking is performed on these segments identical to Lisa.

The third address space is the SPECIAL space, this addresses all devices on the CPU board and the Memory/MMU board. In this SPECIAL address space the addresses are not translated by the MMU but rather the MMU is used only to select a specific segment within this address space. The READ ONLY and STACK functions do not exist in this address space. Also note that devices in the SPECIAL address space may only appear in four different segments, for example, INTERNAL I/O may be mapped to Virtual Segment 12, 44, 76 and/or 108.

Invalid memory accesses through the MMU causes a 68010 BUS ERROR cycle. In addition, the MMU ERROR status can be read from the SYSTEM STATUS register. Note that a bus timeout also causes a BUS ERROR cycle.

MMU Register Contexts

The Whopper MMU contains up to eight MMU register sets, or contexts. This is twice the number as in Lisa. As in Lisa, context 0 is reserved for supervisor state access. Context 7 is reserved for DMA accesses since all DMA accesses are mapped through this context.

The current context is defined by the SEG0, SEG1 and SEG2 control bits. These bits can be set by reading the following locations in the STATUS/CONTROL section of the SPECIAL address space:

ADDRESS TO CLEAR	ADDRESS TO SET	ADDRESS TO READ	BIT TO READ	FUNCTION
003800	003900	003000	8	SEG0 bit
003A00	003B00	003000	9	SEG1 bit
003C00	003D00	003000	10	SEG2 bit

Note that the SEG bits can be read in STATUS REGISTER 0 as indicated above.

The 68010 provides an Alternate Function Code register that can be used to

generate accesses to data in the user's address space while operating in supervisor mode.

The SetUp bit

The SET-UP bit is provided so RAM can be mapped into segment 0. In order to access the MMU registers SET-UP must be set to 1. Note that when SET-UP is a 1, the 68010 can only access devices in the SPECIAL address space, MAIN MEMORY and EXPANSION memory cannot be accessed. Therefore any common data must be stored in a portion of the video memory.

The register drawings show how the bits in the MMU registers are laid out. Note that the Access Control bits are now associated with the Base register rather than the Limit register as in Lisa.

Note that SETUP is set to 1 by the hardware upon power on. The SETUP bit is accessed by:

SETUP BIT	Set to 1: Read from $003E00	All accesses in Segment #12, 44, 66, or 108
	Set to 0: Read from $003F00	

Interrupts

The interrupt system includes seven levels of priority interrupts defined as follows:

LEVEL	DEFINITION
7	NMI, Memory Parity
6	SCC, Timer 1
5	Expansion Slot 1
4	Expansion Slot 2
3	Expansion Slot 3
2	Expansion Slot 4
1	COPS, Timer 0, Vertical & disk

Parity Error Status, COPS, Timer, Vertical, and Hard Disk interrupt status can be read at the READ STATUS location.

To permit several instructions to execute "atomically" a control bit is provided that can disable all interrupts. This function should never be invoked for more than a few instructions. The bit is controlled as follows:

All Interrupts	Disable:	Read $003F00	All accesses in Segment #12, 44, 66, or 108
	Enable:	Read $003E00	

To interact with the COPS in polled mode it is necessary to insure that no interrupts occur; the Disable All Interrupts can be used for this function. It should be noted that DMA can still pre-empt the 68010 when interrupts are disabled, so no DMA transfers should be pending when an "atomic" operation is desired. The Test and Set instruction will work on the Whopper. Finally, when the DMA MMU context is being modified, care should be taken to insure that DMA will not occur, as wild accesses may result.

Bus Timeouts

During all accesses a timer is set to cause a 68010 BUS ERROR cycle in the event there is no response to a bus cycle. Note that if the MMU/Memory board is installed a BUS ERROR cycle can also be generated if a MMU error occurs. The BUS ERROR software can distinguish between a BUS TIMEOUT and an MMU ERROR by reading the

MMU ERROR bit in the SYSTEM STATUS register.

Reliability Goals

A final design goal for the Whopper is that the system be reliable. While it is difficult to quantify reliability the following is proposed:

> The Whopper PC boards will each be tested to demonstrate that they meet the MTBF predicted by performing a stress level MTBF prediction.

Cooperation between the QA department and the engineering department will be required to insure that this goal is met.

Environmental Specifications

The Whopper system will be designed to meet the following environmental specifications:

1 - Operating Temperature: 10 - 40 Degrees Centigrade
2 - Operating Humidity: Less than 90% R. H. non-condensing
3 - Vibration: Shall withstand 1.5 g vibration in shipping container
4 - Shock: Inside shipping carton shock shall not damage HDA

Agency Approvals

The Whopper shall be approved by all applicable agencies, that is:

Safety: UL, CSA, tested to comply with IEC-380
RFI: FCC, Class A and VDE, Class A

Connectors

The Whopper provides the following connectors:

1 - Keyboard Connector (front of unit, 1/4" phone plug)
2 - Power Connector (back of unit, standard motor attachment cap)
3 - Serial Connectors (back of unit, two DB-25 connectors)
4 - Mouse Connector (back of unit, special DB-9 connector)
5 - Video Output Connector (back of unit, RCA jack)
6 - Expansion Connectors (back of unit, three 56 pin ZIF connectors)

In addition to the six types of connectors, one or two holes are provided to permit a card in Expansion Slot 4 to have external connections. The pinout of the Serial Port and Mouse connector is as follows:

Pin	Mouse	Serial A	Serial B
1	SW 1	Gnd	Gnd
2	+5 V	TxD A	TxD B
3	Gnd	RxD A	RxD B
4	L	RTS A	-
5	R	CTS A	-
6	SW 2	DSR A	DSR B
7	SW 0	Gnd	Gnd
8	DN	DCD A	-
9	UP	-	-
10		-	-
11		-	-
12		-	-
13		-	-
14		-	-
15		TxC A	-
16		-	-
17		RxC A	-
18		-	-
19		-	RxD B+
20		DTR A	DTR B
21		-	-
22		-	-
23		-	-
24		Text A	-
25		-	-

Appendices

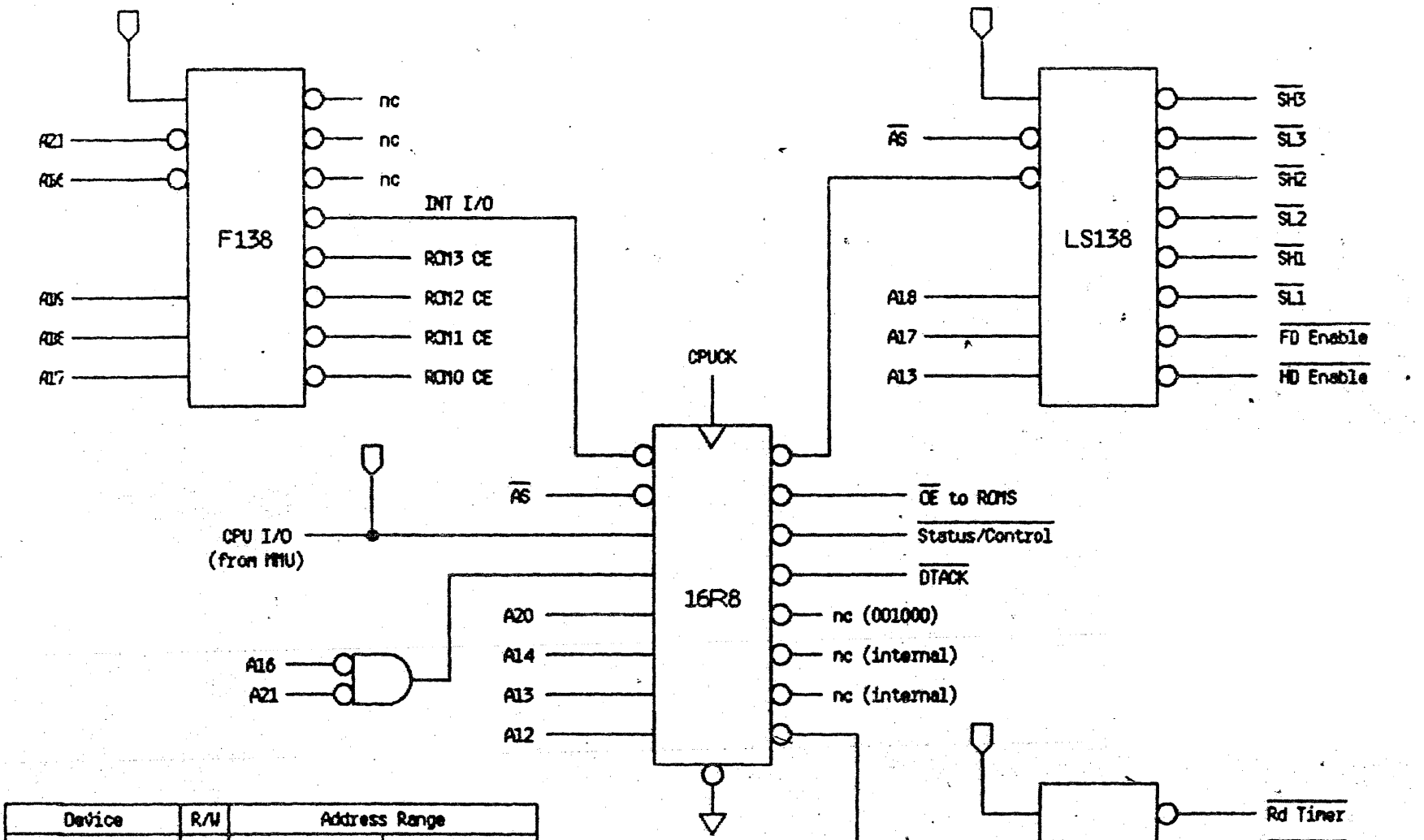
Attached to this ERS are data sheets containing information on the following devices:

1 - IWM chip

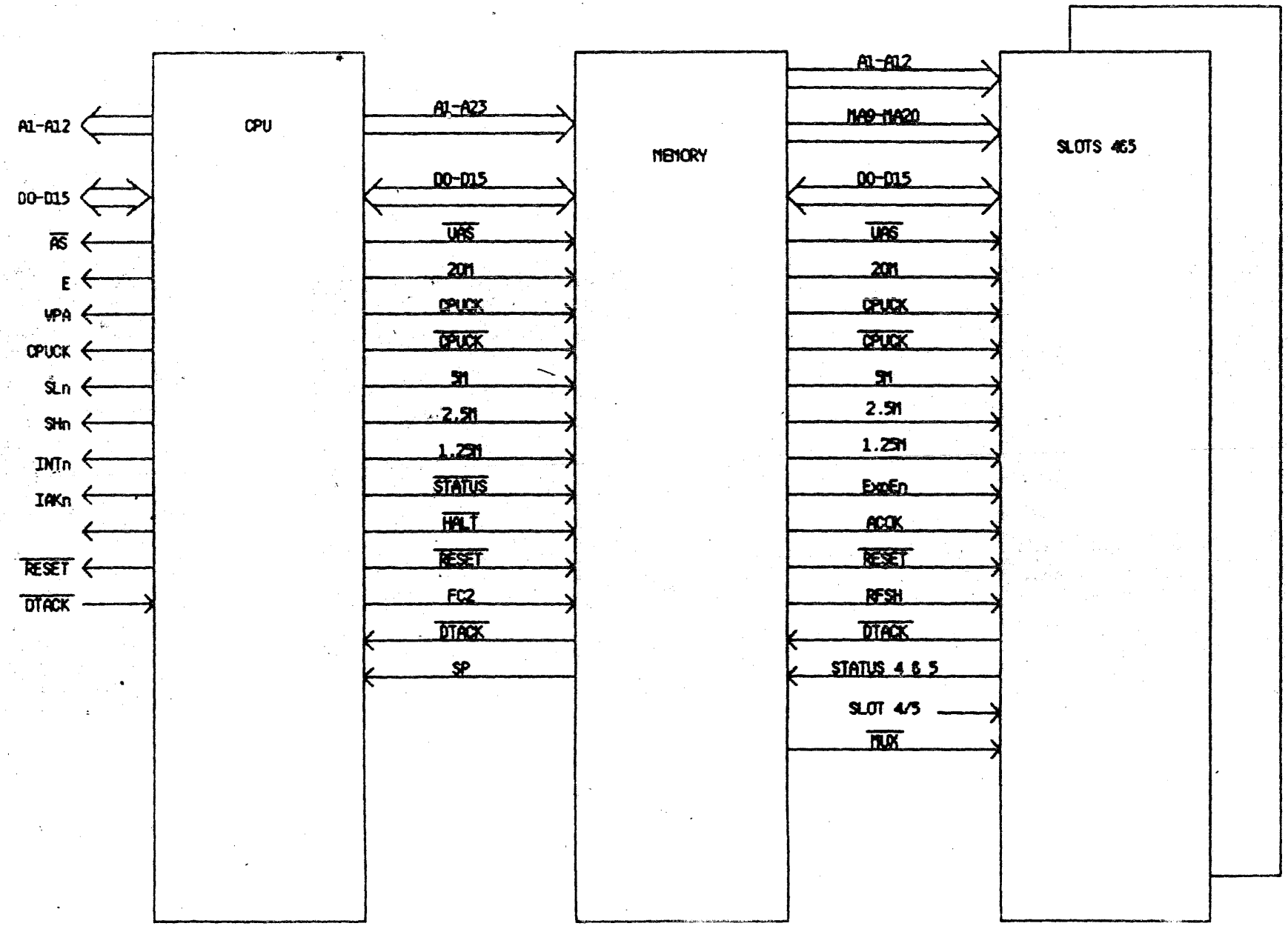
2 - SCC chip

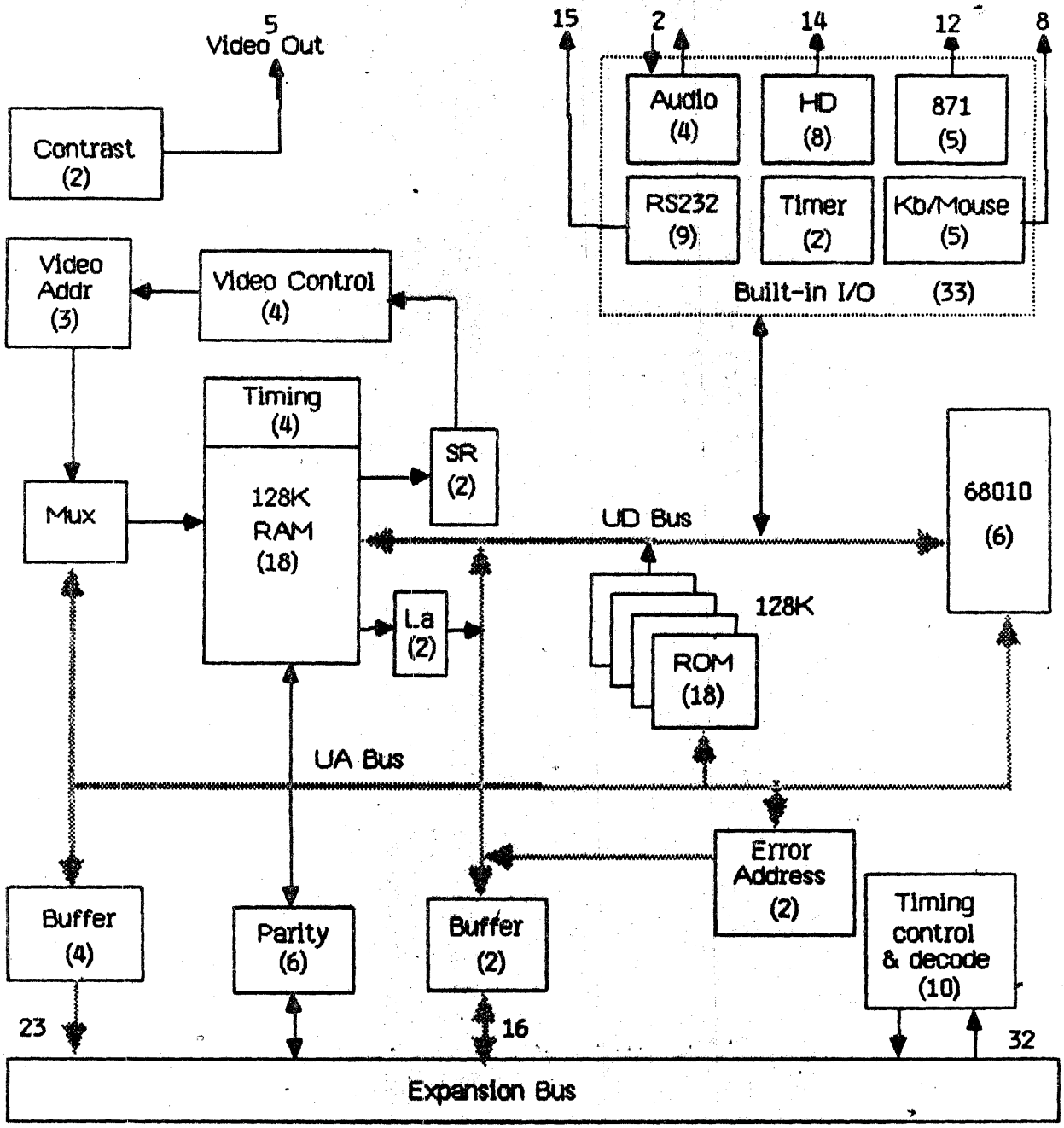
3 - 8253 Timer

4 - 3.5 Inch Mac Sony Disk Drive Interface Description



Device	R/W	Address Range	
Hard Disk	R/W	000000	000FFF
(not defined)	R/W	001000	001FFF
Floppy Disk	R/W	002000	002FFF
Status/Control	R/W	003000	003FFF
Timer	R/W	004000	004FFF
COPS	R/W	005000	005FFF
SCC	R/W	006000	006FFF
Error Addr Latch	RO	007000	007FFF
Video/Audio level	WO	007000	007FFF


CPU
MEMORY
SLOTS 4&5
A1-A12
DO-D15
AS
E
VPA
CPUCK
SLn
SHn
INTn
IAKn
RESET
DTACK
A1-A23
DO-D15
UAS
20M
CPUCK
CPUCK
5M
2.5M
1.25M
STATUS
HALT
RESET
FC2
DTACK
SP
A1-A12
MA9-MA20
DO-D15
UAS
20M
CPUCK
CPUCK
5M
2.5M
1.25M
ExpEn
ACOK
RESET
RFSH
DTACK
STATUS 4 & 5
SLOT 4/5
MUX



LISA 1.75 CPU BOARD