Apple Lisa Computer
Technical Information

# Apple Lisa Computer:

# Hardware Manual

# 1981

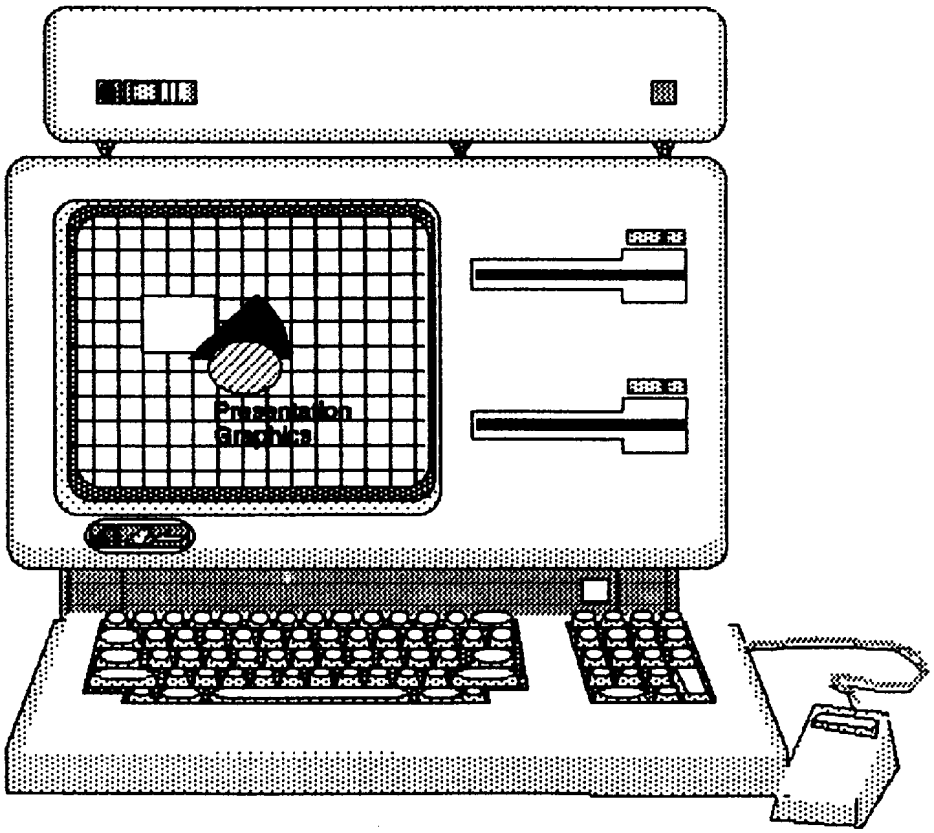
Lisa Computer:
1983 - 1985

Printed by: Macintosh Picture Printer 0.0.5 1999-01-11 Printed: 1999-02-22 12:16:37

Apple Lisa Personal Computer

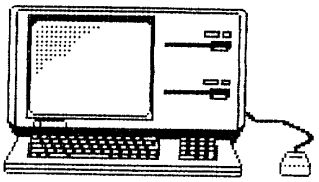
1983 to 1985

# Hardware Manual (Jul 81)




"LisaHWG81_000C1.PICT" 181 KB 1999-02-20 dpi: 72h x 72v pix: 2146h x 2716v

[redacted] [redacted]



LISA

# Hardware

REFERENCE MANUAL

Apple Lisa Computer
1983 - 1985

Larry Birnbaum

{7-July-1981}

408-996-1010

~~Bill Schottstaedt~~
~~Ext. 2079~~

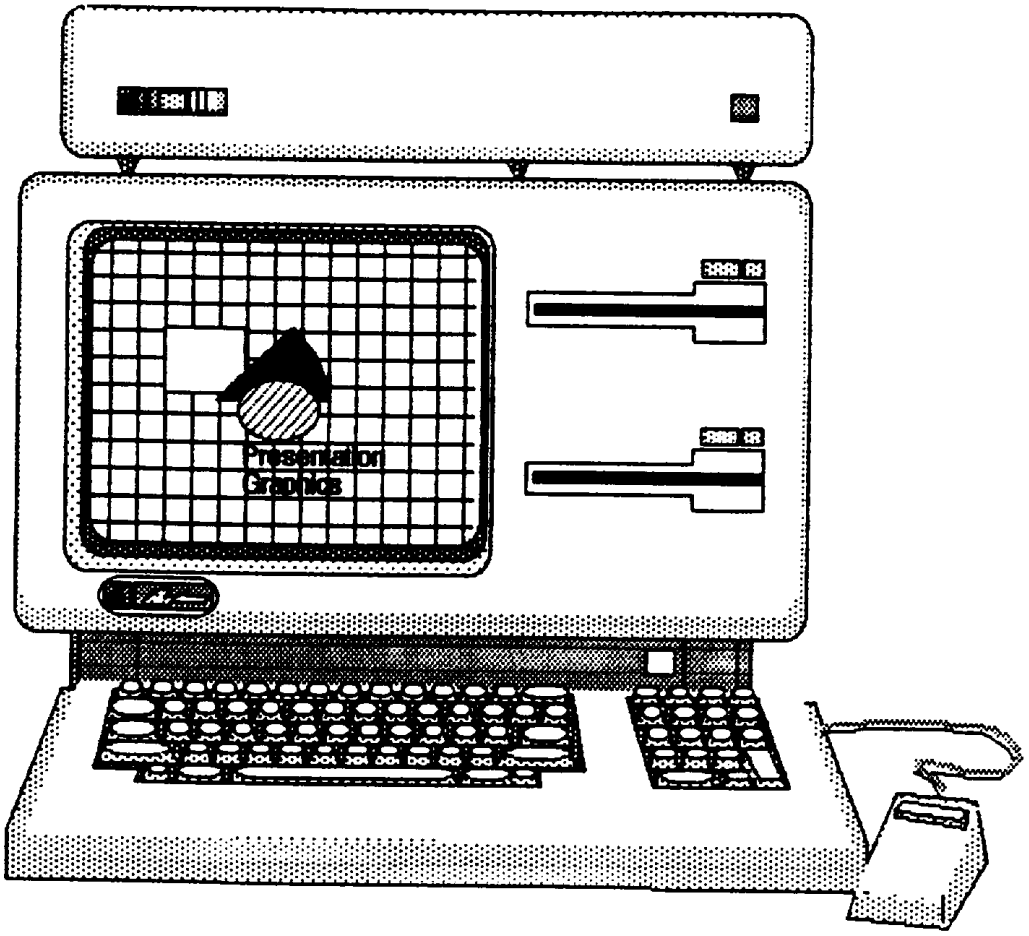
Questions: Paul Baker
X2182

What size are expansion cards? 9.3 x 4.7 5.3 (connector)

NOTE: DO NOT CONTACT THESE PEOPLE AT APPLE!

(CONTAINS 81 PAGES)

[p. 1]




# Apple Lisa Personal Computer
# 1983 - 1985

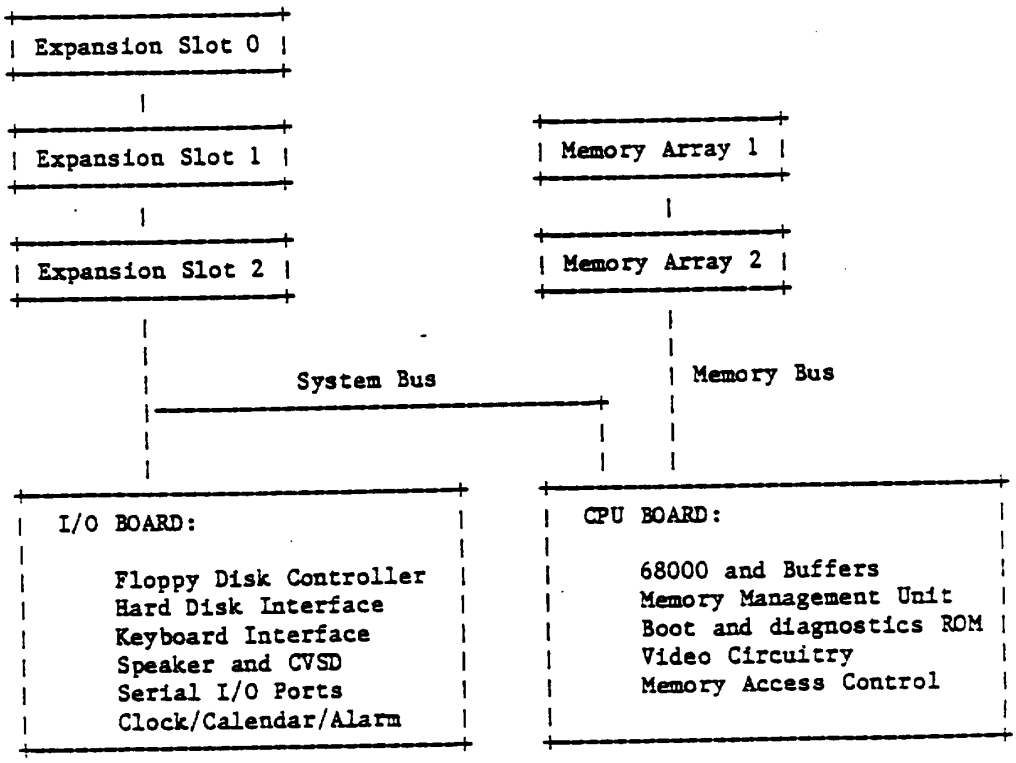
1 TITLE PAGE

2 TABLE OF CONTENTS i-ii

## HARDWARE OVERVIEW

The Lisa hardware can be divided into four functional pieces: the CPU and memory management unit, the I/O circuitry, the memory arrays, and the expansion slots. The main microprocessor is a Motorola MC68000. It communicates with the rest of the Lisa world over two buses: the system bus, and the memory bus. The I/O circuitry includes the built-in parallel interface, two serial interfaces, floppy disk controller, clock-calendar, speaker, and keyboard interface.

SYSTEM BLOCK DIAGRAM




## THE PROCESSOR

The Motorola MC68000 Microprocessor User's Manual fully describes the operation of the processor, including its instruction set and timing. The Lisa 68000 has a 5 megahertz clock. The memory cycle time is 800 nanoseconds; therefore instructions that use a multiple of four clock cycles can execute without a wait state.

## VIDEO DISPLAY

The bit mapped display has 360 dots vertically and 720 dots horizontally. It is refreshed from system RAM. Since the video circuit cannot wait for a long bus cycle to complete, the Memory Bus provides a direct path from system RAM to the video refresh circuit. The CPU board synchronizes CPU access to system RAM with that of the video circuit.

## MEMORY

The Lisa memory management unit addresses 16 megabytes of memory. Hardware on the CPU board divides this address space into 128 segments, each with 128 KBytes. This hardware, called the Memory Management Unit (MMU), takes a logical address generated by the 68000 and transforms it into a physical address indicating a position in RAM or I/O. The transformation involves relocation; therefore the physical location of a segment in memory is unimportant to a program. During the transformation, the MMU performs access checks so that the segment is used only for its intended purpose. The access checks help insure that a runaway process does not destroy too much data before being caught.

There are three physical address spaces: main memory, I/O, and special I/O. The main memory can contain up to two megabytes of RAM. The I/O memory addresses all system I/O devices and expansion slots. The special I/O address space is reserved for the Boot ROM and MMU registers. It is used only when starting the machine and when modifying the contents of the MMU registers.

### SYSTEM MEMORY MAP

| Address Range | Address Space | Function |
|---|---|---|
| $000000 - $1FFFFF | Main Memory | System Main Memory (RAM) |
| $000000 - $001FFF | I/O | Slot 0 Low Decode |
| $002000 - $003FFF | I/O | Slot 0 High Decode |
| $004000 - $005FFF | I/O | Slot 1 Low Decode |
| $006000 - $007FFF | I/O | Slot 1 High Decode |
| $008000 - $009FFF | I/O | Slot 2 Low Decode |
| $00A000 - $00BFFF | I/O | Slot 2 High Decode |
| $xx0000 - $xx3FFF | Special I/O | Boot and self-test ROM |
| $00C000 - $00CFFF | I/O | Floppy disk controller shared memory |
| $00D000 - $00DFFF | I/O | I/O Board Devices |
| $00E000 - $00FFFF | I/O | CPU Board Devices |

## THE SYSTEM BUS

The system bus connects the CPU board, I/O board, and expansion slots, providing a simple interface between the various devices. When an address is put on the bus, a signal indicates the start of the cycle. The system then waits for the addressed device to respond with the acknowledge signal which marks the end of the cycle. Timing on the system bus is non-critical. The bus cycle can be as short as 800 nanoseconds or as long as 50 microseconds. If the addressed device does not respond within 50 microseconds, a timer on the bus asserts the bus error signal. A detailed description of the System Bus timing is given in the section on the expansion I/O slots.

The following pages describe in detail the CPU board and memory management unit, the various built-in I/O devices and interfaces, and the expansion slots. A discussion of the design of device drivers can be found in the Operating System Reference Manual. This manual is intended for programmers who deal directly with the hardware--it is not a service manual, nor does it provide a complete theory of operation. Complete schematics are included, however, in the appendices.

## THE 68000

The 68000 is a 16 bit microprocessor with eight 32 bit data registers, seven 32 bit address registers, a 32 bit program counter, and two 32 bit stack pointers. Its instruction set directly supports a wide variety of data types and addressing modes. Its 24 bit address bus provides a memory addressing range of 16 megabytes. The 68000 used in the Lisa computer has a CPU clock period of 200 nanoseconds. Since the memory cycle time is 800 nanoseconds, and since the processor cannot access memory during video refresh cycles, enough wait states are inserted to bring each instruction execution time to a multiple of four cycles.

## INTERRUPTS

The 68000 has a flexible interrupt handling structure. Interrupts are generated by system errors, program errors, and external devices. When the interrupt is detected, the 68000 enters the supervisor state and begins executing instructions at the address provided by the exception vector table. The source of the exception can provide the index into the vector table, or the interrupt level can be used to provide the exception vector ('autovectoring'). Internal Lisa devices all use autovectors. External devices can use either option.

The interrupt system has seven fixed priority interrupts.

FIXED PRIORITY INTERRUPTS

| Level | Type | |
|---|---|---|
| 7 | Non-Maskable Interrupt | (highest priority) |
| 6 | RS-232 ports | |
| 5 | Expansion slot 0 | |
| 4 | Expansion slot 1 | |
| 3 | Expansion slot 2 | |
| 2 | Keyboard Interrupt | |
| 1 | All other internal interrupts | (lowest priority) |

## EXCEPTION VECTOR ASSIGNMENT

| Exception | Vector Address |
|---|---|
| Reset: Inital SSP | $000000 |
| Reset: Initial PC | $000004 |
| Bus Error | $000008 |
| Address Error | $00000C |
| Illegal Instruction | $000010 |
| Zero Divide | $000014 |
| CHK Instruction | $000018 |
| TRAPV Instruction | $00001C |
| Privilege Violation | $000020 |
| Trace | $000024 |
| Unimplemented Instruction 1010 | $000028 |
| Unimplemented Instruction 1111 | $00002C |
| Reserved, unassigned | $000030 - $00005F |
| Spurious Interrupt | $000060 |
| Other Internal Interrupt | $000064 |
| Keyboard Interrupt | $000068 |
| Slot 2 Autovector | $00006C |
| Slot 1 Autovector | $000070 |
| Slot 0 Autovector | $000074 |
| RS232 Interrupt | $000078 |
| Non-Maskable Interrupt | $00007C |
| TRAP Instruction Vectors | $000080 - $0000BF |
| Reserved, unassigned | $0000C0 - $0000FF |
| User Interrupt Vectors | $000100 - $0003FF |

These addresses are logical addresses. The operating system stores the vectors in segment 0, context 0 in main memory.

Four sources can generate a Non-Maskable Interrupt: power fail, hard or soft memory error, and keyboard reset. Any key on the keyboard can be programmed to cause the keyboard reset.

The floppy disk controller, the hard disk interface, and the video circuitry can generate level 1 interrupts.

Although the power on reset vector is a part of the exception vector table given above, it is actually stored in ROM so that the vectors are available at power on.

## MEMORY

All memory is divided into three parts: main system memory (RAM), I/O memory, and special I/O memory. Main memory can contain up to two megabytes of RAM. I/O memory is used by all system I/O devices. Special I/O memory is used to boot the machine and to run the self-diagnostic tests. The Memory Management Unit transforms addresses generated by the 68000 into actual memory locations.

## MEMORY CONFIGURATIONS

The memory boards contain four rows of RAM chips with 18 chips in each row. This configuration provides byte parity memory. Since the chips can be either 16K or 64K RAMs, the board capacity is either 128K or 512K bytes. The boards can be jumpered to reduce total memory capacity by stuffing only one or two rows. A partially stuffed memory board containing 64K RAMs can yield either 128K or 256K bytes.

The memory boards contain the address recognition circuitry to make them self configuring. An n byte board automatically occupies the next n bytes in the main memory address space. Memory capacity can therefore be increased merely by plugging in another board.

POSSIBLE MEMORY CONFIGURATIONS

| Configuration Number | Board 1 Chips | Board 2 Chips | Board 1 Size | Board 2 Size | Total Size |
|---|---|---|---|---|---|
| 1 | 16K | — | 128K | | 128K |
| 2 | 16K | 16K | 128K | 128K | 256K |
| 2 | 64K | — | 256K | | 256K |
| 3 | 64K | — | 512K | | 512K |
| 4 | 16K | 64K | 128K | 512K | 640K |
| 4 | 64K | 16K | 512K | 128K | 640K |
| 5 | 64K | 64K | 512K | 256K | 768K |
| 5 | 64K | 64K | 256K | 512K | 768K |
| 6 | 64K | 64K | 512K | 512K | 1024K |

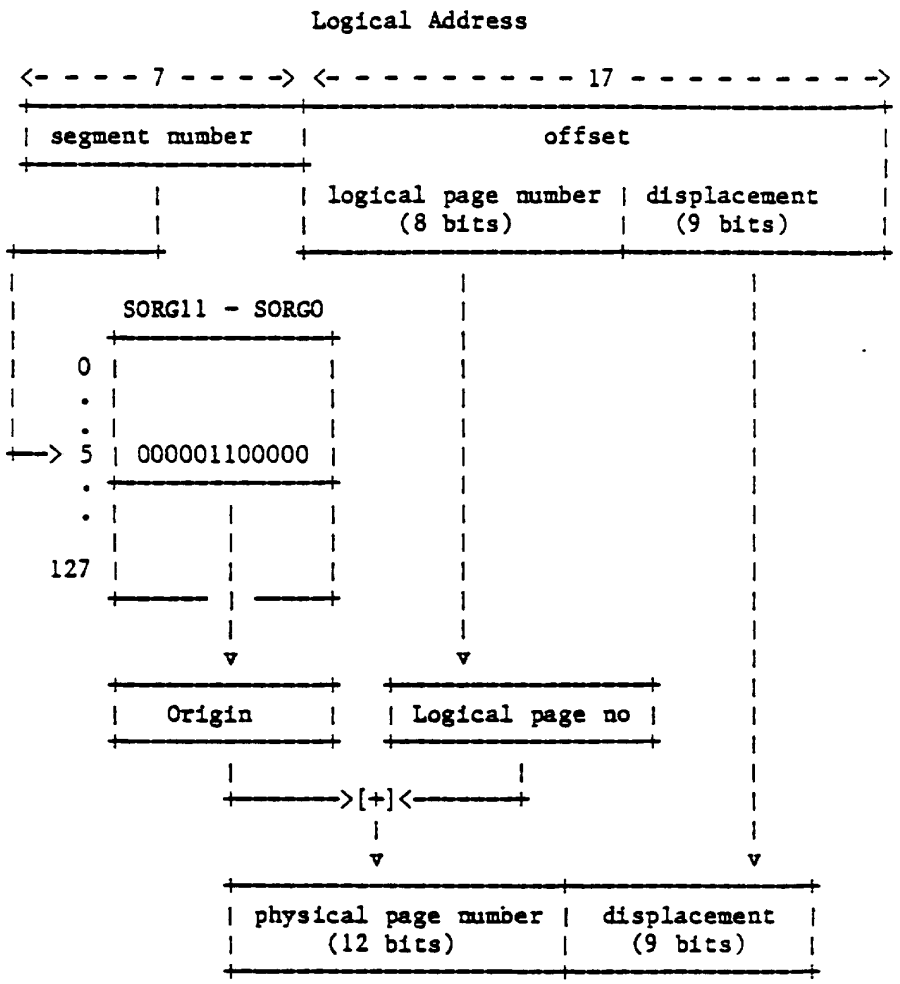
Since only four rows of RAM fit on a board, the maximum memory size is one megabyte.

## THE MEMORY MANAGEMENT UNIT

The Memory Management Unit (MMU) helps the Operating System relocate objects in memory. It also provides memory access controls to prevent a process from accessing memory which is not assigned to it. The mechanism used is segmentation.

The 68000 outputs 24 bits of address information. The seven most significant bits (A23 - A17) select one of the 128 segments. The remaining 17 bits are the offset into the segment. Each segment has two associated registers which describe the origin (SORG) and size (SLIM) of the segment. In addition, each segment is divided into 512 byte pages. The most significant 8 bits of the segment offset determine the segment page number, and the remaining 9 bits are the page offset.

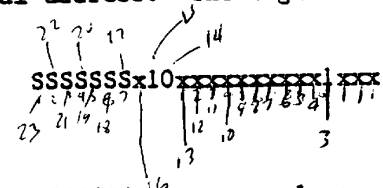
ADDRESS TRANSLATION




Physical Address
(21 bits -- 2 megabytes)

## THE SEGMENT ORIGIN

The Segment Origin Register (SORG) contains the high 12 bits of the address added to the segment offset produced by the 68000 to get the physical address. The register can be read or written. Its address is



SSSSSSSx10xxxxxxxxxxxxxxx (in Special I/O space)

where

| | |
|---|---|
| SSSSSSS | selects the segment |
| x | ignored |

Bits 0 to 11 of the register set the segment base address for segment SSSSSSS. The other 4 bits in the register are not used. For example, a segment whose origin is $000200 has SORG11 - SORG0 set to $001. The address is in the currently selected context. The SEG1 and SEG2 bits determine which context is selected. An MMU setup program should run in supervisor mode.

## CONTEXT SELECTION

The context bits mentioned above permit fast context switching. The MMU has four sets of MMU registers which can be assigned to four different processes. Once these contexts have been loaded, the context bits can be used to control which context is active. Contexts 1, 2, and 3 are general purpose contexts, but context 0 is intended for the use of the Operating System. Whenever the 68000 does an access in supervisor mode, the context is automatically switched to context 0. When the context selection bits are actually set, however, the fact that you are necessarily in supervisor mode is ignored and the desired context is selected. The TRAP instruction, which puts the processor in supervisor mode, can be used to generate a call from a user process to the Operating System.

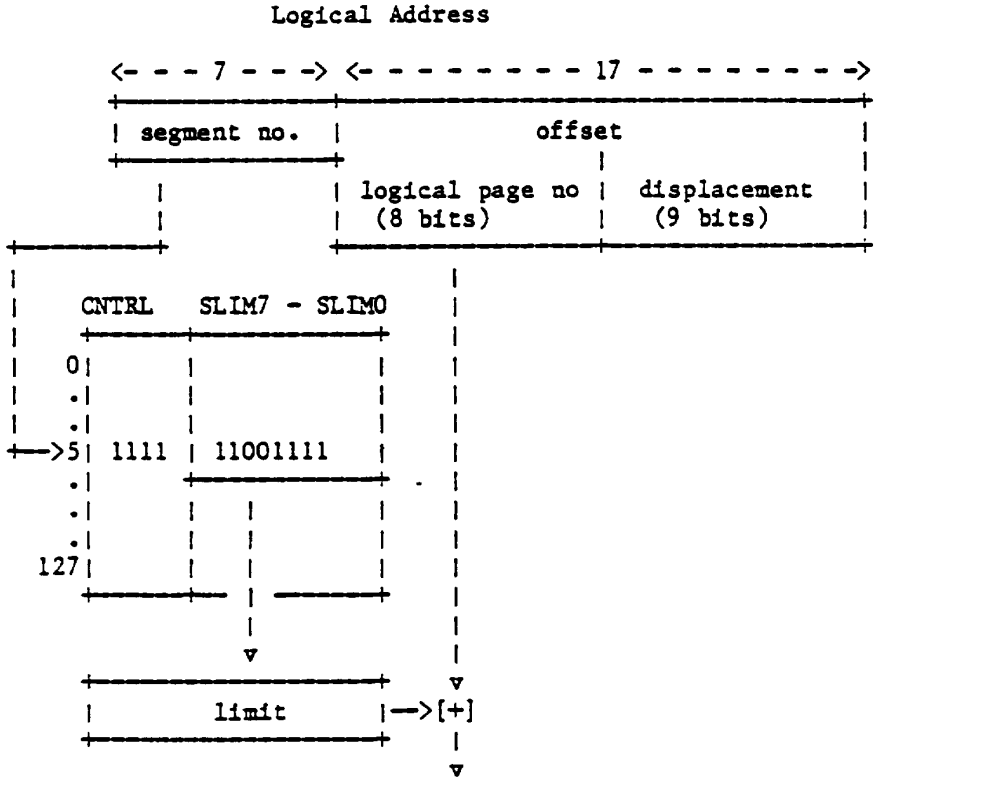
The two bits SEG1 and SEG2 control which context in the MMU is selected. SEG1 is set by a read or write at the address $00E00A (in I/O space), and reset by a read or write at $00E008. SEG2 is set by a read or write at $00E00E and reset at $00E00C. The context selected is determined as follows:

| SEG2 | SEG1 | Context |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |

THE SEGMENT LIMIT

When the MMU receives an address, the address is first relocated and an access to main memory is started at the addressed location. The MMU then checks to see that the relocated address is actually within the limits of the segment being accessed.

LIMIT CHECKING




limit failure if carry out = 1 - normal segment
limit failure if carry out = 0 - stack segment

The Segment Limit Register tells the MMU the segment limit and controls which address space is accessed. If the address is within the size limit of the segment being accessed and the access is to main memory, the cycle is completed. If the address is beyond the segment limit a bus error is generated. If the access required is actually to I/O space, the main memory cycle is terminated and the I/O cycle is performed. The four control bits in the limit check register tell the MMU which address space is being accessed.

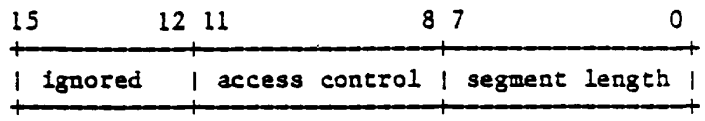
The Limit Register of a particular segment is found at the address:

SSSSSSSx10xxxxxxxxxx0xxx (in Special I/O space)

where

| | |
|---|---|
| SSSSSSS | selects the segment |
| x | ignored |

The limit register is divided into three regions:




Bits 0 to 7 of the limit register set the length of the segment. The value in these bits is interpreted as the segment length in two's complement form. A length of 0 (SLIM7 - SLIM0 = $00) indicates that any access is allowed. A length of 255 (SLIM7 - SLIM0 = $FF) indicates that the segment contains 512 bytes.

Bits 8 to 11 of the limit register are the access control bits.

ACCESS CONTROL BITS

| CNTRL 11 10 9 8 | | Address Space and Access |
|---|---|---|
| 0 1 0 0 | 4 | Main memory -- Read only stack |
| 0 1 0 1 | 5 | Main memory -- Read only |
| 0 1 1 0 | 6 | Main memory -- Read/Write stack |
| 0 1 1 1 | 7 | Main memory -- Read/Write |
| 1 0 0 0 | 8 | I/O space |
| 1 1 0 0 | C | Page invalid (segment not present) |
| 1 1 1 1 | F | Special I/O space |
| any other | | Not allowed (unpredictable results) |

THE PROCESS STACK

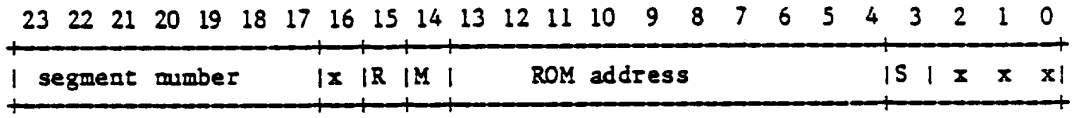
Before the 68000 pushes data on the stack, the value of the stack pointer is decremented. The top of the stack is actually at a lower address than the bottom of the stack. To permit a segment to start at the segment top and grow downward, one of the control bits (8) is used to reverse the map's limit check. It is assumed that the stack pointer is initialized to all ones + 1 in the segment. That is, it "starts" at the bottom (0-th word) of the neighboring segment. Since its value is first decremented, the first actual location accessed is the top word of the segment.

## LOADING THE MMU REGISTERS

To set up the MMU, the Operating System must initialize the SORG and SLIM registers of each of the 128 segments. To set these registers, the memory map must be disabled.

The MMU can be disabled by setting the SETUP bit. SETUP is set at power on when the map registers are uninitialized, and whenever the map registers are being loaded. When the memory map is disabled, the Special I/O address space is enabled, and only the MMU registers and boot ROM can be acessed. In this state the system hardware decodes addresses from the 68000 in the following manner:

SPECIAL I/O ADDRESS DECODE




Bit 16 is ignored. Bit 14 (M) turns the memory map on (1) or off (0). If the map is off, bit 15 (R) selects either the system ROM (0), or an MMU register (1). If the ROM is selected, the low 14 bits are the address in ROM. If the MMU is selected, the high order bits are the segment number of the register to be accessed and bit 3 determines whether the Segment Origin register (0), or the Segment Limit register (1) is accessed. Bits 0 - 2 are unused.

When the system is powered on, the SETUP bit is set. The power on vectors at address 0 are fetched from ROM and can branch to the boot program in the ROM. The boot program can be addressed by any segment number. It is recommended that segment 127 be programmed to decode special I/O and that the boot program jump to segment 127 when it begins execution. The boot program can initialize the MMU registers so that context 0 addresses RAM with the low 16 segments and system I/O with segment 126. Since this makes all the address space in the system accessible, the system can act as though no memory map were present. The boot program can then turn off the SETUP bit, but the program can continue to run from ROM because segment 127 has been programmed to special I/O. The Operating System can then be loaded into RAM and begin normal execution.

When the Operating System wants to load the MMU registers, the program it uses must be located at a logical address that has bit 14 set. If bit 14 is set and the program is in supervisor mode with the SEG bits set correctly, the SETUP bit can be set and the program continues to execute out of RAM as long as the address emitted has bit 14 set. When the actual write to the MMU register occurs, bit 14 must be 0 and bit 15 must be set. Once the MMU has been set up, the SETUP bit must be cleared to return to normal operation. The SETUP bit can be set by a read or write at the address $00E010 in I/O space, and reset by a read or write at $00E012.

MEMORY ERRORS

Two kinds of memory error can be detected during a memory access, Hard Error and Soft Error. A Hard Error is a parity error during a parity memory board access or an uncorrectable error during an ECC memory board access. A Soft Error indicates that a correctable error has been detected on an ECC memory board access. If either the Hard Error (HDER) or Soft Error (SFER) signal is asserted at the end of a memory cycle, a latch on the CPU board latches the address at which the error occurred, and asserts the NMI signal which interrupts the 68000. The address latches latch the high order 16 bits (segment number and page number) of the address. The cause of the error is latched in the system status register. A program can therefore log soft errors in ECC boards and keep track of hard failures in main memory on a page by page basis, and the MMU can be used to map out bad pages.

Detection of either hard or soft memory errors can be disabled. Each error type has an Enable Memory Error Detect Bit. If the bit is set, errors are detected and latched. If the bit is reset, error detection is masked. At power on, both kinds of memory error detection are masked off. To set soft error detection, read or write to the address $00E016 in I/O space. To reset soft error detection, read or write to $00E014. Hard error detection is enabled by a read or write to $00E01E, and disabled by a read or write to $00E01C.

## THE STATUS REGISTER

The status register permits a program to read the status of the system bus. A Bus Error exception routine can read this register to determine the cause of the Bus Error. The register is located at $00F800 in I/O space. It can only be read. Its bits have the following meanings:

STATUS REGISTER BITS

| Bit | Name | Meaning |
|---|---|---|
| 0 | Soft Error | A soft error has occurred (bit=0) |
| 1 | Hard Error | A hard error has occurred (bit=0) |
| 2 | Vertical Retrace | The video circuit has begun a vertical retrace if this bit is a 0. When this bit is set, an interrupt is generated. The interrupt routine should reset this bit just before it returns because the video circuit continues to set this bit for two scan lines after the start of the retrace (about 90 uS). The Enable Vertical Retrace Interrupt bit can be set to mask the interrupt and prevent this bit from being set. |
| 3 | Bus Timeout | A bus timeout has occurred (bit=0). A timer insures that the 68000 does not hold Address Strobe (AS/) low longer than 50 uS to keep the system from being hung by an attempt to access a nonexistent peripheral device. Whenever the timer detects that AS/ has been low for more than 50 uS, a Bus Error is generated and the Bus Timeout bit is set in the system status register. |
| 4 | Video Bit | This bit permits the CPU to read the output of the video circuit for diagnostic purposes. If the Video Bit is 1, the video output is a bright spot on the screen. |
| 5 | Horizontal Sync | This bit permits the CPU to read the state of the horizontal sync signal for diagnostic purposes. When the bit is 1, the display is in a horizontal retrace. |

| | | |
|---|---|---|
| 6 | Serial Number Data | The serial number is stored in the video state PROM as a 56 bit number. It can be read one bit at a time at this location. The serial number is delivered one bit every four CPU clock cycles, so a program that does consecutive memory cycles at the maximum rate can read one bit per cycle. |
| 7 | Serial Number Sync | The serial number read out into bit 6 is synchronized to start when the Serial Number Sync bit is 1. The following 224 CPU clock cycles shift out all 56 bits of the serial number. |
| 8-15 | | Unused |

THE MEMORY ERROR ADDRESS LATCH

The memory error address latch can be read to determine the address at which a detected hard or soft error occurred. If the status register's hard or soft error bits are not set, the address latch contains an indeterminate value. Once the error logging program has read the latch, the latch is reset to permit detection of subsequent errors. At the same time the soft error bits, the bus timeout bit, and the vertical retrace bit are reset. The address latch reset signal also gates the latched address data onto the data bus, so all the status bits are reset when the error latch is read. The latched address is a physical address. The error address latch is located at $00F000 in I/O space. To reset the hard error bit, use the hard error mask bit.

MEMORY DIAGNOSTIC MODE

Two flip-flops are provided to help test the error detection and correction circuitry. DIAG1 asserts the Soft Error line, and DIAG2 asserts the Hard Error line. DIAG1 is set by a read or write to $00E002 in I/O space, and reset by a read or write to $00E000. DIAG2 is set by a read or write to $00E006 and reset by a read or write to $00E004.

The test function performed depends on the type of memory board in use. On a parity memory board, the parity detection circuitry can be tested by the following steps:

1. Assert the Hard Error line (set DIAG2)
2. Write to memory
3. Clear DIAG2
4. Enable Hard Error

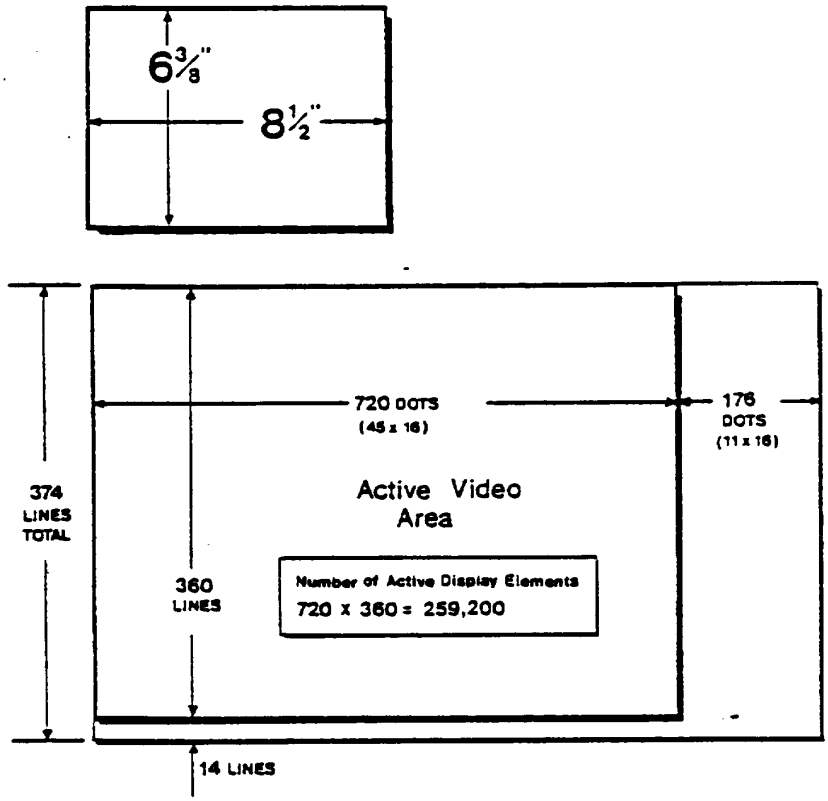
At this point a read from the memory board should generate a parity error. Since the Hard and Soft error status latches are set in this mode, the Hard and Soft error masks should also be set.

VIDEO CIRCUIT

The display is 6.325 inches high and 8.5 inches wide. There are a total of 374 horizontal lines of which 360 are active. 720 of the 886 vertical lines are active. There are, therefore, 259,200 active elements in the full display.

The dot clock frequency is 20.0 MHz, so the dot clock time is 50 nanoseconds. The active line time (the dot clock time times 720) is 36 microseconds, and the inactive line time (dot clock times 176) is 8.8 microseconds. The line frequency is 22,321.429 Hz. The frame frequency is 59.68 Hz. The screen aspect ratio (width/height) is 4/3. The dot aspect ratio is 2/3.

THE VIDEO DISPLAY



The video circuit fetches a 16 bit word from main memory once every four CPU clock cycles, and shifts it out to the CRT to provide a bit mapped display. The data in the bit map is displayed as an image of the bits in memory. A one corresponds to a dark (off) dot, and a zero corresponds to a bright (on) dot. Wait states are inserted into each memory fetch so that it takes a multiple of four cycles.

A periodic interrupt, the vertical retrace interrupt, is generated at the start of each vertical retrace. The display is refreshed at 60 Hz. A state machine in the video circuit counts the number of words displayed and generates the horizontal and vertical sync pulses. The ROM which controls this state machine also contains the system serial number.

The vertical retrace interrupt can be disabled. A write to the address $00E01A in I/O space enables the interrupt and a write to $00E018 disables the interrupt. Since the Reset line resets this bit, the vertical retrace interrupt is masked off at power on.

The video circuit controls the low order 15 bits of the video address to refresh the display. The high order six bits remain fixed. The video address latch at $00E800 in the I/O address space provides the high order six bits. By writing to this register the programmer can position the video memory anywhere in main memory. At power on the register is in an indeterminate state.

| D15 - D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|
| unused | VA20 | VA19 | VA18 | VA17 | VA16 | VA15 |

CONTRAST CONTROL

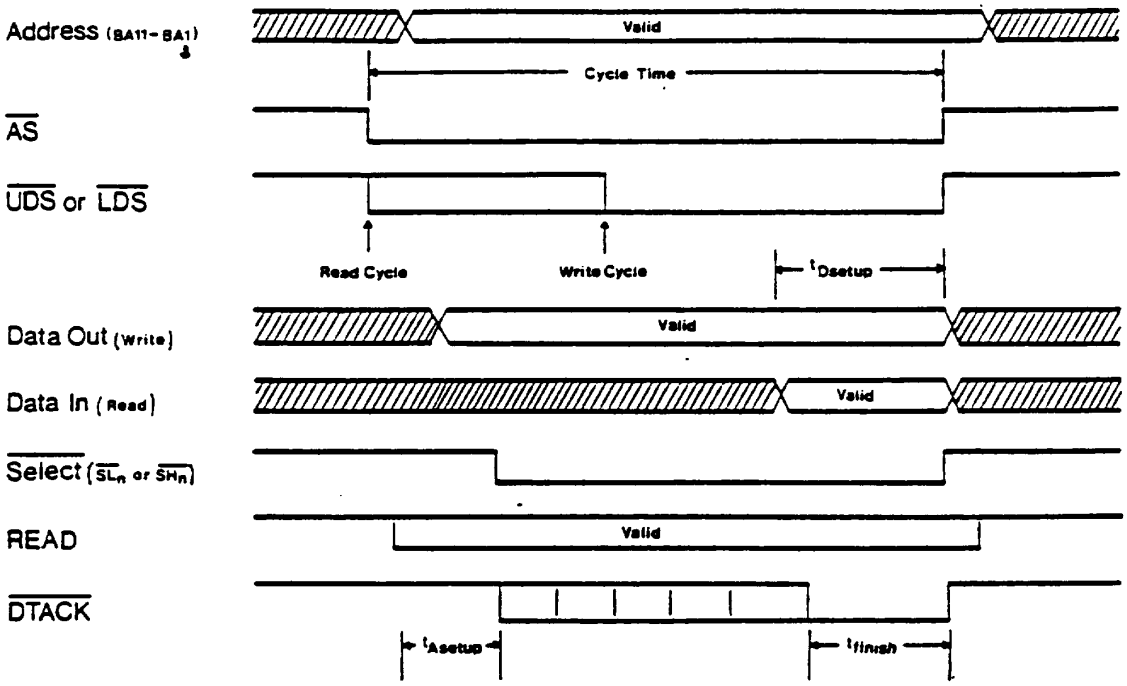
The contrast control circuitry is located on the I/O board. A digital to analog converter controls the CRT display contrast. This converter is controlled by a latch connected to the Slow Data bus (the hard disk interface 6522 Peripheral A Port). To write a value to the contrast latch:

1. Set the DEN bit (PB2) to 1.
2. Put the desired contrast value ($00 to $FF) on the Peripheral A Port of the hard disk 6522.
3. Execute MOVE.L D0,$00D01C. This instruction causes a pulse on the contrast latch clock input.

The address $00D01C in the I/O address space clears the clock input to the contrast latch. The address $00D01E sets the clock input, so the MOVE instruction given above insures that there is a positive edge on the clock input.

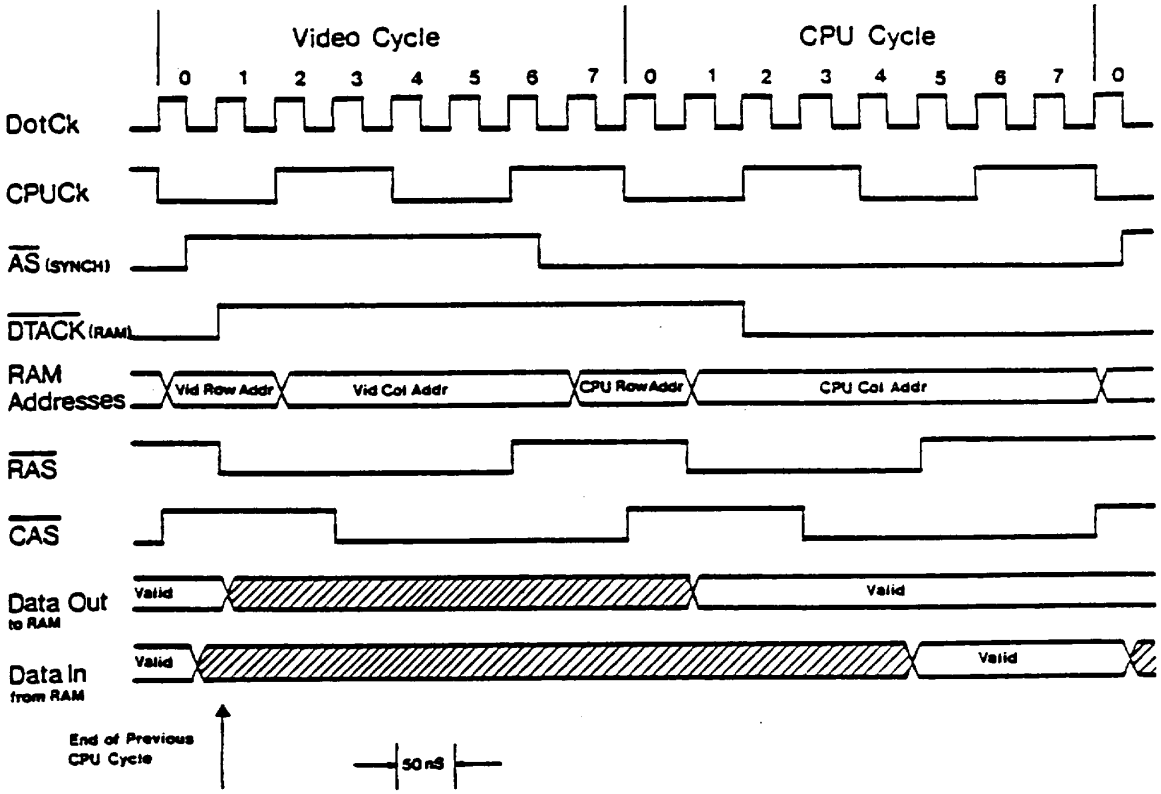
When the system is reset either by the RESET instruction, or by a power-on, the contrast latch is reset and the screen is cleared. The higher the number stored in the contrast latch, the higher the display contrast.

SYSTEM BUS TIMING




| | Min | Max |
|---|---|---|
| Cycle Time | 800ns | 50us |
| TASetup | 100ns | — |
| TDSetup | 100ns | — |
| TFinish | 100ns | 260ns |

MEMORY BUS TIMING



## A SUMMARY OF I/O SPACE LOCATIONS

| | | |
|---|---|---|
| Status Register | | $00F800 |
| Memory Error Address Latch | | $00F000 |
| Memory Diagnostic Bits | | |
| DIAG1 | Set: | $00E002 |
| | Reset: | $00E000 |
| DIAG2 | Set: | $00E006 |
| | Reset: | $00E004 |
| Context Selection Bits | | |
| SEG1 | Set: | $00E00A |
| | Reset: | $00E008 |
| SEG2 | Set: | $00E00E |
| | Reset: | $00E00C |
| SETUP register | Set: | $00E010 |
| | Reset: | $00E012 |
| Enable Vertical Retrace Interrupt | | |
| | Set: | $00E01A |
| | Reset: | $00E018 |
| Enable Memory Error Detect | | |
| Soft Error Enable | | |
| | Set: | $00E016 |
| | Reset: | $00E014 |
| Hard Error Enable | | |
| | Set: | $00E01E |
| | Reset: | $00E01C |
| Video Address Latch | | $00E800 |

## THE INPUT/OUTPUT BOARD

The Input/Output board contains the circuitry for the I/O devices built into the Lisa. These devices are:

Keyboard/Clock interface

Two Serial interfaces

Floppy Disk Controller

PIPPIN Hard Disk Controller

Speaker and CVSD interface

The soft-on and soft-contrast registers are also on the I/O board. I/O devices are accessed through memory-mapped locations in I/O address space.

## THE KEYBOARD INTERFACE

The keyboard interface uses two Control Oriented Processor System (COPS) microprocessors. The COPS on the keyboard communicates key up and key down information to the 68000 over a bi-directional serial interface. The other COPS is on the I/O board. It connects the bi-directional keyboard signal to the 68000 by way of an eight bit parallel port implemented with a 6522 Versatile Interface Adapter.

Since the same circuitry which is connected to the power-on pushbutton, the power supply ON signal, and the power fail signal, also supplies the clock/calendar function, it is possible for software to cause the system to turn itself off and on.

The 6522 has two eight bit parallel I/O ports (Peripheral A Port, or PA0-PA7, and Peripheral B Port, or PB0-PB7). The control lines for these ports are CA1, CA2, CB1, and CB2. The bit assignment is as follows:

6522 Peripheral A Port

| CD7 | CD6 | CD5 | CD4 | CD3 | CD2 | CD1 | CD0 | Lisa Name |
|---|---|---|---|---|---|---|---|---|
| PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 | Synertek Name |

CD0 - CD7 Data bus for communication with the COPS.

CA1 CA1 must be programmed as an input that latches the COPS data in the Peripheral A Port.

CA2 CA2 must be programmed to handshake with the COPS to indicate that the data has been latched.

6522 Peripheral B Port

| CR/ | CRDY | PR/ | FDIR | VC2 | VC1 | VC0 | ENC | Lisa Name |
|---|---|---|---|---|---|---|---|---|
| PB7 | PB6 | PB5 | PB4 | PB3 | PB2 | PB1 | PB0 | Synertek Name |

ENC ENC controls the CVSD Encode/Decode line. When ENC is high, the CVSD circuit is in encode mode.

VC0 - VC2 These bits control the speaker volume. When all three bits are low, the speaker is at its lowest volume. As the three bit number formed by VC0 to VC2 increases in magnitude, the speaker volume increases.

FDIR Floppy Disk Interrupt Request. Before attempting to read from the memory shared by the 68000 and the disk controller, it is necessary to insure that the floppy disk controller is the source of the level 1 interrupt. If it is not, the attempted memory access would disturb the disk operation in progress. The FDIR input allows the disk interrupt line to be sensed to check this condition. If FDIR is 1, the controller is the source of the interrupt, so the 68000 can safely read the shared memory.

PR/ Parity Reset (low true). PR/ resets the parity latch in the hard disk controller. It should be pulsed low before each disk data transfer so that any parity errors which were detected during the data transfer can be passed on to the disk driver. If CR/ (PB7) is high, PR/ is an input from the parallel port.

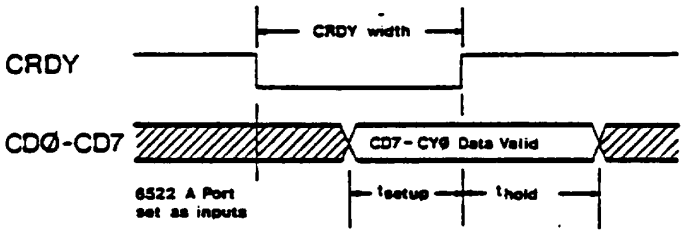
CRDY COPS Ready. This signal undergoes a high to low transition when the COPS is ready to accept a command (see the timing diagram below). There is no handshake on entering a command.

CR/ Controller Reset. When CR/ is low, you can reset the Pippin Controller connected to the hard disk port.

CB1 CB1 is used to clock data transfers between the 6522 Shift Register and the CVSD circuit under the control of T2.

CB2 CB2 is a data I/O pin between the 6522 Shift Register and the CVSD circuit.

COPS COMMAND TIMING




| | Min | Max |
|---|---|---|
| CRDY Width | 16us | 20us |
| TSetup | -8us | 15us |
| THold | 20us | 50us |

## 6522 REGISTER ADDRESSES

| Register | Reg# | Register Function | Address |
|---|---|---|---|
| ORB/IRB | 0 | Output/Input Register B | $00D181 |
| ORA/IRA | 1 | Output/Input Register A | $00D183 |
| DDRB | 2 | Data Direction Register B | $00D185 |
| DDRA | 3 | Data Direction Register A | $00D187 |
| T1C-L | 4 | T1 Low-order Latches/Counter | $00D189 |
| T1C-H | 5 | T1 High-order Counter | $00D18B |
| T1L-L | 6 | T1 Low-order Latches | $00D18D |
| T1L-H | 7 | T1 High-order Latches | $00D18F |
| T2C-L | 8 | T2 Low-order Latches/Counter | $00D191 |
| T2C-H | 9 | T2 High-order Counter | $00D193 |
| SR | 10 | Shift Register | $00D195 |
| ACR | 11 | Auxiliary Control Register | $00D197 |
| PCR | 12 | Peripheral Control Register | $00D199 |
| IFR | 13 | Interrupt Flag Register | $00D19B |
| IER | 14 | Interrupt Enable Register | $00D19D |
| ORA/IRA | 15 | Same as Register 1 but without handshaking | $00D19F |

Register 15 should be used when transferring commands to the COPS. All the 6522 register addresses are in the I/O address space. The T1 timer in the 6522 is reserved for use as an interval timer by the Operating System. For further details about the 6522, read the 6522 data sheet in the Synertek MOS Data Catalog.

The CD bus (bits CD0 to CD7 above) can send commands to the COPS.

CD COMMANDS

| CD7-CD0 | Function |
|---|---|
| 0000 0000 | Turn I/O port on |
| 0000 0001 | Turn I/O port off |
| 0000 0010 | Read clock data |
| 0001 nnnn | Write nnnn to clock |
| 0010 spmm | Set clock modes (see Clock below) |
| 0011 nnnn | Write nnnn to low indicator bits in keyboard |
| 0100 nnnn | Write nnnn to high indicator bits in keyboard |
| 0101 nnnn | Set high nibble of NMI character to nnnn |
| 0110 nnnn | Set low nibble of NMI character to nnnn |
| 1xxx xxxx | Do nothing (null command) |

If the 6522 is not programmed as an output when the CRDY line makes a high to low transition (thereby indicating that the COPS is ready to accept a command), the pullup on the CD bus prevents a spurious command from being entered.

The pin description given above indicates that the CD bus (the Peripheral A Port) should be set up to handshake and to latch data. If the 6522 is set up correctly, when the I/O board COPS has data for the 68000 it will handshake with the handshake lines and generate an interrupt.

The indicator bits can be used to control 8 LEDs. If, for example, these LEDs are installed on the keyboard, they can be used to emulate a VT-100 terminal.

RESET CODES

In addition to the key up and key down information, the keyboard COPS produces several two byte reset codes. The two bytes consist of a reset character ($80) and a code number for the following conditions:

| Reset Code | Condition |
|---|---|
| FF | Keyboard COPS failure detected |
| FE | I/O board COPS failure detected |
| FD | Keyboard unplugged. The reset code of the keyboard ID will follow when the keyboard is plugged back in |
| FC | Clock timer interrupt |
| FB | Soft power switch pressed (do a Soft-off) |
| F0 - FA | Reserved for future use |
| Ey | Clock data follows (five bytes). "y" is the year. Clock information is formatted,<br>$80 $Ey dd dh hm ms st<br>where ddd is the day, hh is the hour, mm is the minute, ss is the second, and t is the tenth of a second. |
| 0 - DF | Keyboard ID number (the current and only keyboard ID is 01). The keyboard ID reset code is produced whenever the keyboard COPS is reset. This happens when the power is turned on and when the keyboard is plugged in. |

## THE KEYBOARD

The keyboard connects to the computer through the jack on the lower right front of the Lisa. Its 80 keys are fully programmable.

The keyboard contains a microprocessor which scans the keys and communicates the occurrence of a key depression or release to the 68000. The keyboard is a true N-key rollover design. An arbitrary number of keys can be depressed without causing phantom key problems. It is the responsibility of the 68000 side of the interface to interpret key up and key down codes to properly handle functions such as shift and auto-repeat. Any key can be programmed to generate an NMI (non-maskable interrupt).

The keyboard's microprocessor contains an eight byte FIFO that buffers keyboard data if the 68000 does not collect it fast enough. Normally, this FIFO keeps keycodes from being lost, but if the keycodes are not read for a long enough time, the buffer will eventually overflow and keycodes will be lost.

## KEYCODES

Seven bits of the keycode byte identify the key which moved. The other bit indicates whether the key went up or down. The keycode format is:

drrr nnnn

If the d bit is 0, a key up transition has occurred. If it is 1, a key down transition has occurred.

The rrr and nnnn bits of the keycode have the following meanings:

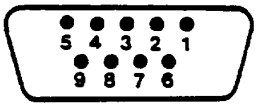
| nnnn \ rrr-> | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|
| 0000 | CLEAR | | _ - | ( 9 | E | A |
| 0001 | / | | + = | ) 0 | ^ 6 | @ 2 |
| 0010 | * | | | U | & 7 | # 3 |
| 0011 | = | | | I | * 8 | $ 4 |
| 0100 | 7 | | P | J | % 5 | ! 1 |
| 0101 | | | BACKSPACE | K | R | Q |
| 0110 | 8 | | | < [ | T | S |
| 0111 | 9 | | | > ] | Y | W |
| 1000 | - | | RETURN | M | RIGHT COMMAND | TAB |
| 1001 | 4 | | 0 | L | F | Z |
| 1010 | 6 | | | : ; | G | X |
| 1011 | + | | | " ' | H | D |
| 1100 | . | | ? / | SPACE | V | OPTION |
| 1101 | 2 | | 1 | , , | C | ALPHA LOCK |
| 1110 | 3 | | | . . | B | SHIFT |
| 1111 | NUMERIC RETURN | | | 0 | N | LEFT COMMAND |

## THE MOUSE

The mouse is an input device with motion sensors and one or more buttons. It sends movement and button information to the computer through the 6522 on the keyboard interface. This information is usually used to provide a fast and convenient pointing and drawing mechanism by tracking the mouse movement in some direct manner with a cursor on the screen.

The mouse is plugged into the nine pin EIA connector in the middle of the connector panel on the back of the Lisa. Since the mouse senses the mechanical movement of the ball as it rolls across a table, it should work on almost any convenient surface.

### MOUSE CONNECTOR PINOUT




1 - Switch L
2 - +5v
3 - Ground
4 - Left
5 - Right
6 - No connection
7 - Switch 0 (always grounded by mouse)
8 - Down
9 - Up

## MOUSE COMMANDS

Mouse commands are sent to the 6522 Peripheral A Port data bus (PA0-PA7). The command format is:

0111 ennn

If e is set, mouse interrupts are enabled. The nnn bits set the time interval that will separate mouse interrupts.

Once mouse interrupts are enabled and the time interval has been set, the COPS interrupts the 68000 whenever the mouse moves. The time between interrupts is about 4 ms times the number given in the nnn bits. For example, the command 01111100 ($7C) enables the mouse to interrupt 16 ms (100 = 4*4 ms) after the mouse begins to move. If the mouse continues to move, interrupts are generated every 16 ms.

MOUSE DATA

Mouse generated data is sent to the 68000 in three bytes:

| | |
|---|---|
| 00 | (mouse data follows) |
| dx | (change in X direction) |
| dy | (change in Y direction) |

The dx and dy values, which are in the range of -128 to 127, are reset every time the 68000 reads them. While the mouse is moving and mouse interrupts are enabled, a new dx and dy value can be read every nnn*4 ms. If the 68000 does not respond to the mouse interrupt and read the mouse data, the dx and dy values are updated and collected in the interface's eight byte FIFO buffer. If the mouse is ignored long enough, the buffer will overflow, and mouse data will become invalid.

The mouse button is returned as keycode d000 0110. If the d bit is 1, the button has been pressed, and if the d bit is 0, the button has been let up. The d bit in the keycode d000 0101 is 1 if the mouse is plugged in, and 0 if it is not. The third mouse "keycode" (d000 0111) is currently unused.

## THE CLOCK/CALENDAR CIRCUIT

The clock/calendar is a part of the keyboard encoder circuitry. The clock resolves to tenths of a second, and needs to be reset only once every sixteen years. In addition, there is an alarm clock that can be programmed to generate an interrupt at any time up to $FFFFF seconds (about 12 days) from the time it is set. This interrupt, for example, can be programmed to generate a system power up.

12.14 days
291.27 hours

The 6522 Peripheral A Port data bus (PA0-PA7) is used to communicate with the clock.

CLOCK COMMANDS

| PA7-PA0 | Function |
|---|---|
| 0000 0010 | Read clock data |
| 0001 nnnn | Write nnnn to clock |
| 0010 spmm | Set Clock modes |

CLOCK MODES

| mm | Function |
|---|---|
| 00 | Clock/Timer disable |
| 01 | Timer disable |
| 10 | Timer underflow generates interrupt |
| 11 | Timer underflow turns system on, if it is off, and generates an interrupt |

If the p bit above is 0, the system is turned off. If the p bit is one, and the system is off, the system is turned back on.

SETTING THE CLOCK

The s bit, if set, enables Clock Set mode, and the 16 nibbles of time data can be entered one digit (nibble) at a time. Only as many nibbles as desired need to be entered during the time setting operation. Once the time has been set, the s bit is cleared, and the operation is terminated. The clock and timer must be stopped while the time is being set. To do this, set the clock mode to Clock/Timer disable ($00) during the operation, then reset it to Timer disable ($01) when the operation is complete. The clock can be left running while setting the alarm, but the timer must be disabled ($01).

The order in which the time information is entered during the time set operation is:

aaaaa y ddd hh mm ss t (8 bytes total)

where:

| | |
|---|---|
| aaaaa | alarm nibbles (0-$FFFFF) (0 - 1,048,575) |
| y | year (0-15) |
| ddd | day (1-366) |
| hh | hour (0-23) |
| mm | minute (0-59) |
| ss | second (0-59) |
| t | tenth of second (0-9) |

The alarm counts down one count per second and generates an interrupt upon the $00000 -> $FFFFF transition. Since the five alarm nibbles are entered first, the alarm can be set without disturbing the clock. All the nibbles are maintained in decimal format except the alarm and year nibbles which are kept in binary format to reduce the amount of code required to handle them.

Note: Alarm setting corresponds to following times:

1,048,575 seconds
17,476.25 minutes
291.27... hours
12.14... days
1.73... weeks

## THE CVSD AND THE SPEAKER

The Lisa has a built-in speaker, and a circuit that performs digital-to-analog and analog-to-digital conversion. On the back of the Lisa is a microphone input jack, and a remote speaker jack.

The 6522 which controls the keyboard interface also controls the circuit which drives the speaker. The circuit uses a Continuously Variable Slope Deltamodulator (CVSD) to provide both audio input and output. The central part of the circuit is a Motorola 3417 CVSD chip.

The CVSD chip records and plays back audio data at a programmable rate. CVSD input can come from a microphone plugged into the back of the Lisa mainframe. The length of the recording depends on the amount of memory available for it. Since the normal CVSD speed is 16,000 bits/second, even a short recording can require a substantial amount of memory. CVSD output can be sent to the internal speaker, or picked up at the remote speaker jack.

To record a signal, the ENC bit (PB0 of the 6522) is set high. To play back a signal, the ENC bit is set low. The T2 timer (one of the 6522 internal registers) is then set to produce the desired clock speed, and data is read or written (played or recorded) to the 6522 shift register. The 6522 can be programmed to generate an interrupt whenever the shift register is full or empty. CB1 clocks the data transfers between the 6522 shift register and the CVSD.

The volume of both the CVSD and the speaker output is controlled by the three VC bits (PB1, PB2, and PB3 in the 6522). The volume is proportional to the magnitude of the 3-bit unsigned integer formed by the VC bits.

In addition to handling voice I/O, the CVSD can be programmed to produce tones from the speaker. The CVSD represents a waveform as a series of 1's and 0's where a 1 indicates a rising signal, and a 0 indicates a descending signal. A series of alternating 1's and 0's therefore produces no sound at all. When its comparator indicates to the CVSD that the signal has been rising faster than the rate at which the CVSD has been rising for several samples, the CVSD increases the slope of its signal. This continuously variable slope enables it to produce a closer approximation of the desired waveform than a simple slope modulator can. Of course, the quality of the approximation depends on the rate at which the waveform is sampled. The higher the sampling rate, the better. See the Motorola MC3417 data sheet for full details. To send a pattern to the CVSD, load the shift register and program the T2 timer to produce the desired frequency. The resultant waveform need not be a square wave.

## THE SERIAL INTERFACES

The Lisa has two built-in RS232-C serial communication ports. Both ports can support local asynchronous commnuications at speeds up to 19.2 Kbaud. In addition, port 1 conforms to RS232-C type D specifications, permitting full modem control with either synchronous or asynchronous protocols.

The serial logic is implemented with a NEC 7201 dual channel communication interface. Refer to the 7201 data sheet for complete information about the operation of the chip. Note that autovectors are used so you do not need to program the 7201 to supply the interrupt vectors.

The 7201 has neither an internal baud rate generator, nor a Ring Indicator input. The baud rate generator is provided by the serial interface. The Ring Indicator input is brought into the channel B DCD (Data Carrier Detect) line where it must be decoded by software. The DCD line can be programmed to produce an interrupt whenever there is a transition on that line.

SERIAL INTERFACE LINES

| Signal Name | Description | RS-232 Pin # Channel A | Channel B |
|---|---|---|---|
| TxD | Transmit Data | 2 | 2 |
| RxD | Received Data | 3 | 3 |
| RTS | Request To Send | 4 | 4 |
| CTS | Clear To Send | 5 | — |
| DTR | Data Terminal Ready | 20 | 20 |
| DCD | Data Carrier Detect | 8 | — |
| RI | Ring Indicator | 22 | — |
| TxC | Transmit Clock Input | 15 | — |
| RxC | Receive Clock Input | 17 | — |
| TEXT | Transmit Clock Output | 24 | — |
| DSR | Data Set Ready | — | 6 |

Both channels are set up to be the Data Terminal Equipment (DTE) end of the communications system. To connect other DTE devices to the Lisa, you need a modem eliminator (a null modem). The Apple /// null modem (590-0029-00) with a communciations card cable can be used to connect the Lisa to a Qume printer through channel B.

## BAUD RATE GENERATION

The baud rate generator produces standard baud rates as well as a baud rate that is under the control of the 6522. The 6522 controlled rate can range from less than one baud up to about 1000 baud. Since the T1 timer in the Hard Disk Interface's 6522 is connected to the baud rate generator via PB7, either data communications channel can use the 6522 timer for baud rate control. The baud rate generator has seven control line inputs (C0-C6) which are set or reset by control pulses.

BAUD RATE CONTROL LINES

Channel 1

| CCC 210 | Baud Rate |
|---|---|
| 000 | External Clock |
| 001 | 6522 control |
| 010 | 300 baud |
| 011 | 1200 baud |
| 100 | 2400 baud |
| 101 | 4800 baud |
| 110 | 9600 baud |
| 111 | 19200 baud |

Channel 2

| CCC 654 | Baud Rate |
|---|---|
| 000 | 6522 control |
| 001 | 300 baud |
| 010 | 600 baud |
| 011 | 1200 baud |
| 100 | 2400 baud |
| 101 | 4800 baud |
| 110 | 9600 baud |
| 111 | 19200 baud |

| C3 | Receiver Clock |
|---|---|
| 0 | same as transmitter clock |
| 1 | external clock (for synchronous communications) |

Control Bit Addresses

| Bit | Set Address | Reset Address |
|---|---|---|
| C0 | $00D002 | $00D000 |
| C1 | $00D006 | $00D004 |
| C2 | $00D00A | $00D008 |
| C3 | $00D00E | $00D00C |
| C4 | $00D012 | $00D010 |
| C5 | $00D016 | $00D014 |
| C6 | $00D01A | $00D018 |

7201 Register Addresses

| | | |
|---|---|---|
| Channel 1 | Data: | $00D041 |
| | Control: | $00D045 |
| Channel 2 | Data: | $00D043 |
| | Control: | $00D047 |

Note that the 7201 baud clock is at 16 times the baud rate in all the cases given above. For 9600 baud, for example, the clock runs at 153.5 KHz. It is possible to program the 7201 to use the xl clock in synchronous communications, so baud rates up to 307 Kilobaud are possible.

## THE FLOPPY DISK CONTROLLER

The built-in floppy disk controller consists of a 6505 based microcomputer and a disk interface similar to that of the Apple II.

The 6505 has 1 KByte of RAM and 4 KBytes of ROM. The RAM is shared with the 68000 address space. The 68000 and 6505 communicate data and commands through this shared memory. Use the MOVEP instruction to convert 16 bit 68000 data to the 8 bit 6505 format. When data moves from the 6505 to the 68000, the shared memory appears in the low byte of each word read by the 68000.

At power on, the 6505 sets the disk control lines to a safe state, does a checksum on the ROM, tests the RAM, clears some of the RAM, and leaves the results of the ROM and RAM tests in the location $00C003. The 6505 then interrupts the 68000 and waits for a command.

## DISK CONTROLLER COMMANDS

The command block consists of 20 bytes in the low 20 words of the 6505 address space. 17 of these bytes are used to pass parameters to the Read/Write/Track/Sector (RWTS) routine. The other 3 bytes are used for communication between the 68000 and the 6505. The command byte appears at the 68000 address $00C001, and at the 6505 address $0000.

The disk controller reads the command byte until a valid command appears, or until an interrupt arrives from the disk drives. The disk drives generate an interrupt when the eject button is pushed and when the disk is in place. When a valid command appears, the controller reads the command, clears the command byte, executes the command, and interrupts the 68000 if necessary. The controller then waits for the next command.

### FLOPPY DISK CONTROLLER COMMANDS

(68000 writes to $00C000)

| | |
|---|---|
| $81 | Execute RWTS using $00C003 - $00C031 as parameters |
| $84 | JSR to routine pointed to by $00C003 and $00C005 |
| $85 | Clear Interrupt Status |
| $86 | Set Interrupt Mask |
| $87 | Clear Interrupt Mask |

When the 6505 generates an interrupt, the 68000 receives it as a level 1 interrupt. The 68000 can discover the state of the floppy disk interrupt request line by reading the location PB4 (also known as FDIR) in the keyboard's 6522. If this bit is low, the floppy disk controller has interrupted the 68000, so the 68000 can examine the shared memory area to determine the results of the command. The 68000 should check the FDIR bit before accessing the shared memory because the 6505 locks the 68000 out of the shared memory during a disk transfer. If the 68000 tries to access the shared memory while it is locked out, a Bus Timeout Error occurs.

## THE EXECUTE RWTS COMMAND

The main part of the code in the disk controller ROM is executed by the Execute RWTS command ($81). The routines involved are very similar to those used by DOS 3.3 (see the DOS 3.3 Manual, Apple Part No. xxxxxxxxx). As in DOS 3.3, an I/O Block (IOB) is used to pass parameters to the RWTS routines.

FLOPPY DISK I/O COMMAND BLOCK

| | | |
|---|---|---|
| $00C003 | Command code | |
| | $00 | Seek |
| | $01 | Read |
| | $02 | Write |
| | $03 | Reserved |
| | $04 | Format |
| | $05 | Clamp |
| | $06 | Examine status |
| | $07 | Unclamp |
| $00C005 | Drive select | |
| | $00 | Drive 2 (Bottom drive) |
| | $80 | Drive 1 (Top drive) |
| $00C007 | Side select | |
| | $0x | Side 1 (Top side) |
| | $1x | Side 2 (Bottom Side) |
| $00C009 | Sector number | (0 to 22) |
| $00C00B | Track number | (0 to 44) |
| $00C00D | Error code | |
| | $00 | No error |
| | $40 | Drive error |
| $00C011 | Version number | (2 bytes) (Operating System Header) |
| $00C015 | Volume number | (2 bytes) |
| $00C019 | File ID | (2 bytes) |
| $00C01D | Data used | (2 bytes) |
| $00C021 | Absolute page | (2 bytes) |
| $00C025 | Relative page | (2 bytes) |
| $00C029 | Forward link | (2 bytes) |
| $00C02D | Backward link | (2 bytes) |
| $00C031 | Previous drive number (drive accessed on last command) | |

Tag bytes (12)

The I/O Block controls the transfer of data from the 6505 to the disk. The transferred data is in a buffer located at $00C501 - $00C7FF on the 68000 side and at $0200 - $03FF on the 6505 side of the shared memory. The 68000 can use the MOVEP instruction to transfer this data from the shared memory to its main memory.

## DISK DATA STORAGE

Programmer's Apple MAC Sourcebook Tom Hogan MB

Data is stored in 512 byte blocks on the floppy disk. The number of blocks in a track depends on the track number. A block is equivalent to a DOS 3.3 sector. The total capacity of each side of the disk is 868 blocks or 444,416 bytes.

"Twiggy"

| Track number | Blocks in track |
|---|---|
| 0 - 3 | 23 |
| 4 - 9 | 22 |
| 10 - 16 | 21 |
| 17 - 22 | 20 |
| 23 - 28 | 19 |
| 29 - 32 | 18 |
| 33 - 38 | 17 |
| 39 - 45 | 16 |

"400K floppy"

| TRACK | sec/TRK | Sec# |
|---|---|---|
| 0-15 | 12 | (0-191) |
| 16-31 | 11 | (192-367) |
| 32-47 | 10 | (368-527) |
| 48-63 | 9 | (528-671) |
| 64-79 | 8 | (672-799) |

800K sectors 2x (2x sided) offset from zero

## THE JSR COMMAND

The JSR command ($84) transfers control from the normal RWTS routines to another program. This program can be located in the ROM, or can be downloaded into the shared memory and then executed. The two bytes after the command byte ($00C003 and $00C005) contain the low and high bytes of the address of the program to be executed as seen from the 6505 side of shared memory. To calculate the 6505 address, shift the 68000 address right one bit, then mask off the high order 9 bits.

## INTERRUPTS

The Clear Interrupt Status command ($85) clears the interrupt line from the 6505 to the 68000 using a mask found at $00C003. A one in the mask byte clears the associated interrupt bit. Several interrupt sources can cause the 6505 to assert the FDIR/ bit. The identity of the interrupt source can be found at $00C081.

FLOPPY CONTROLLER INTERRUPT SOURCE

| | |
|---|---|
| D7 | Set if D6, D5, or D4 is set |
| D6 | Set if RWTS is complete for Drive 1 |
| D5 | Set when the Drive 1 Button is pushed |
| D4 | Set when Drive 1 signals "disk in place" |
| D3 | Set if D2, D1, or D0 is set |
| D2 | Set if RWTS is complete for Drive 0 |
| D1 | Set when the Drive 0 Button is pushed |
| D0 | Set when Drive 0 signals "disk in place" |

The 68000 can clear any of the bits in this word with the Clear Interrupt Status command. D7 and D3 remain set as long as any associated bit is set.

The Interrupt Mask byte at $00C083 determines which drive can interrupt the 68000. Only bits 7 (Drive 1) and 3 (Drive 0) are used. The Set Interrupt Mask command ($86) sets the bits of this mask on the basis of the operand byte at $00C003. The Clear Interrupt Mask command ($87) clears these bits on the same basis. If interrupts are masked off, no interrupts can occur until they are enabled again.

## NON-VOLATILE MEMORY

The 1K floppy disk controller shared memory RAM is backed up by the same power source that backs up the clock/calendar circuit. The last 64 words in the 6505's page 0 are reserved for the "parameter memory" of the system. Since the 6505 does not use these bytes, the 68000 can store data here which it wants to be able to access intact when the system is powered on. The reserved memory appears at the 68000 locations $00C181 - $00C1FF.

## THE BATTERY

The disk controller also controls the charging current for the battery back up. Ni-Cd batteries last much longer if the charging current is pulsed with a 50% duty cycle at 200% of the normal charging current. While the disk controller is waiting for a command, it cycles the charging current at about ten times per second. The charging current is turned off while the controller is executing a command.

As long as the system is plugged in, the backup battery is not in use. If there is a power failure, or the system is unplugged, the battery can last about 20 hours.

## THE HARD DISK INTERFACE

The hard disk interface is a general purpose eight bit parallel port which can be used to connect the Pippin hard disk controller to the Lisa. The Pippin is a 5-1/4 inch Winchester-type disk capable of storing about five megabytes of data. The 6522 which connects the interface to the 68000 bus also provides the programmable baud rate to the serial ports and generates the slow data bus that latches data for transfer to the contrast control latch.

### HARD DISK INTERFACE SIGNAL LINES

| | |
|---|---|
| DD0 - DD7 | Disk Data lines. These eight bi-directional lines communicate data to and from the disk. |
| RW | Read/Write. RW controls the direction that data flows on the Disk Data lines. When RW is high, data moves from the disk to the Lisa. When RW is low, data moves from the Lisa to the disk. |
| PARITY/ | This bit contains odd parity on the data being transmitted. If any parity errors are caught, the data can be retransmitted. |
| PSTRB/ | Processor Strobe line. PSTRB controls the data flow from the Lisa to the disk interface. The timing diagram is given below. |
| CMD/ | The Command line tells the disk controller that there is a command present. |
| BSY/ | The Busy line is high when the disk controller is busy. BSY/ is also used as a handshake line with the CMD/ signal. |
| OCD | Open Cable Detect. If the disk controller is connected to the Lisa, the OCD line is low. If the controller is not connected, OCD is pulled high. |
| CRES/ | Controller Reset. This line is driven by the CR/ bit (PB7) in the keyboard 6522. In addition, when the port is used as a general purpose parallel port, CRES/ is an input which can be read on the keyboard 6522 PR/ bit if the CR/ bit is 1. |
| M3 | M3 is an input that produces the M3 keycode. It provides a keyboard interrupt which the general purpose parallel port can use. |

The 15 hard disk interface signal lines are controlled by a 6522. The relevant 6522 lines are:

| 6522 line | | disk interface | |
|---|---|---|---|
| PA0 - PA7 | Peripheral A Port | DD0 - DD7 | Disk Data |
| CA1 | Control line 1 A | PSTRB/ | Processor Strobe |
| CA2 | Control line 2 A | latch CA1 | |
| PB2 | Peripheral B Port 2 | DEN | Disk Enable |
| PB3 | Peripheral B Port 3 | RW | Read/Write |
| PB4 | Peripheral B Port 4 | CMD | Command |
| PB5 | Peripheral B Port 5 | OCD | Open Cable Detect |
| PB6 | Peripheral B Port 6 | PARITY/ | Parity error |
| PB7 | Peripheral B Port 7 | BRC | Baud Rate Control |
| PB7 on the Keyboard 6522 | | CRES/ | Controller Reset |
| CB2 | Control line 2 B | BSY | Busy |

PA0 - PA7 The slow data bus. These eight bits are buffered with an LS245 to produce the eight Disk Data lines (DD0 - DD7). They are also used to provide data to the contrast control latch.

CA1 CA1 generates the PSTRB/ signal. If CA1 is in pulse handshake mode, and the DEN bit is low, the CA1 line produces the proper pulse to transfer data to or from the disk with each read or write to the Peripheral A Port.

CA2 CA1 is fed directly to CA2 to allow latch mode to be used on the Peripheral A Port while data is being read from the disk.

PB0 - PB7

| BRC | RDP | OCD | CMD | RW | DEN | | | Lisa |
|---|---|---|---|---|---|---|---|---|
| PB7 | PB6 | PB5 | PB4 | PB3 | PB2 | PB1 | PB0 | Synertek |

PB0 - PB1 Unused.

DEN Disk Enable. DEN enables the buffers which drive the disk interface lines. When DEN is low, the buffers are enabled. When DEN is high, they are disabled.

RW RW controls the disk Read/Write line, and sets the data direction on the LS245 buffer for the Disk Data lines. When RW is high, the DD lines are enabled to read from the disk. When RW is low, the DD lines write to the disk.

| | |
|---|---|
| CMD | The CMD bit controls the CMD/ line to the disk. The CMD bit and the CMD/ line have the same level. |
| OCD | When the DEN bit is low, the OCD bit returns the state of the disk interface OCD line. |
| RDP | Read Parity permits a program to read the state of the PARITY/ flip-flop after completion of a data transfer. If RDP is high, there has been a parity error detected. |
| BRC | Baud Rate Control. BRC is the output of the T1 timer that is connected to the Baud Rate Generator in the serial interface. It can be used to produce odd baud rates.<br><br>CR/ is PB7 on the other 6522 (the one on the keyboard). |
| CB1 | Unused. |
| CB2 | CB2 is connected to the disk interface BSY signal. |

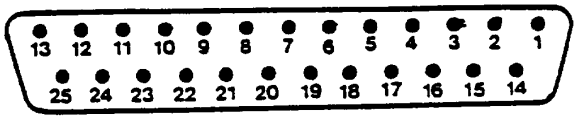
## 6522 REGISTER ADDRESSES

| | | |
|---|---|---|
| ORB/IRB | Output/Input Register B | $00D101 |
| ORA/IRA | Output/Input Register A | $00D109 |
| DDRB | Data Direction Register B | $00D111 |
| DDRA | Data Direction Register A | $00D119 |
| T1C-L | T1 Low-order Latches/Counter | $00D121 |
| T1C-H | T1 High-order Counter | $00D129 |
| T1L-L | T1 Low-order Latches | $00D131 |
| T1L-H | T1 High-order Latches | $00D139 |
| T2C-L | T2 Low-order Latches/Counter | $00D141 |
| T2C-H | T2 High-order Counter | $00D149 |
| SR | Shift Register | $00D151 |
| ACR | Auxiliary Control Register | $00D159 |
| PCR | Peripheral Control Register | $00D161 |
| IFR | Interrupt Flag Register | $00D169 |
| IER | Interrupt Enable Register | $00D171 |
| ORA/IRA | Same as Register 1 but without handshaking | $00D179 |

The register addresses for this 6522 jump by 4 words per register. This peculiar address decode allows a MOVEP.L instruction to transfer 4 bytes to or from the Peripheral A Port during a disk read or write. By decreasing the number of instructions needed to transfer the data, the transfer rate is increased.

The parity flip-flop should be reset before each transfer by clearing, then setting the PRES/ bit in the keyboard 6522.

## THE HARD DISK CONNECTOR




1 - Ground
2 - Ground
3 - RW
4 - Ground
5 - DD0
6 - DD1
7 - blocked
8 - DD2
9 - Ground
10 - Ground
11 - DD5
12 - DD6
13 - DD7
14 - Ground
15 - PSTRB/
16 - BSY
17 - CMD/
18 - PARITY/
19 - OCD
20 - Ground
21 - CRES/
22 - DD3
23 - DD4
24 - Ground
25 - CHK

## I/O SPACE ADDRESSES

| Address | Function |
|---|---|
| $000000-1FFF | Slot 0 Low Decode |
| $002000-3FFF | Slot 0 High Decode |
| $004000-5FFF | Slot 1 Low Decode |
| $006000-7FFF | Slot 1 High Decode |
| $008000-9FFF | Slot 2 Low Decode |
| $00A000-BFFF | Slot 2 High Decode |
| $00C000-1 | Floppy Disk Controller Command Byte |
| $00C003-31 | IOB |
| $00C081 | Interrupt Source |
| $00C083 | Interrupt Mask |
| $00C181-1FF | Parameter Memory ✔ |
| $00C501-7FF | Shared Memory ✔ |
| $00D000 | Serial Interface C0 Control Bit Reset |
| $00D002 | C0 Set |
| $00D004 | C1 Reset |
| $00D006 | C1 Set |
| $00D008 | C2 Reset |
| $00D00A | C2 Set |
| $00D00C | C3 Reset |
| $00D00E | C3 Set |
| $00D010 | C4 Reset |
| $00D012 | C4 Set |
| $00D014 | C5 Reset |
| $00D016 | C5 Set |
| $00D018 | C6 Reset |
| $00D01A | C6 Set |
| $00D041 | 7201 Channel 1 Register Data |
| $00D043 | Control |
| $00D045 | 2 Data |
| $00D047 | Control |
| $00D101 | Hard Disk Interface 6522 ORB/IRB register |
| $00D109 | ORA/IRA |
| $00D111 | DDRB |
| $00D119 | DDRA |
| $00D121 | T1C-L |
| $00D129 | T1C-H |
| $00D131 | T1C-L |
| $00D139 | T1C-H |
| $00D141 | T2C-L |
| $00D149 | T2C-H |
| $00D151 | SR |
| $00D159 | ACR |
| $00D161 | PCR |
| $00D169 | IFR |
| $00D171 | IER |

9 not 1

| | |
|---|---|
| $00D179 | ORA/IRA |
| $00D181 | Keyboard 6522 ORB/IRB register |
| $00D183 | ORA/IRA |
| $00D185 | DDRB |
| $00D187 | DDRA |
| $00D189 | T1C-L |
| $00D18B | T1C-H |
| $00D18D | T1C-L |
| $00D18F | T1C-H |
| $00D191 | T2C-L |
| $00D193 | T2C-H |
| $00D195 | SR |
| $00D197 | ACR |
| $00D199 | PCR |
| $00D19B | IFR |
| $00D19D | IER |
| $00D19F | ORA/IRA |
| $00E000 | Memory Diagnostic DIAG1 Reset |
| $00E002 | Set |
| $00E004 | DIAG2 Reset |
| $00E006 | Set |
| $00E008 | Context Selection SEG1 Reset |
| $00E00A | Set |
| $00E00C | SEG2 Reset |
| $00E00E | Set |
| $00E010 | SETUP Register Set |
| $00E012 | Reset |
| $00E014 | Enable Soft Memory Error Detect Reset |
| $00E016 | Set |
| $00E018 | Enable Vertical Retrace Interrupt Reset |
| $00E01A | Set |
| $00E01C | Enable Hard Memory Error Detect Reset |
| $00E01E | Set |
| $00E800 | Video Address Latch |
| $00F000 | Memory Error Address Latch |
| $00F800 | Status Register |

## THE EXPANSION I/O SLOTS

Three expansion devices can be connected to the Lisa system bus by inserting the expansion cards into the slots at the back of the machine. Slot 0 is the leftmost slot, as you face the back of the machine. The three expansion connectors are 56 pin ZIF connectors. These connectors permit you to install a card without removing the card cage from the chassis.

## SIGNAL DESCRIPTION

The signal description given below describes the basic operation of the bus interface signals. Any signal that has the same name as a 68000 signal (AS/ for example) is just a buffered version of that 68000 signal, and its function on the expansion bus is the same as that described in the 68000 Micrprocessor User's Manual.

EXPANSION I/O CONNECTOR (top view)

Front of Machine

| | | | |
|---|---|---|---|
| +5 Volts | 56 | 55 | +5 Volts |
| Digital Ground | 54 | 53 | Digital Ground |
| +5 STDBY | 52 | 51 | Analog Ground |
| UDS/ | 50 | 49 | LDS/ |
| READ | 48 | 47 | AS/ |
| DTACK/ | 46 | 45 | +12 Volts |
| VPA/ | 44 | 43 | VMA/ |
| BA12 | 42 | 41 | BA11 |
| BA10 | 40 | 39 | BA9 |
| BA8 | 38 | 37 | BA7 |
| BA6 | 36 | 35 | BA5 |
| BA4 | 34 | 33 | BA3 |
| BA2 | 32 | 31 | BA1 |
| BD0 | 30 | 29 | BD1 |
| BD2 | 28 | 27 | BD3 |
| BD4 | 26 | 25 | BD5 |
| BD6 | 24 | 23 | BD7 |
| BD8 | 22 | 21 | BD9 |
| BD10 | 20 | 19 | BD11 |
| BD12 | 18 | 17 | BD13 |
| BD14 | 16 | 15 | BD15 |
| BG out | 14 | 13 | BG In |
| BR | 12 | 11 | E |
| BGACK/ | 10 | 9 | LDMA/ |
| CPUCK | 8 | 7 | RESET/ |
| IAKn/ | 6 | 5 | INTn/ |
| SHn/ | 4 | 3 | SLn/ |
| -12 Volts | 2 | 1 | -5 Volts |

Back of Machine

BD0 - BD15 The 16 Buffered Data lines are the system data bus. These data lines are bidirectional and are pulled up on the motherboard.

BA1 - BA12 The 12 Buffered Address lines select one of 2048 words in both the Low and High Selects. Each I/O card, therefore, has a maximum of 4096 words (8192 bytes) of addressability. The Buffered Address lines are the 12 low address lines generated after the MMU. Thus, the operating system can assign a process a segment that encloses the 8192 addressable bytes on the I/O card, and the process will address the I/O card from addresses $0 to $1FFF independent of the location of the slot.

AS/ Address Strobe indicates that the 68000 has initiated a memory cycle. Due to memory map delay, the high 3 address bits may not be valid when the address strobe is asserted, but the low 8 bits are valid.

UDS/ Upper Data Strobe indicates that the memory cycle being performed affects the upper data byte (BD8 - BD15).

LDS/ Lower Data Strobe indicates that the memory cycle being performed affects the lower data byte (BD0 - BD7).

DTACK/ Data Transfer Acknowledge indicates that the I/O device has performed the data transfer requested by the 68000, and that the cycle can complete.

READ READ indicates the direction of the data transfer on the Buffered Data lines. When READ is high, data goes from the I/O device to the 68000. When READ is low, data goes from the 68000 to the device.

VPA/ Valid Peripheral Address indicates to the 68000 that the addressed device works with a 6800 bus cycle. DTACK/ must not be asserted if VPA/ is asserted.

VMA/ Valid Memory Address indicates that the 68000 has received the VPA and is executing the requested 6800 cycle.

E E is the 68000 equivalent of the 6800 phi2 clock. E is high for 4 clock cycles and low for 6 cycles. The frequency of the E clock is therefore 500 KHz.

| | |
|---|---|
| BR | Bus Request indicates to the 68000 that another device wishes to take control of the bus. |
| BG In/ | The 68000 issues a Bus Grant after the DMA device has been given control of the bus. If no higher priority device controls the bus and the present I/O card has asserted Bus Request, the card can take control of the bus when the Bus Grant signal is asserted. |
| BG Out/ | The Bus Grant signal is propogated to the lower priority devices when the current device is not using the bus. |
| BGACK/ | Bus Grant Acknowledge is produced by the I/O device when it assumes control of the bus. It is asserted until that device is finished with the bus. |
| LDMA/ | Load Direct Memory Access is used to load the DMA high address latch. Since the expansion slot has only 56 pins, the high 8 address lines cannot be sent to memory directly. Instead, a latch is provided on the CPU board to hold these 8 bits. The latch is loaded with the contents of the low 8 data bits (BD0 - BD7) when there is a rising edge on the LDMA/ signal. LDMA/ is pulled up on the CPU board. The DMA device must load this register each time it receives control of the bus, before any DMA cycles occur. |
| CPUCK | The CPU Clock is the 5 MHz signal that drives the 68000. It can be used as a general purpose clock, or to control timing during the DMA transfer. |
| RESET/ | RESET/ tells the I/O device to return to its power-on state. RESET/ is asserted by the CPU when the power is turned on, and when the RESET instruction is executed. When the power is off, this line can be pulled low to turn it back on by using the soft-on feature. |
| +5STDBY | This signal provides a +5 volt power supply to those devices on the I/O board which need to be powered on whenever the system is turned on. The system clock/calendar circuitry is powered by this supply. When the normal +5 volt supply is operating, no current should be drawn from +5STDBY. The maximum current that can be drawn from this supply is 50 ma per expansion board. |

INTn/ INTn/ is the Interrupt output from the expansion board to the 68000. When an expansion device interrupts the system, INTn/ should be held low until it is reset by the 68000.

IAKn/ When the 68000 recognizes the INTn signal, it asserts the Interrupt Acknowledge signal. IAKn/ is asserted when AS is asserted, so to provide a 68000 interrupt vector, the vector is applied to the bus and a DTACK/ signal is generated. If an autovector cycle is desired, the IAKn/ signal is routed to the VPA/ signal, and the 68000 provides the interrupt vector, depending on which slot the interrupt is provided for. To insure that a Bus Error does not occur, an I/O card should always provide an interrupt vector, even if it doesn't think it asserted the interrupt line. To keep spurious interrupts from crashing the system, the interrupt handler should test each interrupt source on the I/O card, and should return from the interrupt if the card did not cause the interrupt.

SLn/, SHn/ The two select signals for each expansion slot select contiguous 2048 word sections of memory. SLn/ selects the low 2048, and SHn/ selects the high 2048 words. The select lines are asserted after the the addresses are valid, and remain asserted until the data bus can be tri-stated on a read cycle.

Power Supplies

The I/O cards are provided with four voltages from the power supply: +12, +5, -5, and -12 volts. Both Digital and Analog Grounds are brought to the board. All three expansion I/O cards can draw a maximum of 75 ma from the -5 volt line, and a maximum of 100 ma from the -12 volt line. The cards can draw up to 10 watts from the other lines.

300 ma @ 12V

10 watts } 3 cards

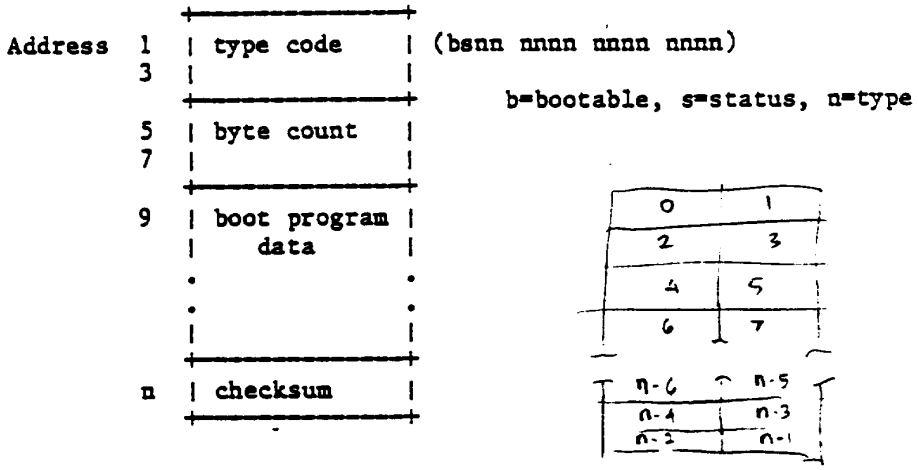
300 ma @12V

10 watts } 3 cards

DEVICE PROTOCOL

To enable the boot ROM to boot the system at power on from an expansion device, every I/O card must conform to certain identification and control protocols. Each card must contain a small ROM which identifies the card to the boot ROM. If the card has a boot program, the program also must be contained in this ROM. The I/O card's ROM is addressed as the lowest significant byte of the SLn select for the card. The ROM contents are:

I/O CARD ID AND BOOT ROM




The type code is a 16 bit female byte sex number (most significant byte first) stored in the first two bytes of the ROM. If the first bit is a 1 (see above), the device claims to be bootable. If the device's status should be checked at boot time, its status bit (the second bit) is a 1. The other 14 bits are the identification number which identifies the card.

Before the boot ROM executes the card's boot program, it loads register A1 with the address of the first word of SLn for the expansion slot. The boot program can therefore access the card by addressing relative to A1.

If the status bit is on, the status program, which can contain device setup code, should begin at the first word of the program data. The boot program, if present, should begin at the third word of the data. The status program should always return to the boot ROM leaving its result in register D0. A result of 0 indicates that the device is happy. The boot ROM checks each card's status bit and executes the status program, as required. It then checks parameter memory to decide which device to try to boot from, and checks that device's boot bit to insure that a boot program is present.

The byte count is a 16 bit female byte sex number that indicates the length in bytes of the boot program data. The boot program is stored as bytes, but is read into main memory as 16 bit female words. The 2 byte checksum should produce a 0 (for all 16 bits) when it is added to all the rest of the data in ROM, including the type code and the byte count.

If the card returns a type code of $FFFF, the boot ROM responds as if there were no device in the slot. This code is provided for devices that do not contain a boot program during the hardware development phase. All production cards, however, must be self-identifying.

Since addressing is done through the SLn decode, the expansion device boot ROM cannot be larger than 2048 bytes. Smaller ROMs can be used.

## DIRECT MEMORY ACCESS

The DMA scheme provided by the Lisa is a slight modification of the system provided by the 68000 alone. Since the BG lines are daisy chained, the prioritization of DMA transfers is left up to the design of the expansion I/O cards and the location of the cards. The hardwired priority for DMA devices is:

| | |
|---|---|
| Highest Priority | Slot 2 |
| | Slot 1 |
| Lowest Priority | Slot 0 |

To insure that the daisy chain is not broken, any card that does not use DMA should tie BG In to BG Out. The higher priority device should not propogate the BG signal to lower priority devices until it has finished using the bus. To insure proper propogation of the daisy chained signal, it is necessary to fill the expansion slots from right to left (from slot 2 to slot 0). Since a DMA device takes over the bus from the 68000, DMA transfers should be limited in time to about 1 ms.

The DMA controller associated with the expansion device controls the AS, UDS, LDS, and READ lines, as well as the data and address buses. The DMA controller should simulate the 68000 control of these lines as closely as possible. The timing on the DTACK/ signal is especially critical. To avoid wait states in the 68000 on normal memory fetches, the memory controller asserts DTACK/ one CPU clock cycle before the data transfer is complete.

GLOSSARY

Asynchronous modem
: A modem which handles asynchronous transmissions. In asynchronous communication each character is transmitted with its own framing information telling the receiver where the character starts and stops. Since each character is a complete message, the time interval between successive characters need not be fixed.

Autovector
: To handle certain interrupts, the 68000 automatically jumps to a location predefined for the given interrupt. The jump to the interrupt handler preloaded at this location is called an autovector.

Baud rate
: The rate at which a Modem sends and/or receives information. 110 baud means the Modem is handling approximately 110 bits per second. If there are two stop bits, a start bit, a parity bit, and a seven bit ASCII character code, 110 bits per second translates into about 10 characters per second.

Bit
: An acronym for binary digit. A bit is a piece of data with only two possible states, 1 or 0.

Block
: A contiguous set of bits.

Boot
: When a computer is turned on, it has to "bootstrap" itself into a useable state. The process of getting the operating system software into place and executing is called booting.

Bus
: A set of parallel wires (traces, paths) which carry related data and control information from one device to another.

Byte
: A group of bits. On the Lisa, a byte is always 8 bits.

Byte parity
: When bytes of data are being moved around, one or more bits in the byte can get improperly flipped. These incorrect bits can sometimes be detected by checking the byte parity. The byte's parity is odd if there is an odd number of 1 bits in the byte. If another bit is available in the byte, the sender can insure that every byte has even parity upon being transmitted. The receiver can then check each byte's parity, and if any are odd, it can inform the system that something has gone wrong.

Card cage
: The metal box in which the printed circuit cards reside.

Checksum
: Similar to byte parity. A number used to ensure that data has not suffered degradation during transfer.

| | |
|---|---|
| Clock | A continuous, regular waveform used to control the timing of logic decisions. |
| CMOS | Complementary Metal Oxide Semiconductor. CMOS combines N-channel and P-channel MOS transistors to give rather high speed operation, good noise rejection, low power consumption, and large fan-out. Since the non-volatile parameter memory must not consume much power, it is implemented with CMOS chips. |
| COPS | Control Oriented Processor System |
| CPU | Central Processing Unit. A Motorola 68000 in the Lisa. |
| CVSD | Continuously Variable Slope Delta Modulator. A kind of analog to digital and digital to analog converter. |
| Cycle | The interval between two clock pulses. |
| Daisy chain | A daisy chain is a method of connecting several devices to a single I/O port. |
| Decode | The opposite of a select. The decoder's input address determines which of its many outputs will go low. |
| DMA | Direct Memory Access. Normal memory access goes through the 68000 and its memory manager. A device can, however, read and write memory locations directly, without any intervention from the CPU. |
| DOS 3.3 | Disk Operating System 3.3, used on the Apple II. The floppy disk controller used by the Lisa is very similar to that supported by DOS. |
| D/A | Digital to Analog. When a digital signal is used to control an analog device, the bits in the digital word must be converted into analog voltage levels. |
| ECC | Error Correction Code. |
| Event | A block of data associated with an exception. |
| FIFO | First In First Out, A queue. |
| Flip flop | A digital circuit used to store one bit of data. |

Female → bytes stored in MSB to LSB order in seq... RAM

FEMALE: Bytes stored in MSB to LSB order in sequential RAM byte order (a/a 8080)

| | |
|---|---|
| Gate | A gate is a switch which controls the flow of data according to some Boolean function of its inputs. An AND gate, for instance, has two inputs, a data line and a control line. If the control line is 0 (false), data from the input line cannot get to the output (anything AND false is false). If the control line is 1 (true), any data on the data line is allowed to pass through the gate. |
| Handshake | Handshaking controls the transfer of data between devices. Each device has a way to tell the other that its side of the operation is complete. For example, a processor writes data to a register, then sends a signal to a device that data is ready to be read. The device reads the data, then sends the processor a signal that it has finished reading and is ready for more. The handshaking insures that the processor does not write new data to the register (clobbering the old data) before the device has had a chance to read the old data. |
| Hang | A computer hangs when it gets into some infinite loop or wait state. There are times when the only recourse is to reboot the machine. |
| Hard disk | A disk that is not floppy. Hard disks look a little like a stack of phonorecords. Higher data densities can be supported by a hard disk because it does not change size and position as much as the floppy disk does. |
| High | A voltage state. A high voltage can signify either true or false, depending on the logic being used. |
| KHz | Kilo-Hertz (1000 cycles per second) |
| IOB | Input/Output Block. A block of memory used to control and communicate with the floppy disk controller. |
| LED | Light Emitting Diode. A semiconductor device that emits light when a small current passes through it. |
| Low | A voltage state (0 volts). Low voltage can signify either true or false, depending on the logic being used. |
| ma | milliamp |
| Mainframe | The main computer. |
| MHz | Megahertz |

Male → swap MSbyte & LSbyte of a word when accessing sequential bytes (a la 8080)

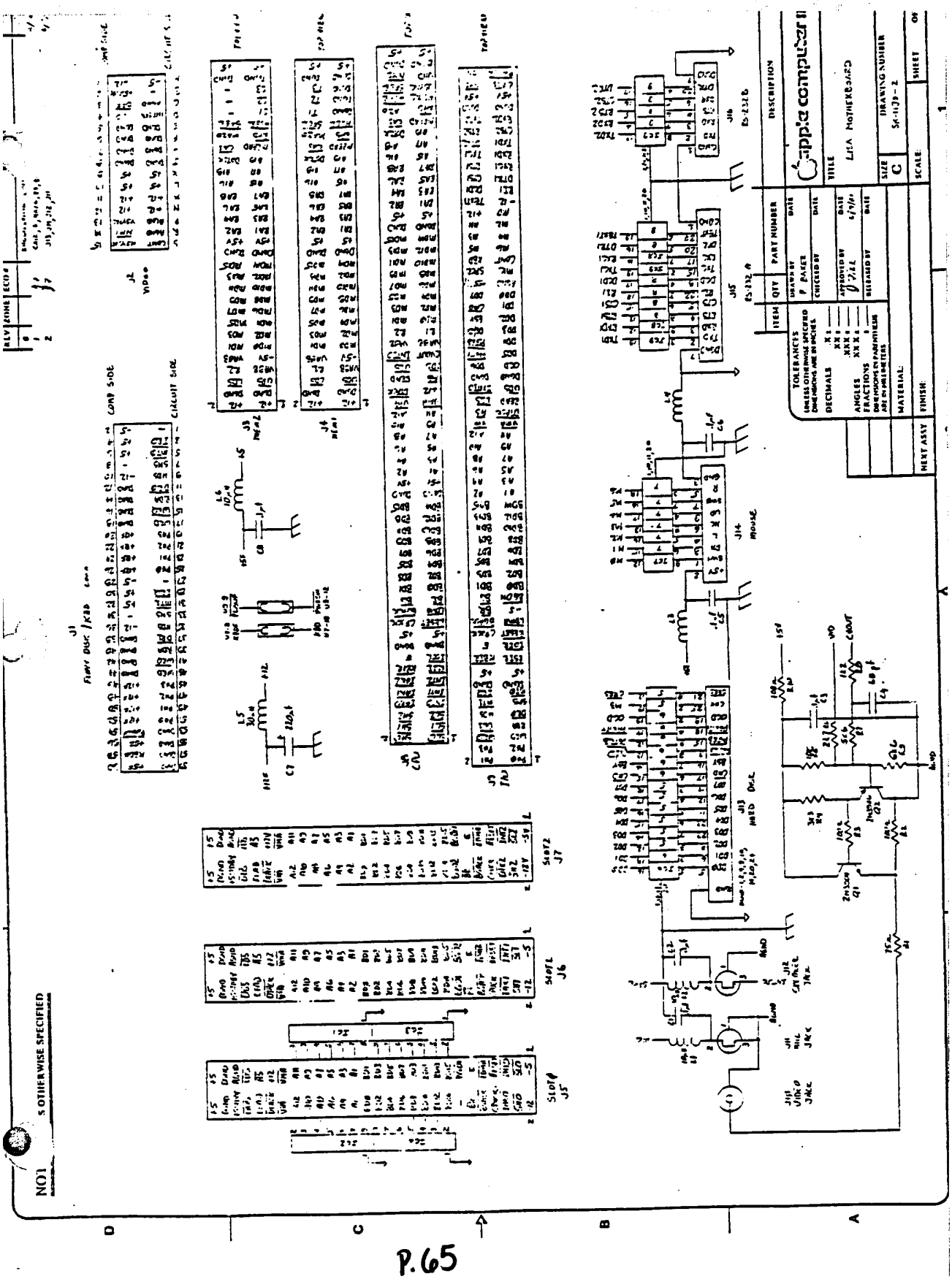
↳ MALE: Swap MSbyte & LSbyte of a word when accessing sequential bytes (a la 8080).

| | |
|---|---|
| Modem | Modulator/Demodulator. To send digital data over a telephone line, the Modem transforms the bit patterns into a stream of modulations of a carrier signal. At the receiving end, the signal is demodulated to recover the data. |
| MMU | Memory Management Unit |
| ms | millisecond. |
| N-key rollover | Nearly all keyboard interfaces work by scanning the keys and forming a two dimensional matrix representation of the state of the keys. The logic involved can tell when two keys are being pressed simultaneously (rollover), but when three or more keys are held down, phantom keys can appear (there are hidden paths through the logic array). N-key rollover design adds a diode in series with every key switch to eliminate the hidden paths. |
| Nibble | A set of bits smaller than a byte. On the Lisa, a nibble is 4 bits. |
| NMI | Non-Maskable Interrupt. When the processor receives an interrupt, it usually checks a mask to see whether it should pass control to that interrupt's handler. A non-maskable interrupt is always honored. |
| ns | nanosecond. A billionth of a second. Also abbreviated as nsec. |
| Page | 512 bytes |
| Parameter memory | A non-volatile block of RAM set aside for such things as the system serial number, configuration data, and user-defined information. |
| Parity | Parity has to do with the number of one bits in a word. If there are an even number, the word has even parity. Parity can be checked to insure that one bit in a word has not been incorrectly flipped during transmission. |
| PC | Program Counter |
| PIA | Peripheral Interface Adapter (6522 on the Lisa) |
| PIPPIN | A sealed 5 megabyte hard disk made by Apple |
| Port | An I/O location. |

| | |
|---|---|
| Privilege violation | The 68000 has two states of privilege, User and Supervisor. Certain instructions (RESET, for example) can occur only in supervisor state. An attempt to execute a priviliged instruction from user state causes a privilege violation interrupt to be generated. |
| PROM | Programmable Read Only Memory |
| RAM | Random Access Memory (actually ROM is also random access memory--RAM, however, is read/write memory). |
| ROM | Read Only Memory. |
| RS232-C | A communication protocol. |
| RWTS | Read/Write/Track/Sector. The controlling routines that drive the floppy disk controller both on the Lisa and the Apple II. |
| Segment | A segment is an independent address space. It may have little obvious relation to physical memory. |
| Shift register | A register is a device which can store information. A shift register is able to shift all its bits left or right. |
| SSP | Supervisor Stack Pointer. See Privilege Violation. |
| Synchronous modem | Synchronous transmission puts the framing information around a group of characters. The transmitter then automatically inserts fill characters into the stream whenever necessary to maintain synchronicity. Because more of the bits are data (there are fewer stop and start bits than in asynchronous transmission), data transfers can go at a faster rate. |
| tristate | A logic output that can be inactive, high or low. The three states are, therefore, active high, active low, and open. |
| us | micro-second. The letter 'u' is used because it looks like the small Greek letter mu (small Roman 'm' is used for milli, and large Roman 'M' is used for Mega). |
| Word | A group of bytes. On the Lisa, a word is usually 16 bits, or two bytes. Long Words are 32 bits or four bytes. |
| ZIF | Zero Insertion Force |

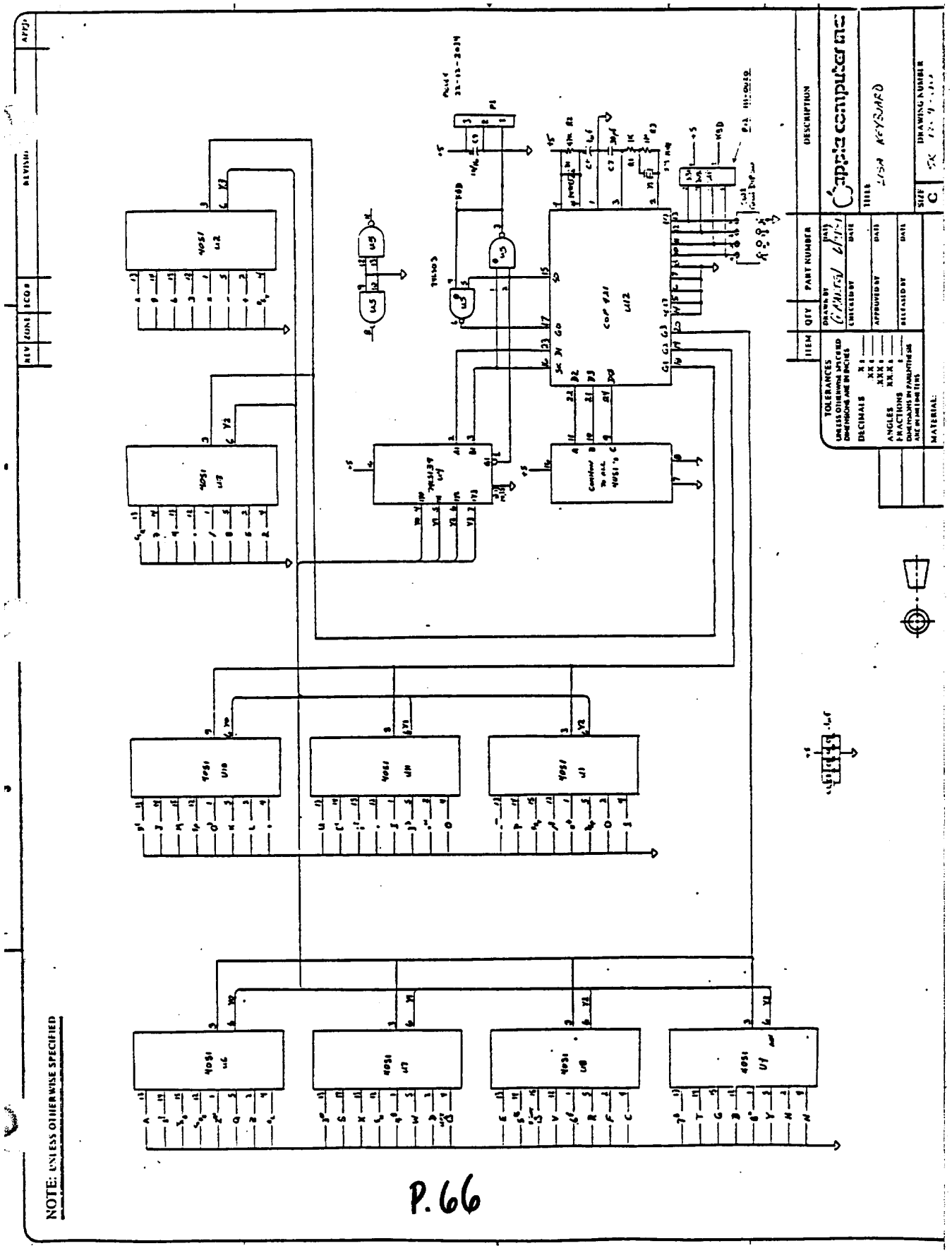
# APPLE LISA SCHEMATICS

{7-July-1981}



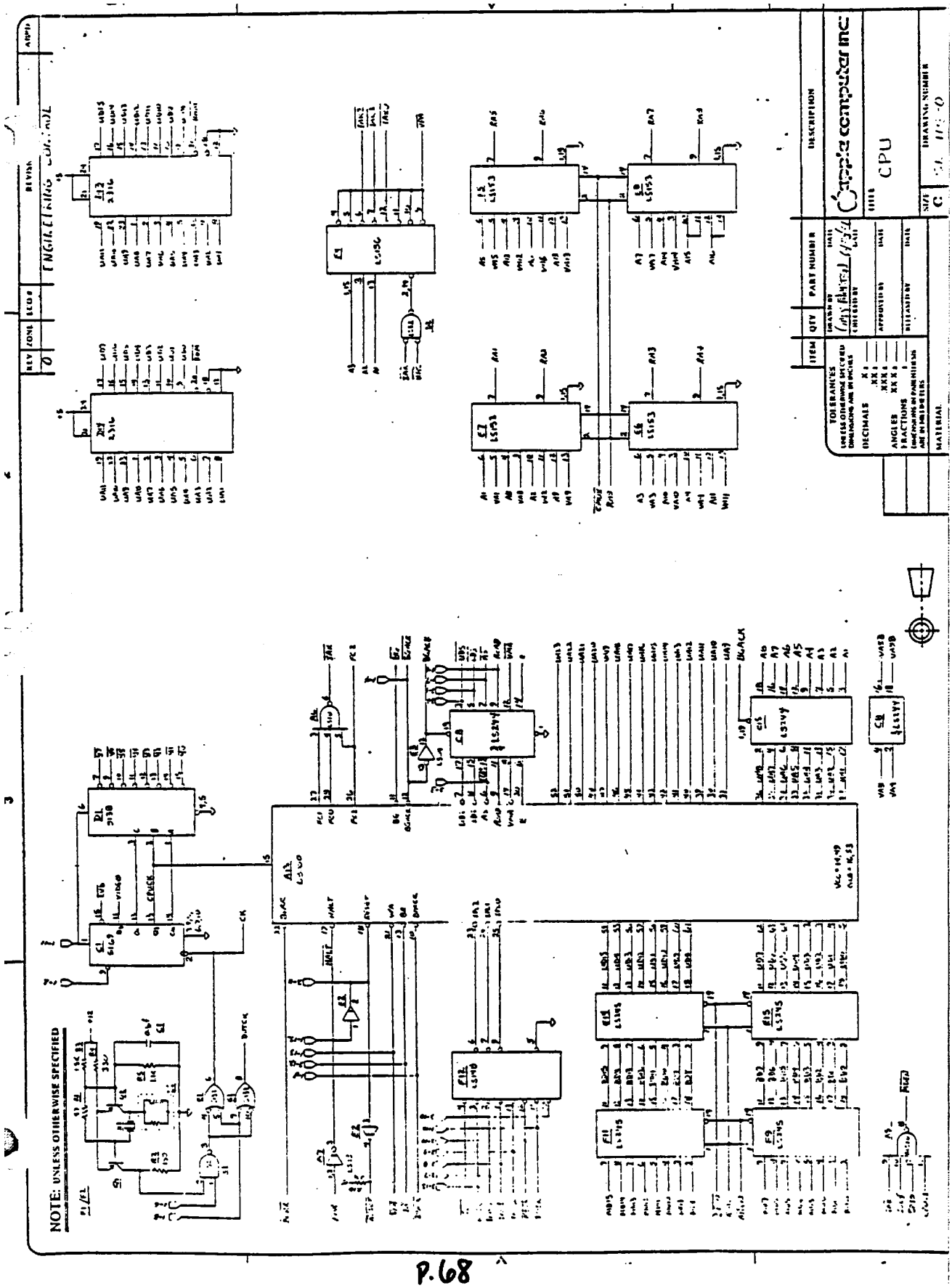
p.65

"LisaHWG81_065.PICT" 303 KB 1999-02-20 dpi: 72h x 72v pix: 2330h x 3151v



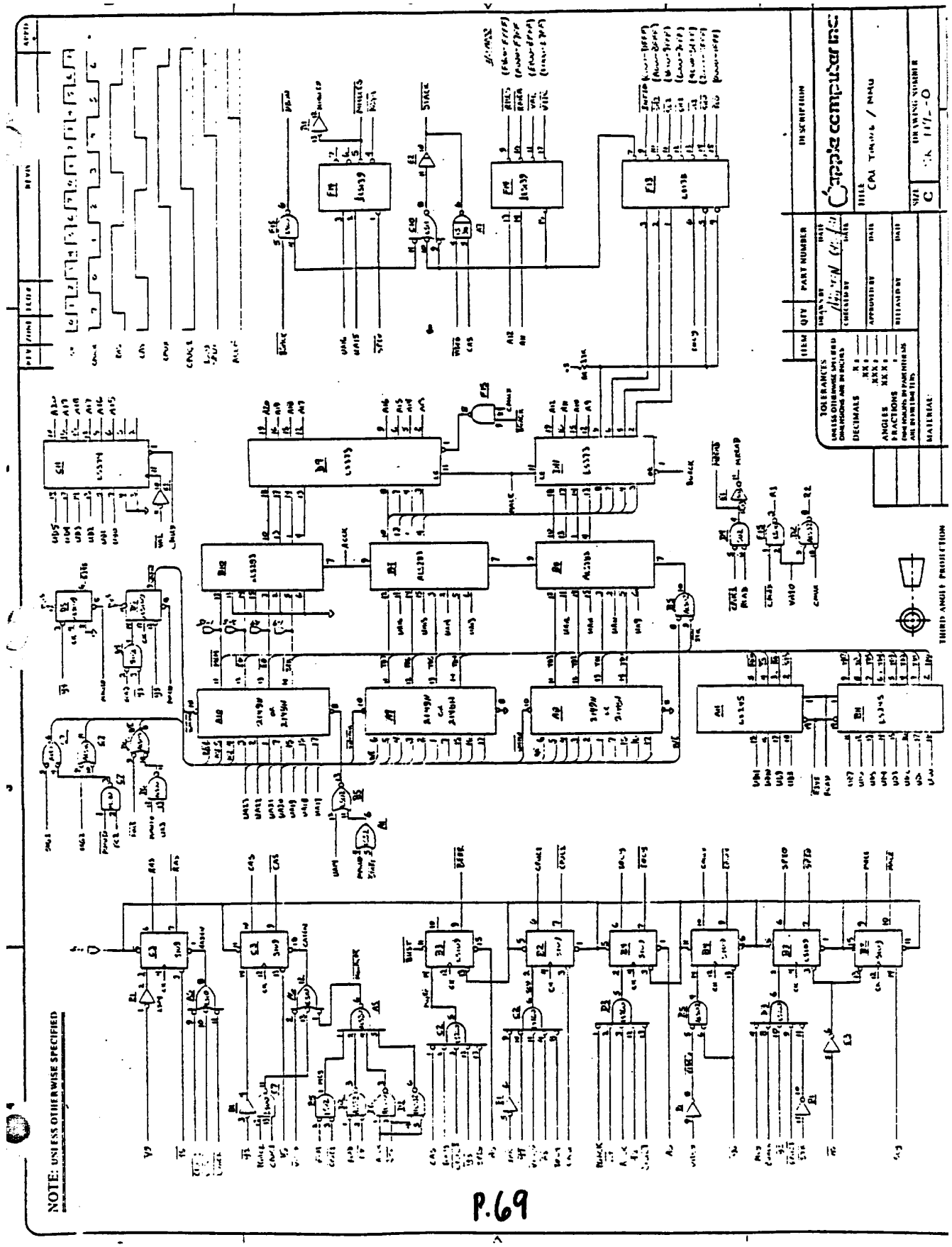

"LisaHWG81_066.PICT" 213 KB 1999-02-20 dpi: 72h x 72v pix: 2296h x 3015v

CPU BOARD

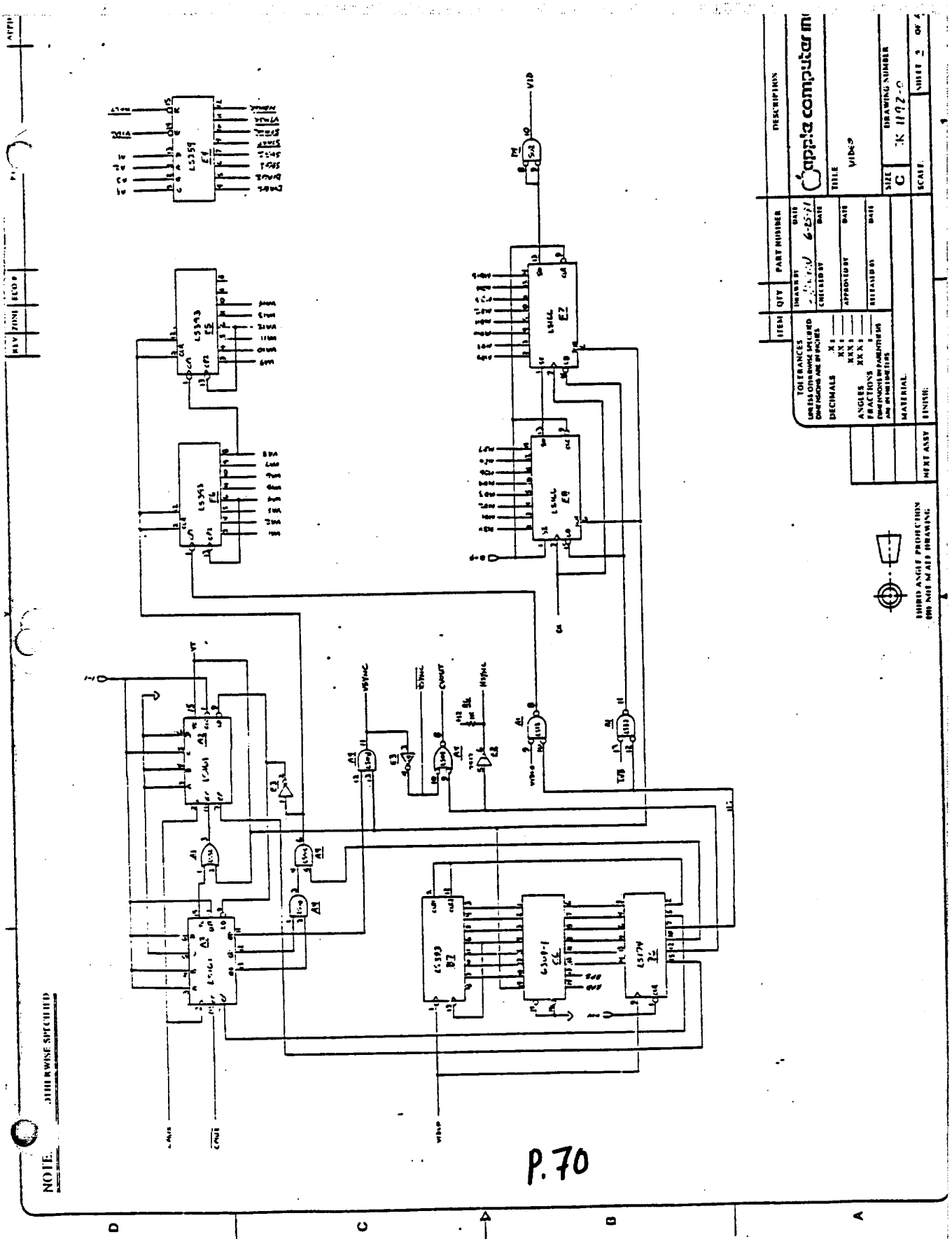



P.68

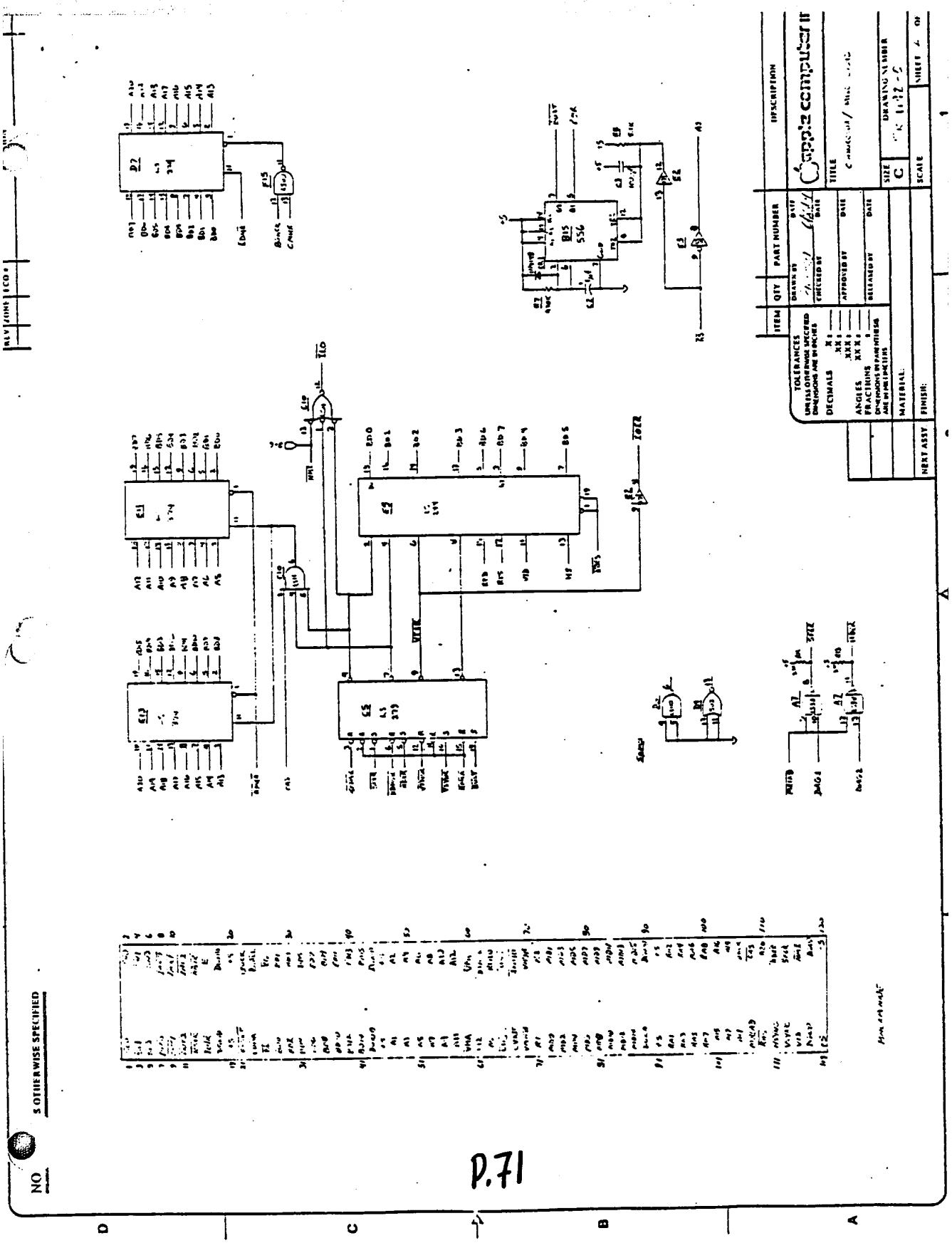
"LisaHWG81_068.PICT" 286 KB 1999-02-20 dpi: 72h x 72v pix: 2285h x 3080v



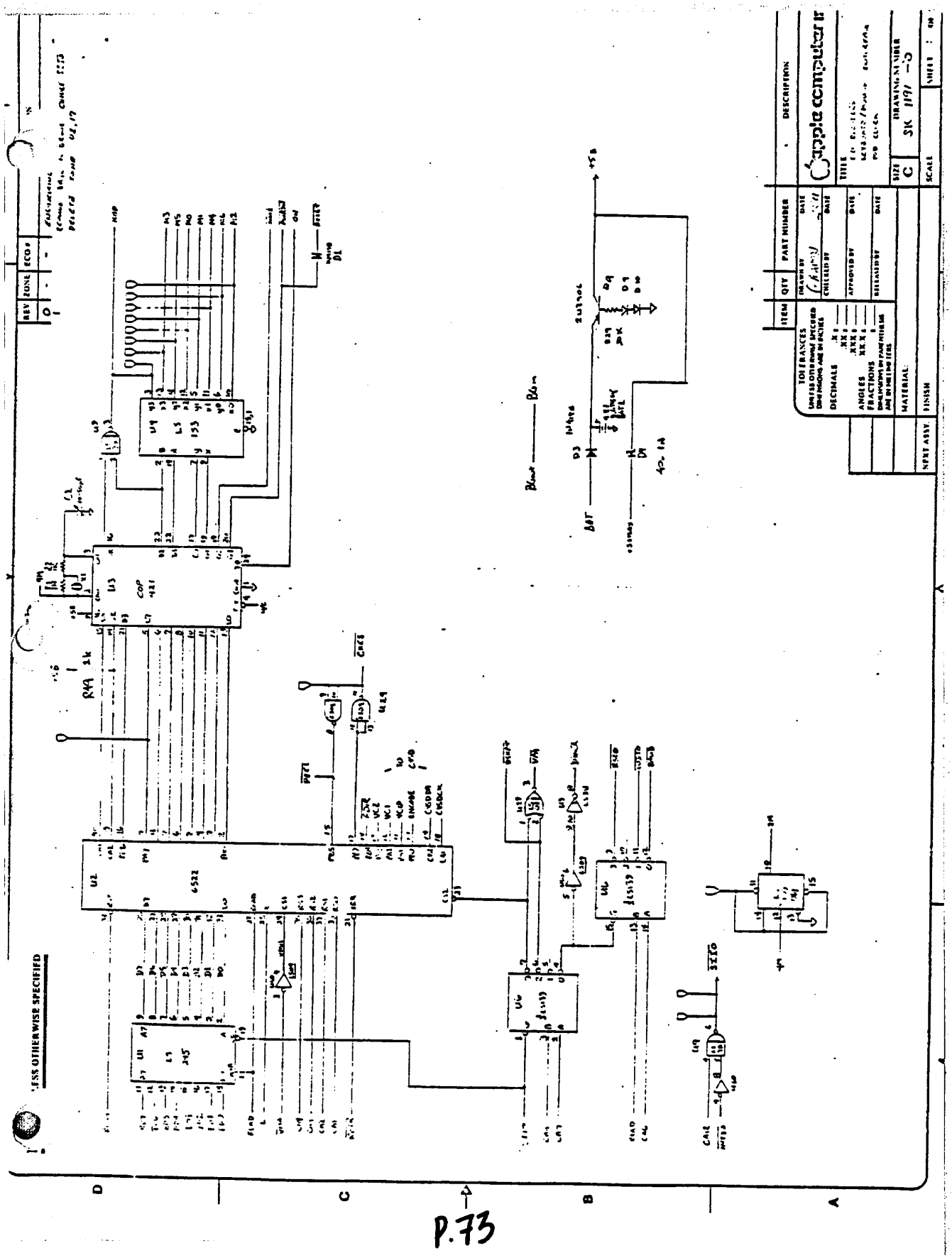

"LisaHWG81_069.PICT" 354 KB 1999-02-20 dpi: 72h x 72v pix: 2311h x 3059v



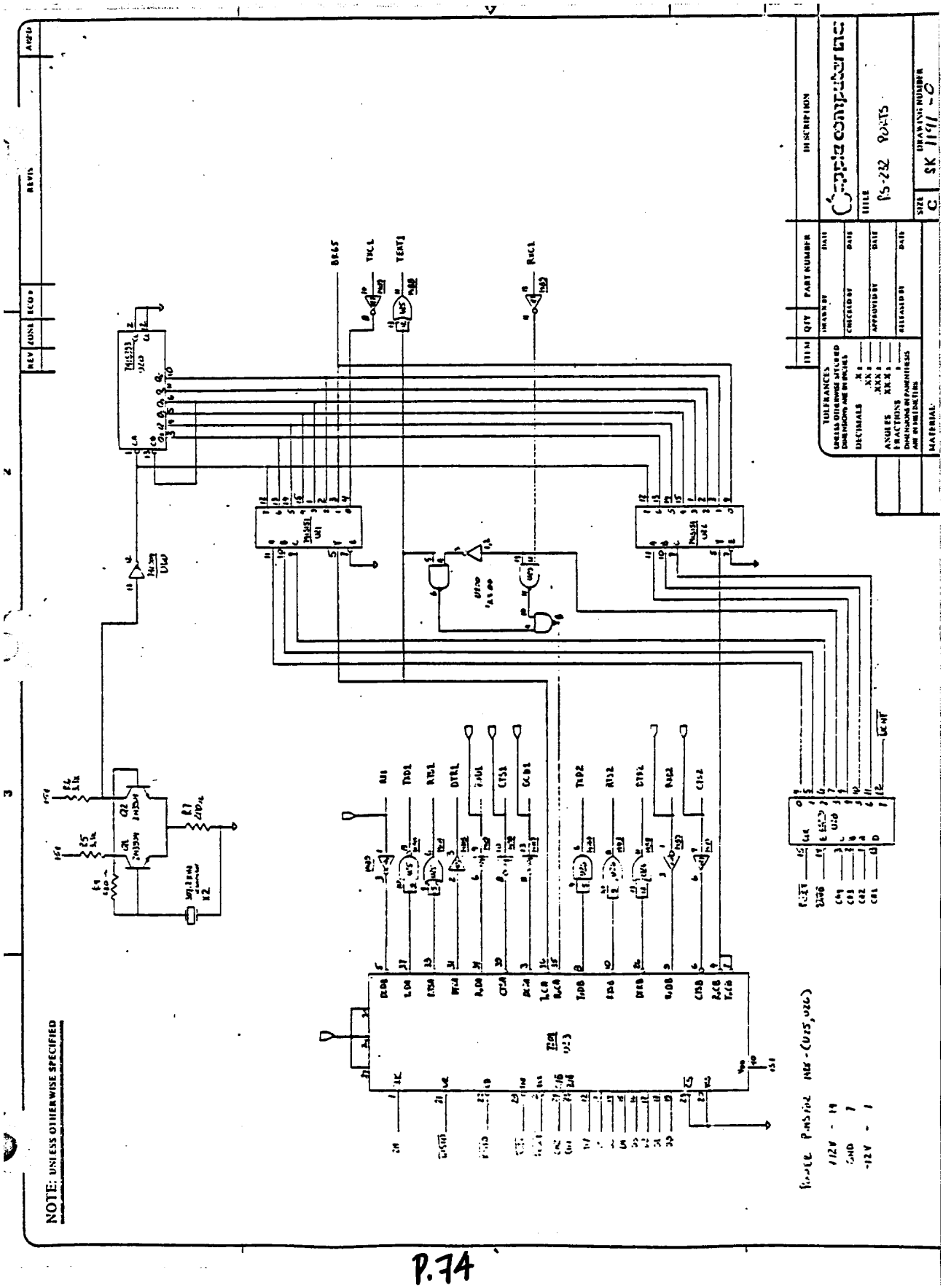

"LisaHWG81_070.PICT" 201 KB 1999-02-20 dpi: 72h x 72v pix: 2366h x 3049v

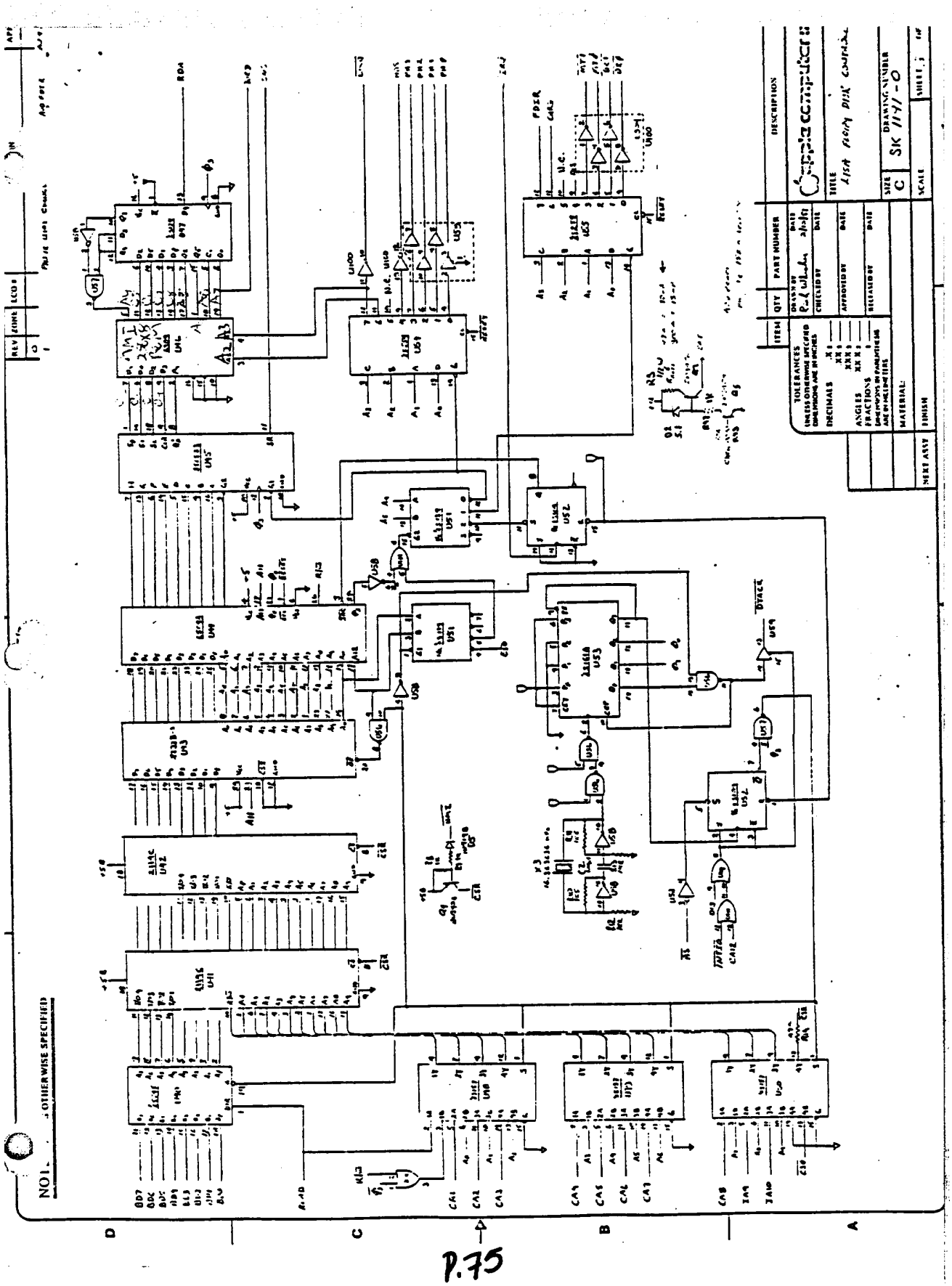

P.71

# I/O BOARD

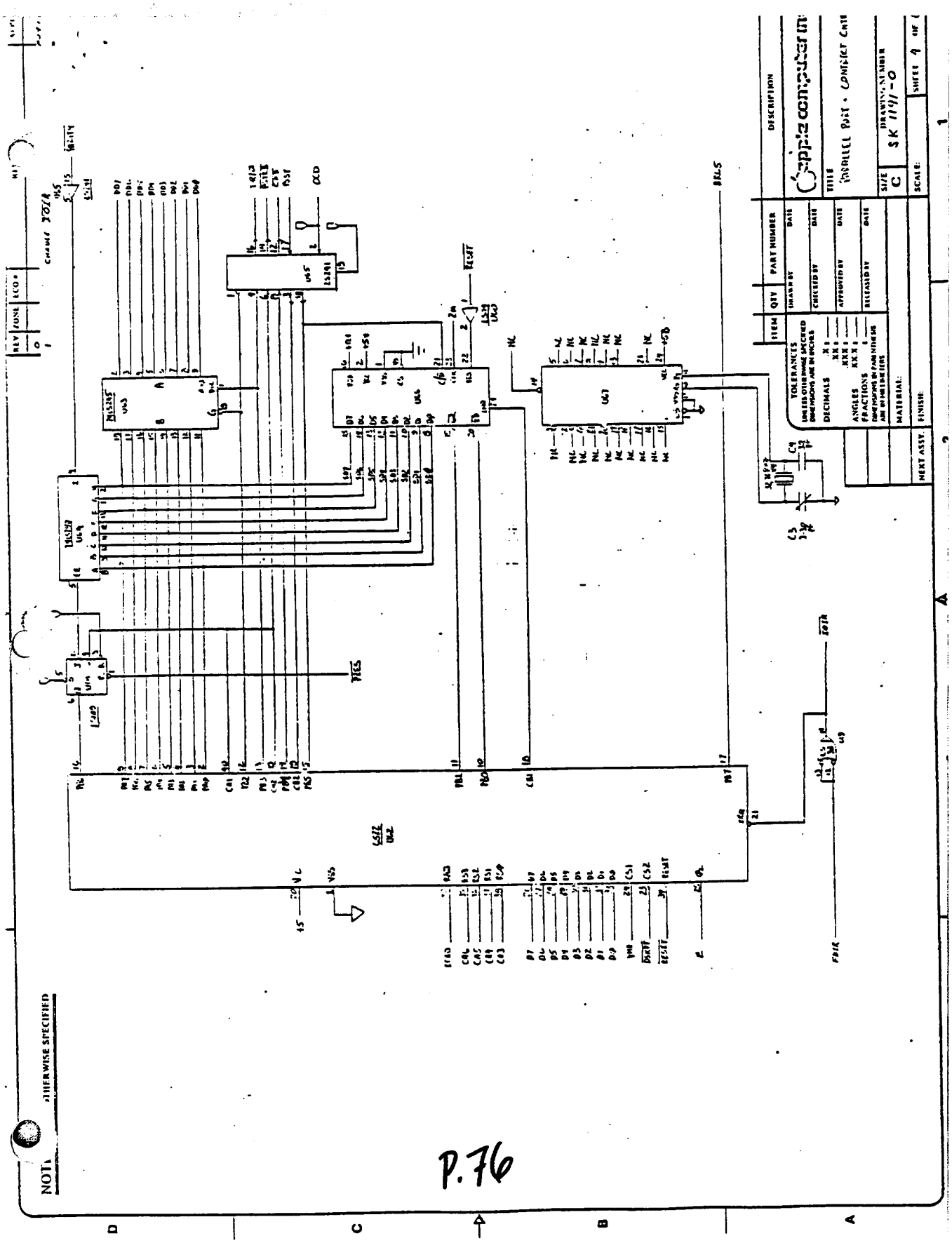



"LisaHWG81_073.PICT" 233 KB 1999-02-20 dpi: 72h x 72v pix: 2367h x 3115v

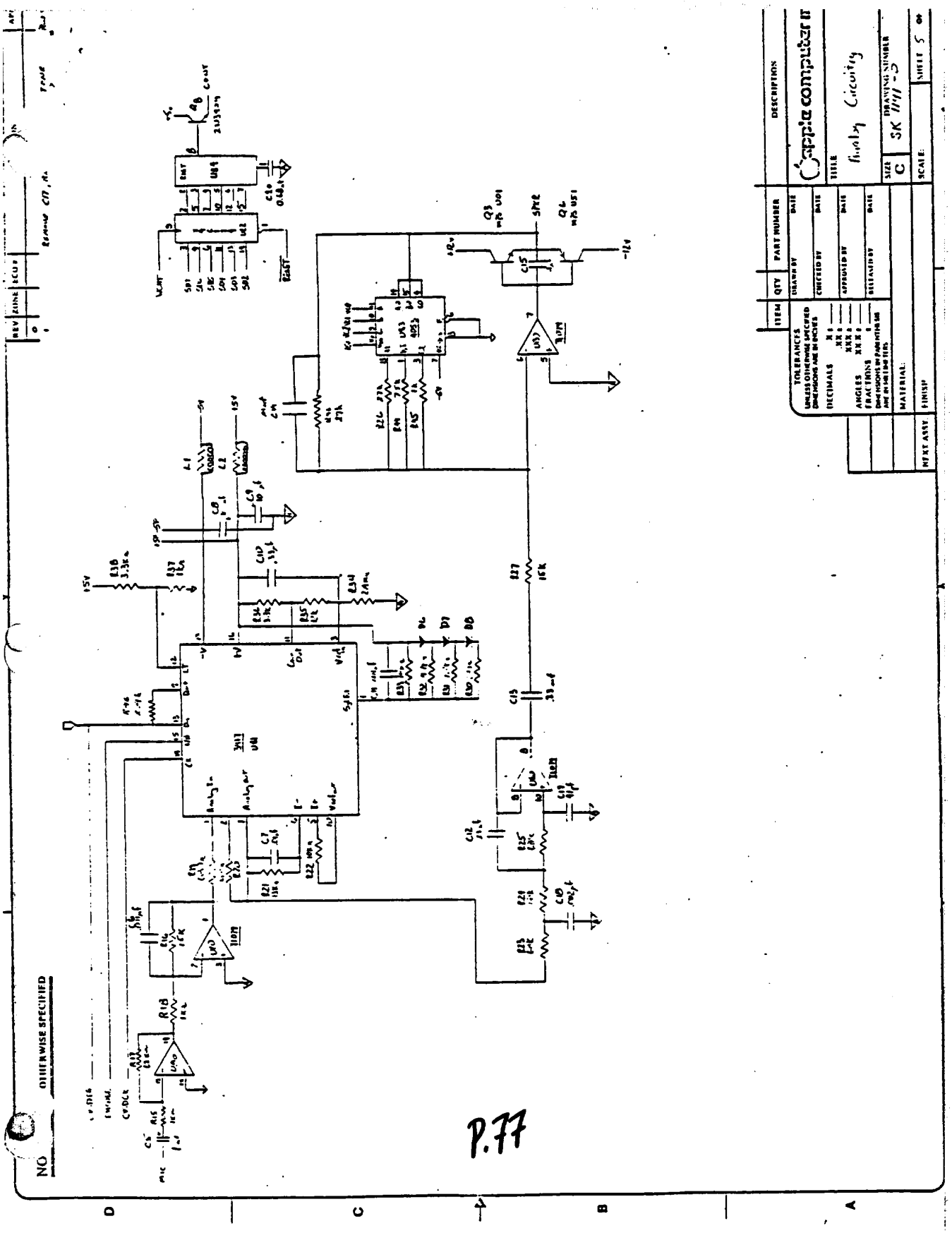

RS-232 PORTS
NOTE: UNLESS OTHERWISE SPECIFIED
P.74



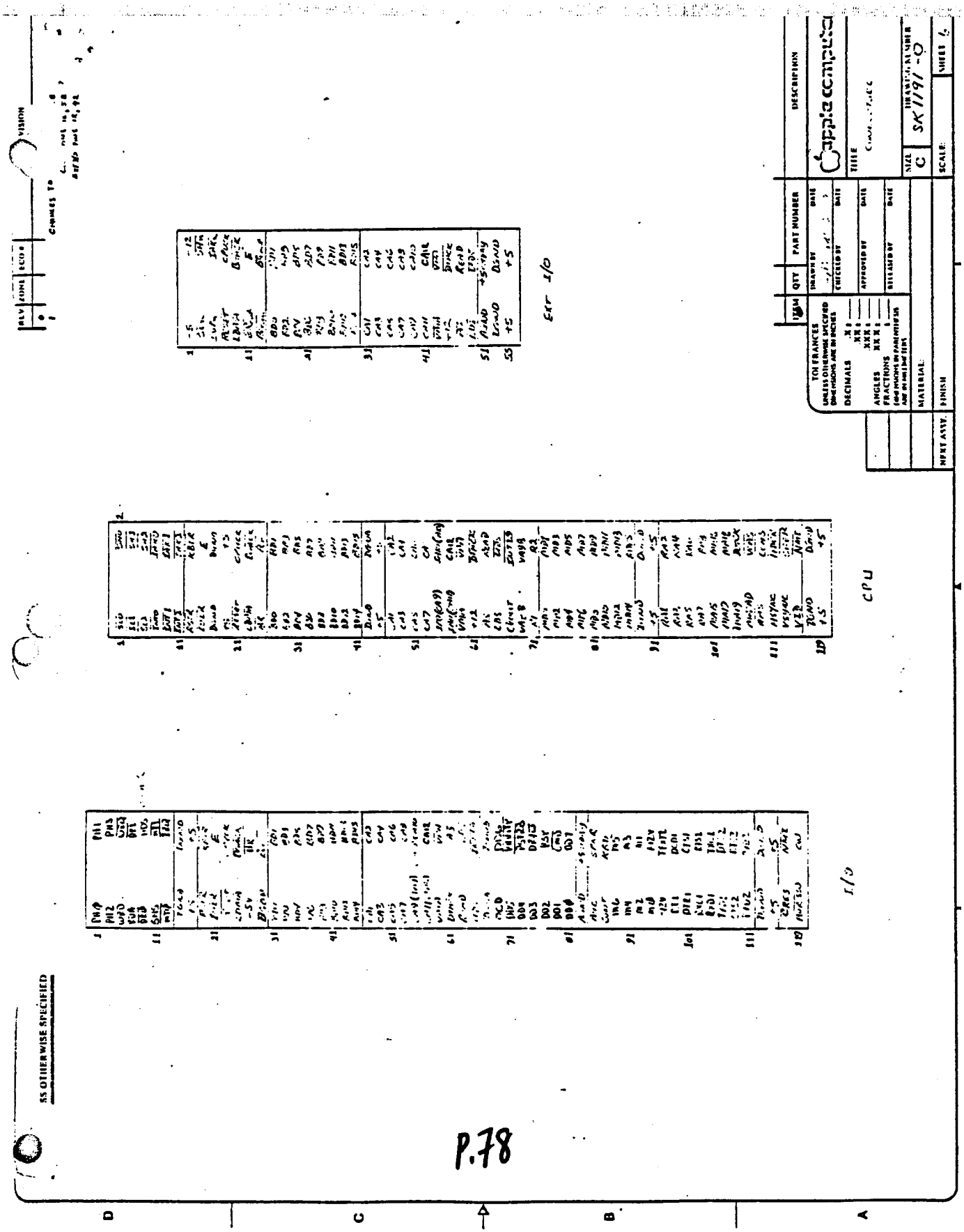

"LisaHWG81_075.PICT" 310 KB 1999-02-20 dpi: 72h x 72v pix: 2337h x 3154v

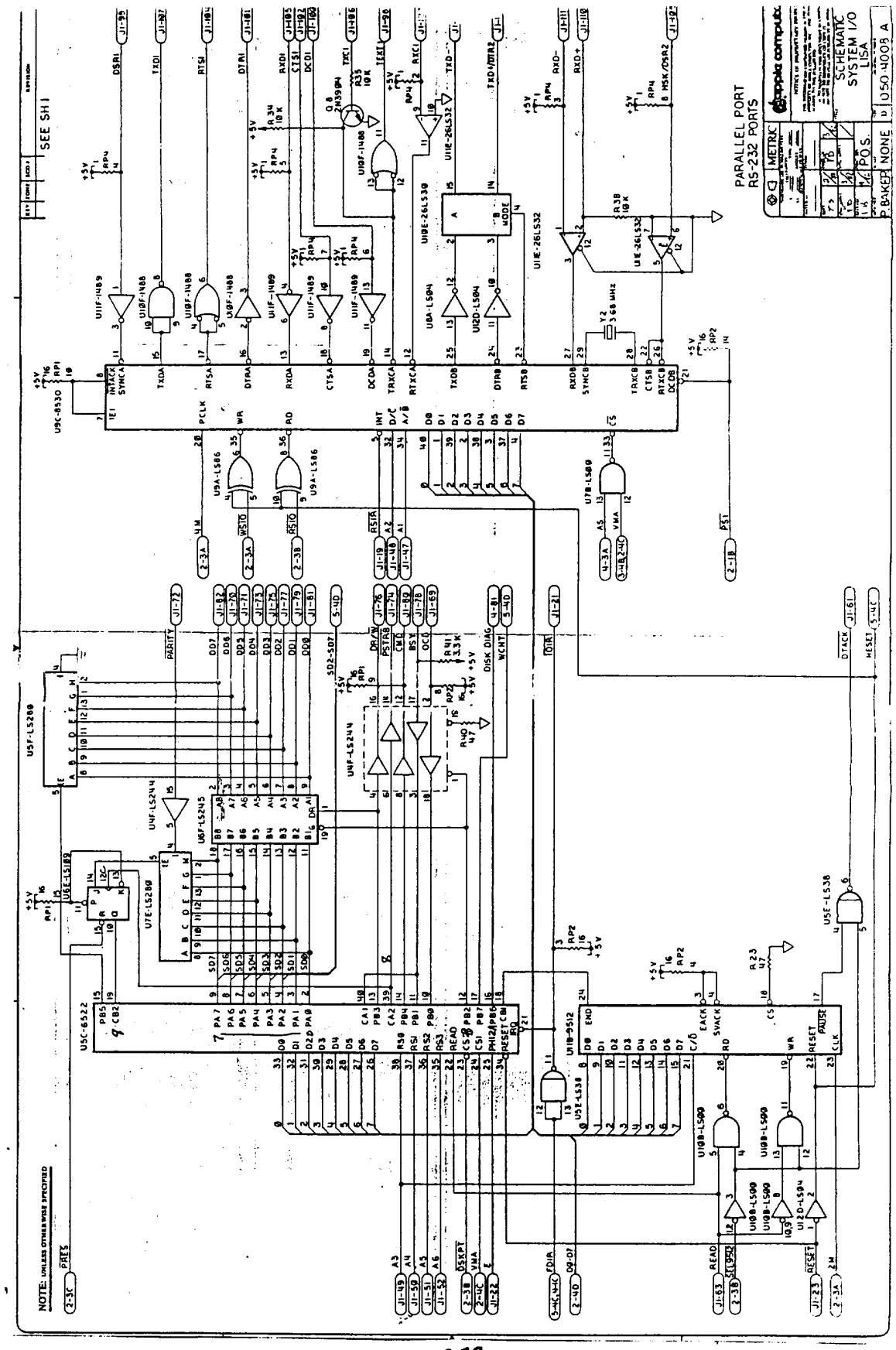



"LisaHWG81_076.PICT" 241 KB 1999-02-20 dpi: 72h x 72v pix: 2361h x 3045v



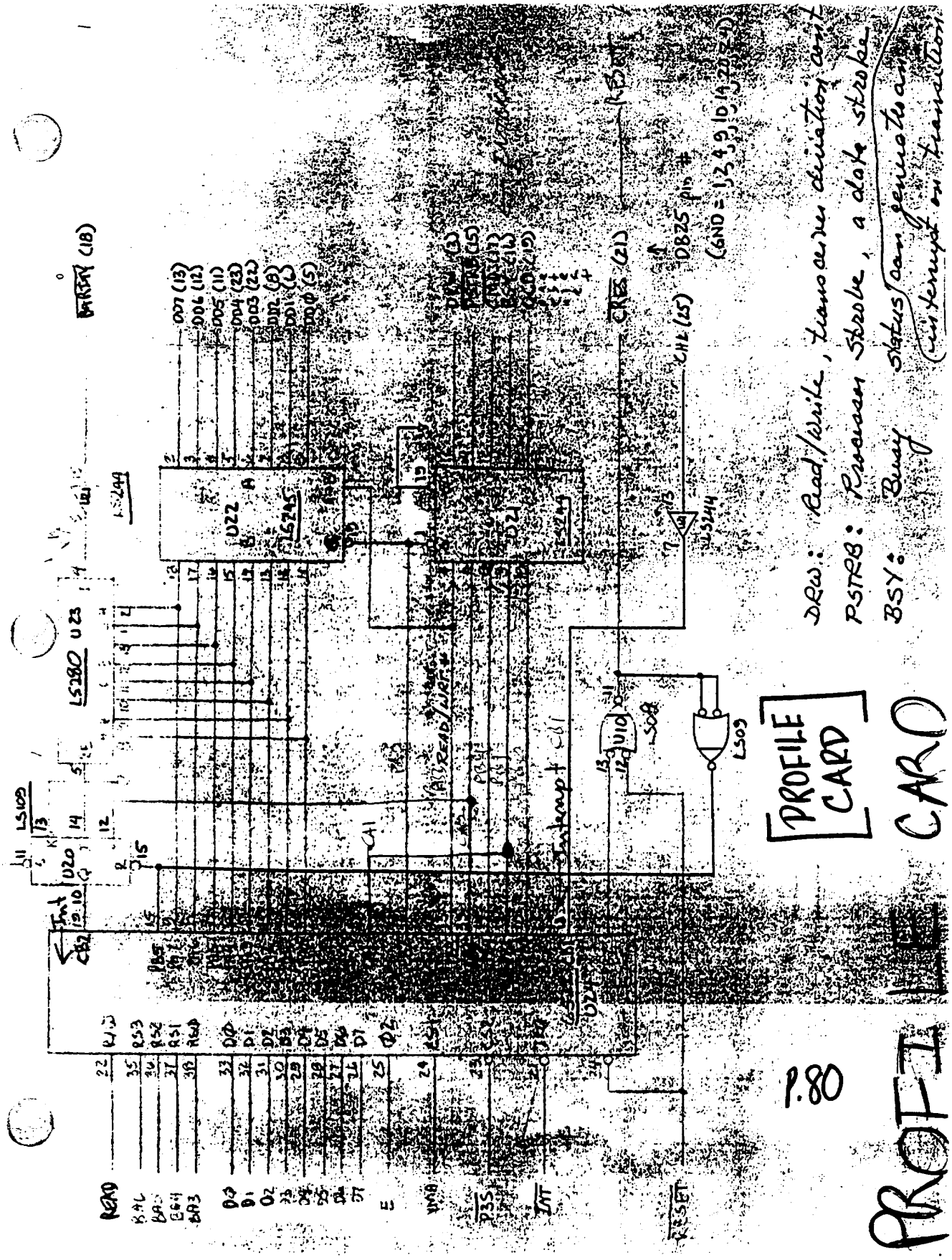

"LisaHWG81_077.PICT" 177 KB 1999-02-20 dpi: 72h x 72v pix: 2351h x 3072v


CPU
I/O
Apple computer
SK1191-0
SIZE C
SCALE:
SHEET 1
UNLESS OTHERWISE SPECIFIED
DECIMALS
ANGLES
FRACTIONS
MATERIAL
FINISH
NEXT ASSY.
DESCRIPTION
DRAWN BY
CHECKED BY
APPROVED BY
RELEASED BY
DATE
QTY
PART NUMBER
ITEM
REVISIONS
ZONE
ECO #
VISION
SS OTHERWISE SPECIFIED
P.78

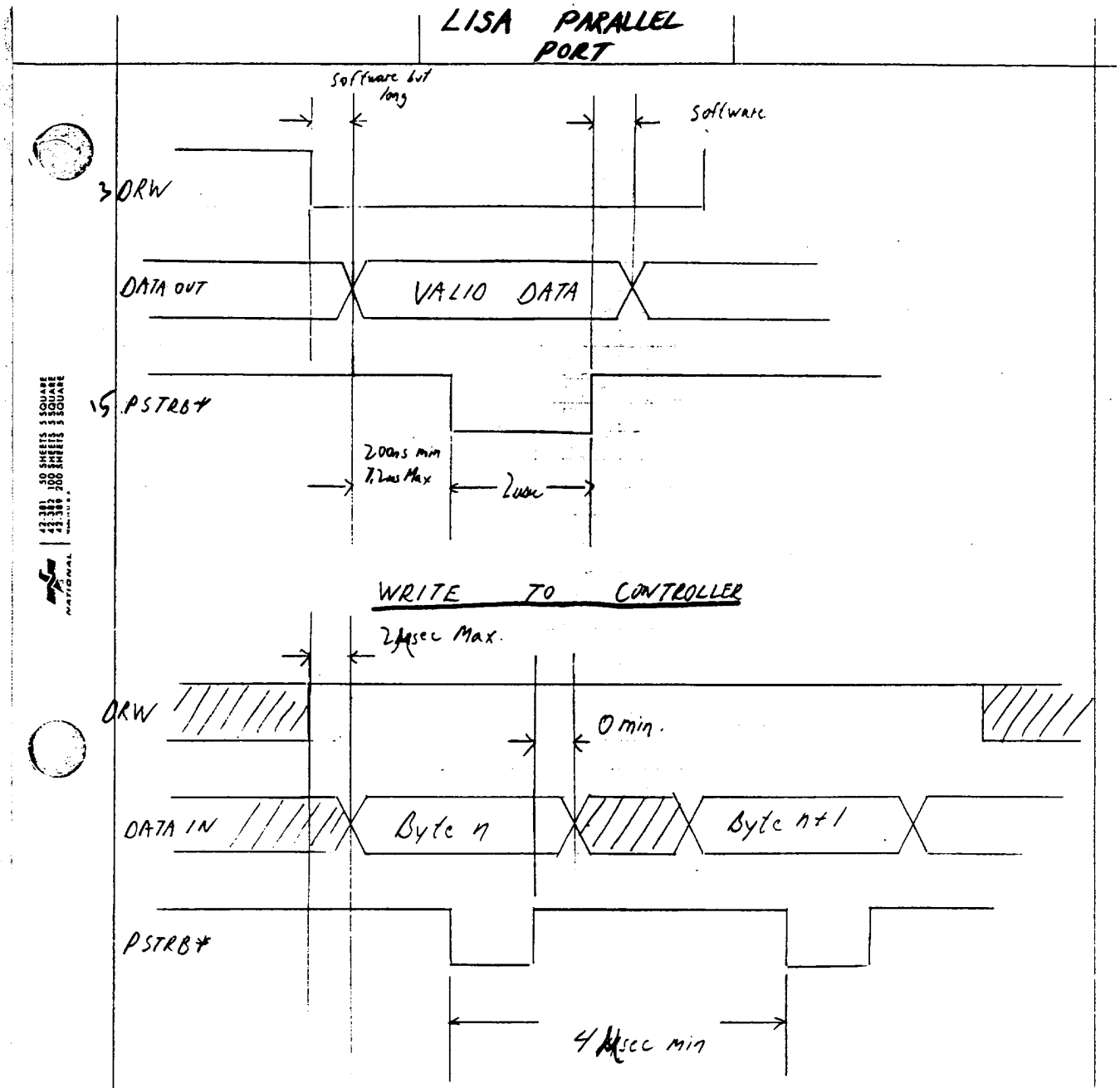



P.79

"LisaHWG81_079.PICT" 332 KB 1999-02-20 dpi: 72h x 72v pix: 2044h x 3069v




"LisaHWG81_080.PICT" 545 KB 1999-02-20 dpi: 72h x 72v pix: 2344h x 3115v




READ FROM CONTROLLER

# LISA PARALLEL PORT

P.81 ≡Finis≡

al 8/5

"LisaHWG81_081.PICT" 134 KB 1999-02-20 dpi: 72h x 72v pix: 2295h x 2993v