E22/3-4.2

A2090A
HARD DISK CONTROLLER
TECH DATA

OCTOBER, 1988 **PN-314880-01**

Commodore Business Machines, Inc.
1200 Wilson Drive, West Chester, Pennsylvania 19380 U.S.A.

TABLE OF CONTENTS

Amiga Hard Disk/SCSI Controller (A2090A) Technical Reference Manual

1.0 Description

The Amiga Hard Disk/SCSI Controller is an intelligent high performance controller designed to interface both ST506 hard disk drives and SCSI devices to the Amiga expansion bus architecture. A background command processor provides high level command interpretation minimizing Host intervention. Data is transferred to and from the Host via DMA (direct memory access) with FIFO allowing high data throughput while maintaining reasonable bus bandwidth for other bus controllers.

1.1 Features

- Auto-boot from hard disk devices
- Support for up to two ST506 hard disk drives
- Full SCSI with MacIntosh Plus compatibility
- High level command interpretation and exceptional handling performed by Z80 processor
- Support for up to 8 heads, 2048 cylinder with 512 bytes/sector
- Individually Programmable Drive Characteristics
- 1:1 sector interleave
- 32 bit ECC for data correction
- Multiple block transfers
- Full auto-config compatibility
- Real time data transfer rates of up to 800ns/byte via DMA

2.0 Specifications

2.1 Performance

Hard Disk (ST506)

Encoding method:	MFM
Cylinder per head:	Up to 2048
Sectors per track:	Up to 17
Sector length:	512
Heads:	8
Drive Selects:	2
Step Rate:	3.2 us to 6.5 ms
Data Transfer Rate:	5.0 Mbit/sec.
Write Precomp Time:	12 nanosec.
Sector Interleave:	1:1
Sector Interleave Across Heads:	1:2
Ecc Polynomial:	32 bits
Burst Error Correction:	11 bits

SCSI

ANSI X3T9.2 compatible
MacIntosh Plus compatible connector

Host Interface

Amiga expansion bus compatible
Full auto-config compatibility

2.2 Power Requirements

+5 Volts ±5%, 3 Amps. Max.

2.3 Environmental

Ambient Temperature:	0 — 55 Deg. C.
Relative Humidity:	20% — 80%

2.4 Connector Pin Assignments

Table 2.1 through Table 2.3 list the pin assignments for the controller board. For pin out and definition for card edge connector refer to Amiga expansion architecture manual.

Table 2.1 — Connectors J1 and J2 — Disk Serial Data Pin Assignments

Ground Return	Signal Pin	Signal Name
2	1	Drive Selected
4	3	Reserved
6	5	Write Protected (J1 Only)
8	7	Reserved
10	9	Cartridge Changed (J1 Only)
12	11	Ground (GND)
	13	MFM Write Data+
	14	MFM Write Data–
16	15	Ground (GND)
	17	MFM Read Data+
	18	MFM Read Data–
20	19	Ground (GND)

Table 2.2 — Connector J0 — Disk Control Signal Pin Assignments

Ground Return	Signal Pin	Signal Name
1	2	Head Select 3
3	4	Head Select 2
5	6	Write Gate
7	8	Seek Complete
9	10	Track 00
11	12	Write Fault
13	14	Head Select 2
15	16	Reserved
17	18	Head Select 1
19	20	Index
21	22	Ready
23	24	Step
25	26	Drive Select 1
27	28	Drive Select 2
29	30	Reserved
31	32	Reserved
33	34	Direction In

Table 2.3 — Connector CN1, SCSI

SCSI Connector (DB-25) Female			
Pin	Name	Pin	Name
1	REQ	14	GROUND
2	MSG	15	C/D
3	I/O	16	GROUND
4	RST	17	ATN
5	ACK	18	GROUND
6	BSY	19	SEL
7	GROUND	20	DBP
8	DBO	21	DB1
9	GROUND	22	DB2
10	DB3	23	DB4
11	DB5	24	GROUND
12	DB6	25	N.C.
13	DB7		

2.0 Reference

— 8727 DMA Specification
— Commodore Amiga A500/A2000 Technical Reference Manual
— Motorola 68000 Technical Manual
— Western Digital WD33C93 SCSI Chip Manual
— American National Standard Committee X3T9.2 SCSI Specification

3.0 Functional Description

The Amiga Hard Disk Controller basically consists of three main subsections:

1. Host Interface
2. ST506 Hard Disk Controller (HDC)
3. SCSI Controller

3.1 Host Interface

The host interface is 68000 compatible with direct memory access and full auto-config capability. Data transfers to and from the host are usually made via DMA thereby allowing real time date transfer rates of 1.6us/byte for the ST506 interface and up to 800ns/byte for SCSI. Addressing for DMA operations is provided by three external address counters. Before any DMA operation can be performed each counter must be pre-set and thereafter will be incremented automatically. For information on initializing the DMA see section 4.0.

The DMA is a Commodore custom LSI chip (8727) with byte to word funneling and a built in 64 byte FIFO. The internal 64 byte FIFO permits real time data transfer to and from the host without holding the bus for an entire sector transfer. This provides very effective utilization of the bus. The average bus requirement for the transfer of an entire sector is 8.9us once every 51.2us. This amounts to only 17% over for CPU and other bus masters.

The interface logic also provides full auto-config and all I/O decode.

For electrical specification and detailed timings refer to Commodore Amiga Technical Reference manual.

3.2 ST506 Hard Disk Controller (HDC)

The ST506 Hard Disk controller is an intelligent background controller capable of high level command interpretation and support of up two ST506 hard disk units. This controller will be referred to in this document as the HDC or the Hard Disk Controller.

The processor for the HDC is a Z80A CPU, with up to 8K of PROM for firmware and 1K of RAM for variable data. Collectively, the above components constitute the "intelligence" of the controller.

The design that has gone into this aspect of the controller has been to enhance performance and increase flexibility while reducing cost. As a result, the majority of operations have been placed in firmware. The only functions performed by "hardware" are those that are too fast for the processor.

The Z80A CPU and its associated PROM and RAM collectively perform the following functions:

1. Power up initialization
2. Diagnostics
3. Error recovery
4. Error reporting
5. Error correction
6. Command processor
7. Disk select
8. Seek
9. Write precomp select, reduced write current
10. Head select
11. Mapping
12. Logical to physical address translation — Physical to logical address translation

3.2.1 The DJC Custom Chip

The DJC is a custom LSI chip. It has been designed to handle all serial data, state machine and DMA functions as described below:

ERROR CORRECTION CODE

The error correction polynomial is a 32-bit code capable of correcting up to 11-bit burst errors.

In keeping with the overall design philosophy, the ECC circuitry generates the write syndrome and validates the read without requiring the processor to handle the data. Calculating this polynomial with the processor would seriously degrade the performance of the ST506 controller. Calculating the reverse polynomial to correct bad data is done by the processor. It is accomplished without any measurable effect on performance because the operation is only done after multiple retries and as such is seldom necessary.

HEADER VERIFICATION

Once a disk has been formatted, the DJC converts the desired record address on the disk. The conversion is done in terms of head, track and sector address, with a CRC code tested to further insure positional integrity. A comparison is then made of the header before a read or write function is performed.

TWO INDEX TIMEOUT

This function insures accurate control over the number of attempts to find a header (i.e., it is not "mislead" by counting false address marks).

MFM ENCODE

The DJC converts all parallel data to serial and then to MFM. This function is followed by Precomp, if selected.

3.2.2 Selectable Precomp

In Precomp, a "string" of pulses is analyzed to determine if they are arranged in the unique manner that could cause them to crowd once written on the disk. It also determines which way the crowding would distort the pulses when read. The write pulse stream is then shifted, early or late, to compensate for the crowding conditions, which normally occur on the innermost tracks of the drive.

Under the processor's control, the DJC precomps the disk MFM data by using external inductive delays. Precomp is selectable and is designed to shift the MFM data early or late by 12 nanoseconds to improve read margins.

The use of this feature should be performed in conjunction with the particular drive manufacturer's specification.

3.2.3 MFM Decode

Data received from a disk drive is MFM, a self-clocking serial data stream which contains a phase locked loop, lock detect, missing clock detect and the data seperator.

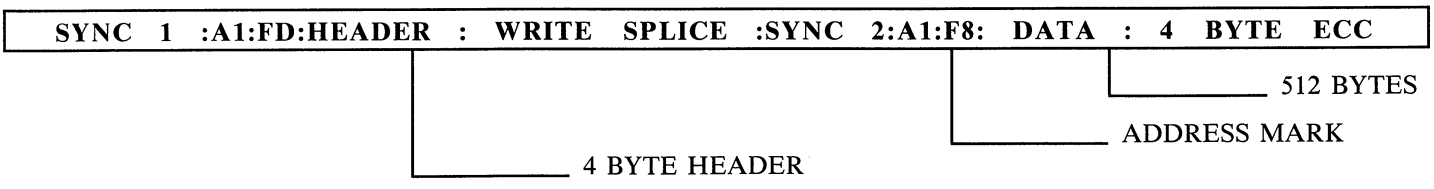
When the DJC asserts Read Gate, the 8465 data seperator will attempt to lock its phase locked loop on the read data. If this does not occur within 4.8 usec, the DJC will turn off Read Gate, causing the 8465 to be placed into the low track rate for increased stability.

The MFM data is now decoded into NRZ data and clock for the DJC. The 8465 decodes a missing clock bit and a hexidecimal A1, FD or an A1, F8 in the sync field. This data indicates the start of a valid header or data field. Receiving any other data causes the DJC to abort the read. Another read would be tried after resyncing the 8465 to 10 MHz.

3.2.4 Sector Format

Figure 3.2 describes the format of a typical sector.

Figure 3.2 — Typical Sector Format



BYTE 1 = HEAD #; BYTE 2 = TRACK ADDRESS; BYTE 3 = SECTOR #; BYTE 4 = CRC

Note: **1. Address Mark is a Hex 1 with a missing clock pulse.**
2. SYNC field 1 is comprised of 16 bytes of zeros.
3. SYNC field 2 is comprised of 15 bytes of zeros.

3.2.5 Error Recovery Philosophy

Extensive measures have been taken in the design of the controller to insure reliable data. Selectable precompensation circuitry and a sophisticated data seperator with two tracking rates are a few examples. Additional effort has been made to reduce the probability of miscorrection (of having bad data flagged as corrected) through design and options made available to the systems integrator.

In a write operation the controller only precomps the unique combinations of data that might cause crowding conditions on the disk. Shifting data early or late by 12 nsec is done to retain as much of the 50 nsec data window as is possible. This reduces the probability of errors occurring.

In a read operation the data seperator phase lock loop (PLL) provides two tracking rates, a high and a low, which allows for quick synchronization with the header address in the first case and stable data transfer in the second. The controller only contributes a maximum of 6 nsec (typically 3 nsec) of window error out of the allowable error window of 50 nsec. This allows the disk drive to have up to 44 nsec of jitter before error recovery/correction is needed.

The controller uses a 32-bit error correction code that enables an error correction span of up to 11 bits. This computer-generated code is considered superior to fire codes because it substantially reduces the chances of miscorrection while providing the full 11-bit correction span.

In data recovery and error correction the ECC syndrome must be stable in order to perform a correction. This insures that multiple attempts are made to recover marginal data before correction data is applied and further reduces the probability of miscorrection on long (greater than 12-bit) error bursts.

The significance of not correcting data unless the ECC syndrome is stable is that 1) noise induced errors are not corrected and 2) real errors are corrected quickly without wasting time on useless retries.

The user can improve data reliability by mapping tracks with flaws and by reducing the error correction span. The latter reduces the odds of miscorrection on large errors (greater than 12 bits) and provides for early detection of a degrading media. The controller can be programmed to report or not report "soft" errors, on reads that took multiple tries but did not need correction.

Monitoring soft errors is probably the best method of early detection. A correction span of seven (7) bits is thereby suggested as an optimum in data integrity. An alternate eleven (11) bit correction span could be used as a means to retrieve the data before the track is mapped.

3.3 SCSI Controller

The SCSI controller uses the Western Digital WD33C93-SBIC which provides the actual interface to the SCSI connector and supports the full SCSI protocol minimizing host responsibilities. The WD33C93 is supported with a flexible architecture allowing either the 68000 (host) or the Z80A (board processor) to control the WD33C93 operations. Data transfer can be done via DMA or host I/O. For detailed information refer to Western Digital WD33C93 manual.

4.0 I/O Definitions

The following I/O addresses refer only to offset location since the actual board location in physical memory is configurable as described in the Commodore Amiga Technical Reference manual. Refer to this manual for details on auto-config I/O descriptions. I/O locations 0 hex through 42 hex are written out as nybbles or 4 data bits (AD12-AD15). I/O addresses 50H - 68H are unique to this board and will be described in this document.

Hex Location	Definitions
00/02	Boardtype and size
04/06	Product number
10/12	
14/16	Mfg # high and low byte
28/2A	Optional ROM vector high byte
2C/2E	Optional ROM vector low byte

40/42	WRITE	READ
15 / 14 / 13 / 12	15 / 14 / 13 / 12	
	Interrupt enable	Interrupt enable
	*SSEL	DON'T CARE
	MRESET	MUST BE ZERO
	*HCBP bit	*CCBP bit
	not defined	INT2 PENDING
	not defined	ZERO
	not defined	ZERO
	not defined	INT FOLLOW

*Signals unique to Amiga Hard Disk/SCSI Controller.

SSEL	Used to select SCSI controller or to ST605 controller. High = SCSI, low = ST506.
HCBP,CCBP	Host command block pointer and Controller command block pointer. Used to handshake address of Command block pointer to ST506.
48H	Base address register.

4.1 I/O addresses unique to board

50H	WRCBP/INTACK — Multiplexed signal. WRCBP strobes the command block pointer register. INTACK clear INTP at end of command.
52H	PROCC — Interrupt ST506 controller to process command. Write only. Data value written from host is XXX1 hex to initiate command execution and a data value of xxx0 hex is written from host to clear pending interrupt from 8727 (DMA).

4.1.1 SCSI Controller

60H	CS — Chip select for the WD33C93 SCSI chip. Used to write to the internal address register and read from the internal status register.
62H	CS — Chip select for the WD33C93 SCSI chip. Used to write and read remaining Control registers in the WD33C93.
64H	SCSI PCSS — Used to initialize the 8727 (DMA) in SCSI mode. Refer to section 5.0 for 8727 commands.
68H	SCSI PCSD — Used to pass data to and from the 8727 in SCSI mode. Refer to section 5.0 for transfer procedures.

4.1.2 Auto-Boot ROMS

8000H-FFFFH	To support AUTO-BOOT from hard disk the A2090A has two 28 sockets for auto-boot ROMS. Two 8 bit wide ROMS are used to support the Amiga's 16 bit bus. ROMS can be either 2764 or 27128 and begin at the base address offset of 8000H.

5.0 Host Interface Protocol

5.1 Interface Protocol

The host interface is via a DMA controller. This DMA device is controlled by the Z80A on the disk controller board or 68000 (host). On the host side there are counters for the address bus that are preset before the beginning of each transfer. Three bytes must be written for the 23 address lines (A23-A1). The MSB (corresponding to A24) of the upper address latch is used to control the host R/W- line for DMA transfers. This line is set high to read from the host memory and low if a write is intended. The DMA logic, contained in one chip, can be configured to transfer a single word (2 bytes) or 256 words (512 bytes). Transfer are always on even byte boundaries.

The method of communicating to the DMA circuit is by two control lines PCSS- and PCSD-, controlled by the Z80 or 68000. PCSS- is always strobed first to strobe in the "state" on the data bus. The state will determine the function to be performed on the succeeding PCSD- strobes. Not all valid states need to be followed by a PSCD- strobe and for each state loaded, PCSD- can be strobed any number of times. When reading the host status for instance, the expected number of PCSD- strobes need not be given, but when writing to the DMA controller the correct number of PCSD- strobes must always be given.

5.2 DMA Commands

The valid commands, for DMA operations, are summarized in Table 5-1 on the following page. All data values are listed in hex. Multiple states can be strobed into the DMA controller as long as no bus contention occurs. Notice that the state bits 4-0 are low in one position only for all the valid states. This implies that any state that does not require transfer of data by the following PCSD- can be combined and set simultaneously. Hence a single word transfer and start DMA cycle can be combined as DE. Some states are mutually exclusive such as F7 (transfer data to or from the FIFO) and EF (reading the DMA status). Similarly state D6 is illegal since word transfer and the FIFO path open will result in BUS contention. State FC is permitted as long as the same data is to be written in the DMA mid address latch and DMA low address counter. Other such valid states can be similarly derived.

Table 5-1: DMA States

Data Strobed by PCSS-	Brief Functional DESCRIPTION	Data Valid PCSD- (R/F)
FB 1111 1011	Load upper DMA address latch	F
FD 1111 1101	Load mid DMA address latch	F
FE 1111 1110	Load low DMA address latch; start DMA on rising edge of LDO; block mode XFER	F
F7 1111 1111	Open path to int. DMA FIFO (64 bytes)	R
EF 1110 1111	Read internal DMA status DB7 = 1 if no DMA or DMA cycle complete DB6 = 1 if byte avail. from or to FIFO DB5 = 1 if no FIFO overflow or underflow	R
9F 1001 1111	Force IREQ- to high impedance	X
BF 1011 1111	Command complete signal to host	X
DF 1101 1111	Set DMA into a single word transfer	X
7F 0111 1111 FF 1111 1111	Reset DMA and clear FIFO followed by FF to ensure proper DMA reset.	X X

5.2.1 Load Upper DMA Address Counter (FB)

The LD2 output of the DMA chip is set low on the rising edge of PCSS- and then set high on the falling edge of PCSD-. This loads the R/W- and the upper 7 address lines A23-A17 from the data bus into a counter on the rising edge of LD2. This 8 bit counter need not be reloaded if its contents are to remain unaltered in the succeeding operations.

5.2.2 Load Mid DMA Address Counter (FD)

Address lines A16-A9 are loaded into another counter in the same manner as above by the rising edge of LD1. This 8 bit counter also need not be reloaded if its contents are to remain unaltered in the succeeding operations.

5.2.3 Load Low DMA Address Counter (FE)

On the falling edge of PCSD-, LD0 is set high to load the address lines A8-A1. The rising edge of LD0 will start the DMA circuit. This also implies a block mode transfer operation, since bits 7-4 are all high. On power-up the DMA controller defaults to the block transfer mode. It should be noted that all three address counters mentioned above are cascaded allowing for the continues transfer of up to 64 Kbytes.

5.2.4 FIFO Access (F7)

This state opens a path to an internal FIFO that is 64 bytes in length. The falling edge of PCSD- will start to shift data out of the FIFO for a read or shift data into the FIFO on the rising edge of PCSD- if the R/W- was set low with LD2. The DMA will initiate host memory access, done a word at a time, whenever the FIFO is half full. A typical memory access without any wait states takes 4 cycles, each cycle being about 140 nS.

5.2.5 Read DMA Status (EF)

The host DMA status must be read before initiating any data transfer, since its FIFO can be shared by another device. At the end of every word or block transfer initiated by the hard disk controller, the status must be read to ensure successful data transfer completion. Status is not read after every word in a block transfer. After the last byte, in a block transfer, has been strobed into the DMA controller approximately 12 uS are needed to ensure that the DMA status lines are all high. To read the status, any number of PCSD- strobes may be used before initiating another DMA cycle. The DMA internal status available after the falling edge of PCSD- is interpreted as follows:

DATA BIT 7: This line will be high if no DMA was requested or a DMA cycle was completed. After completion of a word or a block transfer, this bit will be set high. A low indicates DMA busy status.

DATA BIT 6: This bit is high if a byte of data is available to be read from the FIFO, or if there is a byte to be written and the FIFO is not full. At the end of a block write operation to the disk, since there are no more bytes available, this bit is set low.

DATA BIT 5: This line is low if the FIFO overflowed or underflowed. This may occur during a disk transfer if the DMA circuit does not receive a bus acknowledge signal from another device on the 68000 motherboard, before the FIFO becomes full or empty. Under this condition the FIFO is cleared by the Z80, before any other data transfer can be initiated.

DATA BITS 4-0: These data lines will be logic zero.

5.2.6 Reset IREQ- (9F)

This state will force IREQ- line to high impedance. It is set low by the host.

5.2.7 Command Complete Acknowledge (BF)

This will cause the assertion of the host vectored interrupt line to its active low state to indicate the completion of a command by the HDC.

5.2.8 Word Transfer (DF)

This will set the internal DMA circuit into a single word transfer. On completion of the word transfer, the DMA resets to a block transfer mode. Hence this state must be strobed for every word transfer desired.

5.2.9 Reset DMA (7F)

This state, followed by state 'FF', resets the DMA circuits and clears the FIFO. This state should be strobed on power-up and to clear any FIFO underflow or overflow conditions.

5.3 Host/HDC Command Protocol

Commands are passed to the HDC through the DMA circuit. When the host requires a disk transfer a command block will be setup in the 68000 memory followed by the host asserting the IREQ- line low. The Z80 will then go through a sequence for each IREQ as discussed below:

5.3.1 Step 1: Setting Up The DMA Address

State FB is loaded into the DMA circuit with PCSS- followed by PCSD- with the hex value of desired high ordered address. Bit 7 of the data bus determines the direction of the transfer, a low will cause a write operation to host and a high will cause a read from host.

Then state FD is loaded into the DMA circuit with PCSS followed by PCSD- with the value of desired address on the data bus. This sets up address lines A16-A9.

State DE is loaded with PCSS- for a word transfer. A value of 06 is loaded with PCSD- to point to the 12th and 13th bytes of the command block. On the falling edge of PCSD- the DMA word cycle will begin. Byte 12 must be FF before the command is executed.

5.3.2 Step 2: Reading Data

The state EF is loaded with PCSS- so that on the falling edge of PCSD- internal DMA status will be outputed. The data lines DATA7, DATA6, and DATA5 are examined until they are high indicating completion of the DMA cycle and that data has shifted through the FIFO. For a block write operation to the disk, DATA6 is examined until low. The HDC will sample the status for about 20 mS, until the data bus contains EO or AO, before attempting to clear the FIFO and re-transmit the block of data, if necessary. If the FIFO cannot be cleared after within 20 mS, the command will be terminated in the normal manner, if possible.

5.3.3 Step 3: Reading The Command Block

If byte 12 is an FF, the rest of the command block is retrieved by the HDC. This requires the execution of Step 1 (LDO only) followed by Step 2 four times. The data value for state DE of Step 1 is incremented from 00 to 03, by the HDC for each word transfer to get all eight command bytes.

5.3.4 Step 4: Data Block Transfer

Block transfers are initiated as in Step 1 except that the third state loaded is FE. The state DE was a single word transfer. The direction of transfer is determined by data line DATA7 when initializing the high order address lines. Status is read by the HDC at the end of every block or word transfer, and at the start of every new command.

5.3.5 Step 5: Command Completion

To complete a command status must be returned to the host. The status information returned is that defined by the 'Request Sense' command. To do this, 2 status words must be transferred to the command block. The host DMA is setup for a word transfer, by setting the LD2, LD1, and the LD0 counters similar to the read of the command block byte 12 (see Step 1). The four status bytes: ERROR, CODE, LUN:LADD2, LADD1, and LADD0 are loaded into the FIFO on the rising edge of PCSD-, a word at a time. As usual, the DMA status is examined, between word transfers. If the command, just executed by the HDC required a disk access, then the ADV (address valid) bit is set. Otherwise ADV = 0 to indicate that the LSA, reported in the 4 byte status block, is meaningless. This completes the instruction. The host is acknowledged by writing state BF to set the host vectored interrupt line low. Also IREQ- is de-asserted by the HDC.

6.0 Commands

6.1 Command Block

In the 68000 memory located at an address determined by Amiga DOS is a 16 byte command block. The first byte received through the FIFO is the MSB even byte, followed by the LSB odd byte. During the command block transfer phase, 8 bytes specifying the command are read by the HDC. The command block is organized as follows:

Table 6-1: Host Command Block

BYTE	WORD	7	6	5	4	3	2	1	0
0	0	Command Class			OP Code				
1	0	Logical Unit Number			Logical Sector Address (High)				
2	1	Logical Sector Address (Middle)							
3	1	Logical Sector Address (Low)							
4	2	Block (sector) Count							
5	2	Control Byte (reserved in DMA spec)							
6	3	High Order DMA DB Address (A23-A17)							
7	3	Mid Order DMA DB Address (A15-A9)							
8	4	Low Order DMA DB Address (A1-A8)							
9	4	Reserved							
10	5	Reserved							
11	5	Reserved							
12	6	ADV	Error Type		Error Code				
13	6	LUN			LADD2				
14	7	LADD1							
15	7	LADD0							

Byte 0 must be specified for all commands. Depending on the value of Byte 0, each parameter in Bytes l through 5 may require specification. Table 6.2 specifies the supported commands and their parameters. It also includes information in data transfers required during execution. All other commands are reserved.

6.1.1 Command Class

There are eight command classes. Command class 0 contains the commands used in normal operation. Command class 7 contains the diagnostic commands. Command classes 1, 2, 4, 5, and 6 are reserved for future use.

6.1.2 Operation Code

There are 32 operation codes in each command class. For a description of all the available op codes see the Command Description Section.

6.1.3 Logical Unit Number

This is contained in the upper three bits of Byte 1 specifying one of eight logical unit numbers. Logical units 0 and 1 are hard disk drives 0 and 1 respectively. Logical units 2 to 7 are reserved for future use. The HDC reports an invalid command if the logical unit number is out of range. However, for error reporting, all even LUN's are treated as drive 0 and all odd LUN's are treated as drive 1.

6.1.4 Logical Sector Address

A logical sector address is a 21 bit unsigned integer that specifies a unique physical sector. The one-to-one correspondence between the set of logical sector addresses and the set of physical sectors is computed by the HDC from the Cylinder (C), Head (H), and Sector (S) address, as well as the drive parameters, heads per drive (HD) and Sectors per track (ST):

$$L = (((C * HD) * H) * ST) + S$$

C, H and S can be derived from L, HD, and ST as follows:

$$S = L \text{ Modulo } ST$$
$$H = (L\text{-}S)/ST \text{ Modulo } HD$$
$$C = (((L\text{-}S)/ST)\text{-}H)/HD$$

This field specifies a sector or the first sector for the Read and Write Drive commands. When only a track specification is required, the sector number implied by the Logical Sector Address is ignored. Hence each format type command begins operation at the beginning of the track containing the specified sector. The HDC will report an invalid command, if the logical address specified is out of range.

6.1.5 Block Count

The sector count is a parameter for each data transfer command. It specifies the number of logical sectors to be transferred during any disk READ or WRITE operations. The sector count is an unsigned, non-zero integer. All zeros in the sector count field specify a count of 256.

For a format command, the number of sectors to be formatted per track is specified by this byte. The interleave factor need not be explicitly furnished by the host, since it is implicitly contained in the interleave table furnished by the host.

6.1.6 Control Field

The control field is reserved for future use.

6.1.7 DMA Memory Address

The next three bytes, bytes 6, 7, and 8, make up the 23 bit address which points to the block of 512 byte to be transfered via DMA. This block of memory contains data bytes or specifies an address value as required by the command to be executed. Since the R/W- bit is part of the LD2 memory address counter, address bits A1-A23 are shifted right 1 bit by the HDC before being stored for command execution.

6.1.8 Status and Error Bytes

At the completion of each command the HDC will return status in the last four bytes (12-15) of the command block. The status format is similar to that returned by the 'Request Sense' SCSI command. This four byte block contains error and status information pertaining to the last block of data transferred or a non-disk operation executed by the HDC. The ADV bit will be set, to indicate a valid address, if the last operation required a disk access, otherwise ADV = 0.

The logical unit number returned is simply the contents of the logical unit field, where the error occurred, as defined in the drive control block. For those commands that do not take a logical unit number as an input parameter, the logical unit number returned in the command status byte is not meaningful.

A list of possible error codes, along with their descriptions, follows:

6.1.8.1 Error Bytes

The logical sector address bytes are to be in the same format at that defined in the command block. Bits 3-0 of the error byte is used for the error codes. Bits 4, 5 indicate the error type and 7 is the ADV bit. Bit 6 is not used presently.

Disk Drive Error Codes (Type 0)

0 No Error
1 No Index
2 Seek not complete
3 Write fault
4 Drive not ready
6 Track 0 not found

Controller Error Codes (Type 1)
- 10 Disk read I.D. error
- 11 Uncorrectable data error
- 12 Address mark not found
- 13 Sector not Found, Read
- 14 Sector not Found, Write
- 15 Seek error
- 18 Correctable ECC error
- 1A Format error

Command Error Codes (Type 2)
- 20 Invalid command
- 21 Invalid sector address
- 22 Soft header error in read
- 23 Soft header error in write
- 24 Soft data error
- 28 Soft DMA error

Hardware Error Codes (Type 3)
- 30 RAM failure (HDC)
- 31 ROM Checksum Error
- 32 Host DMA status error

6.1.8.2 Error Code Description

No Error
A code of 00 or 80 is returned if no errors were detected during the execution of the last operation.

No Index (1)
The HDC does not detect index signal from drive.

Seek in Progress (2)
This error code is only returned by the test drive ready command when the target is a hard disk that supports buffered seeks. It indicates that drive is busy doing a buffered seek. No other command will be executed on the selected drive, until the seek is completed.

Write Fault (3)
This error code is returned by the hard disk drives. It indicates that there was write current to the head when the write gate was off. This is a very serious problem and should be fixed immediately. No command will be executed when this condition is detected.

Drive Not Ready (4)
No disk operations are executed unless the drive is ready.

Track 0 Not Found (6)
This error code is only returned by the recalibrate command. It indicates that the track 0 status from the drive did not become active after the maximum necessary steps towards cylinder 0. Besides drive malfunction, this type of error usually occurs if more than 1 disk drive is selected at the same time, either by the HDC or by the option switches on the supported drives.

Uncorrected Data Error (11)
For a Winchester drive this error code indicates one or more error bursts in the data field were beyond the error correction code's ability to correct. It could also mean that the HDC was unable to obtain a match of two consecutive syndromes within eight read attempts. The sector data for the sector in error is sent to the host, prior to any retries and correction algorithms used.

Address Mark Not Found (12)
It indicates that the header for the target sector was found, but its address mark was not detected. This is treated like a data field error, except that no data transfer to the host takes place. If the error persists after 8 attempts, an auto-restore is performed, followed by a reseek, and another 8 attempts to read the desired LSA.

Sector Not Found, Read (13)
The HDC found the correct cylinder and head but not the target sector.

Sector Not Found, Write (14)
The HDC found the correct cylinder and head but not the target sector.

I.D. Not Found (10)
If the ID field cannot be read correctly after all the retries have been exhausted, this error code is set and the operation terminated. The HDC searches for the ID field 8 times.

Format Error (1A)
During a check track command the HDC detects one of the following errors:
1) Track not found.
2) Bad ID.

Illegal Parameters (20, 21)
These error codes, invalid command (20), illegal LSA (21), and illegal LUN (22) are self explanatory.

HDC RAM Error (30)
During internal diagnostic the HDC detects a RAM error.

HDC ROM Checksum Error (31)
During internal diagnostic the HDC detects a ROM checksum error.

Host DMA Error (32)
This error code is set whenever invalid status is read from the DMA during any data or command access. For most operations the status checked is E0 (hex), except for a block write. In this case the valid status checked for is A0.

6.2 Command Description

All commands executed by the HDC are summarized in the table below. Fields of the command block not specified are don't cares. Following this summary is a generalized description of the commands.

Table 6-2: Command Summary

Command Description	Class Opcod	LUN Num	LADD (21)	Int/ BCNT	Control Options	Possible Error Codes
Read Drive Status	00	0-1				RDS
Restore to TK0	01	0-1				06, RDS
Request Status	03	0-1				Last Oper.
Check Trk Fmt	05	0-1	L		R	RDE, RDS, IDA
Format Track	06	0-1	L	B	S	IDA
Read Drive	08	0-1	L	B	R,S	RDE, RDS, IDA
Write Drive	0A	0-1	L	B	R,S	15, 19, RDS, IDA
Seek	0B	0-1	L			RDS, IDA
Set Drive Param.	0C	0-1				20, 32
Change Command Block Address	0F					20, 32
Read Drive Long	E5	0-1	L	B	R,S	RDE, RDS, IDA
Write Drive Long	E6	0-1	L	B	R,S	15, 19, RDS, IDA
Init. Unit 1	CC	1				20, 32

R = 0 Retries/ECC enable
= 1 Retries/ECC disabled

S = 0 Set correction span to 5 bits
= 1 Set correction span to 11 bits

L = Logical Sector Address

B = Block or sector count required

Read Drive Status (RDS) = 02, 03, 04, 20, 32
Illegal Disk Access (IDA) = 20, 21, 22, 32
Read Sector Error (RDE) = 11, 12, 13, 14, 15

6.2.1 Read Drive Status (Class 0, Opcode 0)

Action
Read the drive's status and determine if drive is ready. For Hard disk drives supporting buffered seeks this command is useful for determining the first drive to reach its target track. The command will be aborted, if the drive status read is incorrect.

Possible Error Codes
No error, invalid command, seek in progress, drive not ready, write fault, DMA error.

6.2.2 Restore (Class 0, Opcode 1)

Action

The Restore command positions the heads to cylinder 0. It is usually issued by the host when the drive has been turned on, or before a format drive operation is initiated by the host.

Possible Error Codes

No error, invalid command, Track 0 not found, drive not ready, write fault, DMA error.

6.2.3 Request Status (Class 0, Opcode 3)

Action

Send the host four bytes of error information for the specified drive. The status of the last command executed may have already set the error register but the execution of this command will not set any new bits. If however, the command requesting the status is invalid, then the previous command status will be lost.

Possible Error Codes

No error, invalid command, last operation status, DMA error.

6.2.4 Check Track Format (Class 0, Opcode 5)

Action

Verify that the specified track is formatted with the correct number of logical sectors. A multiple read command is issued by the HDC to verify all the ID fields on that track and the data read back from the disk is discarded. Retries may be enabled if desired.

Possible Error Code

No error, invalid command, invalid sector address, IDNF error, drive not ready, write fault, invalid LUN, seek not complete, DMA not found, uncorrectable data error, DMA error.

6.2.5 Format Track (Class 0, Opcode 6)

Action

The format track command is used for initializing the ID and data fields on a specified track. The current contents of the specified track are overwritten. This command is useful for marking any bad sectors or tracks after the entire disk surface has been formatted. Assignment of alternate tracks or simply not specifying bad logical addresses is best handled by the host driver routines in the interest of flexibility and reducing onboard firmware requirements.

Possible Error Codes

No error, invalid command, invalid sector address, drive not ready, seek not complete, write fault, invalid LUN, DMA error.

6.2.5.1 Interleave Considerations

During this command the sector is set up by the host to contain additional parameter information instead of data. Each sector requires a two byte sequence. The first byte designates if a bad block (80) or good block (00) is to be recorded in the ID field. The second byte indicates the logical sector number to be recorded on the disk, as shown below:

Table 6-3: Interleave Factor Table

Addr in Hex	Data for an Interleave factor of: (HEX) 1	2	3	4
00	00	00	00	00
01	00	00	00	00
02	80	00	00	00
03	01	09	06	0D
04	00	80	00	00
05	02	01	0C	09
06	00	00	80	00
07	03	0A	01	05
08	80	00	00	80
09	04	02	07	01

***Table 6-3: Interleave Factor Table** (continued)*

Addr in Hex	Data for an Interleave factor of: (HEX) 1	2	3	4
0A	00	00	00	00
0B	05	0B	0D	0E
0C	00	00	00	00
0D	06	03	02	0A
0E	00	00	00	00
0F	07	0C	08	06
10	00	80	00	00
11	08	04	0E	02
12	00	00	00	00
13	09	0D	03	0F
14	00	00	00	00
15	0A	05	09	0B
16	00	00	00	00
17	0B	0E	0F	07
18	00	00	80	00
19	0C	06	04	03
1A	00	00	00	00
1B	0D	0F	0A	10
1C	00	00	00	00
1D	0E	07	20	0C
1E	00	00	00	00
1F	0F	10	05	08
20	00	00	00	80
21	10	08	0B	04
All	XX	XX	XX	XX
Rest	XX	XX	XX	XX

These numbers can be from 00 to 10 (hex), or 17 sectors per track or any number that the host wishes to specify that meets the drive track capacity. Bad block marks are shown for sector numbers 1 and 4 in all four interleave factors illustrated. The other requirement of the host is to provide the logical sector number. Using this scheme, sectors can be recorded in any interleave factor desired. Byte four of the command block then specifies the number of sectors to be formatted per track. Also the host is free to choose marking individual sectors or entire tracks bad. At the end of a track format, the host can re-issue the command, for formatting the track across head boundaries as shown below:

Table 6-4: Interleaving Across Head Boundaries

00	01	02	03	04	.....	0E	0F	10
10	00	01	02	03	.....	0D	0E	0F
0F	10	00	01	02	.....	0C	0D	0E
0E	0F	10	00	01	.....	0B	0C	0D

Using the above spiral format approach, the HDC has approximately 1 mS for any processing overhead required. This 1 mS loss in the 1:1 performance across head boundaries, assuming a disk rotational speed of 3600 r.p.m. is reasonable. Across cylinder boundaries, the 1:1 interleave factor cannot be maintained because of the step rates involved. To format the entire disk using the Format Track command the host must update the buffer, if desired, and re-issue the command every track formatted. This is not really a major advantage since the host driver routines can easily re-issue the command in a loop until the entire disk is formatted. This gives the host total flexibility to format the drive using any clever algorithms for formats across head and cylinder boundaries instead of a canned approach.

6.2.5.2 Physical Track Format

The data fields are filled with FF hex, and the ECC is generated as specified by the related coding options. The Gap 3 value is determined by the drive motor speed variation, data sector length, and the interleave factor. The interleave factor is only important when 1:1 interleave is used. The formula for determining the minimum Gap 3 is:

$$\text{Gap 3} = 2 \times M \times S + K + E + V$$

M = motor speed variation (e.g. .01 for ±1%)
S = sector length in bytes
K = 18 for an interleave factor of 1
E = 2 if ECC is enabled
V = number of overhead bytes required for the HDC between sectors
= 9 (for an interleave factor of 1)

To maximize data read back efficiency and maintain the interleave factor of one, as closely as possible, it is required that the physical sector numbers be offset by a sector from track to track, (see table) so that the HDC has a sector length available for overhead to switch heads while on the same cylinder.

6.2.6 Read Drive (Class 0, Opcode 8)

Action

Read the specified number of consecutive sectors beginning with the specified sector in the command block to the host computer. If ECC is enabled, ECC bytes are recomputed by the HDC. After the data is transferred to the host, the recorded ECC bytes are compared to the generated bytes to generate the syndrome bytes. If the syndrome is non-zero, errors have occurred. Error correction is invoked by the HDC if two consecutive syndromes match, otherwise a maximum of 8 retries are attempted by the HDC.

Possible Error Codes

No error, invalid command, invalid sector address, invalid LUN, IDNF error, bad block mark, address mark not found, uncorrectable data error, write fault, drive not ready, seek in progress, DMA error.

6.2.7 Write Drive (Class 0, Opcode A)

Action

The Write Sector command is used to write the specified number of sectors of data from the host computer to the disk, beginning with the specified logical address in the command block. The write operation is identical to the read, except for error handling and reading the host status.

Possible Error Codes

No error, invalid command, invalid sector address, invalid LUN, drive not ready, IDNF error, bad block mark, write fault, seek in progress, DMA error.

6.2.8 Seek (Class 0, Opcode B)

Action

The Seek command positions the R/W head to the cylinder contained in the logical address. No ID field is read to verify start or end position. Seek It is primarily used to move the R/W head to the Shipping zone for transportation of the hard disk.

Possible Error Codes

No error, invalid command invalid sector address, invalid LUN, drive not ready, write fault, DMA error.

6.2.9 Set Drive Parameters (Class 0, Opcode C)

Action

This command points to a 6 byte block of memory, specified by bytes 6 and 7 of the command block, that sets the following parameters for both of the hard disk drives (logical units 0 and 1):

Table 6-5: Set Drive Parameters

D7	D6	D5	D4	D3	D2	D1	D0
User Options				Step Rate			
0	Num. of Heads			CYL. Nums. MSN			
Number of Cylinders LSB							
Precompensation Cylinder / 16							
Reduce Write Current Cylinder / 16							
Number of Sector per Track							

If the above command is not executed after power up or every reset, the HDC will assume the following default parameters:

306. = Number of cylinders (131 hex)
4 = Number of heads
128. = Starting write precompensation cylinder
128. = Reduce write current cylinder
3 mS = Step rate
5 = Maximum length of an error burst to be corrected
17. = Number of sectors per track
8. = Retries & ECC enable

The acceptable range of values for these parameters are as follows:

0 - 2047. Number of cylinders
0 - 7 Number of heads
0 - 255 Sector Numbers
0 - 1023. Starting write precompensation cylinder
5 / 11. Maximum length of error burst to be corrected
0 / 8 Retries

If one of the parameters is out of range, then an "invalid command" error code i generated by the HDC. Bytes 2 thru 5 of table are self explanatory and will not be discussed any further.

User Options

This four bit field can be used to specify options as indicated below:

Bit 7 = 0 5 bit correction span (default value)
= 1 11 bit correction span

Bit 6 = 0 Retries & ECC enabled (default value)
= 1 Retries & ECC disabled

Bit 5 = 0 Not Used

Bit 4 = 0 Not Used

Step Rate

Step Rate 14 = 11.1 usec
Step Rate 15 = 30 usec
All Others = 3 msec

Possible Error Codes

No error, invalid command, DMA error.

6.2.10 Initialize Unit 1 (Opcode CC)

Action

This command with initialize or set drive parameters of unit 1 only. This allows for the HDC to support two different drive types at the same time. The action of this command is identical to the action of the 'Set Drive Parameter' command noted above except that it will effect only unit 1. For command details see section 6.2.9.

6.2.11 Change Command Block (Class 0, Opcode F)

Action

The Change Command Block is used to move the location of the command block from the default on power up to a new location. Bytes 6 and 7 of the command block are used as indirect address pointers for the beginning of a 7 byte block of memory organized as follows:

Table 6-6: Change Command Block Address

D7	D6	D5	D4	D3	D2	D1	D0
0							0
A23			High Order DMA Byte				A16
A15			Mid Order DMA Byte				A08
A07			Low Order DMA Byte				0

Since the host R/W bit, and address bits A23-A17, form the data byte for the host LD2 – counter, the DMA high and middle order address bytes are shifted right 1 bit position before being used. Since a copy of the previous address is not maintained, the command status is returned to the new address location specified and not the old one.

Possible Error Codes
No error, invalid command, DMA error.

6.2.13 Read Long (Class 7, Opcode 5)

Action
Similar to Read Sector except the ECC operation producing the syndrome is inhibited in the HDC. Instead the HDC copies the recorded CHECK bytes from the disk and passes them unaltered to the host. This command is useful in debugging and verifying the ECC hardware and software. To do this first write normally, and then READLONG. The data or the check bits may now be altered by the host and written to the disk using the WRITELONG command. If a READ command were issued, then the HDC should invoke error correction on the data field and correct it as long as the error induced is within the correction capability of the ECC polynomial.

Because there is no storage register on board, this command is implemented only for diagnostic purposes. Also note that the 4 extra checkbytes are to be accessed directly by the host. Hence the diagnostic tester used is required to support a 516 byte block transfer instead of the standard 512 byte block transfer supported by the Amiga system.

Possible Error Codes
No error, invalid command, invalid sector address, invalid LUN, IDNF error, bad block mark, address mark not found, write fault, drive not ready, seek not complete, DMA error.

6.2.14 Write Long (Class 7, Opcode 6)

Action
The Write Long command functions similarly to the Write Sector command except the ECC operation of computing the ECC word is inhibited in the HDC. Instead, the HDC accepts a 32 bit appendage from the host and passes it unaltered to the DJC to be written on the disk after the data. This command is useful for diagnostic purposes only. It allows the generation of a sector containing a correctable ECC error. See the Read Long command description for operation details and system requirements.

Possible Error Codes
No error, invalid commands, invalid sector address, invalid LUN, IDNF error, bad block mark, fault, seek not complete, drive not ready, DMA error.

A2090A HARD DISK CONTROLLER ADDENDUM

Your new A2090A Hard Disk Controller card includes two autoboot EPROMs. These EPROMs are ONLY to be used when the Kickstart Version 1.3, or later, ROM is present in your Amiga® 2000. Use of these ROMs with Kickstart Version 1.2 may interfere with the operation of your computer.

Installation of the EPROMs should be performed by an authorized Commodore Service center. Commodore shall not be responsible or liable for any damages whatsoever caused or occasioned by improper installation or use of these EPROMs.

Installation of the Autoboot EPROMs

The two Autoboot EPROMs for the A2090A are installed in chip locations U50 and U51 (directly under the J1 and J2 cable connectors).

The HI, or Even, EPROM must be installed in U50, while the LO, or Odd, EPROM goes in U51. There is lettering on both the EPROM and the Controller board so that you can differentiate which location is HI (Even) and which one is LO (Odd).

To insert the EPROM:

1) Make sure the EPROM is facing the correct direction. The EPROM is rectangular and will be inserted to conform to the outline drawn on the board. However, there is a small notch on on edge of the EPROM (one of the shorter edges). This notch must align with the notch in the board's socket.
2) Once you have the EPROM oriented in the proper direction, align the pins of the chip with the holes in the socket.
3) Carefully and very slowly, begin to press the EPROM into place. Constantly check to make sure that all the pins are going into the holes and that none of the pins are out of alignment. The EPROM is fully inserted when it is just about flush with the socket.

Autobooting with the A2090A

***NOTE:* YOU CANNOT AUTOBOOT WITH THE A2090A UNTIL THE KICKSTART VERSION 1.3, or later, ROM HAS BEEN INSTALLED IN YOUR A2000.**

When your A2000 has the Kickstart Version 1.3, or later, ROM installed, you can autoboot directly from your hard disk. This means that you will not have to insert a Workbench, or other bootable, disk into our computer when you turn it on. As long as your hard drive has been property prepped and formatted, as explained in the accompanying user's guide, you will be able to autoboot by simply turning on the power to your A2000.

Inserting a floppy disk into the internal floppy drive (df0:), will override the autobooting from the hard drive. This allows you to boot from a floppy disk whenever you desire.

Correction to page 1 of the A2090 Hard Disk/SCSI Controller User's Guide:

Please add the following hard drives to the list of models available for use with the Amiga 2000:

ST506 Drives	SCSI Drives
Miniscribe 8425F	Connor CP340
	Miniscribe 8051S
	Quantum ProDrive 40S
	Rodime RO3057S Type 00B
	Seagate® ST251N*

*You cannot autoboot with a Seagate SCSI hard drive and an A2090A Controller Card because of the long initialization process in the Seagate power-up sequence. (This pertains to all Seagate SCSI hard drives, including the drives listed on page 1 of the Hard Disk/SCSI Controller User's Guide.)

Correction to page 34:

The information below replaces the information in the "Using Multiple Hard Disk Controllers" section (page 34) of the A2090 Hard Disk/SCSi Controller User's Guide:

Multiple Controller Boards

It is possible to install more than one A2090/A2090A Hard Disk Controller into the Amiga 2000 if you would want to use more than two ST506 hard disks and/or seven SCSI devices with your Amiga system.

For each additional Controller, you can install up to two ST506 hard disks and/or seven SCSI devices. The driver name assignments for hard disks connected to additional Controllers follow a two-pass configuration procedure, explained below.

Binding Driver Names to Controller Boards

The first pass of the driver configuration procedure occurs for the A2090A Autoboot boards. (These boards MUST HAVE valid Autoboot EPROMs installed.) The second pass occurs for A2090 boards and A2090A boards without the Autoboot EPROMs installed; this pass takes place when the BindDrivers program of your startup-sequence is executed.

During the first pass of the driver configuration process, Autoboot A2090A boards are configured sequentially, beginning with A2090A board in the slot closest to the processor (nearest to the power supply) and continuing outward for each Autoboot board in turn. The first board is assigned to the "hddisk.device," the second assigned to the "iddisk.device," and additional boards are assigned to the "jddisk.device,""kddisk.device," and so on.

The second pass occurs during the execution of BindDrivers. A2090 (non-Autoboot) boards are also configured sequentially starting with the A2090 board closest to the processor and continuing outward. All non-Autoboot boards receive an assignment to a single driver name. This name corresponds to the next name available after the first pass.

For example, if you had two Autobooting A2090A boards and two A2090 boards installed, the assignments would be as follows:

A2090A board closest to processor	hddisk.device
2nd A2090A board	iddisk.device
Both A2090 non-autobooting boards	jddisk.device

These assignments must be included in the MountList entry for each ST506 or SCSI device. For instance, "device = hddisk.device" or "device = jkkisk.device".

Assignment of Amiga DOS Names to Hard Disks or SCSI Devices

To create unique AmigaDOS name assignments to the actual hard disk devices, a similar two-pass procedure is followed. First, all hard disk drives connected to A2090A boards are named, beginning with the A2090A Controller closest to the processor and continuing outward (as described above). Then, the devices connected to the A2090, non-autobooting, boards are named; again, starting with the board closest to the processor and continuing outward.

The first ST506 drive found as MountList Unit 1 becomes DH0; and the first ST506 drive found as MountList Unit 2 becomes DH1:.

The second ST506 drive found as MountList Unit 1 becomes D10;, while the seond ST506 drive found as MountList Unit 2 becomes DI1:.

Note that the first ST506 drive found as Unit 1 becomes DH0: NO MATTER TO WHICH CONTROLLER BOARD IT IS CONNECTED. This simplifies the creation of start-up sequences which refer to DH0: as the boot device.

SCSI devices follow the same pattern. The first SCSI device found as MountList Unit 3 becomes DH2;, and the first SCSI device found as MountList Unit 4 becomes DH3:. The second SCSI device found as MountList Unit 3 becomes DI2:, while the second SCSI device found as MountList Unit 4 becomes DI3:, and so on.

Note that the first SCSI device found as Unit 3 becomes DH2: NO MATTER TO WHICH CONTROLLER BOARD IT IS CONNECTED. This simplifies the creation of startup-sequences which refer to DH2: as the boot device.

Assignment of AmigaDOS Unit Numbers to Hard Disks/SCSI Devices

To create unique AmigaDOS assignments to actual hard disks or SCSI devices, the now familiar two-pass procedure is also followed. First, all devices connected to A2090A Autobooting Controller boards are given Unit numbers, starting with the devices connected to the Controller board closest to the processor and continuing outward. Then the devices connected to the A2090 boards are named, again starting with the board closest to the processor.

All DOS Units connected to A2090A Autoboot boards are single digit Unit numbers within the range of 1 through 9. The Unit assignments are:

Hard Disk/SCSI Device	Unit Number for MountList entry
1st ST506 drive	1
2nd ST506 drive	2
1st SCSI device	3
2nd SCSI device	4
3rd SCSI device	5
4th SCSI device	6
5th SCSI device	7
6th SCSI device	8
7th SCSI device	9

No matter how many autobooting Controllers are installed, the Unit number stays within the 1 through 9 range. For instance, the 1st SCSI device on the 1st A2090A card (hddisk.device) is given a unit number of 3. The 1st SCSI device on the 2nd A2090A card is also given a number of 3, but that card's device name is "iddisk.device."

The unit numbers for non-autobooting A2090 Controller cards are double digit numbers within the range 01 through 99. With the non-autobooting cards, the unit number is dependent on the card to which the device is attached.

For devices attached to the first A2090 card:

Hard Disk/SCSI Devices	Unit Number for MountList Entry
1st ST506 device	01
2nd ST506 device	02
1st SCSI device	03
2nd SCSI device	04
3rd SCSI device	05
4th SCSI device	06
5th SCSI device	07
6th SCSI device	08
7th SCSI device	09

For devices attached to the second A2090 card, the unit numbers range from 11 to 19. For devices attached to the third A2090 card, the unit numbers range from 21 to 29, and so on for each additional card.

Remember,

1) The device name for the MountList entry is determined by the location of the card and by whether or not it is autobooting Controller. The 1st A2090A card found will be recognized as "hddisk.device". Each additional A2090A card found will be given a name one unit higher alphabetically, "iddisk.device", jddisk.device", etc. ALL A2090 cards go by the same device name; whichever name would follow after all the A2090A cards are named.

2) The AmigaDOS name is dependent on the location of the hard disk SCSI device. The DOS assignments for the 1st ST506 hard disk is always dh0 no matter if the drive is connected to the 1st or 5th Controller card. For instance, if you had the following configuration:

1st A2090A Card —	hddisk.device	
1 ST506 hard drive	Unit 1 (MountList)	dh0:(AmigaDOS)
1 SCSI device	Unit 3 (MountList)	dh2:(AmigaDOS)
2nd A2090A Card —	iddisk.device	
2 ST506 hard drives	Unit 1 (MountList)	di0:(AmigaDOS)
	Unit 2 (MountList)	dh1:(AmigaDOS)
1st A2090 Card —	jddisk.device	
2 SCSI devices	Unit 03 (MountList)	di2:(AmigaDOS)
	Unit 04 (MountList)	dh3:(AmigaDOS)

General Note Regarding Hard Drives and Graphic Displays:

Due to DMA contention, you may notice a performance degradation when using SCSI drives and displaying 4 bitplane graphics. Performance will be improved if you move the 4bitplane screen to the background or if you pull it down. The performance of ST506 hard drives may be affected but to a lesser degree.

MAJOR PARTS LIST

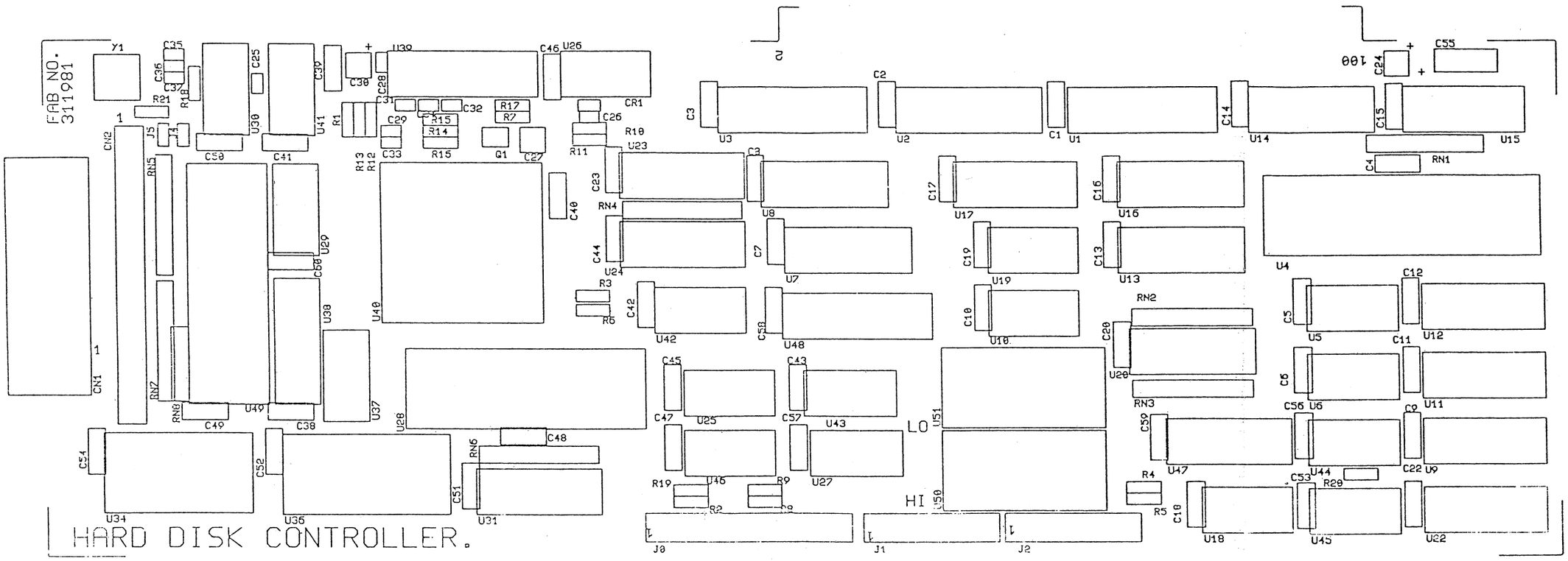
319943-01	A2090 USER GUIDE HDC	311979-02	PCB ASSEMBLY A2090A
318865-01	A2090A HDC ADDENDUM	314880-01	A2090A SERVICE MANUAL
317733-01	PROGRAM DISK		

COMPONENT PARTS LIST
PCB ASSEMBLY #311979-02

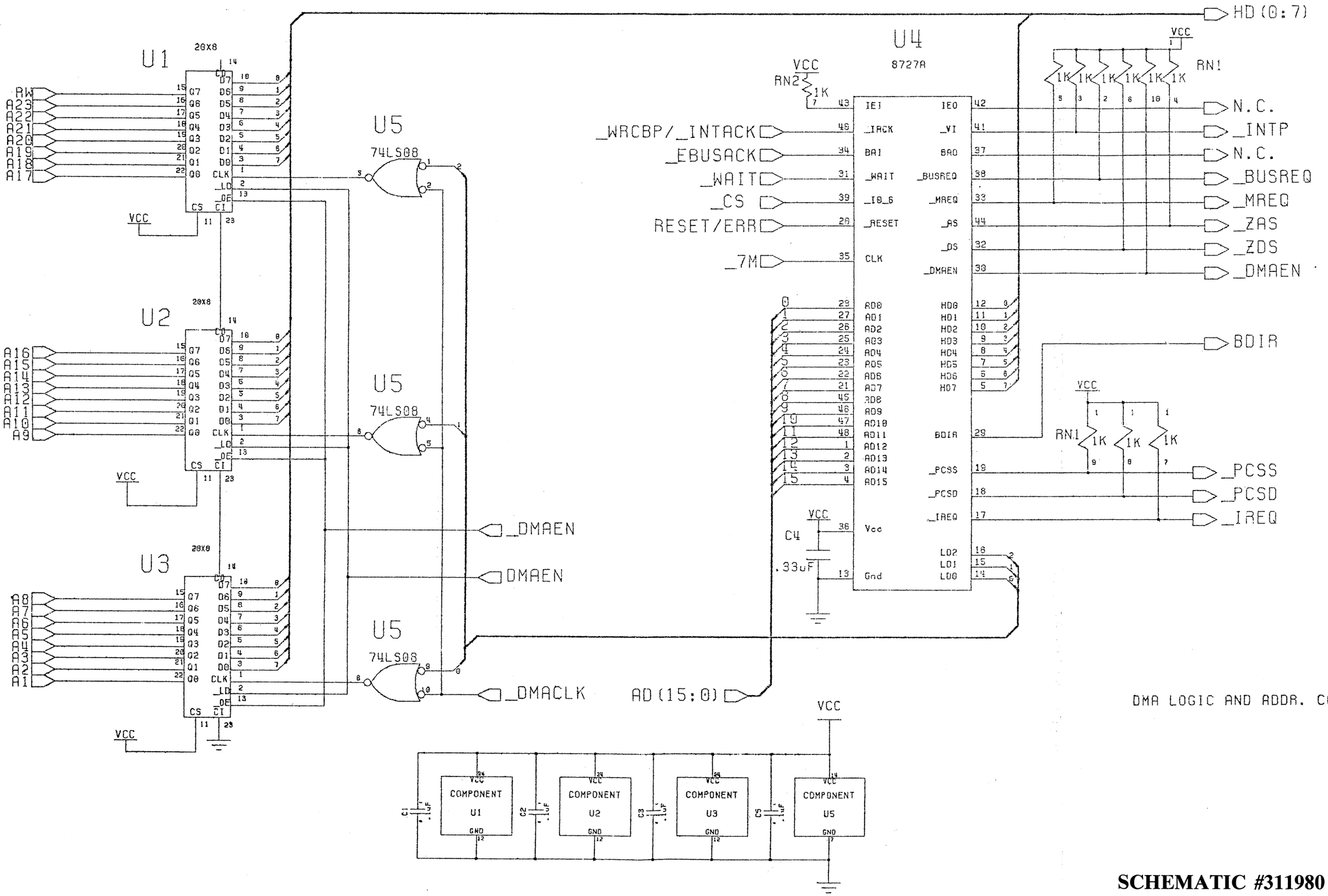
Commodore part numbers are provided for reference only and do not indicate the availability of parts from Commodore. Industry standard parts (Resistors, Capacitors, Connectors) should be secured locally. Approved cross-references for TTL chips, Transistors, etc. are available in manual form through the Service Department, order #314000-01.

IC COMPONENTS		
318096-01	16L8 PAL	U8
390235-01	20L10 PAL	U8
390234-02	20L10 PAL	U48
390234-03	30L10 PAL	U48
390231-01	20 X 8 PAL	U1,U2,U3
315098-01	2764 EPROM, 250NS	U50
315097-01	2764 EPROM, 250NS	U51
390233-01	16L8 PAL	U17
390236-01	16L8 PAL	U7
390236-01	16L8 PAL	SUB: U7
390232-02	16L8 PAL	U38
390237-01	16R6A, PAL	U16
290001-01	74LS245	U14,U15,U31
390089-01	74F245	U47
390011-01	74LS74A	U44
390081-01	74F74	U18
390198-01	74F86	U19
390002-01	7407	U10
901521-02	74LS04	U6
901521-03	74LS08	U5,U29
315046-01	8727, CBM DMA	U4
318041-01	74F521, 8BIT COMPARATOR	U13
901521-43	74LS374	U9,U20
901521-13	74LS244	U11,U22,U23
390158-01	74LS273	U24,U12
251637-05	2016 2K X 8 STATIC RAM, 200NS	U34
318040-01	26LS31, DIFFERENTIAL TRANSMITTER	U26
318043-01	26LS32, DIFFERENTIAL RECEIVER	U27
390207-01	DP8465N-4, DATA SEPARATOR	U39
906150-01	Z10A CPU	U28
390230-02	2764 EPROM, 250NS	U36
390210-01	DJC KNONAN HDC (84 PIN LCC)	U40
390004-01	7406	U42,U43
318042-01	74HCU04	U30
901521-61	74LS76	U37
390156-01	74LS174	U25
324667-02	DDU-6-60 DELAY LINE	U45
901521-11	74LS157	U46
390206-01	WD33C93 SCSI, WESTERN DIGITAL	U49
SOCKETS		
904150-05	28 PIN, .6 CENTER	U36
904150-05	28 PIN, .6 CENTER	U50,U51
904150-06	40 PIN	U28,U49
904150-08	20 PIN	U7,U16,U17,U38,U8
390060-01	24 PIN, .3 CENTER	U1-U3,U39,U48
251313-01	48 PIN	U4
390185-01	CARRIER, PLASTIC LEADED	U40

CRYSTAL, DIODES, TRANSISTORS		
900556-09	CRYSTAL, 20MHz, HC-18/U STYLE OR EQUIV.	Y1
390017-01	DIODE, 1N914	CR1
902707-01	TRANSISTOR, 2N3906	Q1
CAPACITORS		
356500-72	.1UF, 50V, Z5U (MLC)	C28
900020-01	.1UF, 50V, 20% CERAMIC	C1-C3,C5,C6-C23,C28, C38-C54,C56-C60
900020-09	.33UF, 50V, 20% CERAMIC .2IN	C4
900014-02	.001UF, 50V, 10% CERAMIC	C26
900014-03	.01UF, 50V, 10% CERAMIC .1IN	C29,C31
900014-04	.022UF, 50V, 10% MONO/CER .1IN	C33
900402-13	1UF, 35V, 10% MONO/CER .1IN	C30
900402-09	10UF, 20V, 10T TANTALUM .1IN	C27
900019-13	22PF, 50V, 10% MONO/CER .1IN	C35
900019-14	39PF, 50V, 10% MONO/CER .1IN	C36,C37
900050-27	95PF, 300V, 1% MICA .1IN	C32
900018-23	390PF, 50V, 5% CER .1IN	C34
900019-15	100PF, 50V, 10% DIP, CER .1IN	C25
390101-04	22UF, 25V, ELECTROLYTIC, RADIAL	C24
RESISTORS — All values are in ohms-1/4W, 5% unless noted otherwise.		
902410-10	1K OHM, SIP	RN1-RN3
902410-06	3.3K OHM, SIP, 10 PIN, PIN 1 COLUMN	RN6
380388-04	220/330 OHM, SIP, 10 PIN, PULLUP/PULLDOWN	RN4,RN5,RN7
380388-01	220/330 OHM, SIP, 6 PIN, PULLUP/PULLDOWN	RN8
901550-32	1.3K OHM	R15
901550-69	2.5K OHM	R14
901550-49	100 OHM	R4,R5,R16
901550-89	150 OHM	R13
901550-40	620 OHM	R2,R3
901550-01	1K OHM	R1,R8,R9,R11,R19
390170-05	1K OHM, .125W, 1% METAL FILM	R17
901550-53	2K OHM	R7
390170-06	4.75 OHM, .125W, 1% METAL FILM	R12
901550-20	10K OHM	R6,R10
901550-09	10M OHM	R18
901550-17	1.2K OHM	R20
901550-14	330 OHM (FOR LED DRIVER)	R21
MISCELLANEOUS		
390241-06	CONNECTOR, 25 PIN, D TYPE	CN1
903345-01	HEADER, 2 PIN, DUAL IN-LINE	J4,J5
903345-10	HEADER, 20 PIN, DUAL IN-LINE	J1,J2
903345-17	HEADER, 34 PIN, DUAL IN-LINE	J0
903345-25	HEADER, 50 PIN, DUAL IN-LINE	CN2
380120-09	EXTENSION CARD PANEL	
316420-01	LABEL, A2090, FCC ID:	



PCB BOARD LAYOUT #311979

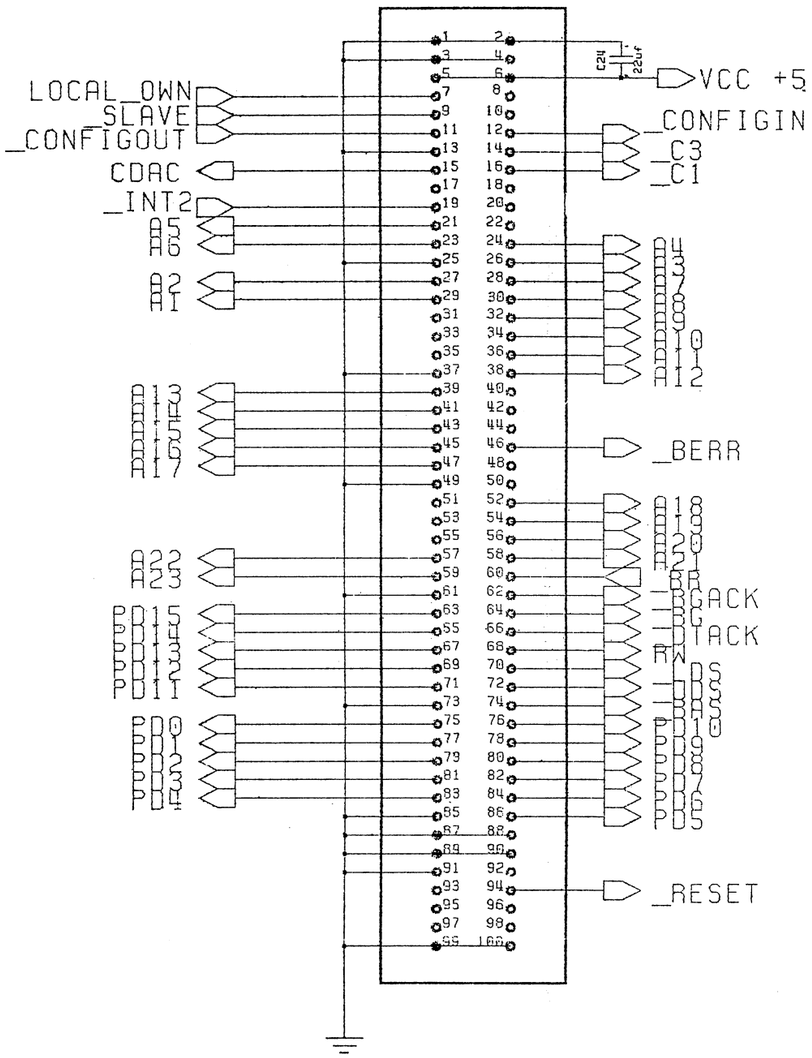


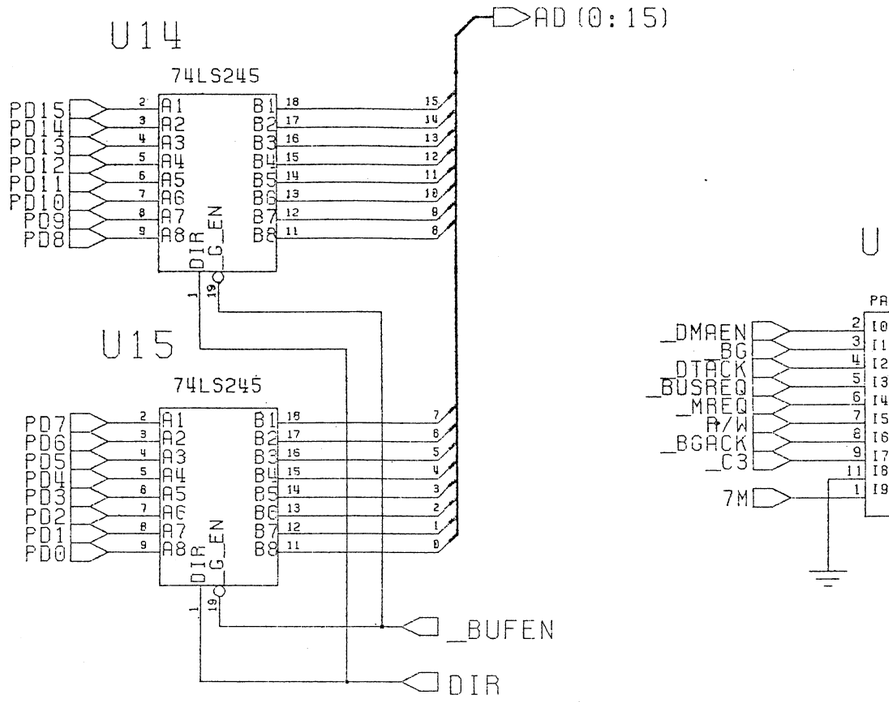
DMA LOGIC AND ADDR. COUNTERS

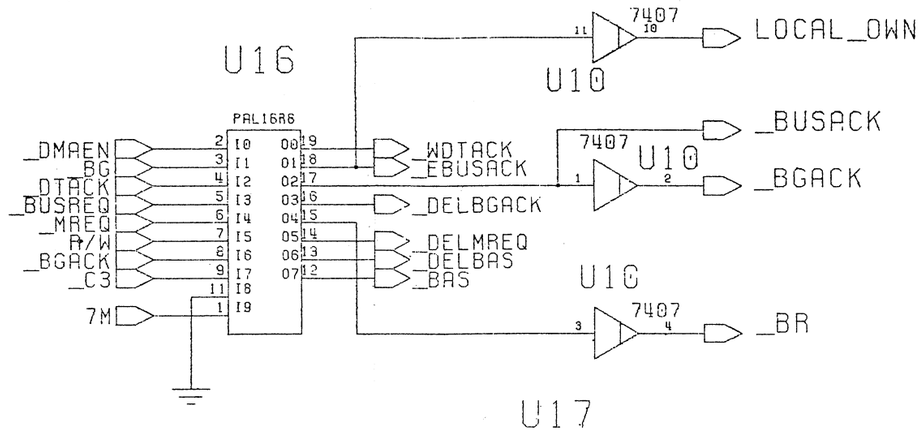
SCHEMATIC #311980 REV. A

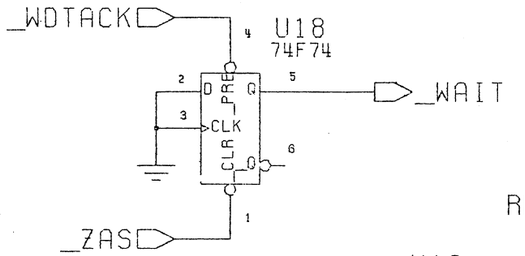
68000 EXPANSION BUSS INTERFACE

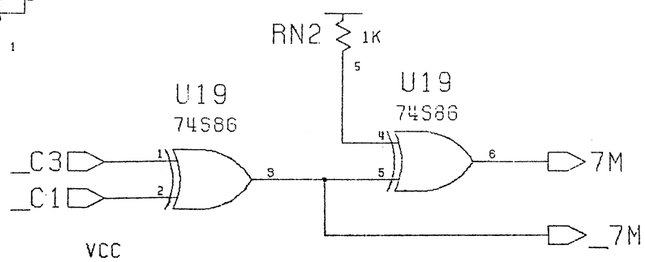
100 PIN EDGE CONN.

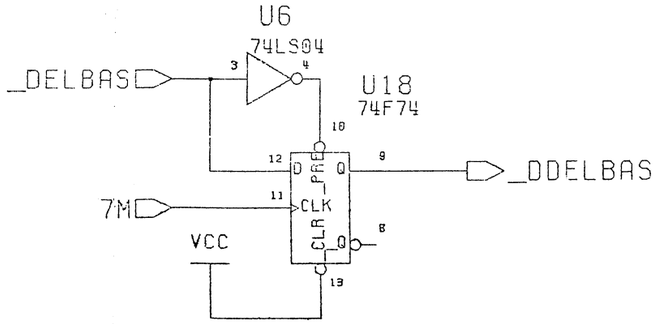


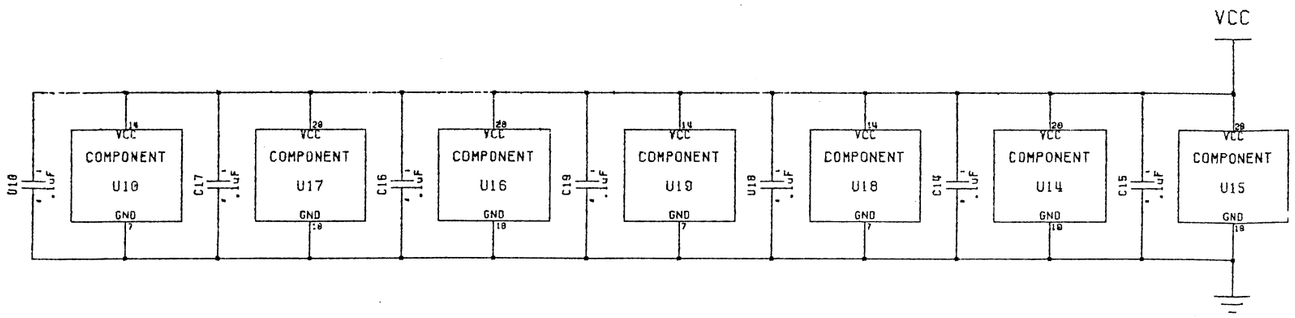






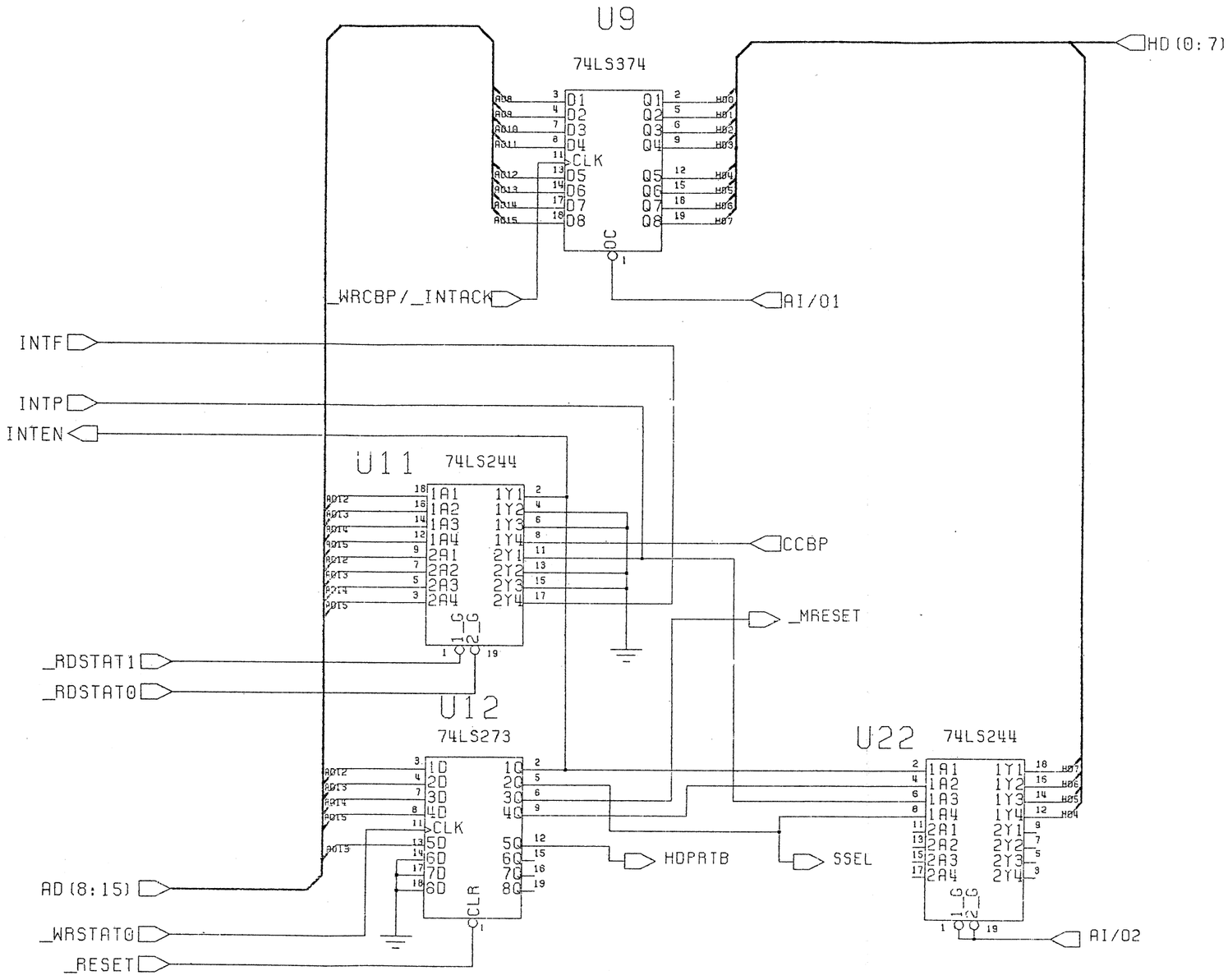


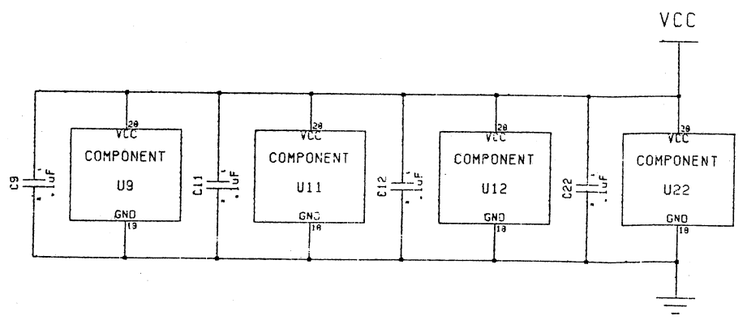




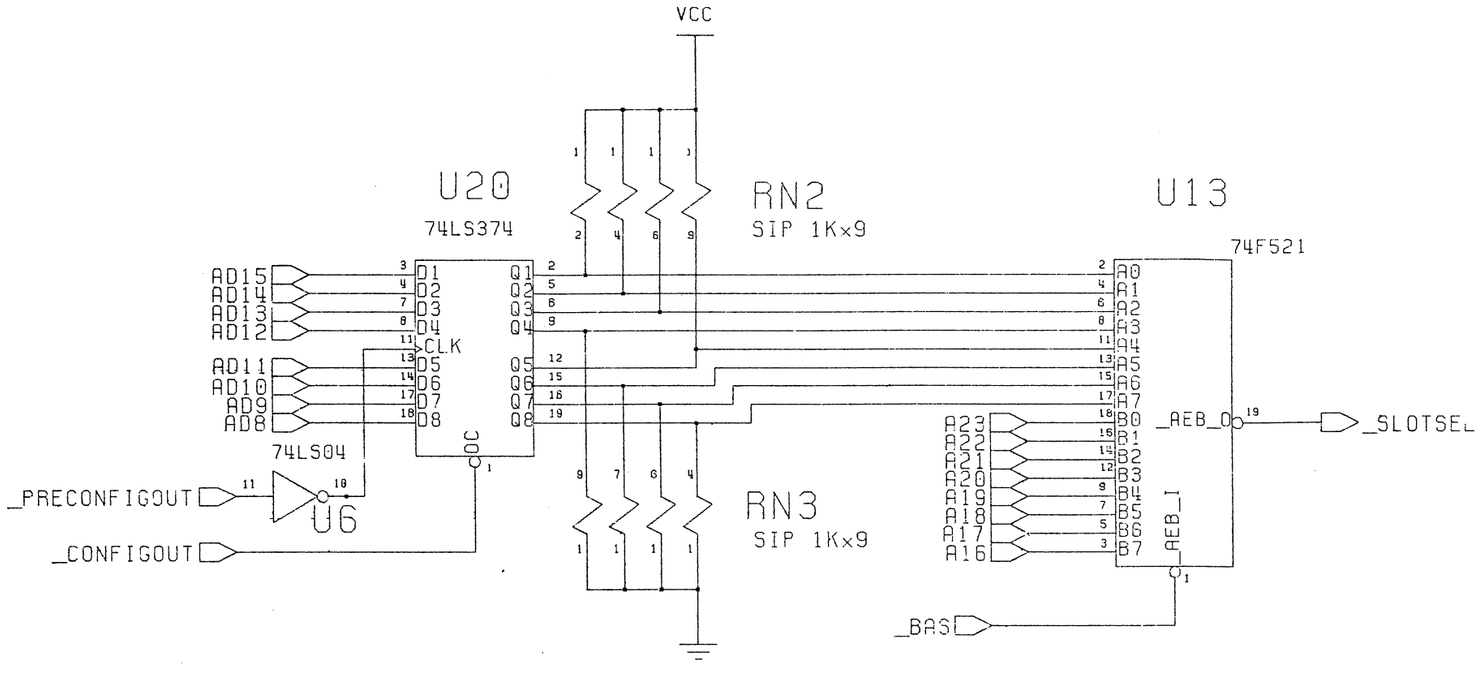
68000 EXPANSION BUSS INTERFACE

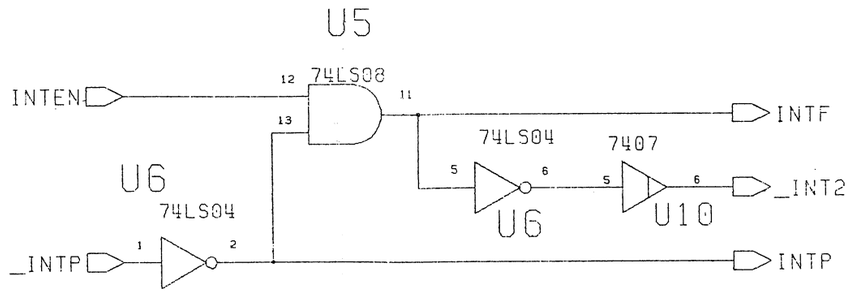
SCHEMATIC #311980 REV. A

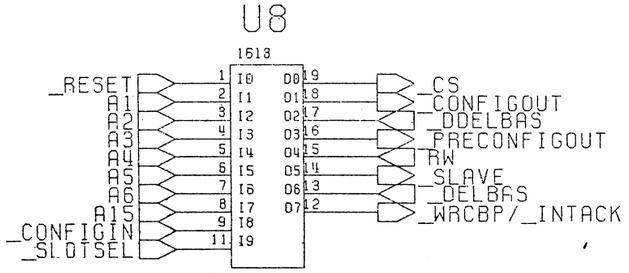


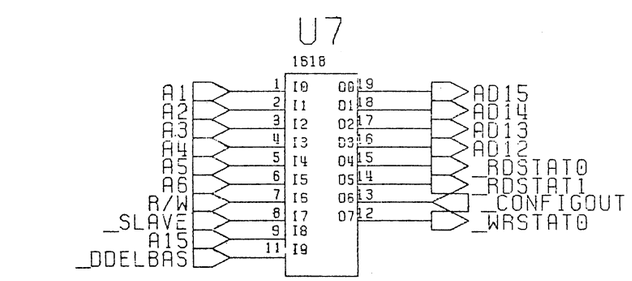


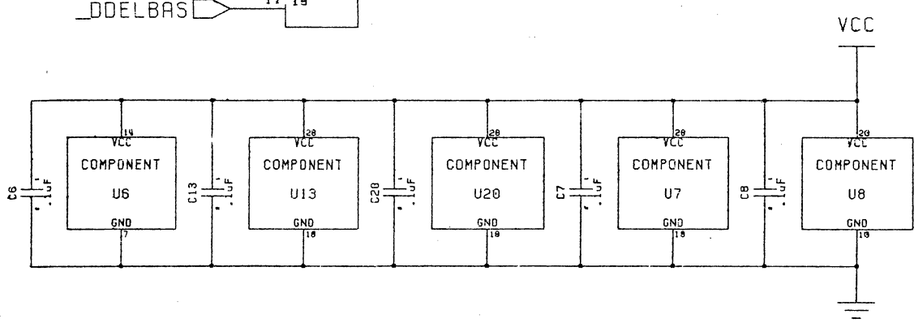
SCHEMATIC #311980 REV. A

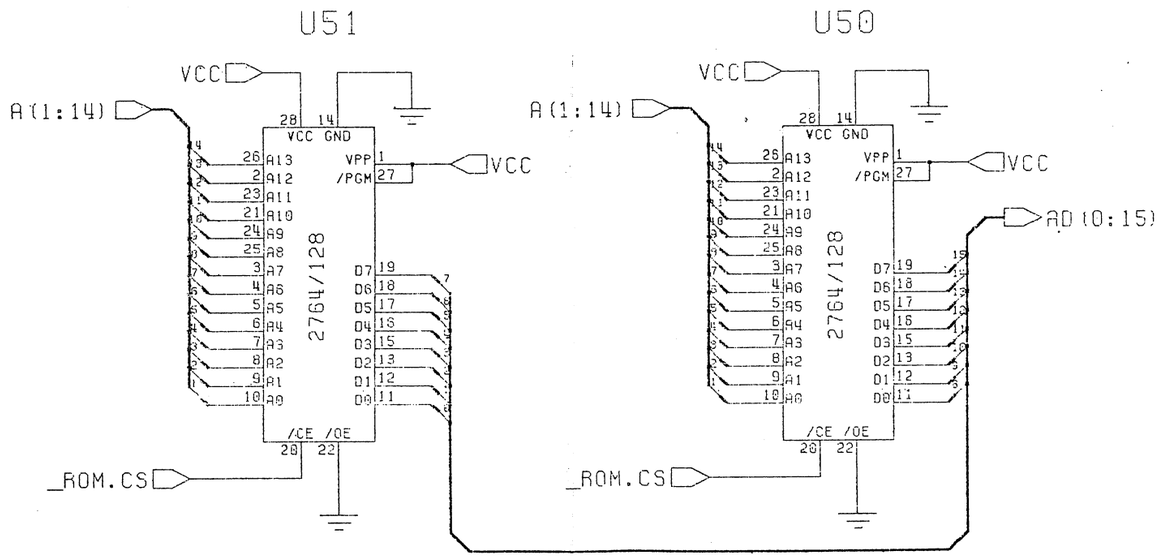






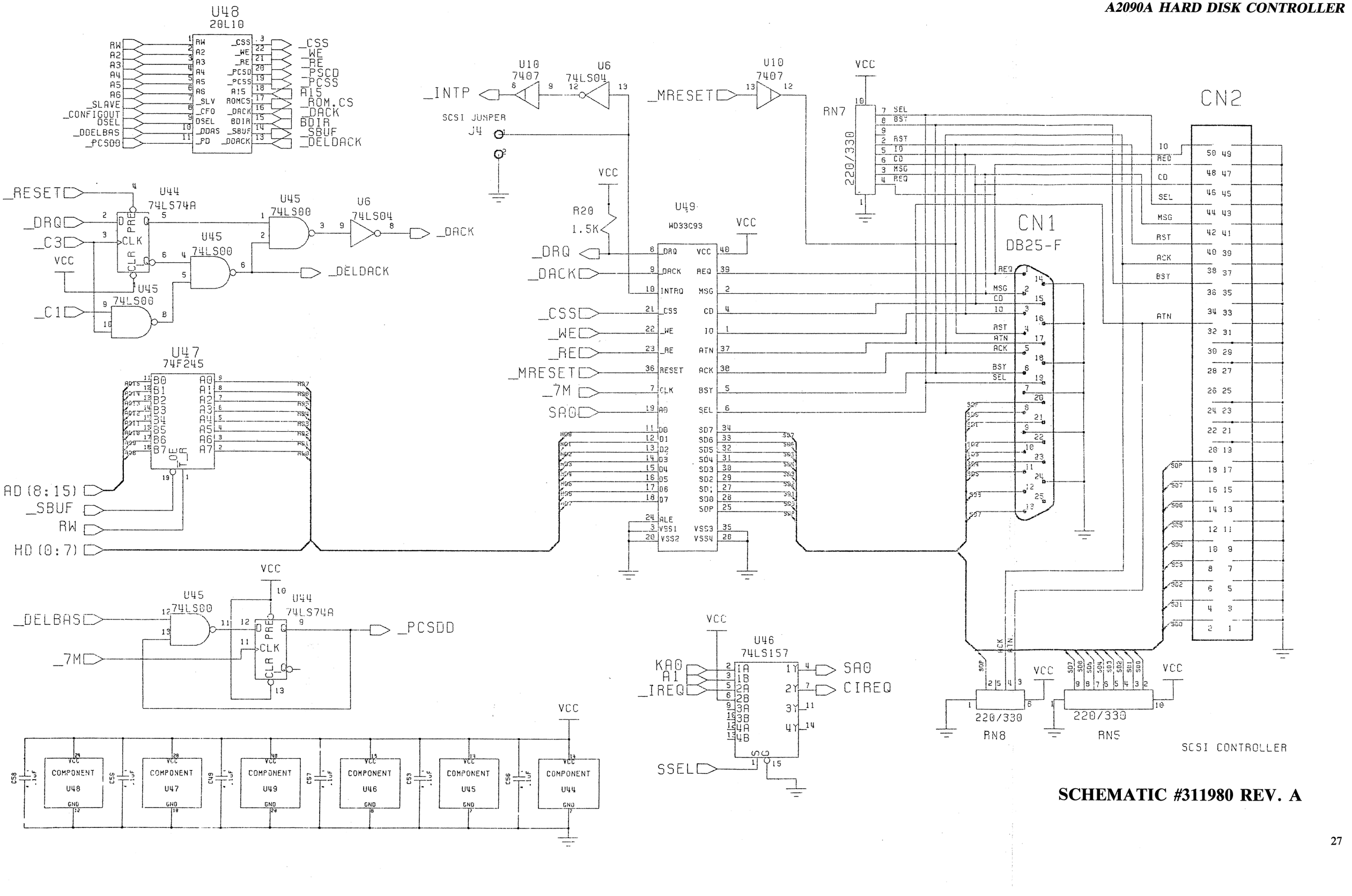




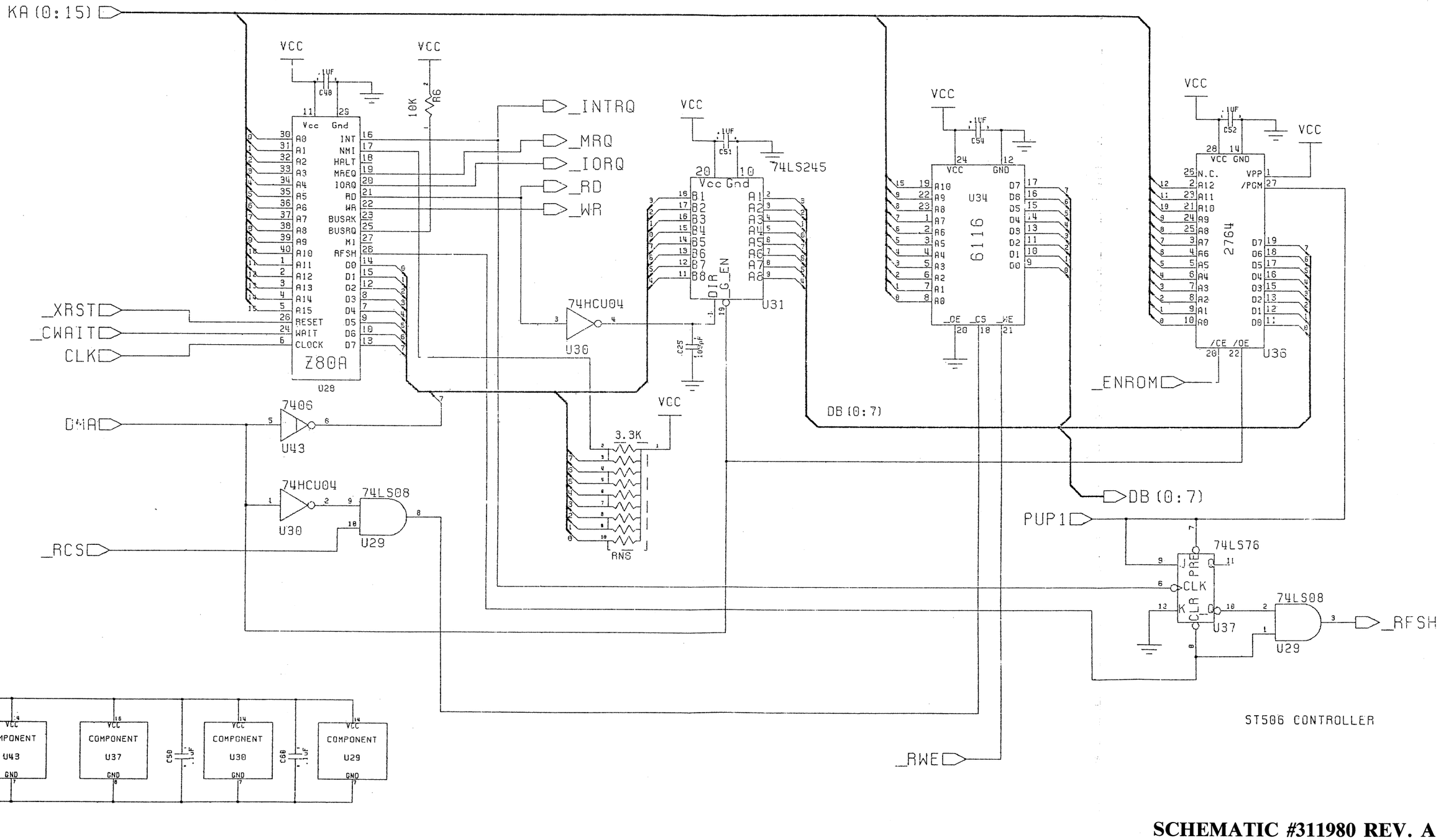


AUTO_CONFIG/BOOT_ROMS

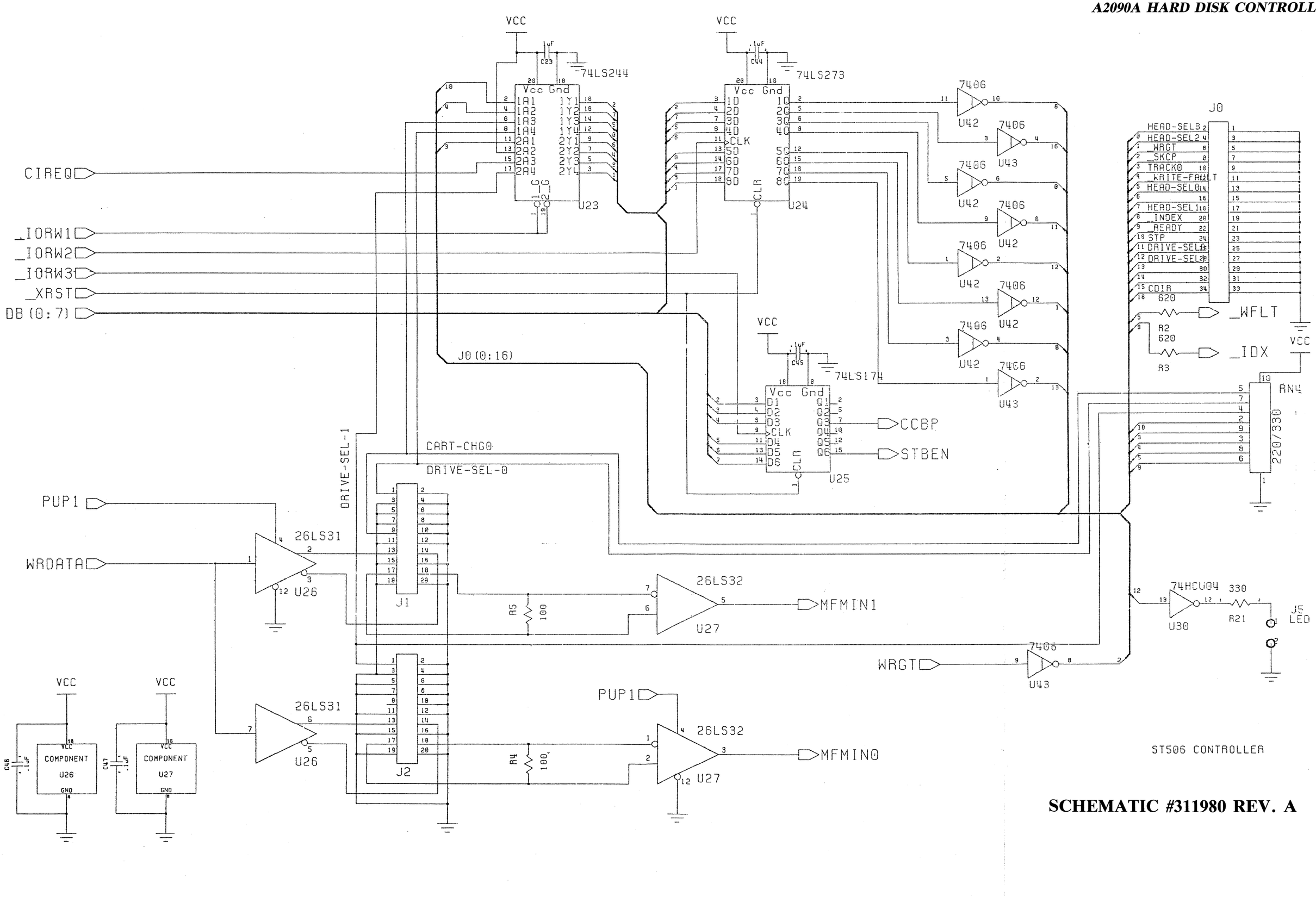
SCHEMATIC #311980 REV. A



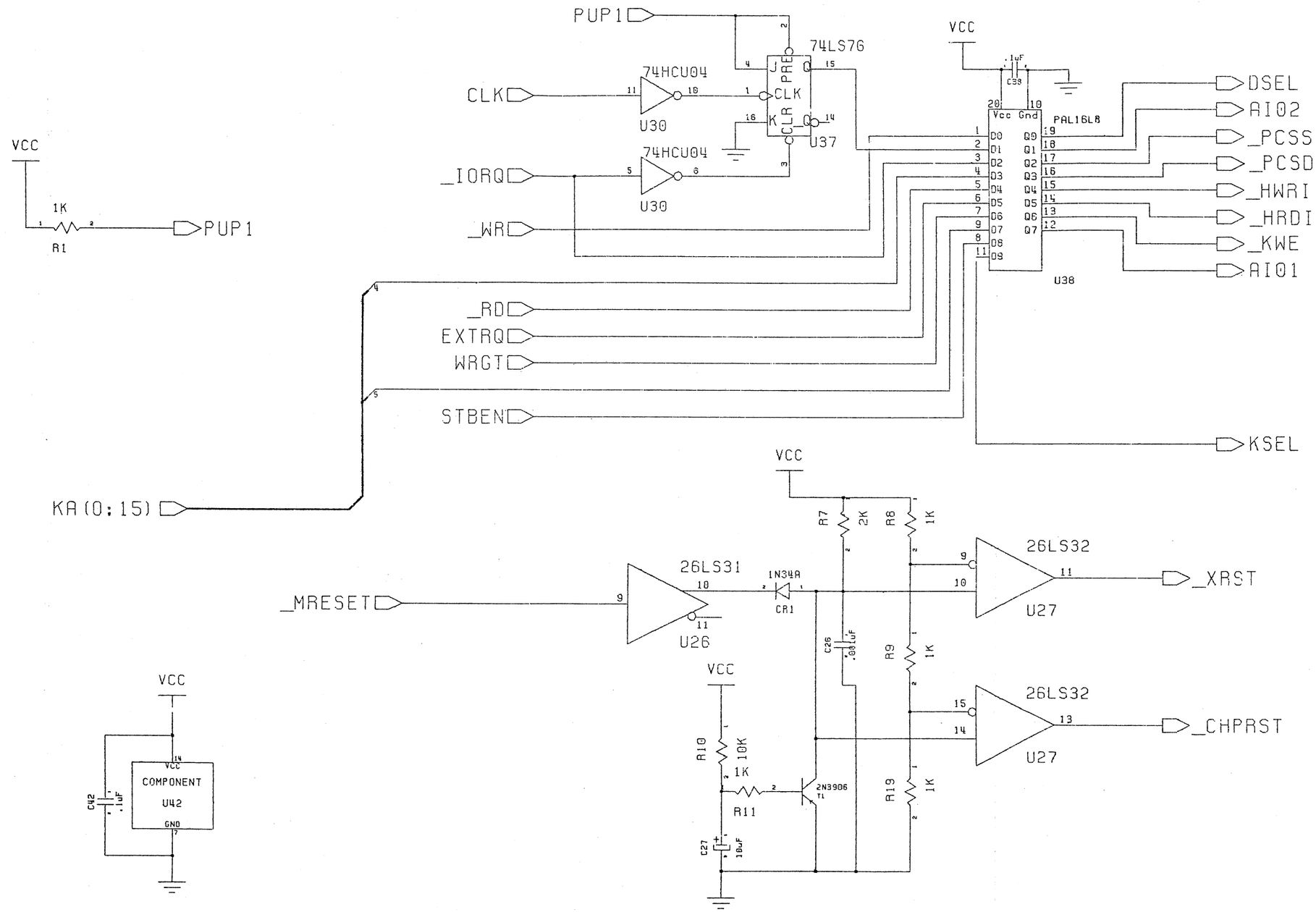
SCHEMATIC #311980 REV. A



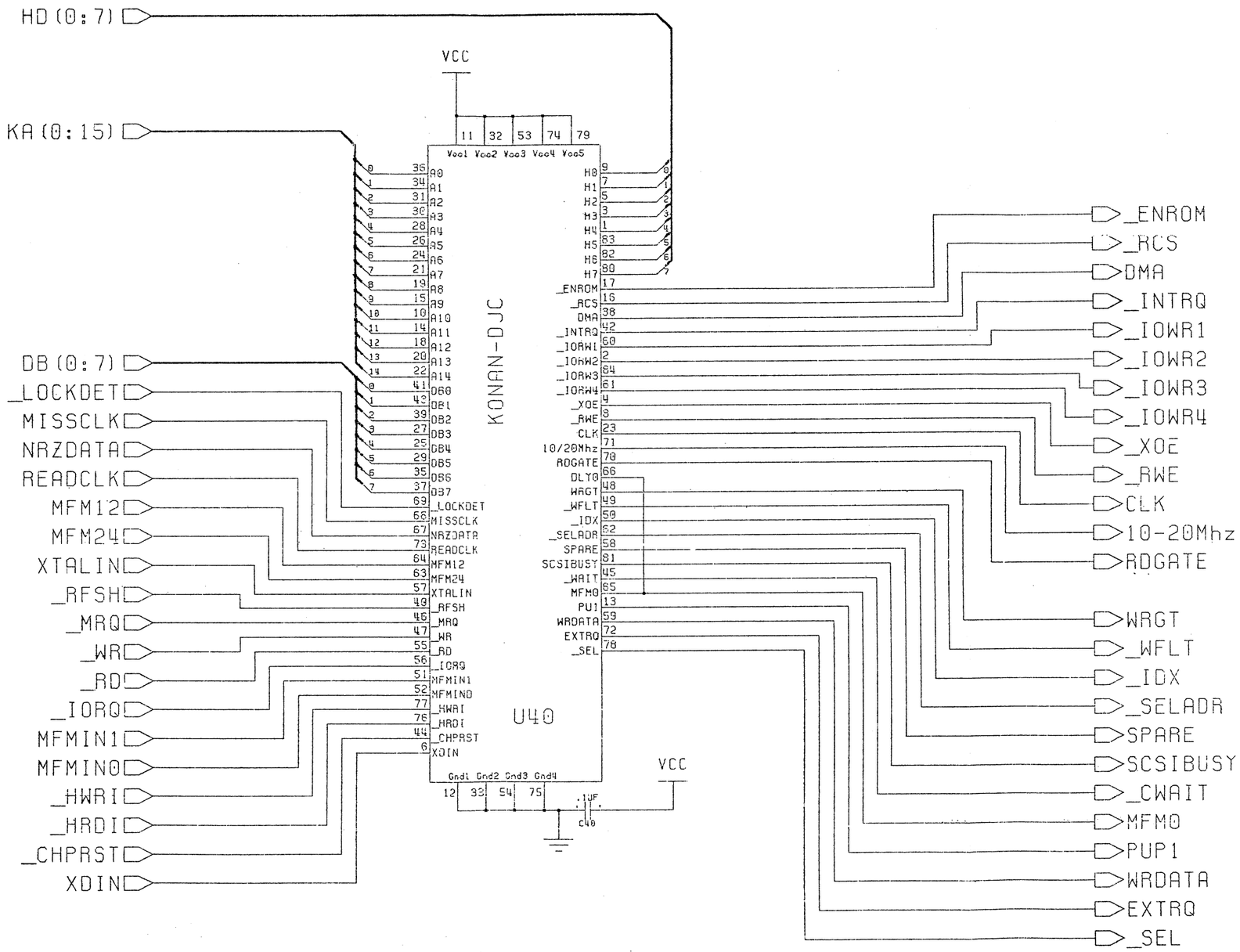
SCHEMATIC #311980 REV. A



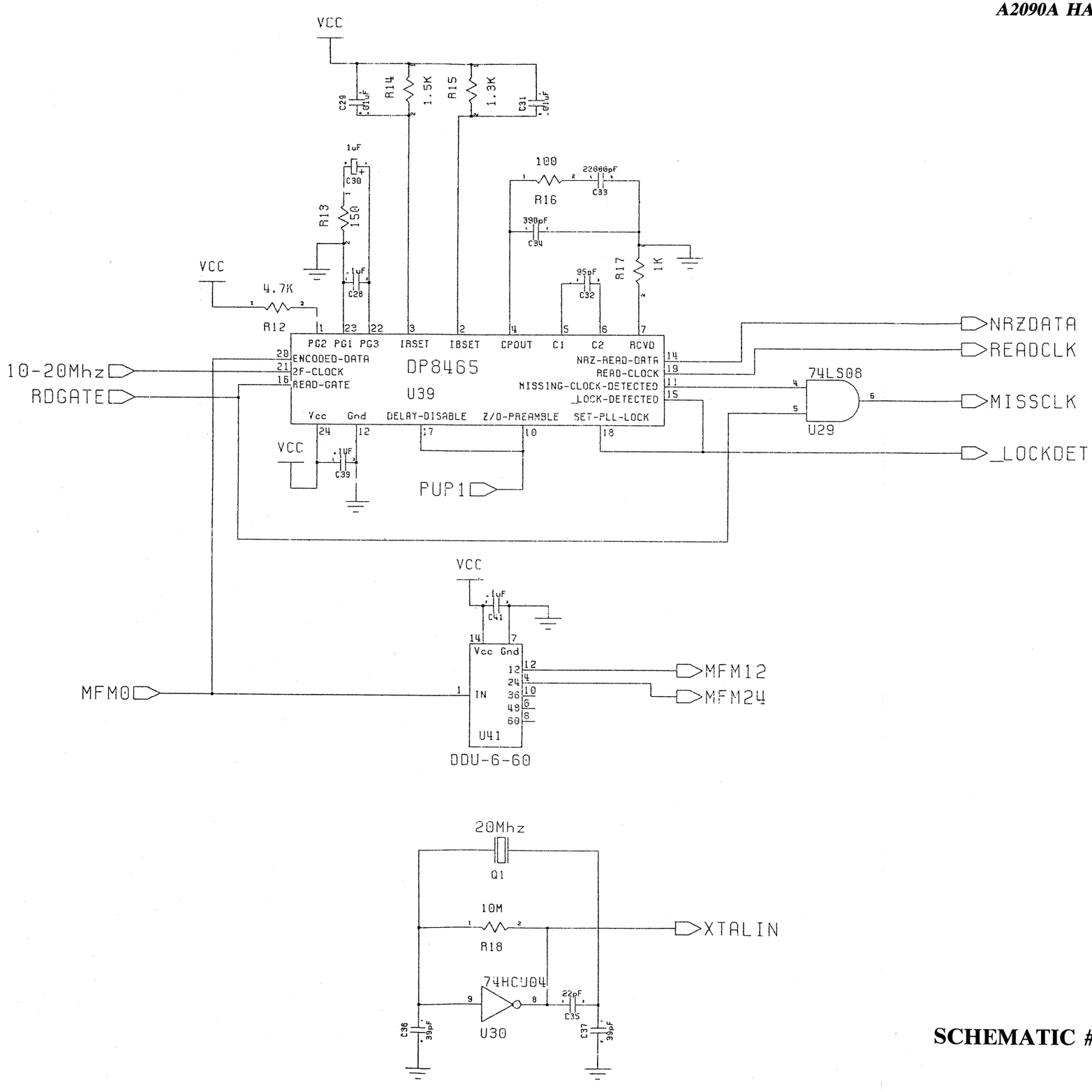
SCHEMATIC #311980 REV. A



SCHEMATIC #311980 REV. A



SCHEMATIC #311980 REV. A



SCHEMATIC #311980 REV. A