

Commodore

A 3000 T

Service Manual

07.91

PN-400407-01



Produced By:

**Commodore International Spare Parts GmbH
Braunschweig, West Germany**

SERVICE MANUAL

A3000T

JULY, 1991

PN-400407-01

INTERNATIONAL EDITION

SERVICE POLICY AND PART NUMBER INFORMATION MAY VARY ACCORDING TO COUNTRY. SOME PARTS MAY NOT BE AVAILABLE IN ALL COUNTRIES.

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

Commodore makes no express or implied warranties with regard to the information contained herein. The information is made available solely on an as is basis, and the entire risk as to completeness, reliability, and accuracy is with the user. Commodore shall not be liable for any damages in connection with the use of the information contained herein. The listing of any available replacement part herein does not constitute in any case a recommendation, warranty or guaranty as to quality or suitability of such replacement part. Reproduction or use without express permission, of editorial or pictorial content, in any matter is prohibited.

This manual contains copyrighted and proprietary information. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of Commodore Electronics Limited.

TABLE OF CONTENTS

SECTION 1 — SPECIFICATIONS

1.1	SPECIFICATIONS.....	1-1
-----	---------------------	-----

SECTION 2 — THEORY OF OPERATIONS

2.1	A3000T SYSTEM OVERVIEW.....	2-1
2.2	A3000T SYSTEM BLOCK DIAGRAM.....	2-3
2.3.1	68030 MICROPROCESSOR.....	2-4
2.3.2	68881 FLOATING-POINT COPROCESSOR.....	2-11
2.4	GARY — GATE ARRAY.....	2-16
2.5.1	ROM — READ ONLY MEMORY.....	2-19
2.5.2	RTC — REAL TIME CLOCK.....	2-21
2.6	8372 AGNUS 2MEG.....	2-22
2.7	CHIP MEMORY.....	2-26
2.8	CIA0 - 8520 (U350).....	2-28
2.9	CIA1 - 8520 (U300).....	2-29
2.10	8364 PAULA.....	2-35
2.11	8373 DENISE HI RES.....	2-36
2.12.1	AMBER.....	2-37
2.12.2	DISPLAY ENHANCER.....	2-39
2.12.3	PAL AND NTSC AMIGA GENLOCK INTERFACE GUIDELINES.....	2-41
2.13	AMIGA EXPANSION BUS.....	2-43
2.14	RESERVED FOR FUTURE USE.....	2-47
2.15	AMIGA 32 BIT LOCAL SLOT.....	2-47
2.16	BUSTER.....	2-48
2.17	SDMAC.....	2-53
2.18.1	RAMSEY	2-58
2.18.2	FAST MEMORY.....	2-61

SECTION 3 — TROUBLESHOOTING

SCHEMATIC #313180, REV. A.....	3-1
SCHEMATIC #313176, REV. A.....	3-2
SCHEMATIC #313177, REV. A.....	3-3
SCHEMATIC #313196, REV. B.....	3-4
SCHEMATIC #313179, REV. A.....	3-5
SCHEMATIC #390885, REV. B.....	3-6
SCHEMATIC #313182, REV. A.....	3-7

SECTION 4 — PARTS

A3000T PCB ASSEMBLY 313181.....	4-1
SHIPPING ASSEMBLIES.....	4-2
MAJOR ASSEMBLIES.....	4-4
PCB COMPONENTS.....	4-5

SECTION 5 — SCHEMATICS

A3000T SCHEMATIC PAGE INDEX.....	5-1
JUMPER SETTINGS.....	5-2
SCHEMATIC #313195-01, REV. A.....	5-3

SECTION 1
SPECIFICATIONS

1.1 SPECIFICATIONS

INTRODUCTION

The A3000T is a tower version of the A3000 personal computer. It employs the same processing elements as the A3000, but in a larger case supporting greater expansion capabilities. A detached keyboard and mouse are provided.

FEATURES

- 25 Mhz 68030 based CPU w/68882
- 1 Meg of Chip Ram, expandable on board to 2 Meg. (32 bit access to CPU).
- 2 Meg of 32 bit wide Fast Ram. (expandable on board to 16 Meg).
- Kickstart in 32 bit ROM.
- Integral 3.5 inch Hard Disk & SCSI controller.
- Many externally accessible drive bays including:
 - 2 3.5 inch devices oriented horizontally (One populated with one standard floppy drive)
 - 2 5.25 inch half height bays oriented vertically (Can be used as a single full height bay)
 - 2 5.25 inch half height bays oriented horizontally (Can be used as a single full height bay).
- Internally accessible drive bays including:
 - 1 Full height 5.25 inch bay adjustable for 3.5 inch devices
- Real Time clock
- On board de-interlacer/scan doubler
- 5 100 pin amiga slots, 2 of which are in line with PCAT slots, and one of which is in line with a new formfactor video slot
- 2 PCAT only slots.

PHYSICAL SIZE

Height: 24 inches
Depth: 17.5 inches
Width: Tower: 7 inches
Feet: 15 inches
Weight: 35 lbs

SYSTEM IO

The IO capabilities of the A3000T are as follows:

EXTERNAL SYSTEM IO

Serial/Parallel/Mouse/Joystick/Keyboard/Stereo Audio Ports

These ports remain unchanged from their A2000 counterparts.

SCSI connector

This connector is the same as that found on the back of the 2090 series of hard disk controllers. It allows communication to a variety of devices including hard disks, optical drives, and laser printers.

Video & VGA connectors

As in all Amigas, the A3000T supplies the standard 23 pin video connector. In addition, the on board flicker fixer provides a standard 15 pin VGA connnecter.

INTERNAL SYSTEM I/O

The following connectors comprise the internal IO capabilities of the machine:

- SCSI hard drive ribbon cable header.
- 7 expansion slots:
 - 1 100 Zorro-III slot inline with Commodore Amiga Video slot
 - 2 100 pin Zorro-III slots
 - 2 100 pin Zorro-III inline with PC/AMIGA bridgecard slots
 - 2 PC/AT only slots.
- Socketed chip memory expansion good for upgrade from 1 to 2 meg of chip.
- Socketed fast RAM expansion for upgrade to 16 Meg.

SECTION 2
THEORY OF OPERATIONS

2.1 A3000T SYSTEM OVERVIEW

CUSTOM CHIPS

The A3000T CPU will require the use of more custom integrated circuitry than any previous Commodore product. In addition to the 3 chip Amiga chipset, there are 5 new custom chips.

These include the system glue chip (FATGARY), a new expansion buss controller (FATBUSTER), a new ram controller with DMA support (RAMSEY), a new improved disk IO controller (SDMAC), and the flicker fixer gate array controller part (AMBER).

FATGARY

As the name implies, this chip is a larger version of the gary chip originally used in the A500 CPU design. Fat gary provides chip select and DASCK support for motherboard resources such as the 8520's, RTC, chipram, roms, and SDMAC.

FATBUSTER

Fat Buster is a new, larger version of the A2000 chip of the same name. As before, it is responsible for expansion buss control and arbitration. However, the new buster also maps 32 bit 68030 memory cycles onto standard A2000 16 bit expansion buss cycles. Fat Buster also provides extensions to the standard A2000 16 bit buss protocol which allow full 32 bit address and data DMA to occur to and from expansion cards. Thus FATBUSTER allows us to max and match existing 16 bit A2000 expansion cards, and a new 32 bit card format on the same expansion buss in a wholly compatible manner.

RAMSEY

RAMSEY is a dual function chip. It provides support for up to 16 megabytes of on board 32 bit wide ram, and provides address generation support for the super DMAC.

SDMAC

SDMAC is a new improved version of the 4701 series of disk & SCSI DMA controllers. SDMAC allows us to provide 32 bit Direct Disk DMA. It is felt that this will yield greatly improved system throughput for large applications, or disk intensive operating systems (read UNIX).

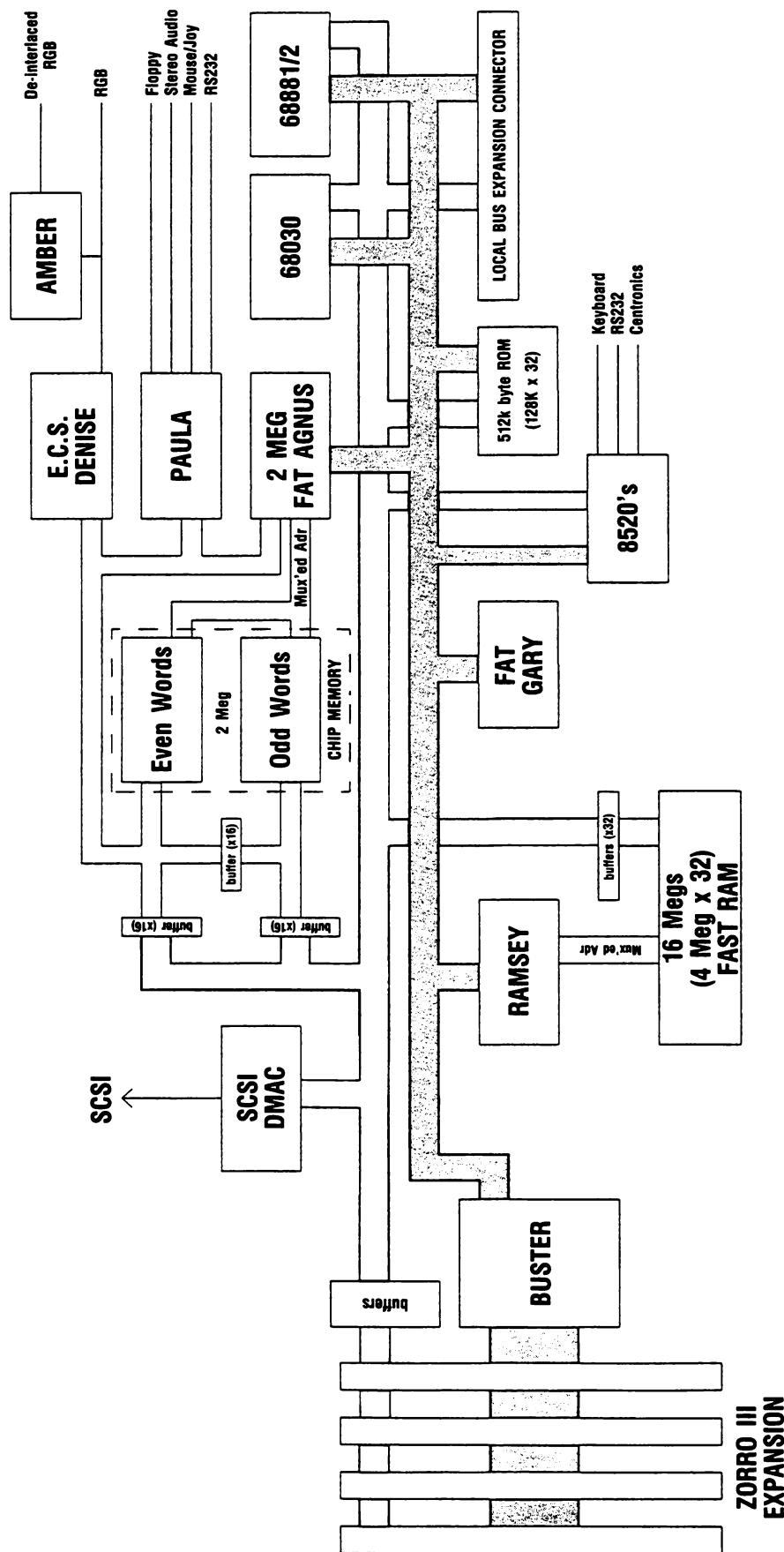
AMBER

Amber is the name of the flicker fixer controller gate array. This device, in conjunction with some specialized video memories, effectively provides a flicker fixer at low cost. This essential capability is to be option due to the sensitive cost of the specialized video drams required.

2.1 A3000T SYSTEM OVERVIEW (Continued)**MEMORY MAP**

The A3000T makes use of the full 32 bit addressing capability of the MC68030. In the lower 16 Meg, (the only region addressable by the 68000), the memory map is basically unchanged from that of the A2000.

\$000000-	\$1FFFFFF	Up to 2 Megs of Chip Memory
\$200000-	\$9FFFFFF	Auto Config Expansion space
\$A00000-	\$BFFFFFF	IO Space (8520s)
\$C00000-	\$D7FFFF	Location of A500 Magic \$c00000 memory. Unused in this machine.
\$D80000-	\$D9FFFF	Custom chip register mirror or real time clock
\$DA0000-	\$DBFFFF	Unused
\$DC0000-	\$DDFFFF	Real Time Clock
\$DE0000-	\$DFFFFFF	Custom chip registers
\$E00000-	\$E80000	Unused
\$E80000-	\$E8FFFF	Autoconfig slot
\$E90000-	\$EFFFFFF	64k autoconfig slots
\$F00000-	\$FFFFFFF	Rom space
\$01000000-	\$07FFFFFF	On board 32 bit ram space. This grows down from \$07FFFFFF.
\$08000000-	\$0FFFFFFF	32 bit ram expansion space. This grows up from \$08000000.
\$10000000-\$	\$EFFFFFFF	32 bit ‘other’ expansion space.



2.3.1 68030 MICROPROCESSOR

The MC68030 is a second-generation full 32-bit enhanced microprocessor from Motorola. The MC68030 is a member of the M68000 Family of devices that combines a central processing unit (CPU) core, a data cache, an instruction cache, an enhanced bus controller, and a memory management unit in a single VLSI device. The processor is designed to operate at clock speeds beyond 20 MHz. The MC68030 is implemented with 32-bit registers and data paths, 32-bit addresses, a rich instruction set, and versatile addressing modes.

The MC68030 is upward object code compatible with the earlier members of the M68000 Family and has the added features of an on-chip memory management unit, a data cache, and an improved bus interface. It retains the flexible coprocessor interface pioneered in the MC68020 and provides full IEEE floating-point support through this interface with the MC68881 or MC68882 Floating-Point Coprocessor. Also, the internal functional blocks of this microprocessor are designed to operate in parallel, allowing instruction execution to be overlapped. In addition to instruction execution, the internal caches, the on-chip memory management unit, and the external bus controller all operate in parallel.

The MC68030 fully supports the non-multiplexed bus structure of the MC68020, with 32 bits of address and 32 bits of data. The MC68030 bus has an enhanced controller that supports both asynchronous and synchronous bus cycles and burst data transfers. It also supports the MC68020 dynamic bus sizing mechanism that automatically determines device port sizes on a cycle-by-cycle basis as the processor transfers operands to or from external devices.

A block diagram of the MC68030 is shown in Figure 1-1. The instructions and data required by the processor are supplied from the internal caches whenever possible. The memory management unit (MMU) translates the logical address generated by the processor into a physical address utilizing its address translation cache (ATC). The bus controller manages the transfer of data between the CPU and memory or devices at the physical address.

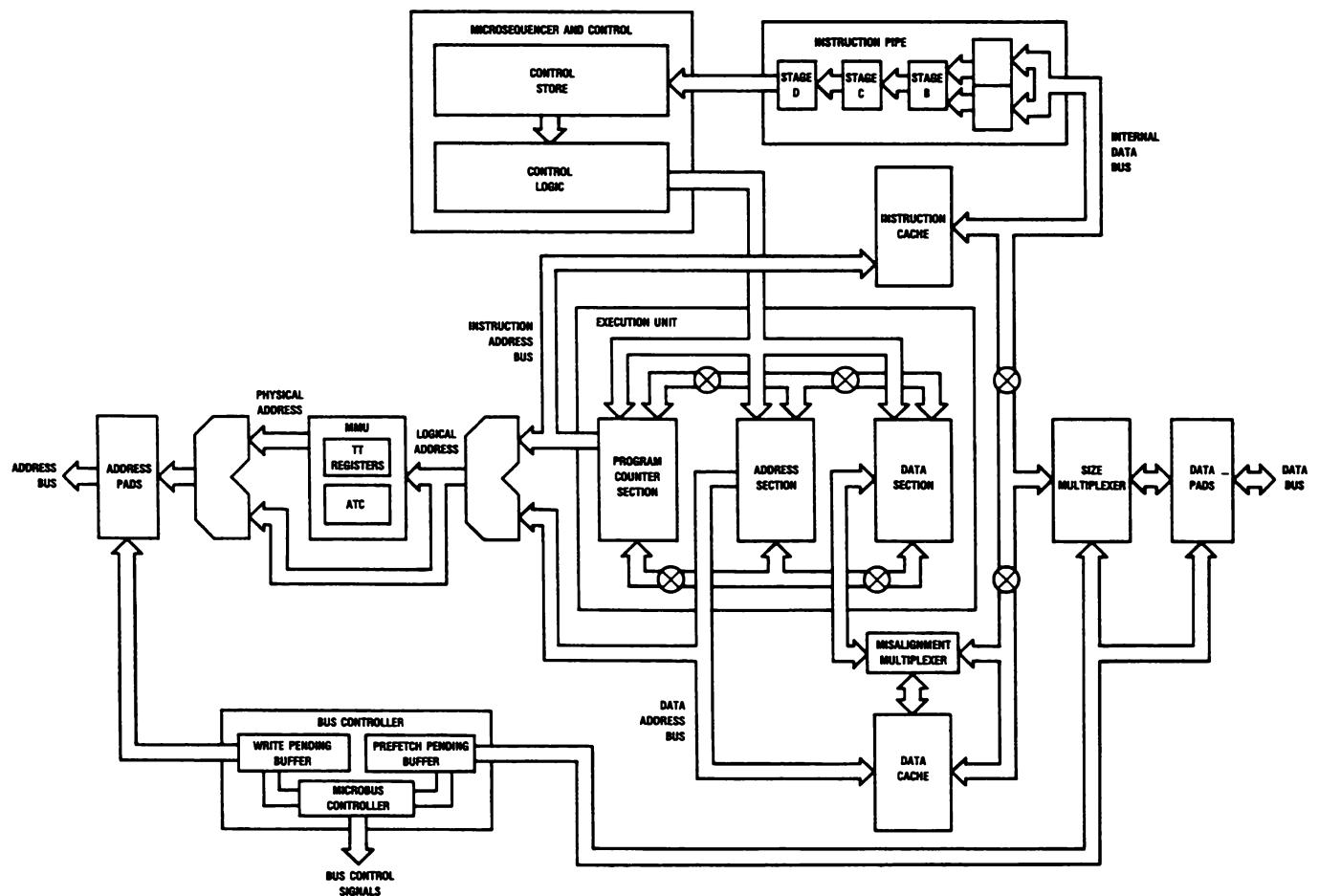


Figure 1-1. MC68030 Block Diagram

2.3.1 68030 MICROPROCESSOR (Continued)

FEATURES

The features of the MC68030 microprocessor are:

- Object Code Compatible with the MC68020 and Earlier M68000 Microprocessors
- Complete 32-Bit Non-Multiplexed Address and Data Buses
- Sixteen 32-Bit General Purpose Data and Address Registers
- Two 32-Bit Supervisor Stack Pointers and Ten Special Purpose Control Registers
- 256-Byte Instruction Cache and 256-Byte Data Cache that can be Accessed Simultaneously
- Paged Memory Management Unit that Translates Addresses in Parallel with Instruction Execution and Internal Cache Accesses
- Two Transparent Segments Allow Untranslated Access to Physical Memory to be Defined for Systems that Transfer Large Blocks of Data between Predefined Physical Addresses, e.g., Graphics Applications (Amiga — Chip/Fast RAM)
- Pipelined Architecture with Increased Parallelism Allows Accesses to Internal Caches to Occur in Parallel with Bus Transfers and Instruction Execution to be Overlapped
- Enhanced Bus Controller Supports Asynchronous Bus Cycles (three clocks minimum), Synchronous Bus Cycles (two clocks minimum), and Burst Data Transfers (one clock minimum) all to the Physical Address Space
- Dynamic Bus Sizing Supports 8-, 16-, 32-Bit Memories and Peripherals
- Support for Coprocessors with the M68000 Coprocessor Interface; e.g., Full IEEE Floating-Point Support Provided by the MC68881/MC68882 Floating-Point Coprocessors
- 4-Gigabyte Logical and Physical Addressing Range
- Implemented in Motorola's HCMOS Technology that Allows CMOS and HMOS (High Density NMOS) Gates to be Combined for Maximum Speed, Low Power, and Optimum Die Size
- Processor Speeds Beyond 20 MHz

Both improved performance and increased functionality result from the on-chip implementation of the MMU and the data and instruction caches. The enhanced bus controller and the internal parallelism also provides increased system performance. Finally, the improved bus interface, the reduction in physical size, and the lower power consumption combine to reduce system costs and satisfy cost/performance goals of the system designer.

MC68030 EXTENSIONS TO THE M68000 FAMILY

In addition to the on-chip instruction cache present in the MC68020, the MC68030 has an internal data cache. Data that is accessed during read cycles may be stored in the on-chip cache, where it is available for subsequent accesses. The data cache reduces the number of external bus cycles when the data operand required by an instruction, is already in the data cache.

Performance is enhanced further because the on-chip caches can be internally accessed in a single clock cycle. In addition, the bus controller provides a two-clock cycle synchronous mode and burst mode accesses that can transfer data in as little as one clock per long word.

The MC68030 enhanced microprocessor contains an on-chip memory management unit that allows address translation to operate in parallel with the CPU core, the internal caches, and the bus controller.

Additional signals support emulation and system analysis. External debug equipment can disable the on-chip caches and the MMU in order to freeze the MC68030 internal state during breakpoint processing. In addition, the MC68030 indicates:

1. The start of a refill of the instruction pipe
2. Instruction boundaries
3. Pending trace or interrupt processing
4. Exception processing
5. Halt conditions

This status and control information allows external debugging equipment to trace the MC68030 activity and interact nonintrusively with the MC68030 to effectively reduce system debug effort.

2.3.1 68030 MICROPROCESSOR (Continued)

PROGRAMMING MODEL

The programming model of the MC68030 consists of two groups of registers: the user model, and the supervisor model. This corresponds to the user and supervisor privilege levels. User programs, executing at the user privilege level, can only use the registers of the user model. System software executing at the supervisor level uses the control registers of the supervisor level to perform supervisor functions.

Figure 1-2 shows the user programming model, consisting of sixteen 32-bit general-purpose registers and 2 control registers:

- General-purpose 32-bit registers D0-D7, A0-A7
- 32-bit program counter PC
- 8-bit condition code register

The supervisor programming model consists of the registers available to the user plus 14 control registers:

- Two 32-bit supervisor stack pointers ISP and MSP
- 16-bit status register SR
- 32-bit vector base register VBR
- 32-bit alternate function code registers SFC and DFC
- 32-bit cache control register CACR
- 32-bit cache address register CAAR
- 64-bit CPU root pointer CRP
- 64-bit supervisor root pointer SRP
- 32-bit translation control register TC
- 32-bit transparent translation registers TT0 and TT1
- 16-bit MMU status register MMUSR

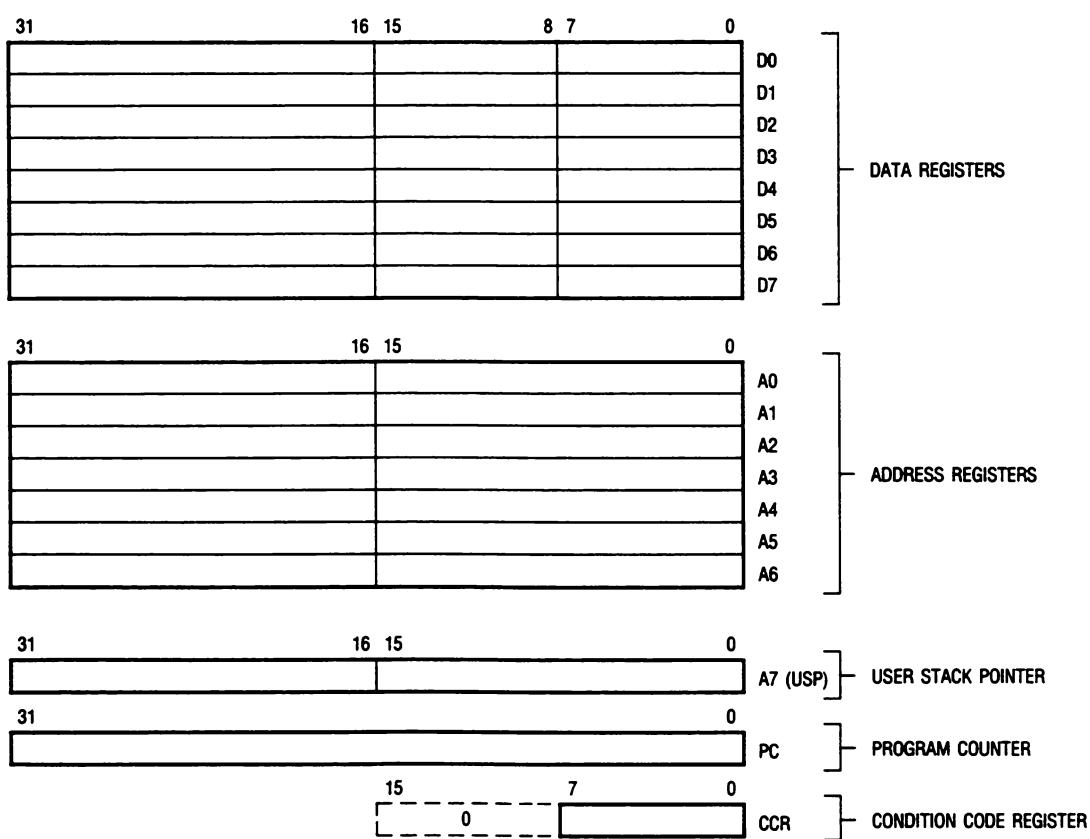


Figure 1-2. User Programming Model

2.3.1 68030 MICROPROCESSOR (Continued)

PROGRAMMING MODEL (Continued)

The user programming model remains unchanged from previous M68000 Family microprocessors. The supervisor programming model supplements the user programming model and is used exclusively by the MC68030 system programmers who utilize the supervisor privilege level to implement sensitive operating system functions, I/O control, and memory management subsystems. Here, in the supervisor programming model, are all the controls to access and enable the special features of the MC68030. This segregation was carefully planned so that all application software is written to run at the non-privileged user level and migrates to the MC68030 from any M68000 platform without modification. Since system software is usually modified by system programmers when ported to a new design, the control features are properly placed in the supervisor programming model. For example, the transparent translation feature of the MC68030 is new to the family supervisor programming model for the MC68030 and the two translation registers are new additions to the family supervisor programming model for the MC68030. Only supervisor code uses this feature and user application programs remain unaffected.

Registers D0-D7 are used as data registers for bit and bit field (1 to 32 bits), byte (8-bit), word (16-bit), long word (32-bit), and quad word (64-bit) operations. Registers A0-A6 and the user, interrupt, and master stack pointers are address registers that may be used as software stack pointers or base address registers. Register A7 (shown as A7', and A7'' in Figure 1-3) is a register designation that applies to the user stack pointer in the user privilege level, and to either the interrupt or master stack pointer in the supervisor privilege level. In the supervisor privilege level, the active stack pointer (interrupt or master) is called the supervisor stack pointer (SSP). In addition, the address registers may be used for word and long word operations. All of the 16 general purpose registers (D0-D7, A0-A7) may be used as index registers.

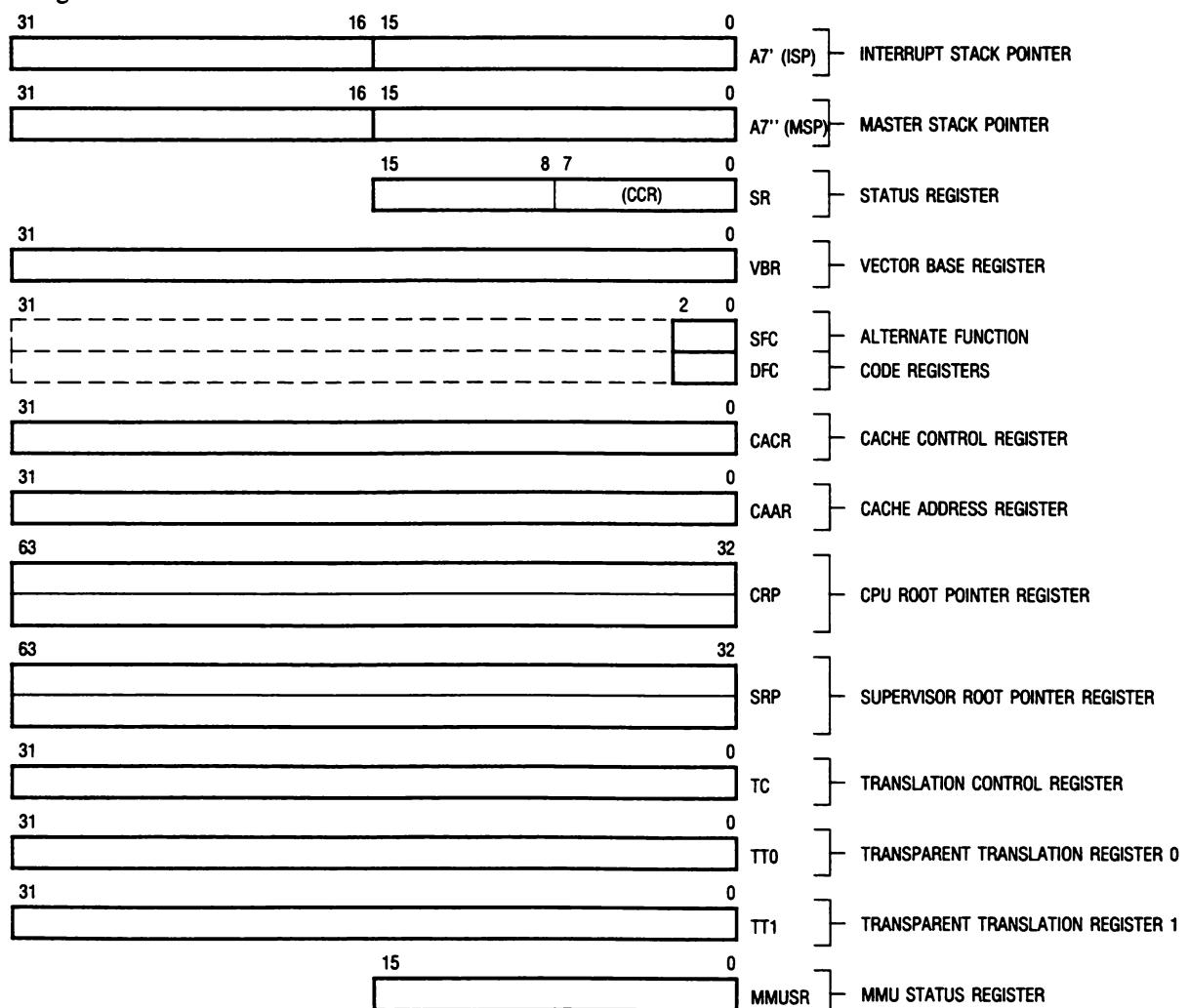


Figure 1-3. Supervisor Programming Model Supplement

2.3.1 68030 MICROPROCESSOR (Continued)

PROGRAMMING MODEL (Continued)

The program counter (PC) contains the address of the next instruction to be executed by the MC68030. During instruction execution and exception processing, the processor automatically increments the contents of the PC or places a new value in the PC, as appropriate.

The status register, SR, (Figure 1-4) stores the processor status. It contains the condition codes that reflect the results of a previous operation and can be used for conditional instruction execution in a program. The condition codes are: extend (X), negative (N), zero (Z), overflow (V), and carry (C). The user byte containing the condition codes is the only portion of the status register information available in the user privilege level, and it is referenced as the CCR in user programs. In the supervisor privilege level, software can access the full status register, including the interrupt priority mask (three bits) as well as additional control bits. These bits indicate whether the processor is in:

1. One of two trace modes (T1, T0)
2. Supervisor or user privilege level (S)
3. Master or interrupt mode (M)

The vector base register (VBR) contains the base address of the exception vector table in memory. The displacement of an exception vector is added to the value in this register to access the vector table.

Alternate function code registers SFC and DFC contain 3-bit function codes. Function codes can be considered extensions of the 32-bit linear address that optionally provide as many as eight 4-gigabyte address spaces. Function codes are automatically generated by the processor to select address spaces for data and program at the user and supervisor privilege levels, and a CPU address space used for processor functions (for example, coprocessor communications). Registers SFC and DFC are used by certain instructions to explicitly specify the function codes for operations.

The cache control register (CACR) controls the on-chip instruction and data caches of the MC68030. The cache address register (CAAR) stores an address for cache control functions.

The CPU root pointer (CRP) contains a pointer to the root of the translation tree for the currently executing task of the MC68030. This tree contains the mapping information for the task's address space. When the MC68030 is configured to provide a separate address space for supervisor routines, the supervisor root pointer (SRP) contains a pointer to the root of the translation tree describing the supervisor's address space.

The translation control register (TC) consists of several fields that control address translation. These fields enable and disable address translation, enable and disable the use of SRP for the supervisor address space, and select or ignore the function codes in translating addresses. Other fields define the size of memory pages, the number of address bits used in translation, and the translation table structure.

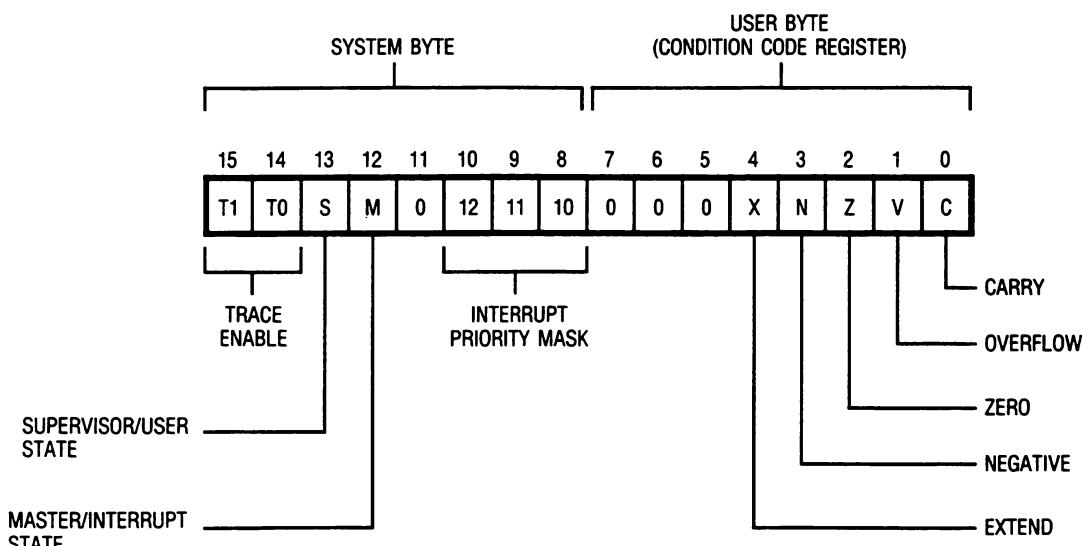


Figure 1-4. Status Register

2.3.1 68030 MICROPROCESSOR (Continued)

PROGRAMMING MODEL (Continued)

The transparent translation registers, TT0 and TT1, can each specify separate blocks of memory as directly accessible without address translation. Logical addresses in these areas become the physical addresses for memory access. Function codes and the eight most significant bits of the address can be used to define the area of memory and type of access; either read, write, or both types of memory access can be directly mapped. The transparent translation feature allows rapid movement of large blocks of data in memory or I/O space without disturbing the context of the on-chip address translation cache or incurring delays associated with translation table lookups. This feature is useful to graphics, controller, and real time applications.

The MMU status register, MMUSR, contains memory management status information resulting from a search of the address translation cache or the translation tree for a particular logical address.

THE MEMORY MANAGEMENT UNIT

The memory management unit (MMU) supports virtual memory systems by translating logical addresses to physical addresses using translation tables stored in memory. The MMU stores address mappings in an address translation cache (ATC) that contains the most-recently-used translations. When the ATC contains the address for a bus cycle requested by the CPU, a translation table search is not performed. Features of the MMU include:

- Multiple Level Translation Tables with Short and Long Format Descriptors for Efficient Table Space Usage
- Table Searches Automatically Performed in Microcode
- 22-Entry Fully-Associative ATC
- Address Translations and Internal Instruction and Data Cache Accesses Performed in Parallel
- Eight Page Sizes Available Ranging from 256 to 32k Bytes
- Two Optional Transparent Blocks
- User and Supervisor Root Pointer Registers
- Write Protection and Supervisor Protection Attributes
- Translations Enabled/Disabled by Software
- Translations can be Disabled with External MMUDIS Signal
- Used and Modified Bits Automatically Maintained in Tables and ATC
- Cache Inhibit Output (CIOUT) Signal can be Asserted on a Page-by-Page Basis
- 32-Bit Internal Logical Address with Capability to Ignore as many as 15 Upper Address Bits
- 3-bit Function Code Supports Separate Address Spaces
- 32-Bit Physical Address

The memory management function performed by the MMU is called demand paged memory management. Since a task specifies the areas of memory it requires as it executes, memory allocation is supported on a demand basis. If a requested access to memory is not currently mapped by the system, then the access causes a demand for the operating system to load or allocate the required memory image. The technique used by the MC68030 is paged memory management because physical memory is managed in blocks of a specified number of bytes, called page frames. The logical address space is divided into fixed-size pages that contain the same number of bytes as the page frames. Memory management assigns a physical base address to a logical page. The system software then transfers data between secondary storage and memory one or more pages at a time.

2.3.1 68030 MICROPROCESSOR (Continued)

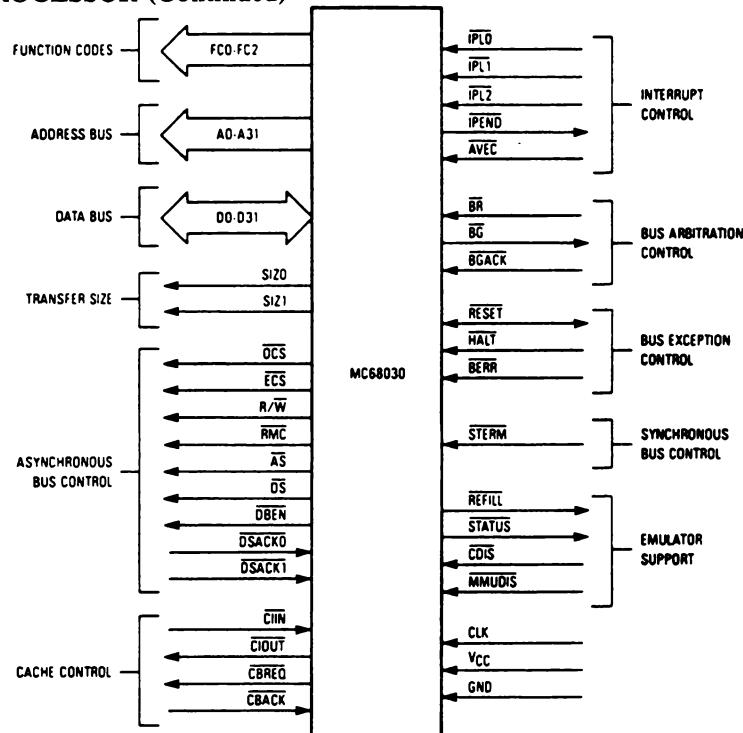


Figure 1-5. 68030 Functional Signal Groups

Signal Function	Signal Name	Input/Output	Active State	Three-State
Function Codes	FC0-FC2	Output	High	Yes
Address Bus	AO-A31	Output	High	Yes
Data Bus	D0-D31	Input/Output	High	Yes
Transfer Size	SIZ0/SIZ1	Output	High	Yes
Operand Cycle Start	OCS	Output	Low	No
External Cycle Start	ECS	Output	Low	No
Read/Write	R/W	Output	High/Low	Yes
Read-Modify-Write Cycle	RMC	Output	Low	Yes
Address Strobe	AS	Output	Low	Yes
Data Strobe	DS	Output	Low	Yes
Data Buffer Enable	DBEN	Output	Low	Yes
Data Transfer and Size Acknowledge	DSACK0/DSACK1	Input	Low	—
Synchronous Termination	STERM	Input	Low	—
Cache Inhibit In	CIN	Input	Low	—
Cache Inhibit Out	CIOUT	Output	Low	Yes
Cache Burst Request	CBREQ	Output	Low	Yes
Cache Burst Acknowledge	CBACK	Input	Low	—
Interrupt Priority Level	IPLO-IPL2	Input	Low	—
Interrupt Pending	IPEND	Output	Low	No
Autovector	AVEC	Input	Low	—
Bus Request	BR	Input	Low	—
Bus Grant	BG	Output	Low	No
Bus Grant Acknowledge	BGACK	Input	Low	—
Reset	RESET	Input/Output	Low	No
Halt	HALT	Input	Low	—
Bus Error	BERR	Input	Low	—
Cache Disable	CDIS	Input	Low	—
MMU Disable	MMUDIS	Input	Low	—
Pipeline Refill	REFILL	Output	Low	No
Microsequencer Status	STATUS	Input	Low	No
Clock	CLK	Input	—	—
Power Supply	VCC	Input	—	—
Ground	GND	Input	—	—

Figure 1-6. 68030 Signal Overview

2.3.2 68881 FLOATING-POINT COPROCESSOR

MC68881 FPCP

This section contains a brief description of the input and output signals for the MC68881/MC68882 (FPCP) floating-point coprocessor. The signals are functionally organized into groups as shown in Figure 1-7.

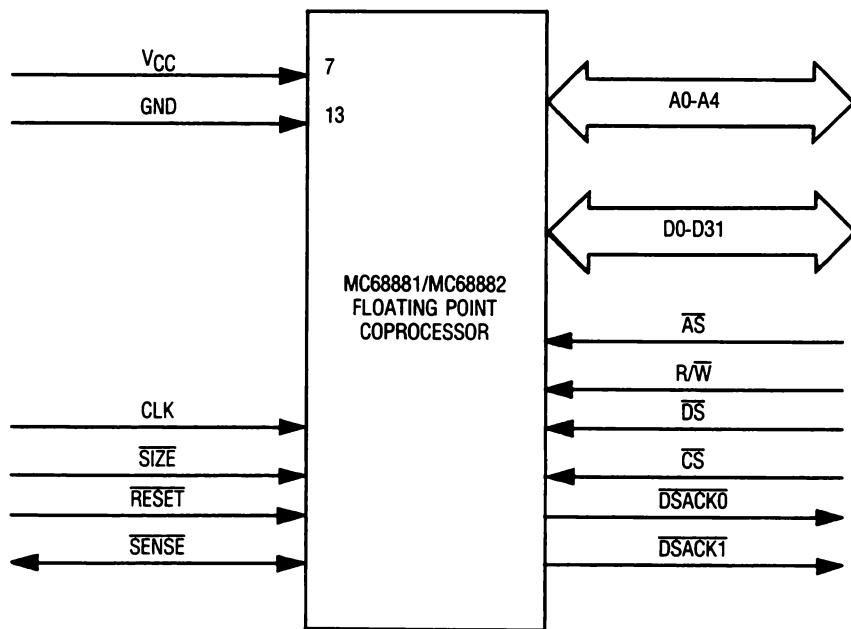


Figure 1-7. MC68881/MC68882 Input/Output Signals

ADDRESS BUS (A0 through A4)

These active-high address line inputs are used by the main processor to select the coprocessor interface register locations located in the CPU address space. These lines control the register selection as listed in Figure 1-8.

When the FPCP operates with an 8-bit data bus, the A0 pin is used as an address signal for byte accesses of the coprocessor interface registers. When the FPCP operates with a 16- or 32-bit system data bus, both the A0 and SIZE pins are strapped high and/or low as listed in Figure 1-8.

DATA BUS (D0 through D31)

This 32-bit, bidirectional, three-state bus serves as the general purpose data path between the MC68020/MC68030 (MPU) and the FPCP. Regardless of whether the FPCP is operating as a coprocessor or a peripheral processor, all interprocessor transfers of instruction information, operand data, status information, and requests for service occur as standard M68000 bus cycles.

The FPCP can operate with an 8-, 16-, or 32-bit system data bus. To operate with the required system data bus size, both the A0 and SIZE pins must be connected specifically for that applicable bus size.

SIZE (SIZE)

This active-low input signal is used in conjunction with the A0 pin to configure the FPCP for operation over an 8-, 16-, or 32-bit system data bus. When the FPCP is configured to operate over a 16- or 32-bit system data bus, the SIZE and A0 pins must be strapped as listed in Figure 1-9.

ADDRESS STROBE (AS)

This active-low input signal indicates that there is a valid address on the address bus, and both the chip select (CS) and read/write (R/W) signal lines are valid.

2.3.2 68881 FLOATING-POINT COPROCESSOR (Continued)

CHIP SELECT (CS)

This active-low input signal enables the main processor access to the FPCP coprocessor interface registers. When operating the FPCP as a peripheral processor, the chip select decode is system dependent (i.e., like the chip select on any peripheral). The CS signal must be valid (either asserted or negated) when \overline{AS} is asserted.

READ/WRITE (R/W)

This input signal indicates the direction of a bus transaction (read/write) by the main processor. A logic high (1) indicates a read from the FPCP, and a logic low (0) indicates a write to the FPCP. The R/W signal must be valid when \overline{AS} is asserted.

DATA STROBE (DS)

This active-low input signal indicates that there is valid data on the data bus during a write bus cycle.

DATA TRANSFER AND SIZE ACKNOWLEDGE (DSACK0, DSACK1)

These active-low, three-state output signals indicate the completion of a bus cycle to the main processor. The FPCP asserts both the DSACK0 and DSACK1 signals when the MPU asserts CS.

If the bus cycle is a main processor read, the FPCP asserts DSACK0 and DSACK1 signals to indicate that the information on the data bus is valid. (Both DSACK signals can be asserted in advance of the valid data being placed on the bus.) If the bus cycle is a main processor write to the FPCP, DSACK0 and DSACK1 are used to acknowledge acceptance of the data by the FPCP.

A4-A0	Offset	Width	Type	Register
0000x	\$00	16	Read	Response
0001x	\$02	16	Write	Control
0010x	\$04	16	Read	Save
0011x	\$06	16	Read Write	Restore
0100x	\$08	16	—	(Reserved)
0101x	\$0A	16	Write	Command
0110x	\$0C	16	—	(Reserved)
0111x	\$0E	16	Write	Condition
100xx	\$10	32	Read Write	Operand
1010x	\$14	16	Read	Register Select
1011x	\$16	16	—	(Reserved)
110xx	\$18	32	Write	Instruction Address
111xx*	\$1C	32	Read Write	Operand Address

Figure 1-8. Coprocessor Interface Register Selection

A0	Size	Data Bus
—	Low	8-Bit
Low	High	16-Bit
High	High	32-Bit

Figure 1-9. System Data Bus Size Configuration

2.3.2 68881 FLOATING-POINT COPROCESSOR (Continued)

DATA TRANSFER AND SIZE ACKNOWLEDGE (Continued)

The FPCP also uses **DSACK0** and **DSACK1** signals to dynamically indicate to the MPU the port size (system data bus width) on a cycle-by-cycle basis. Depending upon which of the two DSACK pins is asserted for a bus cycle, the MPU assumes data has been transferred to/from an 8-, 16-, or 32-bit wide data port. Figure 1-10 lists the DSACK assertions that are used by the FPCP for the various bus cycles over the various system data bus configurations.

Figure 1-10 indicates that all accesses using 32-bit bus with A4 equal to zero are to 16-bit registers. The FPCP implements all 16-bit coprocessor interface registers on data lines D16-D31 (to eliminate the need for on-chip multiplexors); however, the MPU expects 16-bit registers that are located in a 32-bit port at odd word addresses (A1=1) to be implemented on data lines D0-D15. For accesses to these registers when configured for 32-bit bus operation, the MC68881/M68882 generates DSACK signals as listed in Figure 1-10 to indicate to the MPU that the valid data is on D16-D31 instead of on D0-D15.

External holding registers are required to maintain both **DSACK0** and **DSACK1** high between bus cycles. In order to reduce the signal rise time, the **DSACK0** and **DSACK1** lines are actively pulled up (negated) by the FPCP following the rising edge of AS or DS, and both DSACK lines are then three-stated (placed in the high-impedance state) to avoid interference with the next bus cycle.

Data Bus	A4	DSACK1	DSACK0	Comments
32-Bit	1	L	L	Valid Data on D31-D0
32-Bit	0	L	H	Valid Data on D31-D16
16-Bit	X	L	H	Valid Data on D31-D16 or D15-D0
8-Bit	X	H	L	Valid Data on D31-D24, D23-D16, D15-D8 or D7-D0
All	X	H	H	Insert Wait States in Current Bus Cycle

Figure 1-10. DSACK Assertions

RESET (RESET)

This active-low input signal causes the FPCP to initialize the floating-point data registers to non-signaling not-a-numbers (NANs) and clears the floating-point control, status, and instruction address registers.

When performing a power-up reset, external circuitry should keep the **RESET** line asserted for a minimum of four clock cycles after **V_{CC}** is within tolerance. This assures correct initialization of the FPCP when power is applied. For compatibility with all M68000 Family devices, 100 milliseconds should be used as the minimum.

When performing a reset of the FPCP after **V_{CC}** has been within tolerance for more than the initial power-up time, the **RESET** line must have an asserted pulse width greater than two clock cycles. For compatibility with all M68000 Family devices, 10 clock cycles should be used as the minimum.

CLOCK (CLK)

The FPCP clock input is a TTL-compatible signal that is internally buffered for development of the internal clock signals. The clock input must be a constant frequency square wave with no stretching or shaping techniques required. The clock shoud not be gated off at any time and must conform to minimum and maximum period and pulse width times.

SENSE DEVICE (SENSE)

This output pin may be used optionally as an additional GND pin or as an indicator to external hardware that the FPCP is present in the system. This signal is internally connected to the GND of the die, but it is not necessary to connect it to the external ground for correct device operation.

2.3.2 68881 FLOATING-POINT COPROCESSOR (Continued)

POWER (V_{CC} and GND)

These pins provide the supply voltage and system reference level for the internal circuitry of the FPCP. Care should be taken to reduce the noise level on these pins with appropriate capacitive decoupling.

The FPCP is fabricated in Motorola's advanced HCMOS process and is capable of operating at clock speeds of 25 MHz. Although the use of CMOS for a device containing such a large number of transistors allows significantly reduced power consumption in comparison to an equivalent NMOS device, the high clock speed makes the characteristics of the power supplied to the part quite important. The power supply must be as free from noise as possible, and it must be able to supply large amounts of instantaneous current when the FPCP performs certain operations. In order to meet these requirements, more detailed attention should be given to the power supply connection to the FPCP than is required for older NMOS devices that operate at slower clock rates.

Devices Supplied	V_{CC}	GND
D31-D16	H8	J8
D15-D00	B8	B7
Internal Logic, DSACK1, DSACK0	E2, E9	A2, B2, B3, B4*, C3, E10, K3
Separate	—	C1
Extra	A1, B1, J2	A10, D2, F2, H9

Figure 1-11. V_{CC} and GND Pin Assignments

In order to provide a solid power supply interface, four V_{CC} pins, eight primary GND pins, and two secondary GND pins are provided. This allows two V_{CC} and GND pins to supply the power for the data bus, while the remaining V_{CC} and GND pins are used by the internal logic and DSACK drivers. The two secondary GND pins are not intended to provide the main power supply interface, but merely to augment it as required. (One of these pins is the SENSE pin which may be used as an optional GND connection.)

Three V_{CC} and four GND pin positions are reserved for future use by Motorola and should be connected appropriately in order to maintain pin compatibility with all future versions of the FPCP. Figure 1-11 lists the V_{CC} and GND pin assignments.

In order to reduce the amount of noise in the power supplied to the FPCP, common capacitive decoupling techniques should be observed. While there is no recommended layout for this capacitive decoupling, it is suggested that a combination of low, middle, and high frequency filter capacitors be placed as close to the chip as possible. (For example, a set of 10 μ F, 0.1 μ F, and 330 pF capacitors in parallel provides filtering for nearly the entire frequency spectrum present in a digital system.) In a system that utilizes the MC68020 as the main processor, these capacitive decoupling practices should also be observed for the main processor. In particular, the 10 μ F "tank" capacitor should be reasonably close to both devices (since the two devices are typically placed next to each other on a board) to provide for the high instantaneous current requirements of both the MPU and the FPCP.

In addition to the capacitive decoupling of the power supply, care should be taken to ensure a low resistance connection between the FPCP V_{CC} and GND pins and the system power supply traces. In particular, the connections to pins B7 and J8 (the GND pins for the data bus pins) must have very low resistance. This is necessary because a read of the FPCP can cause the data bus drivers to sink very large amounts of current to ground in order to pull the data bus signals low (if the data pattern that is read contains mostly zeroes). If low resistance connections are not provided on pins B7 and J8, the ground potential internal to the package may rise, the fall time of the data signals may be increased, and the low output voltage noise margin may be reduced.

NO CONNECT (NC)

One pin of the FPCP package is designated as a no connect (NC). This pin position is reserved for future use by Motorola and should neither be used for signal routing nor connected to V_{CC} or GND.

2.3.2 68881 FLOATING-POINT COPROCESSOR (Continued)**SIGNAL SUMMARY**

Figure 1-12 provides a summary of all the FPCP signals described in this section.

Signal Name	Mnemonic	Input/Output	Active State	Three-State
Address Bus	A0-A4	Input	High	—
Data Bus	D0-D31	Input Output	High	Yes
Size	SIZE	Input	Low	—
Address Strobe	AS	Input	Low	—
Chip Select	CS	Input	Low	—
Read/Write	R/W	Input	High/Low	—
Data Strobe	DS	Input	Low	—
Data Transfer and Size Acknowledge	DSACK0, DSACK1	Output	Low	Yes
Reset	RESET	Input	Low	—
Clock	CLK	Input	—	—
Sense Device	SENSE	Input Output	Low	No
Power Input	VCC	Input	—	—
Ground	GND	Input	—	—

Figure 1-12. Signal Summary

INFORMATION ON THE 68030, 68881 WAS REPRINTED WITH PERMISSION OF MOTOROLA INC.

FOR MORE INFORMATION ON MOTOROLA TECHNICAL LITERATURE CONTACT:

MOTOROLA SEMICONDUCTOR PRODUCTS
LITERATURE DISTRIBUTION CENTER
P.O. BOX 20924
PHOENIX, ARIZONA, U.S.A. 85036

2.4 GARY — GATE ARRAY

DESCRIPTION

Fat Gary is a custom gate array IC used in the A3000. It is packaged in an 84 pin chip, whose pinout is shown below. Gary provides many different control functions for the system. These functions are:

- Address decoding and timing for ROM.
- Address decoding and timing for Chip RAM.
- Address decoding and timing for chip registers.
- Address decoding and timing for 8520's (CIA's).
- Address decoding and timing for Real Time Clock (RTC).
- Address decoding for the Floating Point Unit (FPU).
- Address decoding for the SCSI/DMA controller IC.
- Address decoding for the local bus card slot of the A3000.
- Monitoring of the bus for timeout conditions.
- Generation of ECLK clock signal.
- Generation of 32 bit and 16 bit data strobe signals.
- Decoding for generation of auto vector (*AVEC) signal to 68030.
- Selection of the Agnus clock source.
- System RESET logic.
- System INTERRUPT control.

ROM

The onboard ROMs are selected in the address range from \$00F80000 to \$00FFFFFF. The ROM's are also selected in the range from \$00000000 to \$0007FFFF when the overlay input signal (OVL) is true (this allows the RESET instructions to be contained in the ROMs). The ROM shows up in data and program space for both the user and supervisor. ROM cacheing is enabled.

ROM TIMING

The ROM timing circuitry provides for 4 different speed settings, settable by 2 jumpers on the motherboard. The *STERM signal is used to terminate the CPU bus cycle during a ROM access. (Refer to 2.5.1)

JP1	JP0	CPU cycles
0	0	5
0	1	6
1	0	7
1	1	8

CHIP RAM

Chip RAM is selected in the address range from \$00000000 to \$001FFFFF. When the OVL input is true (high), chip RAM is not selected for addresses in the range from \$00000000 to \$0007FFFF (ROM appears here while OVL is true). Chip RAM shows up in data and program space for both the user and supervisor. Chip RAM cacheing is disabled. The CPU cycle is terminated using both DSACKs, indicating that the chip ram data is 32 bits wide.

CHIP REGISTERS

The chip registers are selected in the range from \$00DFC000 to \$00DFFFFF. They are also decoded at (shadowed) \$00C00000 to \$00CFFFFF. Earlier Amiga systems configured RAM at these locations by detecting the absence of chip registers. By mapping chip registers into this address range on the A3000, O/S software will not attempt to configure this as RAM space. Chip registers show up in user and supervisor data space. Chip register cacheing is disabled. The CPU cycle is terminated using DSACK1, indicating that the chip registers are 16 bits wide.

2.4 GARY (Continued)

8520's

There are 2 8520 CIA IC's on the motherboard, referred to here as CIA0 and CIA1. CIA0 is selected in the address range from \$00BFE000 to \$00BFEFFF. CIA1 is selected from 00BFD000 to 00BFDFFF. The CIA's show up in user and supervisor data space. Cacheing is disabled. The CPU cycle is terminated by DSACK1, indicating that the CIA is 16 bits wide. In fact, the CIAs are only 8 bits wide, but respond as 16 bits for compatibility. In order to read the 8 bits in on the 68030's D0-D7, CIA0 must be read at an odd word address, and CIA1 must be read at an even word address.

REAL TIME CLOCK (RTC)

The RTC is selected in the range from \$00DC0000 to \$00DCFFFF. It appears in user and supervisor data space. Cacheing is disabled. The CPU cycle is terminated by DSACK1, indicating that the RTC is 16 bits wide. It is actually only 4 bits, but responds as 16 bits for compatibility. In order to read the 4 bits in on the 68030's D0-D3, the RTC must be read at an odd word address.

FLOATING POINT UNIT (FPU)

The FPU performs its own bus cycle termination.

SCSI/DMA CONTROLLER (SUPER DMAC)

The SCSI/DMA controller is selected from \$00DD0000 to \$00DD3FFF. It appears in user and supervisor data space. Cacheing is disabled. In order to get this signal out to the chip as quickly as possible, it is NOT qualified with address strobe (*AS). The SCSI/DMA controller performs its own bus cycle termination.

LOCAL BUS CARD SLOT

This signal is generated in the address range from \$08000000 to \$0FFFFFF. The signal is NOT qualified with *AS. The selection shows up in all address spaces except for CPU space. No bus cycle termination is performed. This signal goes to the 200 pin local bus connector.

BUS TIMEOUT

After the assertion of *AS, a counter in Gary starts running, and is reset by the de-assertion of *AS. If the counter counts down before *AS is de-asserted, Gary terminates the cycle automatically. There are 2 different timer values available, each of which terminates the cycle differently.

There is an 8 bit register in Gary at \$00DE0000 of the user and supervisor data space. When written to, bit 7 selects the timeout mode to be used. Writing a 0 to this bit enables DSACK timeout, and a 1 enables BERR timeout (after a RESET, DSACK timeout is enabled). DSACK timeout counts for 32 C1 pulses (approximately 9 usecs), and then asserts both DSACKs to terminate the cycle. BERR timeout takes much longer, counting for approximately 250 msecs before asserting the *BERR signal. Whenever a bus timeout occurs in either mode, bit 0 of the register at \$00DE0000 is set, and is not reset until the register is read.

The purpose of the timeout is to keep the system from getting hung up if an address is asserted that selects nothing. Using BERR mode allows the system to be informed if this occurs.

The DSACK bus timeout mode was made much shorter, and terminates the bus much more 'discreetly' using DSACKs. This mode was made the default because of compatibility issues. The 1.3 ROMs purposely snoop through a large range of address spaces during boot up, which most of the time are not there. Taking 250 msecs for each one causes it to take forever to boot up. Terminating each bus cycle with *BERR also makes the software get confused. The idea is to get everything running, and then change over to BERR mode.

Since the blitter has the capability of keeping the CPU off of the chip bus for a long time, bus timeout detection is disabled whenever chip RAM or chip registers are being selected.

Automatic bus timeout can be disabled altogether by writing a 1 to bit 0 of the 8 bit register at \$00DE0001. After a RESET, this bit is automatically set to 0 (timeout enabled).

2.4 GARY (Continued)

ECLK CLOCK

The ECLK signal is generated in Gary. It is a free running clock whose frequency is 1/10 of the 7M clock. Normally ECLK is low for 6 7M clocks, and high for 4 7M clocks. However, when the CIAs are accessed the ECLK high time may be shorter than 4 7M clocks. During writes to the CIAs ECLK is high for only 2 7M clocks. During reads ECLK stays high for a minimum of 2 7M clocks, and a maximum of 4 7M clocks. ECLK can vary, but the frequency of ECLK does not change. If the ECLK high time is shortened during CIA access, the difference is made up by increasing the subsequent ECLK low time. Consequently, it is always 10 7M clocks from one rising edge of ECLK to the next.

The ECLK signal is derived from the onboard 28 megahertz oscillator. It is not derived from the XCLK signal when *XCLKE is true. Therefore, ECLK will not change frequency, even when an external system clock source is used (such as when genlocking the video).

DATA STROBES

Gary generates 6 different data strobe signals. 4 relate to 32-bit ports (*UUDS, *UMDS, *LMDS, *LLDS), and the other 2 to 16-bit ports (*UDS, *LDS). Gary decodes SIZ1, SIZ0, A1, A0 and RW signals to determine which of the strobes should be active. Whenever RW indicates the CPU is doing a read, all of the strobes are active. Individual strobes are generated during write cycles.

AUTOVECTOR

When Gary senses that the CPU is doing an exception acknowledge cycle, it generates the AUTOVECTOR signal to tell the CPU to get the exception vectors from RAM.

AGNUS CLOCK SOURCE

For genlocking purposes, an external clock can be fed to Agnus instead of the internal 28 megahertz clock. The external clock (XCLK) and the internal 28 meg clock (28M) are both inputs to Gary. The input signal called *XCLKE controls which signal is fed to Agnus (via the output called AGCLK). When *XCLKE is low, *XCLKE is output to Agnus.

SYSTEM RESET LOGIC

Gary controls reset from 2 different sources. The first is during powerup. The input line called *PWRUP should be kept low via external circuitry until VCC has become stable. The rising edge of *POWERUP causes bit 0 of an 8 bit register at 00DE0002 to be set. The *RESET output is held low for approximately 250 msecs after *PWRUP goes high. This bit can be checked by software to determine what type of reset occurred. The bit can only be reset by writing a 0 back to it (you could just as easily set this bit in order to make the next reset look like a powerup).

The second source of a reset comes from the keyboard. If the input called *KBCLK is held low for at least 60 msecs, the *RESET output will then go low, and will remain low for approximately 250 msecs after *KBCLK goes high again.

SYSTEM INTERRUPT CONTROL

Individual system interrupts are controlled by writing to the INTENA register in PAULA. Since this register resides on the chip bus, the CPU is subject to synchronization delays when attempting to access it. Therefore, an alternate method for shutting off ALL of the interrupts in the system is provided in GARY (GARY provides a bit that can be written to — the actual control is done externally in a PAL). The INTENA register in PAULA is located at the offset of \$09A. The chip registers occupy only 4k of address space. Consequently, they are 'shadowed' in 4 4k chunks from \$DFC000 to \$DFF000. The 16 bit register in GARY to control interrupts is also located at the offset of \$09A, but is selected in the range from \$DF8000 to \$DFB000 (this register is shadowed at each of the 4k chunks as well). Writing a 0 to bit 15 of this register will disable ALL of the interrupts going to the CPU. They are re-enabled by writing a 1 to the bit again (the bit is set after a RESET).

2.5.1 ROM — READ ONLY MEMORY

The A3000 has 512k of ROM configured as 128k x 32. Rom timing is adjustable via J152 and J151. The proper timing for best performance is set when the machine is manufactured and should not be changed unless Roms of different speed are used.

J152	J151	Clocks
0	0	5
0	1	6
1	0	7
1	1	8

The minimum output enable (Toe) and access (Tacc) times for the ROMs is determined by the following:

$$\begin{aligned} \text{Toe} &= (\# \text{cpu cycles}) - 2 * \text{Tcyc} - \text{Tgary} \\ \text{Tacc} &= (\# \text{cpu cycles}) - 1 * \text{Tcyc} \end{aligned}$$

where

$$\begin{aligned} \text{Tcyc} &= \text{period of a single 68030 clock} \\ \text{Tgary} &= \text{max delay thru GARY (30 nsecs)} \end{aligned}$$

V2.0 NEW FEATURES

Outlined here, are some of the new features and changes in the 2.0 version Amiga Operating System. Additional routines are provided on the system diskettes included with the A3000.

Console

- Simple refresh character map console.
- Cut & Paste from/to console.

DOS

- Recode of DOS.LIBRARY in C.
- All important BCPL routines have C interfaces.
- Many new DOS functions (pattern patching, read args, etc.).
- Cut and Paste in CON: windows.
- Auto-Open CON: handler.
- Optional scrolling history in CON window.

Exec

- Optimized autoconfig strategy.
- Improved powerup strategy to manage all available memory, coprocessors and system options.
- Supervisor Stack and ExecBase can be in fast memory.
- More robust method of replacing system modules.
- Memory and ROMs are now fully tested before use.
- Exec tests for the 68030 processor and math co-processor, and enables instruction burst. Support for using the data cache and data burst is also included.
- Preparations are under way for Virtual Memory support by Exec (post 2.0).
- Performance improvements in memory allocation, signal semaphores, interrupt dispatching, and general tuning.

FileSystem

- FastFileSystem in ROM that supports both DOS/0 and DOS/1 disks (DOS/1 = FastFileSystem).
- Notification.
- Record Locking.
- Links.

Graphics

- Support of new resolution modes.
- Support of new monitor scan rates.
- Built-in A2024 support.
- BitMapScale and graphics/scale.h,i.
- Text speed enhancements.
- ColorFont support.
- TextExtent and TextFit.
- XAttr text attribute to specify X size.

2.5.1 ROM (Continued)

Intuition

- New Screens Support: — New modes; Scrolling/Auto Scrolling; Overscan support; Public screens.
- New Look windows.
- New Gadgets — Custom gadgets.
New string gadgets: — Custom editing; Font selections; Color selections; “Activated” color selection; “Replace/fixedfield” modes.
- New State Machine. — No more VERIFY deadlocks; Can call Intuition directly (pass string of InputEvents).
- Independent programmable mouse and key repeat backlog limits.
- StealI() function for taking temporary control from Intuition.
- EZRequester() enhancement to Auto Request.

Keymap

- The usa1 map in ROM (a.k.a. usa)
- Separate keymap.library
- MapANSI() to complement MapRawKey().

Layers

- Improved Damage Control.
- Application backfill for Layer operations.

Timer

- Now has some new functions, requested by popular demand.
- Wait — similar in nature to the current timer wait functions.

Trackdisk

- Improved error handling for marginal disks/drives.
- No-click option on drives that support this.
- Slightly faster.
- Variable number of buffers.
- Ability to get buffer memory back when drive is not in use.
- Does not need as much chip ram (can use fast ram for many operations).

Workbench

- Menu redesign into more logical arrangement.
- Backdrop icons.
- Asynchronous operations like refresh, program load, copy, rename, info, etc.
- Can execute any CLI command.
- More intelligent cleanup routine.
- Programmer callable routines: — AddAppIcon(); AddAppMenuItem(); AddAppWindow(); RemoveAppIcon(); RemoveAppMenuItem(); RemoveAppWindow()
- Faster drawer opening.
- User customizable: — font; colors; backdrop pattern; default icons; user menu.
- Easier to use: — SelectAll; New Drawer; Drag Select; Unsnapshot; Shortcut keys; Workbench startup drawer.
- Can cancel operations like drag, select and drag-select.
- New Look.

2.5.2 RTC — REAL TIME CLOCK

OVERVIEW

The Real Time Clock chip contains a calendar, time counters and alarm functions. Also integrated into the chip is a 26 x 4 bit non-volatile RAM and provisions for a battery backup.

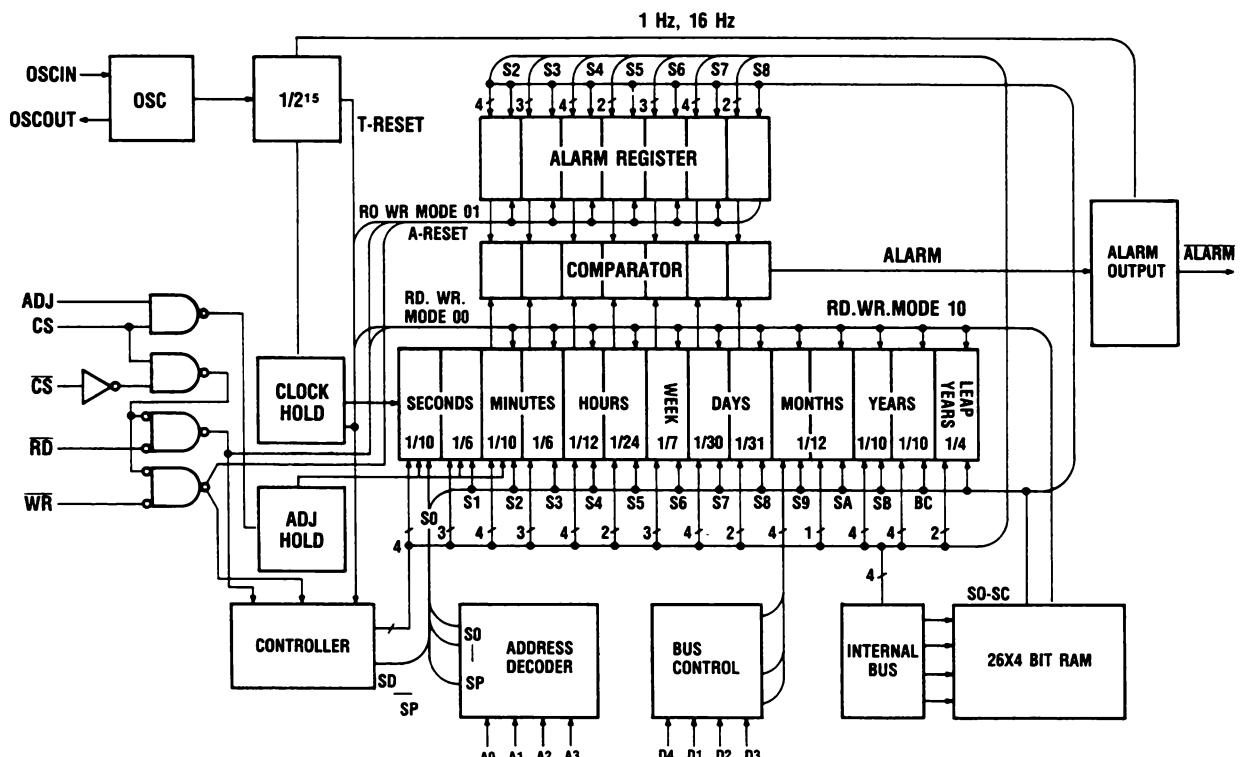


Figure 1-13. Block Diagram

SIGNAL DESCRIPTIONS

Name of Pin	No. of Pin	Function
CS, CS	1, 2	External interface terminals, valid when CS = H and \overline{CS} = L. \overline{CS} is connected to the power-down detector of the peripheral circuitry and \overline{CS} to a CPU address decoder.
ADJ	3	For easy adjustment of the second counter without connection to a CPU. If ADJ is set to high when the second counter registers 0 - 29, the seconds are set to 0, and if ADJ is set to high when the second counter registers 30 - 59, the seconds are set to 0 and the minutes are incremented. This terminal is designed not for edge detection but for level detection. A minimum of 100 μ sec. is required for high-level adjustments.
A ₀ - A ₃	4, 5, 6, 7	Address terminals. Connected to address bus of CPU.
RD	8	I/O control terminal. Low when RP5C01 is read by CPU.
GND	9	0V
WR	10	I/O control terminal. Low when RP5C01 is written by CPU.
D ₀ - D ₃	11, 12, 13, 14	Bidirectional data bus. Connected to data bus of CPU.
ALARM	15	For output of alarm signal or 16Hz/1HZ clock signals. Open-drain output.
OSCIN, OSCO	16 17	For connection to 32.768kHz crystal oscillator circuit.
Vcc	18	+5V power supply terminal

2.6 8372 AGNUS 2MEG

GENERAL

This device is an address generator type IC. Its main function is as a RAM address generator and register address encoder that shall produce all DMA addresses for 25 channels.

The block diagram (Figure 1-14) for this device shows the DMA control and address bus logic. The output of each controller indicates the number of DMA channels driving the Register Address Encoder and RAM Address Generator.

The RAM Address Generator contains an 20 bit pointer register for each of the 25 DMA channels and also it contains pointer restart (backup) registers and jump registers for six (6) of the channels. A full 20 bit adder carries out the pointer increments and adds for jumps.

The priority control logic looks at the pipe-lined DMA request from each controller and stages the DMA cycles based upon their programmed priority and sync counter time slot. Then it signals the processor to get off the bus by asserting the DBR line. The following is a brief description of the device's major operational modes.

A control register determines which 256 possible logic operations is to be performed as the source images are combined and how far they are to be moved (Barrel shifted). In addition to the image combining and movement powers, the Blitter can be programmed to do line drawing or area fill between lines.

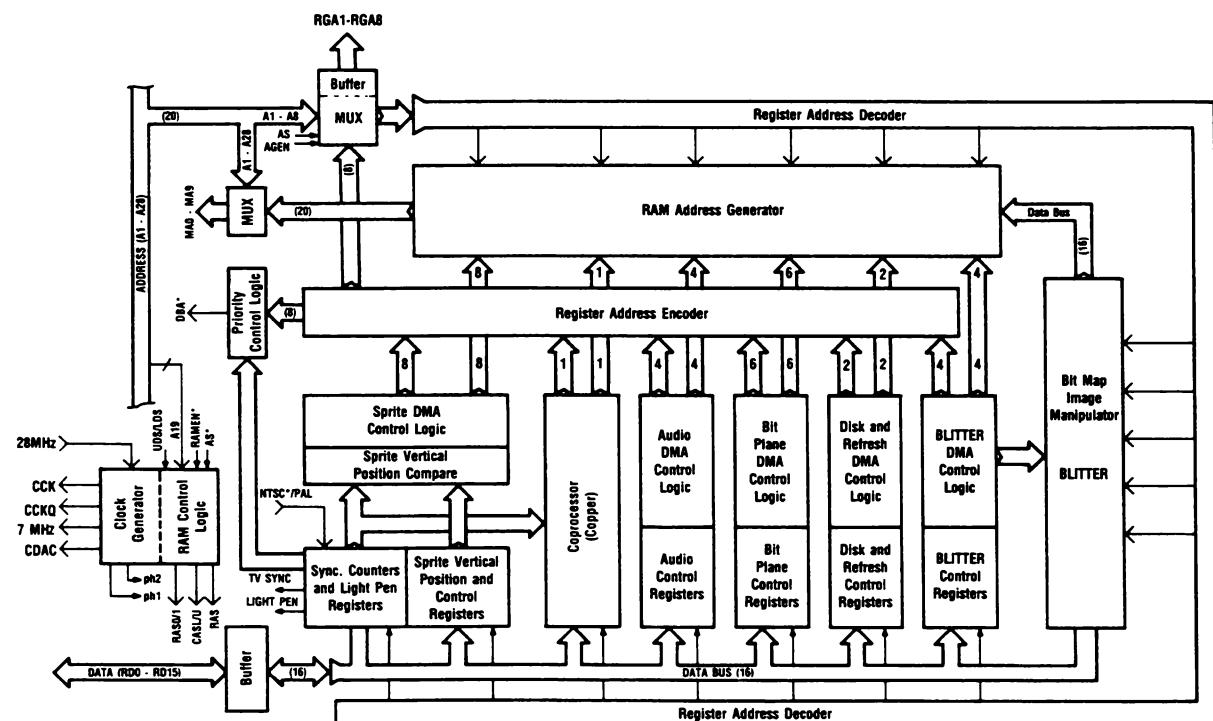


Figure 1-14. Block Diagram

BLITTER

The procedure for moving and combining bit mapped images in memory received the name Bit Blit from a computer instruction that did block transfers of data on bit boundaries. These routines became known as Bit Blitters or Blitters. The Blitter DMA Controller is preloaded with the address and size of three (3) source images (A,B, and C) and one (1) destination (D) in the dynamic RAM. These images can be as small as a single character or as large as twice the screen size. They can be full images or smaller windows of a larger image. The actual pixel resolution is controlled by the BLTSIZH (BLTSIZH and BLTSIZV) registers which contain up to 15 bits for the image height (15 bits = 32K dots max.) and up to 11 bits for the image width (11 bits = 2k words = 32K pixels max.). After one word of each source image is sequentially loaded into the source buffer (A, B, C) they are shifted and then combined together in the logic unit to perform image movement overlay, masking, and replacements. The result is captured in the destination buffer (D) and sent back to the RAM memory destination address. This operation is repeated until the complete image has been processed. The unit has extensive pipelining to allow for shifter and logic unit propagation time, while the next set of source words is being fetched.

2.6 8372 AGNUS (Continued)

BITPLANE ADDRESSING

Some computer bitmap displays are organized so that the bitplanes for each pixel are all located within the same address. This is called pixel addressing. If the entire data word of one address is used for a single pixel with 8 bit planes, the data word will look like this. (numbers are bitplanes): 12345678-----

The data compression can be improved by packing more than one pixel into a single address like this: 1234567812345678 or like this, if there are only 4 bitplanes: 1234123412341234

The IC device, uses a bitmap technique called Bitplane Addressing. This separates the bitplanes in memory. To create a 4 plane (16 color) image, the bitplane display DMA channels fetch from 4 separate areas of memory like this:

```
1111111111111111  
2222222222222222  
3333333333333333  
4444444444444444
```

These are held in buffer register and are used together as pixels, one bit at a time, by the display (left to right). This technique allows reduced odd numbers of bitplanes (such as 3 or 5) while maintaining packing efficiency and speed. It also allows grouping bitplanes into 2 separate images, each with independent hardware high speed image manipulation, line draw, and area fill.

DMA CHANNEL FUNCTIONS

Each channel has an 20 bit RAM address pointer that is placed on the MA memory address bus, and is used to select the location of the DMA data transfer from anywhere in 1M words (2M bytes) of RAM.

An eight (8) bit destination address is simultaneously placed on the register address bus (RGA), sending the data to the corresponding register.

In a typical DMA channel, almost all channels have DRAM as source and chip registers as destination.

The pointer must be preloaded and is automatically incremented each time a data transfer occurs.

Each controller utilizes one or more of these DMA channels for its own purposes. The following is a brief summary of these controllers and the DMA channels they use.

A-Blitter (four (4) channels)

The Blitter uses four (4) DMA channels, Three source and one (1) destination as previously described.

Once the Blitter has been started, the four (4) DMA channels are synchronized and pipelined to automatically handle the data transfers without further processor intervention. The images manipulated in memory, independent of the display (bitplane DMA).

B-Bitplane (six (6) channels)

The bitplane controller continuously (during display) transfers display data from memory to display buffer registers. There are six (6) DMA channels to handle the data from six (6) independent bit planes. The buffers convert this bitplane data into pixel data for the display.

C-Copper (one (1) channel)

The Copper is a co-processor that uses one of the DMA channels to fetch its instructions. The DMA pointer is the instruction counter and must be preloaded with the starting address of the Copper's instructions.

The Copper can move (write) data into chip registers. It can skip, jump, and wait (halt). These simple instructions give great power and flexibility because of the following features.

When Copper is halted, it is off the data bus, using no bus cycles until the wait is over. The programmed wait value is compared to a counter that keeps track of the TV beam position (beam counter) and when they are equal, the Copper will resume fetching instructions.

It can cause interrupts, reload the color registers, start the Blitter or service the audio. It can modify almost any register inside or outside the IC device, based on the TV screen coordinates given by the Beam Counter and the actual address encoded on the RGA Bus.

2.6 8372 AGNUS (Continued)

DMA CHANNEL FUNCTIONS (Continued)

D-*Audio (four (4) channels)*

There are four (4) audio channels, all of which are located outside of the audio DMA Controller section of Agnus. Each controller is independent and uses one DMA channel from the DMA Controller and fetches its data during a dedicated timing slot within horizontal blanking. This is accomplished by a controller asserting the DMAL input on the DMA Controller.

E-*Sprites (eight (8) channels)*

There are eight (8) independent Sprite controllers, each with its own DMA channel and its own dedicated time slot for DMA data transfer. Sprites are line buffered objects that can move very fast because their positions are controlled hardware registers and comparators.

Each sprite has two (2) sixteen bit data registers that define a 16 pixel wide Sprite with 4 colors. Each has a horizontal position register, a vertical start position register and a vertical stop position register. This allows variable vertical size sprites.

The Sprite DMA controller fetches image and position data automatically from anywhere in 2 Megabytes of memory depending on device pin configuration.

Sprites can be run automatically in DMA mode or they can be loaded and controlled by the microprocessor.

Each Sprite can be re-used vertically as often as desired. Horizontal re-using is also allowed with microprocessor control.

F-*Disk (one (1) channel)*

The disk controller, which is located outside of the DMA, uses a single DMA channel from the device. The controller uses the DMA time slot for data transfer and can read or write a block of data up to 128K anywhere in 2 Megabytes of memory depending on device pin configuration.

G-*Memory Refresh (one (1) channel)*

The refresh controller uses a single DMA channel with its own time slots. It places RAS addresses on the memory address bus (MA) during these slots, in order to refresh the dynamic RAM. Memory is refreshed on every raster line.

During the DMA no data transfer actually takes place. The register address bus (RGA) is used to supply video synchronizing codes. At this time RAS1* and RAS are low. CASU* and CASL* are inactive during this cycle.

RAM AND REGISTER ADDRESSING

The device generates RAM addresses from two sources, the processor or the device performing DMA cycles. The processor accesses RAM whenever AS* and RAMEN* are both low. At this time, the device also multiplexes the processor address (A1-A20) onto the MA bus. During row address time A9-A17 and A19 are placed onto MA0-MA8, MA9, respectively; during column address time A1-A8, A18 and A20 are placed onto MA0-MA7, MA8 and MA9, respectively. In the 1 meg configuration, A19 is still used to determine the RAS line to be asserted. If A19 is low RAS0* is active and if high RAS1* is active. In the 2 meg option RAS will always be active on a RAM access. The IC will assert CASL* if LDS* is low or CASU* if UDS* is low.

When the device needs to do a DMA cycle, the device disables the processor from accessing RAM by asserting the Data Bus Request Line (DBR*). At this time, the device multiplexes its generated RAM address onto the MA lines and will activate RAS and the proper RAS0* or RAS1* line unless it is a refresh cycle where all RAS lines are active. During a DMA cycle, the IC device will also assert both CASU* and CASL*, unless it is a refresh cycle where they both remain inactive.

The device also generates RGA addresses from either the processor or device DMAs, each of which is selected by an internal multiplexer. This multiplexer allows the processor to perform a register read/write access when AS* and RGEN* are both low. The device then takes the low order byte of the processor address A1 to A8 and reflects its value on the RGA output bus RGA1 to RGA8. The device will reflect the status of PRW input on the RRW output line, to indicate a memory read or write operation.

During a device DMA cycle, the device prevents the processor from doing a register access by asserting the DBR* line. The device will then place the contents of its register address encoder onto the RGA bus.

2.6 8372 AGNUS (Continued)

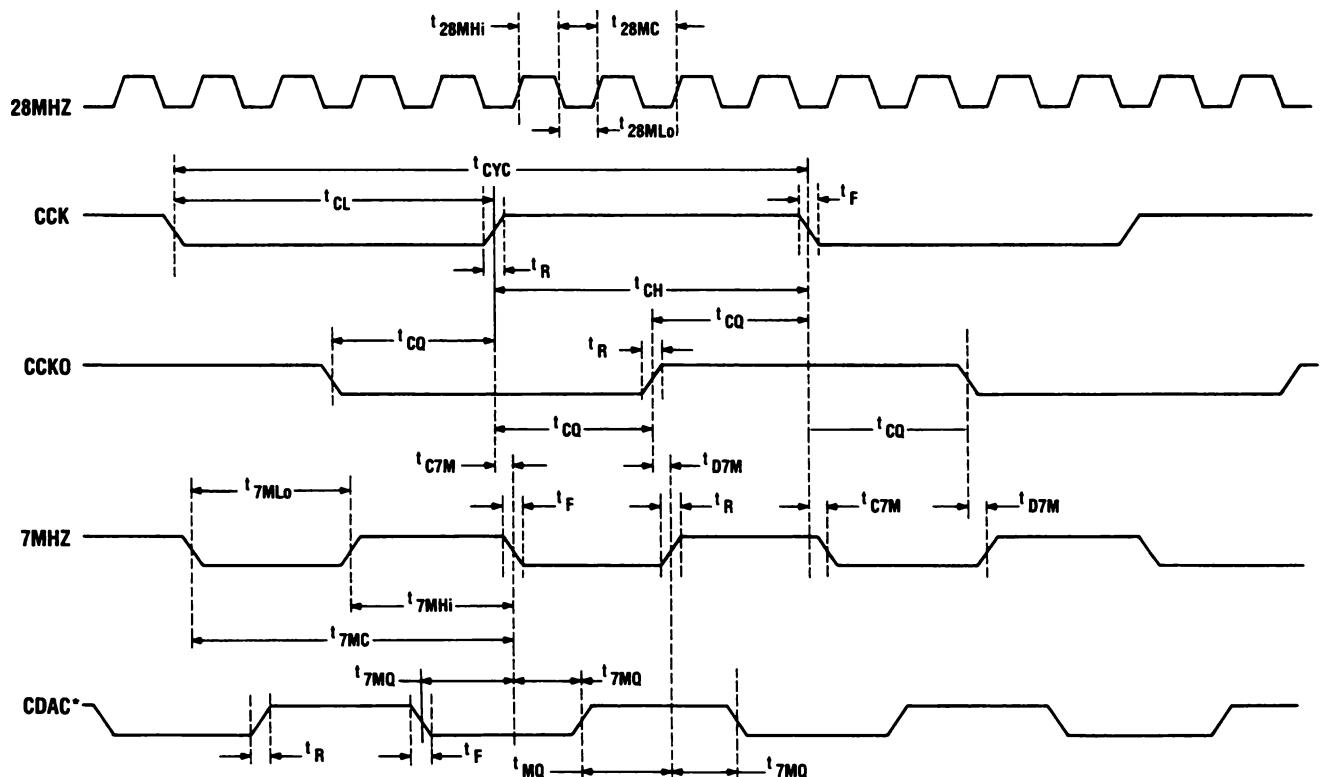


Figure 1-15. Clock Relations

CLOCK RELATIONS (Refer to Figure 1-15)

	SYMBOL	MIN	MAX	UNIT
2.4.1 28MHz clock cycle	t28MC	34.57		ns
2.4.2 28MHz clock high	t28MHi	12.0	22.9	ns
2.4.3 28MHz clock low	t28MLo	12.0	22.9	ns
2.4.4 CCK clock cycle	t _{cyc}	260	290	ns
2.4.5 CCK clock high	t _{ch}	130	150	ns
2.4.6 CCK clock low	t _{cl}	130	150	ns
2.4.7 CCK-CCKQ clock separation	t _{cq}	65	75	ns
2.4.8 7MHz clock cycle	t _{7MC}	130	150	ns
2.4.9 7MHz clock high	t _{7MHi}	65	75	ns
2.4.10 7MHz clock low	t _{7MLo}	65	75	ns
2.4.11 7MHz-CDACQ clock separation	t _{7MQ}	30	40	ns
2.4.12 CCK to 7MHz delay	t _{c7M}	0	15	ns
2.4.13 CCKQ to 7MHz delay	t _{q7M}	0	15	ns
2.4.14 Clock rise time	t _r	0	10	ns
2.4.15 Clock fall time	t _f	0	10	ns

2.7 CHIP MEMORY

CHIP memory contains graphics, sound and other data that is accessible by the Amiga custom chips. Certain software requiring large amounts of this type of data, or many programs executing concurrently that require this type of data, can quickly exhaust the one megabyte of CHIP memory that comes in the A3000 as shipped. Another megabyte of CHIP memory can be added. When adding CHIP memory, all eight of the empty sockets (U267-U274) must be populated with the new RAM chips. In A3000 (unlike A2000) Agnus will function with static column drams.

There are no jumpers to set when adding CHIP memory. If the new RAM is functioning properly (a memory test will be performed at power up), it will automatically be recognized by the operating system. The title bar of the Workbench screen (which shows how much chip memory is available to the system) should reflect the addition of the new memory. If this number is not approximately 1 megabyte larger than it was when the machine was operated without the new RAM, then the RAM is not working properly. Also, if the video screen turns green when the A3000 is turned on, then none of the CHIP memory is functioning properly. In either case, check that the RAM chips were inserted properly, and check for any bent pins.

The A3000 can access up to 2 megabytes of chip RAM. Although all current 8372 FAT AGNUS chips used in the A500 and A2000 are designed internally to support 2 megabytes of CHIP memory, there are separate 1 meg and 2 meg versions. These two versions differ in pinout only. The 2 meg version is designed to be used with 1 meg x 1 DRAMs, and therefore requires an additional multiplexed address bit out (DRA9) and a single *RAS out. Also, in order to access the second meg of memory, another address bit from the CPU (A20) must be fed into the chip. The following table shows which pins have changed.

pin	1 meg	2 meg
35	SCLKIN	A20
56	*RAS1	DRA9
57	*RAS0	*RAS

Since the external clock input pin (XCLKIN) was sacrificed, the clock switch mechanism must be done externally (it is done in FAT GARY).

Eight 256k x 4 DRAMs yield 1 megabyte of memory that is 32 bits wide to the 68030, and 16 bits wide to the custom chips. An additional bank of 8 chips can then be added to double the amount of CHIP RAM.

Since AGNUS only supplies two *CAS signals (*CASH,*CASL), external logic is needed to generate the 8 separate CAS'es that are needed. In order to do this, however, we need to know which bank is being asked for. It is easy to tell what the CPU is asking for since A19 and A20 are available, but the addresses that the custom chips want is internal to the AGNUS chip itself. The proper bank must be determined by 'extracting' A19 and A20 from the multiplexed address coming out of AGNUS. A19 is extracted by latching the value of DRA9 at *RAS time. DRA9 contains the value of A20 at *CAS time.

Figure 1.16 shows the three possible data paths for CHIP RAM data.

The following is a list of required parameters for the new CHIP RAM:

Organization:	256 kilobits by 4 bits (256K x 4)
Speed:	120 nanoseconds or less
Type:	page or static column mode
Package:	20 pin DIP

The following are examples of some of the DRAMs that are acceptable:

TOSHIBA	TC514256 TC514258
OKI	MSM514256 MSM514258
HITACHI	HM514256 HM514258
NEC	uPD424256 uPD424258
SHARP	LH64258
TI	TMS44C256

2.7 CHIP MEMORY (Continued)

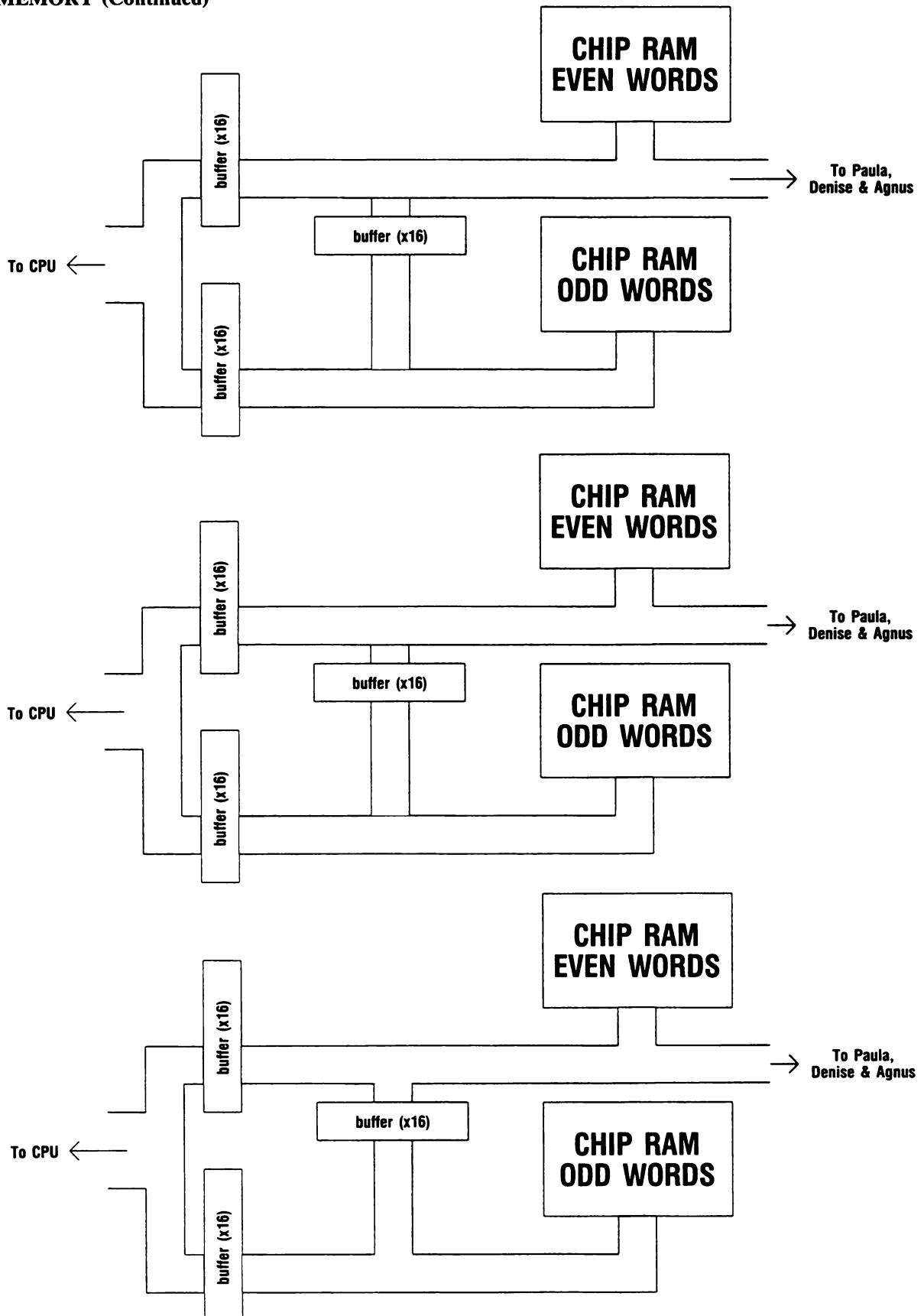


Figure 1-16. Data Paths for CHIP RAM Data

2.8 CIA0 - 8520 (U350)

OVERVIEW

The complexed Interface Adapter, 8520, at location U350 divides the 8 bits of PortA between status signals from the floppy drives, Fire buttons, Power LED, and the Overlay signal to Gary. PortB provides an output port for one byte of data to the Parallel Interface. Keyboard Data and Keyboard Clock signals are also interfaced to the system through this IC.

A functional description of the 8520 is included at the end of 2.9. Detailed information on the use of the 8520 in the Amiga can be found in the AMIGA HARDWARE REFERENCE MANUAL, by ADDISON WESLEY, available at your local Computer bookstore.

2.9 CIA1 - 8520 (U300)

OVERVIEW

The second 8520 located at U300, uses PortA for standard RS232 control signals, Transmit signals to the 1488 line driver at U304 and Receive signals from the 1489 line receiver IC at U305. Note that actual data is passed to and from the 1488/1489 directly to registers in Paula and are not processed by the 8520.

PortB is used to output control signals to the Floppy Drive(s), these include: Drive Select, Motor On, Step, Side, and Direction.

Several Parallel Port control signals are processed by this 8520.

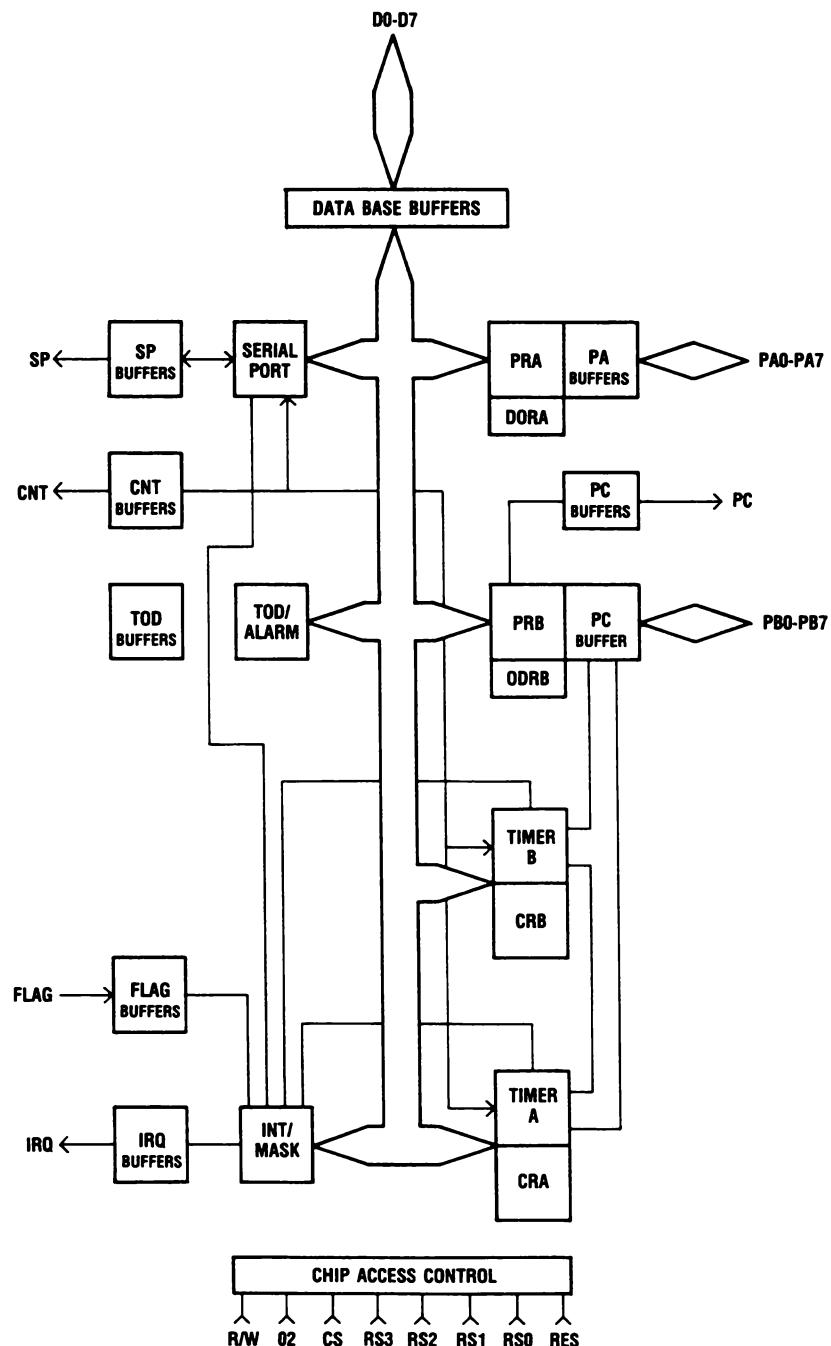


Figure 1-17. 8520 Block Diagram

2.9 8520 FUNCTIONAL DESCRIPTION (Continued)

I/O PORTS (PRA, PRB, DDRA, DDRB)

Ports A and B each consist of an 8-bit Peripheral Data Register (PR) and an 8-bit Data Direction Register (DDR). If a bit in the DDR is set to the corresponding bit in the PR it is an output. If a DDR bit is set to zero, the corresponding PR bit is defined an input. On a READ, the PR reflects the information present on the actual port pins (PA0-PA7, PB0-PB7) for both input and output bits. Port A has both passive and active pullup devices, providing both CMOS and TTL compatibility. It can drive 2 TTL loads. Port B has only passive pullup device and has a much higher current-sinking capability.

HANDSHAKING

Handshaking on data transfers can be accomplished using the PC output pin and the FLAG input pin. PC will go low on the 3rd cycle after a PORT B access. This signal can be used to indicate "data ready" at PORT B or "data accepted" from PORT B. Handshaking on a 16-bit data transfers (using both PORT A and PORT B) is possible by always reading or writing PORT A first. FLAG is a negative edge sensitive input which can be used for receiving the PC output from another 8520 or as a general purpose interrupt input. Any negative transition on FLAG will set the FLAG interrupt bit.

REG	NAME	D7	D6	D5	D4	D3	D2	D1	D0
0	PRA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
1	PPB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
2	DDRA	DPA7	DPA6	DPA5	DPA4	DPA3	DPA2	DPA1	DPA0
3	DDRB	DPB7	DPB6	DPB5	DPB4	DPB3	DPB2	DPB1	DPB0

INTERVAL TIMERS (TIMER A, TIMER B)

Each interval timer consists of a 16-bit read-only Timer Counter and a 16-bit write-only Timer Latch. Data written to the timer is latched in the Timer Latch, while data read from the timer are the present contents of the Timer Counter. The timers can be used independently or linked for extended operations. The various timer modes allow generation of long time delays, variable width pulses, pulse trains and variable frequency waveforms. Utilizing the CNT input, the timers can count external pulses or measure frequency, pulse width and delay times of external signals. Each timer has an associated control register, providing independent control of the following functions.

START/STOP

A control bit allows the timer to be started or stopped by the microprocessor at any time.

PB ON/OFF

A control bit allows the timer output to appear on a PORT B output line (PB6 for TIMER A and PB7 for TIMER B). This function overrides the DDRB control bit and forces the appropriate PB line to an output.

TOGGLE/PULSE

A control bit selects the output applied to PORT B. On every timer underflow the output can either toggle or generate a single positive pulse of one cycle duration. The toggle output is set high whenever the timer is started and is set low by RES.

ONE-SHOT/CONTINUOUS

A control bit selects either timer mode. In one-shot mode, the timer will count down from the latched value to zero, generate an interrupt, reload the latched value, then stop. In continuous mode, the timer will count from the latched value to zero, generate an interrupt, reload the latched value and repeat the procedure continuously. In one-shot mode: a write to Timer High (registers 5 for TIMER A, 7 for TIMER B) will transfer the timer latch to the counter and initiate counting regardless of the start bit.

FORCE LOAD

A strobe bit allows the timer latch to be loaded into the timer counter at any timer, whether the timer is running or not.

2.9 8520 FUNCTIONAL DESCRIPTION (Continued)

INPUT MODE

Control bits allow selection of the clock used to decrement the timer. TIMER A can count 02 pulses or external pulses applied to the CNT pin. TIMER B can count 02 pulses, external CNT pulses, TIMER A underflow pulses or TIMER A underflow pulses while the CNT pin is held high.

The timer latch is loaded into the timer on any timer underflow, on a force load or following a write to the high byte of the prescaler while the timer is stopped. If the timer is running, a write to the high byte will load the timer latch, but not reload the counter.

READ (TIMER)

REG	NAME								
4	TA LO	TAL7	TAL6	TAL5	TAL4	TAL3	TAL2	TAL1	TAL0
5	TA HI	TAH7	TAH6	TAH5	TAH4	TAH3	TAH2	TAH1	TAH0
6	TB LO	TBL7	TBL6	TBL5	TBL4	TBL3	TBL2	TBL1	TBL0
7	TB HI	TBH7	TBH6	TBH5	TBH4	TBH3	TBH2	TBH1	TBH0

WRITE (PRESCALER)

REG	NAME								
4	TA LO	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0
5	TA HI	PAH7	PAH6	PAH5	PAH4	PAH3	PAH2	PAH1	PAH0
6	TB LO	PBL7	PBL6	PBL5	PBL4	PBL3	PBL2	PBL1	PBL0
7	TB HI	PBH7	PBH6	PBH5	PBH4	PBH3	PBH2	PBH1	PBH0

TICK (TOD)

TOD consists of a 24 bit binary counter. Positive edge transitions on this pin cause the binary counter to increment. The TOD pin has a passive pull-up on it. A programmable ALARM is provided for generating an interrupt at a desired time. The ALARM registers are located at the same addresses as the corresponding TOD register. Access to the ALARM is governed by a Control Register bit. The ALARM is write-only; any read of a TOD address will read time regardless of the state of the ALARM access bit.

A specific sequence of events must be followed for proper setting and reading of TOD. TOD is automatically stopped whenever a write to the register occurs. The clock will not start again until after a write to the LSB Event Register. This assures TOD will always start at the desired time. Since a carry from one stage to the next can occur at any time with respect to a read operation, a latching function is included to keep all Time of Day information constant during a read sequence. All TOD registers latch on a read of MSB event and remain latched until after a read of LSB Event. The TOD clock continues to count when the output registers are latched. If only one register is to be read, there is no carry problem and the register can be read "on the fly", provided that any read of MSB Event is followed by a read of LSB Event to disable the latching.

READ

REG	NAME								
8	LSB Event	E7	E6	E5	E4	E3	E2	E1	E0
9	Event 8-15	E15	E14	E13	E12	E11	E10	E9	E8
A	MSB Event	E23	E22	E21	E20	E19	E18	E17	E16

WRITE

CRB7=0

CRB7=1 ALARM

(SAME FORMAT AS READ)

2.9 8520 FUNCTIONAL DESCRIPTION (Continued)

SERIAL PORT (SDR)

The serial port is a buffered, 8-bit synchronous shift register system. A control bit selects input or output mode. In input mode, data on the SP pin is shifted into the shift register on the rising edge of the signal applied to the CNT pin. After 8 CNT pulses, the data in the shift register is dumped into the Serial Data Register and an interrupt is generated. In the output mode, TIMER A is used for the baud rate generator. Data is shifted out on the SP pin at 1/2 the underflow rate of TIMER A. The maximum baud rate possible is 02 divided by 6, but the maximum useable baud rate will be determined by line loading and the speed at which the receiver responds to input data. Transmission will start following a write to the Serial Data Register (provided TIMER A is running and in continuous mode). The clock signal derived from TIMER A appears as an output on the CNT pin. The data in the Serial Data Register will be loaded into the shift register then shift out to the SP pin when a CNT pulse occurs. Data shifted out becomes valid on the falling edge of CNT and remains valid until the next falling edge. After 8 CNT pulses, an interrupt is generated to indicate more data can be sent. If the Serial Data Register was loaded with new information prior to this interrupt, the new data will automatically be loaded into the shift register and transmission will be continuous. If no further data is to be transmitted, after the 8th CNT pulse, CNT will return high and SP will remain at the level of the last data bit transmitted. SDR data is shifted out MSB first and serial input data should also appear in this format.

The bidirectional capability of the Serial Port and CNT clock allows several devices to be connected to a common serial communication bus on which one acts as a master, sourcing data and shift clock, while all other chips act as slaves. Both CNT and SP outputs are open drain, with passive pull-ups, to allow such a common bus. Protocol for slave/master selection can be transmitted over the serial bus, or via dedicated handshaking lines.

REG	NAME	C	SDR	S7	S6	S5	S4	S3	S2	S1	S0
-----	------	---	-----	----	----	----	----	----	----	----	----

INTERRUPT CONTROL (ICR)

There are five sources of interrupts on the 8520: underflow from TIMER A, underflow from TIMER B, TOD ALARM, Serial Port full/empty and FLAG. A single register provides masking and interrupt information. The Interrupt Control Register consists of a write-only MASK register and a read-only DATA register. Any interrupt which is enabled by the MASK register will set the IR bit (MSB) of the DATA register and bring the IRQ pin low. In a multi-chip system, the IR bit can be polled to detect which chip has generated an interrupt request.

The interrupt DATA register is cleared and the IRQ line returns high following a read of the DATA register. Since each interrupt sets an interrupt bit regardless of the MASK, and each interrupt bit can be selectively masked to prevent the generation of a processor interrupt, it is possible to intermix polled interrupts with true interrupts. However, polling the IR bit will cause the DATA register to clear, therefore, it is up to the user to preserve the information contained in the DATA register if any polled interrupts were present.

The MASK register provides convenient control of individual mask bits. When writing to the MASK register, if bit 7 (SET/CLEAR) of the data written is a ZERO, any mask bit written with a one will be cleared, while those mask bits written with a zero will be unaffected. If bit 7 of the data written is a ONE, any mask bit written with a one will be set, while those mask bits written with a zero will be unaffected. In order for an interrupt flag to set IR and generate an Interrupt Request, corresponding MASK bit must be set.

READ (INT DATA)

REG	NAME	D	IRC	IR	0	0	FLG	SP	ALRM	TB	TA
-----	------	---	-----	----	---	---	-----	----	------	----	----

WRITE (INT MASK)

REG	NAME	D	IRC	S/C	X	X	FLG	SP	ALRM	TB	TA
-----	------	---	-----	-----	---	---	-----	----	------	----	----

2.9 8520 FUNCTIONAL DESCRIPTION (Continued)

CONTROL REGISTERS

There are two control registers in the 8520: CRA and CRB. CRA is associated with TIMER A and CRB is associated with TIMER B.

CRA:

BIT	NAME	FUNCTION
0	START	1 = START TIMER A, 0 = STOP TIMER A. This bit is automatically reset when underflow occurs during one-shot mode.
1	PBON	1 = TIMER A output appears on PB6, 0 = PB6 normal operation.
2	OUTMODE	1 = TOGGLE, 0 = PULSE
3	RUNMODE	1 = ONE-SHOT, 0 = CONTINUOUS
4	LOAD	1 = FORCE LOAD (this is a STROBE input, there is no data storage, bit 4 will always read back a zero and writing a zero has no effect.)
5	INMODE	1 = TIMER A counts positive CNT transitions, 0 = TIMER A counts 02 pulses.
6	SPMODE	1 = SERIAL PORT output (CNT sources shift clock). 0 = SERIAL PORT input (external shift clock required).
7	TODIN	1 = 50 Hz clock required on TOD pin for accurate time. 0 = 60 Hz clock required on TOD pin for accurate time.

CRB:

BIT	NAME	FUNCTION
		(Bits CRB0-CRB4 are identical to CRA0-CRA4 for TIMER B with the exception that bit 1 controls the output of TIMER B on PB7).
5,6	INMODE	Bits CRB5 and CRB6 select one of four input modes for TIMER B as: CRB6 CRB5
		0 0 TIMER B counts 02 pulses. 0 1 TIMER B counts positive CNT transitions. 1 0 TIMER B counts TIMER A underflow pulses. 1 1 TIMER B counts TIMER A underflow pulses while CNT is high.
7	ALARM	1 = writing to TOD registers set ALARM, 0 = writing to TOD registers sets TOD clock.

E CLOCK INPUT

The E clock is a TTL, compatible input used for internal device operation and as a timing reference for communicating with the system data bus.

CS — CHIP SELECT INPUT

The CS input controls the activity of the 8520. A low level on CS while 02 is high causes the device to respond to signals on the R/W and address (RS) lines. A high on CS prevents these lines from controlling the 8520. The CS line is normally activated (low) at 02 by the appropriate address combination.

R/W — READ/WRITE INPUT

The R/W signal is normally supplied by the microprocessor and controls the direction of data transfers of the 8520. A high on R/W indicates a read (data transfer out of the 8520), while a low indicates a write (data transfer into the 8520).

RS3-RS0 — ADDRESS INPUTS

The address inputs select the internal registers as described by the Register Map on the next page.

2.9 8520 FUNCTIONAL DESCRIPTION (Continued)**DB7-DB0 — DATA BUS INPUTS/OUTPUTS**

The eight bit data bus transfers information between the 8520 and the system data bus. These pins are high impedance inputs unless CS is low and R/W and 02 are high, to read the device. During this read, the data bus output buffers are enabled, driving the data from the selected register onto the system data bus.

INT — INTERRUPT REQUEST OUTPUT

INT is an open drain output normally connected to the processor interrupt input. An external pullup resistor holds the signal high, allowing multiple INT-outputs to be connected together. The INT output is normally off (high impedance) and is activated low as indicated in the functional description.

RES — RESET INPUT

A low on the RES pin resets all internal registers. The port pins are set as inputs and port registers to zero (although a read of the ports will return all highs because of passive pullups). The timer control registers are set to zero and the timer latches to all ones. All other registers are reset to zero.

REGISTER MAP

RS3	RS2	RS1	RS0	REG	
0	0	0	0	0	PRA Peripheral Data Reg. A
0	0	0	1	1	PRB Peripheral Data Reg. B
0	0	1	0	2	DDRA Data Direction Reg. A
0	0	1	1	3	DDR B Data Direction Reg. B
0	1	0	0	4	TA LO Timer A Low Register
0	1	0	1	5	TA HI Timer A High Register
0	1	1	0	6	TB LO Timer B Low Register
0	1	1	1	7	TB HI Timer B High Register
1	0	0	0	8	Event LSB
1	0	0	1	9	Event 8-15
1	0	1	0	A	Event MSB
1	0	1	1	B	No Connect
1	1	0	0	C	SDR Serial Data Register
1	1	0	1	D	ICR Interrupt Control Register
1	1	1	0	E	CRA Control Register A
1	1	1	1	F	CRB Control Register B

**FOR DETAILED INFORMATION ON 8520 USE IN AMIGA —
REFER TO AMIGA HARDWARE REFERENCE MANUAL BY
ADDISON WESLEY.**

2.10 8364 PAULA

Paula is the Port, Audio and Uart chip. Its main function is the four audio channels. It also contains the I/O ports, (Disk and Pots), Serial Port (Uart), and the Interrupt Control and Status Register.

D TO A CONVERTERS

The four audio channels each have a DMA pointer register, data register, period, (frequency), register and volume register. Each channel has an on chip D to A (digital to analog) converter on the output. The four channels are grouped into a right and left audio output.

DISK CONTROL

The disk controller has registers for data read, data write and control. It also contains a Precompensation Output circuit, a Data separator input circuit with a digital phase lock loop.

UART CONTROL

The serial port uart included in Paula contains Data registers, Control registers, Transmit, (TRN), and receive registers.

POT CONTROL

The four pot ports are general purpose I/O ports. They have counters for simple A to D (digital to analog) conversion of an external capacitor charging, which could also be used for analog joystick controllers.

INTERRUPT CONTROL

The audio, disk and uart controllers all set their own Interrupt Status register bits.

DMA REQUEST LOGIC

The audio and disk controllers also go to the DMA request logic, (remember: they are DMA users), causing the DMAL signal to request DMA cycles from Agnus.

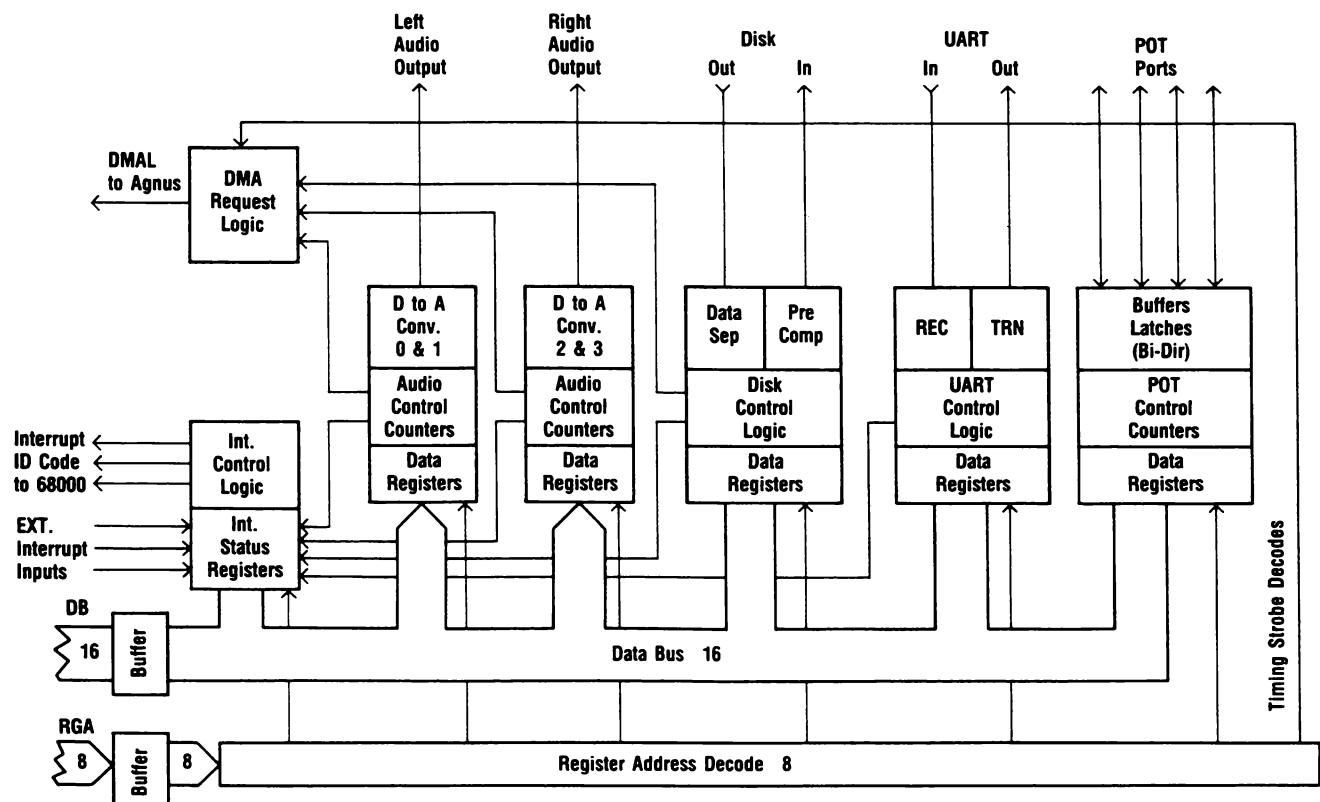


Figure 1-18. Paula Block Diagram

2.11 8373 DENISE HI RES

MAIN FUNCTIONS

- Display data buffer, encode display object to RGB colors.
 - Bitplane & Sprite display. Parallel data from data bus is retained in six (6) Bitplane and eight pairs of Sprite data buffers.
 - Bitplane Data loaded and serialized during display activity.
 - Sprite Data loaded during display inactivity — individual serialization occurs when Sprite position Compare logic detects equality between the Sync Counter and any Sprite Position Register.
 - Six (6) lines of Bitplane & eight (8) pairs of serial data go to Priority control logic which selects only one (1) of the Sprites or one (1) of the separate Bitmap images to produce the five (5) bit color select code at its output. This five (5) bit code then selects one of the thirty-two (32) color registers to produce the twelve (12) bit RGB video output.
 - The Bitplane and Sprite serial lines also go to the Collision Detect Logic, which detects real time coincidence between them, and sets appropriate bits in the Collision Storage register. This register is read and cleared by the 68000.
 - The four (4) “mouse counters” are controlled by the two (2) mouse-joystick connectors. These count the pulses representing the horizontal and vertical motion of two (2) “mouse” controllers, and are read by the 68000.

CHIP ELEMENTS

32 Color Registers; Bitplane Priority and Control Registers; Color Select Decoder; Priority Control Logic; 16 Sprite Serial Lines; Sprite Data Registers; Bit Plane Control Registers; Two (2) Mouse Connectors; Sprite Position Compare Logic; Sprite Horizontal Control Registers; Bit Plane Serializer Collision Detect Logic; Collision Control Register; Collision Storage Register; Buffer — Data Bus; Buffer — Register Address Decode; Bit Plane Data Registers Video: RGB; Sprite Serialization.

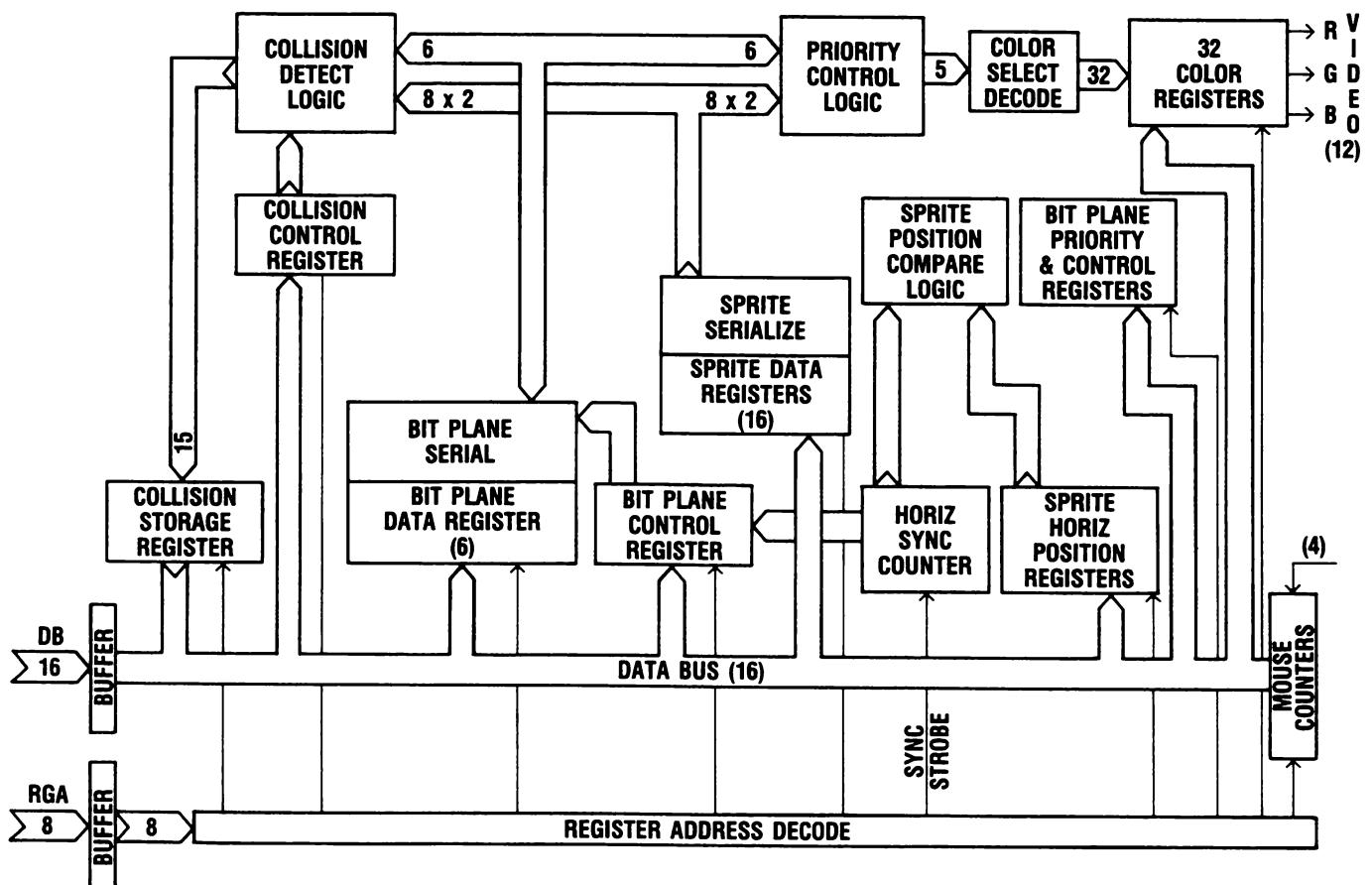


Figure 1-19. Denise Block Diagram

2.12.1 AMBER

INTRODUCTION

The AMBER chip is a custom 68-pin (PLCC), 2 micron CMOS gate array designed to incorporate the necessary circuitry to operate as a video controller in the Display Enhancer circuitry whose function is to either de-interlace or scan-double the Amiga's video data. The AMBER chip will operate in either NTSC or PAL Amiga modes, incorporates genlock support circuitry, and works with the new ECS modes.

FUNCTIONAL DESCRIPTION

The AMBER gate array is the heart of the Display Enhancer system that includes a field store and line memorys, a PLL based pixel clock generator, and an RGB D/A convertor. The AMBER chip itself controls the field store and line memorys based on the operational mode that the Amiga is in which is detected by the two sync signals, _HSYNC and _VSYNC. The AMBER chip functions as the pixel data path selector, depending on the Amiga's mode of operation, and outputs the modified pixel data for the DAC to convert to analog RGB for the display. AMBER does not contain any system accessible registers or I/O ports and is thus not in the Amiga's memory map. Operation is totally dependent on detecting the video display mode that the Amiga is currently working in and automatically modifying the field store, line memorys, and pixel data path in order to improve the display quality for the user. The AMBER gate array also incorporates three different field memory controllers so that several non-compatible types of video field memorys can be used in the system. This allows for flexibility in implementing the most cost-effective solution for enhancing the Amiga's video output.

OPERATION IN AMIGA VIDEO MODES

As the heart of the Display Enhancer circuitry, the AMBER gate array must detect and modify its operation according to the Amiga's video mode. This can best be described by breaking down the various Amiga modes as follows.

NTSC AND PAL DETECTION

Internally, AMBER has both a line and pixel position counter that is slaved to the Amiga by resetting when the high-to-low transition of the _HSYNC and _VSYNC signals is sampled. The vertical counter counts in half-line increments so that proper sync positioning can occur. Using this half-line count AMBER can detect which Amiga video format the graphics system is operating in by decoding the count when the next _VSYNC reset occurs. The decode is set for the NTSC format (either interlaced or non-interlaced modes) and, if not decoded, will assume that the Amiga is operating in the PAL (or PAL like) video format. This is very useful because non-standard video formats can be created using the programmable features of the ECS chip set and the Display Enhancer will try to de-interlace or scan-double the video. In this fashion, under AmigaDos V2.0, the programmer can create a high-quality 70 Hz refreshed, 31.5 KHz display with no system overhead or performance degradation. The AMBER gate array interprets the Amiga's video mode for NTSC as having 910 pixel clocks per line and a 60 Hz (262 or 263 lines in non-interlaced mode, 262.5 lines in interlaced mode) refresh rate. The PAL Amiga video format is interpreted as having 908 pixel clocks per line and a 50 Hz (312 or 313 lines in non-interlaced mode, 312.5 lines in interlaced mode) refresh rate.

INTERLACED AMIGA VIDEO MODE

In the interlaced Amiga video modes (i.e., 320 x 400/512 and 640 x 400/512) the AMBER gate array determines it is in interlaced mode and automatically switches into de-interlace mode. This means that the field store is storing the previous video field, which is then input to a bank of line memorys, and AMBER is combining it with the current incoming video field, which is also input to a bank of line memorys, so that AMBER creates a progressive scan of both previous and current fields at twice the video rate. The horizontal line rate is thus increased from 15.734 KHz NTSC or 15.625 KHz PAL to 31.46 KHz NTSC or 31.25 KHz PAL. This progressive scan or de-interlacer thus removes the interfield and interline flicker, as well as blank scan lines, so that the resulting video is greatly improved. A side-effect of progressive scan conversion however is motion artifacts which are introduced with rapidly moving objects.

2.12.1 AMBER (Continued)

NON-INTERLACED AMIGA VIDEO MODE

In the non-interlaced Amiga video modes (i.e. 320 x 200/256 and 640 x 200/256) the AMBER gate array determines it is in non-interlaced mode and automatically switches into scan-double mode. This means that the AMBER chip is using a bank of line memories to repeat every scan line at twice the video rate to remove visible scan lines from images without introducing motion artifacts. This action is sometimes called a 'line-repeat' function. The horizontal line rate is therefore increased from 15.734 KHz NTSC or 15.625 KHz PAL to 31.46 KHz NTSC or 31.25 KHz PAL. The AMBER gate array can be forced to stay in scan-double mode instead of automatically switching to de-interlacer mode by putting a logic 0 on the N_SCNDBL pin (pin 67). This would allow the system designer to create an inexpensive scan-double only video enhancer for the Amiga which could be upgraded into a full de-interlacer/scan-doubler system by adding the field store memories at the users discretion.

ECS PRODUCTIVITY AMIGA VIDEO MODE

With the ECS chip set the AMBER gate array can detect the increased horizontal video rate when the Amiga is put into productivity mode (either non-interlaced or interlaced) and automatically bypasses the original Amiga video coming out to the ECS Denise chip and sends it to the RGB DAC for the display. This is necessary because the field store and line memories can only sample pixel data every 70nS, however, when in this video mode, the Amiga sends out pixel data every 35nS.

ECS SUPERHIRES AMIGA VIDEO MODE

Using the ECS chip set when in the superhiress video mode (either interlaced or non-interlaced), the field store and line memories have the same sampling problem as with the ECS productivity mode. In this case the AMBER chip cannot detect this as a separate mode in which it should automatically bypass the video data. Therefore the user must toggle a mode switch (to logic 0) which is connected to a pin on the AMBER gate array (pin 65) thereby forcing the chip into bypass (also called 'disable') mode.

NTSC AND PAL GENLOCK SUPPORT FUNCTIONS

The AMBER gate array also is capable of simultaneous operation as a de-interlacer in both NTSC and PAL environments when a genlock is connected to the Amiga system IF the genlock is properly designed as per the Commodore specification. The operation of the AMBER gate array will never disturb the genlock regardless of operational mode. Since the genlock in the system is creating a modified HSYNC and VSYNC sync signals, and only creates these sync signals in an interlaced video mode format, the AMBER chip will only operate as a de-interlacer of the system's video. In this operational mode the AMBER gate array is creating the correct sync signals for output to the monitor and must be set to the 'enable' mode (logic 1) using the bypass switch connected to pin 65 of the gate array.

2.12.2 DISPLAY ENHANCER

BACKGROUND INFORMATION

In Amiga video there are two primary modes of operation, *interlaced* and *non-interlaced*. The Amiga sends out video information in ‘fields’ of horizontal scan lines to make up a display on the monitor. In the interlaced mode, there are two fields, called even and odd, which when taken together make up a complete video frame. These fields are displaced from each other on the screen so that their respective scan lines are interleaved. In other words, the lines of the even field alternate vertically with the lines of the odd field. From top to bottom on the screen there is one line of the even field, then one line of the odd field, another line of the even field, etc., hence the term “interlace”.

The video frame rate is 30 frames per second, so each field (half a frame) takes 1/60th of a second to be drawn on the screen — the standard Amiga hardware is not capable of sending out video data any faster than this. In the time that it takes for the electron beam to draw the second of the current interlaced frame’s two separate fields, the first field has faded in brightness, since the glow of the phosphor dots the beam scans starts to dim as soon as the beam passes to the next pixel. When the refresh rate (how many times per second each field is redrawn) is too low — 30 Hz or less — for the human eye to see as a single image, the perceived result is the annoying flicker of interlace mode. Non-interlace screens are essentially the SAME field scanned TWICE each frame, or sixty times per second, which is rapid enough to avoid the flicker.

With that background information we can now explain how the Display Enhancer removes the flicker from the Amiga’s interlaced display modes, and visible scan lines in non-interlaced display modes.

The Display Enhancer is capable of operation in all Amiga graphics modes, to which it automatically adapts. In the non-interlaced graphics modes (i.e. 320 x 200 and 640 x 200) the Display Enhancer will automatically operate in scan-double mode. This means that the circuit is repeating every line, at twice standard video rate, to increase the raster scan rate from 15.734 KHz NTSC (15.625 KHz PAL) to 31.46 KHz NTSC (31.25 KHz PAL). In this mode of operation, the Display Enhancer removes visible scan lines from all images, fully supporting overscan and HAM displays, yielding a crisp and solid display. Another benefit of this mode of operation is that there are no motion artifacts, (i.e. shearing or ghosting effects) which has obvious value for animation and game software which has rapidly-moving objects.

In the interlaced Amiga modes (i.e. 320 x 400 and 640 x 400) the Display Enhancer automatically switches into de-interlace mode. This means that the Display Enhancer is storing the previous video field and combining it with the current incoming video field at twice the video rate, to remove visible scan lines without adding flicker, while fully supporting overscan and HAM display modes. In this mode of operation you may notice some motion artifacts (i.e. a ghosting effect) with rapidly animated objects.

The Display Enhancer has also been designed to be compatible with the new graphics modes available with the Enhanced Chip Set (ECS) under AmigaDOS 2.0.

The ECS ‘Productivity Mode’ has display of 640 x 400 pixels with 4 colors out of a palette of 64. In this display mode the Display Enhancer will automatically bypass the video data to the multisync monitor since the raster scan rate is already at 31.46 KHz and has no flicker or visible scan lines.

The ECS ‘Superhires Mode’ has a display of either 1280 X 200 or 1280 X 400 pixels with 4 colors out of a palette of 64, and a raster scan rate of 15.734 KHz NTSC (15.625 KHz PAL). In this display mode the Display Enhancer only samples every other pixel and may cause a distorted display, so the card must be put into bypass mode by setting the mode switch to the right.

BASIC BUILDING BLOCKS

The major system blocks of the Display Enhancer are:

- PLL (Phase Locked Loop) clock generator
- 384K Field store and Line memories
- Video controller gate array (AMBER)
- Hybrid D/A converter

2.12.2 DISPLAY ENHANCER (Continued)

PLL (PHASE LOCKED LOOP) CLOCK GENERATOR

The PLL clock generator section consists of an NE564 PLL chip and is designed as a frequency multiplier with a user-adjustable phase control (the "fine tuning" control). In order to create a de-interlaced or scan-doubled image, two clocks that are related to the Amiga's system clock must be generated. These two clocks are 14.3 MHz (14.1 MHz PAL), and the main clock of 28.63 MHz (28.375 MHz PAL) for NTSC-based systems. Since the video slot only provides a 3.58 MHz (3.55 MHz PAL) clock, the frequency multiplier approach was used. The PLL circuit takes the available 3.58 MHz (3.55 MHz PAL) and creates the main clock of 28.63 MHz (28.375 MHz PAL) and, using a divider chain, generates the 14.3 MHz (14.1 MHz PAL) and 3.58 MHz (3.55 MHz PAL) clocks. The PLL then tracks the incoming 3.58 MHz with the divided-down 28.63 MHz clock so that the generated clocks now have a fixed relationship to the Amiga system clock.

Another problem that the PLL circuit must address is that the pixels coming from the Amiga's Denise chip at 14 MHz (or 28 MHz in ECS modes) do not have a constant phase relationship, across process variations, to the available 3.58 MHz clock. This problem requires the use of a phase circuit that must be user-adjusted so that the Display Enhancer's generated clocks correctly sample the incoming pixels from the Amiga, since individual machines vary slightly. The PLL circuit must also be able to handle two different clock frequencies. The NTSC and PAL Amigas' main clock frequencies of 28.636 MHz/28.37516 MHz (less than a 1% difference) require the use of a trimmer capacitor (coarse adjustment control) to set a median frequency which is close enough so that regardless of which system the Display Enhancer is in, the PLL can function correctly.

384K FIELD STORE AND LINE MEMORIES

The Display Enhancer uses three 1-MBit CMOS Frame memories to store a field of 12-bit-wide pixels. The frame memories allow simultaneous read/write action in such a manner that as one new 12-bit-wide pixel is being written, the previous field's 12-bit-wide pixel, at the same line/field location, is being read. The AMBER gate array controls the reading and writing of the frame memories so that this action happens properly. The other major element of the Display Enhancer memory section is the line memories. The line memories are used to double the pixel rate of the incoming video data, as well as for the field-delayed data from the frame memories. The output of each pair of line memories (two for the current line data and two for the field-delayed data) is sent to the gate array controller for multiplexing and reclocking of the video data.

VIDEO CONTROLLER GATE ARRAY (AMBER)

The heart of the Display Enhancer system is the AMBER gate array chip. This chip provides synchronization and control signals for the correct operation of the frame memories and line memories, as well as the proper horizontal and vertical sync signals that control the multisync monitor. The AMBER chip also controls the pixel data path so that the correct line (i.e. current or delayed) is sent to the D/A converter, depending on the operational mode (i.e. de-interlace or scan-double mode). The AMBER chip must also detect whether the Display Enhancer is operating in a PAL or NTSC Amiga system and whether or not there is an external genlock in the system, to adjust the Display Enhancer control functions accordingly.

HYBRID D/A CONVERTER

The de-interlaced/scan-doubled 12-bit-wide digital RGB data coming from the AMBER gate array must be converted to analog RGB for the multisync monitor to display the image. A custom hybrid digital-to-analog converter using a simple, laser-trimmed weighted resistor network is employed for this conversion process. This D/A converter is capable of directly driving a 75-ohm load.

2.12.3 PAL AND NTSC AMIGA GENLOCK INTERFACE GUIDELINES

The following describes the Video Beam Counters “reset” mechanism on the current Agnus devices when an external device (i.e. genlock) applies sync pulses on the HSY* and VSY* pins and the CBM recommended method of interfacing genlock devices to the Amiga for current and future compatibility.

Affected devices: AGNUS 8372 (ECS) and beyond.

HARDWARE INTERPRETATION OF PROPER NTSC AND PAL HSY* AND VSY* RESET PULSES

HORIZONTAL COUNTER

After Agnus is configured to accept external syncs, by setting the ERSY bit in the BPLCON0 register, the horizontal counter will respond as follows:

NTSC MODE

- If the external HSY* reset is NOT applied at the end of a “short line”, the next line will be a “long line”, as normal. If no HSY* is applied at the end of a “long line”, the horizontal counter will roll-over and is held reset at count “00”. The counter will resume counting once the external HSY* pulse is applied.
- If the external HSY* reset is applied at the end of a “short line”, the following line is forced to be another “short line”.
- If the counter reaches the end of a “long line” when the external HSY* is asserted, the next line is a “short line” and will start at count “01” (as opposed to count “00”). Otherwise, if no reset occurs the counter will roll-over to count “00” and stop.

PAL MODE

In PAL mode the counter will operate with “short lines” at all times (“long lines” are disabled) and will respond as follows:

- If the external HSY* is not applied at the end of line, the horizontal counter will roll-over to the beginning of the next line and is held at count “00”, until the next HSY* pulse occurs.
- If the external HSY* is applied when the counter reaches the end of line, the next line will start at count “01”, skipping count “00”.

VERTICAL COUNTER

The vertical counter operates in the same manner for both NTSC and PAL modes. After Agnus is configured to accept external syncs (ERSY bit in BPLCON0), the vertical counter will respond as described below.

INTERLACE MODE (LACE bit is set in BPLCON0)

- If an external VSY* pulse is applied, the vertical counter will be reset to count “000” (first line) and the “long field” condition is set at the beginning of the next horizontal line (at horizontal count “02”) in NTSC mode. In PAL mode, the long field is set under these conditions.
- If the external VSY* pulse is not asserted during the end of a short (long) field, the vertical counter will roll-over and start a long (short) field sequence.

NON-INTERLACE MODE (LACE bit deasserted)

In Non-interlace mode the vertical counter is set to operate with either long or short fields, depending on the last value that the software wrote to the FRAME bit.

In this mode, whenever an external VSY* pulse is applied the counter is reset to line count “000” (first line) at the beginning of the next horizontal line.

PULSE DURATION

The width of the active-low vertical (VSY*) pulse should be less than or equal to one line duration (63.556 uS NTSC, 63.999 uS PAL) but greater than one-half line duration (31.77 uS NTSC, 31.999 uS PAL) and should be asserted during the beginning of a horizontal line, at which time the vertical logic is triggered and the “first line” is started.

The width of the active-low horizontal (HSY*) pulse should be greater than or equal to eight hi-res pixels (0.558 uS for NTSC or 0.563 uS for PAL) for proper operation.

2.12.3 PAL AND NTSC (Continued)

NTSC AND PAL GENLOCK INTERFACE GUIDELINES

In order to synchronize the Amiga to an external source, the genlock device must provide reset pulses to the Amiga's sync lines which have the effect as detailed in the above section. The proper method of thus applying these reset pulses will now be discussed.

NTSC AMIGA MODELS

The genlock device must provide the Amiga with horizontal and vertical reset pulses with the following rates:

- HSY* line: Active-low pulse of proper duration every two lines (i.e. 127.11 uS or 7.8671 KHz)
- VSY* line: Active-low pulse of proper duration every two fields (i.e. 33.36 mS or 29.97 Hz)

Note that for the VSY* reset pulse that at the time of the reset pulse the next field will be an even field (long field).

PAL AMIGA MODELS

The genlock device must provide the Amiga with horizontal and vertical reset pulses with the following rates:

- HSY* line: Active-low pulse of proper duration every line (i.e. 63.99 uS or 15.625 KHz)
- VSY* line: Active-low pulse of proper duration every two fields (i.e. 39.99 mS or 25.0 Hz)

Note that for the VSY* reset pulse that at the time of the reset pulse the next field will be an even field (long field).

BOTH reset pulses must be generated regardless of whether or not source video is being input to the genlock device!!

In addition to providing the Amiga with the proper horizontal and vertical reset pulses, the genlock device must provide the Amiga with a system clock that is synchronized with the external video. One method of doing this would be to multiply up the video source's line duration to the Amiga's system clock frequency, thus making a line-locked clock that also free runs at the correct frequency when no video is present. For correct operation of the Amiga as a multitasking system the proper system clock frequency appearing on the XCLK pin of either the video slot or the 23-pin video port MUST be the following for NTSC and PAL Amiga systems:

- NTSC XCLK: 28.63636 MHz
- PAL XCLK: 28.375156 MHz*

*It has been noted that some PAL genlock designers provide the Amiga's XCLK input with a 28.250 MHz clock frequency. **THIS IS NOT RECOMMENDED FOR PROPER AMIGA OPERATION!**

2.13 AMIGA EXPANSION BUS

POWER CONNECTIONS

The expansion bus provides several different voltages designed to supply expansion devices. These are basically the same for the Zorro III bus as they were for the Zorro II bus, with the exception of one pin, and that the specification has been clarified a bit. Note that all Zorro III PICs must list their power consumption specifications.

Digital Ground (Ground)

This is the digital supply ground used by all expansion cards as the return path for all expansion supplies.

Main Supply (+5VDC)

This is the main power supply for all expansion cards, and it is capable of sourcing large currents; each PIC can draw up to 2.0 Amps @ +5VDC.

Negative Supply (-5VDC)

This is a negative version of the main supply, for small current loads only; each PIC can draw up to 60 mA @ -5VDC.

High Voltage Supply (+12VDC)

This is a higher voltage supply, useful for communications cards and other devices requiring greater than digital voltage levels. This is intended for relatively small current loads only; each PIC can draw up to 500mA @ +12VDC.

Negative High Supply (-12V)

Negative version of the high voltage supply, also used in communications applications, and similarly intended for small loads only; each PIC can draw up to 60 mA @ -12VDC.

CLOCK SIGNALS

The expansion bus provides clock signals for expansion boards. The main use for these clocks on Zorro III cards is bus arbitration clocking. There is no relationship between any of these clocks and normal Zorro III bus activity.

/C1 Clock

This is a 3.58 MHz clock (3.55 MHz on PAL systems) that is synched to the falling edge of the 7M system clock.

/C3 Clock

This is a 3.58 MHz clock (3.55 MHz on PAL systems) that is synched to the rising edge of the 7M system clock.

CDAC Clock

This is a 7.16 MHz system clock (7.09 MHz on PAL systems) which trails the 7M clock by 90° (approximately 35ns).

E Clock

This is the 68000 generated "E" clock, used for 6800 family peripherals driven by "E" and 6502 peripherals driven by Φ2. This clock is four 7M clocks high, six clocks low, as per the 68000 spec.

7M Clock

This is the 7.16 MHz system clock (7.09 MHz on PAL systems). This clock drives the bus master registration mechanism for Zorro III bus masters.

SYSTEM CONTROL SIGNALS

The signals in this group are available for various types of system control; most of these have an immediate or near immediate effect on expansion cards and/or the system CPU itself.

Hardware Bus Error/Interrupt (/BERR)

This is a general indicator of a bus fault condition of some kind. Any expansion card capable of detecting a hardware error relating directly to that card can assert /BERR when that bus error condition is detected, especially any sort of harmful hardware error condition. This signal is the strongest possible indicator of a bad situation, as it causes all PICs to get off the bus, and will usually generate a level 2 exception on the host CPU. For any condition that can be handled in software and does not pose an immediate threat to hardware, notification via a standard processor interrupt is the better choice. The bus controller will drive /BERR in the event of a detected bus collision or DMA error (an attempt by a bus master to access local bus resources it does not have valid access permission for). All cards must monitor /BERR and be prepared to tri-state all of their on-bus output buffers whenever this signal is asserted. An expansion bus master will attempt to retry a cycle aborted by a single /BERR and notify system software in the case of two subsequent /BERR results. Since any number of devices may assert /BERR, and all bus cards must monitor it, any device that drives /BERR must drive with an open collector or similar device, and any device that monitors /BERR should place a minimal load on it. This signal is pulled high by a passive backplane resistor.

2.13 AMIGA EXPANSION BUS (Continued)

SYSTEM CONTROL SIGNALS (Continued)

System Reset (/RESET, /IORST)

The bus supplies two versions of the system reset signal. The /RESET signal is bidirectional and unbuffered, allowing an expansion card to hard reset the system. It should only be used by boards that need this reset capability, and is driven only by an open collector or similar device. The /IORST signal is buffered output-only version of the reset signal that should be used as the normal reset input to boards not concerned with resetting the system on their own. All expansion devices are required to reset their autoconfiguration logic when /IORST is asserted. These signals are pulled high by passive backplane resistors.

System Halt (/HLT)

This signal is driven, along with /RESET, to assert a full-system reset. A full-system reset is asserted on a powerup reset or a keyboard reset; any PIC that needs to differentiate between full system and I/O reset should monitor /HLT and /IORST unless it also needs to drive a reset condition. This is driven with an open-collector output, or the equivalent, and pulled up by a backplane resistor.

System Interrupts

Six of the decoded 680x0 interrupt inputs are available on the expansion bus, and these are labelled as /INT1, /INT2, /INT4, /INT5, /INT6, /INT7. Each of these interrupt lines is shared by wired ORing, thus each line must be driven by an open-collector or equivalent output type. Zorro III interrupts can be handled Zorro II style, via autovectors and daisy-chained polling, or they can be vectored using the quick interrupt protocol. Zorro III cards may use any of these interrupt lines; system hardware supports them all properly. Any card that is designed for operation in both Zorro II and Zorro III backplanes must only use /INT2 or /INT6. These signals are pulled high by passive backplane resistors.

SLOT CONTROL SIGNALS

This group of signals is responsible for the control of things that happen between expansion slots.

Slave (/SLAVEN)

Each slot has its own /SLAVEN output, driven actively, all of which go into the collision detect circuitry. The “N” refers to the expansion slot number of the particular /SLAVE signal. Whenever a Zorro III PIC is responding to an address on the bus, it must assert its /SLAVEN output very quickly. If more than one /SLAVEN output occurs for the same address, or if a PIC asserts its /SLAVEN output for an address reserved by the local bus, a collision is registered and the bus controller asserts /BERR. The bus controller will assert /SLAVEN back to the interrupting device selected during a Quick Interrupt cycle, so any device supporting Quick Interrupts must be capable of tri-stating its /SLAVEN; all others can drive SLAVEN with a normal active output.

Configuration Chain (/CFGINN, /CFGOUTN)

The slot configuration mechanism uses the bus signals /CFGOUTN and /CFGINN, where “N” refers to the expansion slot number. Each slot has its own version of both signals, which make up the *configuration chain* between slots. Each subsequent /CFGINN is a result of all previous /CFGOUTs, going from slot 0 to the last slot on the expansion bus. During the autoconfiguration process, an unconfigured Zorro PIC responds to the 64K address space starting at \$00E80000 if its /CFGINN signal is asserted. All unconfigured PICs start up with /CFGOUTN negated. When configured, or told to “shut up”, a PIC will assert its /CFGOUTN, which results in the /CFGINN of the next slot being asserted. Backplane logic automatically passes on the state of the previous /CFGOUTN to the next /CFGINN for any slot not occupied by a PIC, so there is no need to sequentially populate the expansion bus slots.

Backplane Type Sense (SenseZ3)

This line can be used by the PIC to determine the backplane type. It is grounded on a Zorro II backplane, but floating on a Zorro III backplane. The Zorro III PIC connects this signal to a 1K pullup resistor to generate a real logic level for this line. It is possible, though more complicated, to build a Zorro III PIC that can actually run in Zorro II mode when in a Zorro II backplane. It is hardly necessary or required to support this backward compatibility mechanism, and in many cases it will be impractical. The Zorro III specification does require that this signal be used, at least, to shut the card down and pass /CFGIN to /CFGOUT when in a Zorro II backplane.

2.13 AMIGA EXPANSION BUS (Continued)

DMA CONTROL SIGNALS

There are various signals on the expansion bus that coordinate the arbitration of bus masters. Zorro II bus masters use some of the same logical signals, but their arbitration protocol is considerably different.

PIC is DMA Owner (/OWN)

This is asserted by the bus controller when a master is about to go on the bus and indicates that some master owns the bus. Zorro II bus masters drive this, and some Zorro III slaves may find a need to monitor it, or /BGACK, to determine who is the bus master. This is ordinarily not important to Zorro III PICs, and they may not drive this line.

Slot Specific Bus Arbitration (/BRN,/BGN)

These are the slot-specific /BRN and /BGN signals where "N" refers to the expansion slot number. The bus request from each board is taken in by the bus controller and ultimately used to take over the system from the primary bus master, which is always the local master. Zorro III PICs toggle /BRN to register or unregister as a master with the bus controller. /BGN is asserted to one registered PIC at a time, on a cycle by cycle basis, to indicate to the PIC that it gets the bus for one full cycle.

Bus Grant Acknowledge (/BGACK)

Asserted by the bus controller when a master is about to go on the bus. As with /OWN, most Zorro III PICs ignore this signal, and none may drive it.

Bus Want/Clear (/BCLR)

This signal is asserted by the bus controller to indicate that a PIC wants to master the bus; Zorro III cards can use this to determine if any Zorro II bus requests are pending; Zorro III bus requests do not affect /BCLR.

ADDRESS AND RELATED CONTROL SIGNALS

These signals are various items used for the addressing of devices in Zorro III mode by bus masters either on the bus or from the local bus. The bus controller translates local bus signals (68030 protocol on the A3000) into Zorro III signals; masters are responsible for creating the appropriate signals via their own bus control logic.

Read Enable (READ)

Read enable for the bus; READ asserted during a bus cycle indicates a read cycle, READ negated indicates a write cycle. READ is asserted at address time, prior to /FCS, for a full cycle, and prior to /MTCR for a short cycle. READ stays valid throughout the cycle; no latching required.

Multiplexed Address Bus (A8-A31)

These signals are driven by the bus master during address time, prior to the assertion of /FCS. Any responding slave must latch as many of these lines as it needs on the falling edge of /FCS, as they are tri-stated very shortly after /FCS goes low. These addresses always include all configuration address bits for normal cycles, and the cycle type information for Quick Interrupt cycles.

Short Address Bus (A2-A7)

These signals are driven by the bus master during address time, prior to the assertion of /FCS, for full cycles, and prior to the assertion of /MTCR for short cycles. They stay valid for the entire full or short cycle, and as such do not need to be latched by responding slaves.

Memory Space (FC0-FC2)

The memory space bits are an extension to the bus address, indicating which type of access is taking place. Zorro III PICs must pay close attention to valid memory space types, as the space type can change the type of the cycle driven by the current bus master. The encoding is the same as the valid Motorola function codes for normal accesses. These are driven at address time, and like the low short address, are valid for an entire short or full cycle.

FC0	FC1	FC2	Address Space Type	Z3 Response
0	0	0	Reserved	None
0	0	1	User Data Space	Memory
0	1	0	User Program Space	Memory
0	1	1	Reserved	None
1	0	0	Reserved	None
1	0	1	Supervisor Data Space	Memory
1	1	0	Supervisor Program Space	Memory
1	1	1	CPU Space	Interrupts

Figure 1-20. Memory Space Type Codes

2.13 AMIGA EXPANSION BUS (Continued)

ADDRESS AND RELATED CONTROL SIGNALS (Continued)

Compatibility Cycle Strobe (/CCS)

This is equivalent to the Zorro II address strobe, /AS. A Zorro III PIC does not use this for normal operation, but may use it during the autoconfiguration process. AUTOCONFIG™ cycles always look like Zorro II cycles, though of course /FCS and the full Zorro III address is available, so a card can use either Zorro II or Zorro III addressing to start the cycle. However, using the /CCS strobe can save the designer the need to compare the upper 8 bits of address when snooping for the \$00E8xxxx autoconfig address, so acting as a Zorro II device is often the simpler case. Data must be driven Zorro II style, though if the /DSN lines are respected for reads, /CINH is asserted, and /MTACK is negated, the resulting Zorro III cycle will fit within the expected Zorro II cycle generated by the bus controller. Yes, that should sound weird; it is based on the mapping of Zorro II vs. Zorro III signals, and of course the fact that /FCS always starts any cycle. Also note that a bus cycle with /CCS asserted and /FCS negated is always a Zorro II PIC-as-master cycle.

Full Cycle Strobe (/FCS)

This is the standard Zorro III full cycle strobe. This is asserted by the bus master shortly after addresses are valid on the bus, and signals the start of any kind of Zorro III bus cycle. Shortly after this line is asserted, all the multiplexed addresses will go invalid, so in general, all slaves latch the bus address on the falling edge of /FCS. Also, /BGN line is negated for a Zorro III mastered cycle shortly after /FCS is asserted by the master.

DATA AND RELATED CONTROL SIGNALS

The data time signals here manage the actual transfer of data between master and slave for both full and short cycle types. The burst mode signals are here too, as they are basically data phase signals even through they do not only concern the transfer of data.

Data Output Enable (DOE)

This signal is used by an expansion card to enable the buffers on the data bus. The bus master drives this line to keep slave PICs from driving data on the bus until *data time*.

Data Bus (D0-D31)

This is the Zorro III data bus, which is driven by either the master or the slave when DOE is asserted by the master (based on READ). It is valid for reads when /DTACK is asserted by the slave; on writes when at least one of /DSN is asserted by the master, for all cycle types.

Data Strobes (/DSN)

These strobes fall during *data time*; /DS3 strobes D24-D31, while /DS0 strobes D0-D7. For write cycles, these lines signal data valid on the bus. At all times, they indicate which bytes in the 32 bit data word the bus master is actually interested in. For cachable reads, all four bytes must be returned, regardless of the value of the sizing strobes. For writes, only those bytes corresponding to asserted /DSN are written.

Data Transfer Acknowledge (/DTACK)

This signal is used to normally terminate a Zorro III cycle. The slave is always responsible for driving this signal. For a read cycle, it asserts /DTACK as soon as it has driven valid data onto the data bus. For a write cycle, it asserts /DTACK as soon as it is done with the data. Latching the data on writes may be a good idea; that can allow a slave to end the cycle before it has actually finished writing the data to its local memory.

Cache Inhibit (/CINH)

This line is asserted at the same time as /SLAVEN to indicate to the bus master that the cycle must not be cached. If a device does not support caching, it must assert /CINH and actually obey the /DSN byte strobes for read cycles. Conversely, if the device supports caching, /CINH is negated and the device returns all four bytes valid on reads, regardless of the actual supplied /DSN strobes.

Multiple Cycle Transfers (/MTCR, /MTACK)

These lines comprise the Multiple Transfer Cycle handshake signals. The bus master asserts /MTCR at the start of *data time* if it is capable of supporting Multiple Transfer Cycles, and the slave asserts /MTACK with /SLAVEN if it is capable of supporting Multiple Transfer Cycles. If the handshake goes through, /MTCR strobes in the short address and write data as long as the full cycle continues.

2.14 RESERVED FOR FUTURE USE

2.15 AMIGA 32 BIT LOCAL SLOT

THE CPU SLOT

There is no A2000 style 86-pin CPU slot on the A3000. The processor cards designed for that slot have less power than the CPU contained standard in every A3000. A new 200 pin CPU expansion card connector has been provided as an upgrade path from the base machine to newer processors.

This new connector opens the door for a variety of products from ultra-high speed static RAM cards, to CACHE card products to new 68040 or RISC based processor cards.

2.16 BUSTER

This section details the Level 1 Fat Buster chip used in the Amiga 3000 computer. The Fat Buster is responsible mainly for expansion bus related operations in the A3000. It manages all bus arbitration in the system, all bus buffer control, and bus cycle control. It does all 68000 to 68030 conversions necessary to retain bus level compatibility with the Amiga 2000. The Level 1 Buster supports slave-mode non-burst 32 bit transfers as well.

Please Note: This is preliminary data and subject to change.

INTRODUCTION

The Fat Buster chip is a custom gate array designed as part of the Amiga 3000 system. Its main functions are to provide logical control of the Amiga 3000 bus, arbitration of all system bus masters, and the 68000 to 68030 logical bus conversion necessary to support Amiga 2000 (Zorro II) peripheral devices on the A3000 Bus. This version of the Buster chip directly supports arbitration between five expansion bus devices, one motherboard based SCSI device, and one motherboard based default bus master. The actual bus specification is contained in the document *The Amiga 3000 Expansion Bus*.

FAT BUSTER IN THE A3000

This specification is for the level 1 version of the Fat Buster device. This device fully supports the A2000 bus protocol, and partially supports the new A3000 bus (Zorro III) extention to the basic A2000 bus. Specifically, Fat Buster provides the following functions:

- Collision detects between bus slaves and the local bus.
- Control of address buffer enables.
- Control of data buffer enables, direction, and latching.
- Translation of local bus (68030) cycles into Zorro II cycles.
- Geographical cache enable mapping for Zorro II cycles.
- Translation of local bus (68030) cycles into non-burst Zorro III cycles.
- Translation of Zorro II DMA cycles into local bus (68030) cycles.
- Fair arbitration of five expansion and one local bus master.
- Asynchronous cycle holdoff for support of local bus caches.
- Bus locking for uninterruptable cycles in both Zorro II and Zorro III modes.

Additionally, the level 1 Fat Buster device is designed to be pin compatible with a future upgrade, the level 2 Fat Buster. This level 2 device is expected to provide the following additional features:

- Support for Zorro III Multiple Transfer (burst) cycles.
- Translation of local bus burst cycles into Zorro III burst cycles.
- Support for quickly arbitrated, interruptable, Zorro III bus masters.
- Arbitration for vectored interrupts from Zorro III PICs.

ZORRO II BUS DIFFERENCES

There are some minor differences between the Zorro II implementation provided by the Fat Buster chip, and the Zorro II bus resulting from a 68000 based A2000 implementation. None of these should make a big difference to any Zorro II PICs, but they should be noted. Any currently existing PIC that has a conflict with these differences is probably not designed according to proper Zorro II methodology, and would not be expected to work correctly on the A3000 or any other Fat Buster based computer.

6800 BUS INTERFACE

The major difference between Buster and Fat Buster is that the Fat Buster chip does not support the 68000's emulation protocol for 6800/6502 family peripherals. On the A2000, the /VPA and /VMA lines from the 68000 bus were available for potential use by PICs. The Zorro III bus uses these lines for new functions, and they are no-ops for Zorro II devices on the Zorro III bus. The main reasoning for these changes is primarily that the 6800 emulation mode is unsupported on the A2000 bus, and impossible to use within the Zorro II autoconfiguration mechanism as well.

2.16 BUSTER (Continued)

BUS MEMORY MAPPING

Another change to the Zorro II implementation is that the memory space occupied by the Zorro II bus is programmable via logic external to the Buster device. Current plans for the A3000 call for an additional area of memory, from \$00A00000 to \$00B7FFFF to be added to the Zorro II memory space as I/O memory. All other Zorro II memory is mapped the same as in the A2000 implementation. The one other major difference in mapping conventions is that Fat Buster does not create Zorro II cycles for memory outside of the Zorro II bus space, even if such spaces are within the 68000 address space. So, for example, a CPU access to chip memory would be visible to a Zorro II PIC in an A2000 backplane, but invisible to that same PIC in an A3000 backplane. Since this extra information on the Zorro II backplane can not be legally used by any PIC, however, this is not expected to cause any problems.

CACHE MAPPING

Cache mapping is a new feature, though it is actually implemented similarly on the A2630 (or A2500/30). For areas mapped as Zorro II memory, Buster will create a cachable cycle. It will negate the cache inhibit input to the local bus, and regardless of the actual transfer size, it will force full port width read cycles. For areas of the Zorro II bus space mapped as I/O, Buster will assert cache inhibit to the local bus and drive the data strobes according to the transfer size actually requested by the local bus master.

BUS SYNCHRONIZATION DELAYS

Due to the asynchronous nature of the local-to-expansion bus interface for Zorro II cycles, extra wait states may be occasionally added during locally mastered expansion cycles. Zorro II master access to local bus slaves will almost always result in at least one wait state. Additionally, a Zorro II bus master can only access 32 bit ports on the local bus. In the A3000 design, 32 bit ports are chip memory and fast memory, thus a DMA device will not be capable of accessing any custom chip registers. Attempts to access an 8 or 16 bit port will generate a bus error on both expansion and local buses.

BUS ARBITRATION ORDER

The Zorro II bus is now fairly arbitrated. The normal slot-based order of precedence is given to requesting devices, just as in the A2000 implementation. As always, once a master assumes bus mastership, it has the bus for as long as it wants the bus (of course, trouble results if a device takes the bus over too long). However, once a master gives up the bus, it will not be granted it back until all subsequent requests have been serviced. Bus arbitration by Fat Buster will be slightly slower due to the fairness logic, but it is impossible to jam the arbiter with asynchronous bus requests as with the A2000 Buster. The Fat Buster bus arbiter also holds off bus grants while hidden local bus cycles are in progress, so there is no guarantee of a minimum time between bus request and bus grant provided by Buster, though such a time can be calculated based on the properties of the A3000 system.

SLOPPY CYCLE SUPPORT

The Fat Buster cycle generator contains additional logic to handle *sloppy cycles*. A sloppy cycle is defined as a Zorro II cycle with some component that would naturally extend into a following cycle. Mainly because in this system, a Zorro III cycle may follow a Zorro II cycle, Fat Buster does not permit a new cycle to start until the previous cycle is completely finished. For Zorro II PICs that really follow the Zorro II specifications, this should have no effect. However, any Zorro II PICs that assert signals beyond the end of a cycle, especially critical signals like SLAVE or DTACK, will likely incur wait states on the Fat Buster defined expansion bus. This is not intended as a license for making sloppy expansion card designs, just an acknowledgement that some important devices are sloppy.

FAT BUSTER OPERATION

The Fat Buster design is not especially complicated, and can be broken down into several subsections. There are three main functional subsections of Fat Buster which are somewhat interrelated, but in general pretty distinct modules. These are the *Expansion Cycle Unit*, the *DMA Cycle Unit*, and the *Bus Arbitration Unit*. It is important to note that while this document and the A3000 Expansion Bus Specification both specify features of the Zorro III bus, this attempts to describe the Fat Buster implementation, while the A3000 documents specify the absolute constraints of any implementation of the Zorro III bus. Expansion cards should always be designed using the A3000 document. This document is useful for the understanding of the A3000 design and the Fat Buster II operation only.

2.16 BUSTER (Continued)

THE EXPANSION CYCLE UNIT

The purpose of the Expansion Cycle Unit is to manage the signals necessary to create any locally mastered expansion bus cycles. There are two basic kinds of locally mastered expansion cycles — the Zorro II (A2000 compatible) and Zorro III cycles. The start of any cycle is derived from the local bus activity. Specifically, /AS driven by the local bus master is driven into the Fat Buster chip by the current local bus master. Assuming the /WAIT input is negated and Buster is currently not in the recovery phase that follows any expansion cycle, a new expansion cycle will start. All expansion cycles are started with the assertion of /FCS. At the time /FCS is asserted, /ABOE0, /ABOE1, and /ABOE2 are asserted, and /DBOE0, /DBOE1, and /DBR16 are negated. The external bus cycle typing logic will assert /ADDRZ3, /MEMZ2, or /IOZ2 prior to /FCS if any expansion cycle is mapped at the current address. For all cycles, Buster will monitor the /SLAVEN inputs to create the bus buffer data direction signal /D2P and, if necessary, the bus collision signal, which results in the assertion of /BINT and /BERR. If no collision is detected, Buster will negate the appropriate address buffer enables, assert the appropriate data buffer enables, along with DOE, the expansion buffer enable signal, and the proper data strobes for the type and size of the transfer. Finally, the cycle's termination signal will be read as required and the cycle will be terminated. Any new local bus cycles will be ignored until a full recovery is made from the current expansion bus cycle.

ZORRO III CYCLES

Essentially all expansion bus cycles look like Zorro III cycles until /FCS is asserted. This is no problem for Zorro II PICs, since a Zorro III cycle can not be seen by a Zorro II PIC. After /FCS is asserted and a Zorro III cycle is indicated, /ABOE1 and /ABOE2 are negated — Zorro III PICs latch the higher order addresses on the fall of /FCS. This is the end of the *Address Phase* of the Zorro III cycle. The *Data Phase* follows with the assertion of DOE and both /DBOE0 and /DBOE1. After that, the data strobes are asserted based on the state of the local bus A0, A1, SIZ0, and SIZ1 lines. Nothing changes further until the /DTACK line is asserted by the expansion bus slave. For Zorro III cycles, /DTACK signals the end of the expansion bus cycle. This /DTACK is currently driven as /DSACK0 and /DSACK1 on the local bus; an implementation detail — the Level 2 Buster is likely to drive /STERM as well. /FCS, the data buffer enables, DOE, and the data strobes will be negated after the local bus master negates /AS. On the next state after that, the address buffer lines /ABOE1 and /ABOE2 are asserted, and Buster is in readiness for the next bus cycle. That cycle can not actually start until the bus slave negates /DTACK and /SLAVEN, and the local master negates /WAIT.

ZORRO II CYCLES

A Zorro II cycle is indicated by the assertion of either /MEMZ2 or /IOZ2 by the external bus cycle mapping logic. This will cause Buster to negate /ABOE2 very quickly and start the Zorro II state machine. Buster samples /FCS on the falling edge of CDAC, then clocks that result out on the rising edge 7M to create /CCS, which is equivalent to the 68000's /AS. If the cycle is a read cycle, the data strobes will be driven here too. For /MEMZ2 cycles, both /EDS3 and /EDS2 are driven, to support caching of Zorro II memory devices. For /IOZ2 cycles they are driven based on A0, A1, SIZ0, and SIZ1 on the local bus. Buster drives the data strobes for all write cycles just like the /IOZ2 read cycle, only they are driven on the second rising edge of the 7MHz clock, which is equivalent to the 68000 S4 state. DOE is also driven at S4 time. For write cycles, the data buffer enable /DBOE1 will be driven with /CCS to ensure data is set up by the time the data strobes fall. For read cycles, the data buffer enable /DBOE1 will be driven at S4 time to ensure /D2P has already switched the data buffers around before they are turned on. The data read latch line, DBLT, is asserted for either kind of Zorro II read cycle.

The /DTACK treatment is more complicated for a Zorro II cycle than for a Zorro III cycle. The bus slave has three options. If left along, the /DTACK logic in Buster will assert /DTACK on the bus at S4. The slave may instead assert the /MTCR signal, which it will interpret as the Zorro II XRDY line. This will hold Buster's internal /DTACK generator from creating the /DTACK line as long as it is asserted, adding wait states under the slave's control. Alternately, the slave may assert the CACHE signal, which is interpreted as the Zorro II /OVR line. This prevents any assertion of /DTACK by Buster, allowing the expansion bus slave to directly drive /DTACK with an open-drain or similar buffer. In any case, Buster starts looking for a /DTACK on the falling edge of the 7M clock S4 and every subsequent falling edge of 7M until /DTACK is negated. Once /DTACK is detected, Buster will negate its /CCS. DOE, data strobes, and data enables on the subsequent falling edge of the 7M clock. It expects the external data latch to latch expansion bus data that last falling 7M clock, at which point it will also assert /DSACK1 to the local bus. This mechanism requires the latching buffers for the 16 bit data path, but makes the bus buffer speed the only limiting factor in the speed of the local bus CPU that can be used (by design, of course, gate array technology will likely set more realistic limits).

2.16 BUSTER (Continued)

ZORRO II CYCLES (Continued)

Note that if the local bus master asserts /RMC, Buster will not negate /CCS at the end of a Zorro II bus cycle, but instead maintain /CCS and negate only the data strobes. This will generate the 68000-compatible read-modify-write cycle. Unfortunately, many Zorro II PICs did not properly support this form of bus locking, which results from the TAS instruction. As long as a particular PIC does support Zorro II style bus locking, the use of TAS with that PIC is perfectly acceptable in a Fat Buster driven expansion bus. Zorro II PICs that do not support bus locking will behave unpredictably when accessed by a TAS instruction.

DMA CYCLE UNIT

The DMA Cycle Unit is responsible for translating expansion bus cycles, when appropriate, into local bus cycles. This allows an expansion bus master to access local bus resources as if they were any other slave device. The Level 1 Fat Buster currently only supports the translation of Zorro II cycles into local bus cycles for 32 bit ports alone. This allows Zorro II bus masters to access the local bus chip memory, which is the most important slave resource on the local bus. The Level 2 Fat Buster will add additional support for Zorro III bus masters.

ON-BUS CYCLES

A DMA cycle starts with the assertion of either /CCS or /FCS on the expansion bus; since the Level 1 Buster does not arbitrate Level 2 DMA, this currently can only be /CCS. The external bus cycle mapping logic indicates whether the cycle is on the expansion bus by simply asserting either /MEMZ2 or /IOZ2 very quickly after /CCS is asserted. Should the cycle be on the expansion bus, Buster's main job is to make sure that all data buffer enables are negated. The /DTACK state machine in Buster runs for DMA cycles just as for non-DMA cycles to provide a default /DTACK for those slaves that require it.

EXPANSION-TO-LOCAL TRANSLATIONS

The translation of a Zorro II cycle into the 68030 cycle of the local bus is a rather tricky one. The main problem is that, for write cycles, Buster can not begin a local bus cycle until data is valid on the bus, S4 time, since there is no way to tell just how fast an arbitrary local bus cycle may be[§]. Additionally, the expansion bus cycle can not receive a /DTACK until the local bus terminates a cycle with /DSACK0 and /DSACK1 or STERM. And once the termination signal is received, the local bus data bus be held somehow until the local bus latches it on the falling edge of S6. And there are synchronizations required to avoid any possible metastable conditions.

The first step in the translation process is the start of the Zorro II cycle by the bus master's assertion of /CCS. This acts as the basic enable for the state machine that drives the 68030 compatible cycle on the local bus. The /CCS line asynchronously drives the address buffer enables /ABOE0, /ABOE1, and the /BIGZ line, causing a 32 bit representation of the expansion address to be driven on the local bus. The READ line is also asynchronously gated to R/W on the local bus.

The /CCS line and the data strobes /EDS3 and /EDS2, corresponding respectively with the Zorro II signals /BAS, /BUDS, and /BLDS, are clocked on both edges of the local bus clock CPUCLK to bring them in synchronized to local bus activity. Once synchronized, the appropriate data buffer enables are driven. For a write cycle, the 16 bit expansion data is driven on both sides of the 32 bit local bus by asserting /DBOE1 and /DBR16. For a read cycle, the level EA1, determines which of /DBOE1 and /DBR16 is necessary to drive the appropriate data onto the expansion bus. The SIZ0, SIZ1, A0 and A1 lines are driven to select the proper byte or bytes to access. On the next falling edge of CPUCLK, the /AS line to the local bus is driven. The local bus /DS is also driven if it is a read cycle, while for write cycles /DS follows on the next falling edge of CPUCLK, just like a normal 68030 cycle.

Everything stays this way until a termination signal is received. The /DSACK0 and /DSACK1 lines are sampled on the falling edge of the CPUCLK, while the /STERM line is sampled on the rising edge. Since the /DSACKs are by nature asynchronous, they are actually clocked on both the falling and rising edges of CPUCLK. If /DSACK0 and /DSACK1 do not indicate a 32 bit transfer, a bus error is signaled via /BERR, since only 32 bit ports are supported with this interface.

[§]The DMA, from Zorro II masters to the local bus is not purely arbitrary, but currently at least only supported to chip memory. This is a special case that may allow the expansion-to-local translation to be improved in the Level 2 Buster implementation.

2.16 BUSTER (Continued)

EXPANSION-TO-LOCAL TRANSLATIONS (Continued)

Once a positive termination signal is clocked in, /DTACK is generated on the expansion bus and the state machine locks into its termination state, which can only be cleared by the expansion strobe /CCS being negated. This is to support the fact that /STERM may only be asserted for a single clock period. For write cycles, the 68030 compatible /AS and /DS lines are negated on the first rising CPUCLK after the termination signal is received, just as usual on the 68030 bus. For read cycles, the /AS and /DS lines are kept asserted until /CCS is negated. This is a slight departure from the normal 68030 cycle, but something that all local bus peripherals know about. This is certainly an implementation detail of this system; it was done this way to save on the extra data bus latches and control signals that would have been needed for the cycle to exactly emulate the 68030 cycle while remaining Zorro II compatible.

BUS ARBITRATION UNIT

The Bus Arbitration Unit provides for unambiguous arbitration between the 68030 (or any other primary local bus master as supported with external arbitration logic), a local bus secondary master, and the five possible Zorro II expansion bus masters. All bus arbitration follows the Motorola three-wire convention of request, grant, and acknowledge; Zorro III does not, but that is nothing to worry about at present. All local bus arbitration will always be kept to the Motorola convention, allowing local bus DMA devices like hard disk controllers to work the same regardless of whether arbitrated by Buster or directly by the 68030 in some other system.

BUS REQUEST SYNCHRONIZATION

The local bus request, /SBR, can be driven asynchronously; the bus arbiter will clock this on both edges of the 7M clock, to synchronize it with the main arbiter, which runs based on the 7MHz clock. All the /BRN from the expansion bus are simply latched on the falling edge of 7M, which is safe, based on the Zorro II requirement that all /BRN are to be driven on the rising edge of 7M. The logical OR of all incoming requests at any time driven out on the falling edge of 7M as the /PBR line, which is the bus request to the primary local bus master. The 68030 will internally synchronize this line; other primary bus masters should make sure that they do the same.

BUS GRANT PRIORITIZATION

The primary local bus master, upon receipt of the /PBR line, will asynchronously assert the /PBG line into Buster. Buster gates /PBG with the appropriate signals on the expansion and local buses to avoid issuing a grant during a hidden cycle, then clocks it on the rising and falling edges of 7M to sync it with the rest of the arbiter. At this point, the prioritization logic has chosen from the highest of the pending /SBR or /BG0..../BG4. On the next rising 7M clock, that grant, now enabled by the synchronized /PBG, clocks out as /SBG or /BG0..../BG4. Subsequent bus grants can be generated for other requesting devices as soon as the winning device negates its request and the /PBG line cycles. The /PBG line is also clocked out asynchronously as the /BCLR line, which may be used by expansion bus devices to monitor whether the bus is busy or not.

BUS GRANT FAIRNESS

In the case of multiple bus requests at the same time, a fairness mechanism is employed. As long as /PBR remains in effect (as it will while any request is present) and a master currently owns either local or expansion bus, the arbiter sets a lockout bit for each grant given. This lockout bit prevents that grant from being given again until all pending grants are given. While the Zorro II DMA specification does not allow masters to be rescheduled once they have the bus, this does improve the overall fairness of the system in the event of multiple bus masters.

BUS GRANT ACKNOWLEDGE

In order to simplify the interlocking of the local and expansion buses during DMA < a Bus Grant Acknowledge on one bus is gated over to the other bus. The direction of this gating is determined by the /OWN line, which is asserted as a data buffer direction control by an expansion bus master. The arbiter must be careful to monitor bus activity on both sides of the bus before asserting the alternate BGACK, since each bus can have cycles that are hidden from the other.

2.17 SDMAC

INTRODUCTION

The purpose of this specification is to provide a description of Commodore's 4181-9B Super Direct Memory Access Controller (SDMAC). A description of SDMAC Commands, I/O locations and pin-out is included.

SDMAC DESCRIPTION

The SDMAC is an 84 pin (PLCC), 2 micron CMOS gate-array designed to enhance the performance of the Amiga A3000 computer and to reduce the cost of adding peripheral devices (i.e. SCSI and XT/AT devices) to the Amiga A3000.

Performance is enhanced by providing the computer with a full-speed 16MHz or 25MHz full 32-bit direct memory access controller that runs on the 68030 local fast bus. Data is transferred between Amiga system memory and a peripheral device by utilization of an internal 4 longword (32 bits/longword) FIFO, direct memory access and built-in byte-to-word and byte-to-longword funneling.

Over-all cost of adding peripheral devices to the Amiga is reduced due to the combined functionality of the SDMAC. The SDMAC provides direct connection from the Amiga's full 32-bit local fast bus to either a SCSI controller device (i.e. WD33C93A) or an 'XT/AT' compatible interface. Besides providing direct memory access and bus arbitration, the SDMAC also contains a 16 byte FIFO (eliminating the need for external static buffering), interrupt control logic, and byte or word (8 or 16 bit peripheral data) to word or longword (16 or 32 bit Amiga data) funneling. The SDMAC thus significantly reduces external hardware and manufacturing cost.

I/O DEFINITIONS AND LOCATIONS

The following I/O addresses refer only to offset locations.

SDMAC PERIPHERAL I/O MEMORY MAP

Super-Dmac (SDMAC) Base Address: \$00DD0000 hex

Note that the address decoding for the SDMAC base address is done by the FAT GARY gate array and generates the signal selecting the SCSI (pin 45) chip select pin on the SDMAC for addresses from \$00DD0000 to \$00DD3fff hex.

HEX Location	Symbolic Name	Definition	Type
00	DAWR	DACK Width Register	(write only)
04	WTC	Word Transfer Count Register	(read/write)
08	CNTR	Control Register	(read/write)
0C*	ACR	Address Count Register	(read/write)
10	ST __DMA	Start DMA Transfers	(rd/wr-strobe)
14	FLUSH	Flush FIFO	(rd/wr-strobe)
18	CINT	Clear Interrupts	(rd/wr-strobe)
1C	ISTR	Interrupt Status Register	(read only)
20,24,28,2C	—	Reserved	
30,34,38	—	"	
3C	SP __DMA	Stop DMA Transfers	(rd/wr-strobe)
40 - 4C	PORT0	8 BIT Peripheral Port (SCSI)	(read/write)
50 - 5C	PORT1A	8 BIT Peripheral Port (XT#0)	(read/write)
60 - 6C	PORT2	8 BIT Peripheral Port (XT#1)	(read/write)
70 - 7C	PORT1B	16 BIT Peripheral PORT (AT)	(read/write)

NOTE: ACR and WTC are 32 bit registers. All other registers and ports are 8 bits or smaller and are located on the low order data bus (D0-D7) with the exception of PORT1B which is 16 bits and located on the high order data bus (D16-D31). All registers respond as 32-bits wide (DSACK1 and DSACK0) except PORT1B which responds as 16-bits wide (DSACK0) and ACR which is in the RAMSEY gate array and requires external cycle termination performed by RAMSEY.

*This Address Counter Register is in the RAMSEY gate array. Please see the RAMSEY chip specification for more information.

2.17 SDMAC (Continued)

REGISTER AND COMMAND DESCRIPTIONS

DAWR (Address: \$00) — write only

The DATA ACKNOWLEDGE WIDTH REGISTER is used to determine the pulse width of SDACK and XDACK. SDACK and XDACK are the SCSI and 'XT' peripheral handshaking signals respectively. The pulse width of these signals is specified by the external device manufacturer and determines the data transfer rate to and from the SDMAC to the peripheral device. At reset DAWR is initialized to zero. This pulse width is affected by the system clock rate as detailed below.

Bit7 — x	
Bit2 — x	
Bit1 — DW1	
Bit0 — DW0	

NOTE: x denotes don't care.

SDACK (pin 66) or XDACK (controls pin 71, CSX1) width is determined as follows:

DW1	DW0	DACK WIDTH (number of system clocks)
0	0	1 SCLK period
0	1	2 SCLK period
1	0	3 SCLK period
1	1	4 SCLK period

NOTE: For A3000 machines, set DAWR to 3.

WTC (Address: \$04) — read/write

The WORD TRANSFER COUNTER provides a 24-bit counter addressed as a 32-bit longword for determining the number of data words (16 bits) to be transferred by the SDMAC. A minimum of one to a maximum of 16 million data words can be transferred with a single command. The counter must be initialized prior to beginning a transfer and will automatically decrement once per word transferred until it reaches the terminal count value of zero.

Bit31 — x	
Bit24 — x	
Bit23 — MSB	
Bit0 — LSB	

NOTE: x denotes don't care.

CNTR (Address: \$08) — read/write

The CONTROL REGISTER is an 8 bit register used to set mode and operating parameters of the SDMAC. An external reset will set all register bits to a low state.

Register bit descriptions:

Bit7 — x	
Bit6 — x	
Bit5 — TCEN	
Bit4 — PREST	
Bit3 — PDMD	
Bit2 — INTEN	
Bit1 — DDIR	
Bit0 — IO_DX	

TCEN — Terminal Count Enable, Active High

Activating the TCEN will enable terminal count hardware. If set high, upon reaching a terminal count of zero the SDMAC terminates operation, releases the bus if it has control, sets E_INT high in the INTERRUPT STATUS REGISTER, and forces CSX1 (pin 71) low if the CONTROL REGISTER bit PDMD is set low. If interrupts are enabled, INT_P will also be set high in the INTERRUPT STATUS REGISTER, and the external host interrupt pin, INT_2, will go to an active low state.

2.17 SDMAC (Continued)

REGISTER AND COMMAND DESCRIPTIONS (Continued)

PREST — Peripheral Reset, Active High

PREST is intended to be used to control a reset signal for external peripheral devices. Setting PREST high will cause the SDMAC output pins, _IOW and _IOR to go low. An active low external reset for peripheral devices can then be derived from _IOW and _IOR as follows:

(The external reset can thus be implemented with one 'AND' gate)

External Peripheral Reset = _IOW & _IOR

PDMD — Peripheral Device Mode Select

The SDMAC is capable of controlling two separate types of peripheral interfaces. These two interfaces are mutually exclusive, only one interface can be active at any given time. When PDMD is set high the interface mode is intended for SCSI controller type devices. When set low the interface mode is intended for direct connect to IBM 'AT/XT' type devices. Thus for PDMD, 1=SCSI, 0=XT/AT.

INTEN — Interrupt Enable, Active High

Activating the INTEN will enable the SDMAC to generate external interrupts. If set high, any internally generated interrupt or an external interrupt on INTA (pin 73) or INTB (pin 34), will set INT_P high in the INTERRUPT STATUS REGISTER and will force the external host interrupt, INT_2 (pin 35), to an active low state.

DDIR — Device Direction

DDIR is used to define the direction of data transfers to and from peripheral devices. Direction is determined as follows:

DDIR	DIRECTION
1	Read from host, write to peripheral.
0	Write to host, read from peripheral.

IO_DX — IORDY and CSX1 Polarity Select

The IO_DX bit is used to define the polarity of the SDMAC pins CSX1 (pin 71) and IORDY (pin 72). If set high, the CSX1 output and IORDY input pin are inverted from their default polarities imposed when the SDMAC is reset.

ACR (Address: \$0C) — read/write

The ADDRESS COUNT REGISTER provides a 32 bit address counter for determining the starting address of a DMA transfer and is written as a single 32-bit longword. The counter must be initialized prior to the transfer of data and is incremented by 4 if the SDMAC is transferring to/from a 32-bit memory area or by 2 if the SDMAC is transferring data to/from a 16 bit memory area. This continues until the terminal count (enabled in the CNTR register) or an external END-OF-PROCESS is reached (INTA active). This register is actually in the RAMSEY gate array and thus must be used with the RAMSEY chip or a device that emulates the Address Counter function.

NOTE: The counter can only be preset to a longword aligned boundary (address bits a1 and a0 are always written as 0,0), all other types of non-aligned transfers MUST be done as programmed I/O.

Bit31 — MSB
 |
 Bit0 — LSB

ST_DMA (Address: \$10) — read/write strobe

Any write/read to the START DMA location will cause the SDMAC to begin execution. Execution will continue until either a terminal count is reached or an external EOP is generated, as well as, an error condition occurs, or a SP_DMA command.

FLUSH (Address: \$14) — read/write strobe

A write/read to this location will cause the SDMAC to flush what remains in the internal 4 longword FIFO onto the host bus. The use of FLUSH is needed whenever the SDMAC is not using the Terminal Count mode of DMA transfers which is enabled by the TCEN bit of the control register (CNTR). If the Terminal Count mode of DMA transfer is not enabled by the TCEN bit then the SDMAC can be free-running (started with ST_DMA) and will continue until no more data is presented/requested across the SCSI bus. After an external EOP is generated, a FLUSH command followed by a SP_DMA command terminates the transfer. Note that this location should not be strobed if no DMA transfer was started. FLUSH need only be used when reading from the peripheral device.

2.17 SDMAC (Continued)

REGISTER AND COMMAND DESCRIPTIONS (Continued)

CINT (Address: \$18) — read/write strobe

Any write/read to the CLEAR INTERRUPT location will clear all internally or externally generated interrupts and the system interrupt signal, INT_2, if it was generated internally. CLR_INT has no effect on externally generated interrupts connected to the SDMAC INT2 pin.

ISTR (Address: \$1C) — read only

The Interrupt Status Register is a 9 bit register used to inform the host of interrupt activity and of internal static conditions. All internally generated interrupts are set in-active by a hardware reset.

Bit8	— INTX
Bit7	— INT_F
Bit6	— INTS
Bit5	— E_INT
Bit4	— INT_P
Bit3	— UE_INT
Bit2	— OE_INT
Bit1	— FF_FLG
Bit0	— FE_FLG

INTX — XT/AT Interrupt pending, Active High

This bit is set high whenever the INTB pin (pin 34) is forced low by an XT/AT peripheral device. If the Interrupt Enable Bit is set high (interrupts enabled) in the CNTR register, INTX will also activate INT_P and cause the external interrupt signal, INT_2, to go to an active low state on the host bus.

INT_F — Interrupt Follow, Active High

INT_F is used to indicate a pending interrupt condition from the SDMAC or external peripheral device. It is activated by any of the following interrupt signals: INTX, INTS, E_INT, UE_INT, or OE_INT. INT_F is not effected by the Interrupt Enable Bit in the CNTR register.

INTS — SCSI Peripheral Interrupt, Active High

INTS is used to indicate that an external SCSI peripheral device connected to the INTA pin (pin 73) is attempting to interrupt the host. An active INTS will activate INT_F. If the Interrupt Enable Bit is set high (interrupts enabled) in the CNTR register, INTS will also activate INT_P and cause the external interrupt signal, INT_2, to go to an active low state on the host bus.

E_INT — End-Of-Process Interrupt, Active High

E_INT is used to indicate that either the SDMAC has reached its Terminal Count or a FLUSH command has completed. An active E_INT will activate INT_F. If the Interrupt Enable Bit is set high, E_INT will also activate INT_P and cause the external interrupt signal, INT_2, to go to an active low state.

INT_P — Interrupt Pending, Active High

INT_P is used to indicate a pending interrupt condition from the SDMAC or external peripheral device only when interrupts are enabled. It is activated by any of the following interrupt signals: INTX, INTS, E_INT, UE_INT, or OE_INT. INT_P is only active when the Interrupt Enable Bit is set high.

UE_INT — Under-Run FIFO Error Interrupt, Active High

UE_INT is used to indicate that the SDMAC internal FIFO has an under-run error. An active UE_INT will activate INT_F. If the Interrupt Enable Bit is set high, UE_INT will also activate INT_P and cause the external interrupt signal, INT_2, to go to an active low state.

OE_INT — Over-Run FIFO Error Interrupt, Active High

OE_INT is used to indicate that the SDMAC internal FIFO has an over-run error. An active OE_INT will activate INT_F. If the Interrupt Enable Bit is set high, OE_INT will also activate INT_P and cause the external interrupt signal, INT_2, to go to an active low state.

FF_FLG — FIFO Full Flag, Active High

FF_FLG is used to indicate the status of the SDMAC internal FIFO. Whenever the internal FIFO reaches its full longword count of 4, FF_FLG will go to an active high state and remain there until at least one longword (32-bits) is removed from the FIFO.

2.17 SDMAC (Continued)

REGISTER AND COMMAND DESCRIPTIONS (Continued)

FE_FLG — FIFO Empty Flag, Active High

FE_FLG is used to indicate the status of the SDMAC internal FIFO. Whenever the internal FIFO reaches a longword count of 0, the FE_FLG will go to an active state and remain there until at least one word is entered into the FIFO.

RESERVED (Address: \$20,24,28,2C,30,34,38)

These longword addresses are reserved for future expansion.

SP_DMA (Address: \$3C) — read/write strobe

Any write/read to the STOP DMA location will cause the SDMAC to abort execution. Register contents are unaffected by this operation and no interrupts will be generated.

PORT0 (Address range: \$40 - \$4C) — read/write

PORT0 address range is an internally decoded address range for selecting an external SCSI controller device. Addressing the SDMAC within this range will cause the SDMAC to generate the appropriate interface signals (i.e. chip select, read, write, etc.) to allow communication with external device. I/O definitions within this address range is application dependent and depends on how the user has connected the SCSI controller device to the host bus address logic for register selection of internal SCSI controller registers. In the A3000 machine the programmer should access (read/write) the SCMD register as a byte at \$43 and access (read/write) the SASR register as a longword at \$40 (data in D7-D0). The SDMAC will also allow the programmer to read the SASR register as a byte at \$41 (with data on both D23-D16 and D7-D0) due to the internal and external connections in the SDMAC and the A3000.

PORT1A (Address range: \$50 - \$5C) — read/write

PORT1A address range is an internally decoded address range for selecting one of two external 8-bit 'XT' compatible devices. Addressing the SDMAC within this range will cause the SDMAC to generate the appropriate interface signals (i.e. chip select, read, write, etc.) to allow communication with an external device. I/O definitions within this address range is application dependent.

PORT2 (Address range: \$60 - \$6C) — read/write

PORT2 address range is an internally decoded address range for selecting a second external 8-bit 'XT' compatible device. Addressing the SDMAC within this range will cause the SDMAC to generate the appropriate interface signals (i.e. chip select, read, write, etc.) to allow communication with an external device. I/O definitions within this address range is application dependent. To support a second 'XT' device, the IO_DX bit in the CNTR register must be set to a low state.

PORT1B (Address range: \$70 - \$7C) — read/write

PORT1B address range is an internally decoded address range for selecting an external 16-bit 'AT' compatible device. Addressing the SDMAC within this range will cause the SDMAC to generate the appropriate interface signals (i.e. chip select, read, write, etc.) to allow communication with an external device. I/O definitions within this address range is application dependent. The IO_DX bit in the CNTR register should be set to a high state.

SDMAC TIMING DATA

This device is intended to run synchronously with the 68030 CPU clock. In the A3000 computer the device runs at either 16 MHz or 25 MHz. The maximum clock rate for SCLK is 25 MHz. The SDMAC runs asynchronously with the SCSI peripheral device (in A3000 the WD33C93A runs at 14.3 MHz).

2.18.1 RAMSEY

Ramsey's main function is that of a dynamic RAM controller, allowing the 68030 to interface with up to 16 megabytes of system DRAM. Ramsey also contains logic associated with the DMA controller (super DMAC).

Some of the key features of RAMSEY are:

- 4 megabytes of memory using 32 standard 256kx4 DRAMs (80 nsec).
- 16 megabytes of memory using 32 standard 1Mx4 DRAMs (80 nsec).
- Support of 68030 Burst mode (requires 80 nsec static column DRAMs).
- Page Mode RAM access (requires 80 nsec static column DRAMs).
- Automatic CAS-before-RAS refreshing of DRAMs.
- Multiplexing of addresses.

CONTROL REGISTER

There is a single 8 bit register internal to RAMSEY which can be used to change its mode of operations. This register is readable and writable. It is located at \$00DE0003 of the supervisor data space. Data written into the register does not take effect until the next refresh occurs. Consequently, if you write a value to the register you will have to wait out the refresh interval before the value can be read back. Each of these bits has a default value that it is set to when the *PWRUP bit is low.

bit 0	PAGE MODE	When high, PAGE mode is enabled (default = 0).
bit 1	BURST MODE	When high, RAMSEY will respond to the *CBREQ input and do burst cycles (default = 0).
bit 2	WRAP	If high, all 4 longwords of a burst will be allowed to take place. If WRAP is disabled, then the burst will only continue while A3,A2 are increasing - bursts will not be allowed to wrap to A3,A2 = 00 (default = 0).
bit 3	RAMSIZE	If low, then RAM is 1 megabit (256kx4 or 1Mx1). If high, then RAM is 4 megabit (1Mx4). The default value is determined by the RSIZE input.
bit 4	RAMWIDTH	If low, then RAM is 1 bit wide (1Mx1). If high, then RAM is 4 bits wide (256kx4 or 1Mx4) (default = 1).
bit 5,6	REFRESH RATE	The refresh counter uses the CPUCLK to count out refresh times. The number of clocks between refreshes is determined by the following table:

Bit 6,5	#of clocks	refresh interval (usecs)	
		16 MHz	25 MHz
00	154	9.24	6.16
01	238	14.28	9.52
10	380	22.8	15.2
11	oo	(refresh turned off)	

Since 512 refreshes must be done in 8 msec, the interval between refreshes must be less than 15.625 usecs. During page mode, RAS can only be low for 10 usecs at a time, so the refresh rate should be set to less than 10 usecs when page mode is enabled. The default values are determined by the RSPEED input. If RSPEED is low then the default value for bits 6,5 = 00. If RSPEED is high then the default for bits 6,5 = 01.

bit 7	TEST	This bit is used for testing only.
-------	------	------------------------------------

VERSION REGISTER

There is a read only register at \$00DE0043 that contains the version number of the RAMSEY chip. The 12D RAMSEY returns a version number of \$0D.

2.18.1 RAMSEY (Continued)

RAM MEMORY MAP

\$07C00000-	\$07CFFFFF	RAS0
\$07D00000-	\$07DFFFFF	RAS1
\$07E00000-	\$07EFFFFF	RAS2
\$07F00000-	\$07FFFFFF	RAS3
RSIZE = 0, RAMWIDTH = 1		
\$07000000-	\$073FFFFF	RAS0
\$07400000-	\$077FFFFF	RAS1
\$07800000-	\$07BFFFFF	RAS2
\$07C00000-	\$07FFFFFF	RAS3
RSIZE = 1, RAMWIDTH = 1		
\$07C00000-	\$07FFFFFF	RAS0
RSIZE = 0, RAMWIDTH = 0		

RAM CONTROLLER DESCRIPTION

RAMSEY is designed to work at 2 system speeds — 16.67 MHz and 25 MHz. The RSPEED input is used to tell RAMSEY which speed the system is.

In an effort to improve overall system performance, RAMSEY has several different modes of operation. The two basic modes of operation are referred to as *standard* and *page*. Both standard and page modes can be run with or without *burst* mode, and burst mode can be done with or without *burst wrap* enabled.

Standard mode

In this type of operation both page mode and burst mode are disabled. This mode requires standard 80 nsec page mode DRAMs. Access to the RAMs always takes 5 cycles at 25 MHz, and 4 cycles at 16 MHz.

Page mode

This mode requires 80 nsec static column mode DRAMs. When the RAM is accessed, RAS is held low after the cycle completes. This leaves the current RAM ‘page’ open. While RAS is low, the RAM behaves like a static RAM. Any data within the current page can be accessed by changing the column addresses only. Since the column access time (tAA) is much less than the RAS access time (tRAC), subsequent access to this page of data can be done faster. As long as RAS is held low (10 usecs max) RAMSEY will allow the CPU to access RAM on this page in only 3 cycles (16 and 25 MHz). Comparators inside RAMSEY monitor the ROW address of subsequent accesses. If the ROW address matches (page hit), the RAM can be read in 3 cycles. If the comparators detect that the data being requested is on a different page (page miss), then RAS must be cycled high and low again, opening up a new page in RAM. Since RAS must be cycled high when a page miss occurs, RAM access takes longer (7 cycles at 25 MHz, 6 at 16 MHz).

There is some difference in how page mode is done at 16 MHz versus 25 MHz. At 25 MHz, the page comparator only detects page misses when *AS is low, and RAM is being accessed. Therefore, when a page is opened (RAS held low), it will remain open until the next refresh occurs, or a page miss is detected. At 16 MHz, however, the page comparator will detect a page miss while *AS is high. Therefore, at 16 MHz a page will only stay open as long as consecutive bus cycles access RAM in this page (or a refresh occurs). At 25 MHz the page will remain open even if bus cycles in between accesses to the currently open page occur (such as chip memory, CIA’s, etc.). This was done so that page misses at 16 MHz will only take 5 cycles — addresses are valid 1/2 cycle before *AS (30 nsecs). If it waited until *AS was valid before detecting a page miss, the RAS precharge requirement (tRP) could not be met in 5 cycles.

Burst mode

This mode requires 80 nsec static column DRAMs. In this mode, RAMSEY will respond to the *CBREQ input from the 68030 and allow burst access to RAM. Burst cycles take 2 clocks each.

Burst wrapping

The WRAP bit in the RAMSEY control register is associated with burst mode. The 68030 will always request 4 longword values during a burst sequence. However, if the initial longword is not aligned on a quad longword (A3,A2 not equal to 0,0), the 68030 will read in data which is behind the first data it asked for. Since it is less likely that this data will be used, the WRAP control bit allows you to prevent the 68030 from doing this. If WRAP is low, then the burst will stop after the data with A3,A2 = 1,1 is accessed. If WRAP is high, then all 4 longwords will be bursted.

2.18.1 RAMSEY (Continued)

DMAC SUPPORT

RAMSEY contains the address counters used during DMA via the onboard controller. When *DMAEN becomes low, the address lines become outputs and provide the DMA addresses. The DMA address is incremented on the rising edge of *AS whenever *DMAEN is low. Since DMA to both 32 and 16 bit ports are supported, RAMSEY must monitor how the cycle was terminated so that it can increment the address counter by the appropriate amount. If *STERM transitioned low sometime during the cycle, then the port was 32 bits wide, and the address is incremented by 4. If *DSACK0 transitions during the DMA bus cycle, then the port was also 32 bits wide (*DSACK0 and *DSACK1 are both set low to terminate an asynchronous 32 bit transfer). If neither signal is seen to transition, then it can be assumed that the cycle was terminated by *DSACK1, indicating that the port was 16 bits wide, and the address is incremented by 2.

The address counters are preset before DMA is done by writing to the 32 bit register at location \$00DD000C (this register is readable as well). The counter can only be preset to a longword aligned boundary (bits 1 and 0 are always written as 0,0).

Using conventional 256kx4 80 ns DRAM, the Ramsey gate array will support up to 4 Mbytes of DRAM operating on a conventional memory cycle basis. This controller also supports the use of static column DRAMs, and in this case, the 68030 Burst modes are directly supported.

Ramsey does not support both static column and conventional DRAM simultaneously. It is expected that the user who wishes to use static column DRAM, will first move the megabyte of fast RAM (that came with the A3000) over to extend the chip RAM.

DANGER: The on board memory starts at \$07FFFFFF, and grows downward to \$07F00000 in the case of a single megabyte, or \$07C00000 in the case of 4 Megabytes. This is clearly beyond the 24 bit address space employed on the A500 and A2000 computers. We, at Commodore, sincerely hope you heeded our repeated warnings about not using the upper eight bits of addresses for anything other than addressing.

2.18.2 FAST MEMORY

FAST Memory is non-CHIP memory that resides on the local 32-bit 68030 processor bus. This memory is generally used for program code execution. The FAST memory is tightly coupled to the 68030 via custom circuitry that has been designed to provide very efficient operation. This allows programs to execute very quickly from this RAM.

The A3000 is shipped with a total of 1 megabyte of FAST memory (also known as FAST RAM). The organization of the RAM chips is 256K x 4. A total of 32 RAM chips can be added to the FAST memory section. Using 256K x 4 DRAMs, this results in a total of 4 megabytes of FAST memory. The A3000 also supports the use of DRAMs organized as 1 megabit x 4 (1M x 4). If all 32 of the RAM locations are populated with this size of DRAM, the total amount of FAST memory is 16 megabytes.

WARNING: 256K x 4 and 1M x 4 DRAMs CANNOT be intermixed in the FAST RAM section! If the first bank has 256K x 4 DRAMs, then subsequent banks must have 256K x 4 DRAMs as well. If the first bank has 1M x 4 DRAMs, then the other banks also must have 1M x 4 DRAMs.

There are four *banks* of FAST memory that can be filled. Each bank consists of 8 RAM chips. Using 256K x 4 DRAMs, each bank adds 1 megabyte. With 1M x 4 DRAMs, each bank is 4 megabytes in size. Bank 0 consists of U850-U857, bank 1 U858-U865, bank 2 U866-U873 and bank 3 U874-U881. FAST memory MUST begin in bank 0, and is added in order to banks 1 through 3.

The FAST RAM section consists of 8, 20 pin DIP sockets (U850D-U857D), as well as 32 20 pin ZIP sockets (U850-U881). A maximum of 32 of these locations may be populated with RAM chips. The 8 DIP locations and the 8 lowest numbered ZIP locations are electrically equivalent (both are bank 0). RAM chips can be installed in either the DIP section OR the first bank in the ZIP section, but not both. ZIP sockets have been used in anticipation of the form factor of 1M x 4 RAM chips. The 8 DIP locations that overlap the first bank of ZIPs allow the machine to be shipped with DIP DRAMs. This gives the user the flexibility to use these 8 DRAMs in the CHIP memory if desired.

In the FAST RAM section, the use of static column mode DRAMs allows for slightly improved system performance. It is therefore recommended that the user add this type of DRAM when upgrading the system. In order to use the modes that take advantage of static column mode DRAMs, ALL of the FAST RAM must be static column mode DRAMs. If the FAST RAM section has only page mode DRAMs, or page mode DRAMs and static column mode DRAMs mixed together, then you cannot use these new modes.

There is one jumper (J852) that is associated with the FAST RAM. When using 256K x 4 DRAMs, the shorting bar should connect pins 2 and 3. If the FAST RAM section contains only 1M x 4 DRAMs, then pins 1 and 2 should be connected. If the new RAM is functioning properly it will automatically be recognized by the operating system when the A3000 is powered up. The title bar of the Workbench screen (which shows how much **other** memory is available to the system) should indicate the addition of the new memory. If this number does not reflect the amount of RAM that was added, then it is not working properly. Check that the RAM chips were inserted properly, and check for any bent pins.

The following is a list of required parameters for the new FAST RAM:

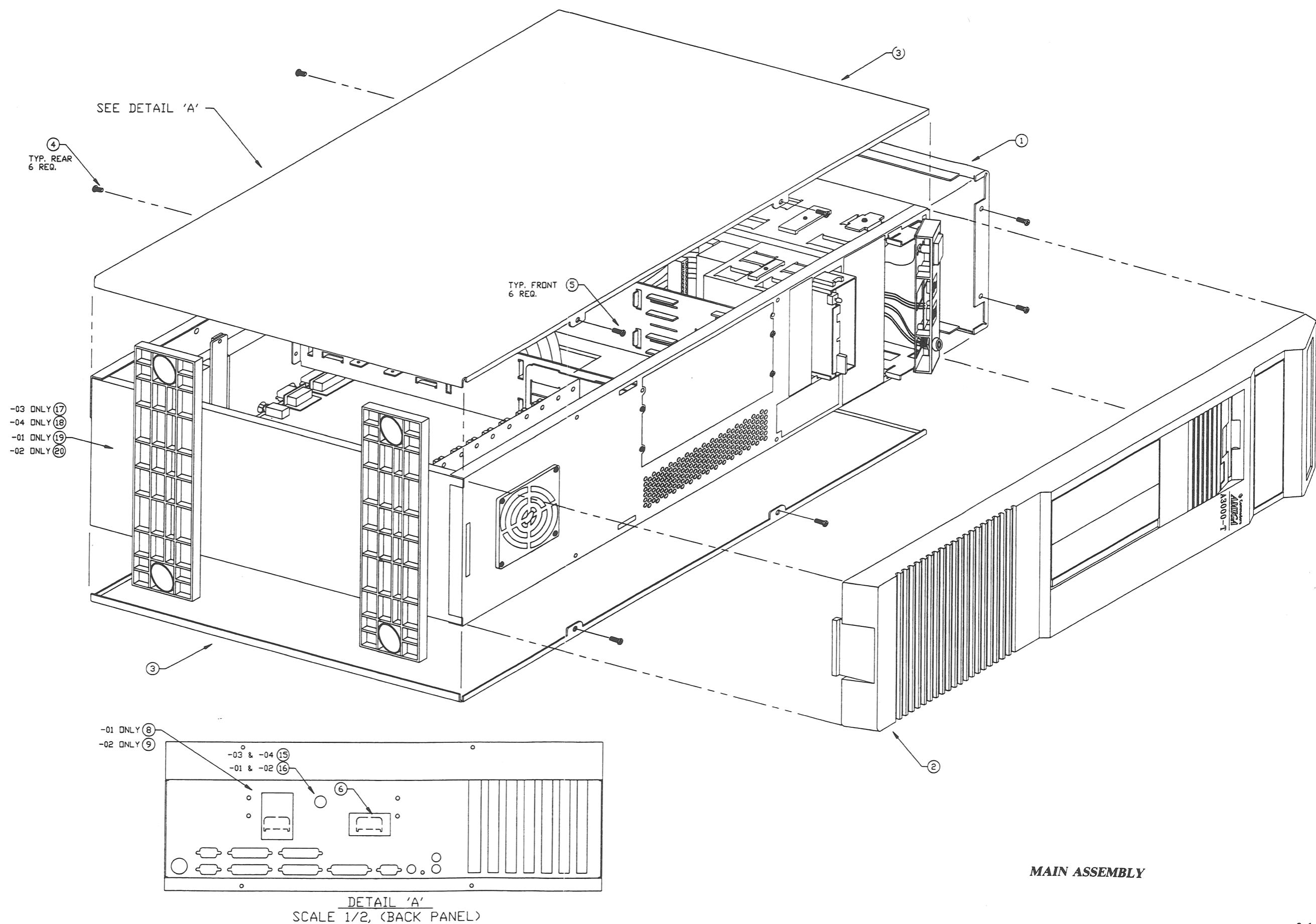
Organization:	256 kilobits by 4 bits (256K x 4) <i>or</i> 1 megabit by 1 bit (1M x 4)
Speed:	80 nanoseconds or less
Type:	page or static column mode
Package:	20 pin DIP (1st bank of 8 chips only); 20 pin ZIP

The following are examples of some of the DRAMs that are acceptable:

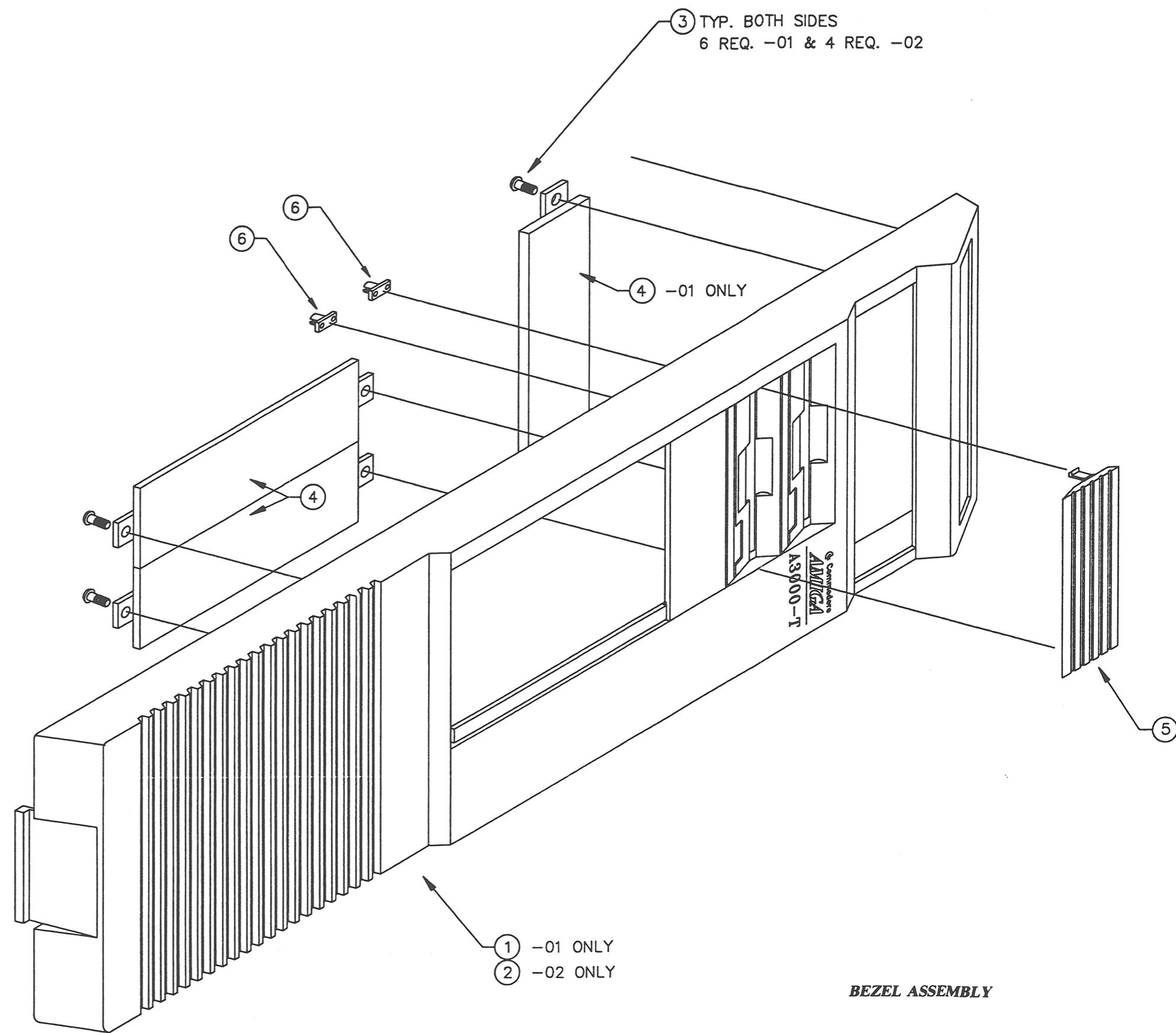
256K x 4 Page Mode		256K x 4 Static Column Mode	
TOSHIBA	TC514256	TOSHIBA	TC514258
OKI	MSM514256	SHARP	LH64258
HITACHI	HM514256	OKI	MSM514258
NEC	uPD424256	HITACHI	HM514258
		NEC	uPD424258
		TI	TMS44C256

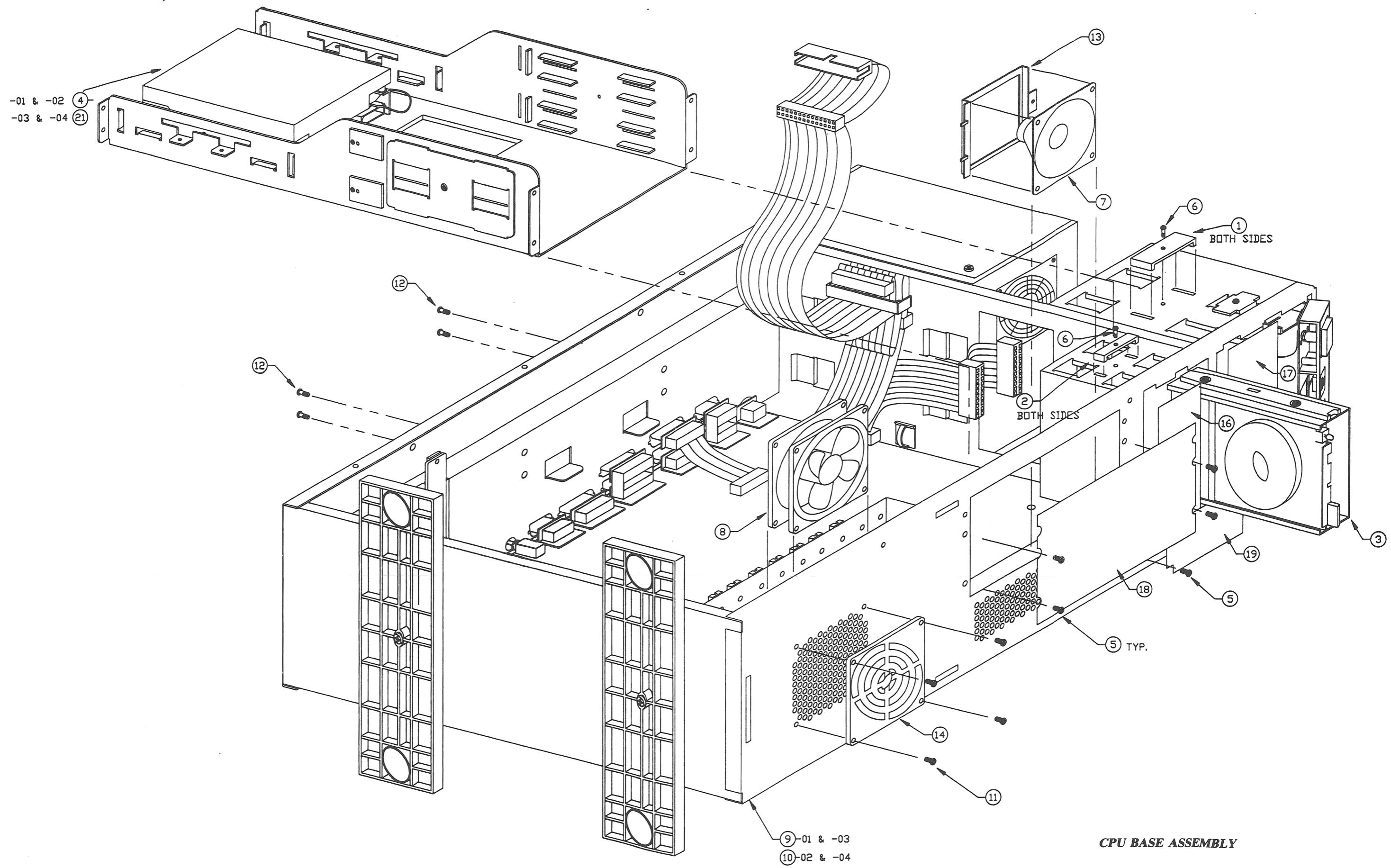
1M x 4 Page Mode		1M x 4 Static Column Mode	
TOSHIBA	TC514400	TOSHIBA	TC514402
HITACHI	HM514400	OKI	MSM514402
NEC	uPD424400		
OKI	MSM514400		

SECTION 3
TROUBLESHOOTING

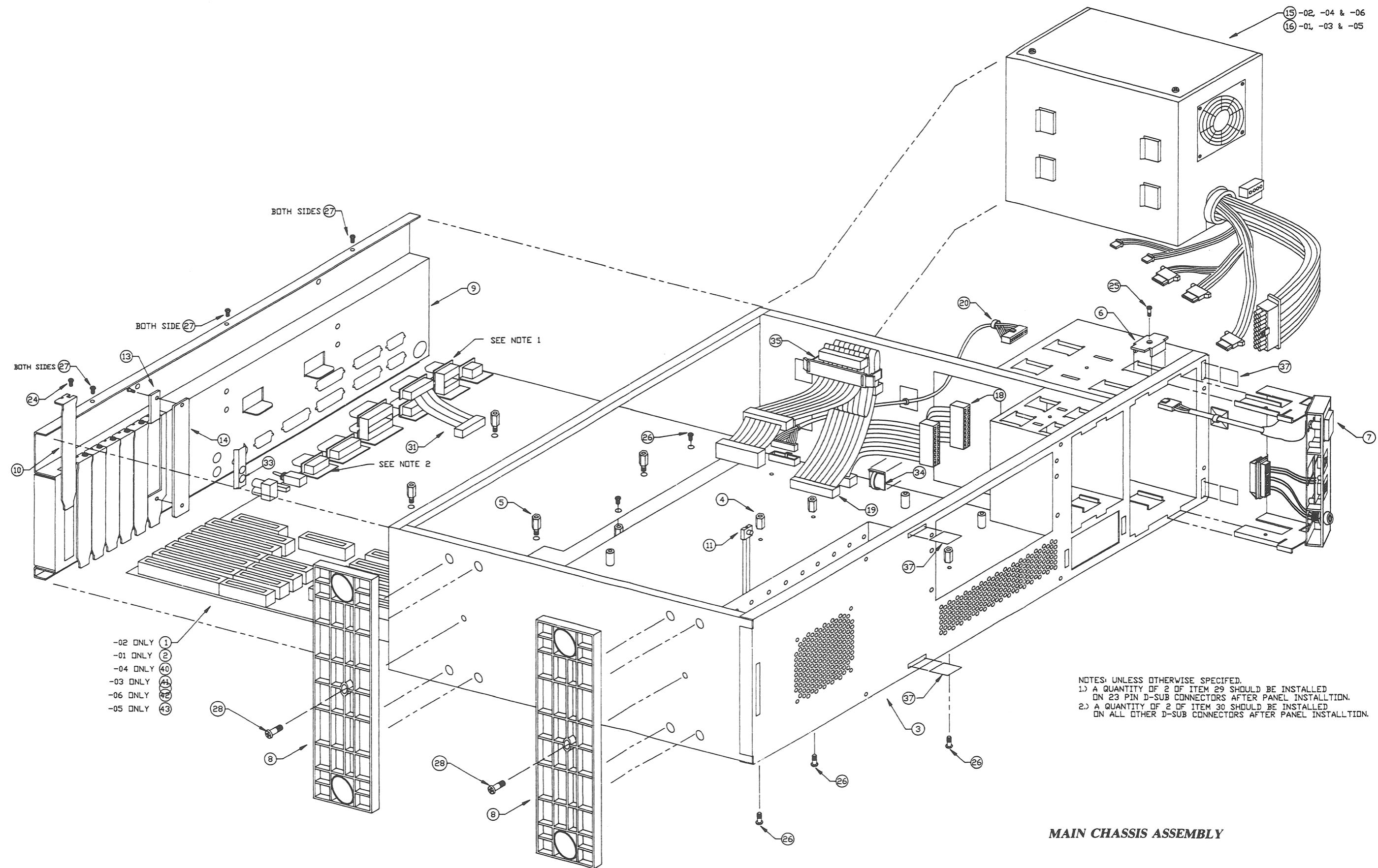


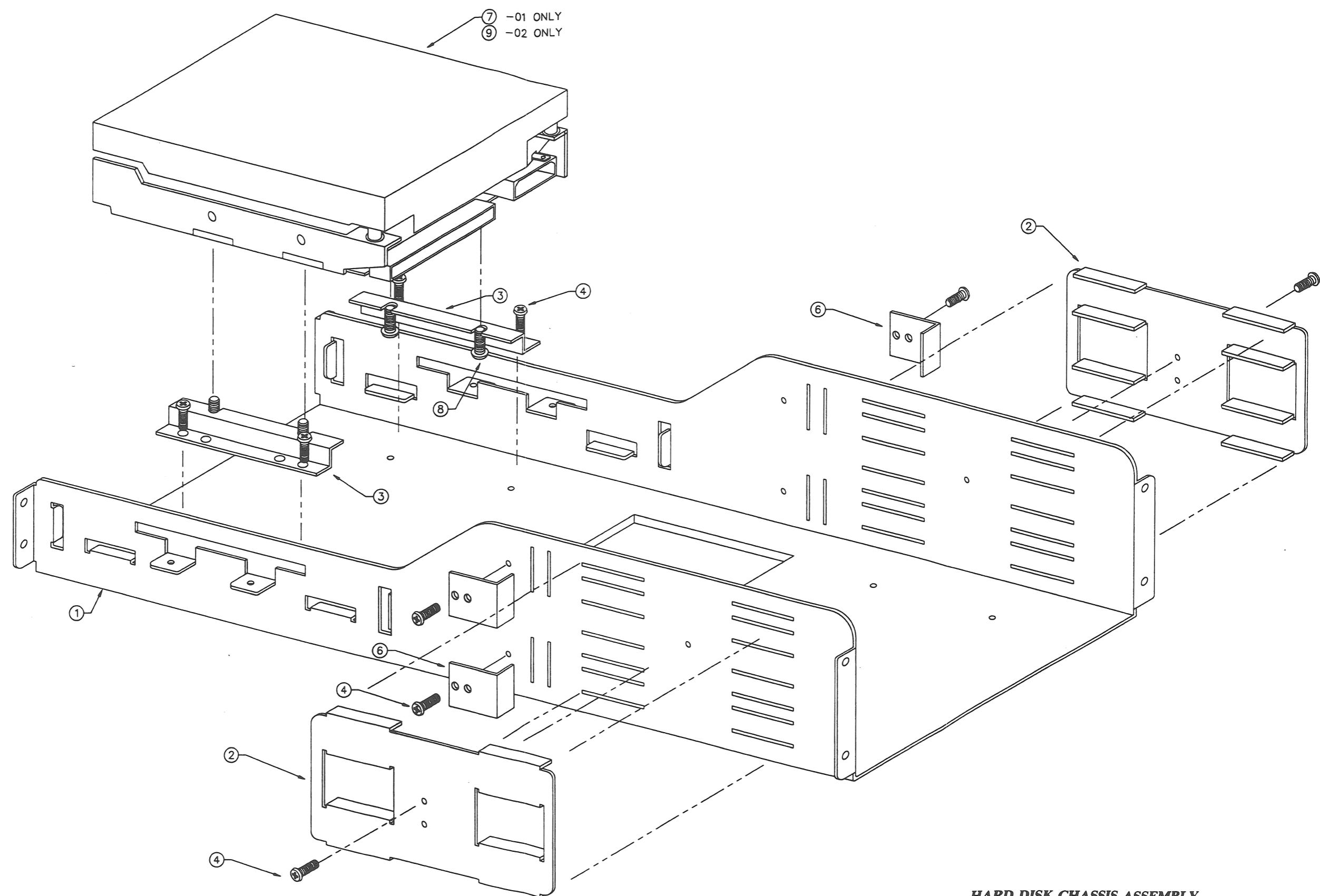
MAIN ASSEMBLY



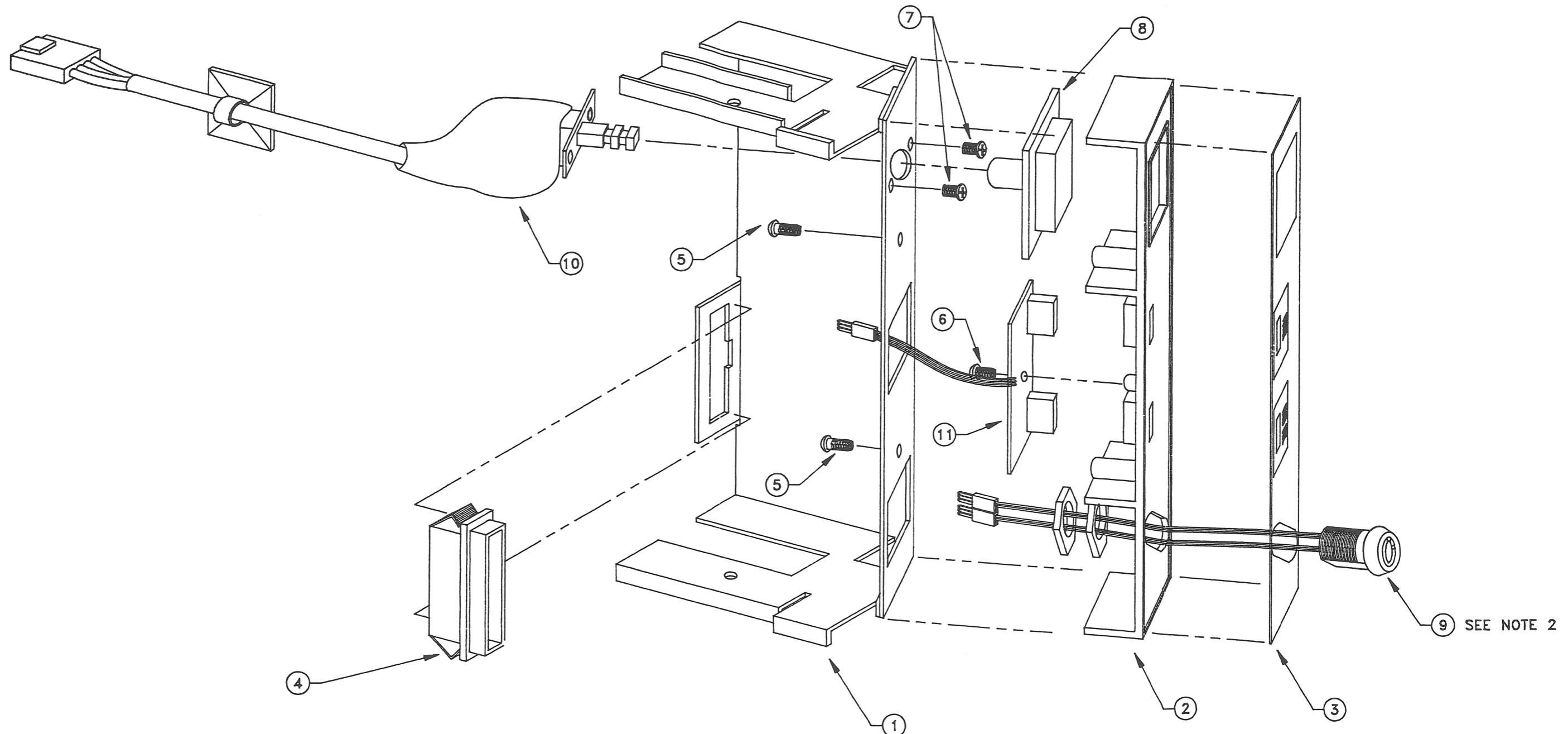


CPU BASE ASSEMBLY





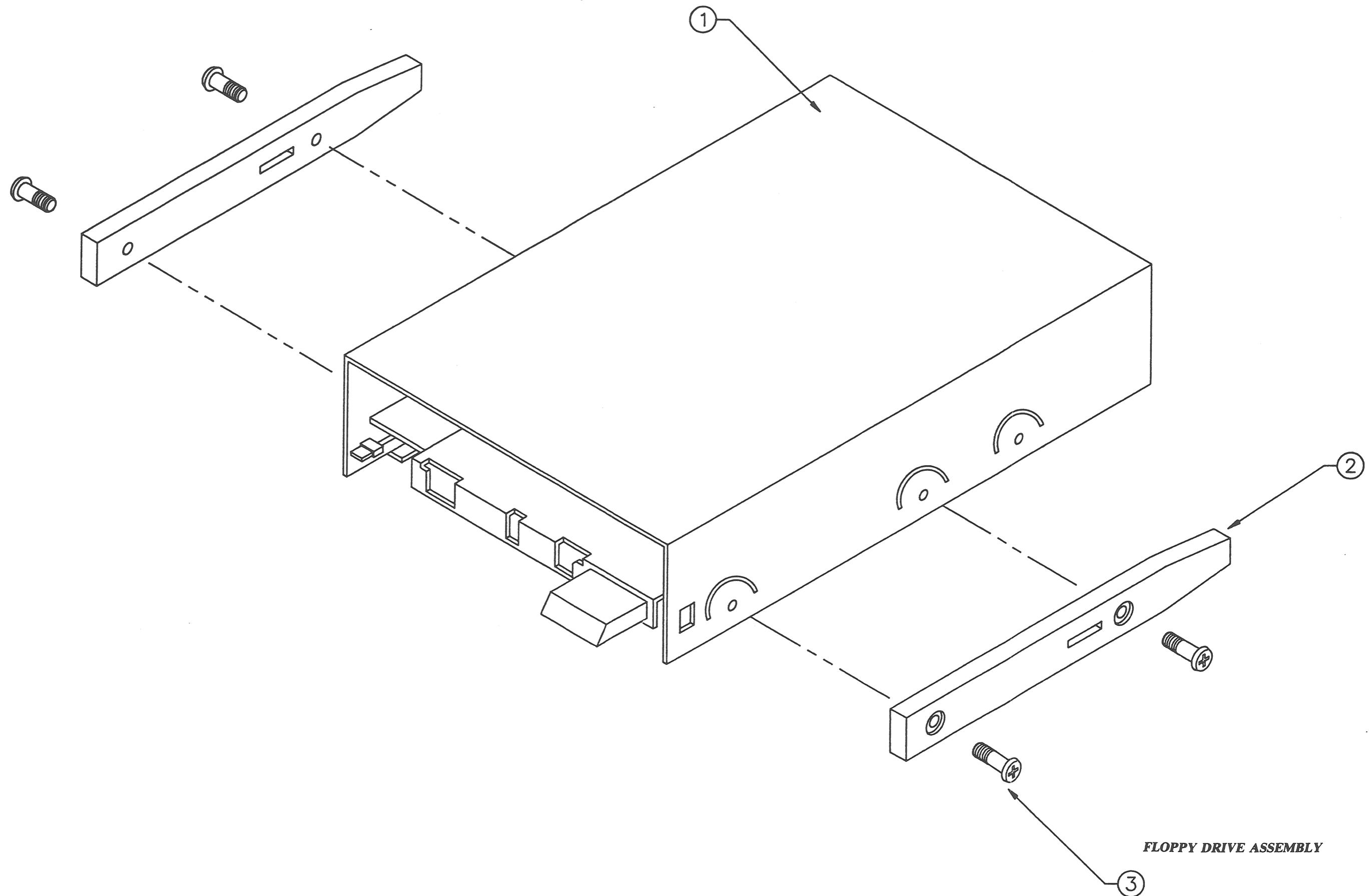
HARD DISK CHASSIS ASSEMBLY



NOTES: UNLESS OTHERWISE SPECIFIED.

- 1.) ITEM 9 COMES WITH TWO RETENTION NUTS, ONE RETAINS THE SWITCH IN ITEM 2 AND THE OTHER GROUNDS IT TO ITEM 1.

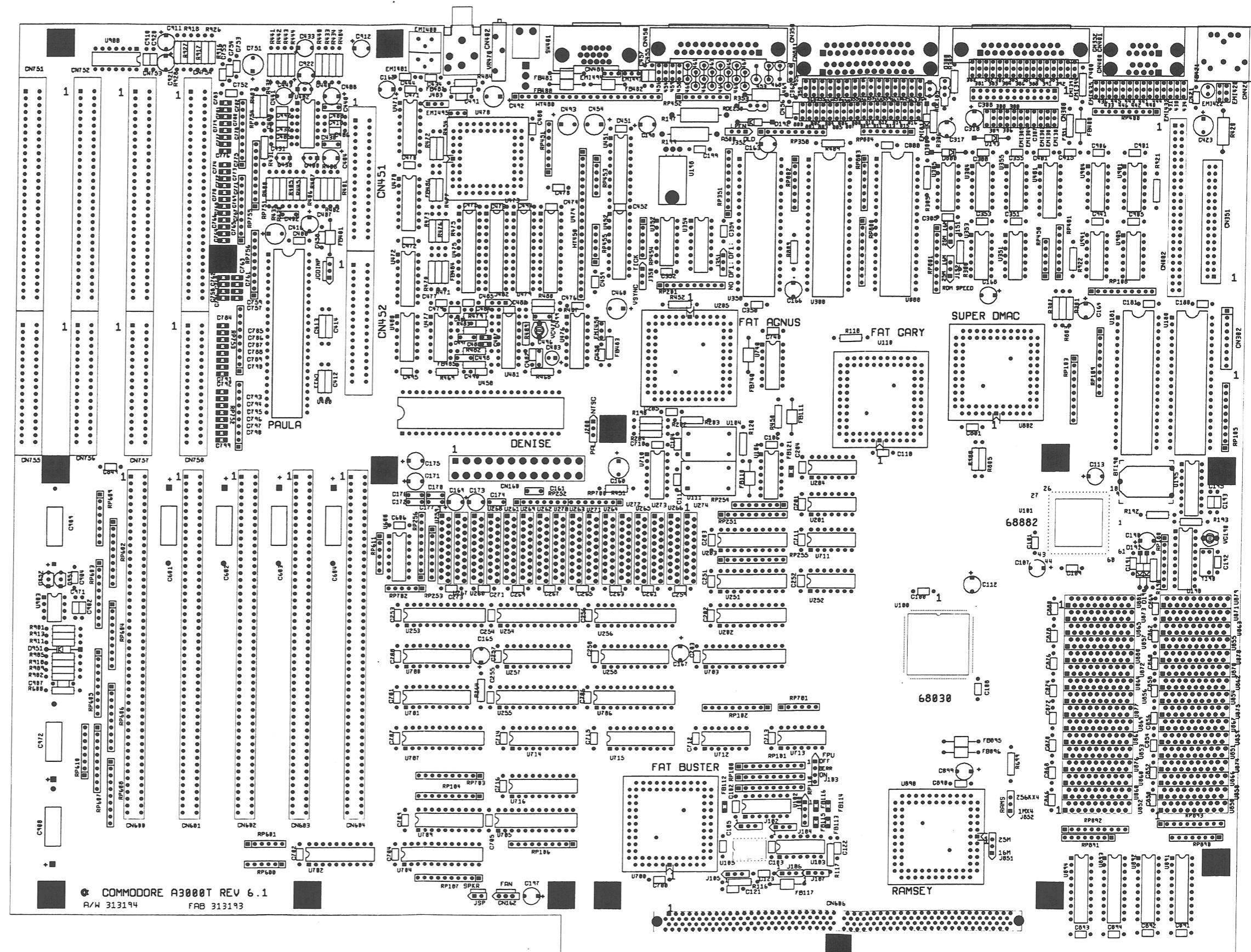
POWER / KEY BRACKET ASSEMBLY



SECTION 4

PARTS

**P
A
R
T
S**



Commodore International Spare Parts List SHIPPING ASSEMBLIES

Commodore part numbers are provided for reference only and do not indicate the availability of spare parts from Commodore. Industry standard parts (Resistors, Capacitors, Connectors) should be secured locally. Part number information may vary according to country, some parts may not be available in all countries. Part Numbers are subject to change, see Parts (Section 2) of current Techtopics for current numbers.

SHIP ASSY SINGLE BOX A3000T		KEYBOARD ASSY A3000T (Continued)
533500-01	A3000T US 4M 3F880K 3H105M	312715-07 KEYBOARD PCB ASSY SWISS
533500-02	A3000T CN 4M 3F880K 3H105M	312715-08 KEYBOARD PCB ASSY DENMARK
533500-03	A3000T US 4M 3F880K 3H210M	312715-09 KEYBOARD PCB ASSY NORWAY
533500-04	A3000T CN 4M 3F880K 3H210M	312715-10 KEYBOARD PCB ASSY FINLAND/SWEDEN
363380-01	BOX SINGLE PACKING	312715-11 KEYBOARD PCB ASSY ICELANDIC
320408-01	BAG FLAT 812 MM X 1066 MM	380130-01 KEYBOARD FOOT ADJUSTABLE (QTY 2)
390926-01	BEZEL CORNER SUPPORT	580131-01 SPRING PLATE (QTY 2)
318170-01	CARD READ THIS FIRST	907272-06 WASHER FLAT 4.3MM (QTY 2)
366064-01	LABEL UPC	906883-04 SCREW PAN HEAD S.T. 2.9 X 9.5 MM (QTY 6)
318240-01	SPACER STYROFOAM REAR PANEL	906805-03 SCREW FLAT HEAD S.T. 2.9 X 9.5 MM (QTY 2)
313254-02	MOUSE ASSY 2 BUTTON, 2.5 METER CABLE	380129-01 FOOT PLASTIC (QTY 5)
903508-17	CORD POWER UL, CSA	380018-02 KEYBOARD CABLE ASSY (PAL)
313180-02	MAIN ASSY A3000T NTSC 105M HD 4M	380018-08 KEYBOARD CABLE ASSY (U.S.)
313180-04	MAIN ASSY A3000T NTSC 210M HD 4M	
313166-02	KEYBOARD A3000T U.S.	
251006-02	BAG FLAT 580 MM X 350 MM	
318243-01	KEYBOARD END CAP	
318893-01	DRYING AGENT (USE W/251006-02)	
324257-01	DRYING AGENT (USE W/320408-01)	
318317-01	KB/SW PARTITION & LIFTING STRAP	
315123-01	SOFTWARE SUB ASSY A3000T U.S.	
315123-02	SOFTWARE SUB ASSY A3000T CN	
318319-01	LINER BOX SUPPORT & FILLER TRAY	
324252-07	TAPE ADHESIVE TRANSP 50 MM	
251006-01	BAG PLASTIC (HOLDS KEYS & SCSI CONN)	
312975-01	ACCESSORY BAG ASSY	
315133-01	SOFTWARE SUB ASSY AS250 U.S.	
366182-01	LABEL CARTON AMIGAVISION	
318246-02	INSERT CHECKLIST	
318245-03	INSERT SYSTEM SOFTWARE INCLUDED	
368077-01	STICKER STARBURST	
368084-01	SEAL COLOR BOX TAMPER EVIDENT (ON TOP & BOTTOM FLAPS)	
363399-01	INTRO TO A3000T MANUAL	
390770-01	CONN ASSY SCSI TERMINATION	
366610-01	LABEL WARNING HEAVY LIFT	
368184-01	PAMPHLET INSTRUCTION SCSI TERMINATOR	
A3000T SHIP ASSY EUROPEAN		
533501-01	A3000T SHIP ASSY EU 4M 3F880K 3H105M	
533501-02	A3000T SHIP ASSY EU 4M 3F880K 3H210M	
363381-01	BOX PACKING	
320408-01	BAG FLAT 812 MM X 1066 MM	
390926-01	BEZEL CORNER SUPPORT	
318240-01	SPACER STYROFOAM	
313180-01	MAIN ASSY A3000T PAL 4M 105M HD	
366064-01	LABEL UPC	
324257-01	DRYING AGENT	
313180-03	MAIN ASSY A3000T PAL 4M 210M HD	
324252-07	TAPE ADHESIVE TRANSP 50MM	
312975-01	ACCESSORY BAG ASSY	
251006-01	BAG PLASTIC	
390770-01	SCSI TERMINATOR ASSY	
366610-01	LABEL WARNING HEAVY LIFT	
368184-01	PAMPHLET SCSI TERMINATOR INSTRUCTION	
KEYBOARD ASSY A3000T		
313166-01	KEYBOARD ASSY A3000T U.K.	
313166-02	KEYBOARD ASSY A3000T U.S.	
313166-03	KEYBOARD ASSY A3000T GERMANY	
313166-04	KEYBOARD ASSY A3000T ITALY	
313166-05	KEYBOARD ASSY A3000T FRANCE	
313166-06	KEYBOARD ASSY A3000T SPAIN	
313166-07	KEYBOARD ASSY A3000T SWISS	
313166-08	KEYBOARD ASSY A3000T DENMARK	
313166-09	KEYBOARD ASSY A3000T NORWAY	
313166-10	KEYBOARD ASSY A3000T SWEDEN/FINLAND	
313166-11	KEYBOARD ASSY A3000T ICELANDIC	
312458-01	KEYBOARD HOUSING TOP	
312715-01	KEYBOARD PCB ASSY U.K.	
312715-02	KEYBOARD PCB ASSY U.S.	
312715-03	KEYBOARD PCB ASSY GERMANY	
312715-04	KEYBOARD PCB ASSY ITALY	
312715-05	KEYBOARD PCB ASSY FRANCE	
312715-06	KEYBOARD PCB ASSY SPAIN	
KEYBOARD ASSY — PART OF 583501		
313166-01	U.K.	
313166-03	GERMANY	
313166-04	ITALY	
313166-05	FRANCE	
313166-06	SPAIN	
313166-07	DUTCH	
313166-08	DENMARK	
313166-09	NORWAY	
313166-10	SWEDEN/FINLAND	
LABEL CARTON — PART OF 583501		
316828-01	U.K.	
316831-01	GERMANY	
316830-01	ITALY	
316829-01	FRANCE	
316832-01	SPAIN	
316836-01	NETHERLANDS	

**Commodore International Spare Parts List
SHIPPING ASSEMBLIES (Continued)**

LABEL CARTON — PART OF 583501 (Continued)		DISK ASSY 3.5" WORKBENCH V2.0 — PART OF 583501 (Continued)	
316835-01	DENMARK	317248-01	SWEDISH
316838-01	NORWAY	317246-01	DANISH
316837-01	FINLAND	317247-01	NORWEGIAN
316839-01	SWEDEN	317245-01	SPANISH
251006-01	BAG FLAT 580 X 350 MM	317249-01	BELGIUM
380426-01	NEOPOLEN FOAM SET	DISK ASSY 3.5" AMIGA EXTRAS V2.0 — PART OF 583501	
380425-03	BOX SHIP CONTAINER	317235-02	U.S.
324252-07	ADHESIVE TAPE TRANSP 50MM	317234-01	INTER
313254-02	MOUSE ASSY 400 DPI 2 BUTTON 2.5M	317236-01	GERMAN
MANUAL USER A3000T — PART OF 583501		MANUAL WORKBENCH V2.0 — PART OF 583501	
363399-01	ENGLISH	363313-03	ENGLISH
363400-01	GERMAN	363314-01	FRENCH
363401-01	FRENCH	363315-01	ITALIAN
363402-01	ITALIAN	363316-01	GERMAN
363405-01	SWEDISH	363317-01	SPANISH
363403-01	SPANISH	CARD WARRANTY — PART OF 583501	
363406-01	DANISH	318365-03	1 YEAR U.S.
363404-01	NORWEGIAN	320046-06	GERMAN
363407-01	DUTCH	325249-01	U.K.
311883-01	DRYING AGENT (USE W/251006-01)	325254-01	FRENCH
368150-01	QUICK CONNECT A3000T	368014-01	AUSTRALIA
390543-01	TOOL, PLASTIC ADJUSTMENT	318882-01	CANADA
318344-02	BINDER WORKBENCH 2.0	318896-01	SOFTWARE LICENSE AGREEMENT
318343-01	SHEET LIFTERS	318708-02	PROGRAM LICENSE AGREEMENT GERMAN
318237-01	DISK HOLDER 3 RING VINYL	380720-01	CARD REMINDER DISK & PRINTER CAUTIONS
316834-01	CARTON LABEL CANADA	318556-02	CARD DISK EXCHANGE CANADA
316833-01	CARTON LABEL BELGIUM	903508-11	CORD POWER VDE MEDIUM BEIGE
315133-01	SOFTWARE SUB ASSY AS250 AMIGAVISION	903508-12	CORD POWER BSI MEDIUM BEIGE
368180-01	CARD BUSINESS REPLY (U.S. ONLY)	903508-17	CORD POWER UL & CSA MED BEIGE
368187-01	CARD COMMAND REFERENCE	367345-04	DISK ASSY A3000 SERIES KICKSTART ENGLISH
DISK ASSY 3.5" WORKBENCH V2.0 — PART OF 583501		335603-03	DISK ASSY A3000 SERIES INSTALL ENGLISH
317954-03	U.S.	367270-02	DISK ASSY WORKBENCH 1.3 A3000,A3000T ENGLISH
317240-01	INTERNATIONAL	367272-01	DISK ASSY EXTRAS 1.3 A3000,A3000T ENGLISH
317241-01	U.K.	318316-02	INSERT IMPORTANT NOTICE ENGLISH
317244-01	GERMAN	318246-02	INSERT CHECKLIST ENGLISH
317242-01	FRENCH	318245-03	INSERT SYSTEM S/W INCLUDED ENGLISH
317243-01	ITALIAN		

Commodore International Spare Parts List

MAJOR ASSEMBLIES

Commodore part numbers are provided for reference only and do not indicate the availability of spare parts from Commodore. Industry standard parts (Resistors, Capacitors, Connectors) should be secured locally. Part number information may vary according to country, some parts may not be available in all countries. See Section 3 for Dis-assembly diagrams.

MAIN ASSY A3000T		SEC. 3 DIAG. #	MAIN CHASSIS ASSY A3000T		SEC. 3 DIAG. #
313180-01	MAIN ASSY A3000T 4M 3H105M (PAL)		313196-01	MAIN CHASSIS ASSY A3000T 4M PAL	
313180-02	MAIN ASSY A3000T 4M 3F880K 3H105M NTSC		313196-02	MAIN CHASSIS ASSY A3000T 4M NTSC	
313180-03	MAIN ASSY A3000T 4M 3H210 PAL		313196-03	MAIN CHASSIS ASSY A3000T 8M PAL	
313180-04	MAIN ASSY A3000T 4M 3H210M NTSC		313196-04	MAIN CHASSIS ASSY A3000T 8M NTSC	
312937-01	TOP COVER	1	313196-05	MAIN CHASSIS ASSY A3000T 16M PAL	
313176-01	BEZEL ASSY	2	313196-06	MAIN CHASSIS ASSY A3000T 16M NTSC	
312938-01	SIDE COVERS	3	313181-02	PCB ASSY NTSC 25 MHZ 4M	1
312878-01	SCREW LT BEIGE M4 X 8.0 LG (QTY 6)	4	313181-01	PCB ASSY PAL 25 MHZ 4M	2
325542-02	SCREW EXT TOOTH M4 X 6.0 LG	5	313181-04	PCB ASSY 8 MEG NTSC	40
316416-01	LABEL WARNING	6	313181-03	PCB ASSY 8 MEG PAL	41
366062-02	LABEL RATING NTSC	9	313181-06	PCB ASSY 16 MEG NTSC	42
366062-04	LABEL RATING NTSC SUB FOR ABOVE		313181-05	PCB ASSY 16 MEG PAL	43
366062-01	LABEL RATING PAL	8	313333-02	MAIN FRAME WELDMENT ASSY	3
366062-03	LABEL RATING PAL SUB FOR ABOVE		390266-06	STANDOFF HEX 6.0 MM X 10.0 MM	4
366062-05	LABEL RATING PAL		390662-02	STANDOFF HEX M/F 6.0 MM	5
366062-06	LABEL RATING NTSC		363719-01	KEY/POWER BRACKET RETAINER	6
316583-06	LABEL HDD TYPE	15	390885-01	KEY/POWER SWITCH SUPPORT BRACKET ASSY	7
316583-04	LABEL HDD TYPE	16	313330-01	FOOT ASSY	8
313177-03	CPU BASE ASSY PAL 210M HD 4M	17	312313-01	REAR PANEL	9
313177-04	CPU BASE ASSY NTSC 210M HD 4M	18	380120-01	EXTENSION CARD PANEL	10
313177-01	CPU BASE ASSY PAL 105M HD 4M	19	251118-01	PCB GUIDE	
313177-02	CPU BASE ASSY NTSC 105M HD 4M	20	313123-01	EXT CARD PANEL VIDEO	13
325090-02	SEAL WARRANTY PLACE OVER 325542-02 SCREW		380739-01	PANEL EXPANSION SLOTS	14
313176-01 BEZEL ASSY A3000T		SEC. 3 DIAG. #	390484-02	POWER SUPPLY (CSA,UL)	16
313176-02	BEZEL ASSY A3000T UX		390484-01	POWER SUPPLY (VDE,BSI,SEV & SAA)	15
363692-01	BEZEL A3000T	1	380742-04	CABLE ASSY FLOPPY DISK	18
363692-02	BEZEL A3000T UX	2	314037-02	CABLE ASSY HARD DISK	19
906883-03	SCREW SELF TAPPING M3 X 6.0 LG	3	390886-01	CABLE HARNESS ASSY	20
312679-03	FDD HOLE COVER BEZEL 5.25" LIGHT BEIGE	4	312993-02	TERMINAL LUG	
363691-01	FDD HOLE COVER 3.5"	5	390329-01	SCREW MACH M3.5 X 0.5 X 5.0 LG (QTY 9)	25
313317-03	LENS LED (YELLOW)	6	906880-02	SCREW MACH M3 X 0.5 X 10.0 LG (QTY 2)	26
CPU BASE ASSY A3000T		SEC. 3 DIAG. #	906880-05	SCREW MACH M3 X 0.5 X 6.0 LG (QTY 15)	27
313177-01	CPU BASE ASSY A3000T (PA 4M FAST 3H105M)		906880-02	SCREW MACH FLT HD M3 X 0.5 X 10.0 LG (QTY 6)	28
313177-02	CPU BASE ASSY A3000T (NTS 4M FAST 3H105M)		906810-03	SCREW MACH M4 X 10.0 LG (QTY 2)	
313177-03	CPU BASE ASSY A3000T (PAL) 4M FAST 3H210M)		390251-03	STAND OFF "D" CONNECT	
313177-04	CPU BASE ASSY A3000T (NTS 4M FAST 3H210M)		390251-04	STAND OFF HEX M/F 5MM	
312948-01	BRACKET SUPPORT RETAINER 5.25" FDD (QTY 2)	1	312463-01	SERIAL PORT ASSY	31
312949-01	BRACKET SUPPORT RETAINER 3.5" FDD	2	312998-01	WIRE CLIP ADHESIVE BACKED	34
313182-01	FDDRIVE ASSY 3.5"	3	390896-01	CABLE ROUTING CLIP 50 POSITIONS	35
313179-01	HDD CHASSIS ASSY 105M	4	390762-02	EDGE PROTECTOR TAPE ADHESIVE BACKED	37
325542-02	SCREW MACH EXT TH M4 X 6.0 LG	5	390798-01 FAN ASSY A3000T		
906800-07	SCREW MACH M3 X 0.5 X 5.0 LG	6	390812-01	DC BRUSHLESS FAN 80 MM	
390728-01	SPEAKER ASSY	7	390813-02	CONN HOUSING 3 POSITION	
390798-01	FAN ASSY	8	390823-01	CRIMP TERMINAL	
313196-01	MAIN CHASSIS ASSY A3000T PAL 4M FAST	9	390902-01	SELF RETAINING FASTENER	
313196-02	MAIN CHASSIS ASSY A3000T NTSC 4M FAST	10	313182-01 FLOPPY DRIVE ASSY 3.5" A3000T		SEC. 3 DIAG. #
906610-02	SCREW MACH #6 - 32 X 1/2" (QTY 4) USE W/FAN & FILTER	11	313315-01	FLOPPY DISK DRIVE 3.5"	1
312878-01	SCREW MACH LT BG M4 X 8.0 LG (QTY 4) USE ON HDD CHASSIS	12	312593-02	F.D.D. SLIDER 3.5"	2
390893-01	SPEAKER BRACKET	13	906800-07	SCREW MACH M3 X 0.5 X 5.0 LG.	3
390892-01	FAN FILTER	14	390886-01 CABLE HARNESS ASSY A3000T		
312833-01	RF SHIELD 3.5" FDD	16	390910-01	SINGLE ROW FEMALE CONN 9 POSITION	
312834-01	RF SHIELD 5.25" FDD	17	903756-06	LEAD WIRE AWG 24 BLUE L = 350 MM	
312832-01	RF SHIELD HDD	18	905051-03	CABLE TIE	
312813-01	RF SHIELD SWITCH OPENING	19	312463-01 SERIAL PORT ASSY A3000T		
313179-02	H.D.D. CHASSIS ASSY 210M	21	390725-01	CONN D SUB 25 PIN MALE IDC	
HARD DISK DRIVE CHASSIS ASSY A3000T		SEC. 3 DIAG. #	903135-08	FLAT CABLE 25 POL L=50 MM	
313179-01	HARD DISK DRIVE CHASSIS ASSY A3000T 105MB		903468-04	CONN DIL FEMALE 26 POL	
313179-02	HARD DISK DRIVE CHASSIS ASSY A3000T 210MB		390885-01 KEY/POWER SWITCH SUPPORT BRACKET A3000T		SEC. 3 DIAG. #
312947-01	HDD CHASSIS	1	363693-01	POWER/KEY SUPPORT BRACKET	
312951-01	PLATE ADJUSTABLE SLIDE	2	363720-01	POWER/KEY BEZEL	2
312950-01	HDD RETAINER	3	390905-01	POWER/KEY BEZEL DECAL	3
906800-05	SCREW MACH M3 X 0.5 X 6.0 LG (QTY 10)	4	390908-01	CONN SINGLE ROW PANEL MOUNT	4
313339-01	HDD SUPPORT STOP (QTY 4)	6	906883-01	SCREW TYPE B M3 X 8 (QTY 2) USE ON BEZEL	5
311836-01	HDD 105MB SCSI	7	906883-02	SCREW TYPE B M3 X 4 (QTY 1) USE ON LED HOLDER	6
906610-01	SCREW MACH #6 - 32 X .250" LG (QTY 4)		906800-07	SCREW MACH M3 X 5 USE ON POWER SWITCH	7
311838-03	HD DRIVE 210MB SCSI	9	390880-01	POWER BUTTON	8
390728-01 SPEAKER ASSY A3000T			312970-02	KEYSWITCH LOCK ASSY	9
390814-01	SPEAKER 1.5W 8 OHM		312969-03	LINE POWER SWITCH ASSY	10
390813-01	CONNECTOR HOUSING 2 POSITION		313303-02	LED HOLDER ASSY	11
390823-01	CRIMP TERMINAL		906115-08	TORROID (WRAP CABLES 312970-02 & 313303-02 TWO TURNS	
200016-21	LEAD WIRE AWG 26 RED L = 340MM				
200016-110	LEAD WIRE AWG 26 BLACK L = 340MM				

Commodore International Spare Parts List

PCB Components

PCB Assembly #313181

Commodore part numbers are provided for reference only and do not indicate the availability of spare parts from Commodore. Industry standard parts (Resistors, Capacitors, Connectors) should be secured locally. Part number information may vary according to country, some parts may not be available in all countries.

IC COMPONENTS			IC SOCKETS (Continued)		
901882-01	1488	U304	390043-01	DIP SHUNTS	J100,J102-J104,J151, J152,J180,J181,J200, J350-J352,J481,J482, J851,J852
901883-01	1489	U305	390719-01	SINGLE SOCKET PINS	U111
390754-01	74HC4066	U901,U904,U941	390768-03	14 PIN DIP .300 (MACHINE PIN)	U103
901522-30	7407	U352,U713,U905	CONNECTORS		
390545-01	74F163A	U476	390995-01	KEYBOARD 5 PIN DIN SHIELDED	CN420
390110-01	74F04	U204,U711	390557-01	200 PIN RECEPTACLE	CN606
390203-01	74F08	U710	390596-03	34 PIN DIL POLARIZED SHROUDED	CN351
390676-01	74ALS74	U480	390596-04	50 PIN DIL POLARIZED SHROUDED	CN802
390108-01	74F373	U251	252122-05	RCA JACK	CN402
390089-01	74F245	U257,U258,U702,U708, U709,U715,U716,U891- U894	390334-01	DB15 VGA	CN480
390077-01	74F32	U712	390242-03	D SUB 23 PIN MALE PCB	CN450
390281-01	74F646	U253,U256,U703-U707	390625-02	DUAL D SUB 9 PIN MALE PCB RIGHT	CN400,CN401
390081-01	74F74	U106,U740	327033-04	D SUB 23 PIN FEMALE PCB	CN352
318049-01	74HC04	U477	903446-04	36 PIN EDGE	CN451,CN452,CN755- CN758
318828-01	74HCT244	U451,U452	903446-02	62 PIN EDGE	CN751-CN754
901521-03	74LS08	U355	903446-06	100 PIN EDGE	CN600-CN604
901521-11	74LS157	U401	390625-01	DUAL DB25 FEMALE	CN350,CN801
901521-63	74LS174	U195,U354	390475-01	POWER 24 PIN	CN160
390198-01	74F86	U252	903345-13	26 PIN DIL .1	CN300
901521-01	74LS00	U353	OSCILLATORS, TRANSISTORS AND DIODES		
901521-06	74LS74A	U201,U351	252344-01	OSCILLATOR 28.375 MHZ PAL	U111
318069-03	8372 FAT AGNUS 2 MEG	U205	325566-14	OSCILLATOR 28.6363 MHZ NTSC	U111
318029-03	8520	U300,U350	325566-12	OSCILLATOR 28.6363 MHZ (SUB FOR 325566-14)	
390538-03	AMBER	U478	900750-01	DIODE 1N4001	D192,D193,D800
390433-01	8373 ECS DENISE	U450	900850-01	DIODE 1N4148	D190,D191,D300,D951
390537-02	DPMC	U802	390192-01	TRANSISTOR 2N3904	Q300
390539-07	FAT BUSTER	U700	902707-01	TRANSISTOR 2N3906	Q400
390540-02	FAT GARY	U110	390254-01	TRANSISTOR MPF102	Q430,Q440
380227-01	74HCT32	U600	902686-02	TRANSISTOR 2222A	Q951,Q952
390530-01	MSM 514221RS-6	U470-U472	900560-01	CRYSTAL 32768 HZ	Y190
390399-04	MC68030 25 MHZ QFP	U100	CAPACITORS		
390434-02	MC68882 25MHZ PLCC	U101	900019-39	7.5PF NPO RAD	C499
390531-01	NE564	U481	900462-20	20PF AXIAL	C192
390086-01	LF347	U402,U900	900019-37	39PF RAD	C497
252127-02	PAULA	U400	900462-29	47PF AXIAL	C458
390532-01	UPD42101C-2	U473-U475,U479	900462-31	56PF AXIAL	C193
390206-03	WD33C93A	U800	900462-37	100PF NPO AXIAL	C421,C459,C491
390525-01	RP5C01	U190	900463-08	220PF AXIAL	C108,C109
390764-01	TL780-05CKC 5 VOLT REGULATOR 2%	U198	900463-23	3900PF AXIAL	C432,C442
390630-02	ROM V1.4 32 BIT LOW WORD	U180	900463-26	6800PF AXIAL	C431,C441
390629-02	ROM V1.4 32 BIT HIGH WORD	U181	900019-22	.01UF RAD	C161,C170,C172,C174, C176,C177,C402
390997-01	74FACTQ240	U105	900463-36	.047UF AXIAL	C411-C414
318099-03	DRAM 256K X 4 120NS ZIP	U259-U266	900019-38	100PF NPO RAD	C496,C498
318099-01	DRAM 256K X 4 100NS ZIP	SUB FOR 318099-03	390082-01	.1UF AXIAL MLC	C103,C105,C106,C191, C195,C201-C205,C251- C258,C300,C305,C350- C355,C362,C400,C401, C409,C410,C415,C434, C444,C451,C452,C470- C482,C484-C486,C490,
318099-05	DRAM 256K X 4 80NS ZIP	SUB FOR 318099-03	900463-12	470PF AXIAL MLC	C494,C495,C606,C901, C904-C906,C941,C907, C740,C123
390668-06	DRAM 256K X 4 ST COL 120NS ZIP	SUB FOR 318099-03	390082-05	.22UF AXIAL MLC	C121,C122
390668-03	DRAM 256K X 4 ST COL 100NS ZIP	SUB FOR 318099-03	390082-04	.33UF AXIAL MLC	C101,C110,C180,C181, C403,C405,C406,C450, C455,C456,C457,C700- C716,C752-C755,C849, C902,C971
390668-04	DRAM 256K X 4 ST COL 80NS ZIP	SUB FOR 318099-03			C100,C102,C104,C111, C199,C259,C261,C263, C265,C800,C801,C850, C852,C854,C856,C858, C860,C862,C864,C866,
390668-04	DRAM 1M X 4 ST COL 80NS ZIP	SUB FOR 318099-03			
390773-01	DRAM 1M X 4 ST COL 80NS ZIP	U850-U865			
390541-04	RAMSEY	U890			
325566-27	OSC 50MHZ	U104			
390555-01	DELAY LINE 25NS 5 TAP	U102			
390776-01	LM386	U903			
390526-02	16L8D PROG	U202			
390527-02	16L8D PROG	U203			
390528-02	16R4A PROG	U701			
390529-03	16L8D PROG	U714			
IC SOCKETS					
904150-07	18 PIN DIP .300	U190			
904150-08	20 PIN DIP .300	U202,U203,U701,714			
904150-06	40 PIN DIP	U180,U181,U300,U350, U800			
251313-01	48 PIN DIP	U400,U450			
390185-02	68 PIN PLCC	U478			
390185-01	84 PIN PLCC	U890,U110,U205,U700, U802			
390724-01	8 PIN SIL	RP802-RP804			
390723-01	20 PIN ZIP	U267-U274,U850-U881			

Commodore International Spare Parts List
PCB Components (Continued)
PCB Assembly #313181

CAPACITORS (Continued)			RESISTOR NETWORKS		
390082-04	.33UF AXIAL MLC (Continued)	C868,C870,C872,C874, C876,C878,C880,C890- C894,C267,C269,C271, C273	326149-01 22 OHM 4 X 8 390227-03 22 OHM 5 X 10 326149-04 47 OHM 4 X 8	RP700,RP890,RP891 RP892,RP893 RP252-RP254,RP451, RP452	
900014-02	1000PF CER RAD	C487,C488,C756-C799	390227-06 47 OHM 5 X 10	RP251,RP255,RP201	
900463-06	CAP 150PF AXIAL MLC	C910,C920	326149-06 68 OHM 4 X 8	RP401	
	UNSTUFFED RESERVE CAPACITORS	C307,C308,C430,C440	902441-17 470 OHM 5 X 6	RP701	
390101-05	RAD ELEC 4.7UF	C190,C404	902441-22 1K 5 X 6	RP600,RP601,RP611, RP702	
900402-01	RAD ELEC 10UF	C483	902422-22 1K 7 X 8	RP180,RP609,RP610	
390101-04	RAD ELEC 22UF	C107,C113,C317,C198, C318,C433,C443,C751, C911,C912,C921,C922, C112	902410-10 1K 9 X 10	RP100-RP110,RP256, RP351,RP703,RP752- RP756	
390101-01	RAD ELEC 47UF	C162-169,C171,C173, C175,C197,C407,C408, C899	902410-06 3.3K 9 X 10	RP350,RP800,RP801	
390101-03	RAD ELEC 470 UF	C160,C416,C423,C454, C460,C492,C493	902441-31 4.7K 5 X 6	RP453-RP455	
900101-47	AXIAL ELEC 220 UF	C601-C604	902410-08 4.7K 9 X 10	RP400,RP751	
900109-09	AXIAL ELEC 100 UF 25V	C909	380388-04 220/330 OHM 8 X 10	RP602-RP608	
900101-04	AXIAL ELEC 10 UF 16V	C908	UNSTUFFED RESISTOR PACK	RP450,RP802-RP804	
900101-38	AXIAL ELEC 470 UF	C972			
RESISTORS — 1/4W ± 5% UNLESS OTHERWISE SPECIFIED					
901550-118	1 OHM	R400,R479,R403,R404	390546-01 VARIABLE RESISTOR 1K 15 TURN	VR470	
901550-64	10 OHM	R484	251029-05 VARIABLE CAPACITORS 2-7 PF NPO	VC470	
901550-63	22 OHM	R120,R450	251029-06 VARIABLE CAPACITORS 6.8 - 45 PF	VC190	
901550-105	33 OHM	R472,R473,R475-R477	380393-01 BATTERY 3.6V 60 MAH NICAD	BT190	
901550-56	47 OHM	R202,R204	390536-02 SWITCH S.P.S.T. TOGGLE RIGHT ANGLE	SW481	
901550-49	100 OHM	R489,R809	390275-03 EMI FILTER 3 PIN 330 PF W/FB	EMI 300-EMI 309, EMI 350-EMI 364	
901550-131	56 OHM	R300	390297-04 EMI FILTER 3 PIN 470 PF W/FB	EMI 430-EMI 436, EMI 440-EMI 446,	
901550-108	360 OHM	R430,R440	390297-02 EMI FILTER 3 PIN 100 PF W/FB	EMI 800-EMI 817	
901550-57	390 OHM	R434,R444		EMI 454-EMI 456, EMI 420,EMI 422,	
901550-58	470 OHM	R190,R192,R468,R699, R909,R911	251842-02 EMI FILTER 3 PIN 100PF	EMI 380-EMI 394, EMI 458,EMI 459	
901550-01	1K	R116,R117,R198,R302, R433,R443,R482,R483, R487,R600,R803,R805, R469,R900,R910	390257-03 EMI FILTER 3 PIN 270PF W/FB	EMI 450,EMI 493- EMI 495	
901550-23	2.7K	R407	252173-01 FERRITE BEAD 1 TURN RADIAL	EMI 400,EMI 401	
901550-02	3.3K	R110,R470,R471,R905	903025-01 FERRITE BEAD	FB453,FB469	
901550-19	4.7K	R203,R264		FB110,FB111,FB117, FB480-FB487,FB740,	
901550-22	47K	R480,R901	252214-01 FERRITE BEAD	FB895,FB896,FB400, FB401	
901550-20	10K	R305,R306,R405,R406, R408,R431,R432,R441, R442,R451,R452,R454, R917,R913,R927	390298-01 FERRITE BEAD	FB460-FB468	
901550-07	100K	R193	390253-03 FERRITE BEAD	FB121,FB112-FB116	
901550-82	470K	R435,R445	390924-02 FUSE POLYSWITCH	FB421	
901550-112	200 OHM	R301		RDE 185,RDE 250 (RAYCHEM RXE090)	
901550-17	1.2K	R401,R402,R921,R922	390280-01 FUSE 3 AMP	F351,F400	
901550-15	27K	R918,R928	390229-03 VIDEO HYBRID	HY450,HY480	
901550-06	33K	R916,R926		(390229-D,390229-E)	
901550-94	68 OHM	R307	903326-03 SIL HEADER .100 1 X 3	J102,J104-107,J151, J152,J200,J350-J352, J482,J483,J851,J852	
901550-139	2.2 OHM	R902	903326-04 SIL HEADER .100 1 X 4	J100,J103	
901600-36	CARBON 1/2W 5% 1 OHM	R420	390822-01 SIL HEADER W/LKG .100 1 X 2	JSP	
901600-15	CARBON 1/2W 5% 47 OHM	R353	390822-02 SIL HEADER W/LKG .100 1 X 3	CN162,JCDINP	
901650-19	CARBON 1 W 5% 33 OHM	R197,R199	390909-01 SIL HEADER W/LKG .100 1 X 9	CN302	
	UNSTUFFED RESISTOR	R409	390775-01 HEAT SINK T0220 W/STUD (USE W/390764-01 IC TL780-05CKC U198)		
			905960-03 NUT HEX M3 (USE W/390775-01)		
			905655-03 LOCK WASHER (USE W/390775-01)		
			904907-01 HEAT SINK COMPOUND (USE W/390775-01)		

SECTION 5
SCHEMATICS

S
C
H
E
M
A
T
I
C
S

A3000T SCHEMATIC PAGE INDEX

TITLE	SHEET
A3000T SCHEMATIC PAGE INDEX	1 of 17
JUMPER SETTINGS	2 of 17
68030 CPU, 68881 FPU	3 of 17
GARY	4 of 17
ROM, RTC	5 of 17
8372 FAT AGNUS	6 of 17
CHIP RAM	7 of 17
8520, SERIAL CONN	8 of 17
8520, PARALLEL CONN, FLOPPY CONN	9 of 17
PAULA, MOUSE CONN	10 of 17
DENISE, VIDEO PORT	11 of 17
AMBER, VGA CONN	12 of 17
EXPANSION SLOTS	13 of 17
CO-PROCESSOR SLOT (FAST SLOT)	14 of 17
FAT BUSTER	15 of 17
DMAC, WD33C93A HDC, SCSI (INT), SCSI (EXT)	16 of 17
RAMSEY, FAST RAM	17 of 17

JUMPER SETTINGS

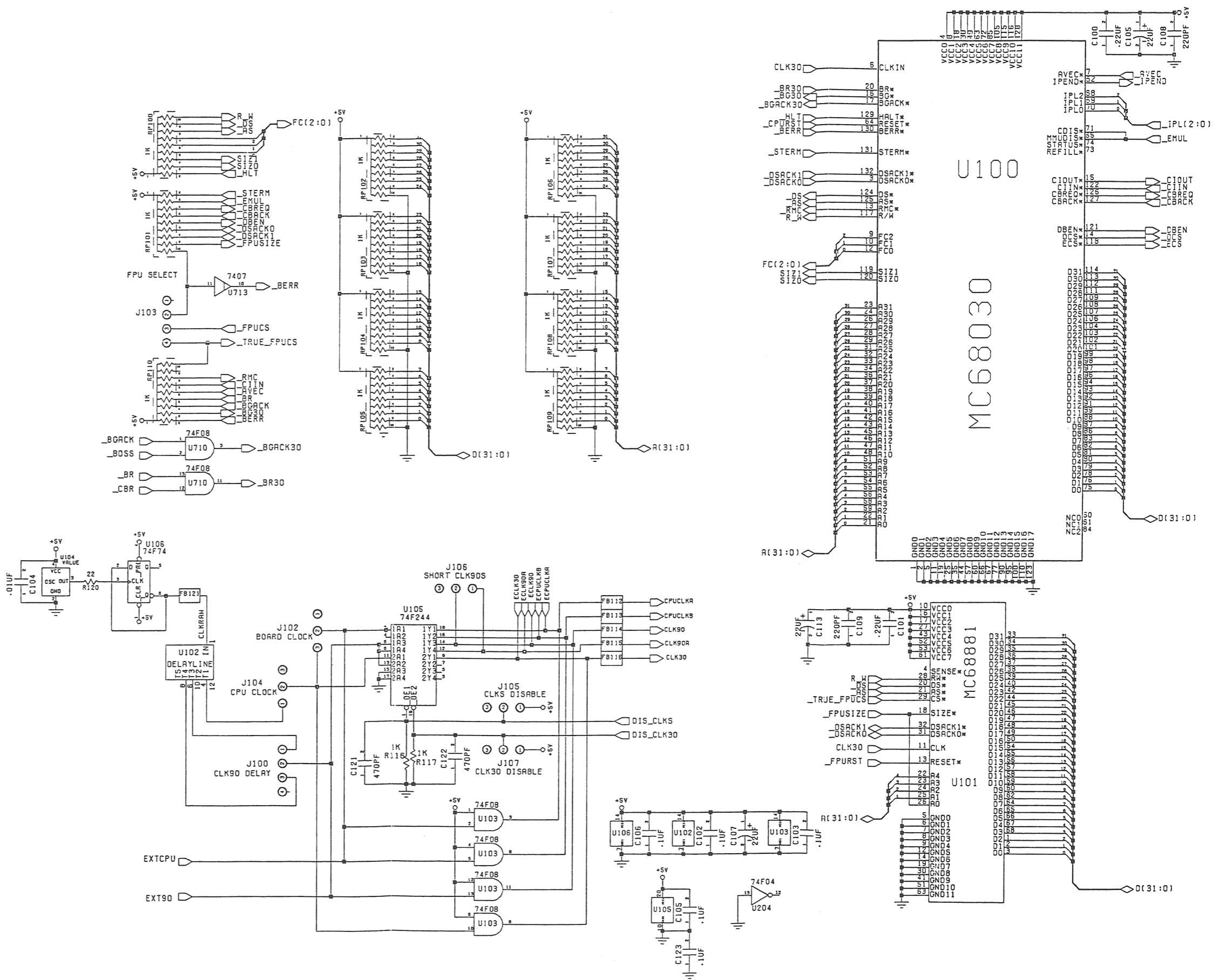
JUMPER	PINS	313181 -01, -03, -05	313181 -02, -04, -06	DESCRIPTION
J100	4	1-2	1-2	CLK90 Delay
J102	3	2-3	2-3	System clock source
J103	4	3-4	3-4	FPU enable & chip select source
J104	3	1-2	1-2	CPU clock source
J105	3	2-3	2-3	Enable/Disable clock driver (except CLK30)
J106	3	2-3	2-3	Short CLK90 signals together
J107	3	2-3	2-3	Enable/Disable clock driver to CPU (CLK30)
J151	3	1-2	1-2	ROM timing (see table below)
J152	3	1-2	1-2	ROM timing
J200	3	1-2	2-3	NTSC/PAL
J350	3	1-2	1-2	Tick clock source
J351	3	2-3	2-3	Floppy drive 1 enable/disable
J352	3	2-3	2-3	Light pen source
J482	3	1-2	1-2	VE PPL loop adjust
J483	3	—	—	Factory test points
J851	3	2-3	2-3	RAM controller speed
J852	3	1-2	1-2	RAM chip size

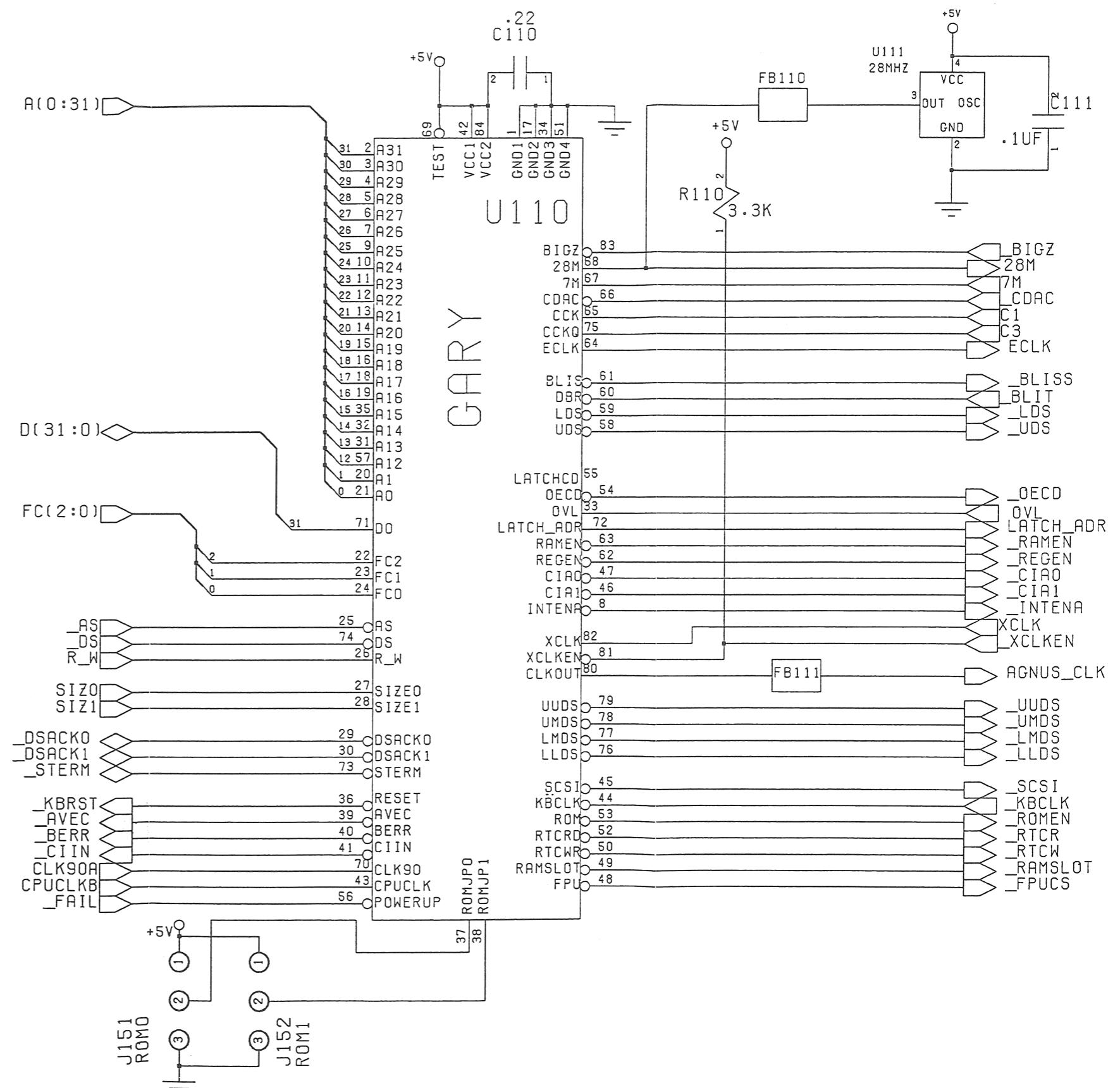
The Rom timing circuitry in GARY provides for 4 different speed settings to match the output enable and access times of the ROMs being used.

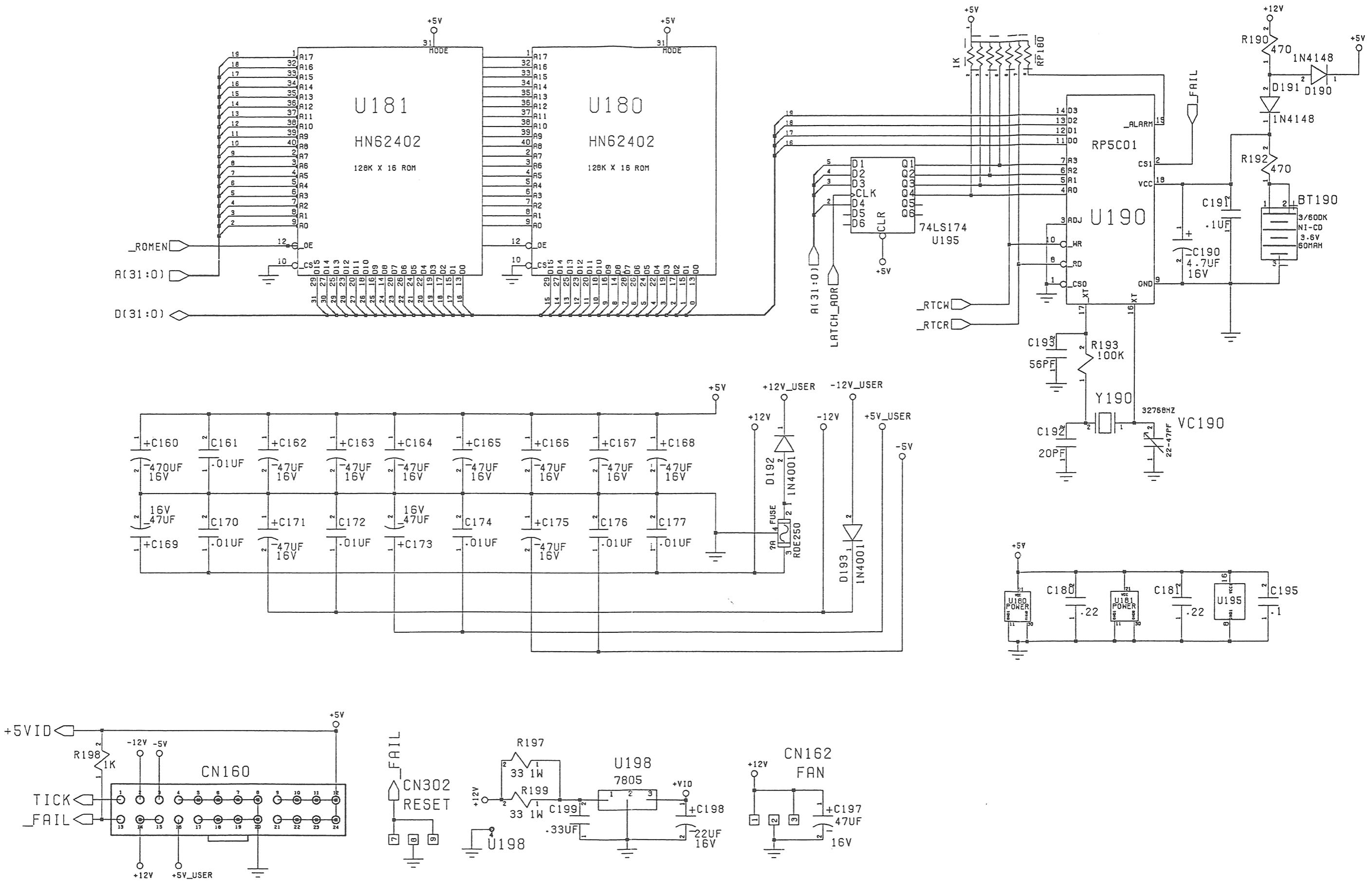
313181 -01 thru -06 are all 25 Mhz systems

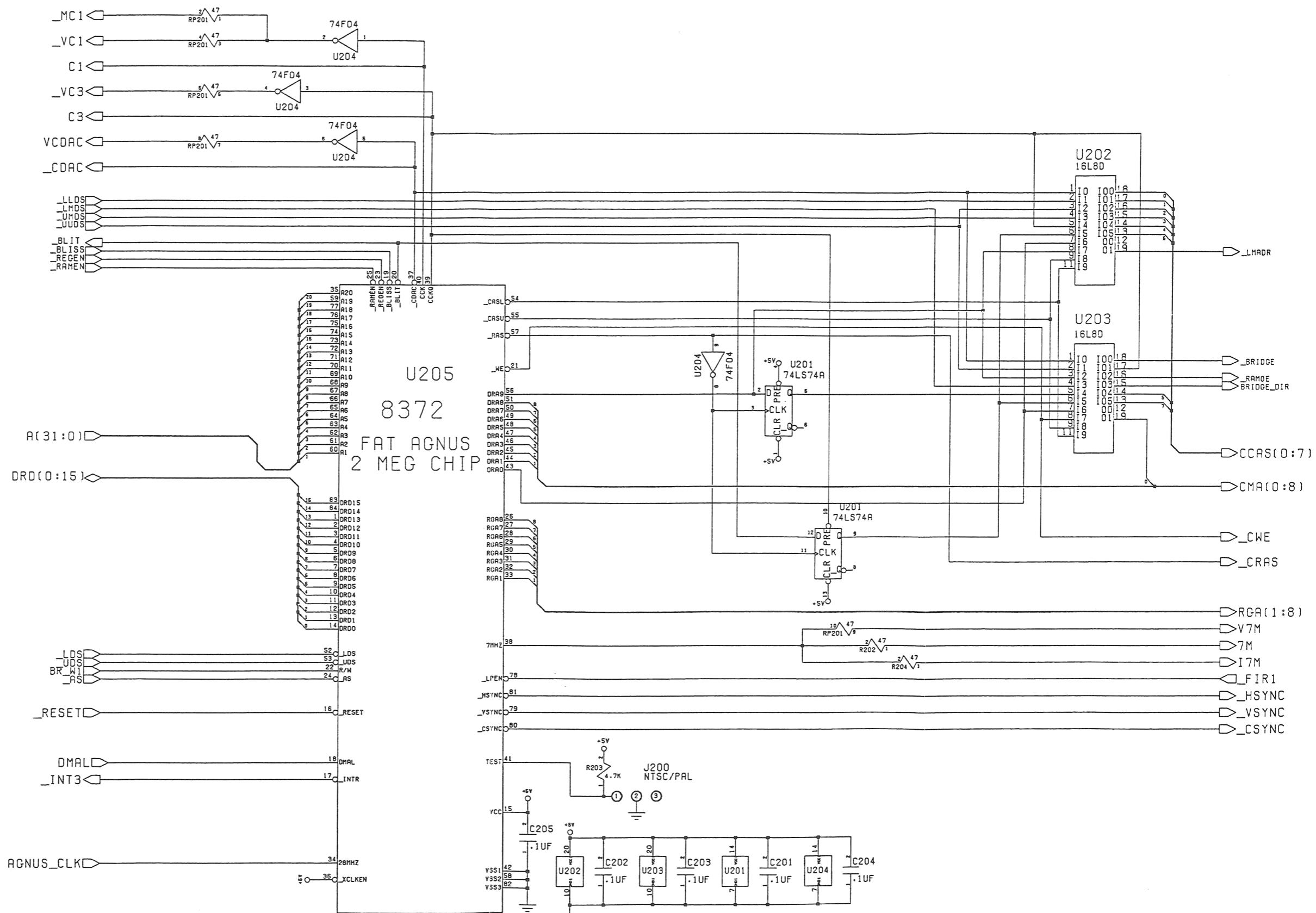
J151	J152	Toe(min)	Tacc(min)
2-3	2-3	90 ns	160 ns
2-3	1-2	130	200
1-2	2-3	170	240
1-2	1-2	210	280

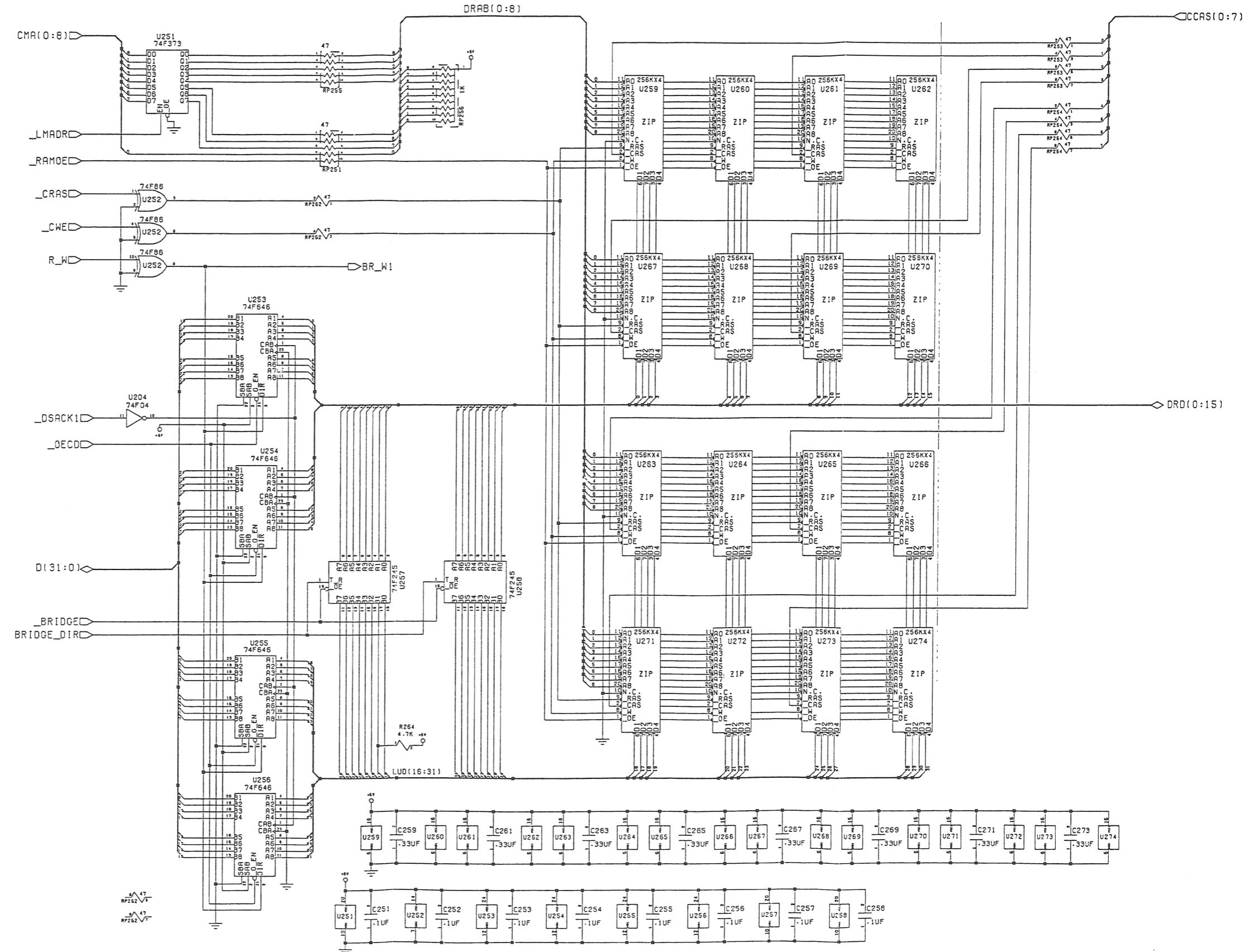
See Page 4-1 for Board Layout.

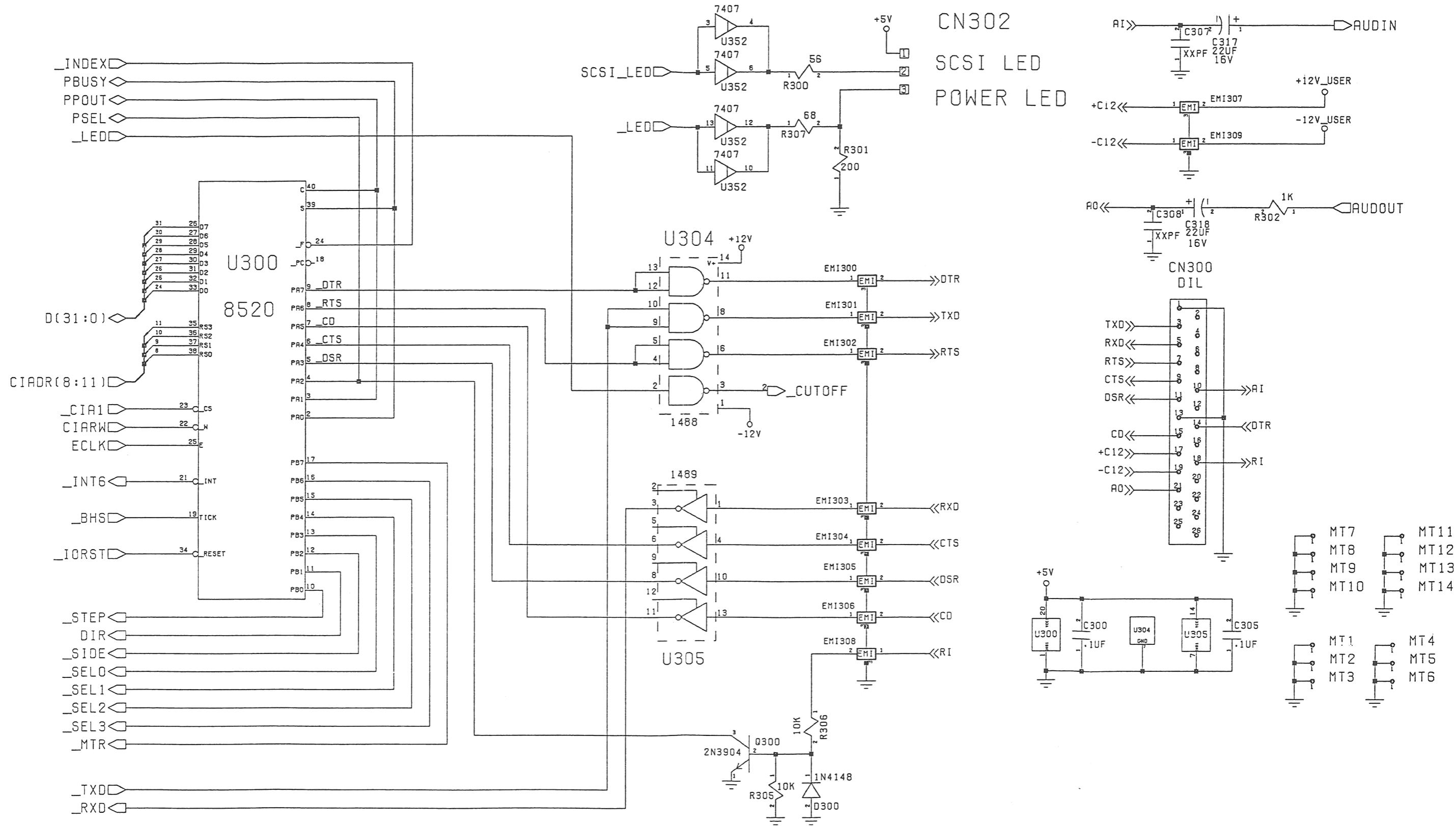


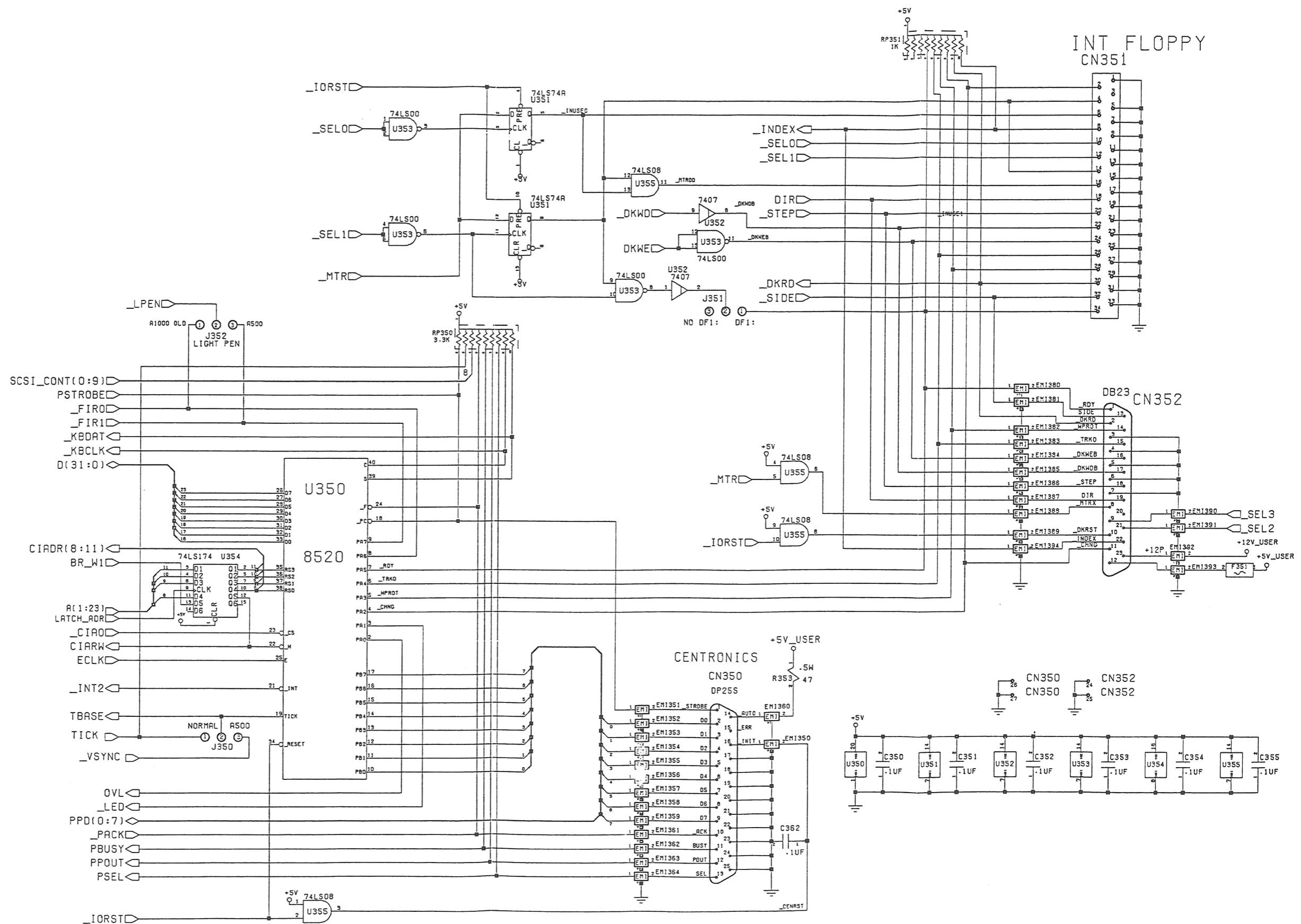


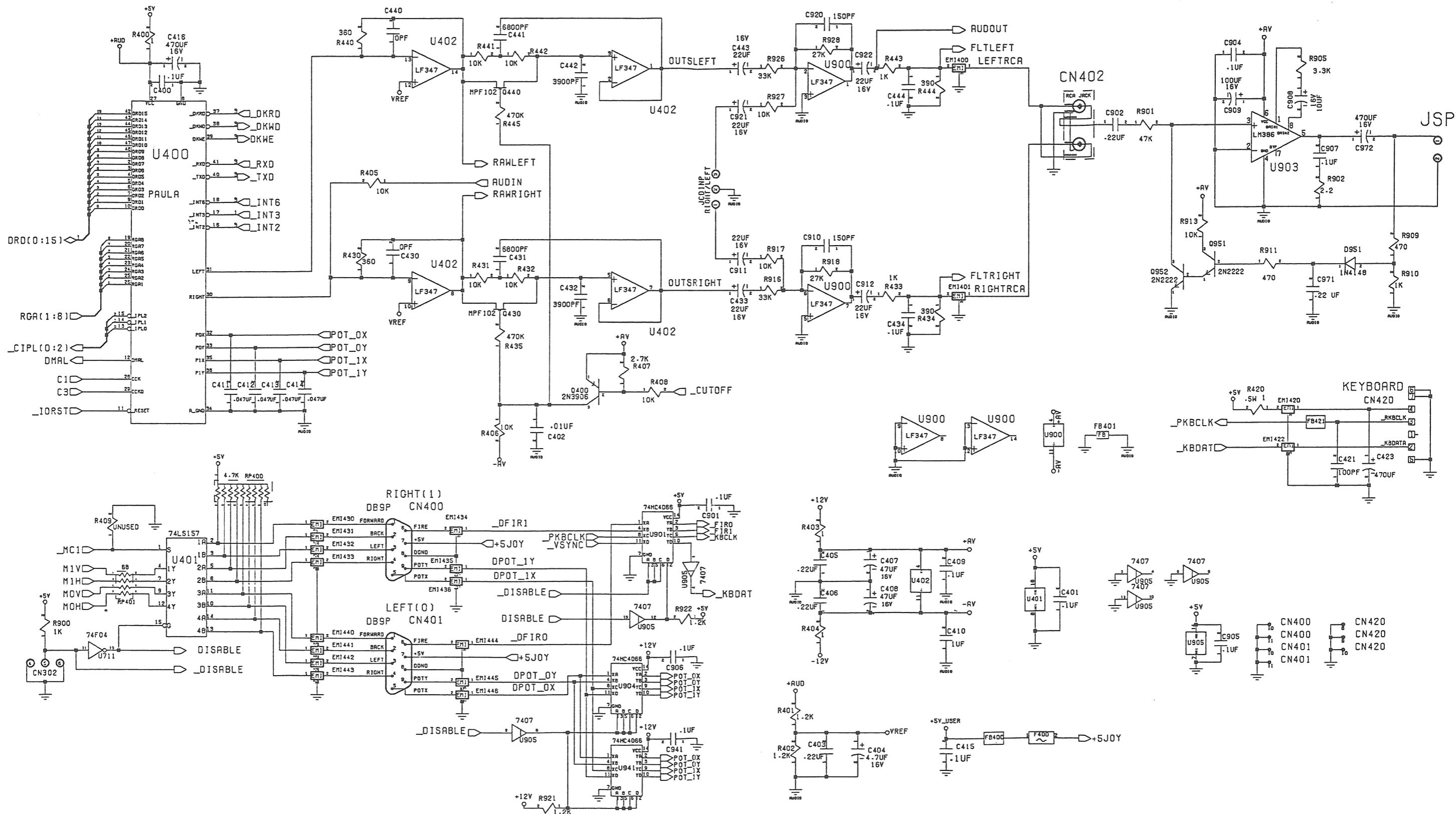


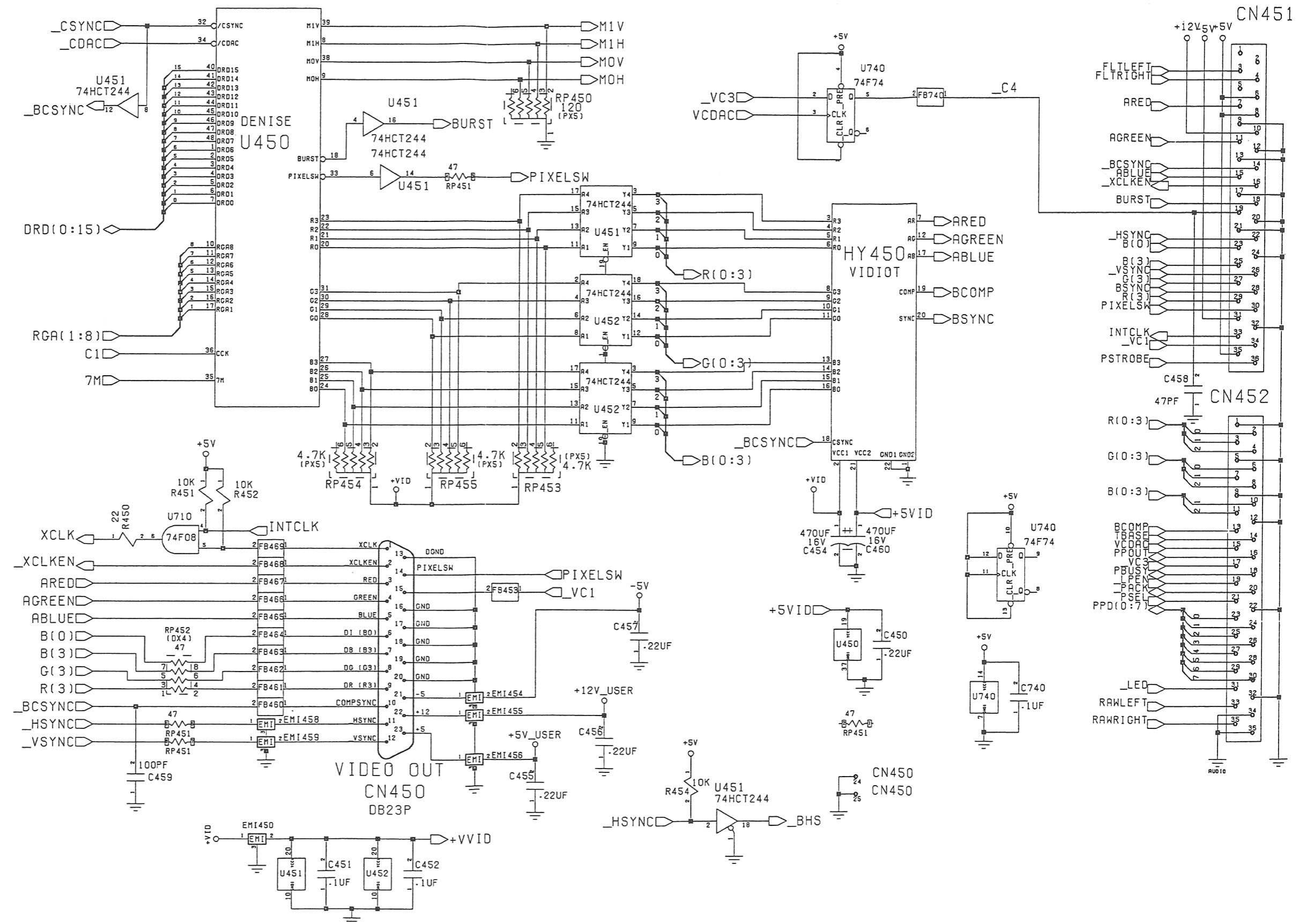


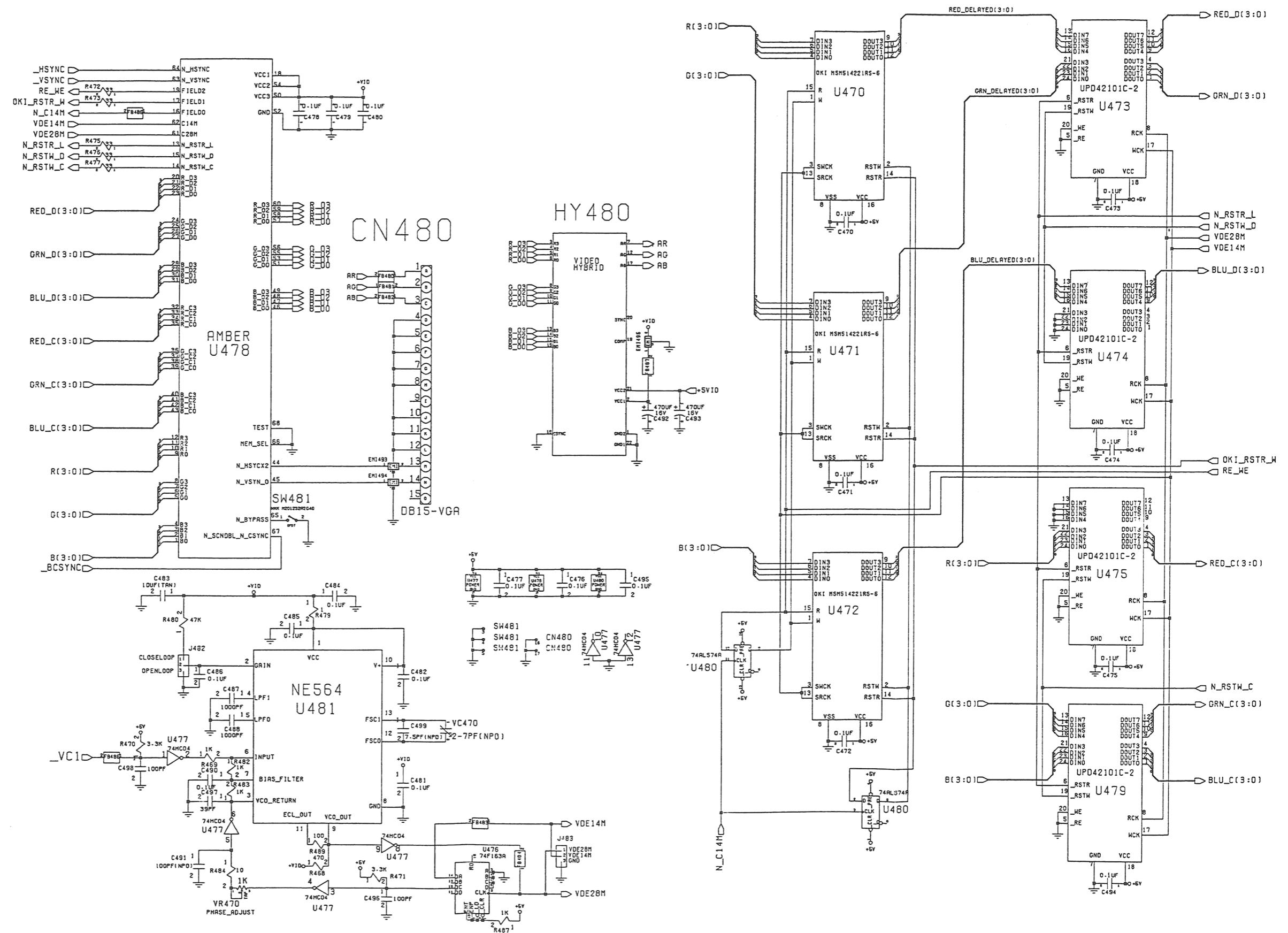


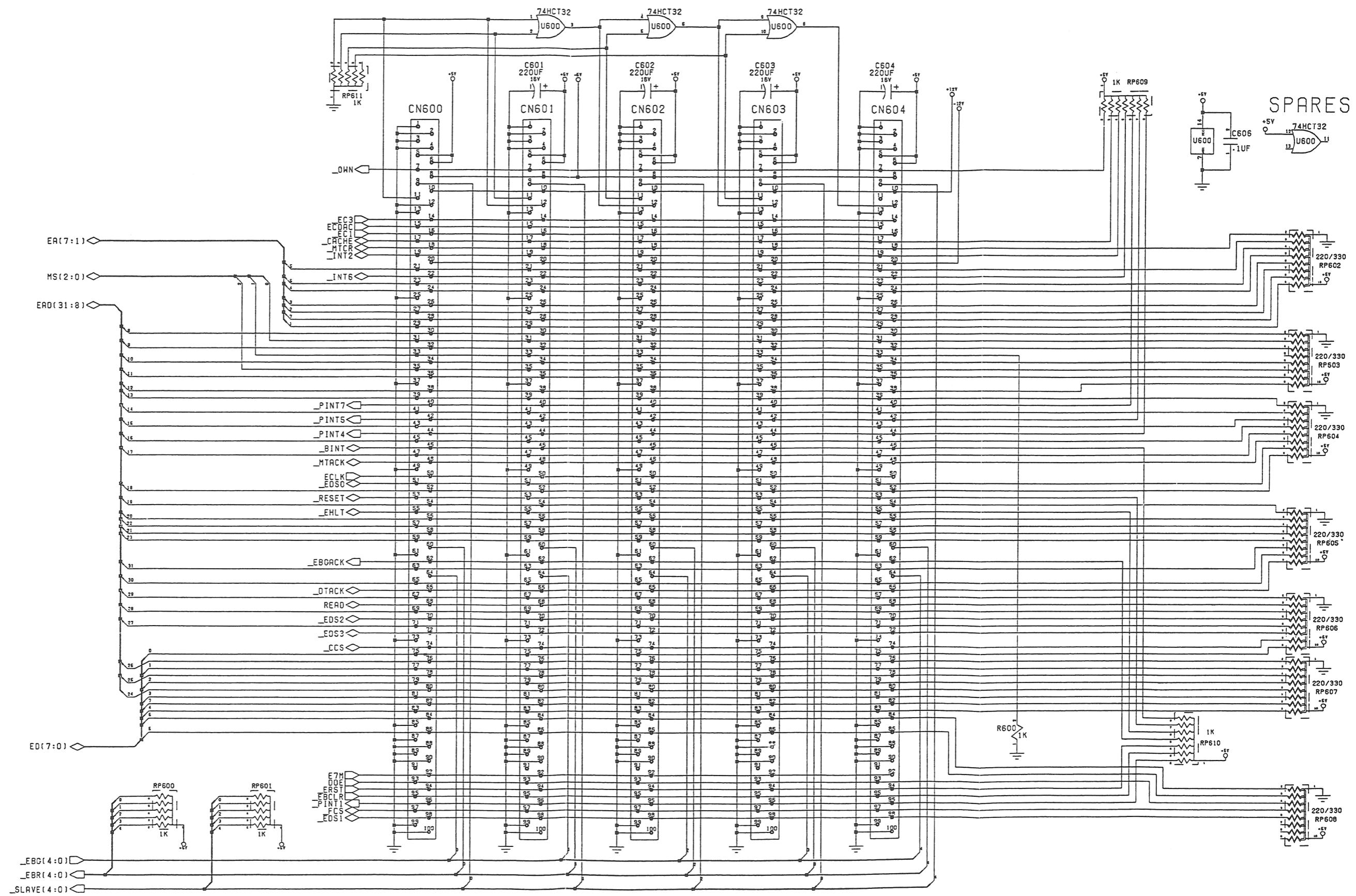








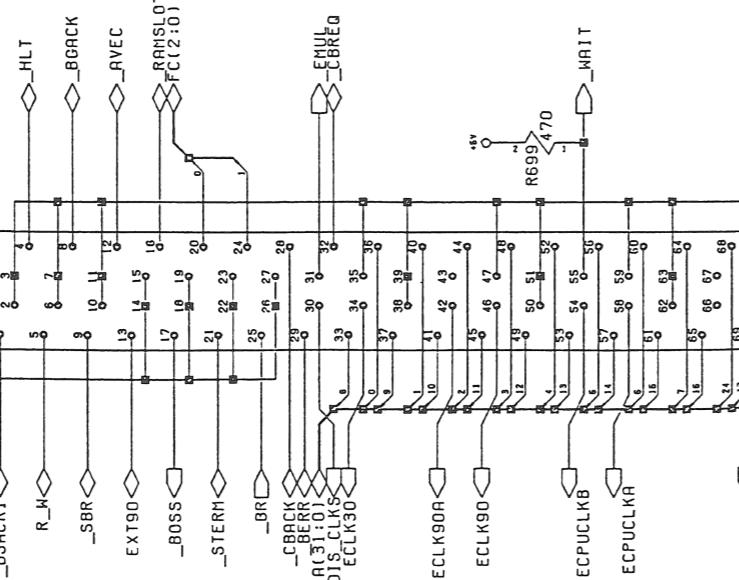




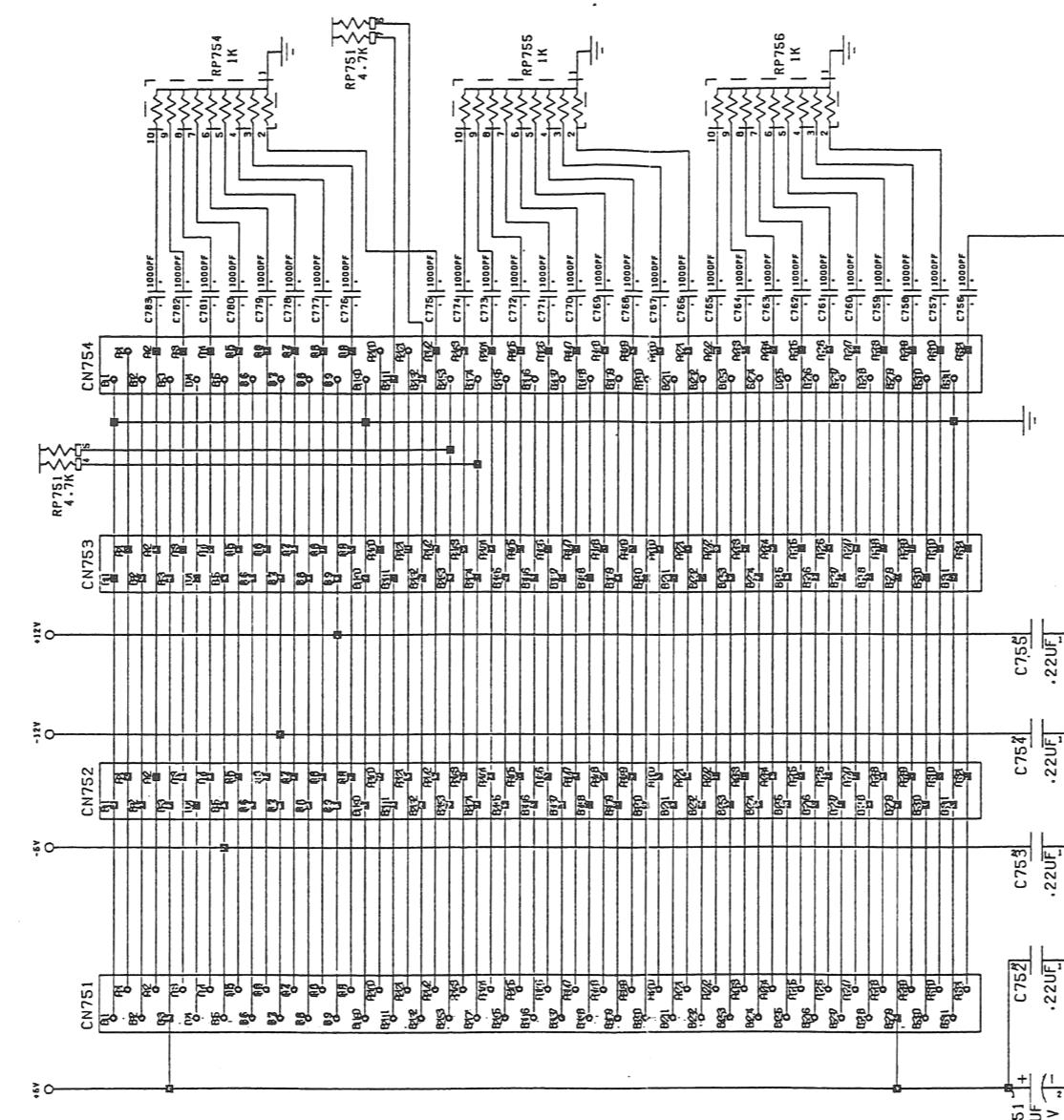
AMIGA 32 BIT LOCAL SLOT

CN606

1

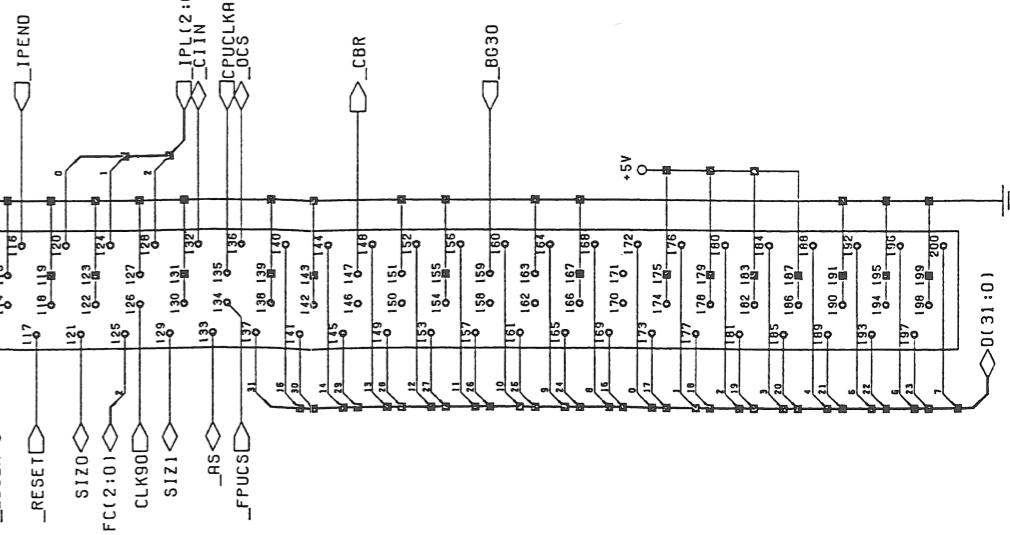


IBM 8 BIT SLOTS

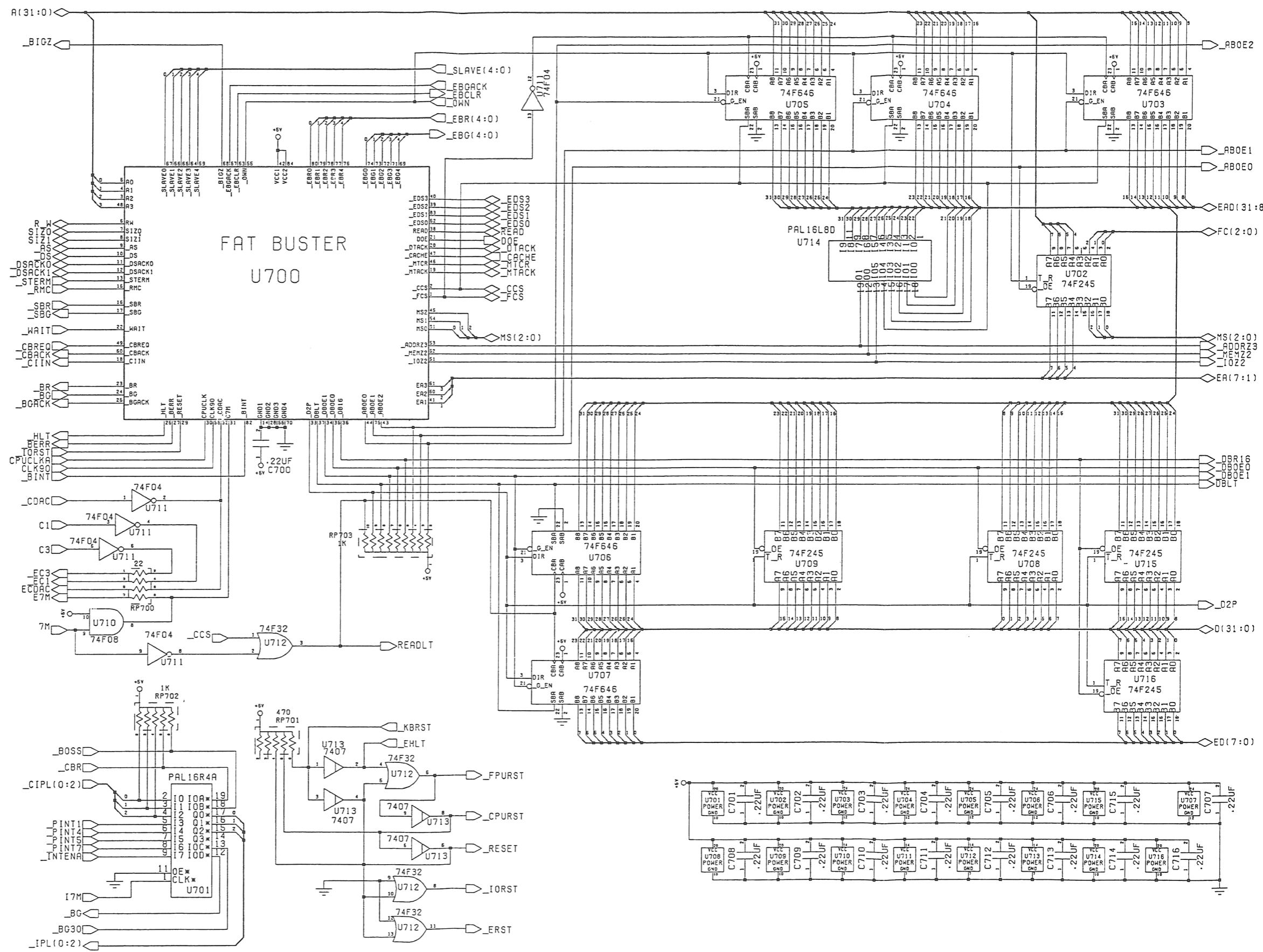


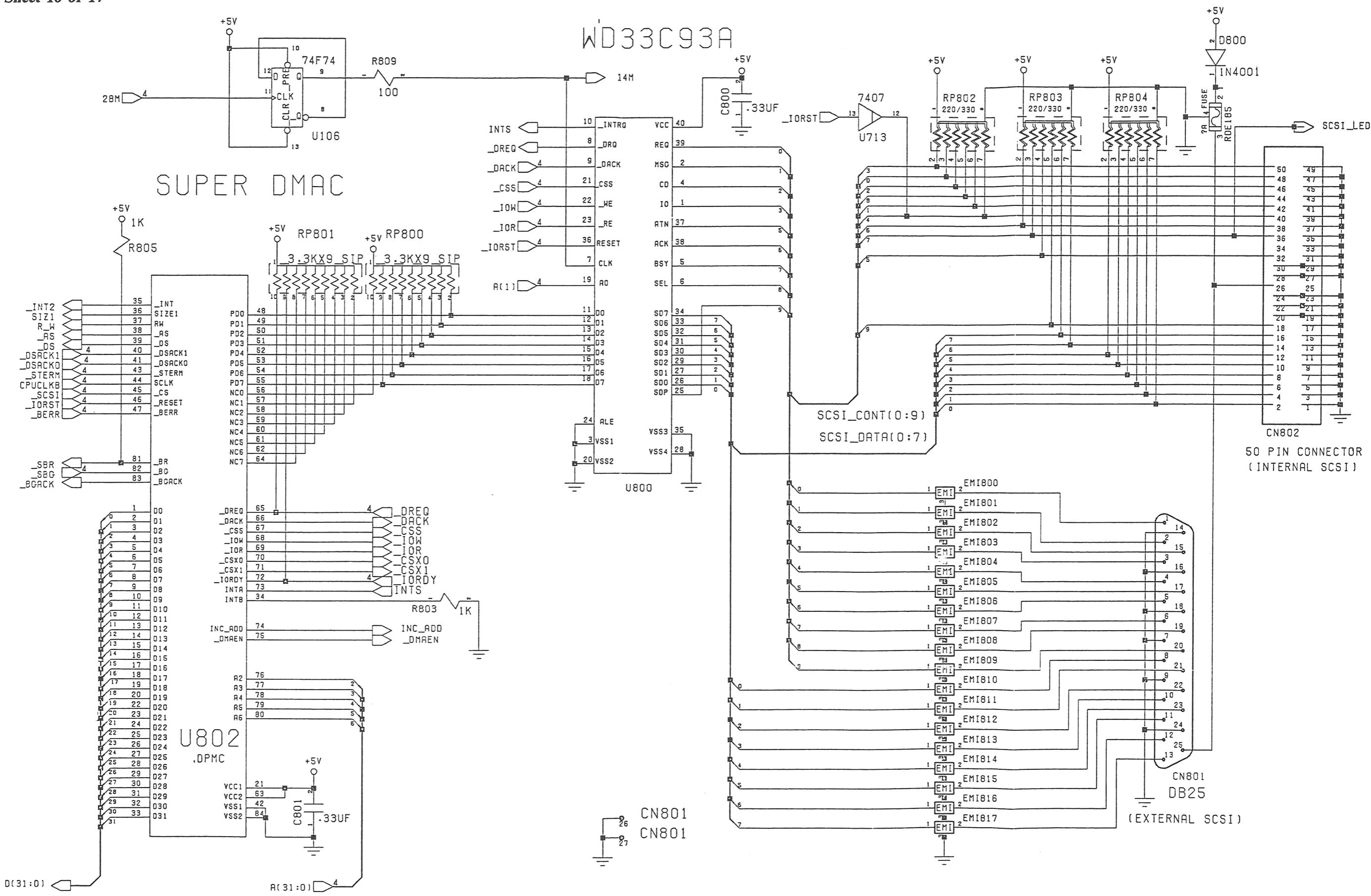
The timing diagram illustrates the logic levels of three signals over time. The horizontal axis represents time, and the vertical axis lists the signals:

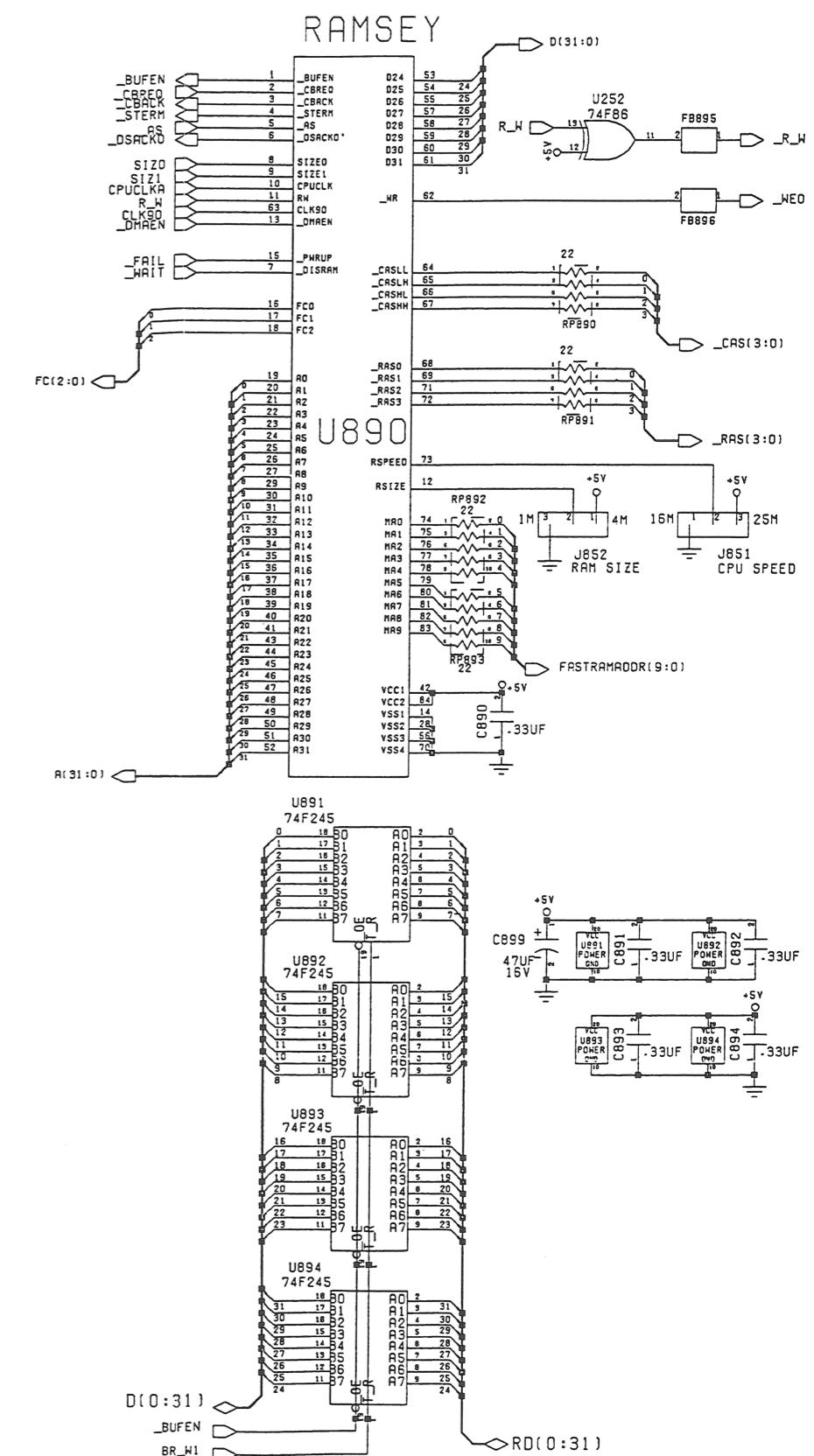
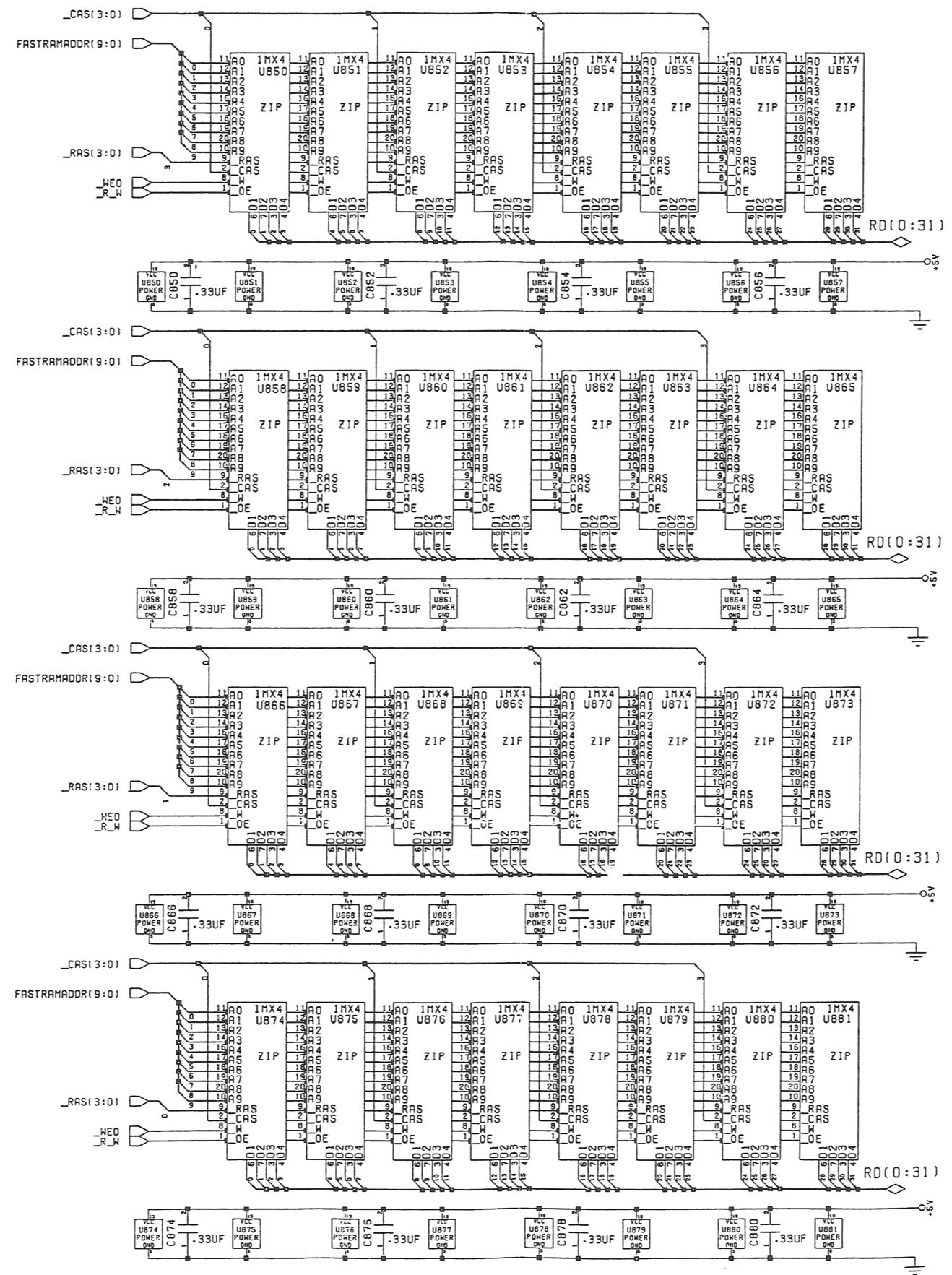
- FPU/RST**: A square wave signal starting at 1 and transitioning to 0 at time 0.
- EBCLR**: A pulse signal starting at 0 at time 0 and transitioning to 1 at time 1.0, returning to 0 at time 1.2, and transitioning back to 1 at time 1.4.
- EXT CPU**: A signal that starts at 0, remains low until time 1.0, then rises to 1, stays high until time 1.2, falls to 0 until time 1.4, and finally rises to 1 again.



IBM 16 BIT EXTENSION SLOTS









**This was brought to you
from the archives of**

<http://retro-commodore.eu>