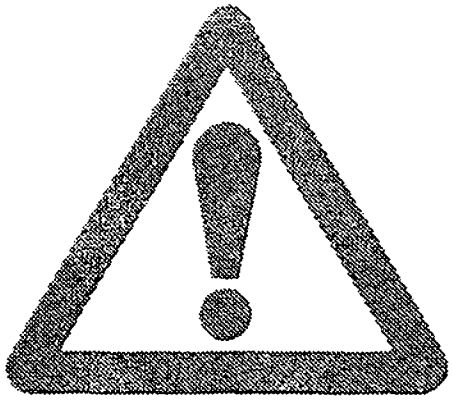
CMI SERIES III

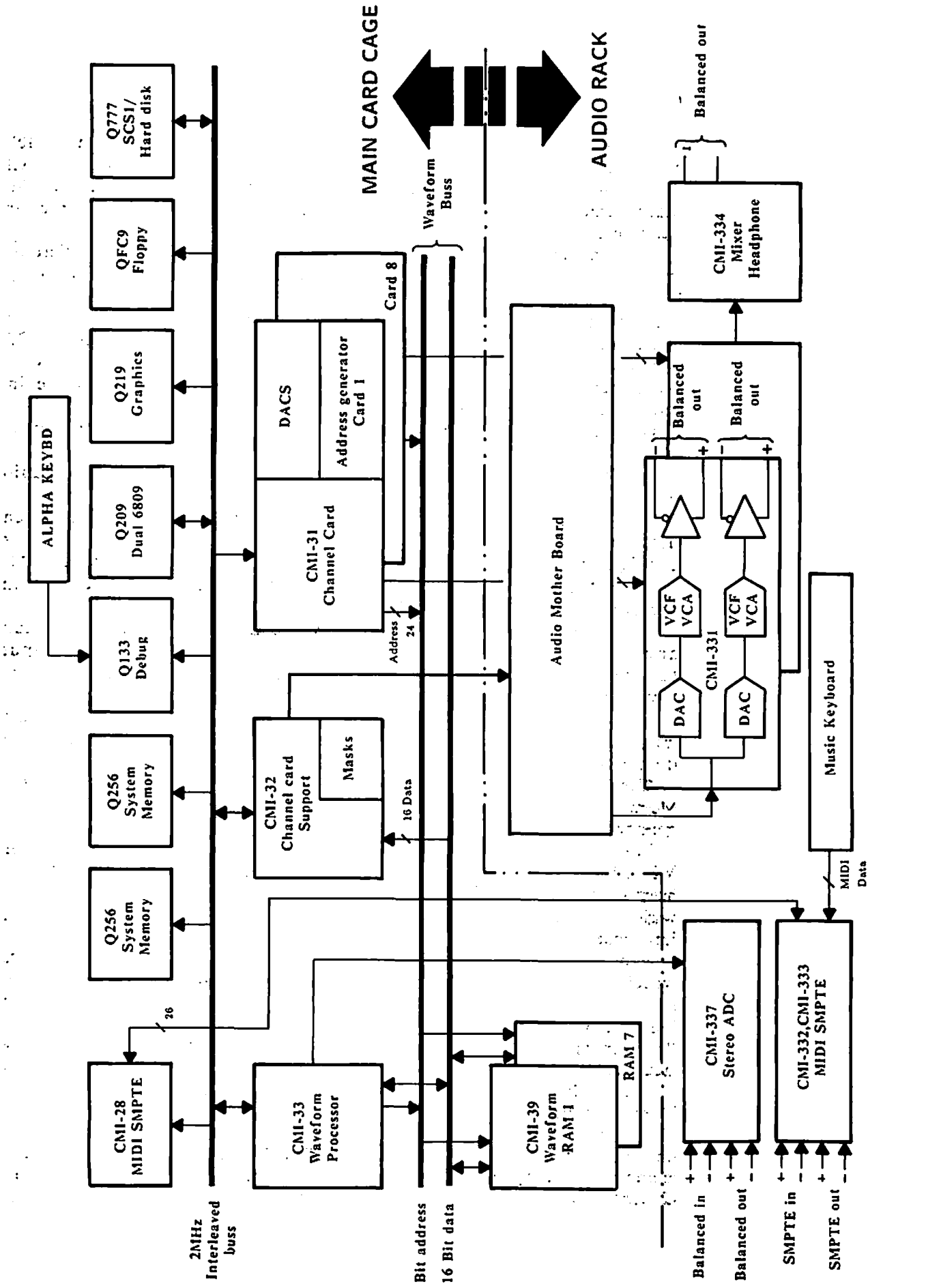
Service Manual



These service instructions are for use by qualified personnel.
Please contact Fairlight or your distributor
before attempting any repairs

Fairlight Pty Ltd., 15 Boundary Street, Rushcutters Bay, N.S.W. 2011
Sydney, Australia
Phone: (02) 331 6333, Telex: AA127998 'FAIRLT'

Contents


MAIN CARD CAGE
AUDIO RACK
Waveform Buss
Q777 SCSI/ Hard disk
QFC9 Floppy
Q219 Graphics
Q209 Dual 6809
ALPHA KEYBD
Q133 Debug
Q256 System Memory
Q256 System Memory
CMI-28 MIDI SMPTE
CMI-33 Waveform Processor
CMI-39 Waveform RAM 1
RAM 7
CMI-31 Channel Card
DACS
Address generator Card 1
Card 8
CMI-32 Channel card Support
Masks
Address 24
16 Data
Audio Mother Board
CMI-334 Mixer Headphone
Balanced out
Balanced out
Balanced out
CMI-331
VCF VCA
VCF VCA
DAC
DAC
Music Keyboard
MIDI Data
CMI-337 Stereo ADC
CMI-332,CMI-333 MIDI SMPTE
26
2MHz Interleaved buss
24 Bit address
16 Bit data
Balanced in
Balanced out
SMPTE in
SMPTE out

General Principles

(Refer to Figure 1)

The C.M.I. is a complex special-purpose computer system which embraces many different hardware and software technologies. All processing and sound generation functions are performed by the Mainframe, while the Graphics Terminal and Keyboards serve as peripherals for operator interfacing.

The mainframe is capable of operating quite autonomously, that is, it is not reliant on any external connections for proper functioning. Under certain conditions it is possible for a fault condition in external peripherals such as keyboards to inhibit proper main frame operation, so the serviceperson should be wary of being misled. Of course, without the peripherals connected it is often hard to know if the system is functioning properly, but this point should be borne in mind when trouble-shooting.

Operator input to the Mainframe comes from three sources: music keyboard, alphanumeric keyboard and graphics pad.

Output devices include the Graphics Display terminal and the audio outputs. A printer may also be used.

The heart of the system is the Central Processing Module, which uses two Motorola 6809 microprocessors in a dual-processor configuration. Both processors share a common buss called the CMI Buss or the CPU Buss which allows them both to communicate with the other processors in the Mainframe.

The Processor Control Module provides EPROM for system startup and bootstrap, RS-232C serial input from the keyboard, serial output to the keyboard and printer, and various other CPU support functions such as interrupt prioritisation.

Main program memory is provided by the 256K RAM card(s). This holds all the operational software, much of which is overlaid from disk as the code exceeds 256K.

The Floppy-disk and Hard disk (SCSI) controllers use Direct Memory Access (DMA) techniques to transfer data between main memory and the two floppy-disk drives.

The Graphics Display is a bit-mapped image of 16K bytes of VRAM. This is displayed as an array of 256 by 512 points. Special hardware provides support functions for automatic vector drawing, which considerably enhances the speed of displaying graphical information.

The digital section contains a second buss called the Waveform Buss which is entirely independent of the CPU buss and is dedicated to sound generation, manipulation and sampling. It has a 23 bit address buss, a 16 bit data buss, synchronous control lines and runs at 3.3 Mhz. It supports up to 14 megabytes of 16 bit waveform RAM. All RAM is accessible by all channel cards in cycles of 16 time slices. This allows 16 channels to run at a maximum of just under 200khz each. The Waveform Processor is the only device that can read and write to the Waveform RAM so it is responsible for loading, saving, sampling and manipulating sounds in waveform RAM. The channel cards only generate read cycles to get data from waveform RAM to the Audio Output modules. The actual waveform data read out of waveform

RAM in response to addresses generated by the channel cards do not return to the channel card but are picked up from the waveform buss by the Channel Support card and sent directly to the the Audio Motherboard and thence the Audio Output Modules. (see System block diagram)

Access to the waveform buss is prioritized in the order Channel cards, Waveform RAM refresh, then Waveform Processor. Channel cards are allowed access to the buss in a cyclic "round robin" manner. The channel card may or may not used its allocated time slice. Unused channel card time slices are allocated to refresh, then Waveform processor access.

The Waveform Processor and Channel cards reside on both CPU and Waveform busses so that they may be controlled by the CPU as well as accessing sounds in Waveform RAM.

The channel cards

Each channel card contains a 2Mhz 6809 with 64 kbytes of program memory and the circuitry to generate 2 channels of wave form pitch, level, filter setting, resonance and address information. The channel cards themselves contain no waveform memory.

Access to the waveform RAM is via fixed cycling allocation. Each channel card is connected to one Audio Module by a 26-way flat cable which carries the control voltages, pitch and clock information used to clock the 16 bit DACs, and control the VCAs and VCFs.

Channel Support Card

This contains the timing generation and time slice allocation logic for the channel cards and the waveform buss. It also contains the channel card addressing logic, channel masks, some channel card program RAM refresh logic and master pitch oscillator. The time slice logic generates 8 equal time slices that are subdivided internally to the channel cards, each channel card generates 2 channels 8 time slices apart.

e.g. channel card 1 generates channels 0 and 8, channel card 2 generates 1 and 9 etc.

The Channel Support card also contains a timer which has one output bussed to all channel cards where it will cause FIRQs (Fast Interrupt Requests). This real time clock is used for channel card envelope timing.

Waveform Processor

This is a 10MHz 68000 with 8k words of ROM, 256 k words of RAM, optically isolated serial interface to the ADC module, a waveform buss interface and a CPU buss interface. It also contains the Waveform Ram refresh arbitration logic. Access to waveform RAM is allowed during unused channel card time slices. 68000 code can run in waveform RAM if required.

Waveform RAM

Each Waveform RAM card contains 1M words of 16 bit waveform RAM. This RAM is refreshed by an on-board counter and logic on the Waveform Processor. Refresh cycles are only granted if no channel accesses are requested. An 8-bit mode is also supported for doubling sample time on low resolution samples.

General Interface Card

This board contains 4 MIDI in/out channels, SMPTE in/out, and synchronization clocks and clicks for the world beyond Fairlight. The music keyboard connects to this via one of the MIDI input channels. A 10MHz 68000 processor unscrambles control frames from key depresses and sends commands directly to the channel cards to play without intervention from the CPU. This processor also plays a major role in running music sequencers.

Audio Output Modules

These plug into the Audio Motherboard from the rear of the CMI Mainframe. Each contains 2 channels of 16 bit DAC, voltage controlled filter and voltage controlled amplifier and line driver. The waveform data come via a single flat cable from the Channel Support card to the Audio Motherboard then along the audio buss to each of the Audio Modules. Control clocks and voltages come directly from the channel cards via individual flat cables terminated in PC-mounting sockets on the Audio Motherboard. A switchable mix facility allows the two channels on each Audio module to be mixed.

Audio Mixer Module

The Mixer plugs into the Audio Motherboard from the rear of the CMI Mainframe and provides a single equally mixed output of all 16 channels to both a line socket and the headphone amplifier.

MIDI Support and SMPTE Support Modules

Both these modules plug into the Audio Motherboard from the rear of the machine. The SMPTE support module contains the analogue I/O circuitry required for SMPTE time code plus an electronic metronome which is controlled by data from the PIA output of the CPU Control Module and whose output is mixed onto the headphone amplifier.

The MIDI support module contains the optical isolation circuitry required for MIDI and provides mounting for the 5pin DIN connectors. Connections from both modules go to a socket on the Audio Motherboard which receives the flat cable connected at the other end to the General Interface Card.

Hardware/Software Relationships

This section gives a summary of the operational concepts involved in each of the CMI's major functions. This information should help relate a particular software function to the appropriate piece of hardware.

The software system is divided into two main sections, the resident software and overlays. The resident part is responsible for all the real-time functions such as sound generation, keyboard input processing and graphics pad operation. The overlays are used for the various control and sound manipulation functions provided by the display pages. Changing pages on the CMI loads a new overlay for that page from disk. Some pages use further overlays themselves, so that when certain functions are invoked from a particular page for the first time, a disk access will be made as the overlay is loaded.

Both 6809 processors access 65K bytes of program RAM, switched from the 256K memory board, so that some of the code may be executed by each processor individually, and both processors can share common data structures. Processor 2 carries out the non real-time functions such as disk I/O and graphics display.

The 68000 Waveform Processor is concerned with movement of data into and out of waveform memory and manipulation of data in waveform memory.

The 68000 MIDI Processor is concerned with starting and stopping audio channels.

A broad description of a range of specified functions follows.

System Startup/Boot

When power is first applied to the system, a power-on reset signal is generated for about a half second by a timer located on the Processor Control card, Q133. At the end of this time, both processors fetch restart vectors from EPROMS, also on the Q133 card and start executing the startup procedure in EPROM. Processor 1 initialises all the registers of the peripheral controller devices such as PIAs and ACIAs Processor 2 initialises the Graphics Display, clears the screen, loads disk driver firmware into system RAM from ROMs on the QFC - 9 and/or Q - 777 controller modules, and displays the LOAD SYSTEM DISK IN DRIVE greeting. While this is happening the processor internal to the Music keyboard also starts up. The Music keyboard LED display first diplays the message POWER ON, and then the message SERIES III. Processor 1 then loops, waiting to be triggered by Processor 2, which in turn loops waiting for a disk to be inserted in drive 0, as indicated by the appropriate status bit from the Floppy-disk Controller Card QFC9.

When the system disk has been correctly inserted, processor 2 executes the first stage of the bootstrap loader firmware (located on the Q133 card). This involves reading in the boot block, which is a special sector on the system disk. The code stored in the boot block is then executed which completes the boot load by loading the operating system and the Page 1 overlay. When Page 1 starts up, the message PAGE 1 READY is sent to the music keyboard display.

Disk Operations

The CMI uses one eight-inch double-sided floppy disk drive and one or two 5.25" hard disc drives.

Floppy disk

Floppy disk format is soft sectored, 128 bytes per sector (single density), or 256 bytes per sector (double density). FM recording is used for extra reliability. The floppy drive itself is controlled by a Western Digital WD1791 L.S.I. controller located on the Floppy Disk Controller Card QFC9.

The Floppy disk driver EPROM is located on the QFC9 card. Routines in this EPROM provides utilities including read sector, write sector, and verify C.R.C. which are called by the RAM-resident disk-operating system.

In the event of a disk error being detected during a read or write operation, the software will perform a number of re-tries, including head relocation, to try to recover from the error. If the error persists, an error message is displayed.

Hard disk

The hard disk is controlled by the Q777 SCSI TSmall Computer Systems Interface) card. The hard disk driver firmware is located on the Q777 card. Hard disks are 85 Meabyte or 140 Megabyte, expandable as required.

Graphics Display, Graphics Pad

The graphics display is generated by writing a bit-mapped image to the dedicated 16K byte VRAM. This block of RAM is mapped in and out of the processor memory space under software control. The graphics pad sends special format ASCII characters through the same path as normal ASCII characters from the alphanumeric keyboard. These are then converted into graphics coordinates, and VRAM addresses, in software, which also generates the graphics pad cursor.

Command entry

Data arriving from the alpanumeric keyboard is fed to the ACIA on the Q133 Processor control card. Alphanumeric characters are passed to Processor P2. They are then processed by the OS9 Operating System.

Loading/Saving Sounds

Sounds are stored on hard disk. Each voice file occupies up to 14 megabytes of disk space. The Voice files and other user files are stored in the directory /CMIF/CMIFILES. When a file is loaded, the directory is searched and the address of the file found. The Voice is then DMA'd into the system memory under the control of Processor 2, and DMA'd from the system memory through to the waveform buss via the waveform processor. *Saving sounds to disk operates by the reverse process.*

Sound Sampling

Audio input for sampling is fed to either or both of the Right and Left Line Inputs at the rear of the CMI III. The CMI - 337 Sample card performs the analogue to digital conversion. There is a single analogue to digital conversion circuit, which switches between left and right channels during stereo conversions. The sampling rate is a

maximum of 100KHz in mono mode and 50 kHz in stereo mode. The sample rate is governed by the frequency of a pulse stream coming from the Channel Card in channel one position. The sample rate is therefore established by software which sets up channel one to operate at the sampling frequency specified on the Sample Page.

As each digital conversion is made, the digital sample is passed serially to the waveform processor via a 10 way cable, and thence to waveform memory. When the **Sample** command is issued, the waveform processor starts conversions and loops until the data read is of a greater absolute value than the number specified as **Trigger Level**. It then begins transferring data to the waveform RAM, until the number of samples made is equal to the number specified as **Sample Number**.

Playing Music from the Keyboard

Three byte MIDI frames are transmitted serially from the CMI music keyboard to the Port D ACIA on the CMI - 332 MIDI Support Module. At present Fairlight keyboard MIDI is confined to MIDI Channel 1. When other MIDI keyboards are used they can be configured to transmit through channels 1 to 16 of Ports A, B and C on the CMI - 332. MIDI frames then pass to the CMI - 28 General Interface card. The CMI - 28 is concerned with starting and stopping output channels. When MIDI data comes directly from the music keyboard, Processors 1 & 2 are not involved.

Music Keyboard Functions

As well as sending music key depression/release data to the mainframe, the music keyboard has a number of ancillary functions.

A multiplexed analog-to-digital converter samples the level of the three faders and two control wheels on the left-hand end of the keyboard as well as the three pedal inputs on the rear. Whenever one of these changes its level by more than a certain amount, a packet of MIDI control data is transmitted to the Mainframe giving the device number and the new level.

The two switches on the left of the keyboard and the three switches which plug into the rear of the keyboard are also scanned, and when any of these are opened or closed, suitable MIDI data is sent to the Mainframe.

Pressing a key on the numeric keypad on the right-hand end of the keyboard sends a character to the Mainframe in exactly the same way as an alphanumeric key depression.

The alphanumeric LED display on the music keyboard is driven by the serial link coming from the Mainframe. The processor in the keyboard controls the displaying of individual characters as well as <rubout> and <clear>. When messages longer than the 12 digits of the display are required, a horizontal scrolling routine in the CMI system software is used.

Playing Programmed Music Sequences

When programmed sequences are played, P1 and P2 are involved. Data is read from disk and converted into MIDI frames and timing information. The MIDI data is fed into the MIDI input queue of the CMI - 28, and output under the control of the timing information.

ELECTRICAL

Power Requirements
- Mains Voltage: 100-120 or 200-250 switch selectable
- Mains Current: 2 amps @ 240V, 4 amps @ 120v
- Mains Frequency: 50/60 Hz

AUDIO

Channel Outputs
- Connector type: Cannon XLR 3 pin (balanced)
- Number of channels: 16 (maximum per mainframe)
- Output level: +4dBm
- Output impedance: 600 ohms
- Output load: Greater than 600 ohms

Mixed Line Output
- Connector type: Cannon XLR 3 pin (balanced)
- Output level: +4dBm with one channel playing
  Switchable to -20dB attenuation.
- Output impedance: 600 ohms
- Output load: Greater than 600 ohms

Headphone Output
- Connector type: 1/4" Stereo Phones
- Amplifier: Stereo 500mW
- Signal: from mixer output.

Click Input
- Connector type: Cannon XLR 3 pin
- Level: 1 volt (min) to 20 volt (max) p-p
- Frequency range: 200 Hz to 8KHz
- Impedance: 10k ohms

Click Output
- Connector type: Cannon XLR 3 pin
- Output signal : 5 volt square clock

SMPTE Input
- Connector type: Cannon XLR 3 pin
- Level: -20 to +10dBm
- Speed: 24 fps to 30 fps
- Impedance: 10k ohms

SMPTE Output
- Connector type: Cannon XLR 3 pin
- Output signal : +15dBm
- Impedance: 33 ohms

MIDI Inputs
- MIDI standard opto-coupler receivers.
- MIDI standard 5-pin DIN sockets

MIDI Outputs
MIDI standard open collector current loop drivers
MIDI standard 5-pin DIN sockets

Multi-Sync Outputs
Four sync outputs, pins 1,3,4 and 5 of the DIN socket
TTL open collector signals

Clock or Drum-machine Controller Output
TTL open collector signals
Compatible with drum machines such as Roland
Pin 1; Run/Halt Pin 2; Earth Pin 3; Clock
Pin 4; Reset/Start Pin 5; n/c

Sampling Inputs
Connector type: Cannon XLR 3 pin
Input signal: Balanced
Sensitivity: -18 dBm required for full scale conversion

DIGITAL

Processors: Dual 6809 CPU
68000 Waveform Processor
68000 General Interface Processor
8 6809 Channel processors

Memory: 512K bytes CPU system RAM
16K Video RAM
512K bytes Waveform Processor Private RAM
64K bytes program RAM on each channel card
16K bytes General Interface Private RAM
2-14M bytes Waveform RAM

Floppy Disk: Mitsubishi M2896-63
8 inch double sided, single/double density
Soft sectored, 128/256 bytes per sector

Hard Disk: 70M or 140M byte (unformatted) expandable as required
SCSI bus compatible

Graphics Display: Bit mapped VRAM 512 x 256 pixels
Composite video output
1 volt p-p nominal
75 ohms impedance

Input/Output: Serial RS232C, 9600 Baud plus MIDI

MECHANICAL

Dimensions: Width 750 mm
Depth 480 mm
Height 345 mm

Weight: 45 kilograms, depending on optioning

Introduction

Presented here is a list of cable connections and their allocated pins so that users may wire up their own cables. external cable connections only are shown, internal cables are presented elsewhere in this manual.

Cable schedule, Power Connections

The CMI is equipped with a standard IEC type three pin mains input power connector. Two pins are for active and neutral 100 to 240 volts AC only with the third being the earth or ground connection.

For feeding external devices a socket version of the IEC type connector is connected to the load side of the CMI mains switch. This provides up to 150VA of switched mains power.

SCSI Connector

The CMI comes equiped with an industry standard 50 way connector for devices on the SCSI bus.

VDU Socket

The VDU socket is a 5 pin 'Belling Lee' type L1904A socket. This feeds the VDU with power and video signals.

Pin no.	Function
1	16 volts AC power supply
2	16 volts AC power supply
3	no connection
4	video signal to monitor
5	video ground

Keyboard Power

The music kayboard receives its power supply from the CMI mainframe via this connector. A 'CANNON' type 7 pin socket is used with the pins allocated as follows:

Pin No.	Function
1	10 volts supply return
2	10 volts supply return
3	+10 volts DC supply
4	+10 volts DC supply
5	-20 volts DC supply
6	20 volts supply return
7	+20 volts supply

Printer 1 and Printer 2

These sockets are for external connection of printers to the CMI. They are industry standard 'DB25S' type connectors.

Pin no.	Function
1	Protective or chassis ground
2	Data out
5	Clear To Send
7	signal ground
20	Data Terminal Ready

Keyboard Connector
The CMI music and alpha-numeric keyboards connect up via an industry standard 'DB9S' type connector.

Pin No.	Function
1	+20 volts supply for alpha KBD only
2	MIDI from music keyboard +
3	-20 volts supply for alpha KBD only
4	MIDI from music keyboard -
5	Ground
6	RS-232C data to CMI
7	Ground
8	Protective ground
9	Keyboard data from CMI

MIDI Inputs
The MIDI inputs are a standard 5 pin 180 degree DIN type socket.

Pin No.	Function
1	no connection
2	no connection
3	no connection
4	MIDI in +
5	MIDI in -

MIDI Outputs
The MIDI outputs are a standard 5 pin 180 degree DIN type socket.

Pin No.	Function
1	no connection
2	ground
3	no connection
4	MIDI out +
5	MIDI out -

SYNC OUT
The SYNC OUT socket provides 4 sync outputs.

Pin No.	Function
1	sync out 1
2	ground
3	clock out
4	sync out 2
5	sync out 3

CLOCK out
The CLOCK output socket is a 5 pin 180 degree DIN type connector.

Pin No.	Function
1	Run/Stop
2	ground
3	clock
4	Reset/start
5	no connection

SMPTE IN
This socket is an 'XLR type three pin socket.

Pin No.	Function
1	ground
2	SMPTE in -
3	SMPTE in +

SMPTE out
This connector is a panel mount 'XLR' type three pin plug.

Pin No.	Function
1	ground
2	SMPTE out -
3	SMPTE out +

METRONOME
This is a panel mount three pin 'XLR' type plug.

Pin No.	Function
1	ground
2	ground
3	metronome out

CLICK in
A three pin 'XLR' type socket is used for Click Track input.

Pin No.	Function
1	ground
2	no connection
3	Click Track in

CLICK out
An 'XLR' type three pin panel mount plug is used to connect Click Track out to external devices.

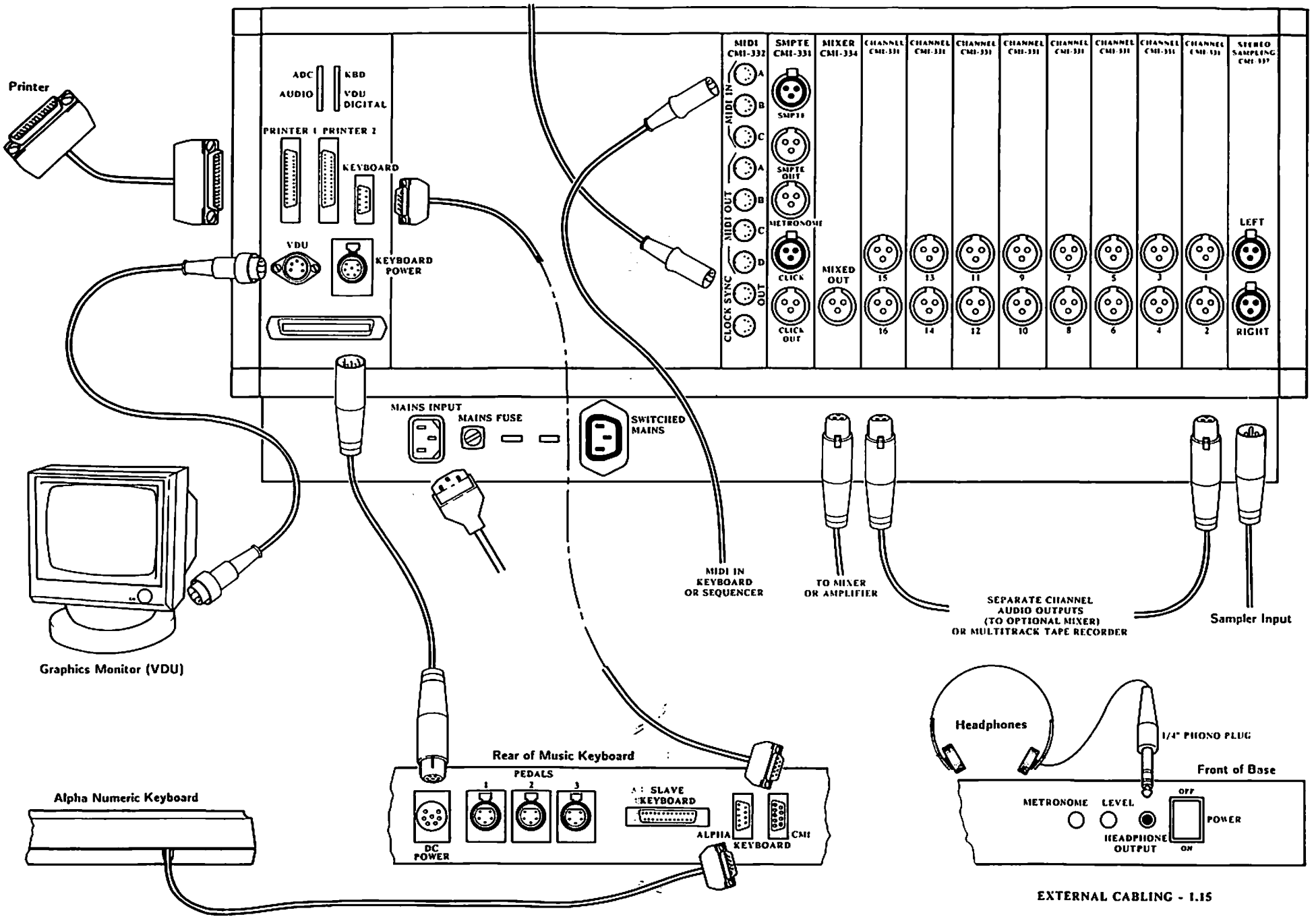
Pin No.	Function
1	ground
2	ground
3	Click Track out

MIXER And AUDIO Outputs
The Audio and Mixer outputs are connected via 'XLR' type three pin panel mounted plugs.

Pin No.	Function
1	ground
2	Output -
3	Output +

SAMPLER Input
The Sampler connects via three pin 'XLR' type sockets.

Pin No.	Function
1	ground (floating)
2	Audio Input -
3	Audio Input +



EXTERNAL CABLING - 1.15

Digital Card Cage 2

Contents

Q209 2.2

Dual 6809 Central Processor

Introduction

The Q209 contains the dual 6809 processors , on board processor communication hardware entailing indivisible instructions, processor readable identification / map state , interprocessor interrupts, automatic map switching FUSE register and hardware trace logic to enable single stepping for software debugging.

The Dual Processor card multiplexes each processor onto a common address and data buss in an interleaved manner, each processor therefore may simultaneously access the same memory location without any contention, if the memory is mapped onto both processors. (See Q256 functional description)

The memory addresses are issued to the buss 225 nanoseconds prior to the access cycle, allowing addresses to be mapped by the memory card, to allow for accessing greater than 64K of RAM.

Many global timing signals are issued from the processor for general buss control .

Master Timing Signals

(refer schematic Q209-00)

All system timing signals are derived from crystal-controlled 40MHz oscillator Q1. Flip-flop 10F derives two opposite phase 20 MHz square waves. Quad D-type latch D2, together with the NAND gate in 7F, forms a 10 state Johnson, or twisted-tail ring counter. Each state is of 50ns duration. The system signals are decoded by NAND gates in 8E from the output of this counter.

Dynamic Memory Timing Signals

(refer schematic Q209-00)

Four non-inverting buffers of 10A are driven by latch E1 to provide CAS(Column Address Strobe), RAS (Row Address Strobe), CA (Column Address , active low) and RA (Row Address, active low). RAS is delayed relative to CAS by about 20ns by the propagation delay of 11E. RA and CA are complementary.

Data and Address Buss Multiplexing

(refer schematic Q209-03)

Flip-flop 6F , along with associated gating, generates the 6809's E signals. The system address buss is multiplexed by the ADDRESS signals ADD1 and ADD2 , (active low). One-of-four decoders 3E and 4E are used to enable the appropriate address and data buffers, to perform the multiplexing. The data buffer enable signals WRITE1, WRITE2, READ1, READ2 are generated by logical combinations of R/W , VMA , processor phase 2 and DMA lines. The address buss is actually multiplexed 4 ways, as the vectored interupt system may also acquire the buss' least significant bits of the address buss for either processor's vector fetch cycle. The address buffer enables are a function of the Address signal and the Interupt acknowledge.

Phase 2 reference and Address references for each Processor are feed to the buss via buss drivers.

Interrupt Strobe Generation
(refer schematic Q209-02)

Dual D-type flip-flop 9D and 3-input AND gate 8D feed Interrupt Strobe pulses to the buss. These are used by the Priority Interrupt Control Units (PICUs) used to provide vectored interrupts, and also to strobe the vector address latches 8A and 9A. The PICUs are located on the Q133 card. These signals strobe the priority latches continuously, until an interrupt is acknowledged. In this way the Interrupt Priority is maintained at its latest level regardless of delay between an interrupt request being received by the PICU and the associated vector-fetch cycle being executed.

Direct Memory Access
(refer to drawing Q209-00)

DMA requests for each processor are clocked into flip-flop 11D on the falling edge of the phase 2 signal of the respective processor. DMA acknowledge is sent to the buss via buffers and drive signals to the processors are suspended in the phase 1 state for the duration of the DMA cycle. The maximum permissible DMA duration is 5 microseconds. Worst-case DMA latency is 1 microsecond. Latency is the time required to service the request.

Vector-Fetch Decoders
(refer to drawing Q209-01)

The vector state of the processors are decoded by the one-of-four decoders, 2D and NOR gates in 1D. These correspond to addresses in the range FFF0 to FFFF. They correspond to the processor fetching vectors FIRQ, NMI, SWI1, SWI2, SWI3, IRQ and RESTART. The Restart vectors come from ROM so when this is sensed the ROM is enabled and the ram disabled. This is achieved by the ROMEN signal on buss pin 44. On detection of an Interrupt Request vector address from the processor, decoder 2D causes the normal address buss drivers for bits 1 to 4 to be disabled and the Interrupt Address buffers to be enabled in lieu.

Processor System Control

These general functions are controlled through ports at the following locations ...

$FC5E	Indivisible instructions	read
$FC5E	Various CPU functions	write
$FC5F	Map status and CPU ID	read
$FC5F	Automatic map switching FUSE	write

The ID can be read to determine the status of the memory map switching hardware. The CPU ID bit can be read by the CPU to find out which CPU is running the program.
The bits are defined as

D0	CPU ID 0=P1 1=P2
D1	P1 map status 0=map B, 1=map A
D2	P2 map status
D3	zero
D4	n/c (indeterminate)
D5	n/c
D6	n/c
D7	n/c

The "Various CPU functions" is an 8 bit register in which each bit may be independently written to. This register is at location 6D, and it is decoded by devices at 9B and 7A. When written to, the bit address is selected by the 3 least significant bits of the data byte. The state of data bit 3 determines whether the bit is set or cleared.

The four functions provided per processor from this register are:-

0+P interprocessor interrupt
2+P hardware trace
4+P map switch select
6+P fast interrupt request

where P is "0" for processor 1 and "1" for processor 2

Fast interrupts may be generated either by an external signal or from the bus. The on-card FIRQ must be reset by the processor concerned, by writing a reset bit to the register.

Automatic Map Switching

(refer schematic Q209-02)

The memory cards support hardware selectable memory maps. The processors can control the A/$\overline{B}$ select lines, allowing automatic switching between "user(B)" and "system(A)" maps.

Whenever an interrupt or processor restart occurs , the A map will be automatically selected. During an interrupt, the switching will occur after the registers are stacked and before the interrupt vector is fetched.

A FUSE location is provided which causes the map to be switched after a specified number of CPU clock cycles have elapsed. These counters are at 8C for CPU1 and 9C for CPU2. The data written to these counters is buffered by the buffer at 7C. The map changed to is determined by the value of the map switch select bit. The map switch will occur after the Nth CPU cycle after the FUSE register write. The delay is required so that a known number of instructions can be executed for house keeping before the CPU's memory is swapped.

Hardware also selects the A map whenever DMA occurs.

Hardware Trace

An NMI may be generated after each instruction execution for software debugging. This is done by flip-flops 5D and half of AND gate 4F. This function is enabled under software, by access to $FC5E.

Enabling this function inverts the NMI signal from the front panel, so that if the trace hardware is left in the triggered state, front panel NMI requests will still be recognized, but on the opposite edge (since NMI is an edge triggered input).

Indivisible Instructions

(refer to schematic Q209-00)

For the test-and-set and double byte load/store instructions to be effective, the processor not executing the instruction must not be able to alter the flag memory location in question. To this end, the execution of these read/modify/write instructions effectively hangs the other processor for its duration, thus preventing race conditions .

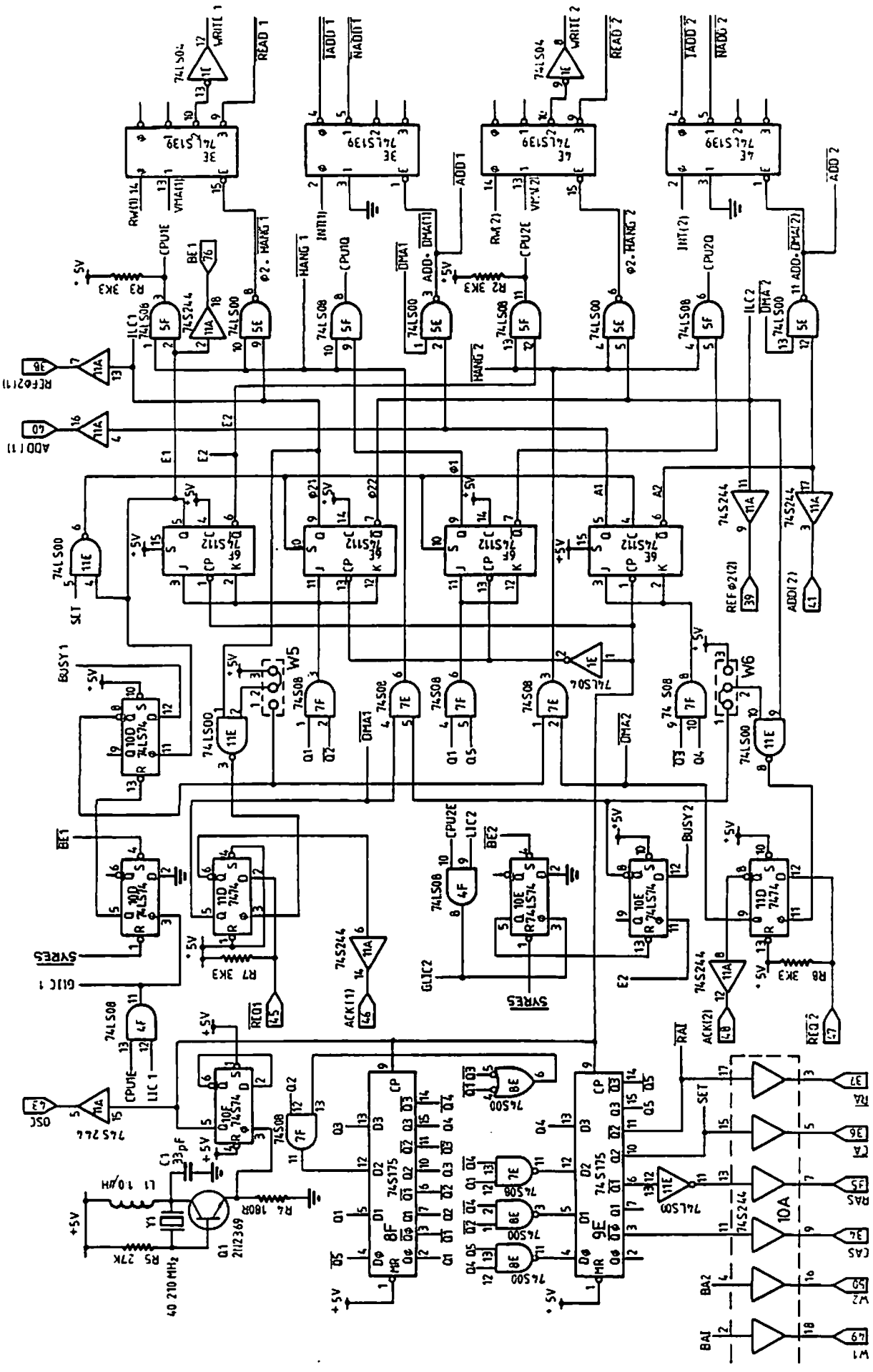
This is achieved by flip-flops 10D, 10E, associated gating and the BUSY outputs from the processors. The BUSY outputs are active when the test and set instruction is executed , and the other processors clock is stopped for its cycle, in the same manner as for DMA transfers. The flip-flops asociated with this are reset at power on to enable the clocks to the 6809's to allow them to be internally reset, at power on reset. To enable this function the instruction to be made indivisible must be immediately preceded by a read from hardware location $FC5E. No interrupt must be allowed to occur between the read and the instruction. This function is automatically disabled at the end of the instruction following the read.

Link Options

The links have the following functions:

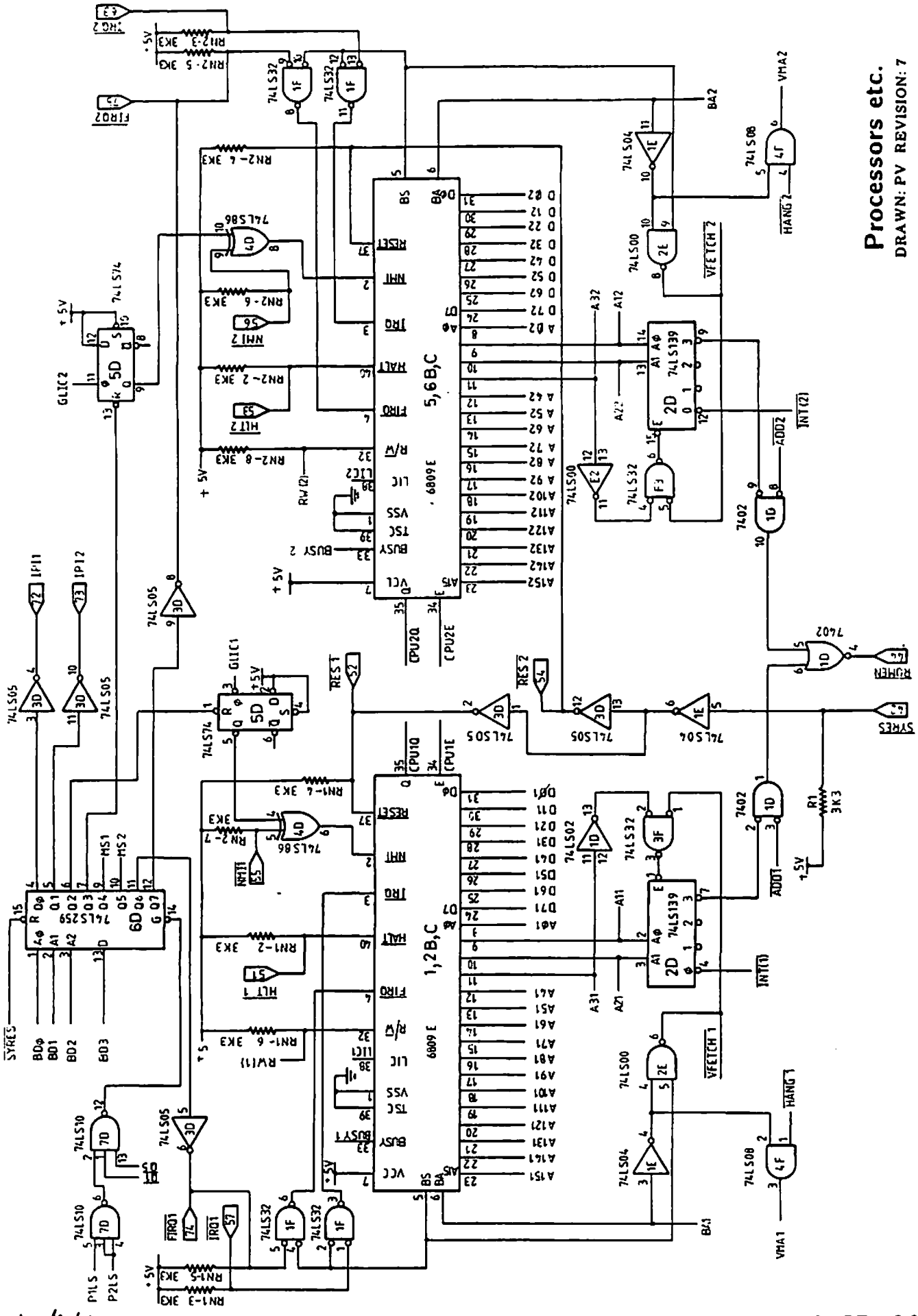
Option	CPU	LINK	Function
W1	P1	1-2 *	enable map select output
		2-3	disable
W2	P1	1-2 *	enable DMA to select A map
		2-3	disable
W3	P2	1-2 *	enable map select output
		2-3	disable
W4	P2	1-2 *	enable DMA to select A map
		2-3	disable
W5	P1	1-2	disable P1 DMA during indivisible P2 cycles
		2-3 *	enable
W6	P2	1-2 *	disable P2 DMA during indivisible P1 cycles
		2-3	enable

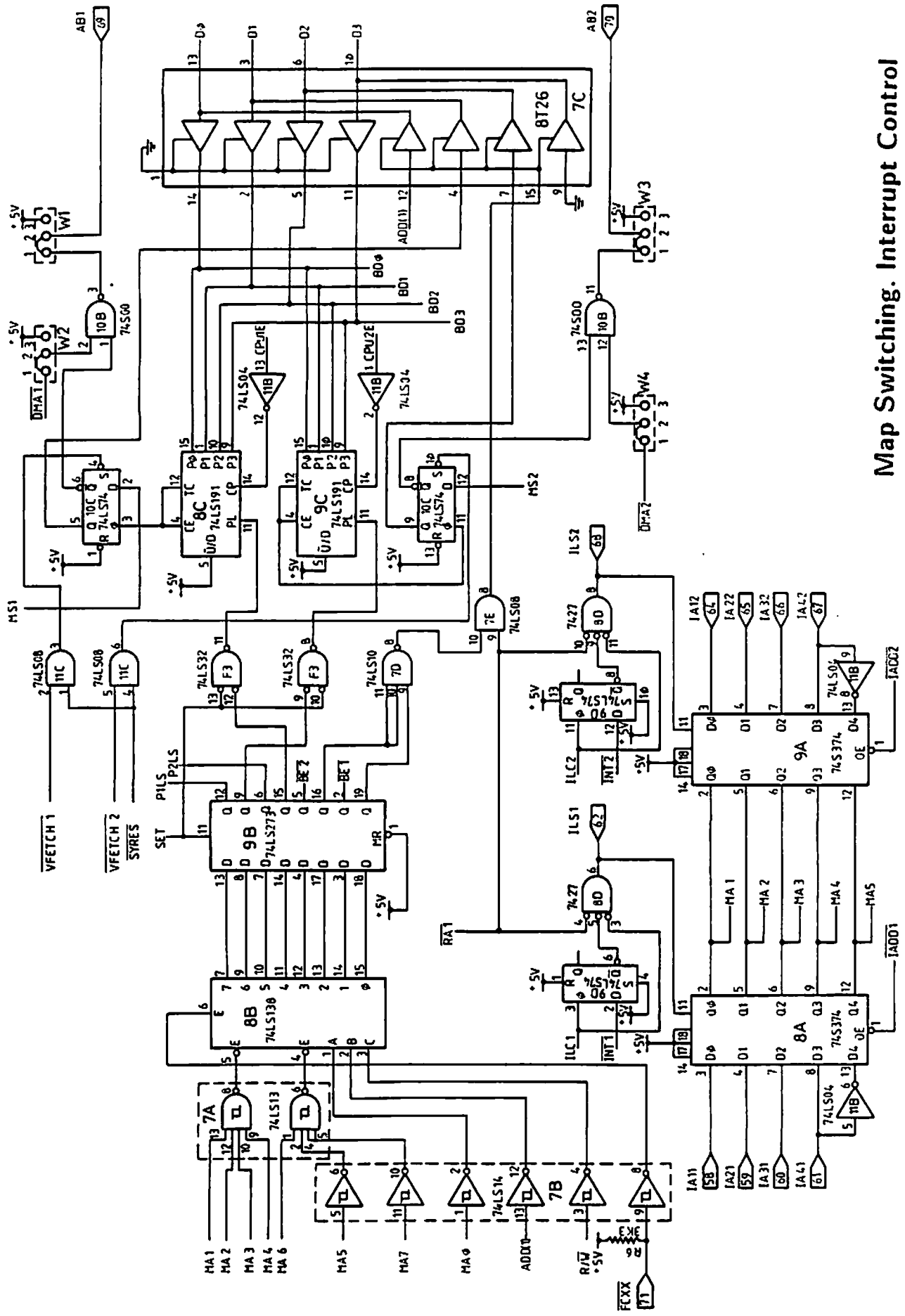
These links are set by PCB traces in the positions marked by * . Links

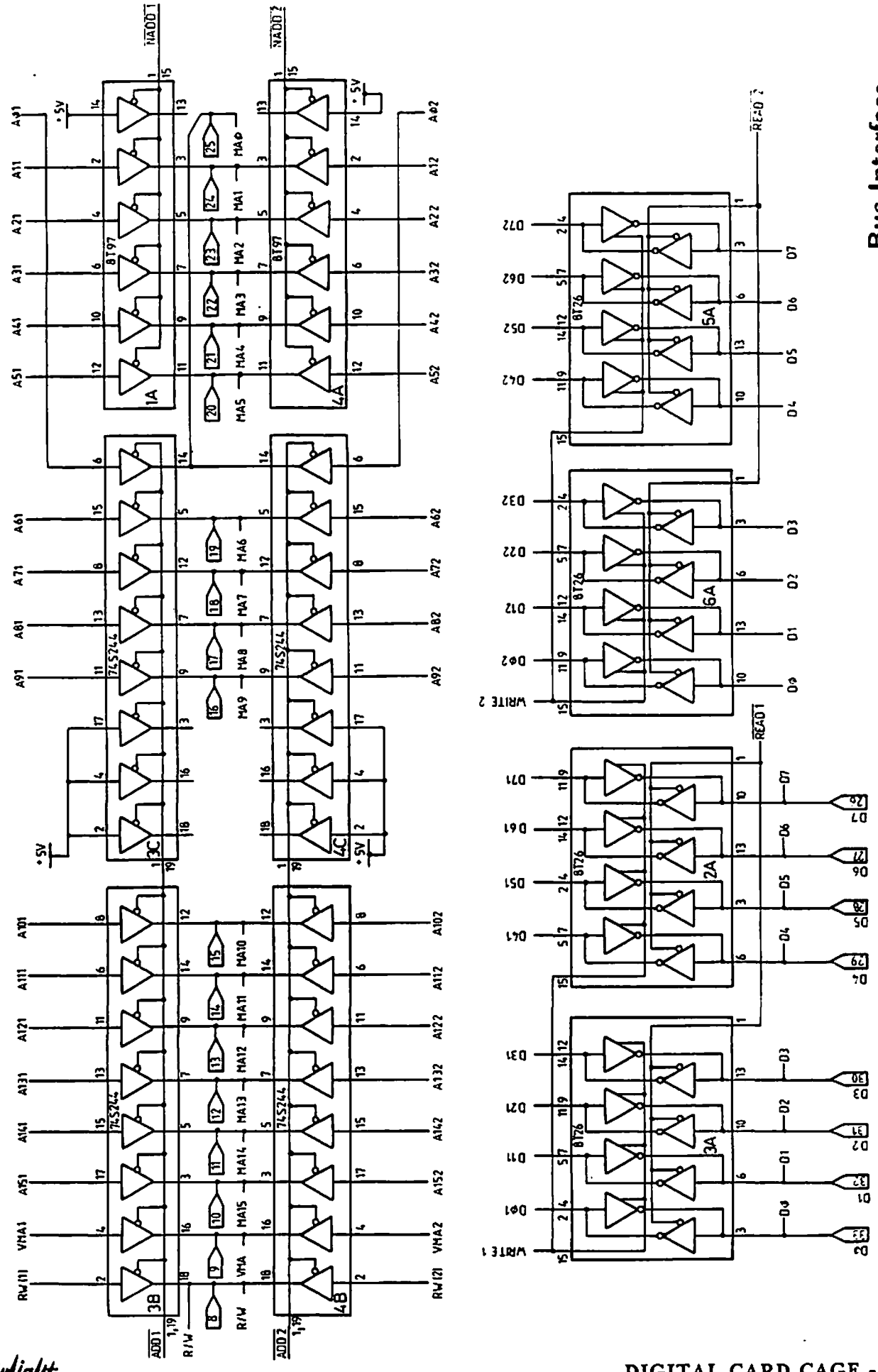


Clock and Timing Generation, DMA Control

DRAWN: PV REVISION: 7


Processors etc.
DRAWN: PV REVISION: 7
5,6B,C
1,2B,C
6809E
74LS139
74LS259
74LS74
74LS86
74LS32
74LS04
74LS08
74LS00
74LS10
74LS05
74LS02
7402
RN2-4 3K3
RN2-5 3K3
RN2-6 3K3
RN2-2 3K3
RN2-8 3K3
RN2-7 3K3
RN1-4 3K3
RN1-2 3K3
RN1-6 3K3
RN1-5 3K3
RN1-3 3K3
R1 3K3
+5V
VMA2
VMA1
HANG 2
HANG 1
VFETCH 2
VFETCH 1
INT(2)
INT(1)
ADD2
ADD1
RQMEN
SYNES
BA2
BA1
A32
A12
A31
A21
A11
RW(2)
RW(1)
BUSY 2
BUSY 1
CPU2Q
CPU2E
CPU1Q
CPU1E
RES 1
RES 2
NMI 2
NMI 1
HLT 2
HLT 1
FIRQ2
IRQ2
FIRQ1
IRQ1
BD0
BD1
BD2
BD3
MS1
MS2
IP11
IP12
P1LS
P2LS


Map Switching. Interrupt Control
DRAWN: PV REVISION: 7


Bus Interface
DRAWN: PV REVISION: 7
NADD1
NADD2
ADD1
ADD2
R/W
READ
WRITE 1
WRITE 2
74LS244
8T26
8T97
1A
2A
3A
4A
5A
6A
3B
4B
3C
4C
+5V

Q133 CPU Control Card 2.3

Introduction

The CPU Control Card provides several support functions required by the CPU card. These include startup and bootstrap ROM, 4 serial communication ports, interrupt prioritisation, dynamic RAM refresh, day/date/time of day clock, P1 DMA daisy chain, and a parallel port.

Address Map

The Debug Card occupies the last 4K bytes of the 65K byte memory addressing space and is set up as follows:-

ADDRESS (HEX)	FUNCTION
F000-F7FF	Rom0 common rom
F800-FBFF	Rom1 processor unique
FC00-FCEF	Available for peripherals
FC80-FC8F	ACIA registers
FC90-FC97	Timer (6840)
FCF0-FCFF	PIA registers, user and clock
FCFC	CPU#1 interrupt prioritiser
FCFD	CPU#2 interrupt prioritiser
FD00-FEFF	Shared 512 byte RAM
FF00-FFFF	Unique 256 byte RAM for each processor

Restart and Interrupt Vectors

RAM space allocated uniquely to each processor provides independent restart and interrupt vectoring. The vector locations are as follows:

ADDRESS (HEX)	VECTOR
FFFE/F	Restart
FFFC/D	NMI
FFFA/B	SWI1
FFF8/9	Unused
FFF6/7	FIRQ
FFF4/5	SWI2
FFF2/3	SWI3
FFF0/1	Unused
FFEE/F (lowest)	IRQ level 7
FFEC/D	IRQ level 6
FFEA/B	IRQ level 5
FFE8/9	IRQ level 4
FFE6/7	IRQ level 3
FFE4/5	IRQ level 2
FFE2/3	IRQ level 1
FFE0/1 (highest)	IRQ level 0 (highest)

Debug Monitor

The Q133 contains two 2K ROMs that contain all the basic driver and initialization routines, such as loading the Disc drivers and Q256's maprams.

The monitor ROM occupies 1K bytes from F000 to F3FF and may be accessed by either processor. Processor-unique workspace RAM is used by the monitor so both processors can be executing the monitor independently.

Commands	
/	Reopen last open address as a 1-byte unit
AAAA	Open 2-byte unit at address AAAA
	Reopen last open address as a 2-byte unit
$A	Open CPU accumulator A
$B	Open CPU accumulator B
$X	Open CPU index register X
$P	Open CPU program counter
$H	Open user SWI handler address
$C	Open CPU Condition Code register
$D	Open CPU D register (A,B concatenated)
$Y	Open CPU index register Y
$U	Open CPU User Stack pointer U
$S	Open CPU Stack pointer S
$R	Open program segment Relocation Register
$G	Open CPU direct page register
$F	Open monitor flag byte
<return>	Close the open location
<linefeed>	Close current, open next location
v	Close current, open previous location
>	Close current, take branch offset and open
@	Open location pointed by current location
AAAA;B	Insert a breakpoint at address AAAA
;L	List all active breakpoints
AAAA;D	Delete breakpoint at address AAAA
;C	Clear all breakpoints
AAAA;T	Insert tracepoint at location AAAA (non-stopping breakpoint)
AAAA;K	Kill tracepoint or breakpoint at AAAA
AAAA;G	Start a user program at address AAAA
;P	Proceed from breakpoint, abort, or call
AAAA;O	Calculate branch offset from open location to address AAAA
HH;F	Fill memory from beginning address to end address
BEG ADDR	User prompt for beginning address
END ADDR	User prompt for end address
<CTRL X>	Abort current command line, take no action
<	Close current location, return to sequence start and open
AAAA,R	Relocate address AAAA by register R. R may be any of the CPU registers, the user relocation register, the monitor flag byte or the currently open location
AAAA.	Relocate address AAAA by Relocation Register $R
:	Same as linefeed (CTRL J) except that no new line is taken, and neither the address nor contents of the next location is displayed
AAAA#LL	Memory dump of LL lines (16 bytes/line) starting from, address AAAA
'<ASCII chr>	Input ASCII character value instead of hex value for any of the above commands

The 6809 monitor will also accept input of signed hex numbers.

System Boot/Disk ROM

This ROM is used by CPU#2 for disk booting operations and occupies locations F800 to FBFF in the unique ROM space for CPU#2.

The following functions calls are provided:-

* Boot load QDOS operating system from disk
* Initialise disk controller
* Read full last sector
* Read partial last sector
* Read verify (CRC check only)
* Write and verify CRC
* Restore head (seek track 0)
* Seek to specified track
* Write test
* Write D.D. mark to sector
* Write sectors and verify CRC
* Write sectors and don't verify CRC
* Check and abort if non-recoverable error

This ROM contains the code to load the actual disk drivers into system RAM. The driver routines themselves are stored in RAM after being loaded from the ROM on the QFC9 floppy controller card, the Q077/Q087 Hard Disk card if present and the Q777 SCSI Controller card.

Address Decoding

(refer schematic Q133-00)

The System Address Buss is buffered by non-inverting buffers A1 and A4. NAND gate B1 generates an output (asserted LOW) when an address in the range FXXX is detected. This is fed out to the buss on edge connector pin 60B. Further decoding by combinational logic at B4, D4, B3, C2 and C1 generate select signals for the two EPROMS, $\overline{\text{ROM0}}$, $\overline{\text{ROM1}}$.

Selection of the on-card static RAM and peripheral devices in the FCFX range are also decoded.

These six select signals are latched by hex flip-flop B6. Hex flip-flops C7 and D7 latch the 11 low-order address bits, as well as the READ/WRITE signal. When any of the on-board devices are read from, inverting data buss transceiver A5 drives the buss. (See drawing Q133-03). At other times, A5 buffers the data into the card.

RAM Refresh Control

Rate multiplier C10 is configured to produce a 1 microsecond pulse every 16 microseconds. This output generates a DMA request for Processor 1 ($\overline{\text{RDMA}}$), via DMA hardware at C8, B10 ,B5 and C4. The refresh has the highest priority in the P1 DMA daisy chain.

The $\overline{\text{ENL}}$ signal, (Enable Next Level), indicates to the next device along the daisy chain when it may make DMA requests. It normally goes low every second P1 cycle, but if a refresh request is pending, the low pulse is inhibited.

When this request is acknowledged by the ACK1 buss signal from the Q209 CPU (asserted HIGH) flip-flop B10 generates a /REF (Refresh, asserted LOW) signal on the buss, which signals a refresh cycle to the dynamic RAMs in the system. At the same time, the output of the refresh address counter A2 is driven onto the buss by tri-state buffers A3. At the completion of the refresh (DMA) cycle, the refresh address counter is incremented ready for the next cycle.

Static RAM

(refer schematic Q133-01)

A small amount of static RAM is provided for use as scratchpad during disk calls and monitor firmware execution. It is organised as follows:

CPU #1 FF00-FFFD
CPU #2 FF00-FFFD
Both FD00-FEFF

The addressing function for this purpose is generated by multiplexer C9 which is driven by an OR funtion of address bits 8 and 9. The RAM itself is in the form of two 1K X 4 devices at D8 and D9.

EPROM

Four kilobytes of U.V. erasable ROM are used. These are 2716s/2516 single 5 volt supply type.

Their functions are:

Location	Address Range	CPU #	Function
D2	F800-FBFF	1	Startup
D4	F800-FBFF	2	Disk boot
D5	F400-F7FF	Both	I/O functions
D6	F000-F3FF	Both	Debug monitor

ACIAS (Asynchronous Communications Interface Adapter)

(refer schematic Q133-02)

6551 ACIAs atE4, E5, E1, E2, E3 are used to receive and transmit serial data. The BAUD rate is determined internally via internal dividers, from the baud-rate generator master 1.8432MHz oscilator at D1.

Interrupts generated by the ACIAs go to the system buss via pin 68B of the edge connector.

Data input and output level conversion for the RS232 standard is provided by circuitry on Sheet 3.

The 6551 used for keyboard data is at location E2, 3. The ACIAs at E4 and E5 have optional RS422 tranceivers at F7 and F8 as well as RS232, at F6 and F5. The 555 timer at D10 is used in the RS422 buss timeout control.

PIA (Peripheral Interface Adapters)

(refer schematic Q133-01)

PIA (F10,11,12) is used to provide two general purpose parallel ports. Peripheral connections are made through a 26-way ribbon cable connector on the front of the card.

Interrupts from the PIA are presented to the buss via pins 66B and 67B.

PIA (E9,10,11) is used to interface the clock/calender chip at E12. This clock has a 3.7 volt Lithium cell to maintain the time when the computer is turned off. The battery is not rechargeable and must be replaced when flat. Battery life is approximately 3 years.

Diodes CR5 and CR4 isolate the battery from the 5 volt supply, so that the battery is only connected to the clock when the 5 volt supply drops. Transistors Q2, Q1 on drawing Q133-03, and associated components interface the PIA's signal levels to the clock and control the power-down function of the clock so that no false writes occur at power-on and off. An optically isolated power down signal is available at connector pins 61B and 62B, from the opto-isolator at A11.

Manual Controls

Restart, halt and interrupt controls are provided on the front-panel card Q137. The sole use is for system debugging. In normal use all signals from the Q137 are inactive.

Activating either HALT switch on the front panel sends /HLT1 or /HLT2 to the corresponding processor on the Q209 CPU. When halted, the Buss Available signals from the CPU card W1 and W2 drive open-collector buffers B12 to turn on the WAIT LEDs on the card.

The system can run without a front panel being connected.

Power-on Reset

555 Timer A12 is used to generate a system-reset signal on power-up or manual restart from the front panel console, if restart is enabled on both processors. This is a low-going pulse of about 500 milliseconds on buss pin 42.

Interrupt Priority Logic

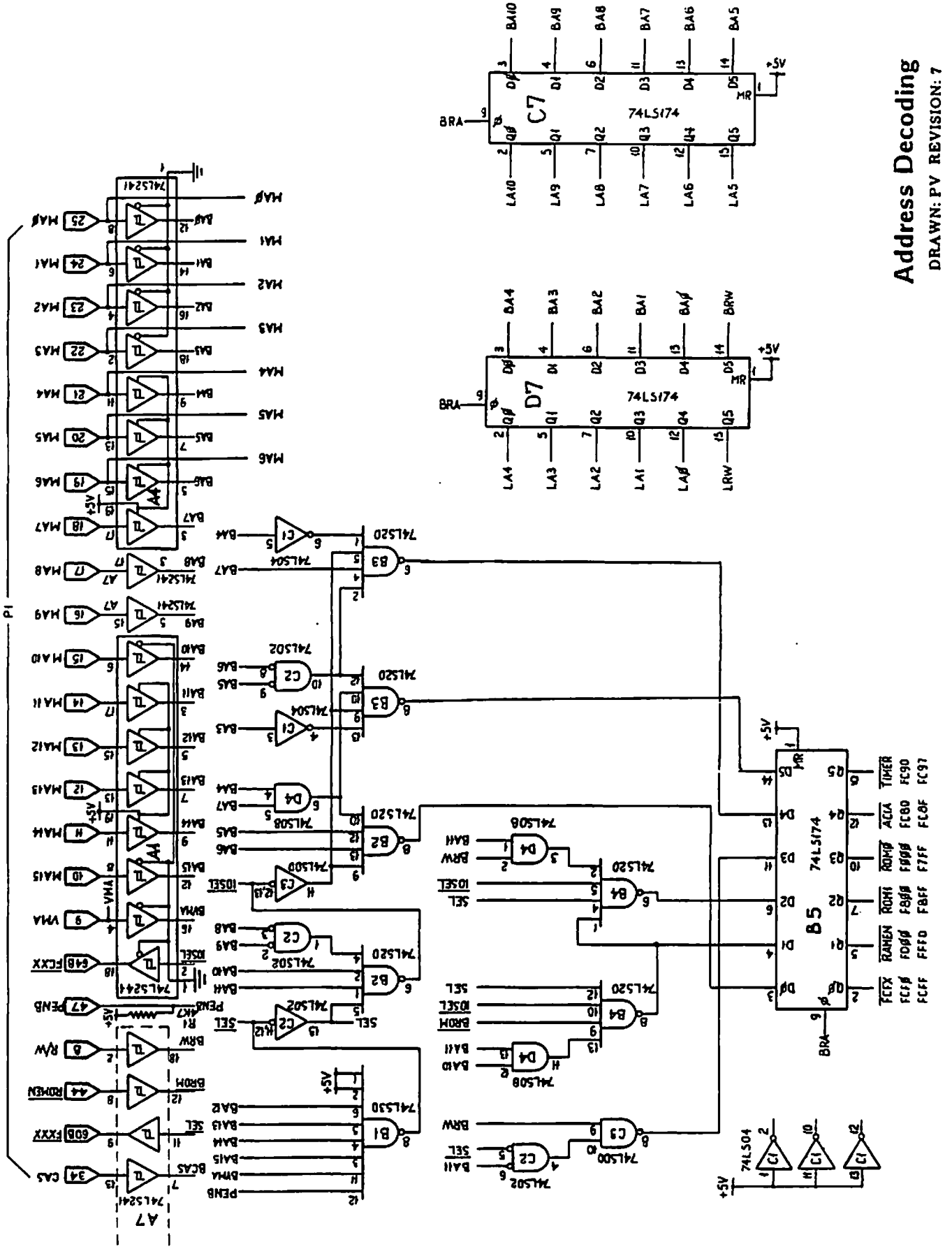
8214 Priority Interrupt Control Units (PICU) are used to latch interrupt requests and generate a priority level which is used by the CPU card to create an interrupt vector address. Each processor has its own PICU.

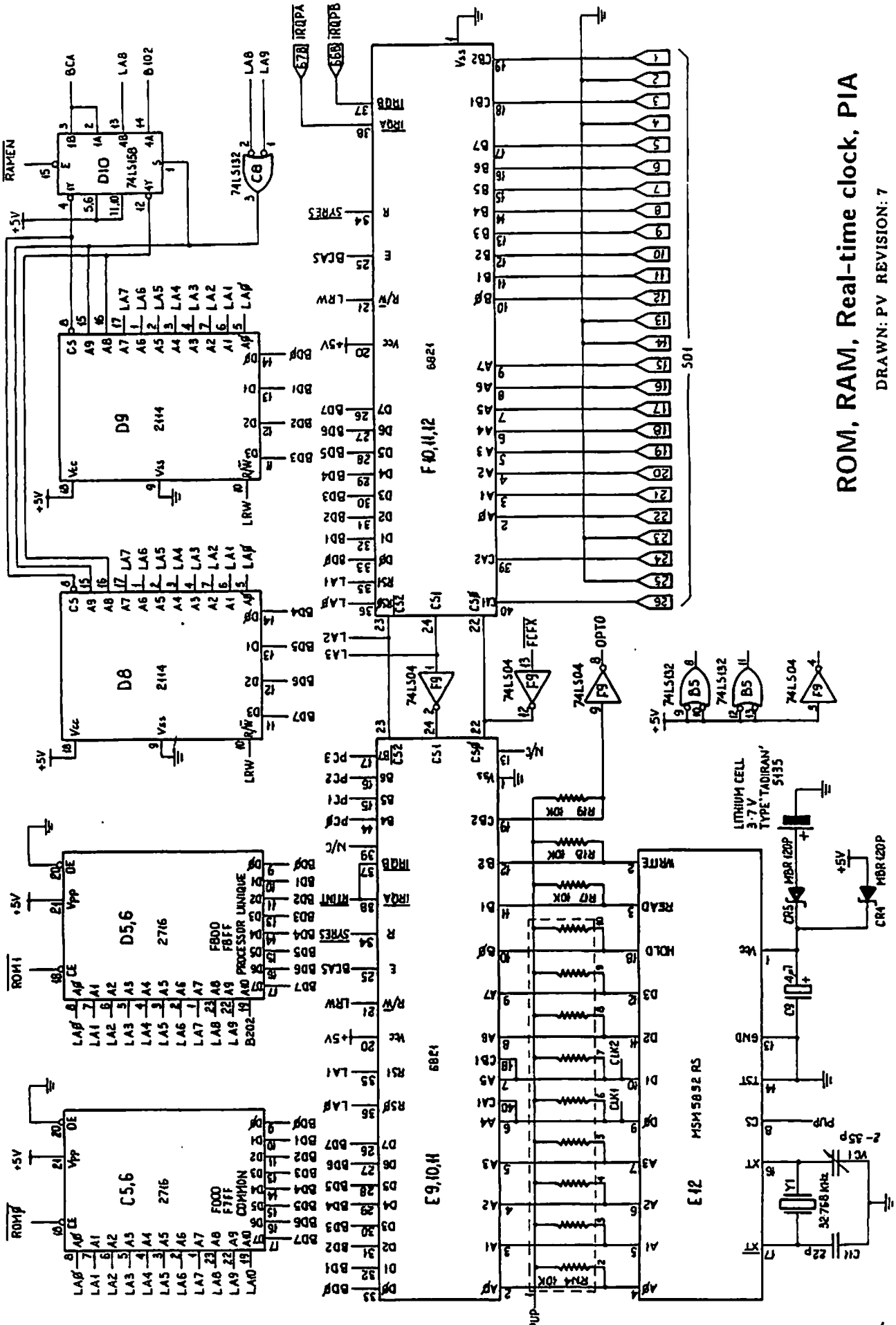
The priority level for each PICU is established by writing the complement of the desired priority level into the status register. The address for CPU 1 is FCFD, for CPU 2 it is FCFC. Decoding for this purpose is performed by one-of-eight selector B8.

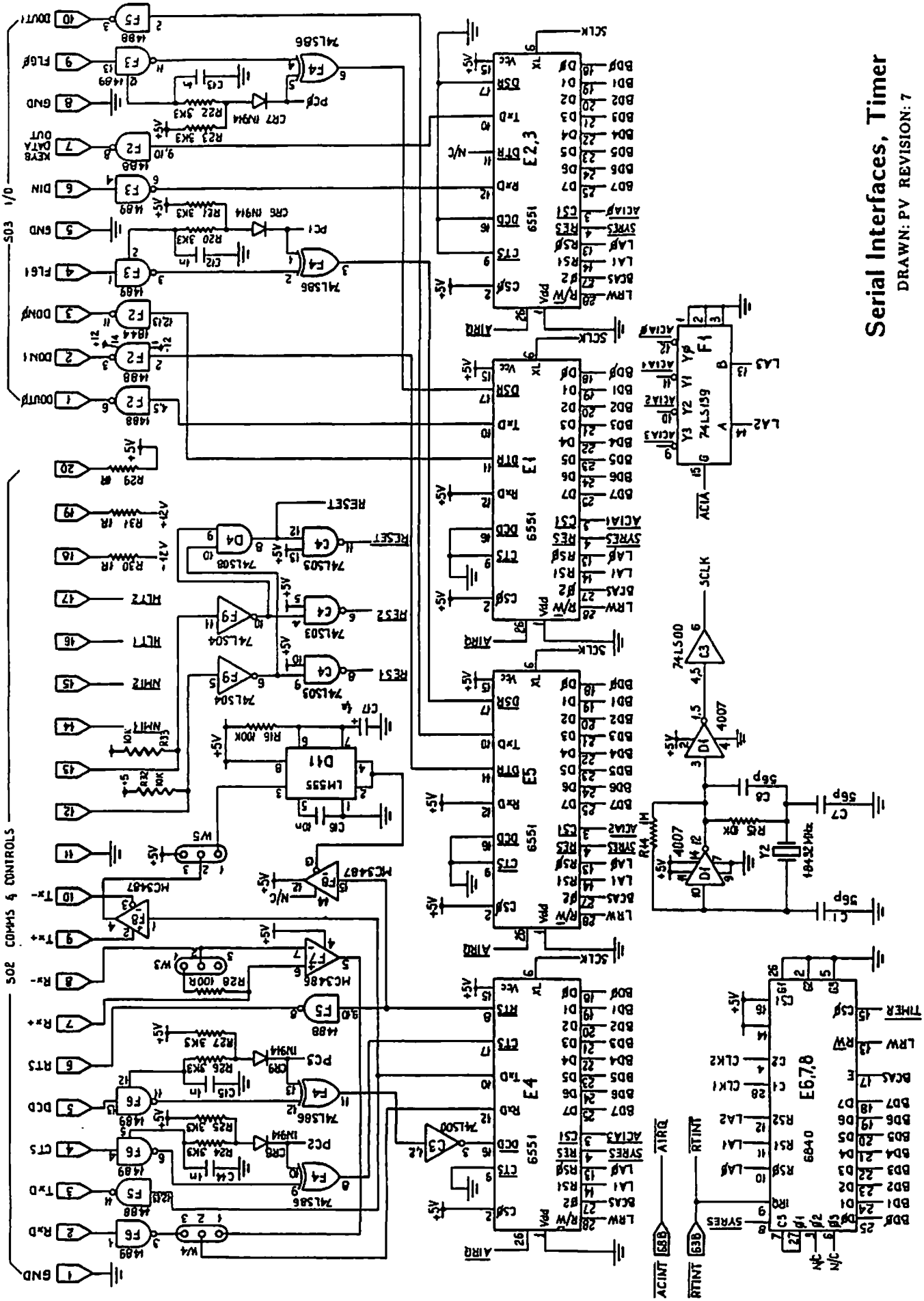
Interrupt requests generated by the PICU are latched by flip-flops B9, which are reset when the PICUs are written to to establish the new priority level mask.

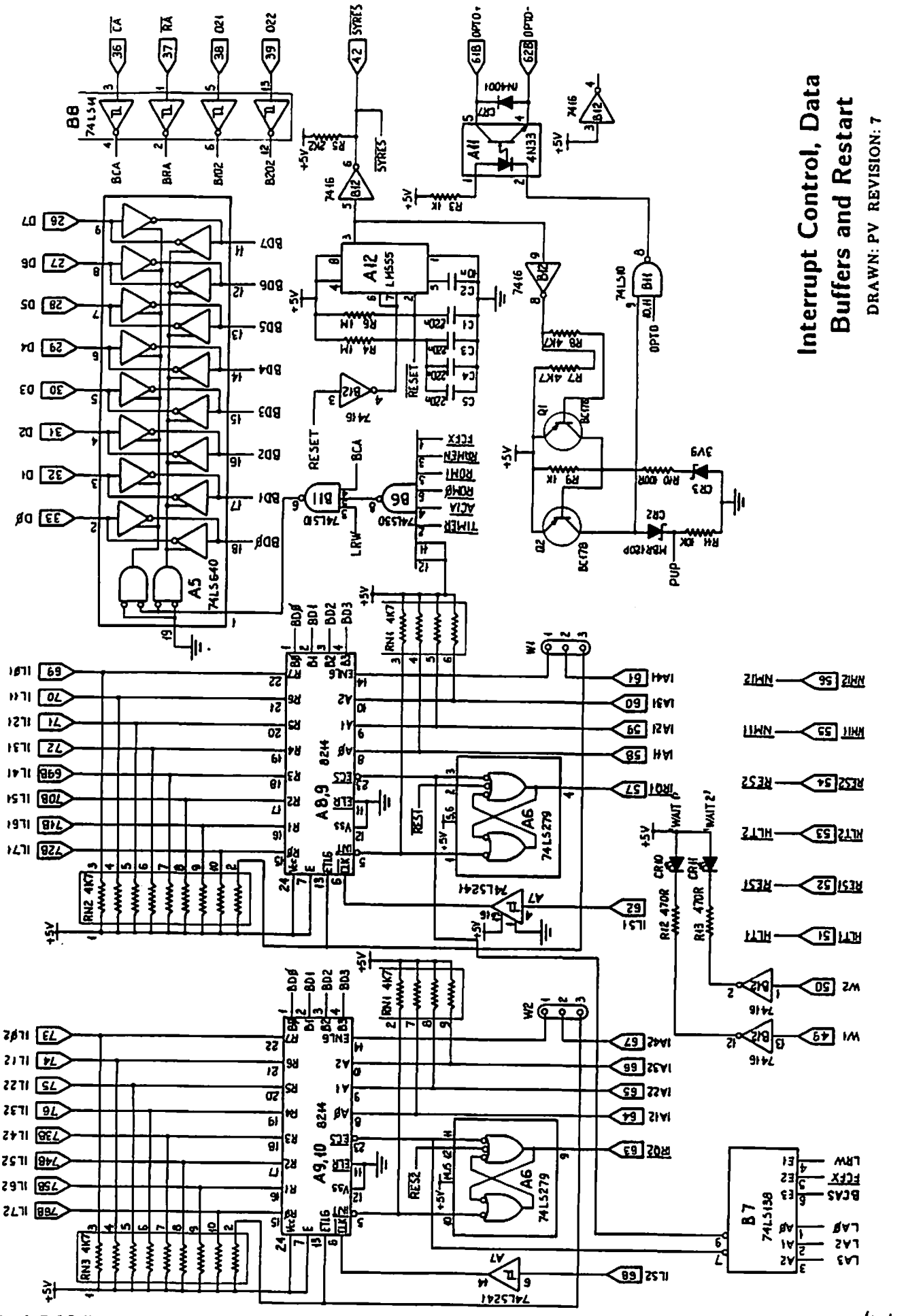
The PICUs are clocked by Interrupt Latch Strobe signals from the bus (ILS1 and ILS2).

Each PICU supports up to eight levels of interrupt.


Address Decoding
DRAWN: PV REVISION: 7
C7
D7
74LS174
74LS20
74LS04
74LS08
74LS02
74LS30
74LS241
74LS244
74LS240
74LS04
B5
A7
A4
A1
P1
+5V
BRA
MR
BA10
BA9
BA8
BA7
BA6
BA5
BA4
BA3
BA2
BA1
BA0
BRW
LA10
LA9
LA8
LA7
LA6
LA5
LA4
LA3
LA2
LA1
LA0
LRW
MA0
MA1
MA2
MA3
MA4
MA5
MA6
MA7
MA8
MA9
MA10
MA11
MA12
MA13
MA14
MA15
VMA
FCXX
PENB
R/W
ROMEN
FXXX
CAS
SEL
BCAS
BRW
BROM
BMA
BA11
BA12
BA13
BA14
BA15
IOSEL
RAMEN
ROMP
ACIA
TIMER
FC90
FC97
FC9F
FCFF
FFFD
FBFF
FBØØ
FØØØ
FCFX
FCFØ

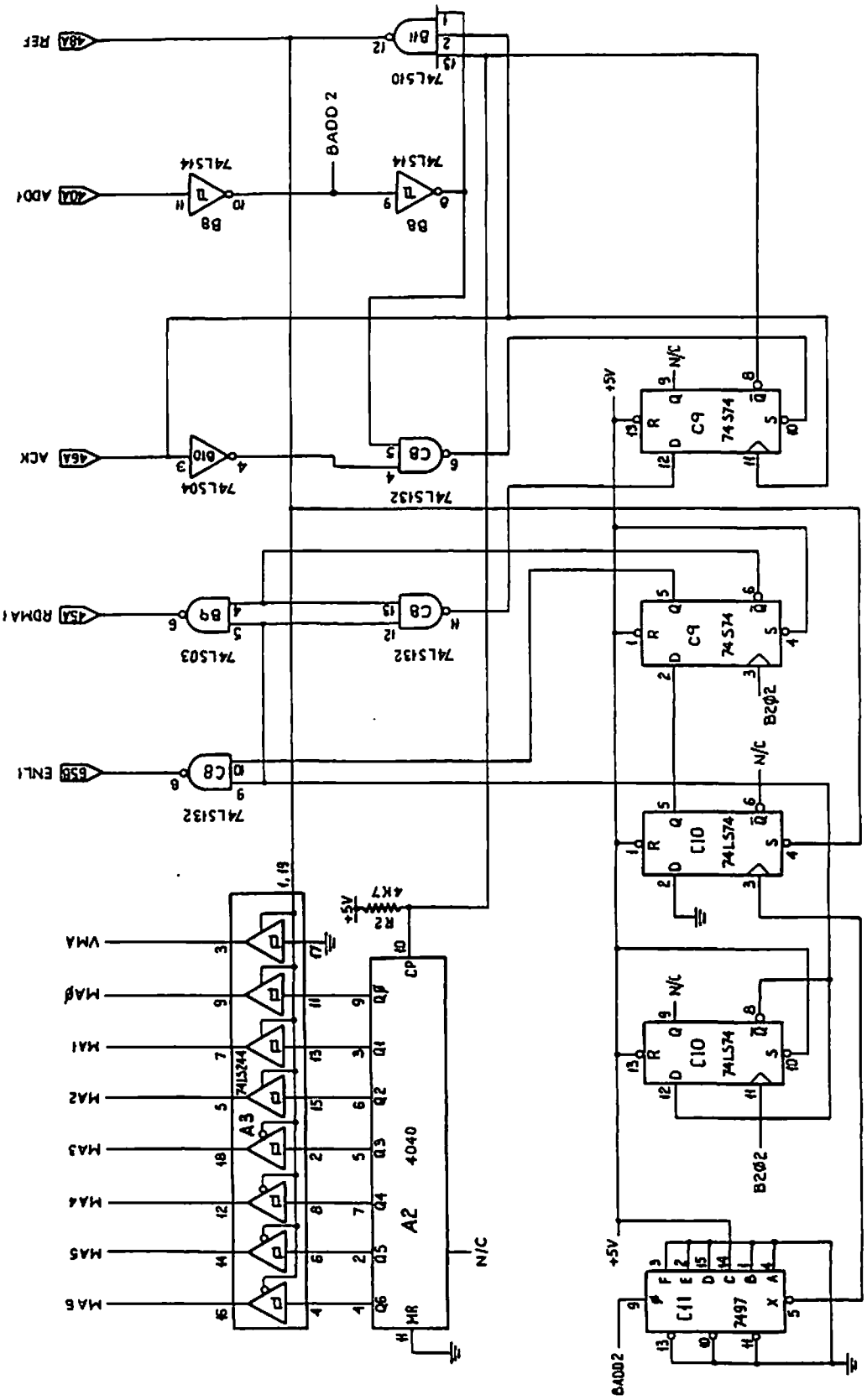

ROM, RAM, Real-time clock, PIA
DRAWN: PV REVISION: 7
D9
D8
D5,6
C5,6
E9,10,11
E12
F10,11,12
DIO
CB
IRQPA
IRQPB
RAMEN
ROM1
ROM2
LITHIUM CELL 3.7 V TYPE TADIRAN 5135
MBR120P
MSM5832RS
32.768 KHz
PROCESSOR UNIQUE FB00 FBFF
COMMON FA00 FBFF


Serial Interfaces, Timer
DRAWN: PV REVISION: 7
S03 I/O
S02 COMMS & CONTROLS
E2,3
E1
E5
E4
E6,8
6551
6840
74LS139
74LS00
4007
LM555
MC3487
MC3486
74LS86
74LS04
74LS03
74LS08
4488
4489
1N914
SCLK
ATIRQ
RESET
ACIA
THIER
LRW
SYNCS
ACINT
RTINT
Y2
18.432MHz


Interrupt Control, Data
Buffers and Restart
DRAWN: PV REVISION: 7

Memory Refresh

DRAWN: PV REVISION: 7



Q256

256K Ram Card

2.4

Note - In this document active low signals are indicated by a slash (/) in front of the signal name. This document is updated for the Rev. 2 Q256.

Introduction

The Q256 is a 256K x 9 bit dynamic RAM, organised as four blocks of 64K and mapped in 2 or 4K chunks. 32 different mappings from "processor space" to physical memory space may be set up. The mapping selected for any given cycle is automatically switched according to the current machine state. The machine state comprises which processor is on the buss, the user state/system state output of the processor, and which DMA channel is active, if any.

The ninth bit in the memory is a parity bit. Parity generation takes place automatically upon writing to the RAM and parity error detection is automatic when reading. If a parity error is detected, an interrupt is generated. A status register records that a parity error occurred, the physical memory block in which it occurred, and the upper five bits of the processor address which was active at the time of the error. The error status bit is automatically cleared after the register has been read.

The map selection logic also generates three control signals: 1) PENB is a universal buss signal which enables or inhibits access to all peripherals on the buss. This signal is fed via the motherboard back into the Q256 since the map selection logic and the mapram are themselves peripherals. The output signal is forced active after power up to ensure configuration of the mapping system is possible, and released by the first read of the parity status register.

2) VENB is a video RAM enable bit which allows accesses in the range $8000 to $BFFF to read or write to the graphics RAM or user RAM which may be mapped into this area instead.
3) PERGEN is a bit used to artificially generate parity errors to for testing purposes.

Options

Option blocks W1 and W2 allow selection of mapping on 2K or 4K byte boundaries as marked on the component overlay. If 4K mapping is selected, every second double-byte mapram location is not used. W1 and W2 must be configured identically. Default links on the PCB are for 2K mapping.

Option block W3 is for debugging purposes and need only be installed if a faulty card is crashing the data buss of the test machine. The default PCB link to +5V must be cut. There are two non-default options. Option 3 (GND) permanently inhibits the data buss driver buffer. The memory can still be written to but read data will only get as far as the buss output buffer. This facility allows the operating system to boot and test programs to run on a healthy board while the faulty board runs in parallel without driving the data buss e.g. for signature analysis.

Option 1 enables the data buss permanently and is not normally used. The default link to DBEN prevents the buss driver from outputting data only until the first read from the status register. This allows the restart ROM program to initialize the memory mapping before enabling memory mapping.

A four-way DIP switch is provided to allow multiple Q256 boards to be installed. Only switches 1-3 are used, so up to 8 boards can be installed. Close a switch for each zero in the board number, i.e. card 0 has all switches closed. SW1 is the LSB.

Map Selection Logic

(refer schematic Q256-00)

The function of the map selection logic is to encode the current system state and generate a five-bit map selection number. It also generates the peripheral enable output signal (PENBOUT), the video ram enable signal (VENB) and a parity error generate signal (PERGEN) which forces an artificial parity error for testing purposes.

There are six possible states for each processor:

A or System state, no DMA

B or User state, no DMA

DMA on any one of four DMA channels. (Processor automatically switches to A state for DMA cycles).

The A/B/DMA state is encoded as three bits and the processor phase signal is added as a fourth bit and presented as an address via the multiplexor IC 4B to the map selection RAM (mapsel) ICs 5D and 6D. Thus each state corresponds to a location in the mapsel and its output data is the map selection number.

DMA claim signals (DMACpn, where p=processor and n=DMA channel) occur when a DMA peripheral has received a DMA acknowledge from the processor and arrive during the data phase preceding the actual DMA cycle. IC 13A is an 8 to 3 line encoder but since there are four channels for P1 and four for P2 the most significant output line only duplicates the processor phase signal and is not used. The GS output indicates that some DMA channel is active. Each processor has its own A/B line (AB1 and AB2). The nand multiplexor 9A selects whichever is relevant for the next cycle. This signal, plus the combinatorial logic of 10B, 11A and 11B and the buffered phase signal ($\overline{B\emptyset 22}$) provide the four-bit state number to the multiplexor 4B.

The mapsel RAM occupies locations FC40-FC4F. Since there are six possible states per processor only twelve locations are actually used, as follows:

FC40	P2 B state
FC43	P2 A state, no DMA
FC44	P2 A state, DMA channel 1
FC45	P2 A state, DMA channel 2
FC46	P2 A state, DMA channel 3
FC47	P2 A state, DMA channel 4
FC48	P1 B state
FC4B	P1 A state, no DMA
FC4C	P1 A state, DMA channel 1
FC4D	P1 A state, DMA channel 2
FC4E	P1 A state, DMA channel 3
FC4F	P1 A state, DMA channel 4

Access to these locations for initialization of the mapsel is decoded by IC 1B, along with the peripheral enable input PENBIN. The output of IC 1B is latched on rising BRA by IC 4A (drawing Q256-03) to produce LFC4X.

Writing to the mapsel RAM is the most time-critical of all operations on the Q256. Normally the entire current data phase is available to generate the map selection number for the next cycle but when writing, the current data phase must be used to write to the mapsel ram as well. This is achieved by making mapsel write cycles very short. To choose locations within $FC4X the mapsel address multiplexor IC 4B is switched over to the lower four latched address bits on the falling edge of ADD2ø2 (data and address busses are in phase). The data is written into the mapsel ram on the rising of BRA (IC 10B). The write pulse is removed and the multiplexor switched back to the current state number only 50nS later by the falling of BCAS (IC 10A).

The lower five bits of the mapsel RAM are the map select number (MAPSEL0-4). This plus the controls PENBOUT, VENB and PERGEN are latched on the rising of ADD2ø2 which begins the next address cycle.

The peripheral enable signal (PENBOUT) and the graphics enable (VENB) come from bits 7 and 5 of the mapsel ram respectively. After power up or system reset they are both forced active (high) by the flip-flop IC 14B so that access to the mapsel ram is ensured for initialization. This flip-flop also inhibits the data buss driver IC 9B (drwg Q256-03) if W3 has been linked to option 2 for debugging. The flip-flop is triggered as soon as either processor reads the parity status register and will remain set until the next SYRES.

Format of data written to the mapsel RAM is as follows:

Bit 0-4 Map select number MAPSEL0-4. (Inverted)
5 VENB (Write 0 to enable graphics ram)
6 PERGEN (Normally 0, 1 to force a parity error)
7 PENBOUT (Write 0 to enable peripherals)

Address Translation

(refer to Drawing Q256-01)

This section performs the mapping from the 64K processor address space onto the 256K physical address space. The outputs LMAP0-4 and MAP5,6 plus standard address lines MA0-10 constitute the 18 bit address required to uniquely access any location in 256K.

All mapping data is stored in the two 2148 static RAMs (mapram), each containing 1024 x 4 bits. If 2K mapping is selected, the lower five address lines of the mapram come from the upper five processor address bits via multiplexor ICs 2B, 3B and 3D. This divides the 64K processor space into 32 blocks of 2K each. In 4K mode only the top 4 processor lines are used and the LSB of the mapram address is tied low. The upper five mapram address lines come from the latched map select number, thus any one of 32 different complete mappings may be selected. The data outputs from the mapram include a card select bit (CSEL), a two-bit 64K rank select (MAP5,6), and a five bit page select which is latched on rising BRA (LMAP0-4). The page select bits become the upper physical address bits.

Although the mapram is 1K bytes in size it in fact occupies 2K of address space, from $F000-F7FF. Even locations in this range are dummy locations which serve only to set up the CSEL flip-flop, IC 1E. Writes to odd locations then store this single bit along with 7 bits of mapping data in the mapram. So a single 16-bit write to the mapram maps one 2K or 4K page of processor space to any 2K or 4K physical page in the 256K available. The processor space being mapped is determined by the address within $F000-F7FF written to and the physical page selected is given by the data written.

Valid addresses in the mapram range are decoded with the buffered peripheral enable signal (BPENBIN) by IC 2A and the write map (WMAP) signal is latched by IC 4A (drwg Q256-03). The mapram address lines are switched over to LA1-LA10 on the rising of $\overline{\text{ADD2ø2}}$ by IC 2D. The 2148s have common data-in and data-out lines multiplexed by the write ($\overline{\text{W}}$) input. If an odd address is being written, the buffered data buss is driven into the 2148s from IC 4C and the $\overline{\text{W}}$ input strobed by the upper LS20, IC 1D on the rising of $\overline{\text{BRA}}$. The write pulse is terminated by falling $\overline{\text{ADD2ø2}}$, and data hold time to the 2148s is provided by the disable delay of IC 4C.

Writing to an even address triggers the CSEL flip flop IC 1E on rising $\overline{\text{BRA}}$. The data to this flip flop is the result of the three-bit comparison of BD0-2 with the card select no. set up on the DIP switch, qualified by BD7 which may be used as an overall page enable bit. Thus mapram data format is as follows:

Even locations
- Bit 7 Page enable
- Bits 5-3 Unused
- Bits 2-0 Card select

Odd locations
- Bit 7 Unused
- Bits 6,5 Physical 64K block select
- Bits 4-0 Physical 2K block select
  (Bit 0 not used in 4K mapping configuration)

Note that when mapping multiple pages of memory in or out the CSEL flip-flop need only be set up once, after which only writes to odd locations in mapram are required.

Memory Block Decoding
(refer schematic Q256-02)

This section generates RAS and CAS controls for each of the four memory ranks according to mapping outputs MAP5,6 and the card select signal CSEL.

MAP5,6 are decoded to a 1-of-4 block select signal by IC 1E provided a valid address is on the buss (VMA), the Debug's Ram inhibit ($\overline{\text{RAMINH}}$) is inactive, and neither $FC4X nor the mapram are being accessed. IC 5E latches this select along with the refresh signal REF, read/write (mapsel, mapram, or status), read/write memory, and access memory signals on rising BRA. IC 2E generates the READ signal to drive the data buss while $\overline{\text{BRA}}$ is high (see drwg Q256-03).

Refresh cycles cause the system RAS signal to be distributed to all four memory ranks simultaneously through ICs 6F and 6E. No rank select and thus no CAS is generated during refresh cycles. During a valid memory access one rank is selected and CAS is routed to that block through ICs 5F and 6E. The R/W signal drives all ranks through ICs 5F and 3E.

BUSS Interface
(refer schematic Q256-03)

The buss address lines are buffered by ICs 6B and 5A and latched on rising BRA by ICs 4A and 5B for mapram and mapsel ram writing and status register reading, along with address decode signals WMAP and $\overline{\text{FC4X}}$.

Unmapped address lines MA0-10 (and MA11 in the 4K mapping configuration) plus mapping outputs LMAP0-LMAP4 (excluding LMAP0 for 4K mapping) are multiplexed onto the RAM address lines by ICs 5C and 6C.

IC 9B latches the RAM outputs on the falling edge of CAS and drives the buss when READ is high unless the debug option W3 prevents this. Data is buffered off the buss to mapsel, mapram, main RAM and parity generator by IC 7B.

Parity System
(refer schematic Q256-04)

The parity bit is automatically generated by IC 7D from the data on the buss when writing to RAM. This bit is written into the appropriate 6665 when the corresponding CAS signal is generated (see drawing Q256-02).

A RAM read cycle causes the parity bit from the selected block to be read out of the parity RAM while IC 7D simultaneously regenerates the same parity bit from the data actually leaving the main RAM and coming back in again through data buffer IC 7B. Thus parity errors may be caused by buss errors as well as RAM faults. The parity memory bit and the regenerated parity bit are compared by exclusive or gate IC 4F. The result of this comparison is only valid for about 150nS and is clocked by rising $\overline{\text{BRAS}}$ into flip flop 14B to generate the parity error interrupt (PERRINT) and the parity error status bit (PERR). Further clocking to the flip flop is disabled by the $\overline{\text{Q}}$ output going low.

On each RAM cycle, the rank select lines MAP5, 6 and 5 upper processor address lines are latched by IC 7A. As soon as a parity error occurs this latching is disabled by the /Q output of the parity flip-flop so that the physical block and the processor 2K address space in which the error occurred is recorded.

The parity system status register occupies the same space as the mapsel RAM: FC40 to FC4F. (The mapsel is write only, the status register is read only). The lower 3 bits of the latched address are compared with the module select lines MS0-2 from the DIP switch by IC 2F so that in a multi-card system parity registers on cards 0-7 are at FC40-7 respectively. A read from the status register enables the latch 7A and buffer 6A onto the data buss on the rising of /BRA and the parity error is cleared by a very short pulse (approx 20nS) on the rising of BRA at the termination of the read cycle. The flip flop is also cleared automatically by /SYRES.

The parity error generate signal PERGEN is used to artificially create a parity error for testing purposes. The I input of IC 7D is simply a ninth data input so that if it is 0 when a RAM location is written it must also be 0 when that location is read, otherwise an "error" will be generated. When testing the parity system, a location is written with the I input 0 and the read back just as the I input is switched to 1. PERGEN is the control for this and comes from the mapsel logic. Since the mapsel outputs are of a lookahead nature, the PERGEN signal is delayed by half a cycle by first latching on rising ADD2ø2 at IC 4D then clocking on rising /ADD2ø2 at flip flop IC1E.

In the diagnostic tests the A state is initialised with PERGEN reset and the B state is initialised with PERGEN set. After writing a location in the A state, the processor switches to the B state at the instant of reading back the location.

Memory Arrary

(refer schematic Q256-05)

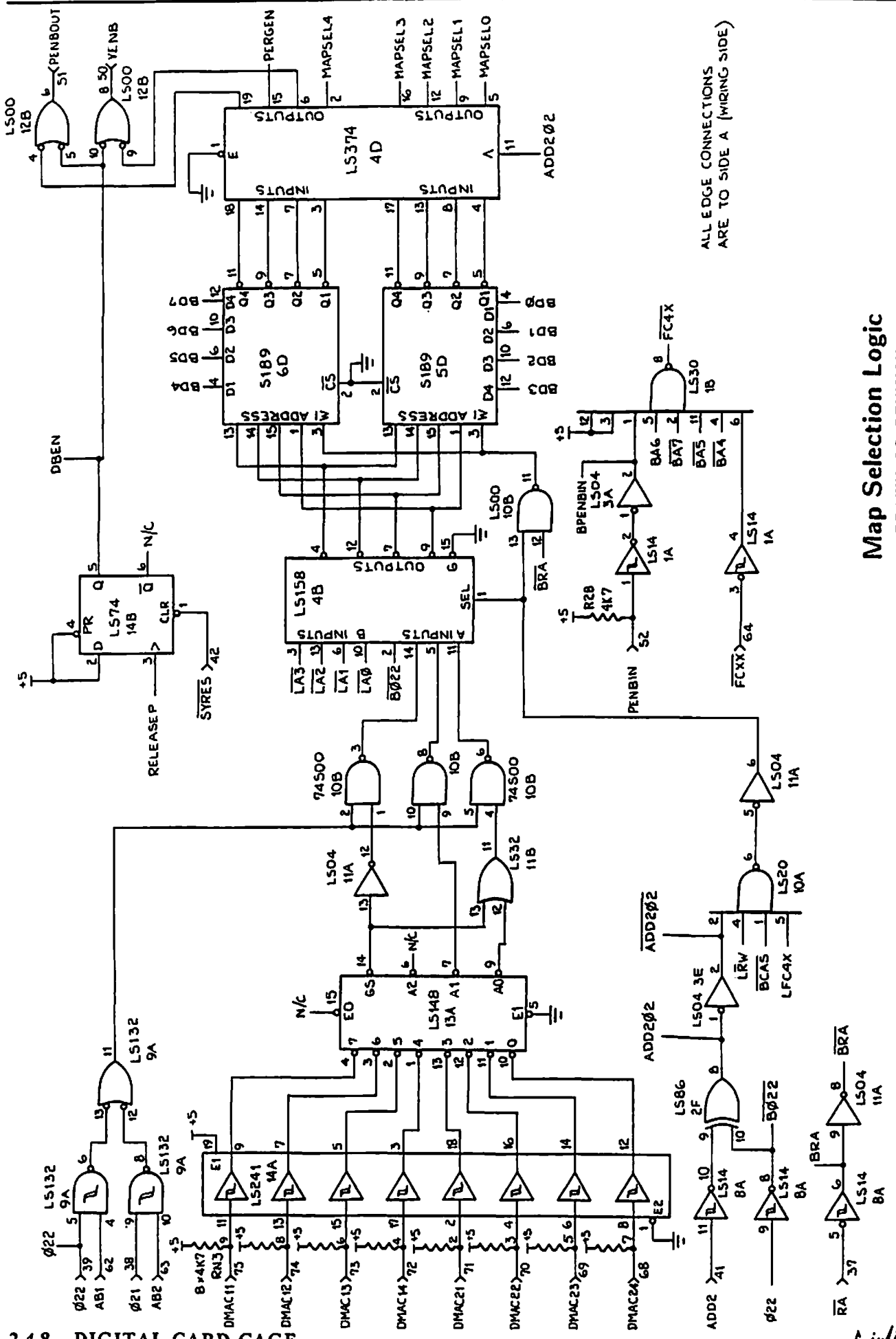
The dynamic memory array consists of four ranks of nine RAM ICs. Each IC is a 64K x 1 bit device, so each rank forms one byte plus parity bit. Which rank is accessed depends on which /RAS and /CAS lines are driven low by the decoding circuitry.

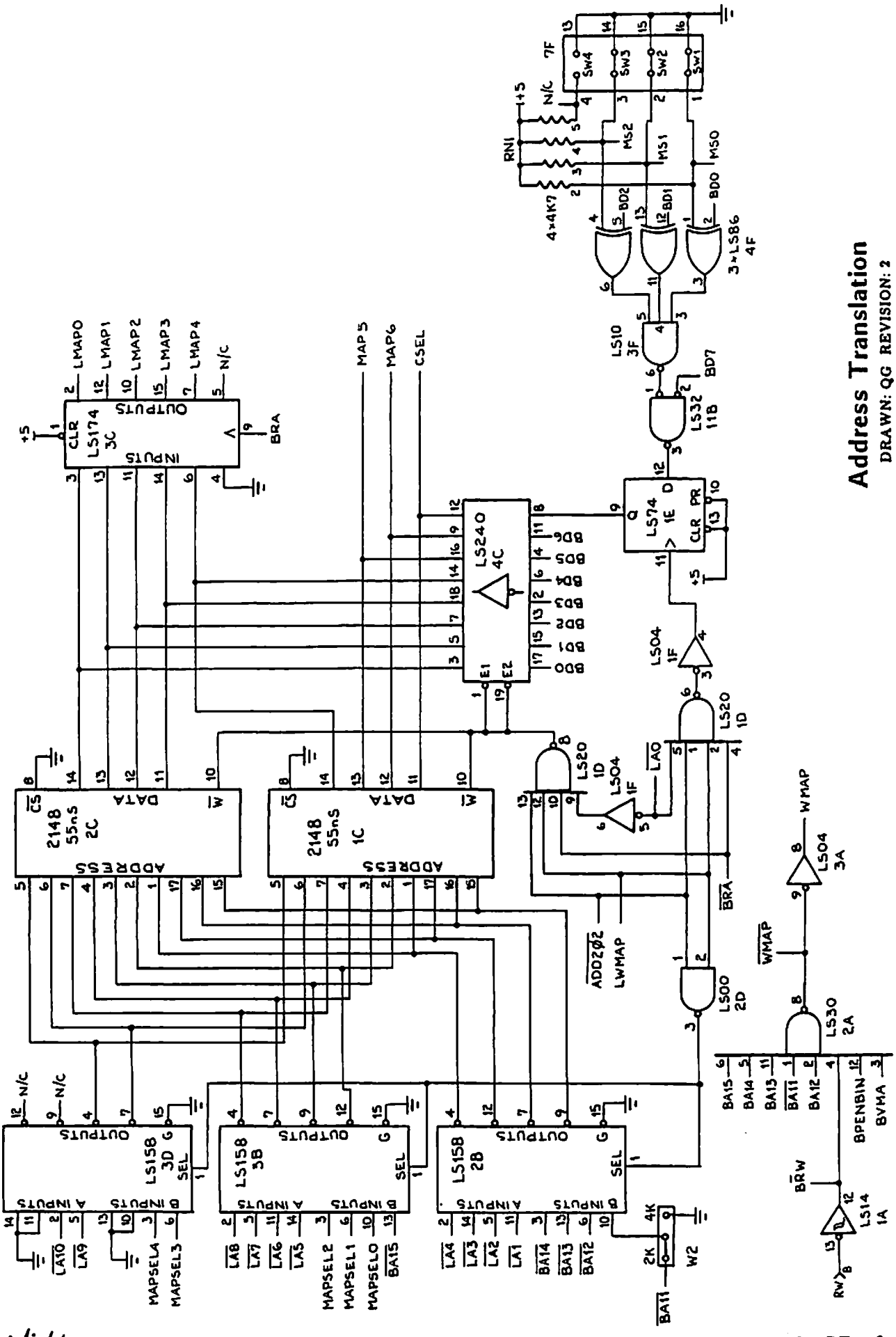
The RAM chips each have 8 address lines. Since 64k space requires 16 address bits the full address is multiplexed onto the 8 lines. The lower 8 bits ("Row" address) is latched into the RAM chips when /RAS goes low. After a period of address hold time the address multiplexor switches over and drives the upper 8 bits ("Column" address) into the RAMs. This is latched into the RAMs when /CAS goes low.

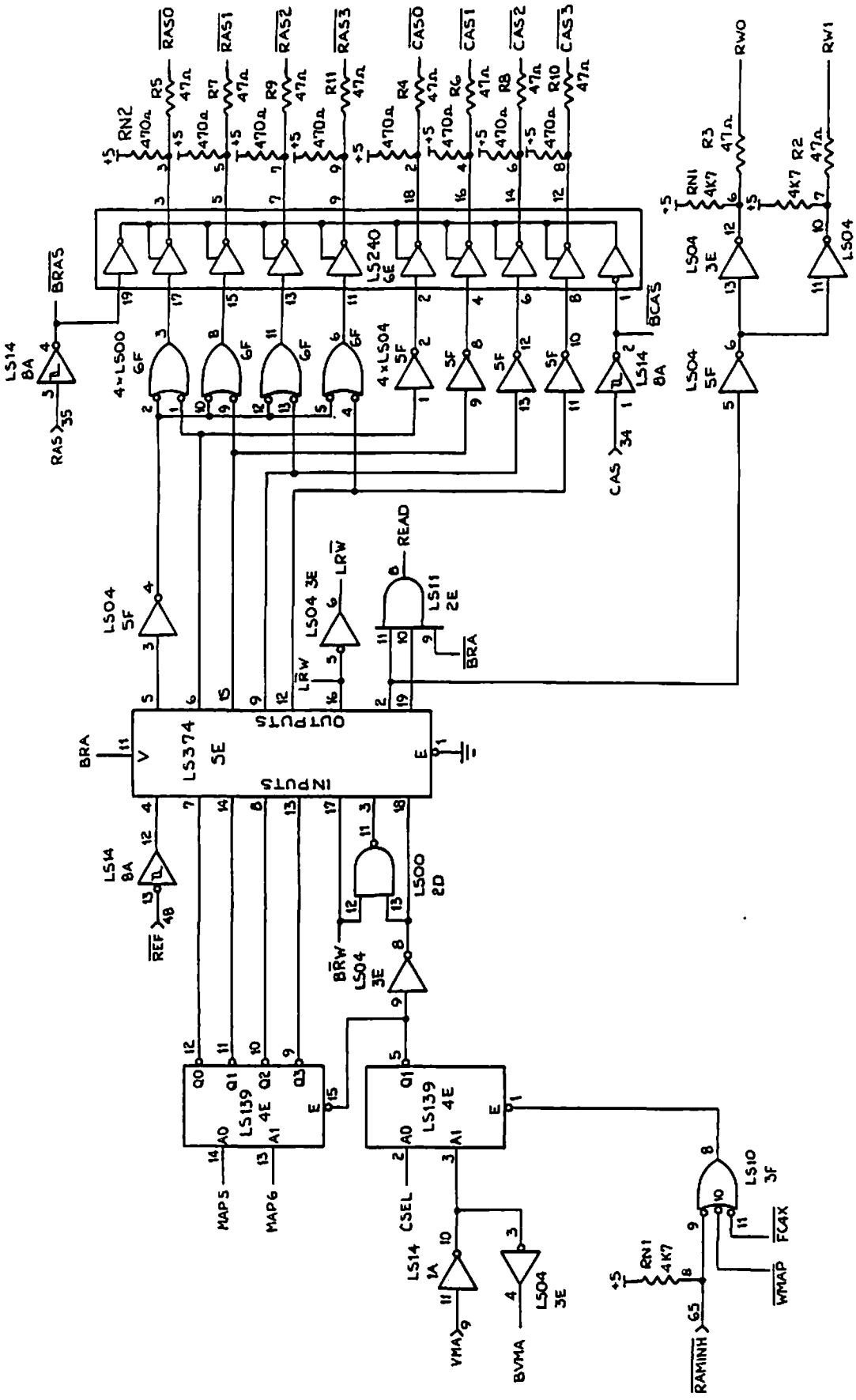
During a write cycle, data is latched into the RAMs on the falling edge of /CAS. In the case of a read cycle, data out becomes valid within 75nS of the falling edge of /CAS. The data lines are driven by the selected rank of RAMs while /CAS is low, and go tri-state at other times.

The state of the /W signal while /CAS is low determines whether the cycle is read or write.

Due to the capacitive input impedance of the MOS RAM ICs, the address and data input lines are driven through series resistors to limit the voltage undershoot.


Map Selection Logic
DRAWN: QG REVISION: 2
ALL EDGE CONNECTIONS ARE TO SIDE A (WIRING SIDE)
LS374 4D
LS189 6D
LS189 5D
LS158 4B
LS574 14B
LS148 13A
LS241 14A
LS132 9A
LS00 12B
LS00 10B
74S00 10B
LS04 11A
LS04 3E
LS04 3A
LS32 11B
LS20 10A
LS86 2F
LS30 1B
LS14 1A
LS14 8A
R8 4K7
RN3 8x4K7
PENBOUT
YENB
PERGEN
MAPSEL4
MAPSEL3
MAPSEL2
MAPSEL1
MAPSEL0
ADD2Ø2
DBEN
RELEASEP
SYRES
PENBIN
BRA
FC4X
FCXX
LRW
BCAS
LFC4X
ADD2
Ø22
Ø21
AB1
AB2
DMAC11
DMAC12
DMAC13
DMAC14
DMAC21
DMAC22
DMAC23
DMAC24
RA

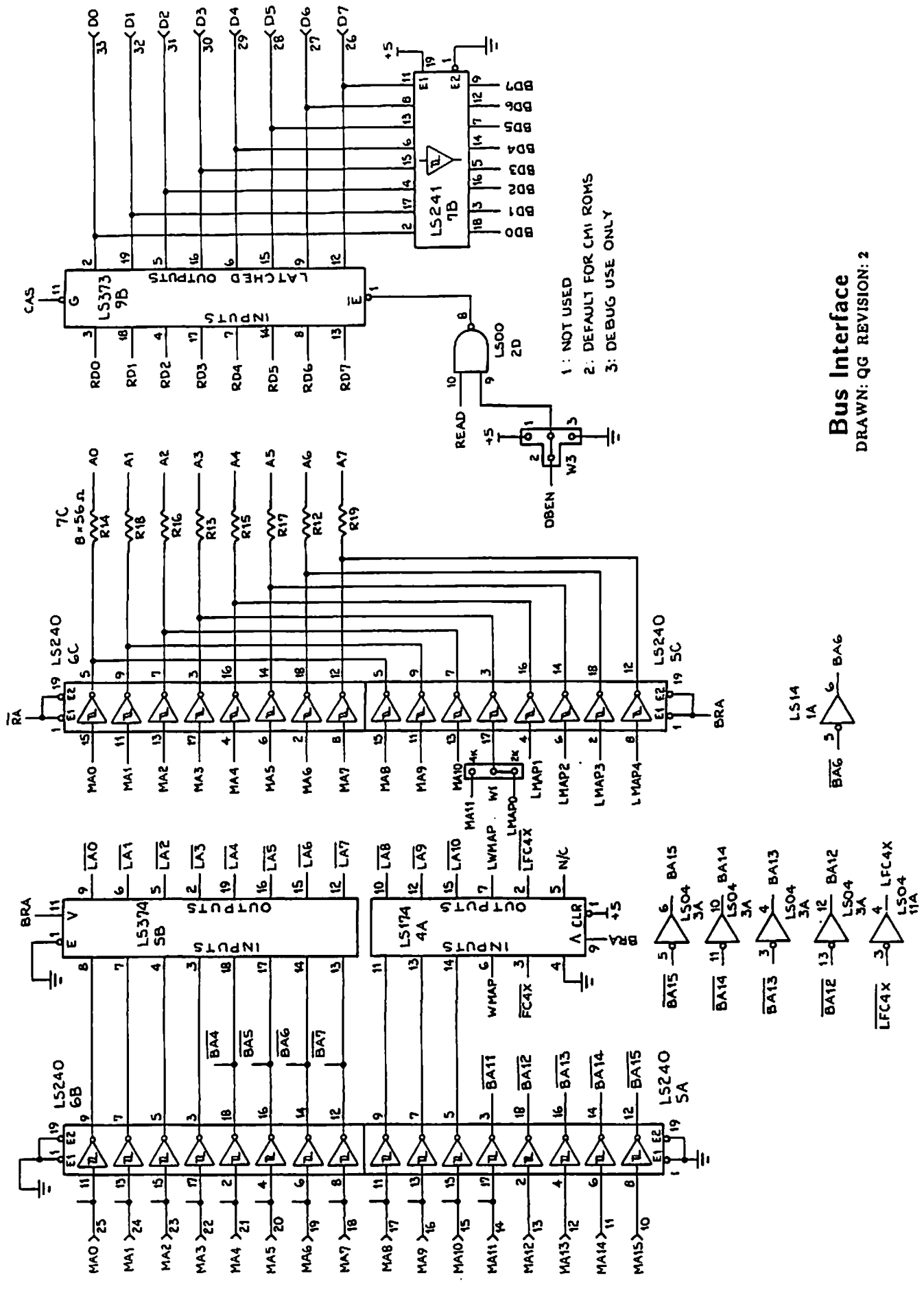

Address Translation
DRAWN: QG REVISION: 2
7F
SW4
SW3
SW2
SW1
+5
N/C
RN1
4x4K7
MS2
MS1
MS0
BD2
BD1
BD0
3-LS86
4F
LS10
3F
LS32
11B
BD7
LS74
1E
D
CLR
PR
Q
LS240
4C
BD6
BD5
BD4
BD3
BD2
BD1
BD0
E1
E2
LS04
1F
LS20
1D
LAO
LMAP0
LMAP1
LMAP2
LMAP3
LMAP4
N/C
OUTPUTS
LS174
3C
CLR
INPUTS
BRA
MAP5
MAP6
CSEL
2148
55nS
2C
1C
CS
DATA
W
ADDRESS
ADD2Ø2
LWMAP
WMAP
LS04
3A
LS00
2D
LS30
2A
BA15
BA14
BA13
BA11
BA12
BPENBIN
BVMA
BRW
LS14
1A
RW
LS158
3D
3B
2B
SEL
A INPUTS
B INPUTS
LA10
LA9
MAPSEL4
MAPSEL3
LA8
LA7
LA6
LA5
MAPSEL2
MAPSEL1
MAPSEL0
BA15
LA4
LA3
LA2
LA1
BA14
BA13
BA12
4K
2K
W2
BA11

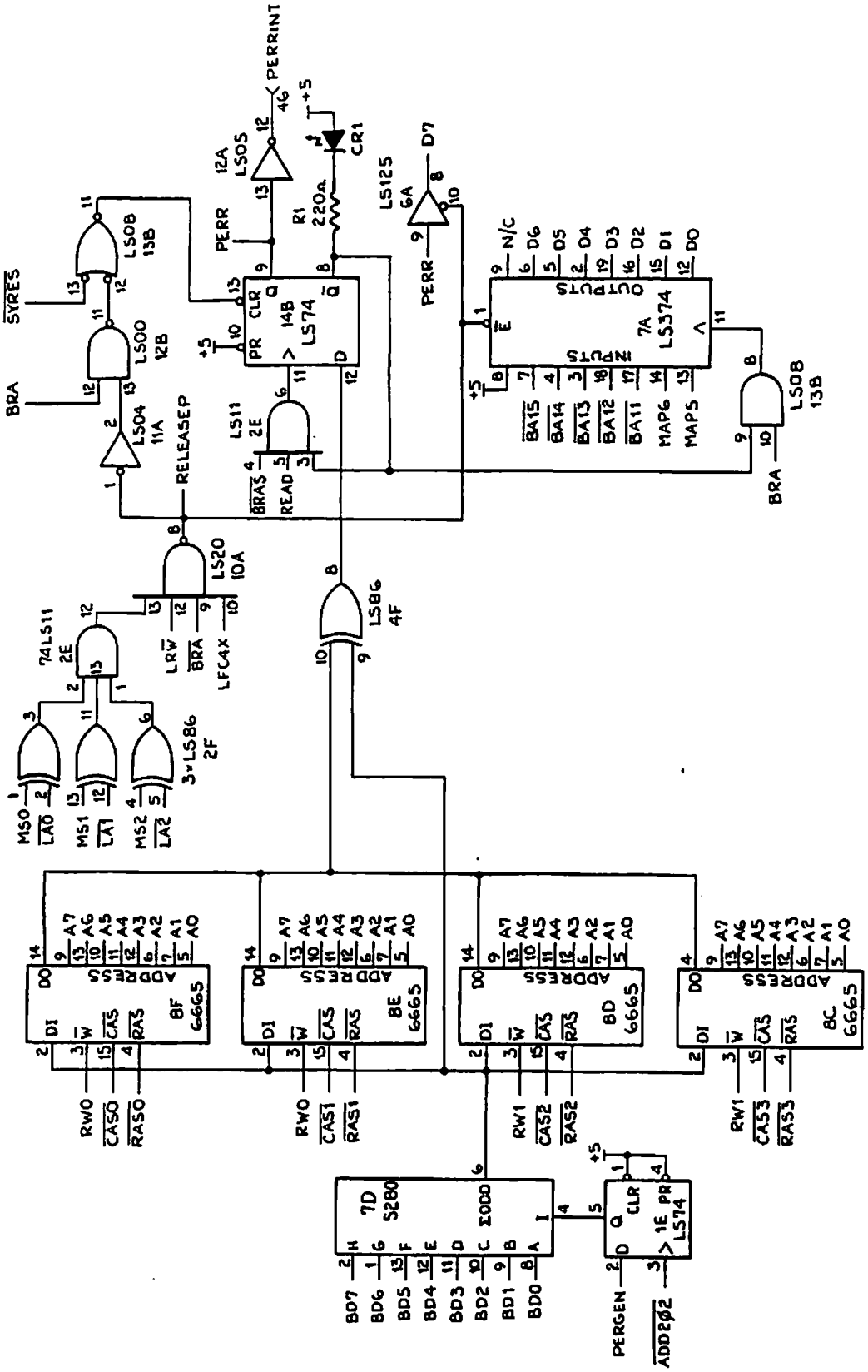


Memory Block Decoding

DRAWN: QG REVISION: 2

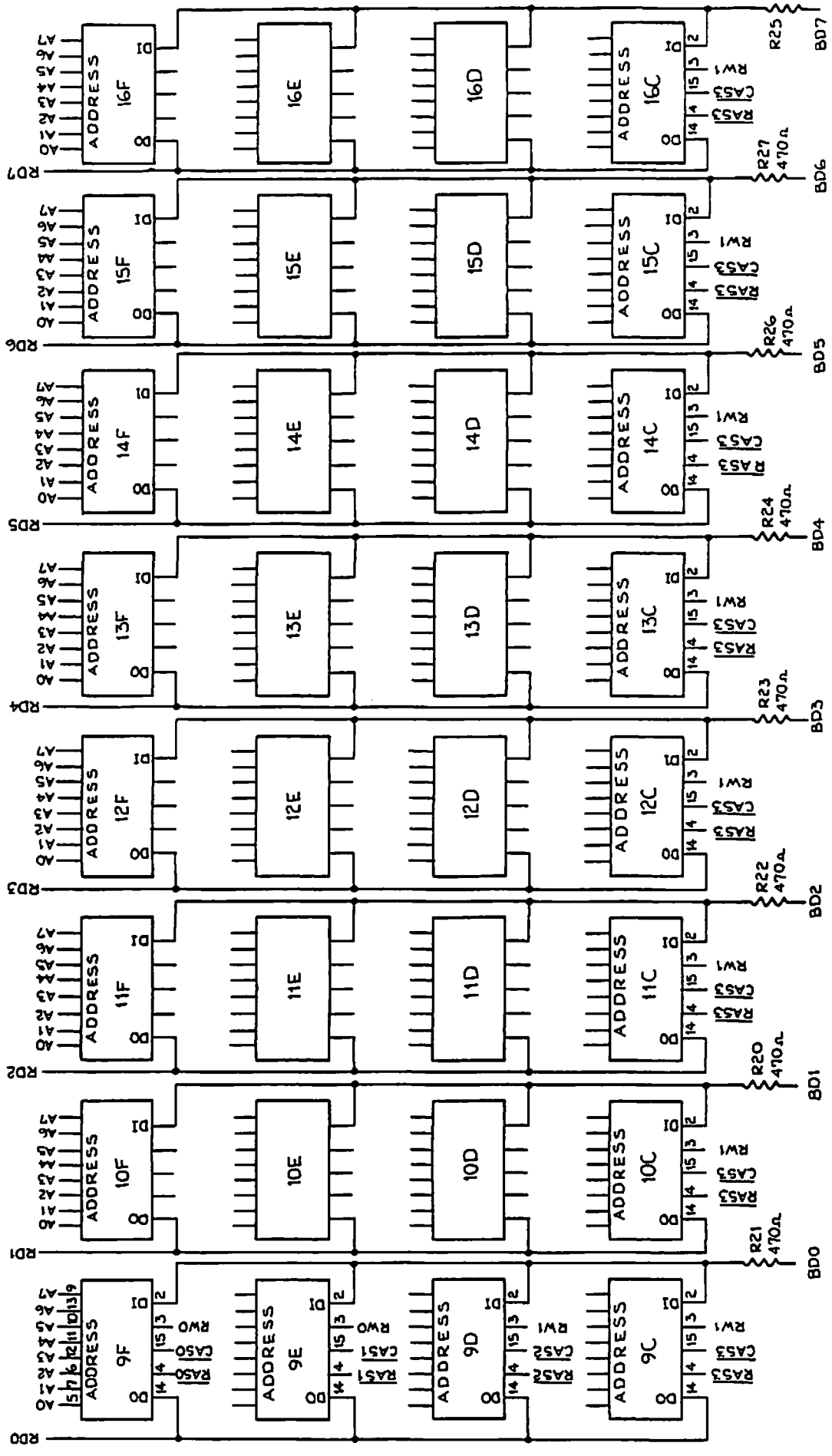
NOTE 70nS ACCESS TIME 2148's MAY BE USED AT 1C AND 2C PROVIDED AN 5139 IS USED AT 4E


Bus Interface
DRAWN: QG REVISION: 2
1: NOT USED
2: DEFAULT FOR CM1 ROMS
3: DEBUG USE ONLY
LS373 9B
LS241 7B
LS00 2D
LS240 6C
LS240 7C
LS240 5C
LS374 5B
LS174 4A
LS240 6B
LS240 5A
LS14 1A
LS04 3A
LS04 11A
CAS
READ
DBEN
BRA
BRA
W1
W3



Parity System

DRAWN: QG REVISION: 2


Memory Array
DRAWN: QG REVISION: 2
ALL IC's 6665
N.B. Vcc +5V - PIN 8
Vss GND - PIN 16
TIE PIN 1 TO +5V
ADDRESS
RAS0
RAS1
RAS2
RAS3
CAS0
CAS1
CAS2
CAS3
RW0
RW1
R20 470Ω
R21 470Ω
R22 470Ω
R23 470Ω
R24 470Ω
R25 470Ω
R26 470Ω
R27 470Ω
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
RD0
RD1
RD2
RD3
RD4
RD5
RD6
RD7

QFC9 2.5

Floppy Disc Controller

Introduction

The floppy interface card interfaces the bit parrallel/serial buss of floppy disk drives to the CMI's interleaved parellel buss. The interface is a combination of device driver software, controlling disk data format and initialization of data transfer parameters, and the hardware which carries out the transfers without processor intervention.

Data is stored on the floppy disk itself on its magnetic coating, in concentric rings. In a standard, 8 inch floppy there are 77 such rings on each side called tracks. Tracks are divided into data blocks called sectors. Sectors in Fairlight disk formats are either 128 or 256 bytes per sector, depending on operating system being used. The smallest amount of data that can be read to or from a disk is one sector. Sectors on a disk may be randomly accessed.

Track 0 is outermost. The controller automatically restores to this track on power on, and on reaching track 0, signals the controller by a mechanical switch generated signal. All head movement is relative to this reset state.

Floppy drives transfer data serially. They also have parallel control lines to control drive number selection, head stepping direction, head load, disk write and disk write enable. The head of the drive must be lowered to the disk surface before a transfer may take place, this is operation refered to as "head load".Index pulses are generated by the drive so that the controller knows the location of the rotating disk. This pulse occurs once per revolution , so the start of tracks can be determined by the controller. Also, the controller generates pulses that are used to step the head in and out to position it over the required ack.

The Floppy Disk Controller/Formatter uses the WD1791 LSI controller. It is software selectable to double density, double sided in addition to single density, single sided. It is designed to work with CPU #2's, transferring data to and from memory by DMA on Processor 2. The processor is not involved with transferring data to and from the disk. Once a data transfer is set up the processor may continue processing other tasks until the interupt for "command complete" is issued by the controller.

Address Map
(refer schematic QFC9-01)

The controller is accessed through two locations, in a memory map which enables access to peripherals. An address register is set up to point to the required controller register. All data is read or written through a single data register.

ADDRESS (HEX)	READ	WRITE
FCE0	data	data
FCE1	status register	address register

The 7 controller registers are:

00 control register
02 DMA address (low byte)
04 DMA address (high byte)
06 byte count to read/write (low byte, inverted)
08 byte count to read/write (high byte, inverted)
0A command location to load device driver ROM into RAM WD1791 L.S.I
0C cmd (write) status (read)
0D track
0E sector
0F data

The definitions of the control register bits are:

0 DS0 drive select address bit 0
1 DS1 drive select address bit 1
2 enable interrupt (active high)
3 enable DMA address incrementing (active low)
4 DMA transfer direction (1= to disk)
5 side select
6 retrig head load timer
7 DENS density selection

The definitions of the control status bits are:

0 0
1 n/c
2 n/c
3 ready
4 two sided
5 disk change
6 interrupt
7 device driver loading (active low)

Commands

The extensive instruction set of the 1791 LSI can be obtained from the manufacturers data sheets for the 1791. This device handles all data conversions between the disk drive and the C M I buss.

DMA address Counters

(refer schematic QFC9-02)

Sixteen bit counter chain C1 to C4 is used to provide the address for DMA transfers. The starting address for each disk transfer is established by writing the appropriate byte address to the address register then writing the address byte to the data register and then repeating for the other address byte. This causes the address to be preset into the DMA address counters by means of parallel-load strobe pulses STAL (low byte) and STAH (high byte). The incrementing of the DMA counters may be inhibited under software control, so that disk data may be dumped directly into the data portholes on channel cards.

DMA Byte Transfer Counters

(refer schematic QFC9-04)

Sixteen bit counter chain C5 to C8 is used to transfer the required number of bytes to or from disk. It must be initialized with the inverse of the number of bytes to be transfered. Any number may be specified up to a maximum of 65,535 bytes. Only those bytes specified will be tranferred to memory on a disk read. This allows less than a sector to be read from disk, and saves the software overhead required to handle partial sector reads. The read takes place but the buss VMA signal goes inactive after the required number of bytes have been transferred, so disabling memory writes. The VMA disable signal is generated from the ripple carry out on this counter chain, by buffering /FINPS, (Finished Partial Sector).

When a transfer occurs, the DMAC (Direct Memory Access Claim) line is generated so that the memory card swaps maps, allowing data to be dumped into memory currently not mapped into the processor's address space. This signal is generated by the components around flip-flop A11.

Data Buffers

(refer schematic QFC9-02)

Data is propagated from the system data bus via latch B6 which hold the data across the processor 1 phase. This latched data also becomes the DATA FROM BUS via buffer B5, to the floppy-controller LSI.

Data written to the system control register at 00 is latched by B7. This controls such functions as drive select and DMA direction.

Address Decoding

Address range $FCE0-$FCE1 is decoded by gates B1, E1, B2, E1 and latched by D2.

Address $FCE0 is used to enable the internal data buss to read and write to controller functions.

Address $FCE1 data is latched by B8 and with the access to FCE0 generates the internal chip selects and read/write strobes through C9.

Inverting buffer E5 and open collector drivers E6, E7 are used to interface the 1791 LSI controller to the disk drive cable. Incoming disk status signals are pulled up by 150 ohm terminating resistors.

Controller LSI

The Interupt Request from the LSI is gated with the Interupt Enable to provide an open-collector interupt signal for the system I.R.Q. on buss pin 63A.

DMA Logic

(refer schematic QFC9-03)

Data requests from the 1791 or Device Driver rom loading are synchronised with Processor 2 Phase 2 using flip-flops C1 and A10. This sets up a DMA request to the processor (RDMA).

DMA cycles are granted by ACK acknowledge signal.

Flip-flop A11 only allows a DMA cycle to occur every second Processor cycle (the floppy drive can not transfer at that rate but this is a system constraint on other DMA devices in the DMA daisy chain).

The DMA daisy chain is controled by /ENL and /EDL. Respectively these stand for, Enable Next Level and Enable Dma Level. When /EDL is active, a DMA request may be requested by the highest priority device. The /ENL signal informs the next device in the daisy chain that it may make a request if higher priority devices have not.

Depending on which function has been requested (Reset, Read, write) the required DTB (Data to Bus), and ATB (Address to Bus) signals are issued.

Control Register

The control register contains the drive number select bits, density selection, interupt enable, increment DMA address enable, data transfer direction, side select and retrigger head load delay.

This register is the latch at B7.

The "retrig head" signal is used to reactivate the head load delay when the drive number has been changed, to allow for head bounce.

Master Oscillator

(refer schematic QFC9-05)

The LSIs used on the card require a master 16MHz clock. This is generated with the components around the 74S04 at F5. The FDC9229 generates the 2 MHz clock for the 1791, by dividing this internally.

Write Precompensation

In double density operation there may be a time shift applied to the data when it is being written to the inner disk tracks (>45). The amount of shift is determined by the LSI and the floppy disk suport device 9229. They produce a programmable delay of 1 to 3 clock cycles.

The amount, if any, is specified by the drive manufacturer. Links W4, W5, W6, W7 select the amount. The amount on inner and outer tracks can be independently set.

W5	W6	W7	Precompensation Value (in,nS)
0	0	0	0
0	0	1	62.5
0	1	0	125
0	1	1	187.5
1	0	0	250
1	0	1	250
1	1	0	312.5
1	1	1	312.5

The precompensation value is normally set to 0 on inner and outer tracks. It is more important on inner tracks as the bit density on the disk is greater.

W4 selects minifloppy drives and also requires the board to be made with a 34 pin connector (or a special cable) and an 8MHz crystal.

Data Separator

The serial data stream that comes from the drive is in a synchronous form. It has embedded the required data as well as clock pulses and synchronization "marker bytes".

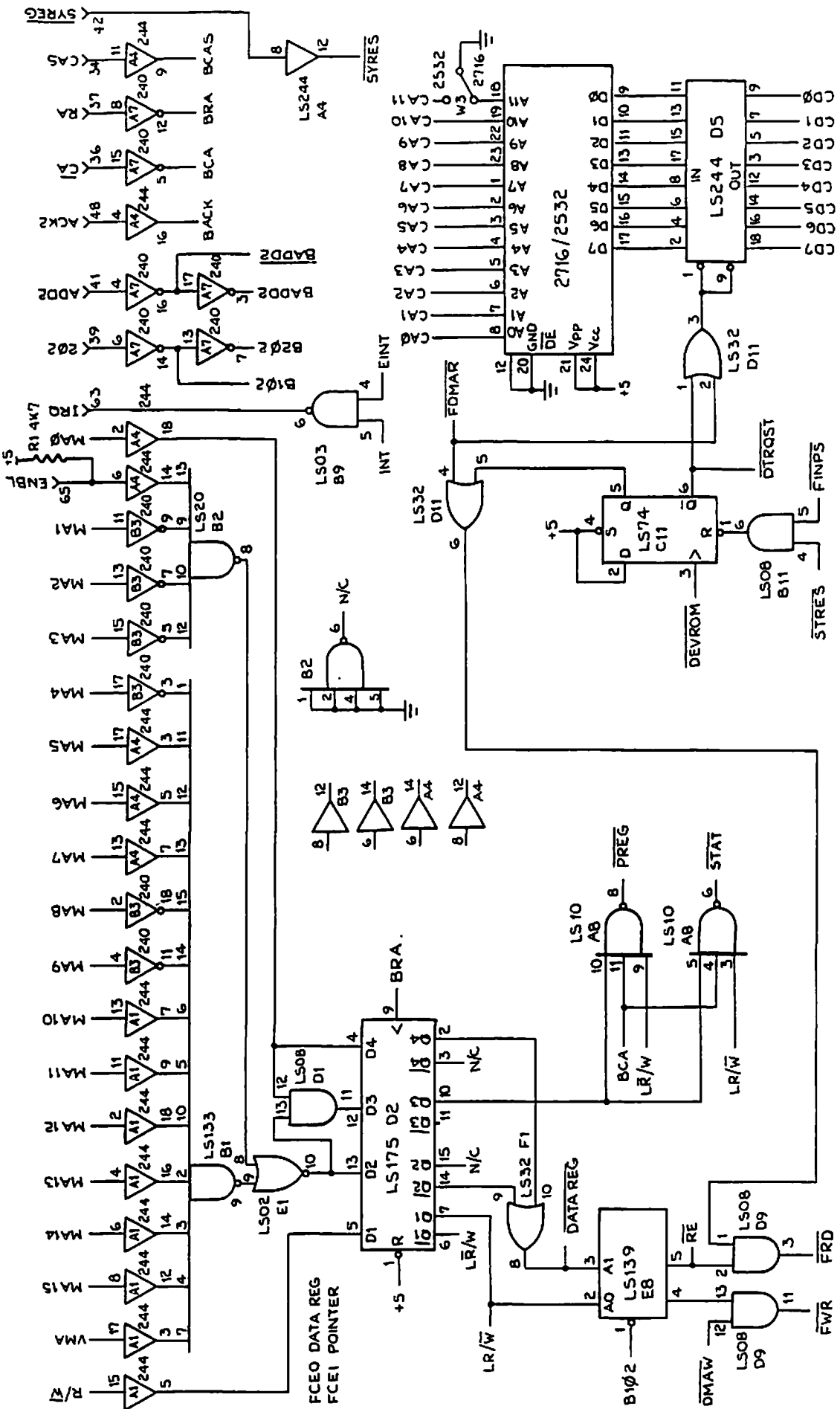
The data separator is used to generate the data window from the FM (single density) or MFM (double density) encoded READ DATA supplied by the disk drive. The separator tracks, so that incoming data is always in the center of the data window. This data window informs the controller chip which bits in the data stream it is receiving are data and which are clock bits just used in the data encoding scheme.

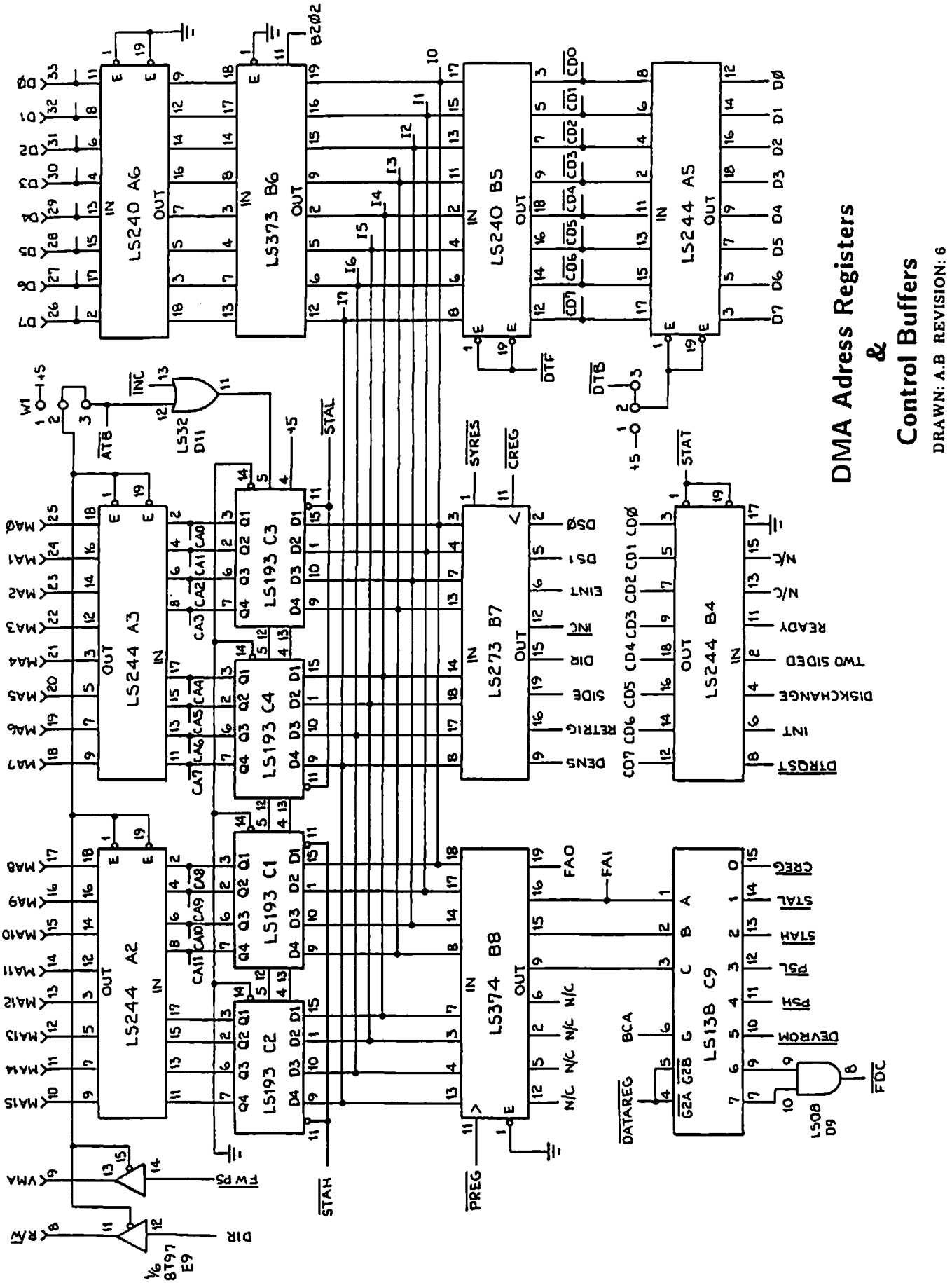
The separator is a digital phase locked loop in the FDC9229 chip at E10. This chip does all the work of data separation.

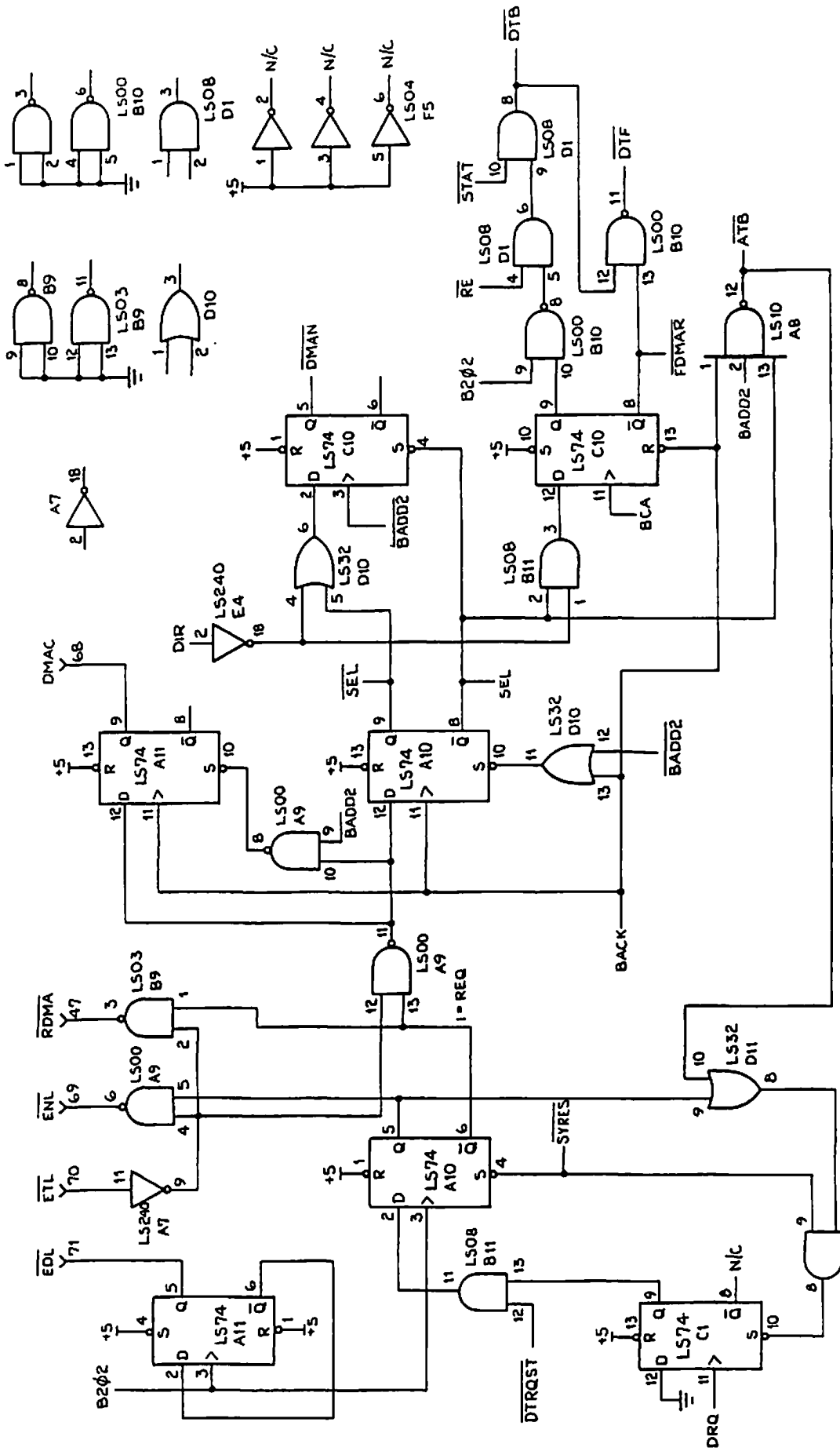
Device Driver ROM

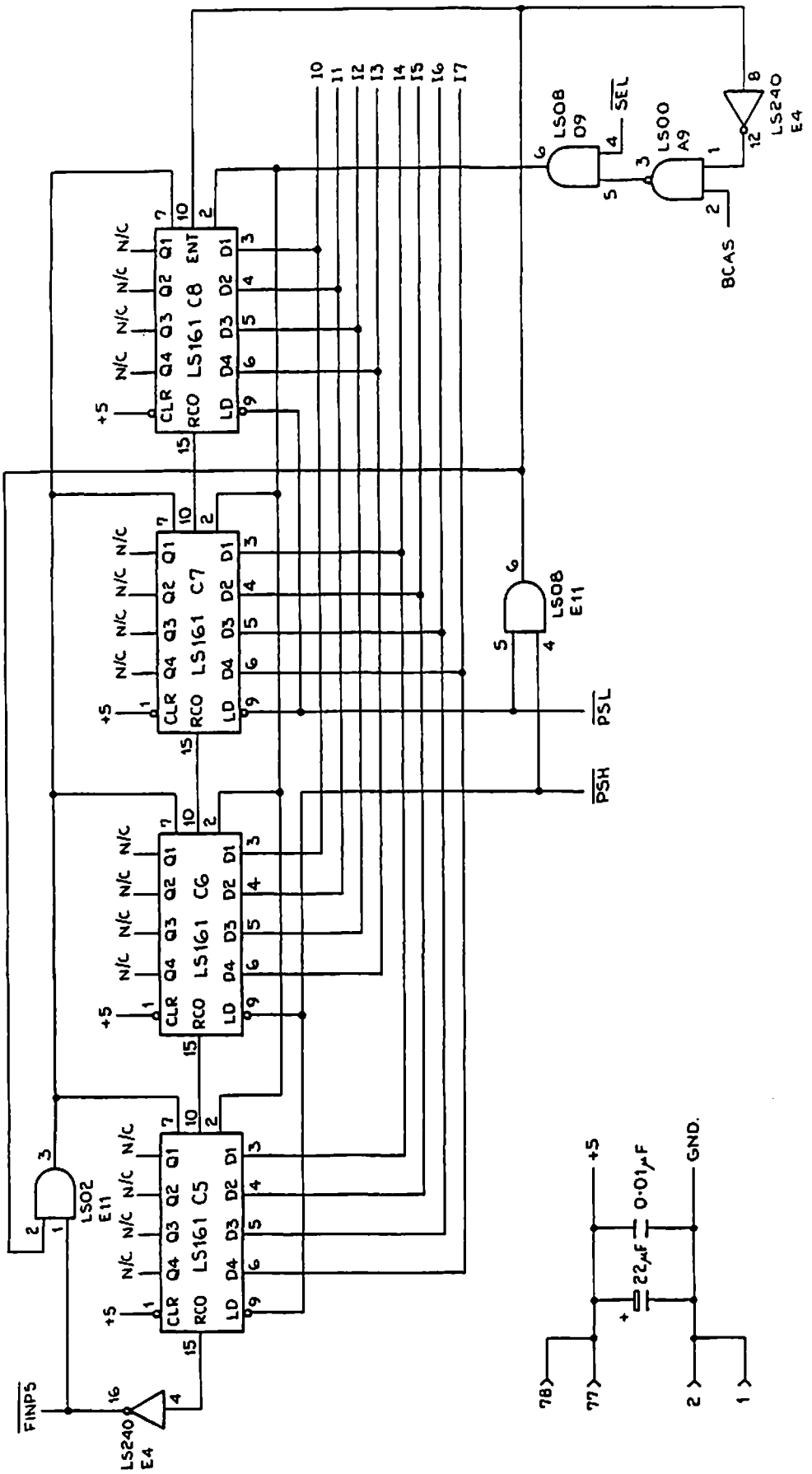
The disk controller software may be placed in a 2K or 4K EPROM on the controller card. This EPROM is not in the processor's directly addressable memory. It is executed by reading the software into RAM. This is done by DMA. The EPROM is copied into RAM as if reading a disk, except much faster.

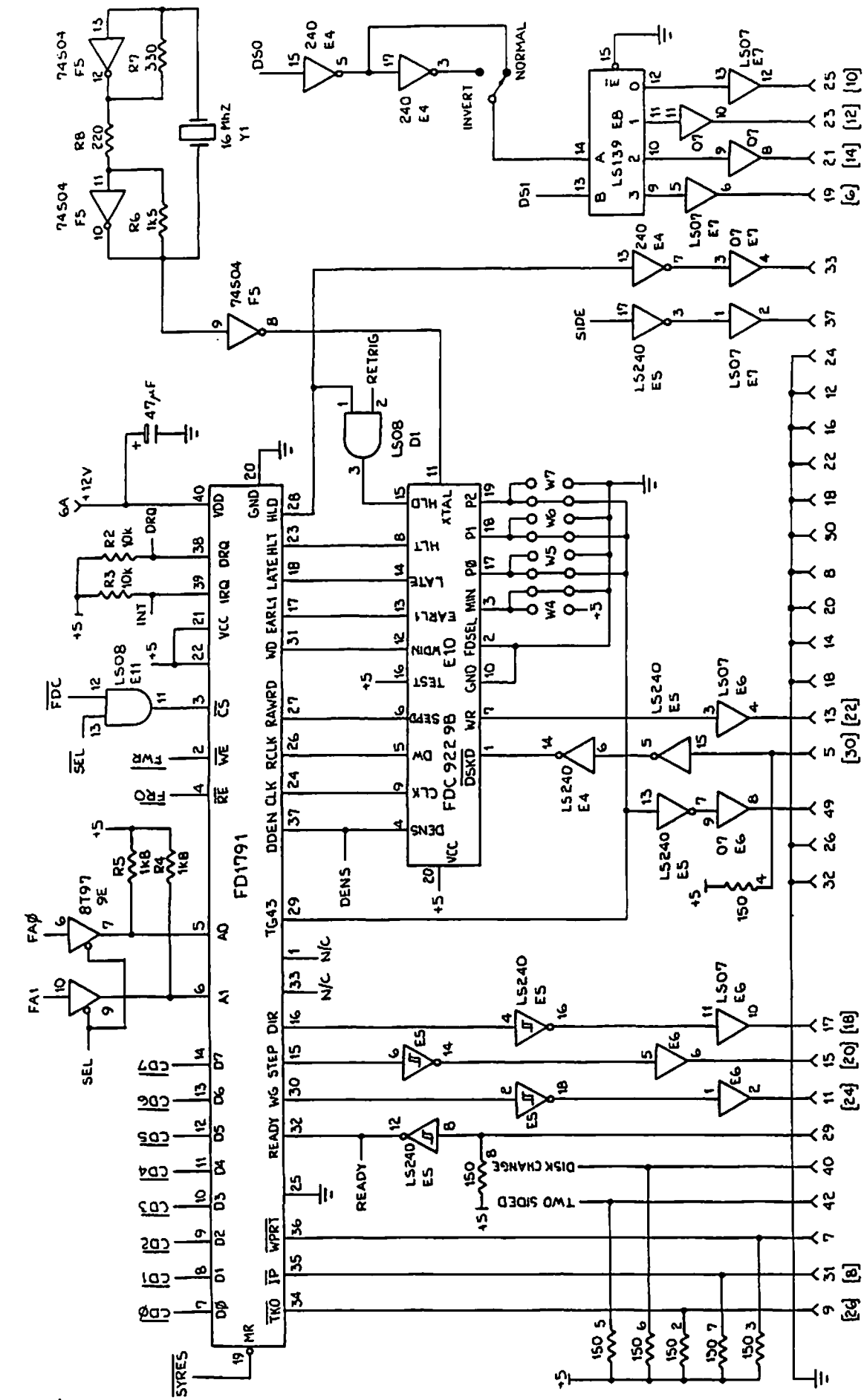
The least significant DMA counter lines are used as addresses on the EPROM, so the EPROM can only be loaded into memory on 2k or 4k boundaries. The flip-flop C11 and gates in D11 and B11 produce a DMA write to memory request that is terminated after the byte counter times out.


Address/Control Buffers
Devicer Driver ROM
DRAWN: A.B REVISION: 6
2716/2532
LS244 D5
LS32 D11
LS74 C11
LS08 B11
LS03 B9
LS20 B2
LS133 B1
LS02 E1
LS175 D2
LS32 F1
LS08 D1
LS08 D9
LS139 E8
LS10 A8
R1 4K7
SYREG
SYRES
BCAS
BRA
BCA
BACK
BADD2
B2Ø2
B1Ø2
EINT
INT
IRQ
ENBL
FDMAR
DTRQST
FINPS
STRES
DEVROM
PREG
STAT
BRA.
N/C
DATA REG
FCE0 DATA REG
FCE1 POINTER
LR/W
RE
FRD
FWR
DMAW
R/W
VMA
ALL UNMARKED NON INVERTING BUFFERS LS244
" " INVERTING " LS240
WITH PINS 1 AND 19 GROUNDED.


DMA Adress Registers
&
Control Buffers
DRAWN: A.B REVISION: 6
LS240 A6
LS373 B6
LS240 B5
LS244 A5
LS244 A3
LS193 C3
LS193 C4
LS273 B7
LS244 B4
LS244 A2
LS193 C1
LS193 C2
LS374 B8
LS138 C9
LS08 D9
B2Ø2
INC
LS32 D11
ATB
DTF
DTB
STAL
STAH
SYRES
CREG
STAT
PREG
TW PS
DIR
FDC
DATAREG
BCA
DEVROM
PSH
PSL
STAH
STAL
CREG
FA0
FA1
READY
TWO SIDED
DISKCHANGE
INT
DTRQST
DSØ
DS1
EINT
INC
DIR
SIDE
RETRIG
DENS
1/6 8T97 E9


DMA Logic Daisy Chain
DRAWN: A.B REVISION: 6
LS00 B10
LS08 D1
LS04 F5
LS03 B9
LS74 C10
LS74 A10
LS74 A11
LS74 C1
LS32 D10
LS32 D11
LS240 E4
LS240 A7
LS08 B11
LS10 A8
DMAN
BADD2
SEL
DIR
DMAC
BACK
1=REQ
RDMA
ENL
ETL
EDL
SYRES
DTRQST
DRQ
B2Ø2
RE
STAT
DTB
DTF
ATB
FDMAR
BCA


Byte Transfer Counter
DRAWN: A.B REVISION: 6
LS08 09
LS00 A9
LS240 E4
BCAS
SEL
LS08 E11
PSL
PSH
LS02 E11
FINP5
LS240 E4
LS161 C8
LS161 C7
LS161 C6
LS161 C5
CLR
RCO
LD
ENT
+5
GND.
0·01μF
22μF


FDC/Buffers PLL
DRAWN: A.B REVISION: 6
FD1791
FDC 9229B
NUMBERS IN [ ] REFER TO 5¼" DISK 34 WAY CONNECTOR
WHICH HAS ALL ODD PINS GROUNDED.
INVERT
NORMAL
RETRIG
DISK CHANGE
TWO SIDED
READY
SIDE
DENS
SYRES

Cable Connector Signals

DRAWN: A.B REVISION: 6

1		2	GND
3		4	HDL3
5	READ DATA	6	HDL2
7	WRITE PROTECT	8	
9	TRACK Ø	10	
11	WRITE GATE	12	
13	WRITE DATA	14	
15	STEP	16	
17	DIRECTION	18	
19	DS4	20	
21	DS3	22	
23	DS2	24	
25	DS1	26	
27		28	HDL1
29	READY	30	
31	INDEX	32	
33	HEAD LOAD	34	HDLØ
35	ALT	36	IN USE
37	SIDE SELECT	38	
39		40	DISK CHANGE
41		42	TWO SIDE
43		44	
45		46	
47		48	
49	LOW CURRENT	50	

Q219 2.6

Light pen/Graphics System

Note The lightpen system is no longer used, however the hardware support is still present on the Q219 card. For this reason we have not deleted references to the lightpen and associated circuitry.

Introduction

The Q219 provides a graphics display and light pen input system. The 16 kilobytes VRAM is displayed as an array of 512 pixels (horizontal) by 256 lines (vertical). The lightpen can detect points within this array.

To increase graphics drawing speed, extensive use is made of special hardware functions which provide automatic address incrementing or decrementing along either or both axes and individual pixel writing at the current address.

Video RAM is arranged such that consecutive bits in each byte correspond to consecutive pixels in the horizontal line. With 512 pixels per line, 64 bytes of VRAM is required per line. Thus there are 64 bytes between a given pixel and the pixel vertically below it. The first byte in VRAM ($8000) appears at the top left corner of the screen.

Locations FCD0 to FCDC perform store and auto-increment/decrement functions as follows:

FCD0 just store at current position
FCD1 store and inc Y
FCD2 store and dec Y
FCD3 store as byte at current Pos
FCD4 store and inc X
FCD5 store and inc X,inc Y
FCD6 store and inc X,dec Y
FCD7 store as byte,and inc X
FCD8 store and dec X
FCD9 store and dec X,inc Y
FCDA store and dec X,dec Y
FCDB store as byte and dec X
FCC4 scroll latch (port A of PIA)

The pattern contained in the register stored at the particular location above may be varied to produce solids or various forms of dotted or dashed lines on the screen.

The current position is established by directly accessing the VRAM with a dummy read. Subsequent use of the special locations then work from that position on the screen in various directions.

The byte mode operations are provided for the writing of alpha-numeric data to the screen. The VRAM arrangement explained above means that the fastest way to place characters on the screen is to write a series of horizontal slice bit pattern to consecutive vertical locations. Note that byte mode operations only work in the vertical (x) direction. This direction requires the actual VRAM address to be advanced by 64 bytes. Byte mode in the (y) horizontal direction is achieved by storing directly into successive locations in VRAM.

All non-byte operations store one point on the screen. What is written to this point (0 or 1) depends on what is in the corresponding bit position of the accumulator used to store the data.

Dot clock Generation

All timing is derived from a crystal oscillator and a counter chain.

The dot clock is constucted around inverter F2 and the 10.38MHz crystal. The string of counters at F9 to F11 and E11 to E13 count the dotclock DOTCLK and with the aid of SROM produce all video related signals.

The screen area is organized as 84 bytes on the line and 304 lines in the frame. The actual displayed area is 64 bytes on the line and 256 lines. One byte time is 8 DOTCLK pulses or 770ns.

The SROM prom at E10 generates the horizontal sync pulse and an advanced blanking signal. The horizontal blank signal (HBLNK) is the 7th bit of the horizontal byte count. The horizontal sync pulse lasts for 6 byte times and starts at the 68th byte position on the line. The vertical sync is generated by gates in E9, F6, D13 and E11. This pulse starts at line 272 and lasts 4 lines.

Gate in E11 generates the last line count (LASTL) at line count 304 and resets the entire counter chain to zero, the first byte and first line of the displayed area.

Video FIFO and Shift Register

(refer schematic Q219-02)

To compensate for the fact that bytes from memory need to be fetched at greater than 1 MHz, (1.185MHz i.e., 64 bytes need to be displayed in one active line time of 54uS) for displaying, a FIFO register (first-in, first-out) is used. Data from the VRAM (MO1-MO8) is latched by octal latch B9. When FIFO input is not full, the IR signal causes a data request to be clocked through flip-flop E6, which in turn generates a shift in (SI) pulse to the FIFO. This occurs every microsecond until the FIFO input queue is full, as indicated by IR returning to a logic zero.

Data is shifted out of the FIFO to the parallel-in serial-out shift register F7 by SO (Shift Out) pulses coming from the bit counter chain and SROM. They start one byte time before the line is unblanked and stop one byte time before the line is blanked. One byte is transferred from the FIFO to the shift register every eight picture bits. The picture information is shifted out by clock pulses from the bit rate oscillator (DOTCLK).

Video Output

Video data from the shift register is EXCLUSIVE-ORed with the INVERT signal from the PIA at D, E4. With a logic 1 at this pin the picture appears normally with set bits in the VRAM displayed as bright pixels. If a logic zero is applied, the picture is inverted (negative) so that ones in VRAM appear as black pixels on a BRIGHT background.

The output from this exclusive-OR gate passes through another, and is gated with the lightpen cursor signal INV.

The output from this gate is serrated at the bit rate by ANDing with DTCLK* at F6. This is done to avoid horizontally adjacent dots merging to appear brighter than vertically adjacent ones.

The serrated video is then blanked by gating with the combined horizontal and vertical blanking signals by F5 and F6. This removes unwanted (inactive) display time from the display. Sync is added to the video by combining the horizontal and vertical sync pulses in F5 and resistor network R10 and R9. This is buffered for low impedance driving by transistor Q1. The composite video signal is available at both 10-way connectors at the front of the card.

Address Decoding

The Lightpen/Graphics Card occupies two areas of memory space. The VRAM uses 16K bytes from $8000 to $BFFF. The various auto-incrementing portholes use 15 bytes $FCD0 to $FCDB, and the PIA and TIMER use FCC4 to FCDF. Addresses in the FCCX to FCDX range are decoded by half of NAND gate A2. VRAM addresses in the $8000 range are decoded by the other half of A2. These are enabled by AND gate D3, and gating together VMA, ENBL and ADD1/ADD1* which makes VRAM processor unique.

Either processor can always access the PIA and Timer on the card, if peripherals are mapped in. See Q256 RAM card for explanation of mapping and peripheral control.

The select signals from this gating are latched, along with the five least significant address bits and the read/write line, by octal latch B3.

The 16 auto-increment functions are decoded from the four least significant address bits by dual one-of-four selector C3.

Data Buffers

Octal buss Buffers B8 and transceiver B7 interface the card to the system data buss. Buffer enables and control signals are generated via the buss control prom BROM. The buss control prom is used to simplify the complicated enabling of the buffers as they must be enabled at several different addresses and modes of operation of VRAM access.

Addressing counters

The address being accessed within VRAM is determined by the state counter chain B4, B5, B6, A4 and A5. These are up-down counters which provide the auto-increment (and auto-decrement) functions. The starting address for any operation is established by any access to the VRAM. This is achieved by the counters being parallel-loaded by ADLD (address load) that occurs when a read is made between $8000 and $BFFF, in a map that contains the graphics ram.

Horizontal incrementing or decrementing is achieved by pulses YUP and YDN clocking the low-order 9 bits of the counter chain. The 3 lowest bits are used to select the bit within the byte. Vertical incrementing or decrementing is achieved by clocking the highest 8 bits of the counter.

Bit Selection

The 16K byte of VRAM is bit addressable, that is each row of chips can be individually written to.

For Byte mode operations, the BYEN signal is TRUE, and when a memory write is performed, the WROM A6 (write control prom) activates all the write lines causing the whole byte to be written irrespective of the current bit position within the byte.

For Bit mode operations, only the bit determined by the contents of A5 will have its W* signal activated .

Vertical Scrolling

(refer schematic Q219-05)

Counter chain D5, D6 and D8 generate the 14 bit address for the VRAM display operation. Each time a byte is fetched into the FIFO an INCA (Increment Address) pulse clocks the counters on to the next byte. The starting line is preset into counters D5 and D6 during line 304 by signal SYNR* produced by F13 and SROM. (This also resets the FIFO and allows it to be filled with the data for the first displayed line before the line is unblanked. This prefilling is necessary as the FIFO is emptied faster than it can be filled, during the active part of a line.)

The starting line number comes from port A of the PIA. Using this side of the PIA allows the starting line to be read from the latch directly via software.

VRAM Address Multiplexing

(refer schematic Q219-06)

Quad data multiplexers C6 to C8 multiplex the addresses of the VRAM between processor accesses (addresses from counter chain B4, B5, A4, B6 and A5) and video display addresses (counter chain D5,D6 and D8).

Switching is done by BADD2X signal which is derived from phase two of the processor to which access has been enabled.

Light Pen Introduction

(refer schematic Q219-04)

The Light Pen is used to generate co-ordinate data from Hit and Touch signals coming from the light pen. These come onto the card via the lower 10-way connector.

he video chain when a touch and hit occur from the lightpen. The co-ordinates are then available to be read from the card by the processor.

pen, which generates a Touch signal. The interface can be configured to generate an interrupt when the touch signal is activated.

The card can also be configured so that the Hit signal causes the picture information to be inverted, which generates a "cursor" on the display at the position the pen is currently "seeing".

The Hit signal from the pen is de-glitched by the interface so that random noise due to external interference will not cause false triggering.

Hit Deglitcher

Deglitching operates by ensuring that light pen hits are repeated a minimum number of times on successive display lines.

Counter E2 is normally in its terminal count state, counting being inhibited by RCO being high, which forces P low. A Lightpen Hit pulse arriving at gate D1 will RESET the counter via its clear input, CLR. This will clock the bit and byte latches via LPBITS through gate D3 and flip-flop E6 after being synchronized with the valid bit clock.

The counter then procedes to count line blanking pulses, HBLNK. Once two lines has been counted, the next Hit pulse will cause a logic 1 to be clocked by flip-flop D2. On the next byte parallel load, PLOAD*, this is propagated to the second flip-flop D2, generating a LPSTB pulse to clock the line latch. If no Hit occurs during the third line after the first hit, the count continues to its terminal state (16) without a LPSTB being generated.

Hit and Touch Receivers

The Hit and Touch lines from the lightpen are TTL compatible signals, active low. They are terminated at the receiving end by resistor networks R1, R2, R6 and R7. Depending on the state of the mode bit TFH the Touch signal may be ANDed with the Hit signal to generate a valid Hit pulse.

Note that gating is arranged to disable the cursor when the Touch is asserted. This is to ensure that the pen has something to "see" when activated.

The lightpen has its own supply regulator to reduce noise sensitivity. This is provided by a 78L05 device.

Co-ordinate Latches

(refer to drawing Q219-03)

The current bit within the byte is counted by F9. The byte on the line is counted by F10 and F11. Their contents are latched by D9 and D10 when a valid hit occurs by LPBITS, (deglitched Hit).

The least significant bit within the byte is available at bit 7 of port B of the PIA. The rest are read by the processor when LPL1* is active.

The line in the frame is counted by E13 and E12 and latched at D12 by LPSTB. This octal latch is read when LPL2* is active.

Control PIA

(refer schematic Q219-01)

Peripheral Interface Adapter D, E4 provides read/write ports for a variety of different functions. The side A is configured as outputs and are used to control the screen scrolling. The B side is input and output and controls screen inversion, processor access to VRAM, TFH (touch for hit) mode, lightpen cursor enable, lightpen touch status and the least significant bit within the byte counter.

The PIA is also used to generate interrupts. Two separate interrupt outputs are possible, one for Touch and one for Hit. These are generated by the PIA CA1 and CB1 outputs respectively.

Timer

An M6840 Timer is provided for general system use. This device contains three independent 16-bit counters which can be configured under software control to count external events or internal clock pulses. Counter 1 is clocked by HBLNK to count lines, counter 2 is clocked by VSYNC to count frames, and timer 3 is clocked by a 1MHz system timing signal to count microseconds.

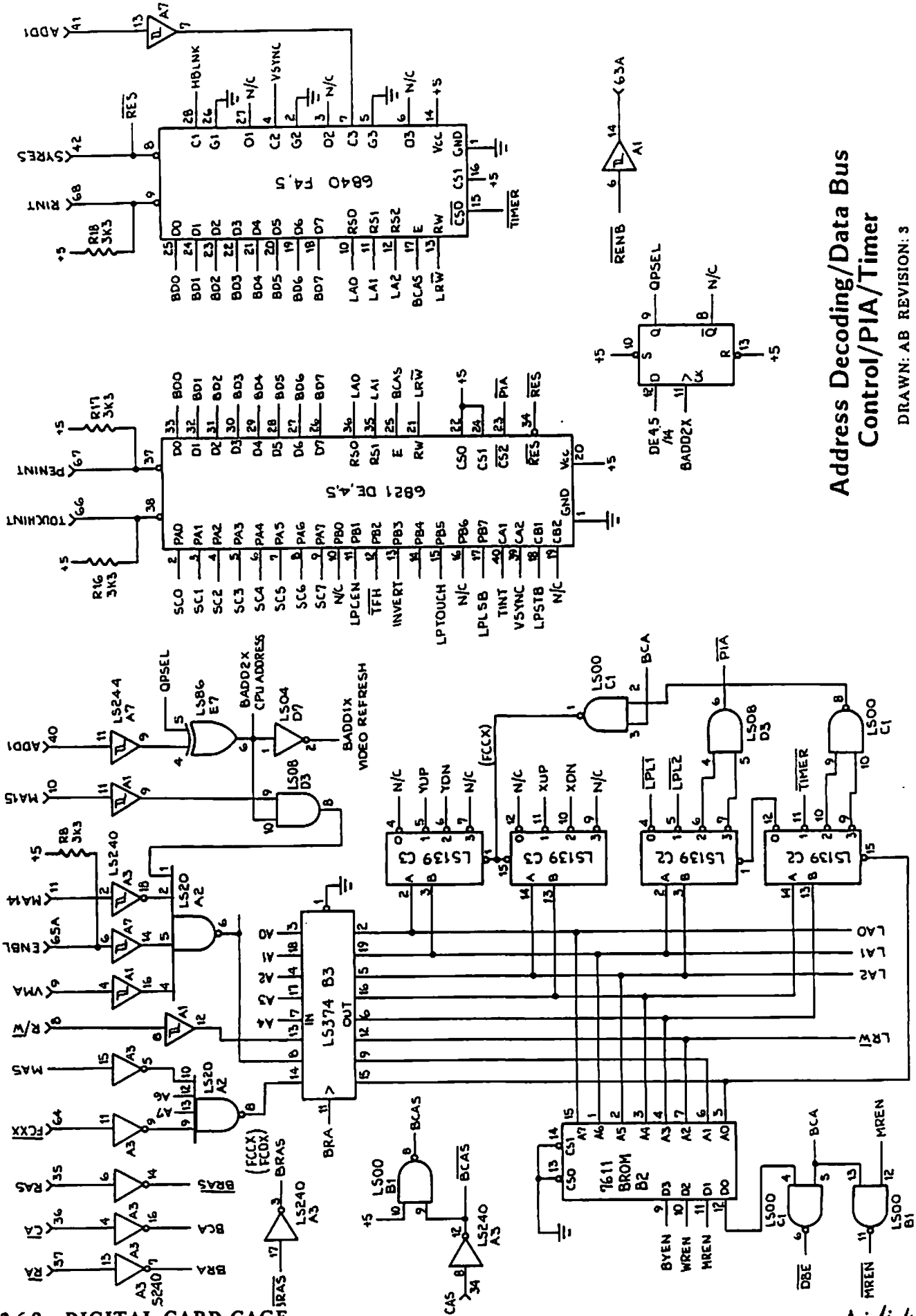
For full operational specifications of the 6840 refer to manufacturers data sheets.

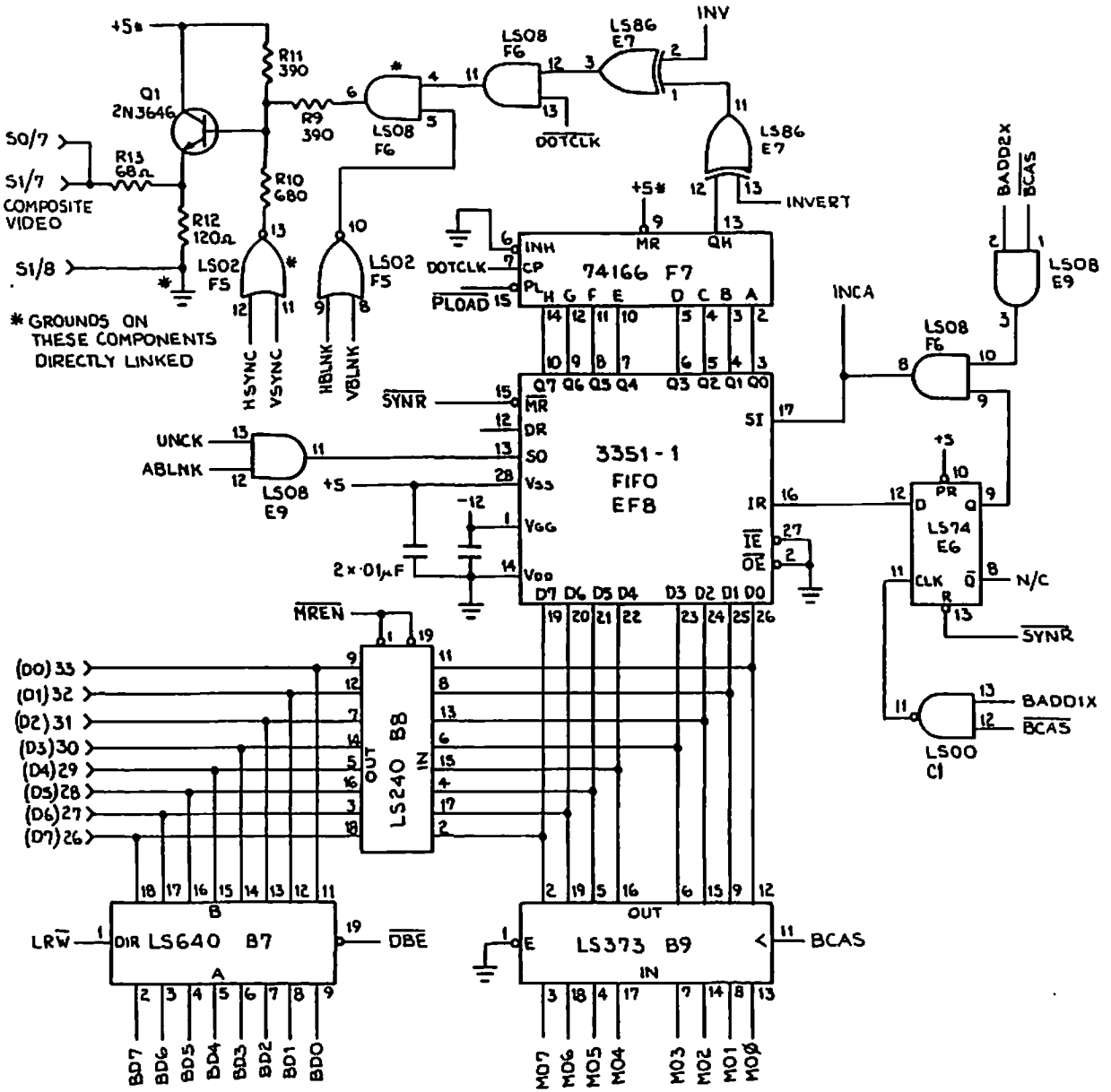
Video Memory Ram

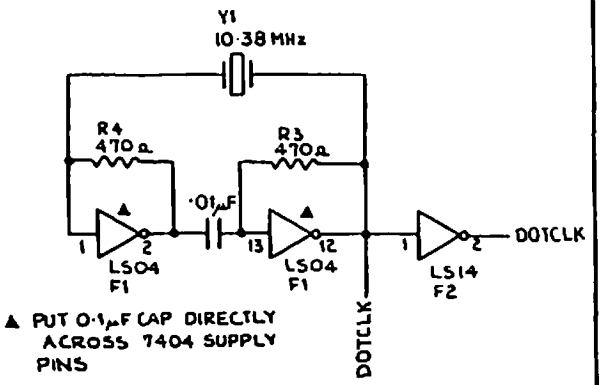
(refer to drawing Q219-06)

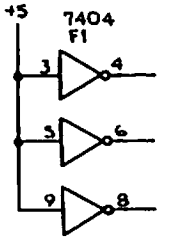
The 16 kilobytes of VRAM is provided by eight 4116 16,384 bit dynamic MOS devices. The timing required for these devices is preset in the system timing signals. These signals are buffered and fed directly to the rams as $\overline{\text{BRAS}}$, $\overline{\text{BCAS}}$ and $\overline{\text{BCA}}$. Memory refresh is done in the continuous access for screen refresh.

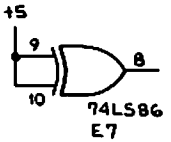
Whenever the VRAM is directly accessed, the decoding at A2, 7D, 7E generates RENB, (active low). This disables the memory card in the 16K block used by VRAM and saves duplicating maps in the Q256 ram card.

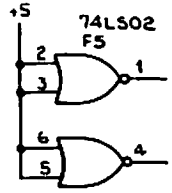

Address Decoding/Data Bus
Control/PIA/Timer
DRAWN: AB REVISION: 3
6840 PA.5
6821 DE.A.5
LS139 C3
LS139 C2
LS374 B3
7611 BROM B2
LS244 A7
LS86 E7
LS04 D7
LS08 D3
LS00 C1
LS240 A3
LS20 A2
LS00 B1
ADD1
RES
SYRES
RINT
R18 3K3
HBLNK
VSYNC
TIMER
RENB
QPSEL
DE4,5 /4
BADD2X
R17 3K3
PENINT
TOUCHINT
R16 3K3
PIA
R8 3K3
BADD2X CPU ADDRESS
BADD1X VIDEO REFRESH
(FCCX)
(FCDX)
BRAS
BCAS
RAS
CAS
BCA
BRA
LA0
LA1
LA2
LRW
BYEN
WREN
MREN
DBE
MRIN





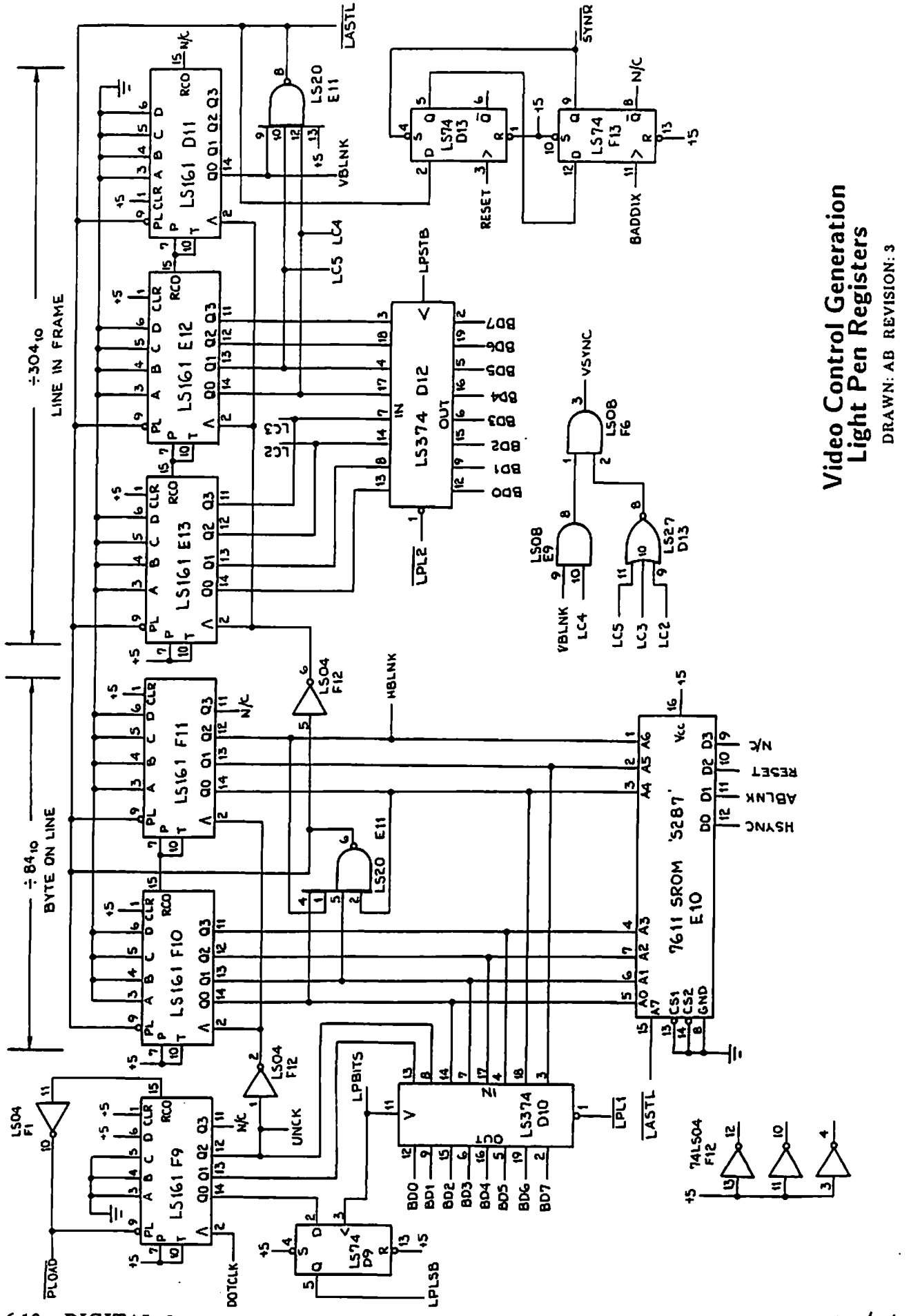


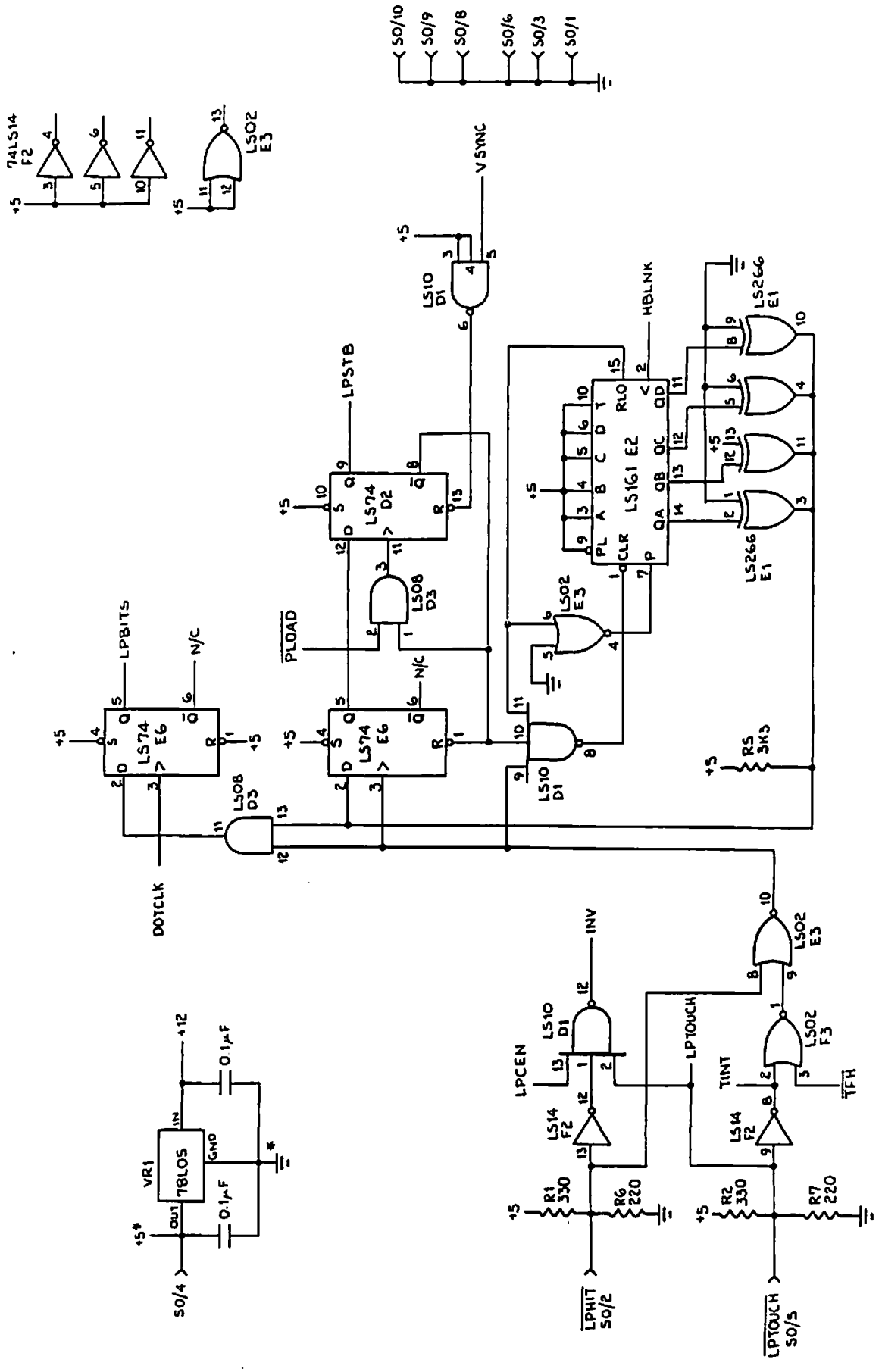


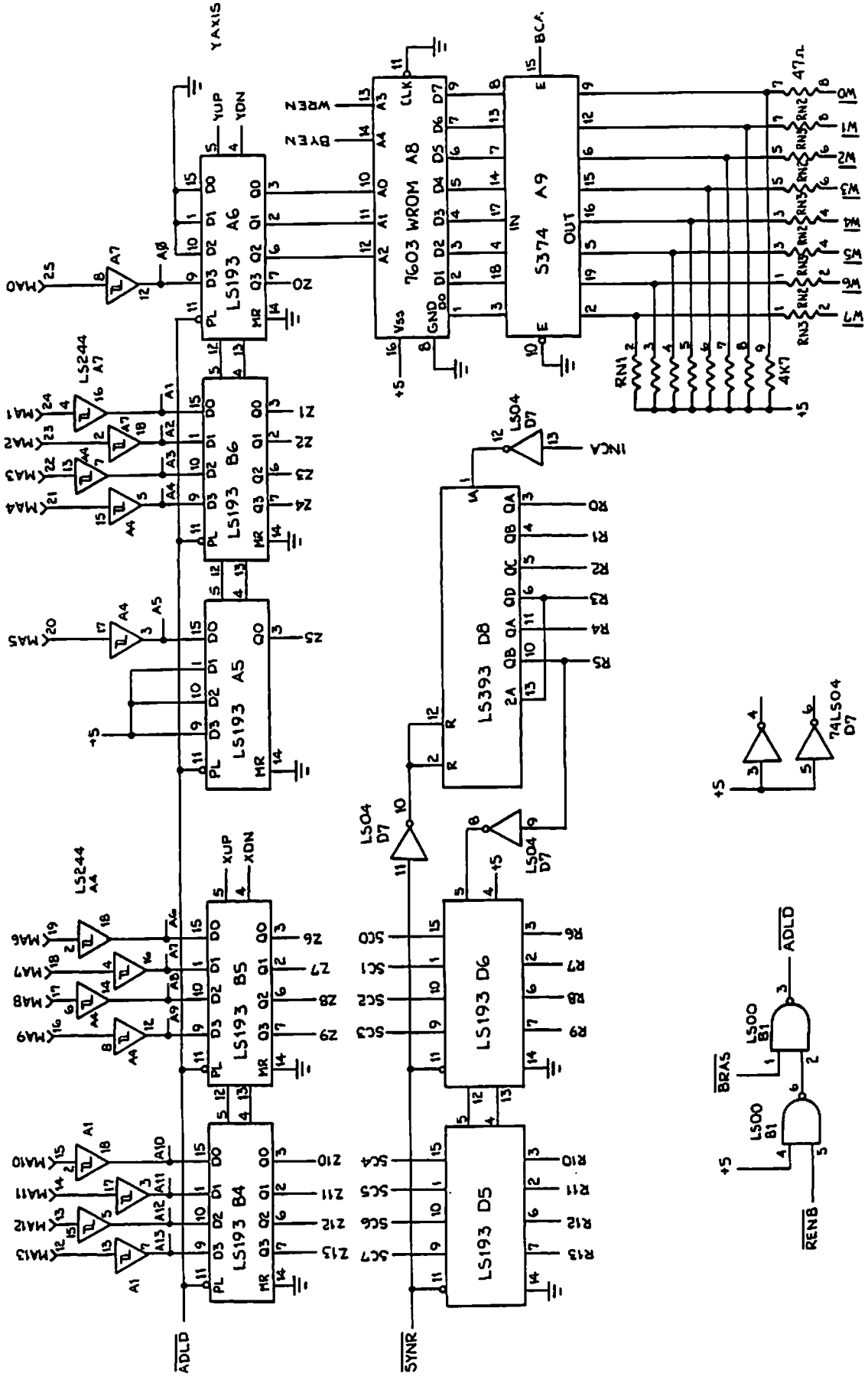


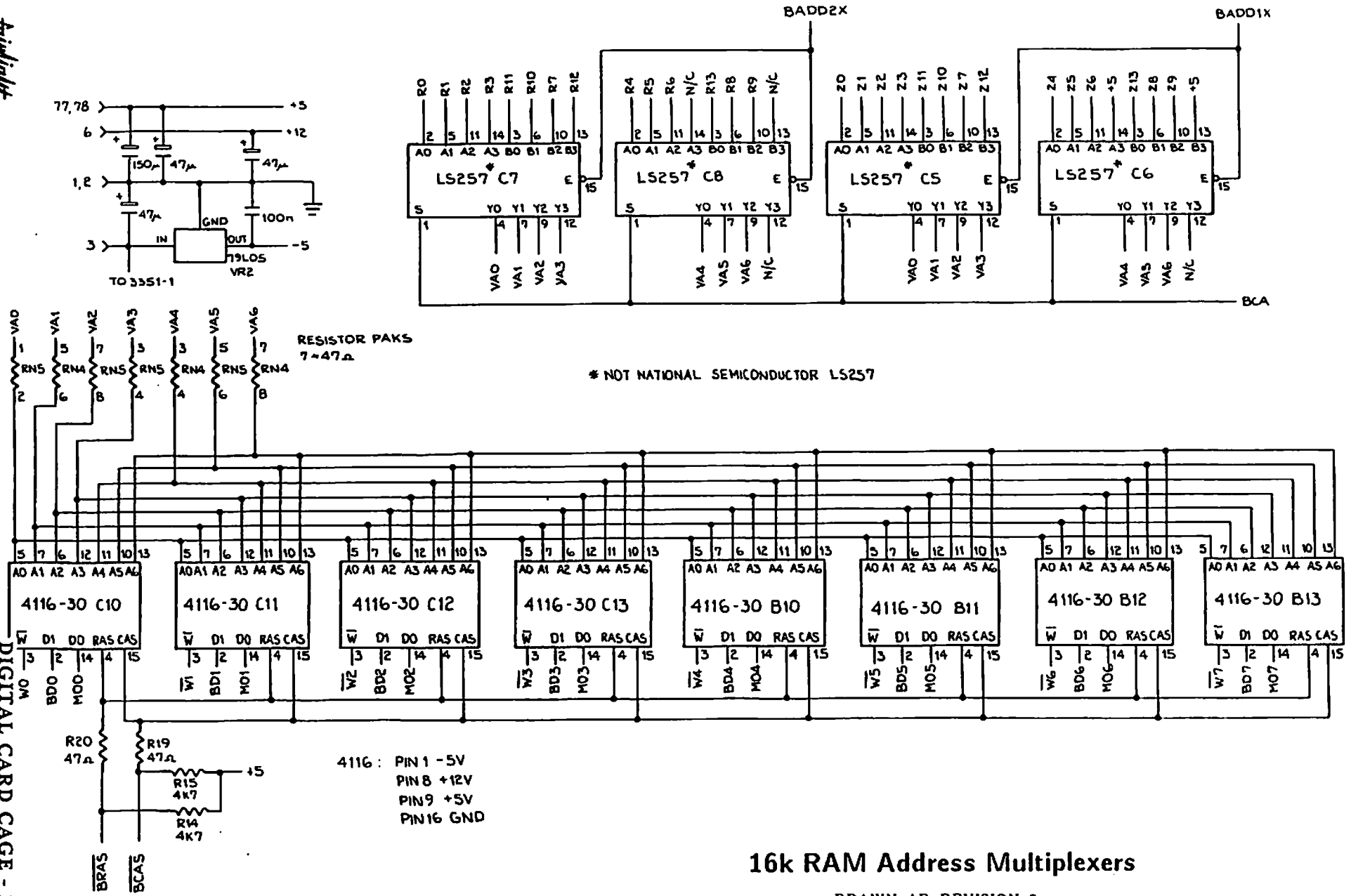
Master Clock, Video Output Data Bus Buffers

DRAWN: AB REVISION: 3


Video Control Generation
Light Pen Registers
DRAWN: AB REVISION: 3
÷304 10
LINE IN FRAME
÷ 84 10
BYTE ON LINE
LS161 D11
LS161 E12
LS161 E13
LS161 F11
LS161 F10
LS161 F9
LS374 D12
LS374 D10
LS374 D9
LS574 D13
LS574 F13
7611 SROM '5287'
E10
LS20 E11
LS04 F12
LS04 F1
LS08 F6
LS08 E9
LS27 D13
74LS04 F12
LASTL
SYNR
VBLNK
RESET
BADDIX
LC5 LC4
LPSTB
LPL2
VSYNC
HBLNK
UNCK
LPBITS
LPL1
LPLSB
PLOAD
DOTCLK
HSYNC
BLNK
RESET
N/C
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7


Lightpen Interface
DRAWN: AB REVISION: 3
SO/10
SO/9
SO/8
SO/6
SO/3
SO/1
74LS14 F2
LS02 E3
VSYNC
LS10 D1
LPSTB
LS74 D2
LS08 D3
HBLNK
LS161 E2
LS266 E1
PLOAD
N/C
LPBITS
LS74 E6
RS 3K3
DOTCLK
INV
LPCEN
LPTOUCH
TINT
LS02 F3
LS14 F2
R1 330
R6 220
R2 330
R7 220
LPHIT SO/2
LPTOUCH SO/5
TFH
VR1
78L05
0.1μF
+12
+5
SO/4


VRAM Addressing Write Control Logic
DRAWN: AB REVISION: 3
YAXIS
YUP
YDN
XUP
XDN
WREN
BYEN
7603 WROM A8
5374 A9
LS193 A6
LS193 B6
LS193 A5
LS193 B5
LS193 B4
LS193 D8
LS193 D6
LS193 D5
LS244 A7
LS244 A4
LS04 D7
74LS04 D7
LS00 B1
4K7
RN1
RN2
RN3
ADLD
SYNR
BRAS
RENB



16k RAM Address Multiplexers

DRAWN: AB REVISION: 3

Q137 Front Panel Control

2.7

Introduction

This board contains the controls used for system debugging and reset. It contains 2 push button swithes, one for RESET and one for NMI.The remaining toggle switches activate halting of each CPU and enabling of the appropriate labelled function.

Operation

The 555 timer is wired as a low frequency oscillator and is used to flash the LED associated with each "armed" switch.

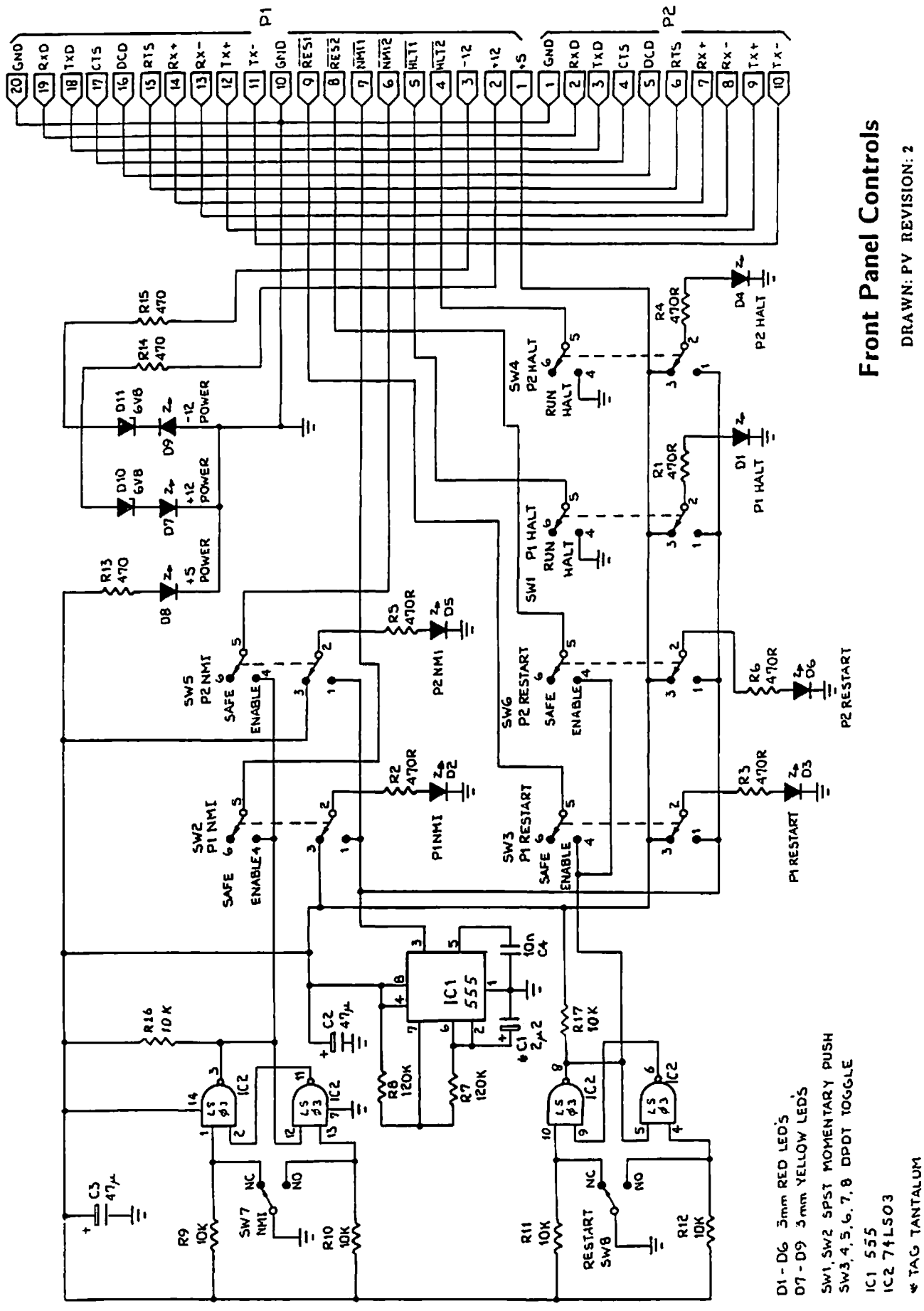
The CMOS NAND gate is used for debouncing the pushbutton switches.

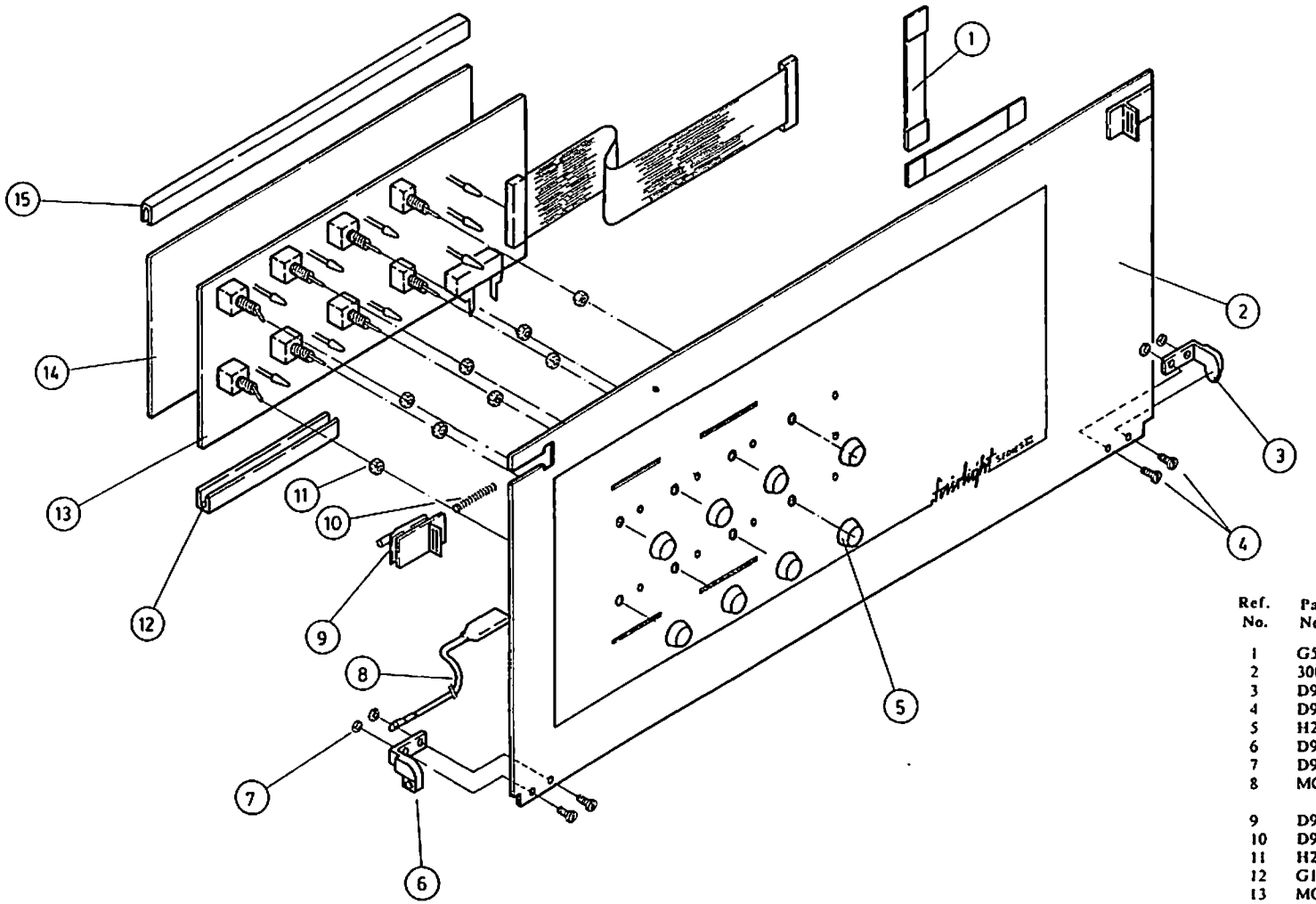
Each switch has two poles. One activates the function while the other is used to connect the oscillator to the LED.

For a function to operate the associated LED must flash.

Communications Port

The front panel also carries a 10 way flat cable connector, P2, that contains serial communications lines from the Q133 card.


P1
P2
Front Panel Controls
DRAWN: PV REVISION: 2
D1 - D6 3mm RED LED'S
D7 - D9 3mm YELLOW LED'S
SW1, SW2 SPST MOMENTARY PUSH
SW3,4,5,6,7,8 DPDT TOGGLE
IC1 555
IC2 74LS03
* TAG TANTALUM



Ref. No.	Part No.	Description
1	G5116	CLAMP, FLAT CABLE
2	300/089	PANEL, FRONT
3	D9002	HINGE BRACKET, R/H
4	D9005	SCREW, VERO, M3 X 8
5	H2006	NUT, DRESS
6	D9002	HINGE BRACKET, L/H
7	D9006	NUT, VERO, M3
8	MC020	CABLE, EARTH6Y
9	D9003	HINGE BOLT, VERO
10	D9004	SPRING, VERO HINGE
11	H2007	NUT, SWITCH
12	G1332	STRIP, SHORT PLASTIC
13	MQ137	CARD, FRONT PANEL
14	G0007	INSULATOR
15	G1332	STRIP, LONG PLASTIC

Q777 SCSI Interface

2.8

Introduction

The SCSI Interface card adapts the CMI interleaved parallel buss to a SCSI peripheral interface buss which is a high speed 8 bit parallel buss with handshaking. The interface card also contains an EPROM with appropriate driver software which the CMI operating system can load into RAM.

The SCSI Interface contains hardware to implement either Initiator operation or Target operation (as defined in the SCSI specification) under software control. DMA transfers of data between the SCSI buss and CMI memory are implemented by the interface for Initiator operation. In this mode a hard disk or similar device will be selected for either reading or writing and the data transfer will proceed at the maximum rate. The SCSI Interface may be configured to have a SCSI address in the range of 0 - 7 and an external select of this device will generate an interrupt.

The SCSI Interface will normally be used to connect to storage devices such as hard disks and streaming tapes, however any device which conforms to SCSI specifications may be attached to the interface. All communications with SCSI devices attached to the buss conform to a defined message protocol which transfers command, data and status information in distinct phases as defined in the SCSI specification.

Address Map

The Interface is accessed through two locations, FCE2 and FCE3 in a memory map which enables access to peripherals. A 3 bit address register (FCE3) is written to point to additional registers. All access to these registers is subsequently performed through location FCE2.

ADDRESS	READ	WRITE
FCE2	DATA	DATA
FCE3	STATUS	POINTER

Status Register

A read from address FCE3 will return the contents of the status register. This register indicates interrupt status, EPROM status and SCSI buss state.

bit 7 - high indicates EPROM transfer not occurring
bit 6 - high indicates an interrupt condition is present
bit 5 - MESSAGE active on SCSI buss
bit 4 - ACKNOWLEDGE active on SCSI buss
bit 3 - COMMAND active on SCSI buss
bit 2 - BUSY active on SCSI buss
bit 1 - REQUEST active on SCSI buss
bit 0 - DIRECTION active on SCSI buss

Pointer Register

A write to location FCE3 will set the pointer register to select additional registers on the interface. Only bits 1 - 3 are used by the interface. The state of other bits will have no effect.

Data Registers

There are 10 additional registers which may be addressed throug location FCE2 after setting the pointer register. Eight of these are output registers and two are input registers.

POINTER	OUTPUT	INPUT
00	control register	input register
02	DMA address low	X
04	DMA address high	X
06	DMA count low	X
08	DMA count high	X
0A	select EPROM load	X
0C	SCSI data out	SCSI data in
0E	target register	X

Control Register

This register controls DMA operation, interrupts and SCSI operations.

bit 7 - enable interrupts to the CMI
bit 6 - enable DMA operation
bit 5 - enable SCSI buffers
bit 4 - disable auto-increment on DMA address counters
bit 3 - SCSI ATTENTION
bit 2 - SCSI BUSY
bit 1 - SCSI SELECT
bit 0 - SCSI RESET

Input Register

This register provides information about the current state of the SCSI interface and also identifies the preset SCSI address of the card.

bit 7 - interrupt present due to SCSI select
bit 6 - bit 2 of SCSI address
bit 5 - bit 1 of SCSI address
bit 4 - bit 0 of SCSI address
bit 3 - SCSI ATTENTION
bit 2 - SCSI BUSY
bit 1 - SCSI SELECT
bit 0 - SCSI RESET

DMA Address Low
Low byte of DMA start address.

DMA Address High
High byte of DMA start address.

DMA Count Low
Inverted low byte of DMA count.

DMA Count High
Inverted high byte of DMA count.

Select EPROM
A write to this register will enable the logic which loads the contents of the EPROM into CMI RAM. It is necessary to set up the DMA logic before writing to this location.

SCSI Data Out
This register drives the buffers onto the SCSI data lines.

SCSI Data In
This register contains the current state of the SCSI data lines.

Target Register
This register controls the hardware implementing SCSI target operation. In normal operation this register is cleared on reset, disabling all target functions.
bit 7 - MODE active enables Target operation
bit 6 - not used
bit 5 - SCSI MESSAGE
bit 4 - SCSI DIRECTION
bit 3 - SCSI COMMAND
bit 2 - not used
bit 1 - SCSI REQUEST
bit 0 - not used

Address Decoding
Address range $FCE2 - $FCE3 is decoded by gates A1, B3, A4, B1, B2, D1, E1 and latched by F2. NAND gates E2, E2 are used to generate read and write strobes for the status and pointer registers respectively.
Bits 1, 2 and 3 from the pointer register E5 are decoded by E4 and strobed by signal DREG* from F3. Where necessary, select lines are gated with read or write signals by OR gates B8, F3.

Data Buffers
Data from the system data bus is buffered and inverted by C6 then latched by D6 which holds the data across the processor 1 phase. The latched data from D6 (I0-I7) is used by all output registers and DMA counters.
Data read onto the system data bus is taken from input registers A5, C9, C8 and C7 via data bus D0-D7.

DMA Address Counters

Sixteen bit counter chain D3, C3, D2, C2 is used to provide the address for DMA transfers. The starting address for each DMA transfer is established by writing the appropriate byte address to the pointer register followed by writing the DMA address byte to the data register. This is repeated for the other DMA address byte. When DMA transfers take place, the counter chain is automatically incremented by the signal ATB*. DMA address incrementing may be disabled under software control via bit 4 of the control register and gate F4. This feature allows disk reads and writes from a single memory location as implemented on the Floppy Disk control card.

DMA Byte Counters

Sixteen bit counter chain D5, C5, D4, C4 is used to transfer the required number of bytes to or from CMI memory under DMA control. The counters are configured to count up to $0000, so for correct operation it is necessary to load the inverse of the required count. Any number may be specified up to a maximum of $FFFF bytes. Only the specified number of bytes will be transferred to or from memory, regardless of the actual size of the transfer requested by the SCSI bus. This allows less than a sector to be read from disk saving the software overheads necessary for partial sector reads. When the counter reaches 0, the signal FINPS* is generated from ripple carry out. This signal is buffered out to the system bus as VMA by 3 state gate C1 disabling further memory accesses.

DMA Logic

DMA requests may come from either the Device Driver ROM logic or the SCSI interface. These requests are synchronised with processor 2 phase 2 by flip-flop A8. This sets up a DMA request to the processor (RDMA*). DMA cycles are granted by the BACK acknowledge signal on a daisy chain basis through ETL* and ENL*.

When a transfer occurs, the DMAC (DMA CLAIM) line is asserted so that the memory card swaps maps, allowing data to be dumped into memory currently not mapped into the processors address space.This signal is generated by flip-flop B10 and gates A10, A10. During a DMA cycle, onboard select strobes are generated by flip-flop B9 and gates A9, A10. During DMA writes either ROM buffer A5 or SCSI buffer C7 will be enabled depending on the state of flip-flop B10. DMA reads will always be directed at the SCSI port. NAND gate E2 generates the address strobe ATB* which enables the address buffers onto the CMI buss.

SCSI Bus Interface

The SCSI bus is physically a 50 way ribbon cable where each signal is terminated by a 220/330 ohm resistor network. The interface is capable of driving and receiving all SCSI signals except PARITY which is not implemented. Open collector NAND gates E6, E7, E8, E9 and D10 are used as drivers. The SCSI data bus is buffered by C7, and the remaining signals are buffered by hex inverters E10 and E11. All SCSI drivers are gated as three groups:- SCSI data, SCSI control and SCSI target control. These groups are activated by signals EDATA, ESCSI and ECONT which are derived from control latches D8 and D9. Both these latches are reset by SYRES* ensuring that the interface is disabled upon powerup.

Data transfers to and from the SCSI bus are initiated by REQ being asserted. Data is read off the bus by strobing buffer C7. Data written to the SCSI bus is latched by D7 and is held until the next write. This is necessary as SCSI data transfers are asynchronous and data must be maintained on the bus until the target releases REQ. Gates C11 and D11 ensure that data is only enabled onto the SCSI bus as requested by SCSI signal DIR. When a byte is transferred while a request is pending, the strobe sets flip-flop F11 driving ACK to complete the data transfer.

The SCSI interface can generate interrupts from two sources:- status available from target and select from external device. The second of these interrupts is only used in systems which implement multiple initiators, and may be disabled by link W4.

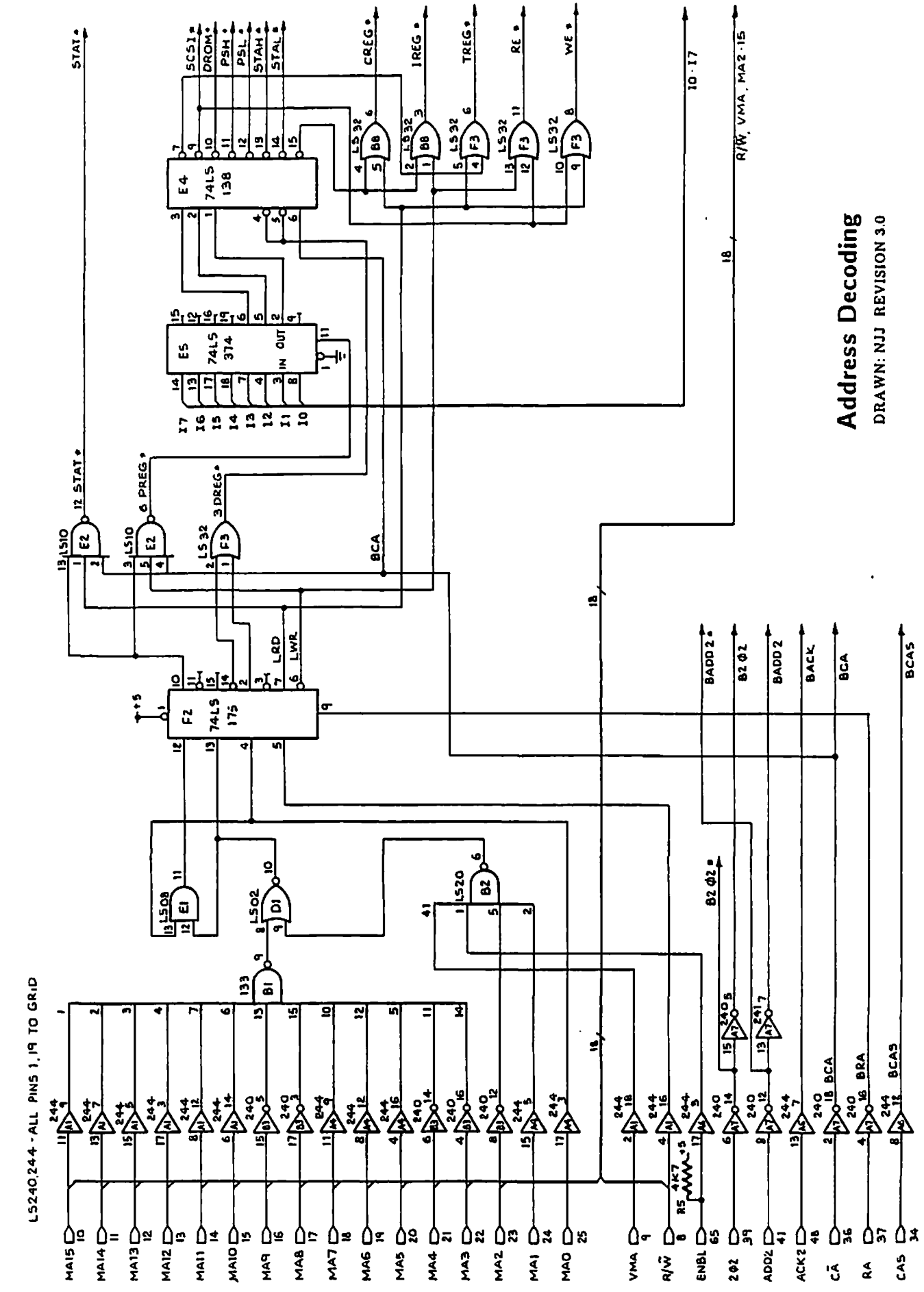
The status interrupt is latched into F11 by the SCSI bus request to transfer a command byte. The latch is enabled by EDMA going high and held clear when this signal is off.

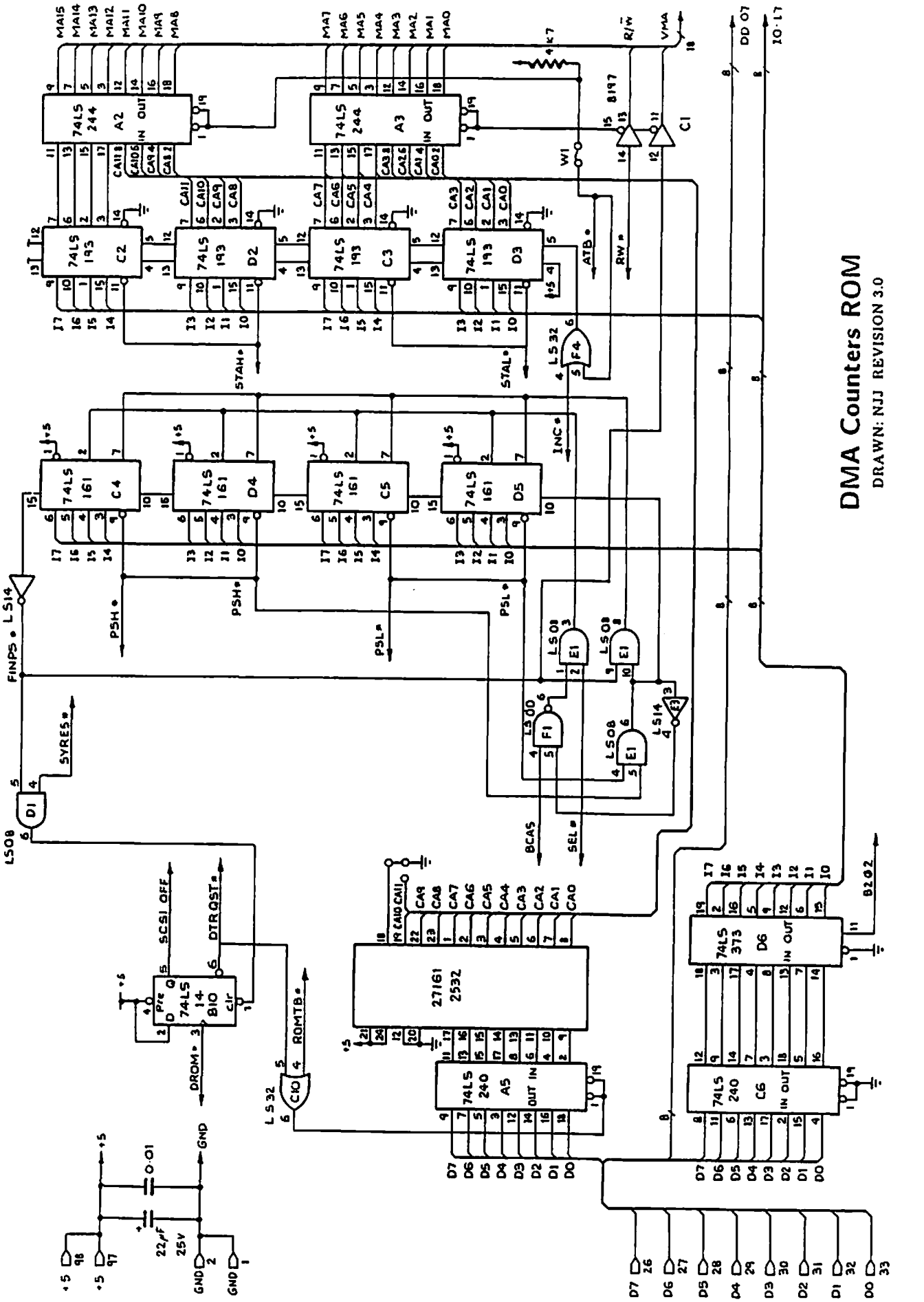
The select interrupt is generated by SEL and the SCSI DATA line set by J0-J2 being simultaneously asserted. B7, C10 and D1 are responsible for this interrupt. Both interrupts are ORed and drive gate D10 onto the CMI buss. Signal EINT is will globally disable all interrupts from this board.

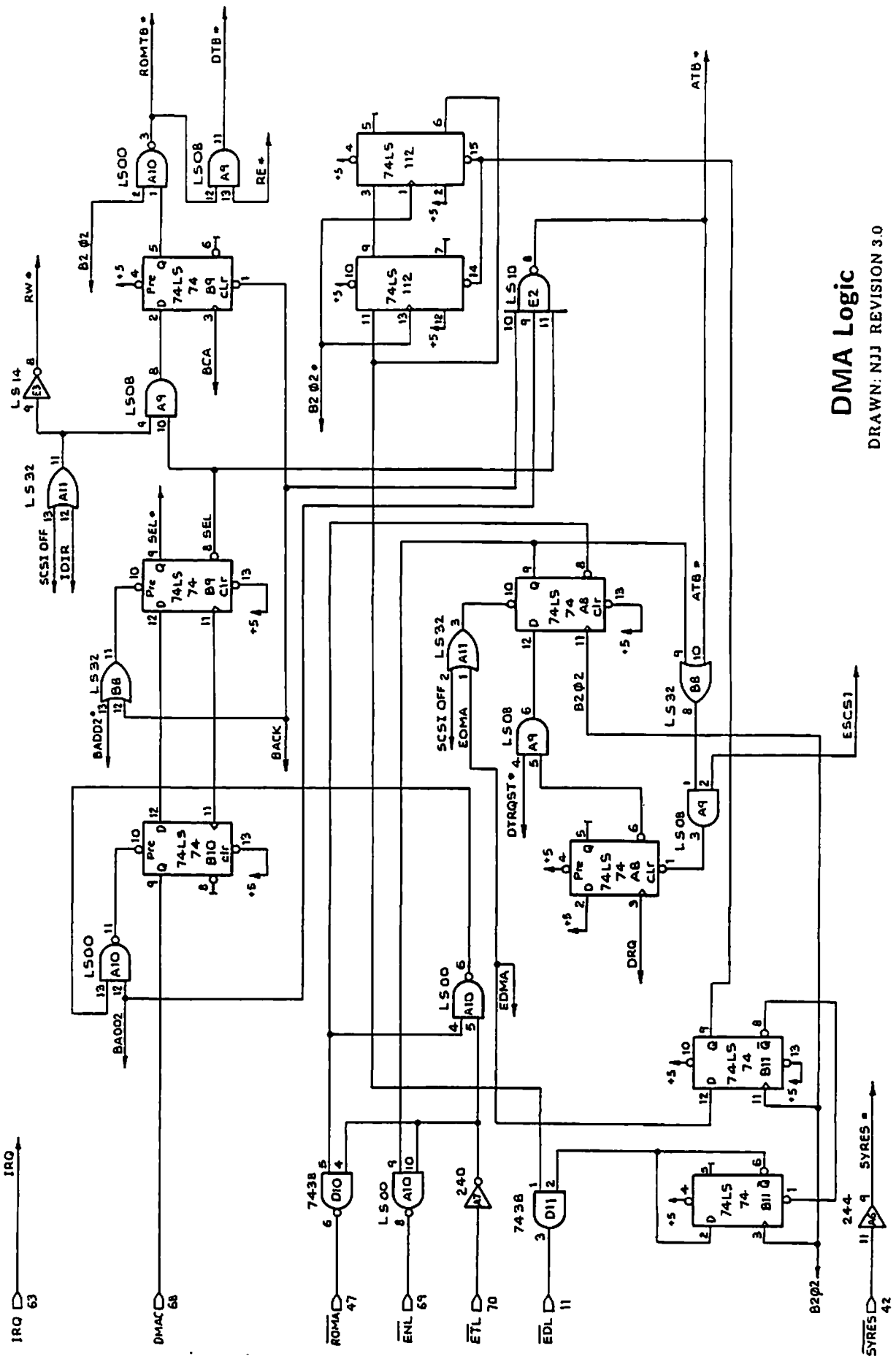
All signals associated with the SCSI interface are read through buffers C8 and C9.

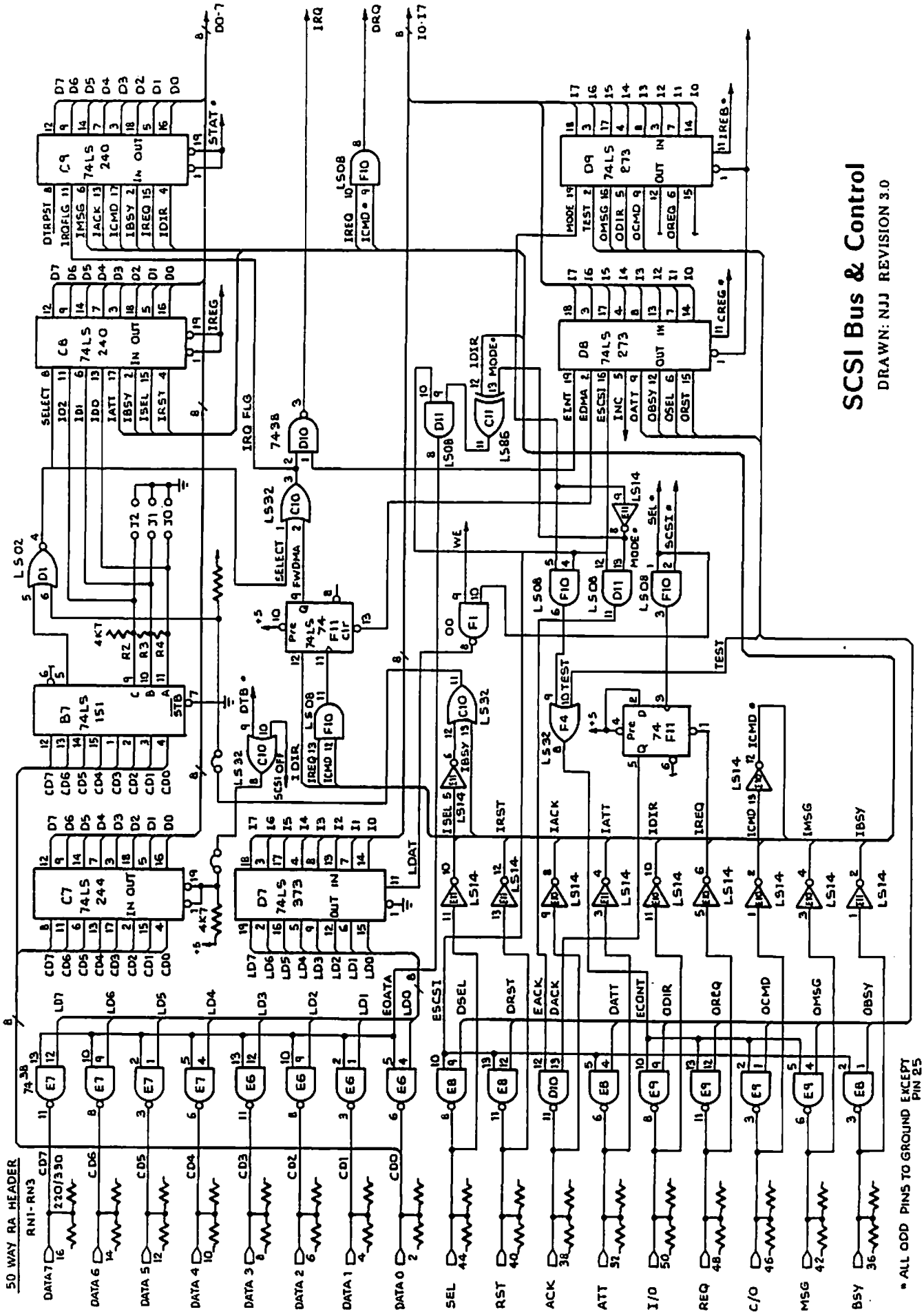
Device Driver ROM

The SCSI controller software may be placed in a 2K or 4K EPROM on the SCSI interface card. This EPROM is not in the processors directly addressable memory - it must be loaded into RAM under DMA control. The least signifigant DMA counter lines are used as addresses on the EPROM, so the EPROM will always be loaded into memory on 2K or 4K boundaries. The DMA transfer is terminated when the byte counter reaches 0000.


Address Decoding
DRAWN: NJJ REVISION 3.0
LS240,244 - ALL PINS 1, 19 TO GRID
STAT*
SCS1*
DROM*
PSH*
PSL*
STAH*
STAL*
CREG*
IREG*
TREG*
RE*
WE*
E4 74LS 138
E5 74LS 374
F2 74LS 175
PREG*
DREG*
LRD
LWR
BCA
BADD2*
B2Ø2
BADD2
BACK
BCAS
B2Ø2*
BRA
R/W, VMA, MA2-15
MA15
MA14
MA13
MA12
MA11
MA10
MA9
MA8
MA7
MA6
MA5
MA4
MA3
MA2
MA1
MA0
VMA
R/W
ENBL
2Ø2
ADD2
ACK2
CA
RA
CAS

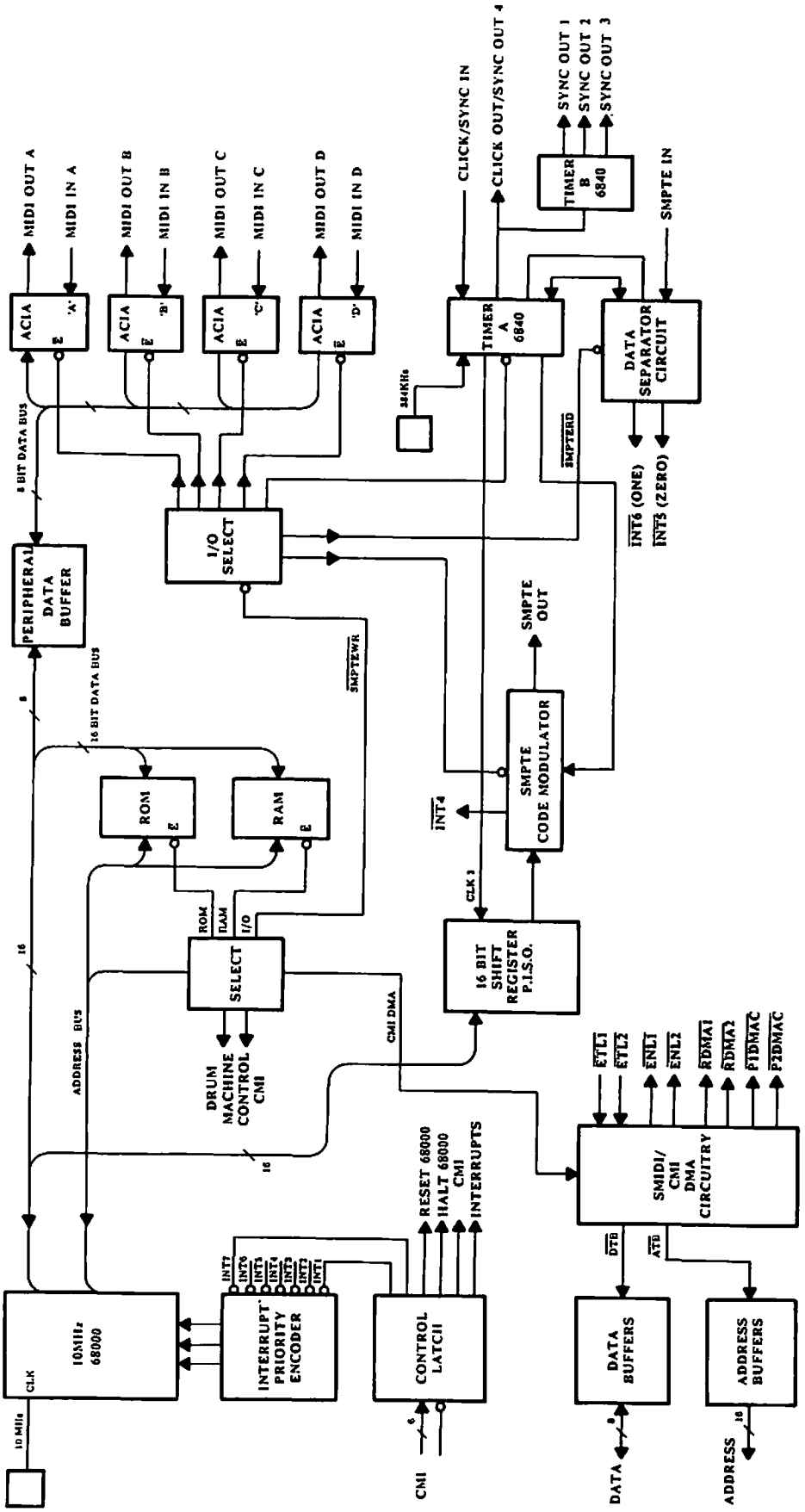

DMA Counters ROM
DRAWN: NJJ REVISION 3.0


DMA Logic
DRAWN: NJJ REVISION 3.0
ROMTB*
DTB*
RE*
ATB*
B2 Ø2
B2 Ø2*
RW*
BCA
SEL*
SEL
SCSI OFF
IDIR
BADD2*
BACK
SCSI OFF
EDMA
DTRQST*
ESCS1
DRQ
EDMA
BA002
IRQ
DMAC
SYRES*
B202
74LS 74 B9 Clr
74LS 74 B10 Clr
74LS 74 A8 Clr
74LS 74 B11 Q
74LS 112
LS00 A10
LS08 A9
LS14 E3
LS32 A11
LS32 B8
LS10 E2
7438 D10
7438 D11
240 A7
244 A6
IRQ 63
DMAC 68
ROMA 47
ENL 69
ETL 70
EDL 11
B202
SYRES 42


SCSI Bus & Control
DRAWN: NJJ REVISION 3.0
50 WAY RA HEADER
RN1-RN3
220/330
4K7
* ALL ODD PINS TO GROUND EXCEPT PIN 25
DATA7
DATA 6
DATA 5
DATA 4
DATA 3
DATA 2
DATA 1
DATA 0
SEL
RST
ACK
ATT
I/O
REQ
C/D
MSG
BSY
DO-7
IO-I7
IRQ
DRQ
IRQ FLG
LDAT
EDATA
ESCSI
DSEL
DRST
EACK
DACK
DATT
ECONT
ODIR
OREQ
OCMD
OMSG
OBSY
IACK
IATT
IDIR
IREQ
ICMD
IMSG
IBSY
IRST
ISEL
TEST
WE
DTB*
SCSI OFF
IDIR
SEL*
SCSI*
MODE*
IREB*
CREG*
STAT*
C9 74LS 240
C8 74LS 240
C7 74LS 244
B7 74LS 151
D7 74LS 373
D9 74LS 273
D8 74LS 273
E7 74LS 38
E6
E8
E9
D1 LS02
D10 7438
C10 LS32
D11 LS08
C11 LS86
F1 74LS74
F11
F10 LS08
F4 LS32
LS14
R2
R3
R4
+5
J2
J1
J0

CMI-28 2.9
General Interface Card

Block Diagram



Terminology
SMIDI Card: General Interface card or SMPTE/MIDI card.
CPU: Dual 6809 processor (P1 & P2) system usually running OS9.
System RAM: Q256 256K memory with MMU used by the CPU.
68K: Refers to the SMIDI processor - a Motorola 68000.
Hexidecimal numbers are in the form nnnnnnh.
Flip-flops packaged in 74LS74's are referred to as *a* - pins 1 to 7, and *b* - pins 8 to 13.
Titles of the form CMI-28-x refer to the circuit diagrams of the SMIDI card, x being the sheet number (1 to 7). Use the diagrams in conjunction with this text.

Introduction

The SMIDI Card plays a central role in the Series III CMI, for all the sequencing is controlled by this card. Of course it has its other function of interfacing the CMI to other digital musical instruments via MIDI and synchronizing sequences to film and video via SMPTE and Click/Sync tracking.

The SMIDI card plugs into slot 18 of the Series III motherboard next to the Waveform Processor Card. A 26-way cable connects the card to two cards (CMI-332, CMI-333) in the Audio Rack.

The CMI-332 contains the MIDI input and output buffers and the DIN sockets. Drum machine controllers and multi-sync outputs also are on this board.

The CMI-333 contains the SMPTE analog circuitry (input/ouput) as well as Click In and Out and Metronome Out.

The facilities built in to the SMIDI card are as follows:

- 10MHz 68000 processor
- ROM 8K/16K x 16 bits.
- Private RAM 8K/32K x 16 bits.
- DMA interface to the CPU buss
- 7 bit control latch
- 4 ACIA (68B50)
- 2 PTM (68B40)
- Circuitry for SMPTE reading and generating
- Drum machine controllers

ROMs

The pair of on-board 8-bit ROMs are arranged in parallel to provide 16 bit code. Presently 2764 ROMs (called *KMON-E* and *KMON-O*) are installed, occupying the bottom 8K words of 68K address space (see memory map). The bottom 400h bytes are reserved for 68K exception vectors including reset PC and Supervisor Stack Pointer which are loaded when the processor comes out of reset. The rest of the ROMs are occupied by the 68K monitor, 68K/CPU IO routines, and self tests. Refer to software documentation for further information on these items.

Private RAM

The 8K (or 32K) words of Private RAM on the CMI-28 is accessible only by the 68K. Its base address is 80000h (see memory map).

CPU DMA Interface

The two channels of the DMA interface to the CPU bus appear as two 64K slices of the 68K's 16M memory space. Accesses in the range 40000h to 4FFFFh steal P2 cycles, and select the P2 DMA memory map. Accesses in the range 50000h-5FFFFh steal P1 cycles, and select the P1 DMA memory map. Appropriate initialization of the Q256 memory maps thus allow the 68K to access any physical area of system memory or peripherals whether or not they are mapped into the CPUs' logical spaces. Refer to Q256 documentation for details about system memory management.

Data size mismatch between the 68K and the CPU bus is handled by the hardware. A 16-bit access by the 68K is handled by two separate 8-bit DMA transfers across the CPU bus and the 68K receives one *Data Transfer Acknowledge* ($\overline{\text{DTACK}}$) when both transfers have been completed. Which byte goes first is indeterminate; in any case each byte is always transferred to or from the right place in CPU space.

Access by various devices to the CPU bus is arbitrated by two daisy chains, one for each 6809. Higher priority devices may prevent access by the SMIDI to the CPU bus indefinitely. Refer to a separate document which describes the daisy chain allocation for Series III.

ACIA's and Programmble Timers

The SMIDI card has four ACIA's (68B50) for MIDI input and outputs. They can be access by the SMIDI processor at addresses 60020h to 60050h. The two timers (68B40) can be accessed by the SMIDI processor at addresses 60000h and 60010h.

SMPTE Reading and Generating

To generate SMPTE the 68k processor writes the appropriate SMPTE word to a Parallel-In Serial-Out shift register which is clocked according to the second timer in Timer A which divides a 384kHz clock. A further section of the circuitry reads SMPTE, the timing in this case is controlled by the first timer in Timer A.

Miscellaneous Items

The indivisible Read-Modify-Write instruction TAS is not supported by any of the memory interfaces on the SMIDI. The address strobe signal $\overline{\text{AS}}$ is used as the bus cycle terminator and as this is not negated in between the read and write cycles of the RMW instruction, the 68K would hang.

A no-wait state memory access is executed by the 68K in 8 states, or 400nS at 10MHz.

Processor, ROMs, and Decoding (ref. CMI-28-1)

The 10MHz processor clock (PCLK) is generated by the 20MHz crystal oscillator and made symmetrical by the flip-flop D10*b*. Detailed description of the operation of the 68K can be obtained from the Motorola literature but the following is a very brief indication of how 68K buss cycles proceed. A cycle is initiated by driving the address onto the 23 address lines A1 to A23 following the falling edge of the clock and asserting the address strobe $\overline{AS}$ 50nS later, by which time the address lines should be stable. There is no A0 address line signal. A1 to A23 determine which 16-bit word in memory is to be accessed. Whether the high byte or the low byte or both is to be accessed is determined by the *Upper and Lower Data Strobes*, $\overline{UDS}$ and $\overline{LDS}$, so the least significant address bit A0 is implied by these two data strobes. With a 24 bit effective address width, the 68K can access 16 megabytes of memory.

In a read cycle, $\overline{UDS}$ and/or $\overline{LDS}$ are asserted at the same time as the address strobe. For a write cycle, the processor puts its data on the buss 50nS after $\overline{AS}$ and asserts the data strobe(s) 50nS later still, by which time the data should be stable.

Nothing happens now until the accessed device returns a *Data Transfer Acknowledge* ($\overline{DTACK}$) signal. The processor samples $\overline{DTACK}$ and recognises that it has been asserted on the falling edge of the clock. One clock cycle later, on the next falling edge, the buss cycle is terminated. In the read case, the data is latched into the processor, and address and data strobes negated. In the write case, the strobes are negated then the data buss tri-stated 50nS later. Each half clock cycle which the processor spends waiting for the return of $\overline{DTACK}$ is called a wait state.

The upper seven address lines A17-A23 are used in the decoding circuitry E2 (LS32) and E12 (LS259), which divides the address space as shown on the memory map. All decoder outputs depend upon $\overline{AS}$ being asserted so the entire address is stable whenever any output is active. The first 128K bytes is to select ROM's, the second 128K bytes is unused. The next 128K bytes is occupied by the DMA to CMI memory via P1 or P2. The ACIA's and Timers are addressed next, then the private RAM.

The ROMs must reside at the bottom of the memory space because after reset the 68K boots up by fetching its supervisor stack vector and restart program counter from the locations 0 and 4 respectively. 450nS ROMs are currently used requiring the insertion of 6 wait states (each wait state is 50nS long) into each ROM access cycle. The counting of the wait states is performed by the LS161 counter at F1. While $\overline{EPROM}$ is not asserted, the counter is held preset at 13 but when $\overline{EPROM}$ comes active, the counter is released while the ROM(s) are enabled. On the second rising edge of the clock after this, the counter reaches 15 and generates the ripple carry out which clocks the flip-flop at G1*b* to assert $\overline{DTACK1}$. When the processor terminates the cycle, $\overline{AS}$ is negated so BAS goes low and clears the flip flop to remove $\overline{DTACK}$. $\overline{EPROM}$ is negated so the LS161 is put back into preset ready for the next ROM cycle.

The delay of $\overline{\text{EPROM}}$ through the decoding circuitry plus the two counted clock cycles, plus the delay between $\overline{\text{DTACK}}$ and the processor latching the data in provides the required 450nS access time. The complete cycle takes 700nS from bus inactive to bus inactive again.

Memory - ROM and RAM
(refer schematic CMI-28-2)

There are four 28-pin sockets for ROM and static RAM. The minimum configuration is 8k words of ROM (2x2764)and 8k words of static RAM (2x6264). The ROM can be configured for 16k words by breaking the link (LK1) between pin 27 of the ROM's and +5v and join the link LK1 to A15 from the processor and plugging in the appropriate two 27128's. Similarly the RAM can be arranged to accommodate 32k static RAM chips (e.g.MK4856 pseudo-statics) by breaking the links LK2 and LK3 to +5v and connecting A14 and A15 to pins 26 and 1 of the RAM chips (via LK3 and LK2), respectively. Further, there is an option for 64k words of RAM; by soldering two 32k RAM chips on top of each other except for pin-20, the chip select, which should be connected to the pads provided from the select circuitry, the OR gates pins 3 and 11 of D12 (LS32). All these memory expansions will depend on the availability of the chips mentioned.

Note; when plugging in the ROM's, they should be labelled *odd* and *even*; the even one should be plugged into E5,6 (near the 68K) and the odd one into E8 (between the RAM chips).

Memory addressing: ROM starts at 000000h and RAM starts at 080000h.

Static RAM is fast enough (150ns) to not need a delay on the $\overline{\text{DTACK}}$ line. So that when RAM is selected $\overline{\text{DTACK}}$ is also enabled.

Control Latch and Interrupts
(refer schematic CMI-28-3)

The 68K can be reset, halted and interrupted on two levels via a write only latch at FCA0h in the CPU address space. This address is decoded by the large NAND gate B3. The decoder output is latched by flip-flop B2*b* on the rising of BRA and data is written into the LS259 latch B4 at the end of the system CAS pulse. The 74LS14 (A5) permanently buffers all incoming data to the latch.

The first 3 data bits from the CPU (D0-D2) are used as an address for the 74LS259 and the fourth data line D3 provides the one-bit data for this latch. With this arrangement if several processors attempt to write to the SMIDI latch there will be no clash.

The bit assignments are as follows:

Latch Address;			Data;	Function;
D2	D1	D0	D3	- Active for:
0	0	0	0	68K Interrupt Level 1
0	0	1	-	n/c
0	1	0	1	CMI Interrupt 2
0	1	1	0	68K Interrupt Level 7
1	0	0	1	CMI Interrupt 1
1	0	1	0	Halt 68K
1	1	0	0	Timer Sync switch
1	1	1	0	Reset 68K

All the interrupts to the 68K from the control latch, the ACIA's, Timers and SMPTE circuitry are fed to the priority encoder C2 (LS148) its 3-bit output is fed directly to the 68K processor. Level 7 (all bits low) is the highest priority (and non-maskable) while level 0 (all bits high) means no interrupt. Interrupts are cleared by the 68K writing either to the control latch itself via the CPU bus (to clear interrupts 1 and 7), or reading ACIA or Timer status or to address 60060h for SMPTE read, or to 60070h for SMPTE write.

To reset the processor, both $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ must be asserted. The minimum reset period required by the 68K is 10 clock cycles i.e. 1uS.

The state of the $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ signals are indicated by the LEDS near the front of the SMIDI card. Both off means that the processor is running.

System Reset ($\overline{\text{BSYRES}}$) clears the latch, thus putting the 68K into reset. The level 7 interrupt also generated will be ignored while the processor is reset. A manual control panel may be plugged into 26-way socket on the SMIDI board with a $\overline{\text{HALT}}$/RUN toggle switch and RESET momentary switch.

The 68K has a very powerful interrupt vectoring system which permits the interrupt service routine vector to be provided by the interrupting device. Many such devices may therefore generate interrupts without the need for extensive polling procedures to find out which device is requesting. This facility is not required on the SMIDI and the only interrupts used are the seven Auto-vector interrupts determined by the interrupt level on $\overline{\text{IPL0}}$-$\overline{\text{IPL2}}$. The function code outputs FC0-FC2 are all high during an interrupt vector fetch and this is detected by the AND gate 74LS11 (D2), on drawing CMI-28-1. This results in $\overline{\text{VPA}}$ being asserted which tells the processor to use the autovector instead of an interrupting device-supplied one.

68k/6809 DMA Buss Interface

(refer schematic CMI-28-4)

Communications between the 68k processor and the 6809 CPU is achieved by DMA *Direct Memory Access* on the system buss. The 68k waits until no higher priority device is occupying the buss and then either 6809 (P1 or P2) is temporily hung while the 68k executes a normal buss cycle writing to or reading from memory or a peripheral on the buss. In this manner the entire 64k address space of each 6809 processor appears as a small slice of the 16 megabyte address space of the 68k. Software then defines various protocols for the different processors to pass messages and data to one another by simply placing them in system memory.

The DMA interface provided on the 68K SMPTE/MIDI Card is a very flexible one. It automatically handles either 8 or 16 bit data transfers (doing double cycles across the 8-bit CMI buss in the latter case) and can do so on either P1 or P2 cycles, selecting any desired memory mapping which has been set up on the Q256 memory card.

DMA is initiated by the 68K when it accesses any address in the range 040000h to 05FFFFh. These addresses are decoded by the LS259 (E12) on drawing CMI-28-1 and result in the $\overline{\text{CMI}}$ signal being asserted (low). Since the rest of the interface circuitry is not activated yet, PACK (to be explained later) will be low and a low will be presented at the data input of flip flop C12*a* whose function is to synchronise the transfer with the CMI buss.

Address line A16 is used to select which 6809 processor's buss cycle(s) are to be used for the transfer. The timing signals for both processors are input to LS241 buffer A7 which is wired as a multiplexer:-

If A16 is low, P2ø2 is enabled through to become Pø2, ADD2 becomes PADD and so on.

If A16 is high, P1's timing signals are enabled instead.

By this means, the address range specified above is split in two: from 040000h to 04FFFFh the transfer automatically occurs on P2 buss cycles, while from 050000h to 05FFFFh it occurs on P1 cycles. Refer to the 6809 CPU doc cycles.

Thus at the beginning of the data cycle of whichever processor is selected, the Pø2 signal clocks the LS74, recording the fact that a DMA cycle is required.

All DMA devices are interconnected on the motherboard in a *daisy chain*. Each device is assigned a given priority in the chain and must wait until no higher priority device is already using the buss. The 6809 CPU is the always the last device in the chain. There are two separate daisy chains in the CMI system, one for each 6809 CPU. Since the 68K SMIDI card can perform DMA on either CPU's cycles, it is a member of both chains. $\overline{\text{ETL1}}$, $\overline{\text{ENL1}}$ and $\overline{\text{RDMA1}}$ are the chain signals for P1, $\overline{\text{ETL2}}$, $\overline{\text{ENL2}}$, $\overline{\text{RDMA2}}$ are for P2. Which set is used is again selected by the state of A16 at the time of transfer.

The selected $\overline{\text{ETL}}$ *Enable This Level* signal is low when no higher priority device is occupying the buss. After the $\overline{\text{CMI}}$ signal has been latched, nothing happens until this signal is low, whereupon the $\overline{\text{RDMA}}$ *Request DMA* is driven low through the selected LS10 gate. Any DMA device pulls this open collector line low to to request buss access to the CPU. At the same time, the selected $\overline{\text{ENL}}$ (Enable Next Level) signal is inhibited. Normally, the low on $\overline{\text{ETL}}$ comes in and goes out again on $\overline{\text{ENL}}$ to indicate to lower priority devices that the buss is available but when the 68K requires a transfer $\overline{\text{ENL}}$ is held high to hold up the lower devices.

The CPU acknowledges that it will hang and release the buss for the next cycle by asserting ACK1 or ACK2; the selected ACK signal becomes PACK. When a request has been generated (C12*a* $\overline{Q}$ hi) and this level is enabled (ENL low), the rising edge of PACK clocks a low into flip flop B11*a* to generate DCYCLE. This signal indicates that the next buss cycle is definitely going to be a 68K DMA transfer and remains asserted until the end of the address phase of the actual DMA cycle.

The other half *b* of B11 is also clocked by PACK to generate the P1 or P2 DMAC *DMA Claim* signal as selected by A16. This signal goes to the Q256 RAM card to select the memory mapping which has been set up specifically for the 68K. In this way the 68K may have access to part or all of the same physical memory space as the 6809 CPU or it may have access to an entirely different part of physical memory as required by software. The DMAC signal is asserted during the data cycle preceding the actual transfer.

The address phase of the DMA cycle is indicated when ATB *Address To Bus* is asserted by the LS10 B10. At this time the lower 15 bits of the 68K address buss are enabled on to the CMI buss through the two LS244's A2 and A3 to select the required location within the 6809 address space. VMA is driven high through LS125 B1 to indicate a *Valid Memory Address* and the 68K R/W line is driven through the same buffer to indicate a read or write cycle.When the 68K performs 8-bit memory accesses, the UDS and LDS signals (*upper* and *lower data strobes*) indicate whether an even or odd address is being accessed. The sense of these signals are clocked into JK flip flop H12*a* at the beginning of DCYCLE to generate HIBYTE and LOBYTE. The latter signal becomes the least significant address line driven onto MA0 through A3. In the case of 16-bit accesses, the hardware automatically requests two successive DMA accesses across the 8-bit CMI buss. Both UDS and LDS are asserted so that the JK outputs HIBYTE and LOBYTE simply toggle on each access. It does not matter which byte transfers first and in fact this depends on the initial state of H12*a*. LOBYTE directs the data to or from the odd or even address and both signals control whether the higher or lower 8 data lines are directed to the data buss.

The data buss interface consists of Schmitt bidirectional buss transceiver LS640 A6 and bidirectional driver/latches E,D5 and E,D6 (LS646s). The data phase of the DMA transfer is indicated by the assertion of DTB *Data To Bus* at the rising edge of BRA when a DMA cycle is in progress. This is performed by flip flop C12*b*. DTB enables the buss transceiver A6 and the direction is determined by the 68K R/W signal.

If the 68K is writing to the CMI buss, E,D5 or E,D6 simply act as buffers to transfer the high or low 68K data signals (PD0-15) through to A6. HIBYTE or LOBYTE plus CMI being asserted will drive the G input of the appropriate LS646 for the duration of the DMA cycle (LS02 and LS32 gates B8 and A8).

When the 68K reads from the CMI buss, E,D5 or E,D6 must latch the data in from the buss to hold it until the 68K terminates its own cycle and latches the data internally, about 50nS after the end of the DMA cycle. 100nS before the end of the data phase, the CMI timing signal CAS goes low, resulting in a rising edge on $\overline{BCAS}$. Data from memory is guaranteed to be valid at this time. B10 generates the LDATA *Latch Data* signal which is ANDed with either HIBYTE or LOBYTE to latch the data coming into the A side of E,D5 or E,D6. The output of the latch (B side of the selected LS646) is driven onto the PD lines until the 68K completes its cycle and negates $\overline{CMI}$.

Termination of the transfer after single or double DMA cycles is controlled by the two flip flops in LS74 C10:

(i) In the single (8-bit) transfer case, either UDS or LDS will be low. This will cause the LS10 A10 to output a high, and $\overline{DTACK2}$ will be generated as soon as LDATA occurs. The 68K will then terminate its cycle immediately, after only one DMA cycle.

(ii) In the double DMA cycle (16-bit) case, both $\overline{UDS}$ and $\overline{LDS}$ are high so $\overline{DTACK2}$ will not be generated until the first flip flop in C10 is set. Initially this flip flop is reset. At the first LDATA pulse a high is clocked in but $\overline{DTACK2}$ is not generated because of the propagation delay through to the next flip flop. Since $\overline{DTACK2}$ is not asserted, the 68K still waits with address and address/data strobes asserted.

If writing, the data remains asserted by the 68K but both address and data are removed from the CMI buss when $\overline{ATB}$ and $\overline{DTB}$ are negated respectively. If reading, the first byte read in is latched and held by E,D5 or E,D6. Since $\overline{CMI}$ will still be asserted and PACK will have been negated, the whole process of waiting for daisy chain priority and DMA requesting begins again in order to perform a second DMA cycle. The second cycle can be held up indefinitely by higher priority devices using the buss after the first cycle. When the second LDATA edge comes along the high on the LS10 output is clocked into the second C10*a* flip flop and $\overline{DTACK2}$ is asserted.

On the next falling edge of PCLK, the 68K recognises that $\overline{DTACK}$ has been asserted. On the second falling edge of PCLK the data is latched internally for a read, and the address and strobes are released. The low on BAS resets the flip flops at C10.

Debugging Notes for the DMA Circuitry

If the timing circuitry of the DMA interface is faulty, the most likely result is that $\overline{DTACK2}$ will never be generated and the 68K will simply hang which makes debugging easy. In this case, check first that the address decoding is generating $\overline{CMI}$, then that the daisy chain signals are present. Then look for an 800nS pulse on $\overline{DCYCLE}$ (pin 5 B11*a*), indicating that DMA cycles are actually ocurring. Continue through to the $\overline{ATB}$, $\overline{DTB}$ and $\overline{LDATA}$ signals, checking not only that they are generated but also that they get to their respective destinations in the circuitry.

If the DMA cycles are being synchronised and timed correctly check that the address buffers and data buffer/latches are being enabled and clocked at the correct times.

If all timing circuitry is correct, the last possibilty is data or address buss shorts, open circuits or faulty drivers. Special test ROMs are available which cause the 68K to repetitively copy bytes and words from one location to another in CMI memory. The 6809 monitor can then be used to deduce which data or addresses cause problems.

SMIDI Card Peripheral Select
(refer schematic CMI-28-5)

The ACIA's and Timers work from an 8-bit data buss with (asynchronous) interfacing circuitry.
Initially the flip-flops (F2) are cleared causing a high $\overline{\text{DTACK3}}$ output setting the LS646 transceiver (G4) into the transparent mode. The direction of data flow is determined by the R/W line with the IO selected. Without IO line selected it appears in write mode. The peripheral is selected by the LS138 enabled by the $\overline{\text{CS}}$ signal. The first flip-flop F2*a* is clocked on the first falling edge of E with the IO select and the data strobe high (ie either $\overline{\text{LDS}}$ or $\overline{\text{UDS}}$ low). The Q output of F2*a* is applied to the NAND gate (H1), asserting $\overline{\text{CS}}$. Selecting the peripheral at this time ensures that the peripheral has adequate address setup time.

On the next falling edge of E, the $\overline{\text{Q}}$ output of F2*b* is clocked low asserting $\overline{\text{DTACK3}}$ and latching data in the transceiver (G4) The asserted $\overline{\text{DTACK3}}$ signal deselects the peripheral by causing $\overline{\text{CS}}$ to go high. Flip-flop F2*a* is cleared by IO going low when the access terminates. Clearing flip-flop F2*a* also initializes the interface circuitry for the next access.

The ACIA's are selected by E10 (LS138) and appear at addresses; 60020h, 60030h, 60040h and 60050h. They share a common interrupt level (level 3). Their transmit and receive data lines are wired to the 26-way connector to be connected to the MIDI drivers and opto-coupler receivers.

The programmable timers appear at the general addresses; 60000h, and 60010h has an interrupt to the 68K (level 2, $\overline{\text{INT2}}$).

SMIDI ACIA's and Timers
(refer schematic CMI-28-6)

There are four different peripheral circuits on the SMIDI card. Firstly there is the four ACIA's (68B50) (G7-11) which are the MIDI ports A,B,C, and D, respectively. Then there is the Timer (68B40) (G5,6), *Timer A* which is used in conjuction with the SMPTE read and generate circuits (which are the other two peripheral circuits) as well as the Click In and Out. There is another Timer (68B40) (H,G3), *Timer B* which does the Multi-sync outputs.

The ACIA's and Timers are driven also by the E (enable) signal from the 68K. The frequency of this clock is one-tenth of the 68K clock (10MHz) with a 60/40 duty cycle (6 clocks high, 4 clocks low)

The Sync switching circuitry consisting of the EXOR gate acting as an inverter (C1) and the two switched gates (LS125) (B1) provide a facility to *divide by one*, which the Timers do not have. When the SYNCSW line is low (from the Control Latch (B4)) the signal coming

in at Click In will be connected to Sync Out 4 directly as well as clocking the third timer in Timer A (G5,6) and clocking all timers in Timer B. When SYNCSW is high the clock signal coming in at Click In is first divided by timer 3 in Timer A before clocking the three timers in Timer B.

The ACIA's can interrupt the 68K on interrupt level 3, ($\overline{INT3}$).

The Timers can interrupt the 68K on interrupt level 2, ($\overline{INT2}$).

SMPTE Generating Circuitry

(refer schematic CMI-28-7)

An oscillator (3.84MHz) is divided by 10 (G2,H1,G1*a*) to provide a standard for generating the 3 different rates of SMPTE code (24,25 and 30 fps) All three are denominators of 384,000. Further division, depending on the frame rate selected, is done by the second timer in Timer A(G5,6) giving the signal CLK2, which is the bit rate for a SMPTE *one* (ie 160 bits per frame). This is in turn divided by 2 (C11*b*) giving CLK1 which is the bit rate for a SMPTE *zero* (ie 80 bits per frame). When a SMPTE word is ready it is written to the Parallel-In-Serial-Out registers LS165 (C7,C8) at address 60070h (through B9,B8 and B7). When this writing takes place the interrupt $\overline{INT4}$ if it has been asserted is now cleared. The data in the shift registers (C7,C8) is clocked out by CLK1, a 4-bit counter LS161 (D11) is also clocked which causes the interrupt on level-4 (INT4) when it reaches its terminal count of 16. Now, if a *zero* is shifted out from C8 the flip-flop C11*a* is toggled at the rate determined by CLK2, but if a *one* is shifted out from C8 the flip-flop C11*a* is toggled at the CLK1 rate. Thus, the word stored on the shift registers is output in SMPTE form.

$\overline{INT4}$ is cleared by writing to the shift registers (C7,C8), i.e. by signal $\overline{SMPTERD}$ at address 60060h.

SMPTE Reading Circuitry

(Refer schematic CMI-28-7 & Timing Diagram)

SMPTE code coming from tape, being converted to TTL signal levels by the CMI-333 board, are received by the CMI-28 through pin 17 of the 26-way connector. The edge-detecting circuitry consisting of the EXOR gates (C1) and the resistor-capacitor combination create a pulse (at pin 6 of C1) for every up or down transition of the incoming signal.

The required output from this SMPTE data separator is to have one interrupt occur for every SMPTE *one* read and another interrupt for every SMPTE *zero* read. This process can be followed through with the timing diagrams. The 68B40 timer is set, according to the frame rate of the SMPTE being read, to 3/4 of the time for one bit cell. The circuit then detects whether there has been a transition in that time or not. If there has been a transistion then an interrupt on level 6 occurs and a *one* is read, if no transition occurred then an interrupt on level 5 occurs and a *zero* is read.

$\overline{INT6}$ and $\overline{INT5}$ are cleared by the $\overline{SMPTERD}$ signal through accessing address 60070h. This will reset flip-flops in D1 making the outputs of the NAND's E1 high.

Pin Connections for the 26-way Connector

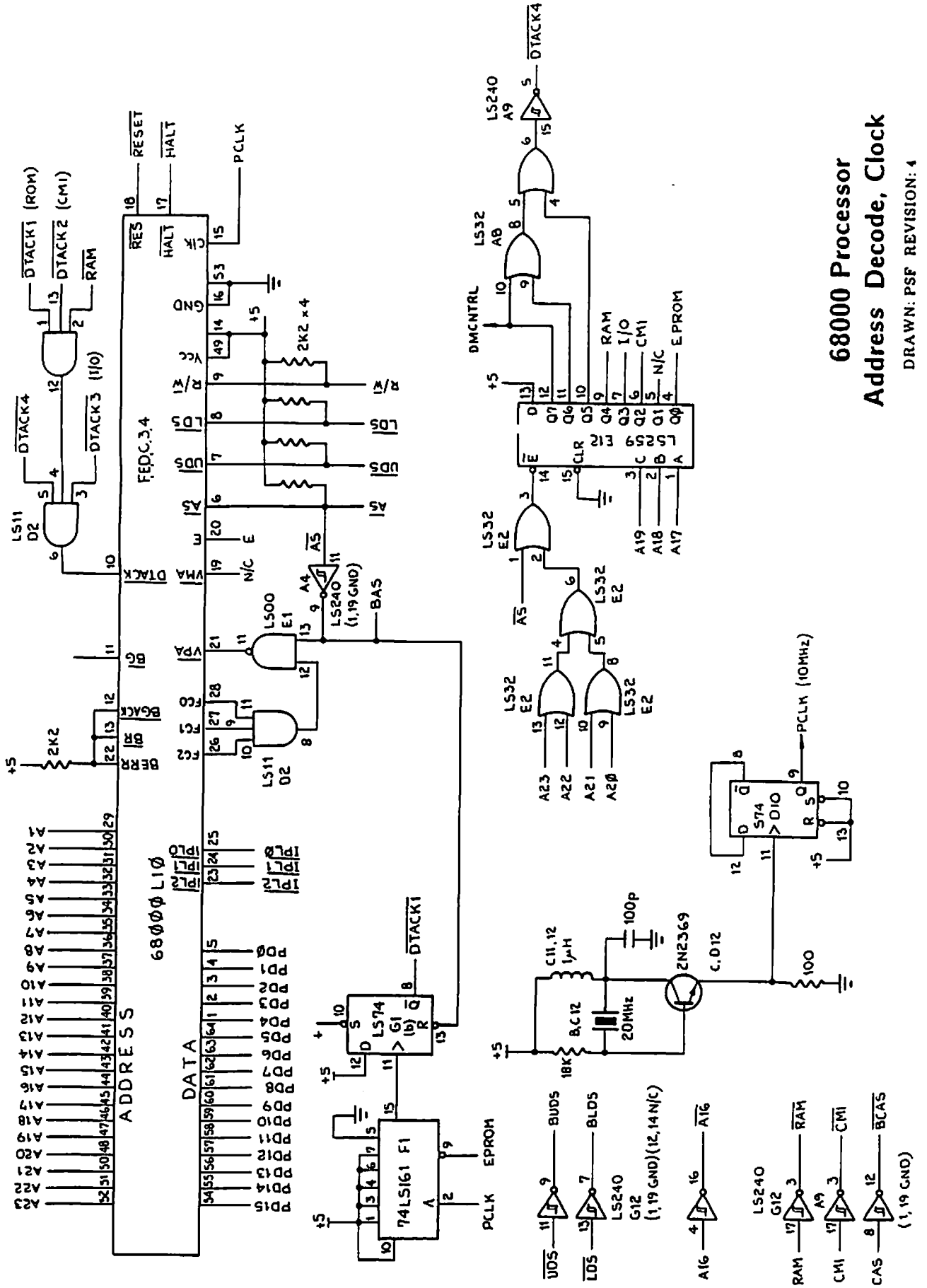
(between the CMI-28 and CMI-332 and CMI-333)

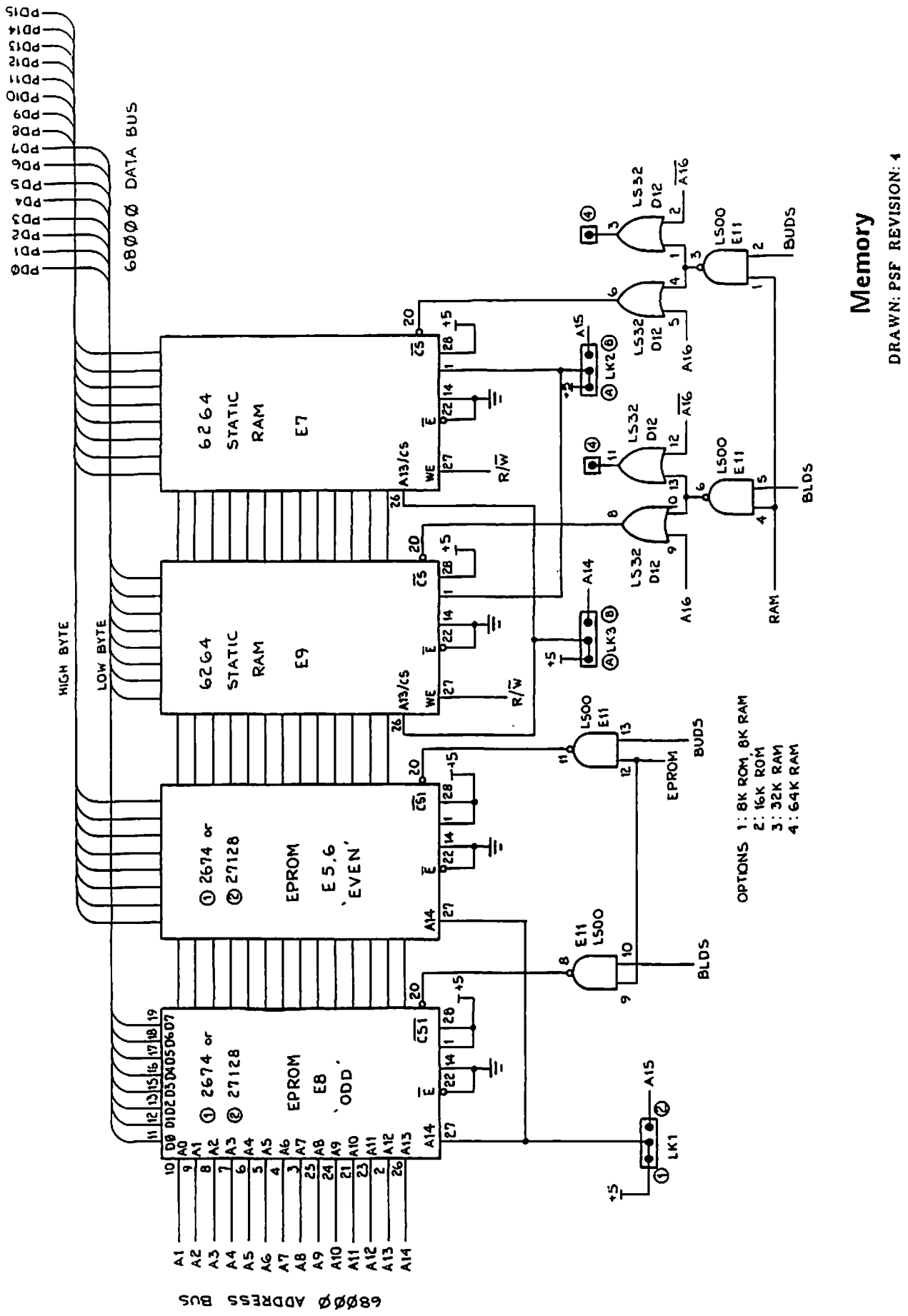
Pin 1	MIDI out A.	
Pin 2	+5 volts.	
Pin 3	MIDI in A.	
Pin 4	SYNC out 1.	
Pin 5	MIDI out B.	
Pin 6	SYNC out 2.	
Pin 7	MIDI in B.	
Pin 8	SYNC out 3.	
Pin 9	MIDI out C.	
Pin 10	Digital Ground.	
Pin 11	MIDI in C.	
Pin 12	Digital Ground.	
Pin 13	MIDI out D.	
Pin 14	RESET/START.	
Pin 15	MIDI in D.	
Pin 16	RUN/STOP.	
Pin 17	SMPTE code in.	
Pin 18	Digital Ground.	
Pin 19	SMPTE code out.	
Pin 20	CLICK out; SYNC out 4.	
Pin 21	CLICK in.	
Pin 22	(CMI332-3) Analog Ground.#	(CMI28) n/c.*
Pin 23	(CMI332-3) +15 volts.#	(CMI28) CPU Halt switch.*
Pin 24	(CMI332-3) -15 volts.#	(CMI28) Digital Ground.*
Pin 25	(CMI332-3) n/c.#	(CMI28) CPU Reset switch.*
Pin 26	(CMI332-3) n/c.#	(CMI28) Digital Ground.

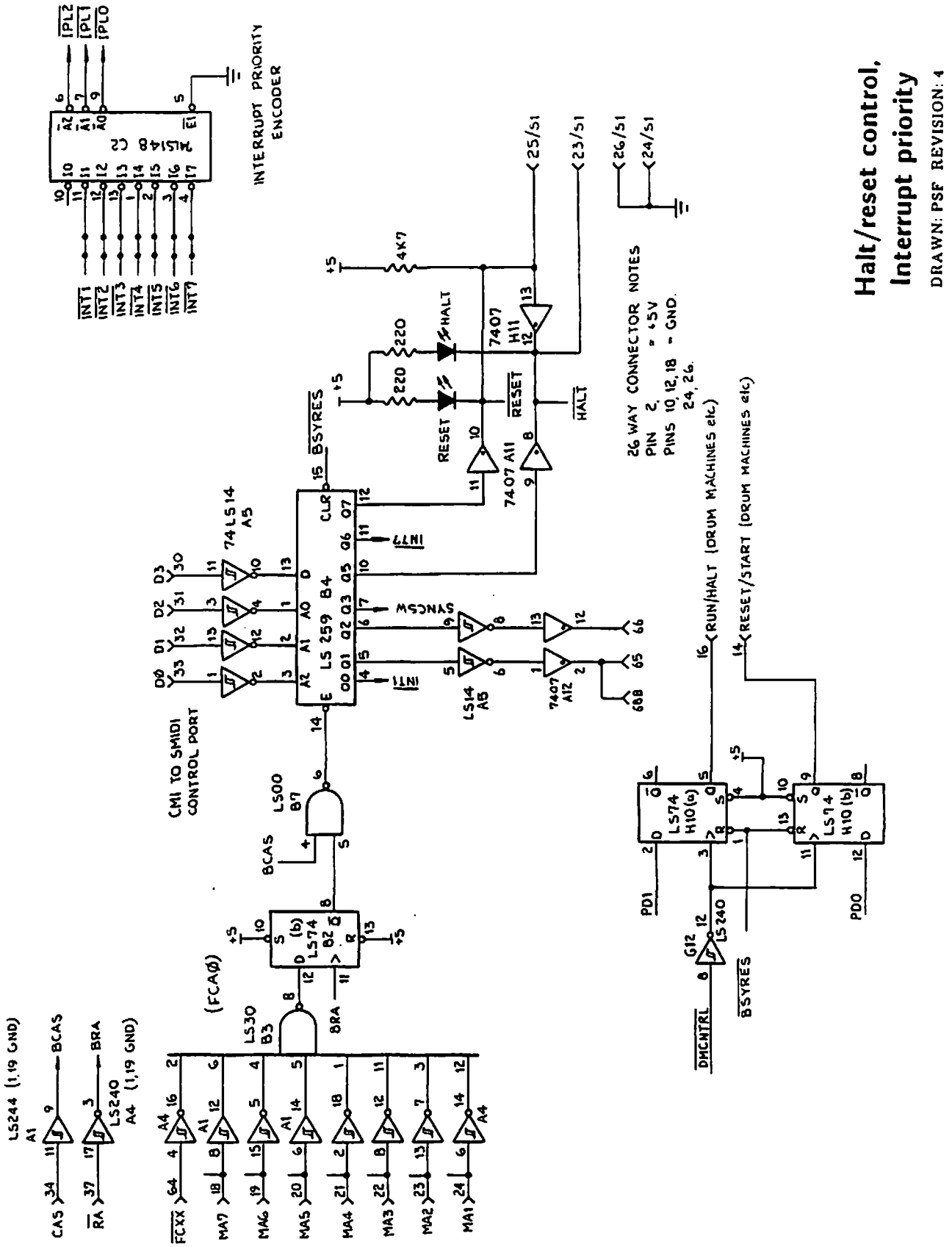
Notes:

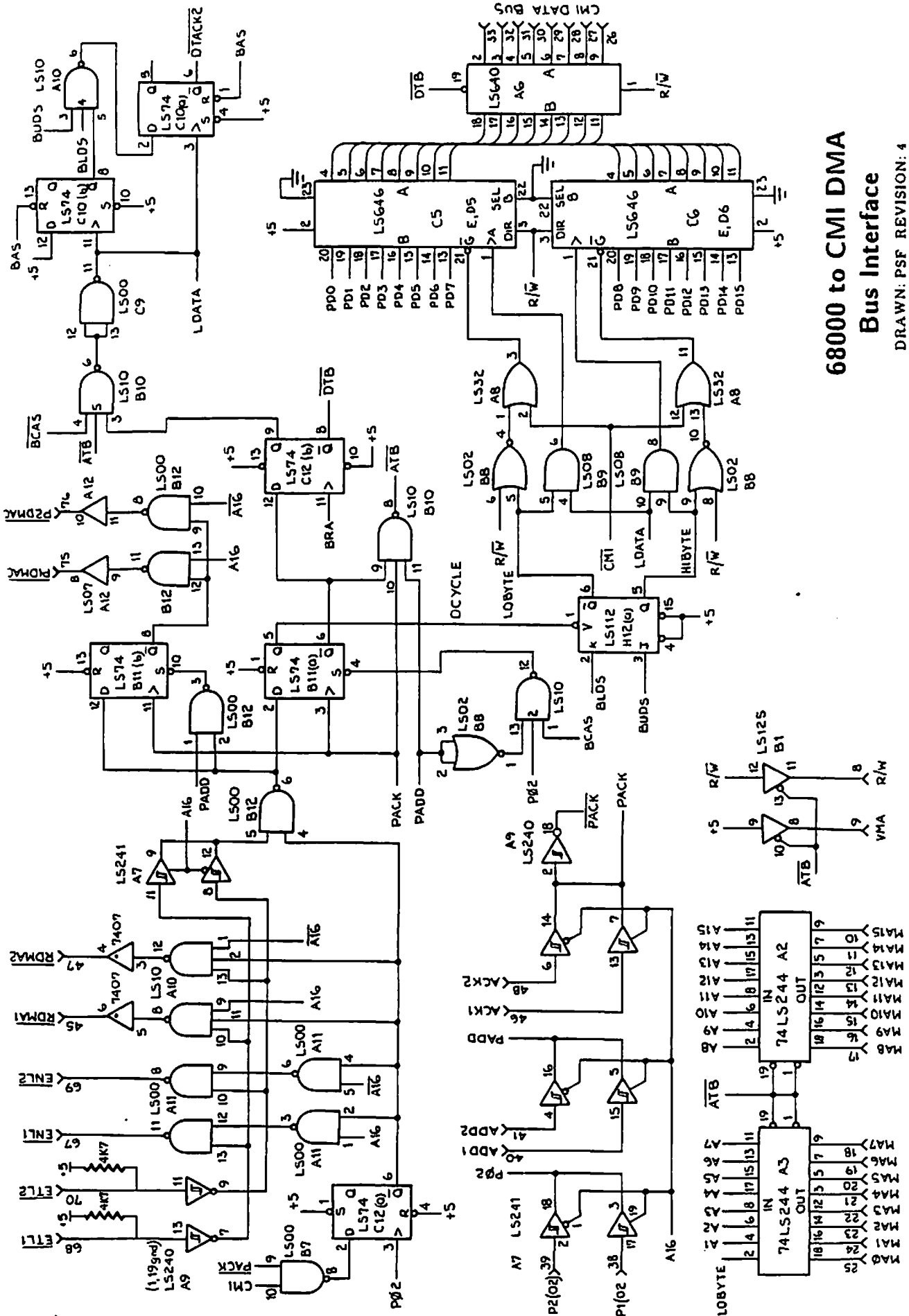
 # - these connections are on Audio Rack only.

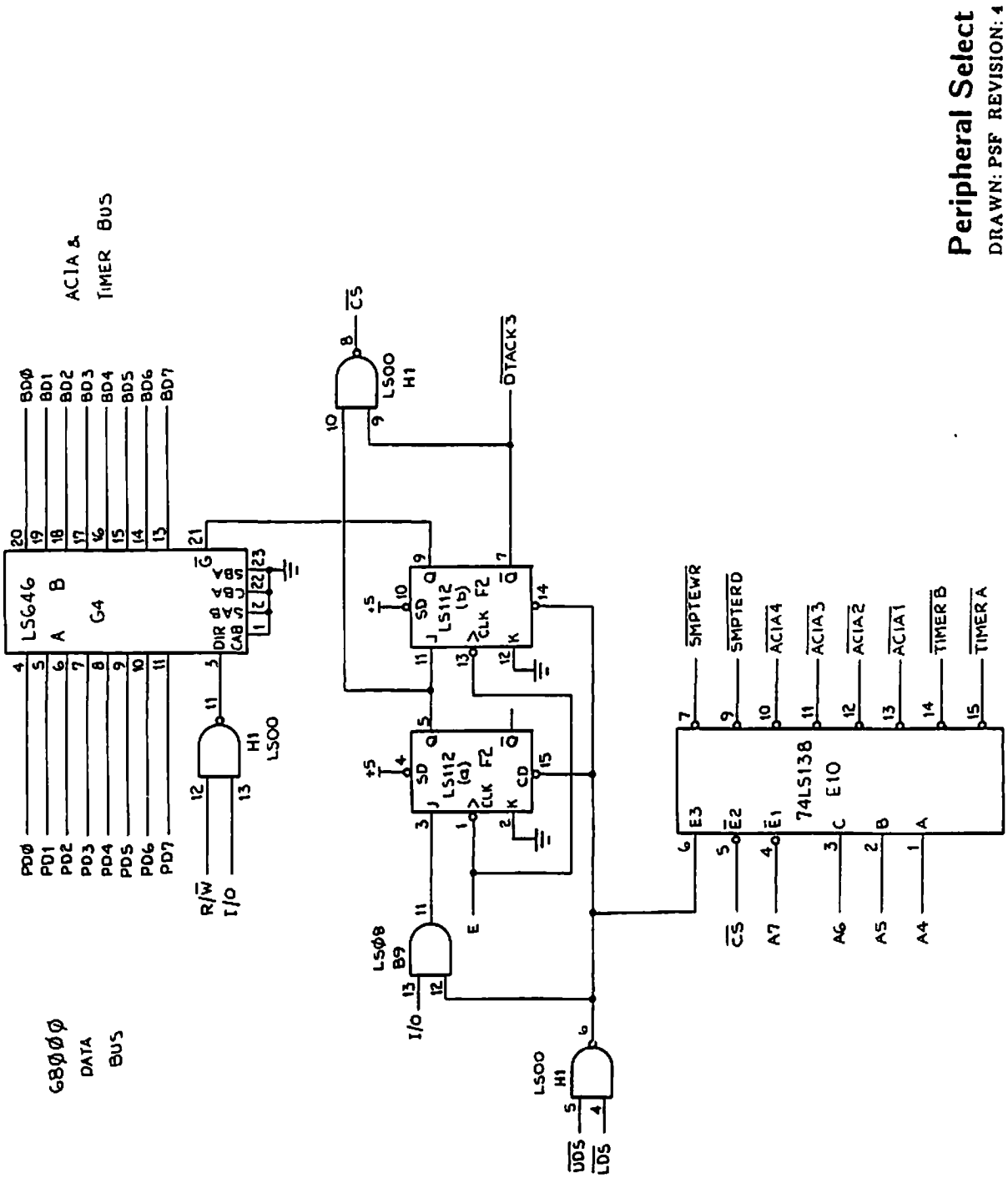
*** - these connections (from the CMI28 board only) are for debugging purposes only. If two push-button switches are connected between pins 23 & 24 and pins 25 & 26, they can be used to manually halt and reset the 68K processor, respectively.**

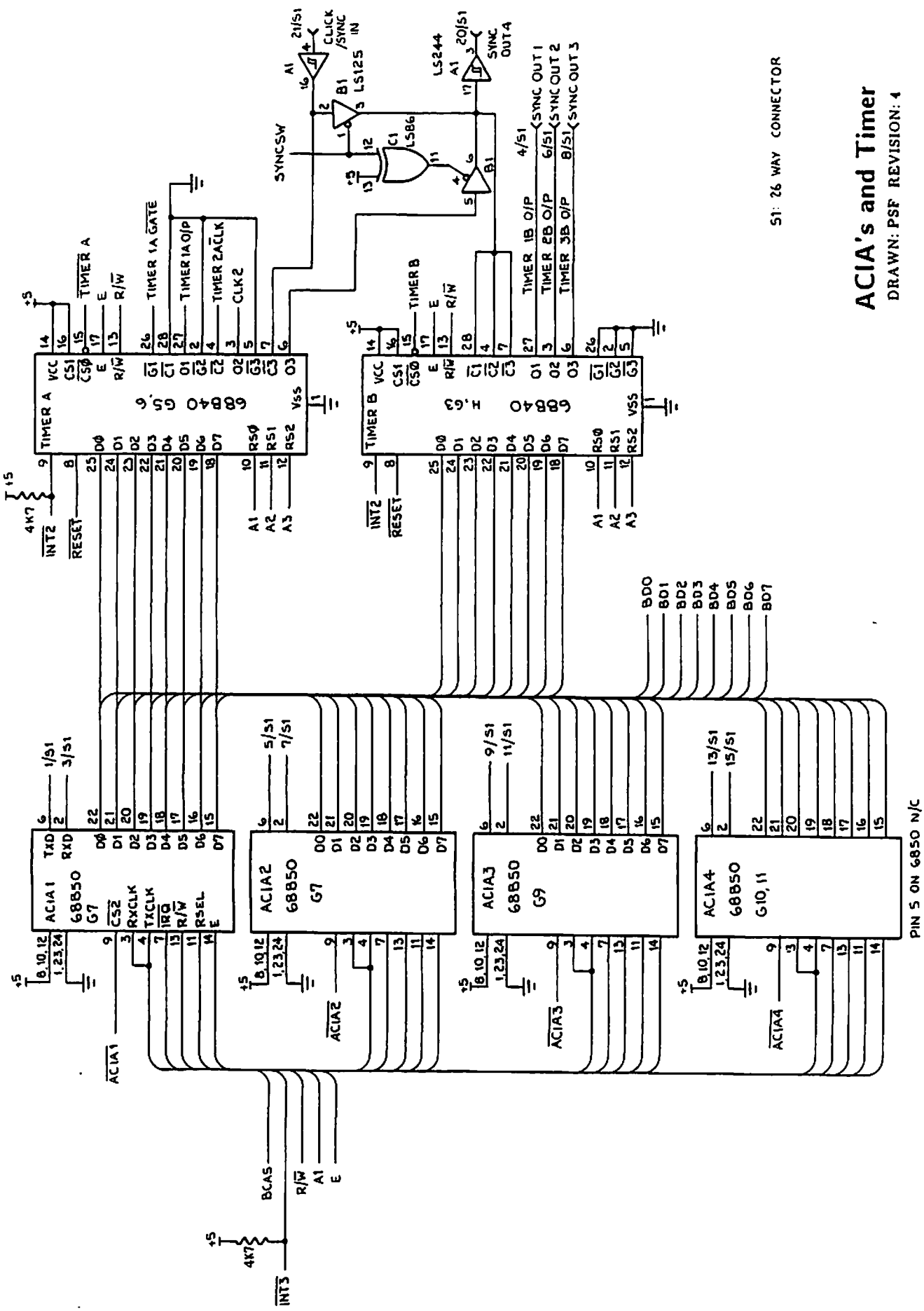

68000 Processor
Address Decode, Clock
DRAWN: PSF REVISION: 4
DTACK1 (ROM)
DTACK2 (CMI)
RAM
DTACK4
DTACK3 (I/O)
LS11 D2
RESET
HALT
PCLK
RES
CLK
GND
Vcc
R/W
LDS
UDS
AS
E
VMA
N/C
DTACK
BG
BGACK
BR
BERR
FC0
FC1
FC2
IPL0
IPL1
IPL2
2K2
2K2 x4
+5
FED.C.34
68000 LIØ
ADDRESS
DATA
A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19 A20 A21 A22 A23
PD0 PD1 PD2 PD3 PD4 PD5 PD6 PD7 PD8 PD9 PD10 PD11 PD12 PD13 PD14 PD15
LS00 E1
LS11 D2
LS240 A4 (1,19 GND)
BAS
LS74 G1 (b)
DTACK1
74LS161 F1
PCLK EPROM
LS32 A8
LS240 A9
DMCNTRL
LS32 E2
LS259 E12
RAM
I/O
CMI
N/C
EPROM
A19 A18 A17
A23 A22 A21 AØ
LS74 D10
PCLK (10MHz)
C11,12 1μH
100p
ZN2369
C.D12
B.C12
20MHz
18K
100
UDS
LDS
LS240 G12 (1,19 GND)(12,14 N/C)
A16
RAM
CMI
CAS
BCAS
(1,19 GND)

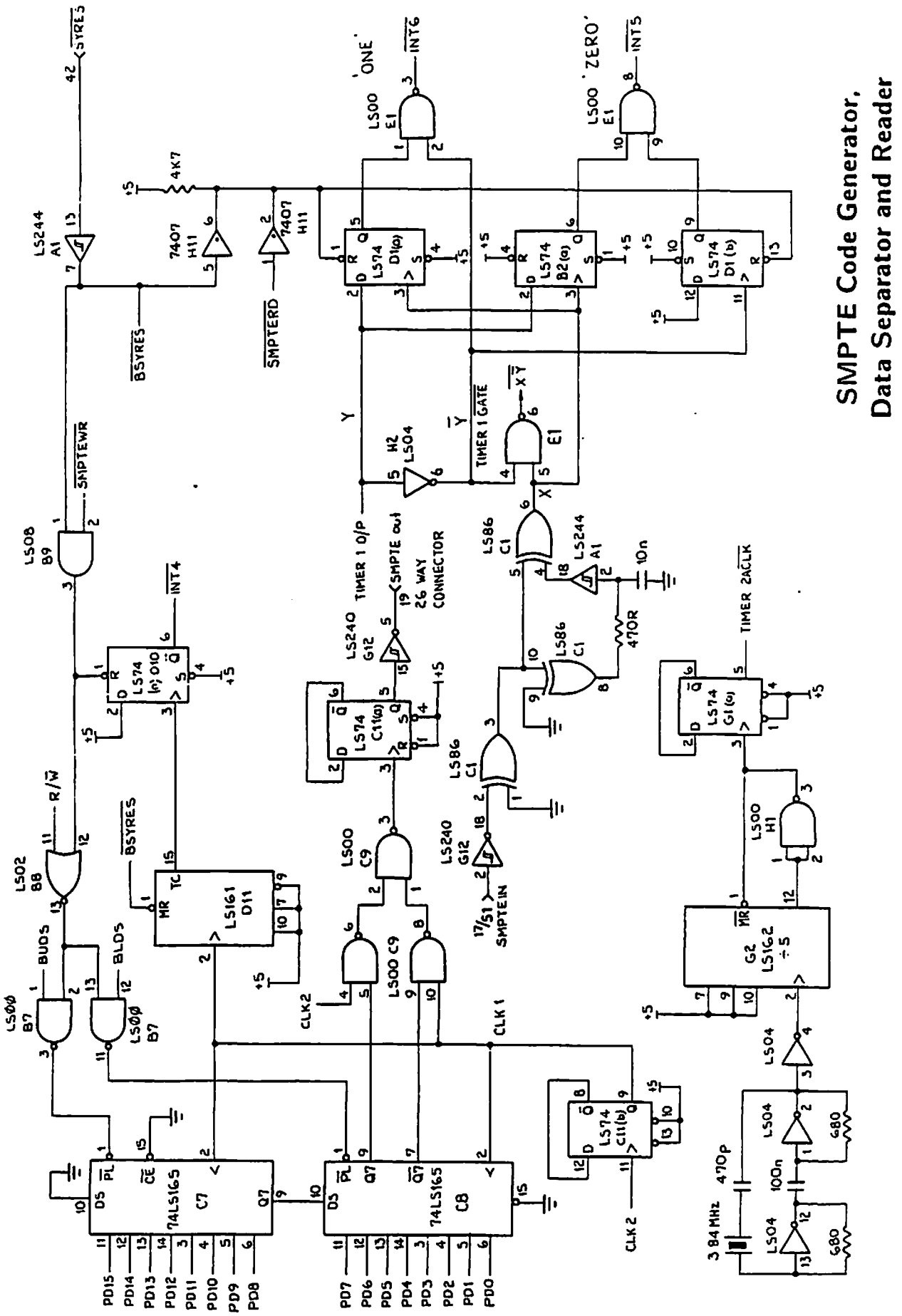

Memory
DRAWN: PSF REVISION: 4
68000 DATA BUS
68000 ADDRESS BUS
HIGH BYTE
LOW BYTE
6264 STATIC RAM E7
6264 STATIC RAM E9
EPROM E5,6 'EVEN'
EPROM E8 'ODD'
① 2674 or ② 27128
R/W
A13/CS
WE
LS32 D12
LS00 E11
BUDS
BLDS
RAM
EPROM
LK1
LK2
LK3
OPTIONS 1: 8K ROM, 8K RAM
2: 16K ROM
3: 32K RAM
4: 64K RAM


Halt/reset control, Interrupt priority
DRAWN: PSF REVISION: 4
INTERRUPT PRIORITY ENCODER
26 WAY CONNECTOR NOTES
PIN 2, = +5V
PINS 10,12,18 - GND.
24,26.
CMI TO SMIDI CONTROL PORT
RUN/HALT (DRUM MACHINES etc)
RESET/START (DRUM MACHINES etc)
LS244 (1,19 GND) A1
LS240 A4 (1,19 GND)
74LS14 A5
LS259 B4
LS00 B7
LS30 B3
LS74 H10(a)
LS74 H10(b)
7407 H11
7407 A11
7407 A12
LS14 A5
LS240 G12
4K7
220
HALT
RESET
BSYRES
BCAS
BRA
(FCAØ)
SYNCSW
CAS
RA
FCXX
MA7
MA6
MA5
MA4
MA3
MA2
MA1
PD1
PD0
DMCNTRL


68000 to CMI DMA
Bus Interface
DRAWN: PSF REVISION: 4
CMI DATA BUS
LS640 A6
LS646 C5 E,D5
LS646 C6 E,D6
LS74 C10(a)
LS74 C10(b)
LS74 C12(a)
LS74 C12(b)
LS74 B11(a)
LS74 B11(b)
LS112 H12(a)
LS241 A7
LS240 A9
LS244 A2
LS244 A3
LS125 B1
LS32 A8
LS02 B8
LS08 B9
LS10 B10
LS00 B12
LS00 A11
LS00 B7
LS07 A12
7407
4K7
DTACK2
BAS
LDATA
BUDS
BLDS
BCAS
ATB
DTB
R/W
CMI
LOBYTE
HIBYTE
OCYCLE
PADD
PACK
P02
ACK1
ACK2
ADD1
ADD2
P2(02)
P1(02)
RDMA1
RDMA2
ENL1
ENL2
ETL1
ETL2
PRDMAC
RDMAC
BRA
VMA

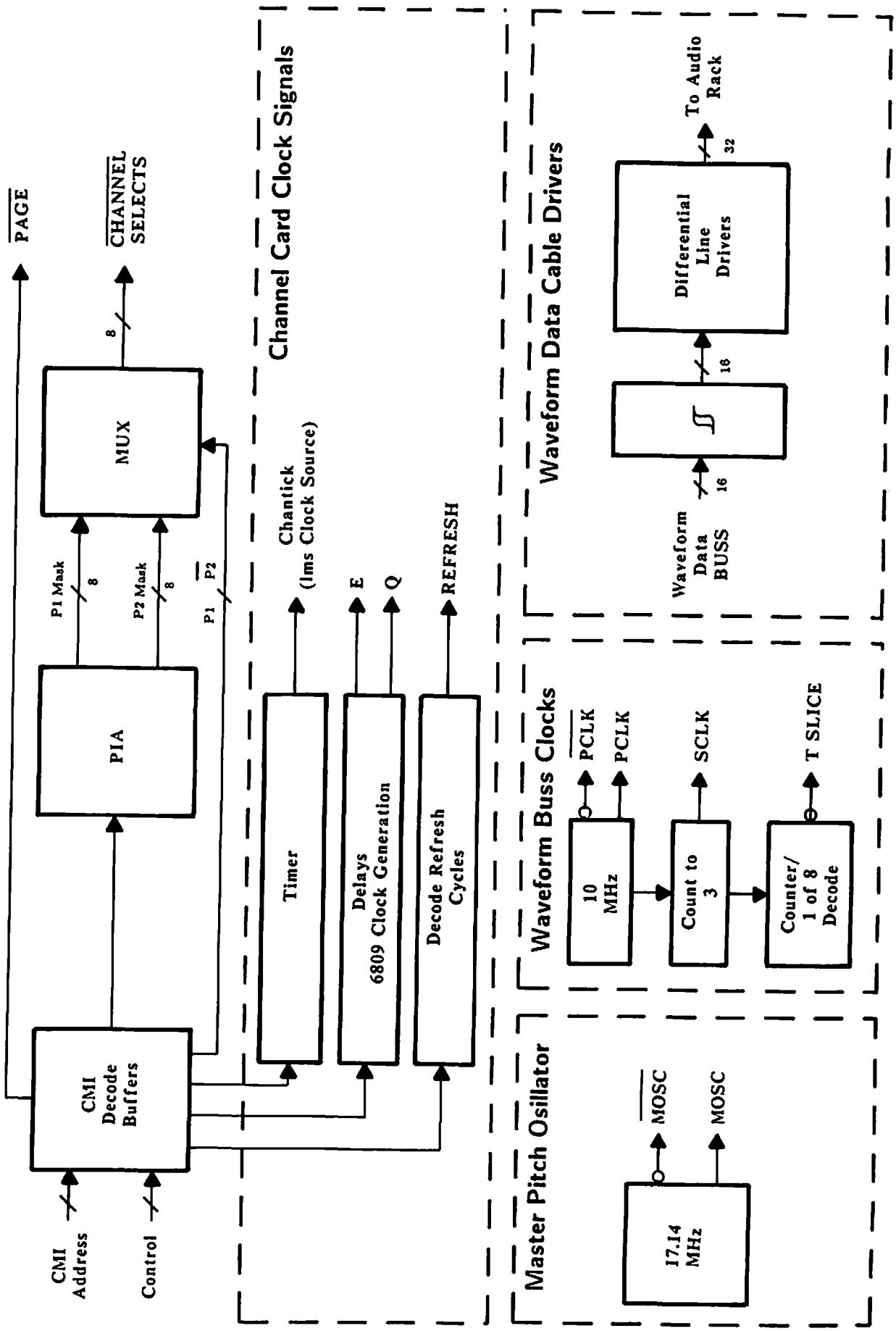

Peripheral Select
DRAWN: PSF REVISION: 4
ACIA & TIMER BUS
68000 DATA BUS
LS646
G4
LS112 (a)
LS112 (b)
74LS138
E10
LS00
LS08
DTACK3
SMPTEWR
SMPTERD
TIMER B
TIMER A
R/W
I/O
E
CS
UDS
LDS


ACIA's and Timer
DRAWN: PSF REVISION: 4
51: 26 WAY CONNECTOR
PIN 5 ON 6850 N/C
SYNCSW
CLICK /SYNC IN
SYNC OUT4
SYNC OUT 1
SYNC OUT 2
SYNC OUT 3
TIMER A
TIMER B
6840 G5,6
6840 H,G3
ACIA1 68850 G7
ACIA2 68850 G7
ACIA3 68850 G9
ACIA4 68850 G10, 11
4K7
INT2
RESET
INT3
BCAS
R/W
A1
E


SMPTE Code Generator,
Data Separator and Reader
DRAWN: PSF REVISION: 4
'ONE'
'ZERO'
TIMER 1 O/P
SMPTE out
26 WAY CONNECTOR
TIMER 1 GATE
TIMER 2 CLK
SMPTEIN
SMPTEWR
BSYRES
SMPTERD
R/W
BUDS
BLDS
CLK 1
CLK 2
3.84 MHz
74LS165 C7
74LS165 C8
LS161 D11
LS162 G2
PD15
PD14
PD13
PD12
PD11
PD10
PD9
PD8
PD7
PD6
PD5
PD4
PD3
PD2
PD1
PD0

CMI-32 2.10

Channel Support Card


Channel Card Clock Signals
Waveform Data Cable Drivers
Waveform Buss Clocks
Master Pitch Osillator
PAGE
CHANNEL SELECTS
8
MUX
P1 Mask
8
P2 Mask
8
P1
P2
PIA
CMI Address
Control
CMI Decode Buffers
Chantick (1ms Clock Source)
Timer
E
Q
Delays 6809 Clock Generation
REFRESH
Decode Refresh Cycles
Waveform Data BUSS
16
Differential Line Drivers
16
32
To Audio Rack
10 MHz
PCLK
PCLK
Count to 3
SCLK
Counter/ 1 of 8 Decode
T SLICE
17.14 MHz
MOSC
MOSC

Introduction

The Channel Support Card (CSC) provides the main timing signals for the control of the Waveform buss, Waveform Processor and Channel cards. It also contains the data drivers to connect the Waveform data buss to the audio board DACs. These functions are independent of each other.

Channel Card Access and Control

The Channel Cards (CCs) are accessed from the dual CPU buss via the CSC. The CSC decodes the dual CPU busses upper address lines to form CC select lines. The CCs may be accessed either individually to set their control registers or selectively simultaniously to their memory. This is done so that common data may be written to more than one channel cards memory at once.

The channel mask, that allows simultanious access, contains a bit for each channel card. Channels with their mask bit set, will be accessed when a device on the dual buss accesses the locations for the channel cards. As there are 2 distinct and independent buss states on the dual buss, there are two mask registers, called MASK1 and MASK2.

The CSC also generates the 2Mhz E and Q signals that the 6809E requires. These signals are generated externally so that the CCs on board proccessor is syncronized with the dual CPU buss. This is required to make all external accesses to the channel card transparent to the on board 68B09Es.

Waveform Buss Control

The timing of the Waveform Buss (WBUSS) is dependent on just one signal, SCLK, Slice Clock. This divides the WBUSS into 300nsec slices to be shared between the WP, CCs and refresh. It is used by the WRAM, WP and CCs.

Waveform Buss access by Channel Cards

The Channel cards are allowed access to the WBUSS in a "round robin" fashion. The WBUSS is divided into 8 300nsec time slices. The signal TSLICE defines the first slice in the sequence and is input into CC number 1, the most right hand CC. Each CC forms a bit in a shift register that passes this TSLICE signal to the next CC. Thus allowing them on SCLK cycle every 8 to access the WBUSS.

Address decoding and Masks

(refer schematic CMI-32-00)

Address lines MA8 to MA15 are buffered and gated with VMA and Peripheral enable, ENBL, to generate the card select at B2 pin 9. Decoder B3 and OR gates in A1 decode for the timer, Pia, CC register selects and CC memory page select. The Page select signal is qualifierd by BRA at B10 to drive all the CCs.

The multiplexers at C7 and B7 select the Mask register appropriate to the current buss cycle. This is achieved by routing the adress phase signal, ADD1, to the select input of C7 and B7, pin 1.

The multiplexer outputs are ANDed with the outputs of decoder A7, after which they are qualified by latch A10, and bussed to the Channel cards. A7 decodes address lines MA4 to MA6 to generate the 8 selects for the register mode access on the 16 byte boundaries required.

The enable for C7 and B7 can be generated under two conditions. When the Channel registers are accessed via the mask, and when the channels memory is accessed, always by the mask. Hence the AND gate driving the enable, pin 15, of C7 and B7.

Timing Generation

(refer schematic CMI-32-01 and timing diagram)

The 2Mhz E signals is generated by ORing buss signals RA and RAS. The Q signal is E delayed by 125ns by a delay line.

Because of the delays involved in multiplexing internal buss signals on the CCs, the CCs require a $\overline{RAS}$ signal delayed by 25ns with respect to the equivalent RA buss signal. This is also generated by a delay line, and bussed to all CCs.

Refresh for the CCs and the WBUSS is synchronized to system refresh by detecting when the first device on the P1 DMA daisy chain requests the CMI buss. This is done by gating REFEN, ADD1 and ACK1 with F/F C1 and NAND gate C2. This generates an active high pulse 500ns long every 16 microseconds, during the P1 refresh cycle.

The master pitch oscillator for all the channel card pitch generation, is based around the 34MHz crystal and transistor Q2. This is divided by 2 and driven to all channel cards via the differential line drivers in D6.

The WBUSS timing chain starts with the 20MHz crystal and transistor Q1. This is divided by two and bussed to the WBUSS via differential line drives.

SCLK is generated in D1 which is a counter that counts to 3. It is bussed to the WBUSS via a buffer in A11 and series resistor R42. R42 cuts down the ringing and undershoot, on the bussed line.

SCLK is then fed to counter E1 to generate the inputs to the 1 of 8 decoder E2. This generates the TSLICE signal used in buss arbitration by the CCs. It goes low once every 8 SCLKs.

Timer Mask Registers

(refer schematic CMI-32-03)

The 6840 is used to generate a real time clock interupt to all the channel cards via its counter 1. This results in a 1KHz square wave on pin 27 of the 6840.

The PIA is used for the MASK registers as it allows the current MASK value to be read by the software.

Bidirectional Data buffer buffers the CMI data buss to the 6840 and 6821 only.

The latches B4 and B10 hold the addresses and 6840, 6821 chip select signals across the data phase of the CMI buss.

Data cable drivers

(refer schematic CMI-32-04)

The WBUSS data is buffered in schmitt receivers A8 and A9 and feed to the differential line driver devices, MC3487, and then to the 34 way IDC socket. The most significant bit of the data is inverted by

the connections to E10 pins 13 and 14, to allow the data in WRAM to be 2's compliment and the audio DACs to be offset binary. The data is differential to evade corruption of the 3.3MHz data stream.

Channel Card Addressing

Via the MASK

Channels may be accessed via the channel mask at $E091 to $E093. The mask registers are the two sides of the PIA. A channel is selected when its mask bit is set to 1. Bit 0 is channel 1, bit 7 is channel 8.

$E080	channel cards via mask
$E090	6821 Channel Mask PIA base address
MASK1	A side is P1 mask
MASK2	B side is P2 mask

There is no hardware to stop reading from multiple channels .
DO NOT READ WITH THE MASK WITH MORE THAN ONE CHANNEL SELECTED.

Direct Channel Card Register Access

The channel cards registers may also be accessed individually with the cards having the following base addresses.

$E000 channel card 1
$E010 channel card 2
$E020 channel card 3
$E030 channel card 4
$E040 channel card 5
$E050 channel card 6
$E060 channel card 7
$E070 channel card 8

Channel Card Memory Access

The channel cards 64k memories are accessed in 256 byte PAGEs.

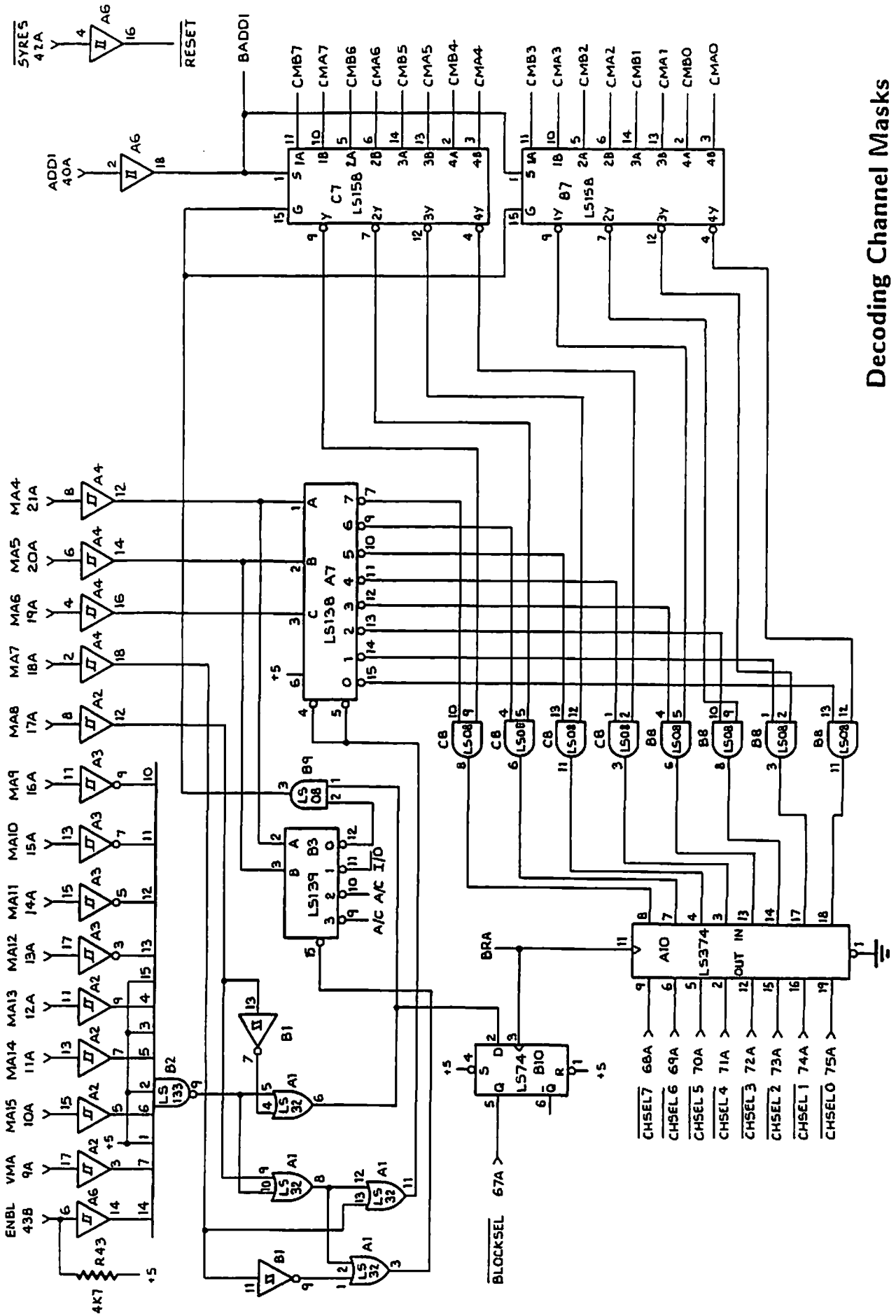
The PAGE to be accessed is written to the register in the channel card, one register for P1, another for P2.

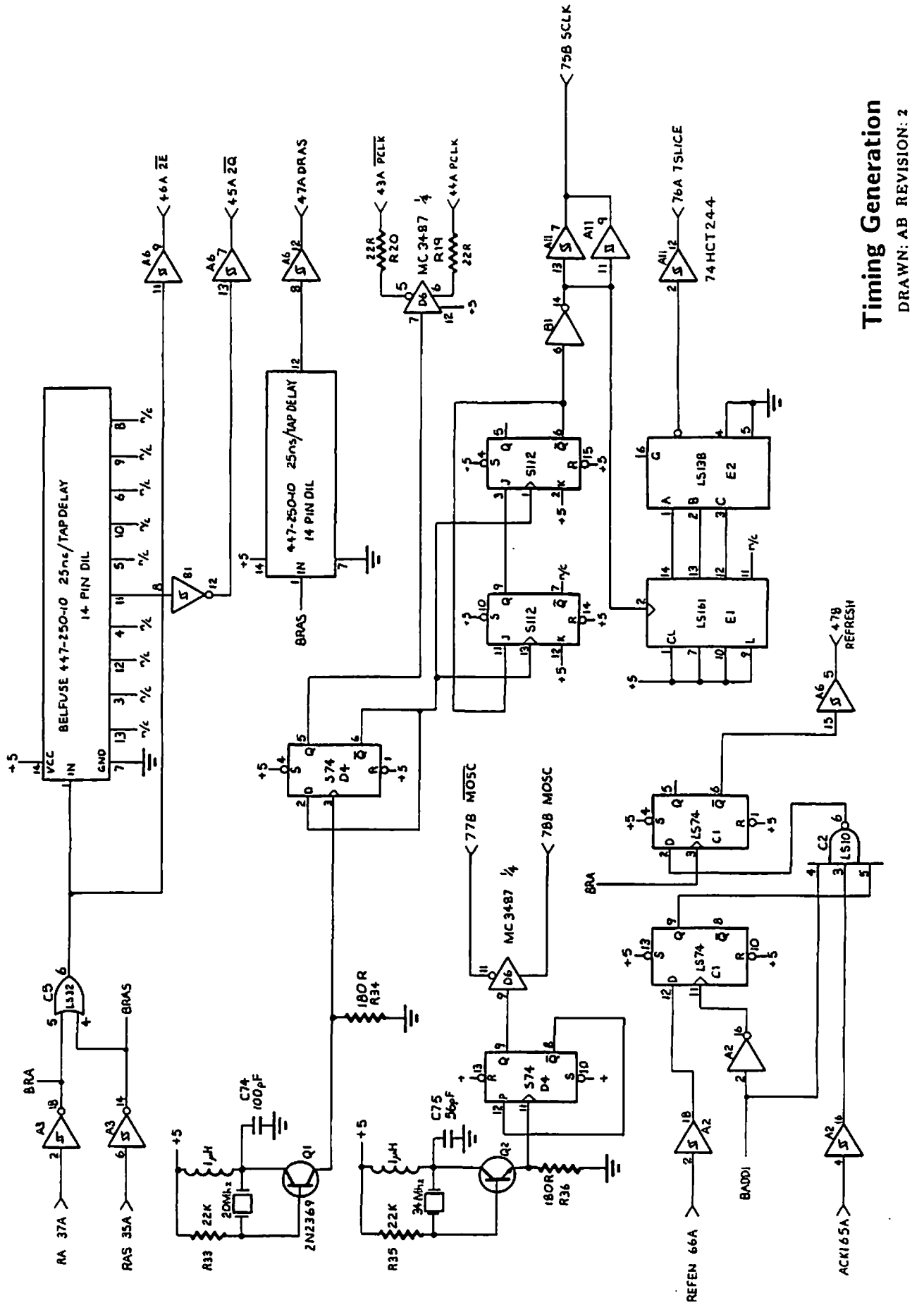
The upper 8 bits of address of the channel cards page registers are used as the upper address bits of the accessed location. The lower 8 bits of the LD or ST instructions address determines the byte within the block.

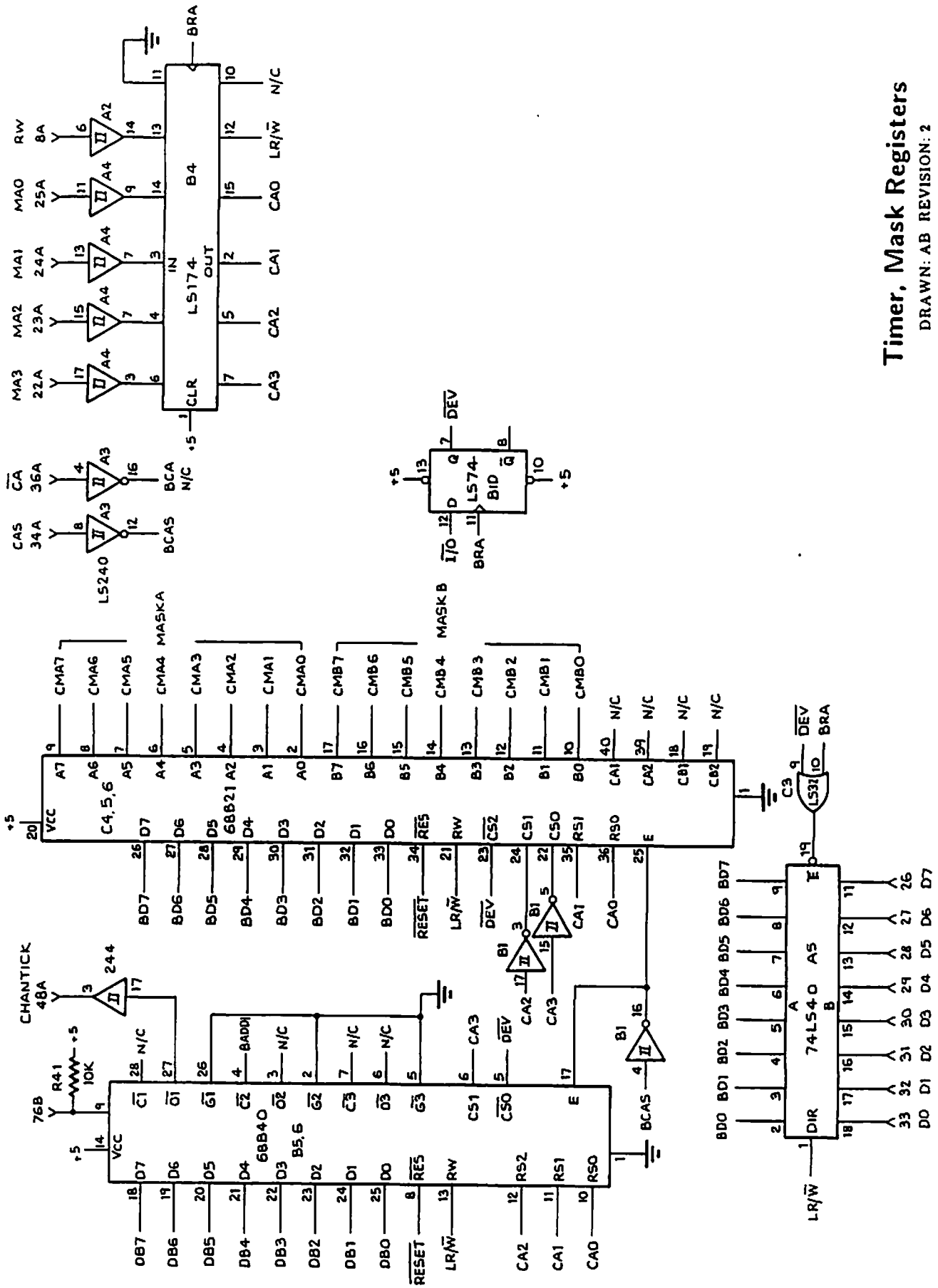
$E100 to $E1FF Page of Channel Memory

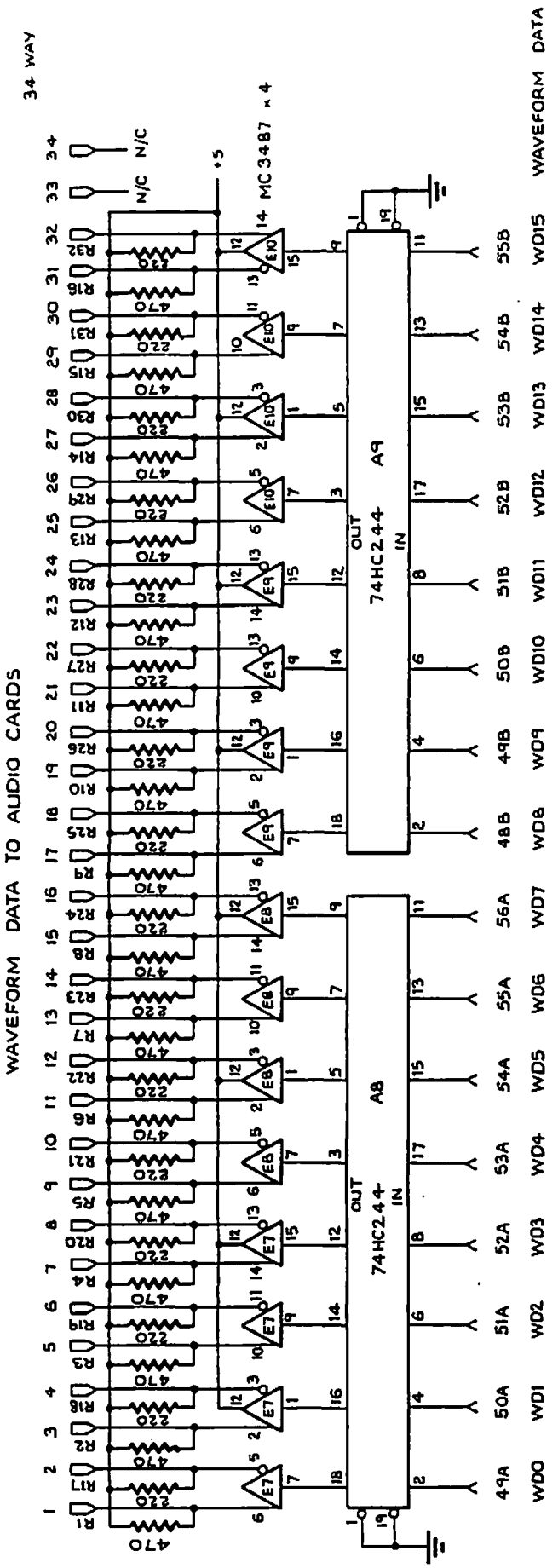
Timer

Timer 1 has its output bussed to all channel cards for its real time clock. Timer 1 is set to run continously at 1KHz. This results in channel card ramps being updated at 1ms intervals. Timer 2 has its input connected to the 1 MHz ADD1 buss signal. Timer 3 has no input or outputs connected. The 6840's IRQ output is connected to an interrupt input on the CPU Control Card.


Decoding Channel Masks
DRAWN: AB REVISION: 2
SYRES 42A
RESET
BADDI
ADDI 40A
CMB7
CMA7
CMB6
CMA6
CMB5
CMA5
CMB4
CMA4
CMB3
CMA3
CMB2
CMA2
CMB1
CMA1
CMB0
CMA0
C7 LS158
B7 LS158
MA4 21A
MA5 20A
MA6 19A
MA7 18A
MA8 17A
MA9 16A
MA10 15A
MA11 14A
MA12 13A
MA13 12A
MA14 11A
MA15 10A
VMA 9A
ENBL 43B
A7 LS138
B3 LS139
A/C A/C I/O
B9 LS08
B1
B2 LS133
LS32
R43 4K7
+5
C8 LS08
B8 LS08
BRA
A10 LS374
OUT IN
B10 LS74
BLOCKSEL 67A
CHSEL7 68A
CHSEL6 69A
CHSEL5 70A
CHSEL4 71A
CHSEL3 72A
CHSEL2 73A
CHSEL1 74A
CHSEL0 75A


Timing Generation
DRAWN: AB REVISION: 2
BELFUSE 447-250-10 25ns/TAP DELAY 14 PIN DIL
447-250-10 25ns/TAP DELAY 14 PIN DIL
MC 3487 1/4
74 HCT 244
LS138
LS161
LS74
LS10
S112
S74
LS32
75B SCLK
76A TSLICE
78 REFRESH
77B MOSC
78B MOSC
BRA
BRAS
RA 37A
RAS 35A
REFEN 66A
BADDI
ACK16 5A
2N2369
180R R34
180R R36
22K R33
22K R35
100pF C74
56pF C75
1µH
20MHz
34MHz


Timer, Mask Registers
DRAWN: AB REVISION: 2
LS174
B4
LS240
LS74
B1D
74LS40
68B21
68B40
C4,5,6
B5,6
MASKA
MASK B
CHANTICK
48A
76B
R41
10K
BRA
BCA
BCAS
DEV
I/O
LR/W
RESET
CA0
CA1
CA2
CA3
CB1
CB2
CMA0
CMA1
CMA2
CMA3
CMA4
CMA5
CMA6
CMA7
CMB0
CMB1
CMB2
CMB3
CMB4
CMB5
CMB6
CMB7
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
+5
N/C



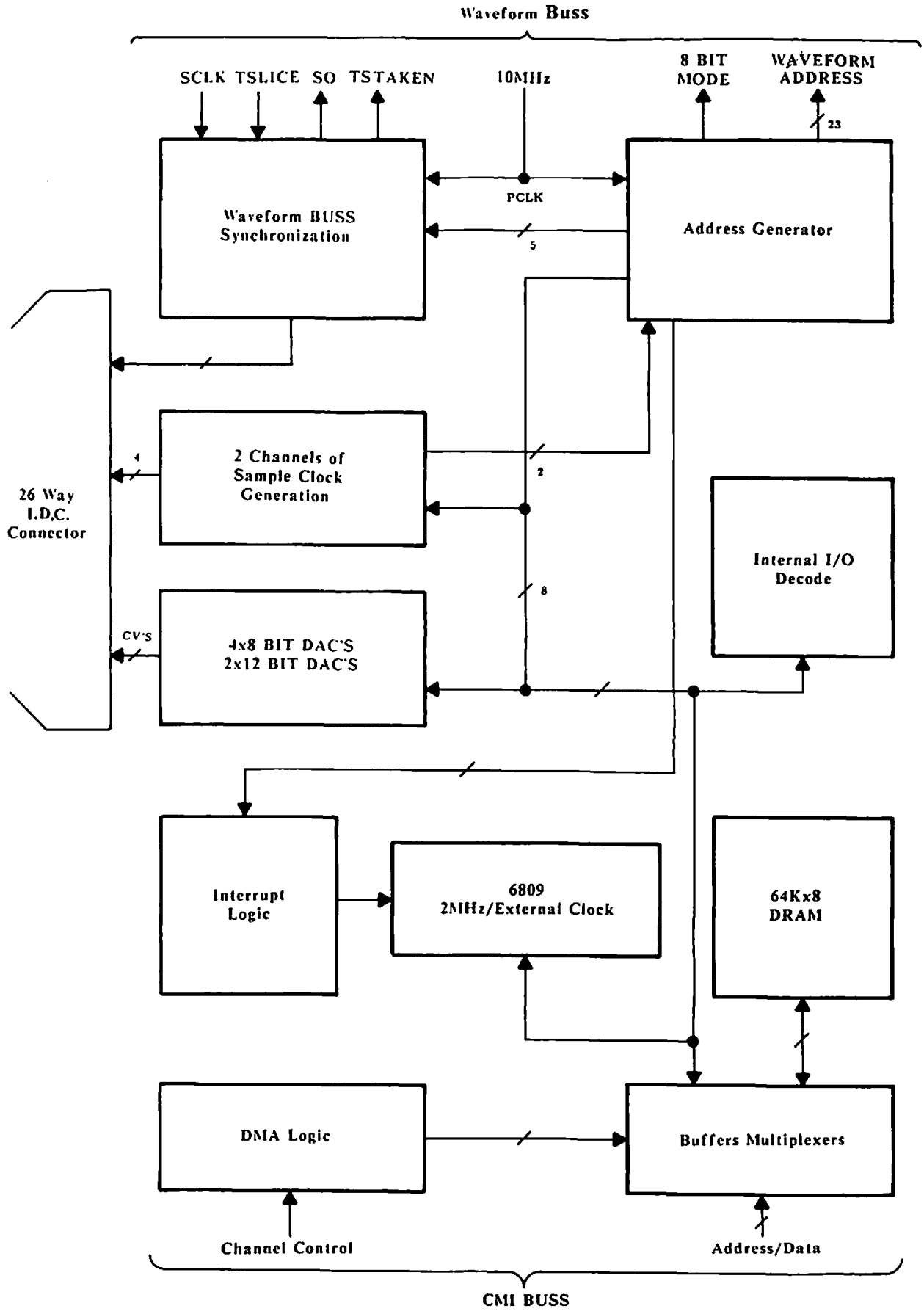
Data Cable Drivers

DRAWN: AB REVISION: 2

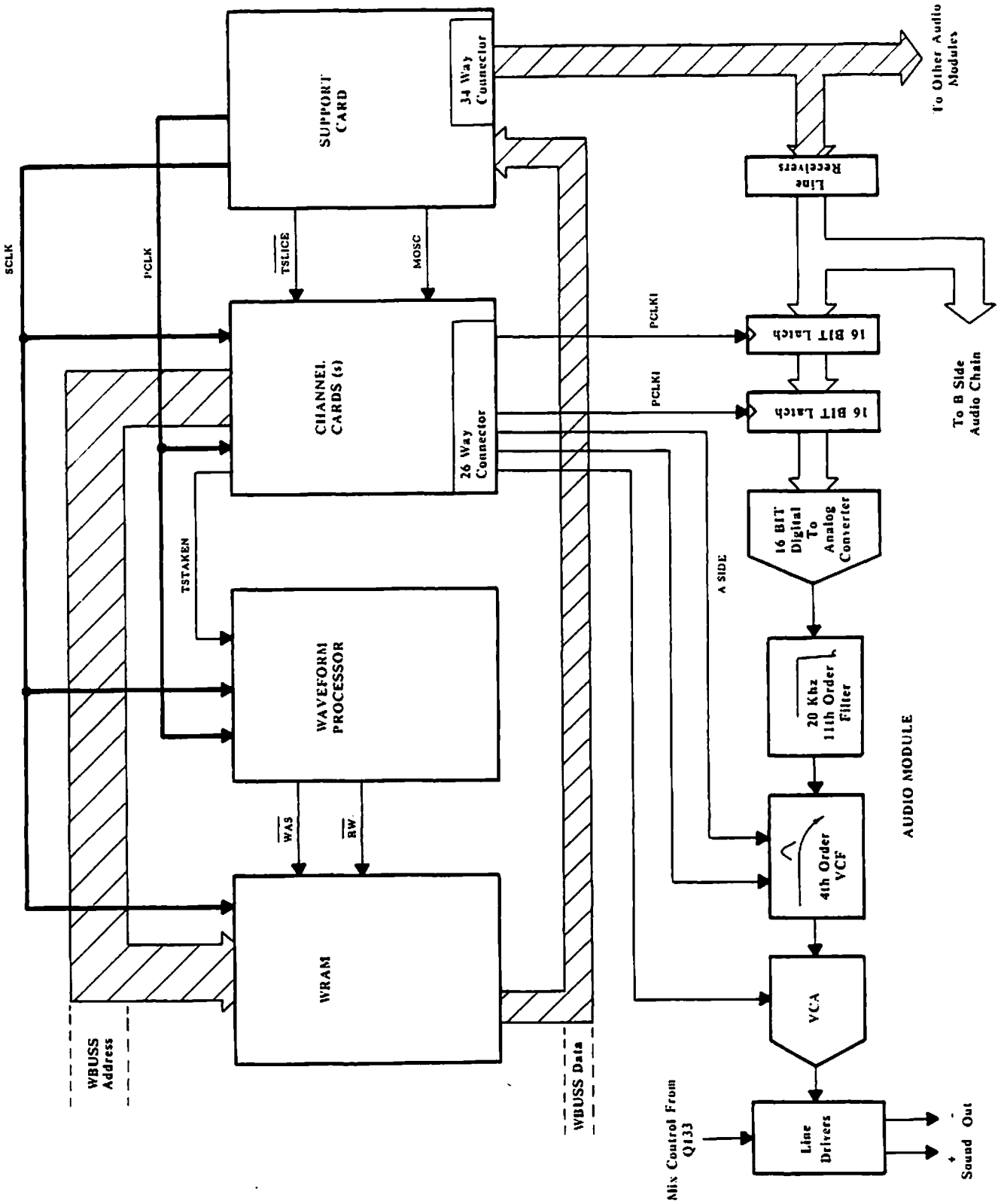
CMI-31 2.11

Channel Card

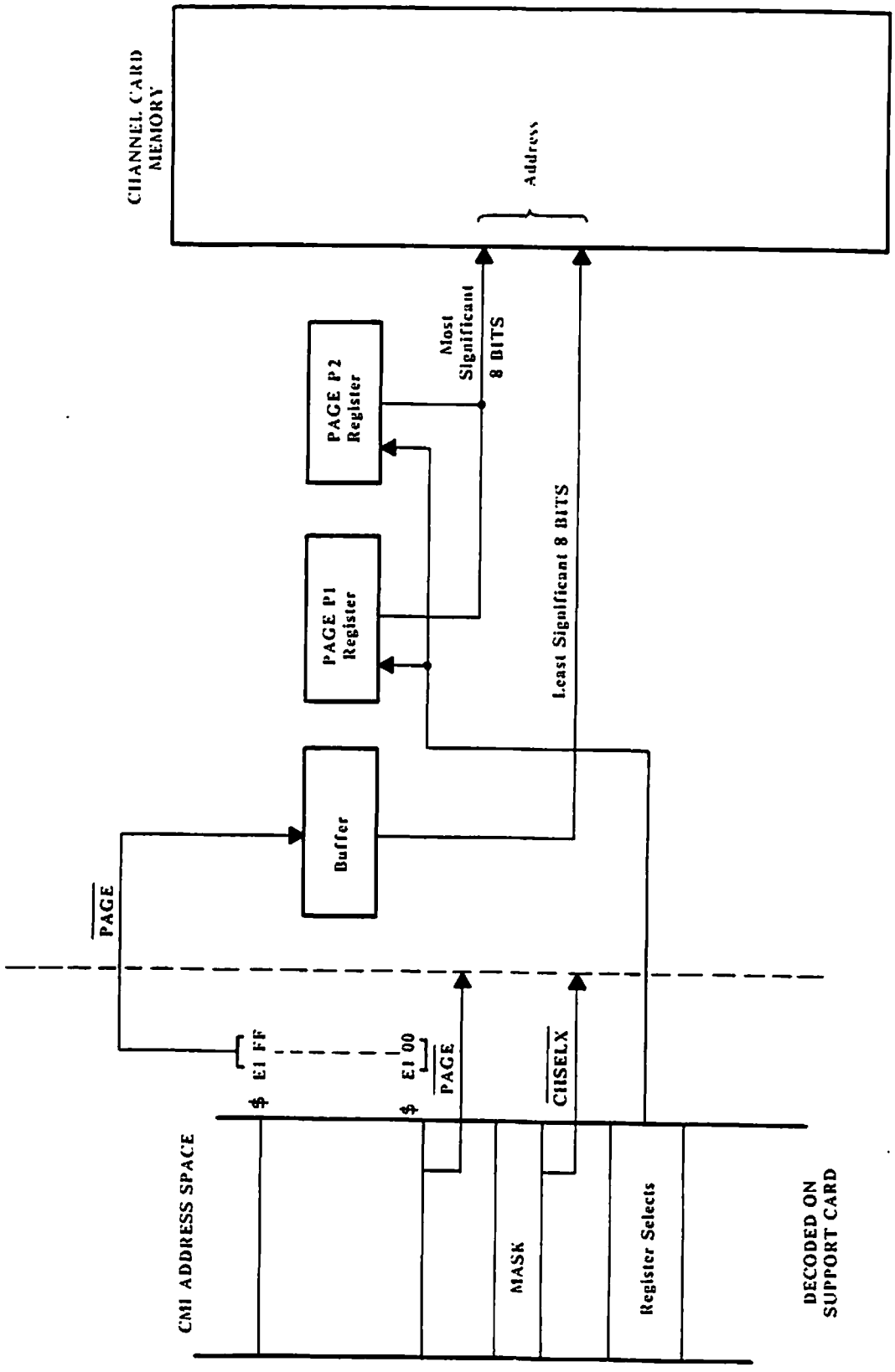
Channel Card Block Diagram



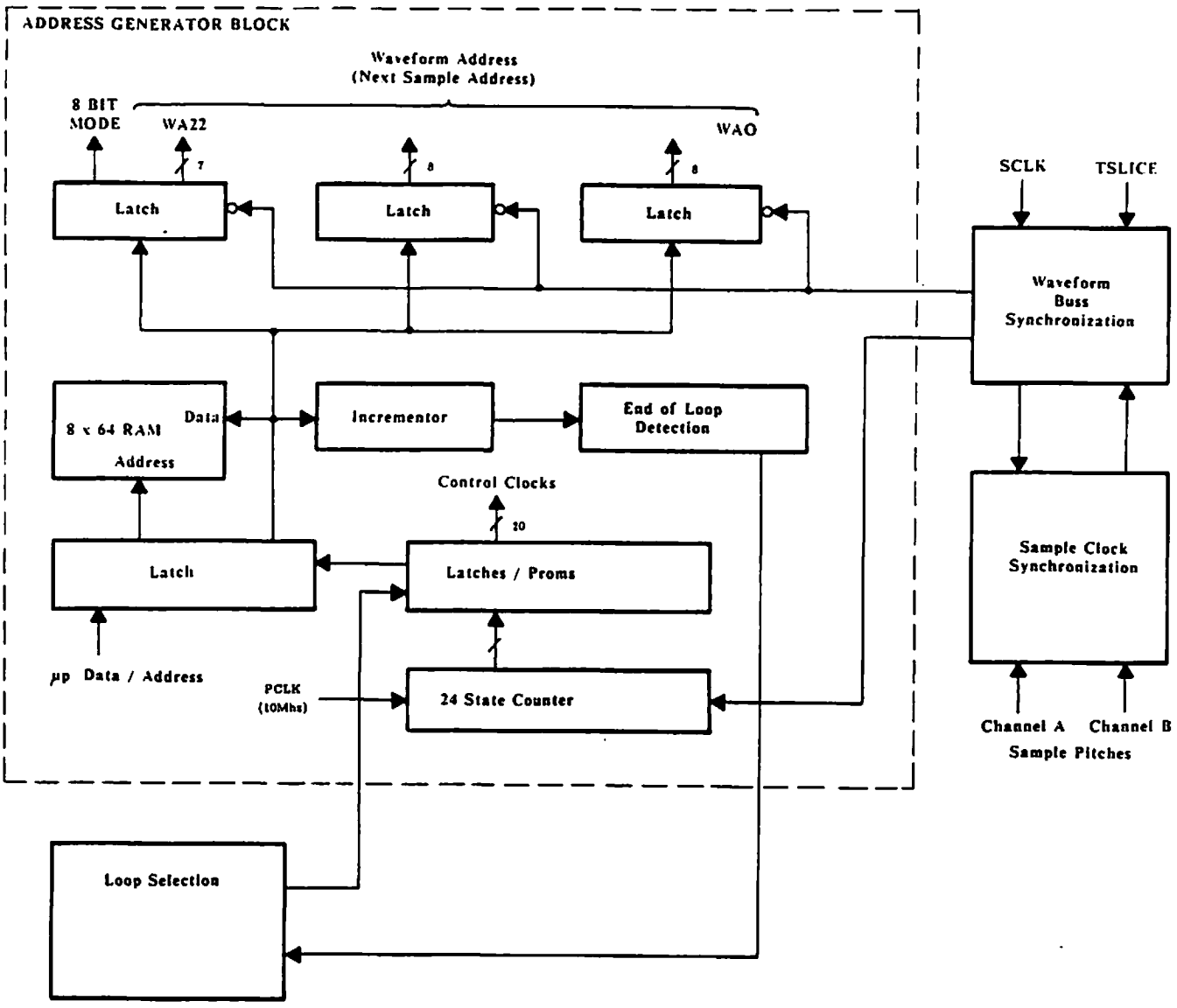
Important Modules during Sound Playing



Memory Access Block Diagram



Address Generator Block Diagram



Terminology
WP: Waveform Processor
GIC: General Interface Card
WBuss: Waveform Buss
WRAM: Waveform RAM (2M - 14Mbytes)
CC: Channel Card
CCP: Channel Card Processor
CSC: Channel Support Card
CPU: Dual 6809 processor system running OS9.
Hexidecimal numbers are in the form nnnnnnH.
Active low signal names are preceded by a "/" character

Introduction

The Channel Card (CC) provides the sample addresses, sample clocks and control voltages used in outputing data to the Audio Modules and controlling the resulting audio waveforms amplitude and high frequency content. It carries on these functions as a self contained computer, once instructed what functions to perform by the WP, GIC or CPU.

The CC can be divided into 2 main sections. The control of the 6809 and its memory by the main CPUs through the CPU buss interface, and the on board 6809s I/O, which generates signals for the WBUSS and the Audio Rack.

Channel Control

The CC contains a 2MHz 68B09E and 64K of dynamic ram. To the main CPUs it looks like a block of 16 registers and two pages of 256 bytes of memory, mapped into its peripheral space.

The generation of the CC select signals occurs on the CMI-32, Channel Support Card (CSC).

The CCs registers allow the main CPUs to reset and interrupt the on board 68B09E and select which two pages of memory are mapped into the main CPUs address space.

All accesses to the CCs registers and memory are made transparently to the Channel Card Processor (CCP) by cycle stealing. This allows the fastest update of control parameters with the minimum overhead.

Refresh also steals one cycle every 16 microseconds from the CCP.

Channel Card Peripherals

The CCP initializes and controls the sample address and clock generating hardware collectively referred to as the Address Generator (ADDGEN). It also addresses 6 DACs used to generate control voltages used in the control of the Voltage Controlled Filters (VCF) and the Voltage Controlled Amplifiers (VCA) on the Audio Modules.

Waveform Buss Interface.

CCs are allowed access to the WBUSS in a "round robin" buss sharing scheme. The CCs internally share their WBUSS grant between the two audio channels contained on each CC.

The ADDGEN is granted access to the WBUSS via the CSCs buss arbitration signal, $\overline{\text{TSLICE}}$. During active CC WBUSS cycles, the WP is informed via $\overline{\text{TSTAKEN}}$ that it or refresh cannot access the WBUSS.

Processor, Address Registers and Multipexers
(refer schematic CMI-31-01)

The CPU's data buss is buffered at A6 to form the internal Memory Data Buss (MDB). The CCPs data buss is connected to the MDB via bidirectional buffer B8. All onboard RAM access occurs via the MDB. B8 is only disabled during CC access by external CPUs, during ram refresh and when the CCP accesses the ADDGEN and DACs. The buss between B8 and the CCP is the peripheral buss (PB) and is where the DACs and ADDGEN are connected. External CPU data cannot get onto the PB from the MDB, and hence control the ADDGEN.

The MDB also connects to the octal latches at A8 and A7 which are the PAGE registers used in external access to the CC's RAM. Latch A7 contains the page address for P1, A8 for P2.

The addresses from the CCP, via buffers B6 and B7, the page registers, and buffer B5 (connected to the least significant 8 bits of the dual CPU buss) all connect together to form the 8 bit address buss for the dynamic RAMs.

During refresh cycles, the least significant 7 bits of the CPU address buss contain the contents of the refresh counter from the CMI-133. During CC memory accesses by external CPUs, the least significant 8 bits of the CMI address buss contains the address within the page to be accessed. The Buffer B5 is enabled during these memory cycles to form the row address for the RAMs.

When the CC is accessed, the two card select control signals, $\overline{\text{CHSEL}}$ and $\overline{\text{PAGE}}$ become active. The channel select is buffered and ANDed with the refresh signal in D9, which in turn is ANDed with the externally generated E and Q to feed the CCP. During refresh and external accesses, the CCP has its clock stopped (D9 pin 11 low), allowing a memory cycle for refreshing and updating of memory or registers.

Address Decoding
(refer schematic CMI-31-02)

The CCs control registers are decoded from the CPU buss by 1 of 8 decoder C3. It is enabled when the channels $\overline{\text{CHSEL}}$ is low and $\overline{\text{PAGE}}$ is high. From this decoder the external CPU can read the channels run status, reset the CCP, write to the PAGE registers and interrupt the CCP. The CCP loses one cycle for each access.

The LS133 at C6 decodes the CCP address buss to define the ADDGEN and DAC I/O locations.

The CCPs I/O is decoded in 2 64 byte blocks starting at FE80H. The first block contains the control latch, pitch registers, CCPs interupt status register, real time clock acknowledge and DACs. The second decodes the ADDGEN registers.

Whenever these I/O loactions are accessed, pin 9 of C6 will go low and disable the CCP's MDB transiever.

Channel B Pitch and Card Status Register
(refer schematic CMI-31-03)

The latches at E5 and E6 contain the pitch for the B audio channel. There are 3 bits to select which octave and 12 bits to select the pitch within the octave. The pitch within the octave bits feed Rate Multipliers at F6 and F5.

The Rate multipliers are clocked at 17.14 MHz. The output of this chain is pin 6 of F6. This is a train of pulses with an average frequency of the required pitch, but in a much higher octave. This clock has considerable jitter as the rate multipliers effectively drop clock pulses from the 17.14 MHz stream to change frequency.

This clock feeds a binary counter E3 that generates the clocks for 8 octaves, at 16 times the required sample rate. The 1 of 8 selector E4 uses the octave select bits to select the appropriate output of E3.

The selector feeds a divide by 16 counter which acts as a jitter filter, to average out the periods of the sample clock, to reduce distortion in the output waveform.

The output of the jitter filter, E9, is ANDed with the RUNB control signal to generate the sample clock for channel B.

The 4 bit latch and buffer at B11 and A11, are used by the external CPUs to determine the current status of the channel card. The upper 4 bits of this register allow the external CPUs to determine if an interrupt came from this channel card, that the CCP is reset, and if either of the two audio channels controlled by this card are currently being used.

The least significant 4 bits allow the CCP to send 4 bit status codes to the external CPUs.

Channel A Pitch and the control registers
(refer schematic CMI-31-04)

The pitch generation for channel A is identical to Channel B but uses the components at E8, E7, F8, F7, D6, D5 and E9.
The main pitch oscillator is bussed from the CSC as a differenti-al line. This is received at A9 by a 26LS32. This receiver then drives the two channels of pitch generating ratemultipliers.

The control latch is located at C10. This octal latch outputs all the control lines to control, channel running, zippa filter time constant, enabling of end of loop interupts and selection of the ADDGENs loops. It is reset when the CMI is turned on to stop the channels running, and making any sound before they are initialized.

The F/F at D2 is used to control the $\overline{\text{RESET}}$ input of the CCP. It can only be written to by external CPUs and is reset, the CCP stopped, at power on.

Control Voltage Generation
(refer schematic CMI-31-05)

The Control Voltage DACs all use a precision 10.000V source as their voltage reference. The +15V, -15V and analog ground (AGND) are cabled to the CCs via 26 way flat cables. There is no connection between the CCs digital ground DGND and AGND on the CC. They are connected together in the Audio Rack. The back to back diodes

and resistor on the CC connecting DGND and AGND are to stop the two from drifting too far apart if the AGND and DGND connection is broken.

The DACs cannot be accessed by external CPUs, only the CCP.

The AD7226 at F3 is a quad 8 bit voltage output DAC. It receives the 2 least significant address lines to select its 4 DACs on byte boundaries. This DAC produces 2 channels of filter cutoff and resonance Control Voltage (CV). Each is 0 to 10 volts.

The filter cutoff frequency range is from 20Hz to 20kHz with control values of 0V (00H) to 10V (FFH).

The filter resonance changes from flat, 0V (00H) to oscilation at 10V (FFH).

The AD7548s at F1 and F2 are 12 bit, right justified DACs with an 8 bit buss interface. They are connected so that the 16 bit value written to them is only latched into the 12 bit DAC after pin 15, DACOUT, is accessed as a seperate operation. The output of the current to voltage converters is a CV with the range 0 to -10 Volts. The control range is approximately 95db. 0V (X000H) is maximum attenuation. -10V (XFFFH) is no attenuation, 0db.

When a sound is being played out, its amplitude control voltage will be updated every 1ms and thus the DAC should be accessed at that rate.

These CVs have a software selectable time constant or "zippa filter" .This allows for fast attacks on envelopes and "quiet " amplitude control at other times. This is selected by the SATA and SATB control latch outputs.

Program Ram And Address control

(refer schematic CMI-31-06)

The 64k of DRAM and the address multiplexors are controlled from the PAPLA, (Page Addressing Programable Logic Array). The PAPLA has as inputs CPU buss signals and the 2 CC select signals. The DRAMs require 16 address lines and to get these into the 16 pin package they are latched internally and use an external 8 bit buss and 2 latching strobes, refered to as Row Address Strobe (/RAS) and Column address strobe (/CAS). The $\overline{\text{CAS}}$ signal also doubles as the data enable strobe. Refer to the Memory data books for exact details.

Because of the propagation delay in the PAL, the Row Address Strobe for the RAMs must be delayed in time with respect to the buss RAS signal. This is done on the CSC to generate ABRAS.

Depending on the state of the PAPLA inputs, it will enable to the DRAM's address inputs , the CCPs addresses, P1 PAGE register and the least significant bits of the external address buss, P2 PAGE register and the least significant bits of the external address buss, or the refresh address from the external address buss.

When there are no external accesses or refresh, then the CCPs address is enabled to the RAM.

$\overline{\text{CAS}}$ to the RAMs is disabled whenever an external access to the CCs control registers occurs.

RAM refreshes use dummy reads from the current refresh address.

Address Generator and Control

(refer schematics CMI-31-07.08, Address Generator and Control Signals)

The address generator (ADDGEN) calculates the sample addresses when playing out waveform ram. It calculates 24 bits of waveform address for the 2 audio channels on the card.

The ADDGEN's 24 bit output is interpreted as 23 bits of address and a "mode" bit by the WRAM. The most significant bit is used to select 8 or 16 bit sounds, and must be set accordingly when issuing loop start addresses.

Each audio channel has 2 loops associated with it. Each loop is initialized with a start address and a loop length, in samples. Two loops are needed to allow the contents of one loop to be played while the other is being initialized.

This allows such things as playing the sound in segments that are not sequential. The switching from one loop to the other only occurs at the end of the loop.

The hardware that generates the 24 bit addresses for the waveform does so 8 bits at a time. The machine is basically a small 8 bit wide RAM, B13 and B14, incrementer, B11 and B12 and logic to generate the sequence of clock and enable signals to do the required operations. The sequence of clocks results in operations on 24 bit numbers of load, store, NOP and add 1.

To do a 24 bit increment with this hardware the following sequence must be performed.

The least significant byte is loaded into the incrementer and incremented by one, by enabling the count input of the counter and clocking the counter. The ripple carry out (RCO) of the incrementer is clocked into a F/F to determine if the counter overflowed. This incremented byte is then written back to its original RAM location.

The middle byte is then loaded into the incrementer. The output of the F/F that clocked the RCO from the first byte increment is then fed into the count enable input of the incrementer and the counter clocked. This byte will only be incremented if the least significant byte overflowed. The RCO from the incrementer is then clocked into another F/F and the middle byte written back to its original RAM location.

The most significant byte is then loaded into the incrementer. The outputs of the two RCO F/Fs are ANDed together and feed into the count enable input of the incrementer, and the counter is clocked. The most significant byte will only be clocked if the middle and least significant bytes overflowed. The most significant byte is then written back to its original location.

Address increments, however, are syncronized to the channnels pitch clocks. This means that not all cycles allocated to incrementing the addresses, actually result in the address being incremented. A signal syncronizing the pitches to the ADDGEN hardware disables the incrementing of the address bytes on unused address calculation cylces.

The following is a general outline of the address calculation sequence for one channel.

```
while power applied
do
  write 24 bit waveform address to address latches A14,A12,A13

     if end of loop flag is 1, F13
       then
            reset loop counter to start
            value and clear end of loop flag
       else
           inc loop counter

     end

     if end of loop
       then
            reset wave address to start
            value and set end of loop flag
       else
            inc wave address
            clear end of loop flag
     end
end
```

Address Generator Clock generation

The state counters are E14 and E13. They are made to count to 24 by being reset after 24 clock cycles by the waveform buss syncronization logic, via signal $\overline{\text{SRES}}$. The output of the counters go to the address inputs of the control PROMS, at D14, D15 and D13. These Bipolar PROMS contain the clock and address sequences needed to control the clocking and enabling of the ADDGEN hardware. The PROM outputs are deglitched by the octal latches on their outputs, C14, C15 and C13. There are 15 clock signals generated by the clock generating PROMs. There are 6 address bits generated by the ADDGEN PROM and loop select and channel select lines.

Pitch Clock Synchronization

The sequence of F/Fs and multiplexer at F9, F11 and F15 syncronize the incrementing of waveform addresses to the rising edges of the pitch clocks. The calculation of the next sample address is carried out by the address generator on the next cycle allocated to that calculation after the rising edge of that channels pitch clock. The result is that COUNTEN goes active for one cycle.

The COUNTEN signal is used to enable the increment operation. The ADDGEN is normally going through the sequence that does the address calculations, but nothing in fact is incremented because of the inactive COUNTEN signal.

Active COUNTEN will also pull the TSTAKEN (time slice taken) active and inform the WBUSS control logic on the Waveform Processor that the next waveform memory cycle will be used by a CC.

At the end of the memory cycle the channel card will also generate the clock to clock data into the Audio Rack, via the DCLK1, DCLK2 and PCLK1, PLCK2.

The operations performed in each clock cycle are:-

0 read WADDRESS1 write to output latch 1
1 read WADDRESS2 write to output latch 2
2 read WADDRESS3 write to output latch 3

	if endofloopflag set	if no endofloopflag
3	read looplenght1	read WORDCOUNT1
4	nop	inc
5	write WORDCOUNT1	write WORDCOUNT1
6	read looplength2	read WORDCOUNT2
7	nop	inc
8	write WORDCOUNT2	write WORDCOUNT2
9	read looplength3	read WORDCOUNT3
10	nop	inc
11	write WORDCOUNT3	write WORDCOUNT3
12	nop	

The end of loop F/F is set here if the end of a loop has been reached. It is clear otherwise.

	if end of loop then set endloopflag	if not end of loop clear endloopflag
13	read STARTLOOP1	read WADDRESS1
14	nop	inc
15	write WADDRESS1	write WADDRESS1
16	read STARTLOOP2	read WADDRESS2
17	nop	inc
18	write WADDRESS2	write WADDRESS2
19	read STARTLOOP3	read WADDRESS3
20	nop	inc
21	write WADDRESS3	write WADDRESS3
22	write to control ram from proccessor	
23	nop	

The parameters are organized in the ADDGEN ram so that changing the upper two address lines to the ram, makes the ADDGEN calculate values for the other channel or the other loop.

The current loop for channel A and B is set in F/F C16 by the CCP. These F/F are set to loop1 whenever the channel is stopped.

The end of loop flags, F13, are also set when a channel is stopped. This is done because the only time the loop values are initialized is when an end of loop occurs.

When the CCP writes the loop values to the ADDGEN ram, the CCP's data and least significant 6 bits of address are latched at C11 and C12. This has to be done because the write cycle for the CCP last for 500ns while that for the ADDGEN lasts 100ns. The ADDGEN and the CCP are also not synchronized, so the CCP's data is latched until the ADDGEN is able to write it to its RAM, which happens every 2.4 microseconds. When the write to ADDGEN ram occurs, the latch containing the CCP's address (C12) is enabled instead of the addresses from the ADDGEN PROM (C13).

The 20 test pins are used to connect a logic analyser to the ADDGEN machine for debugging.

Waveform Buss Syncronization

(refer schematic CMI-31-09)

The WBUSS is controlled by the SCLK signal. The Channel cards take turns in using the WBUSS by only using the cycle following an active $\overline{\text{TSLICE}}$ signal on their $\overline{\text{TSLICE}}$ input. Each card uses half of F/F F14 as a bit in a shift register that is shifted left on each rising edge of SCLK. So that only one CC has an active $\overline{\text{TSLICE}}$ signal at a time.

The current channel being calculated by the ADDGEN is determined by half of G15. The other half of G15 is used to steer the data clock to the correct audio channel on the audio board.

CCs only use their allocated WBUSS slice, if COUNTEN is valid. If a WBUSS cycle is granted and COUNTEN is active then $\overline{\text{TSTAKEN}}$ and $\overline{\text{ATB}}$ will be generated as will the DCLKs to the Audio rack.

This will result in a read from Waveform RAM and the data being clocked into the first set of latches on the audio board for the apropriate channel. The WP generates the read and waveform address strobe signal, to the RAM on receiving the $\overline{\text{TSTAKEN}}$ signal from the channel card.

The gated sample clocks are cabled to the Audio Rack via differentian lines.

Channel Card Processor Interupts

(refer schematic CMI-31-10)

The "chantick" from the 6840 on the Channel Card Support is used to provide a real time clock reference to all channel cards. This is an FIRQ to the channel card processors at the 1 msec rate set by the CSC. The FIRQ is used to time the VCA (and VCF) ramp generating software. The rising edge of chantick sets half of F/F E1 which causes an FIRQ to the CCP. This is the only source of FIRQ on the card. The CCP resets this F/F in its interupt service routine by accessing the $\overline{\text{RTC}}$ location.

The CC can interrupt the CMI by accessing its /ACK location which will reset F/F D2. This interupt will be reset when the CMI reads the CC's status register.

There are 3 sources of IRQ on the CC to the CCP. They are the interrupt from the CMI, the end of loop interupt from channel A and the end of loop interupt from channel B. Because of this, a status register is required to determine the source of the IRQ. This is half of the buffer at A5. Valid interupts are cleared when the status register is read. Shortly after the status read pulse occurs, pin 8 of E1 will go low briefly. This will reset the interupt F/Fs D1 and half of F10, that were valid before the previos CCP cycle. This double buffering of interupts is to stop interupts from being lost, if they occurred during the status register read.

Channel Card Equates and Definitions
Addressing from external CPUs.
The hardware equates for the card are:-

```
CNTRL  EQU    2  0= reset CCP
STAT   EQU    0  processor interupt and status byte
PAG2   EQU    5  Page of CC RAM to be accessed by P2
PAG1   EQU    6  Page of CC RAM to be accessed by P1
ATT    EQU    A  access to interupt CCP
```

STAT bit definitions

```
0    command bit
1        "
2        "
3        "
4    RUNB
5    RUNA state
6    channel card processor reset line
7    active interupt from this channel card
```

Support card Mask related equates

```
CHANS  EQU    E080  access channels via the masks
CHAN1  EQU    E000  direct access to channel 1
CHAN2  EQU    E010        "
CHAN3  EQU    E020        "
CHAN4  EQU    E030        "
CHAN5  EQU    E040        "
CHAN6  EQU    E050        "
CHAN7  EQU    E060        "
CHAN8  EQU    E070  direct access to channel 8
```

On board Processor's I/O

The Channel Card Processors equates

Control voltage DACs

```
AFILT  EQU    FE80   8 bit filter cutoff channel A
BFILT  EQU    FE82   8 bit filter cutoff channel B
ARES   EQU    FE81   8 bit filter Q at cutoff
BRES   EQU    FE83   8 bit filter Q at cutoff
AVOL   EQU    FE85   12 bit right justified
BVOL   EQU    FE89   12 bit right justified
DACOUT EQU    FE8C   access to output above two to DACs
```

Rate multipliers

```
APITCH EQU    FE90   15 bits, right justified
BPITCH EQU    FE92   15 bits, right justified
```

Control latch

```
CNTRL  EQU    FE8E
 0      channel B loop select  0= loop 1
 1      channel A loop select  0= loop 1
 2      enable end of loop channel B IRQ
 3      enable end of loop channel A IRQ
 4      volume zipper noise filter B (1= on)
 5      volume zipper noise filter A (1= on)
 6      run channel B
 7      run channel A
```

Address Generator

Each location is write only and each byte must be written no faster than every 2.4 microseconds
All are 24 bit values, stored least significant byte first.
The most significant bit of the start addresses determine 8 or 16 bit accesses
0 = 8 bit
1 = 16 bit

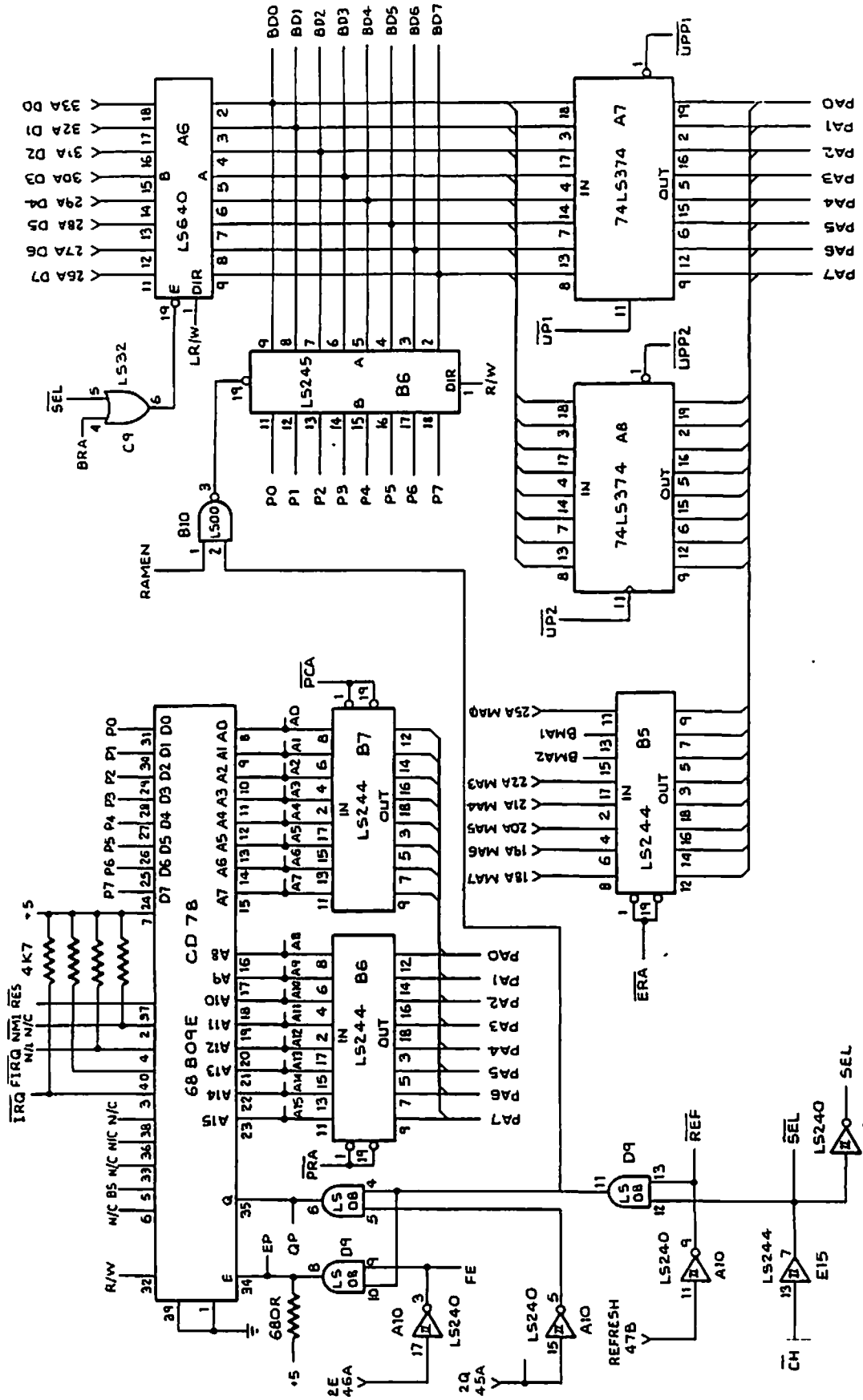
```
ASTRT1 EQU    FEC0   start address channel A loop 1
ALOOP1 EQU    FEC3   length loop 1 channel A
ASTRT2 EQU    FED0   start address channel A loop 2
ALOOP2 EQU    FED3   length loop 2 channel A

BSTRT1 EQU    FEE0   start address channel B loop 1
BLOOP1 EQU    FEE3   length loop 1 channel B
BSTRT2 EQU    FEF0   start address channel B loop 2
BLOOP2 EQU    FEF3   length loop 2 channel B
```

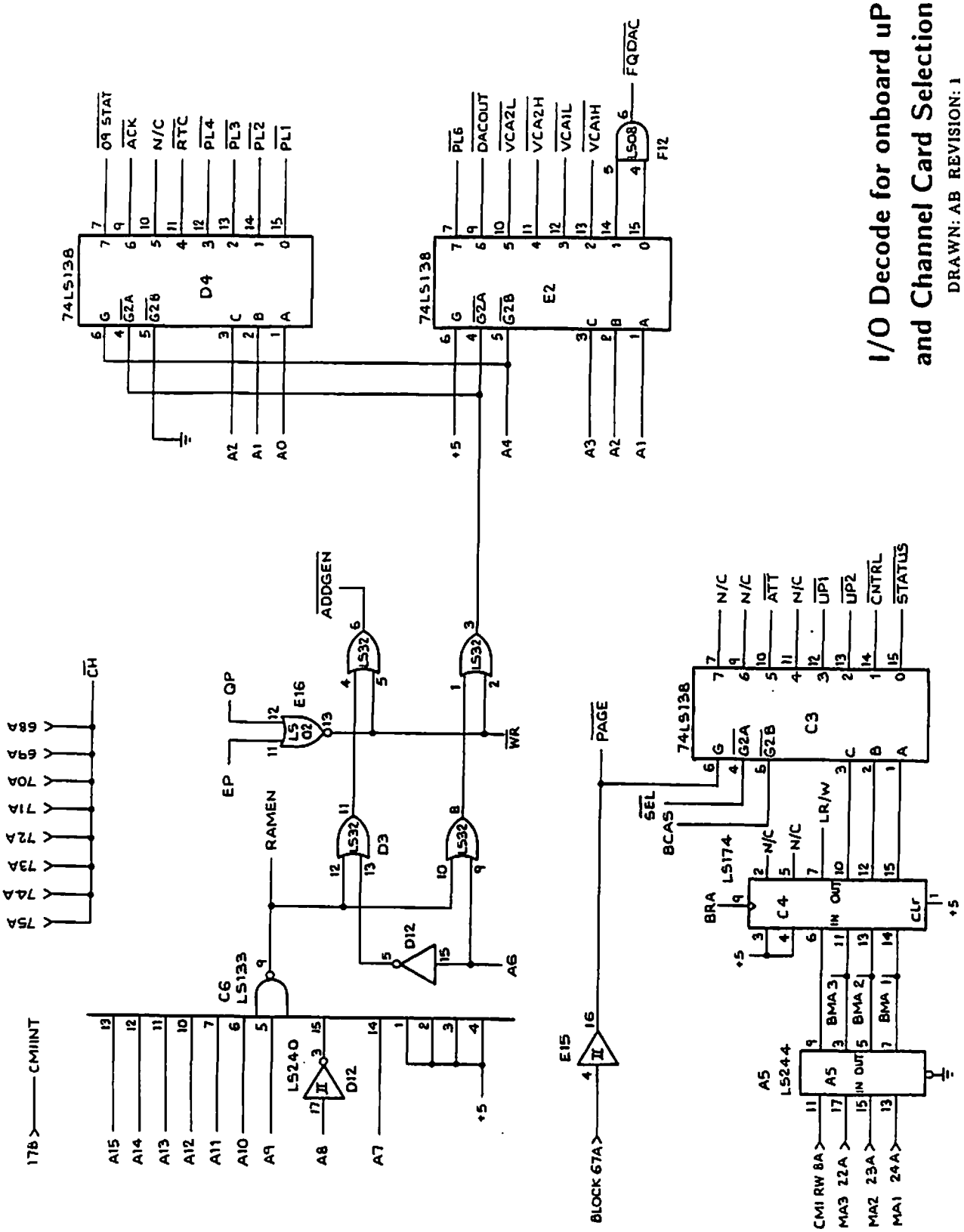
Communications Interrupts

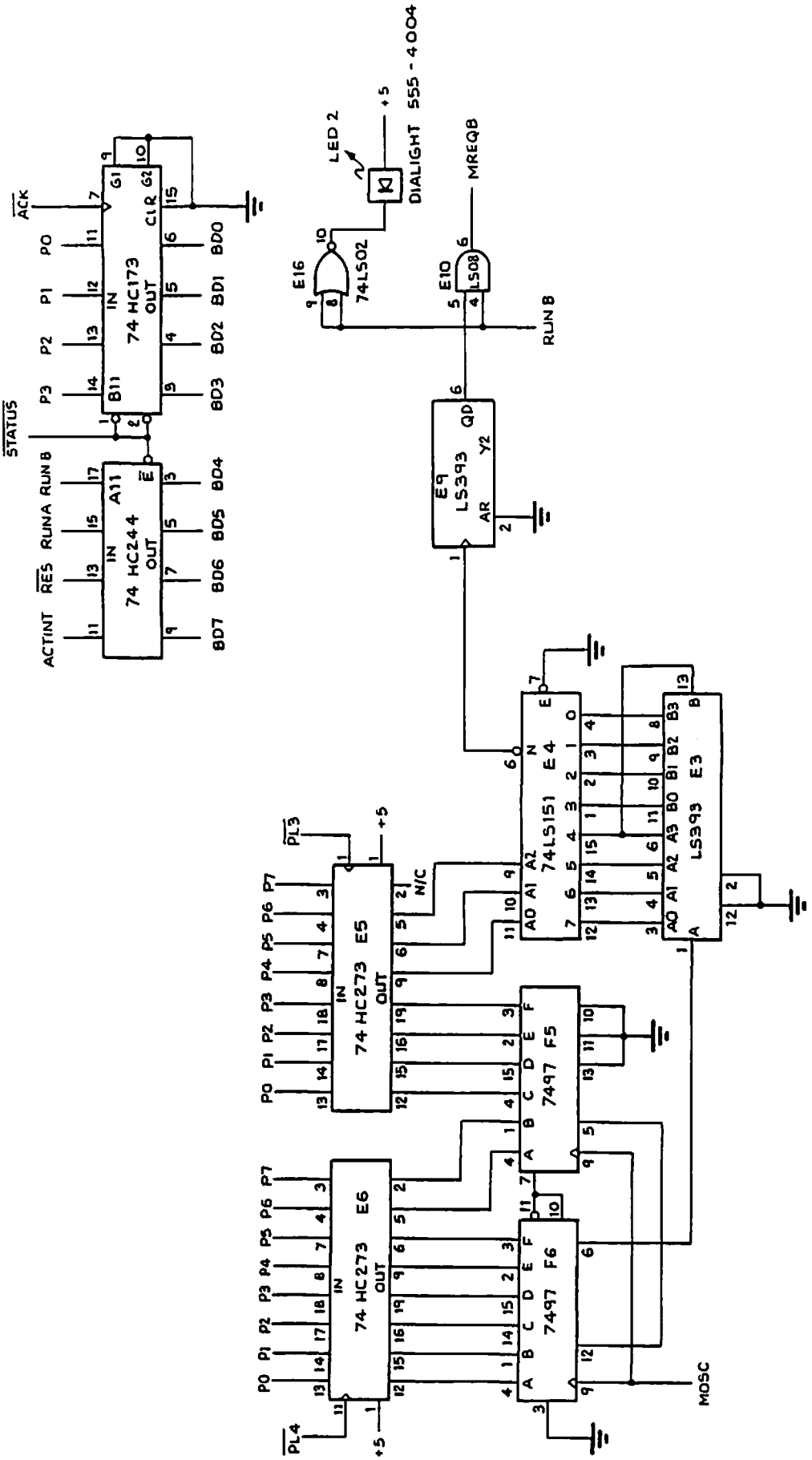
```
COM    EQU    FE96  read or write to interrupt CMI

STAT   EQU    FE97  read for interrupt status
*      Bit definitions
*      4      chantick happened
*      5      end of loop B  interrupt happened
*      6      end of loop A  interrupt happened
*      7      command interrupt happened

RTC    EQU    FE94  access to reset real time FIRQ
```

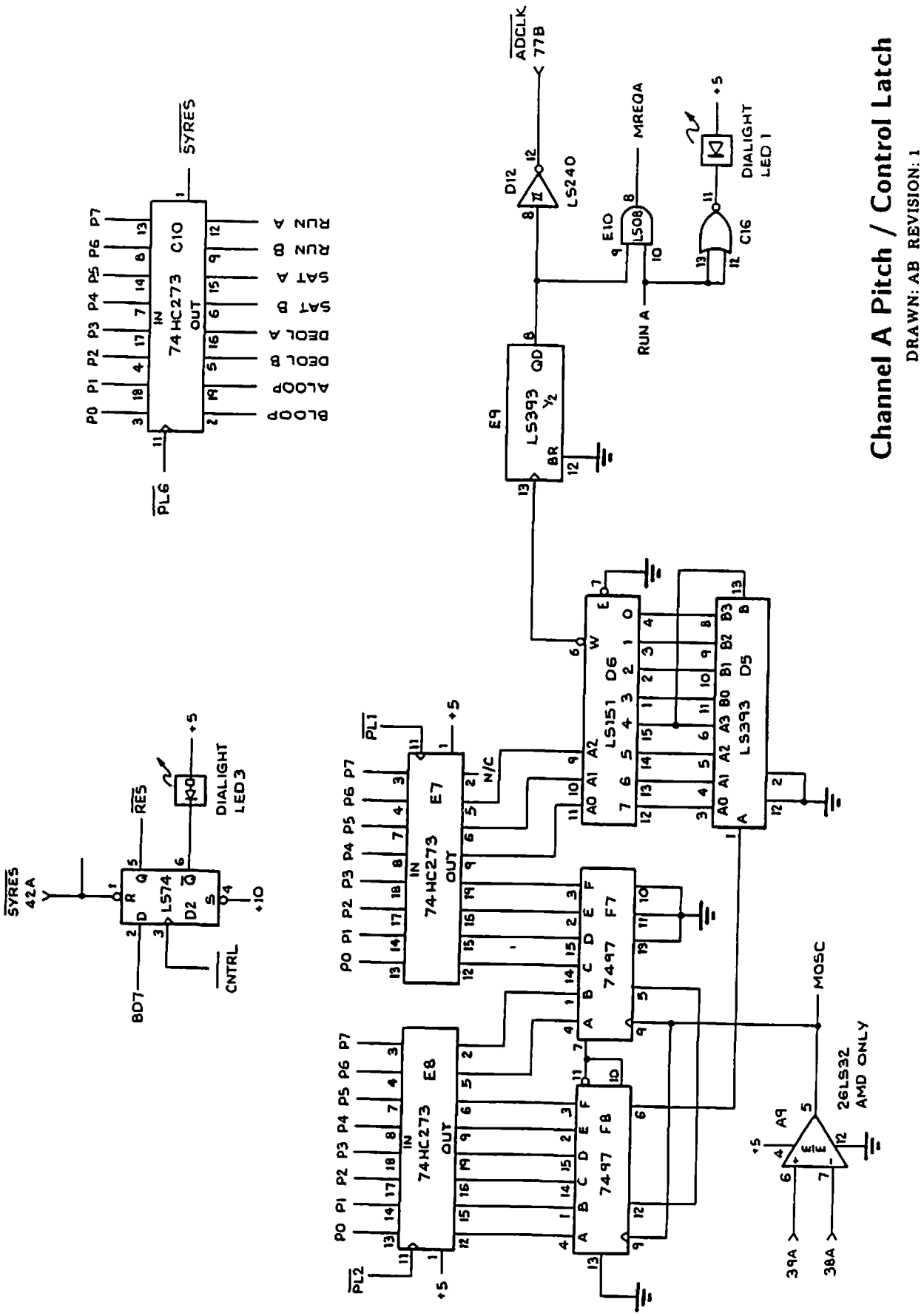
Address Generator 64 Byte RAM Organization.
Unlabeled locations are unused.

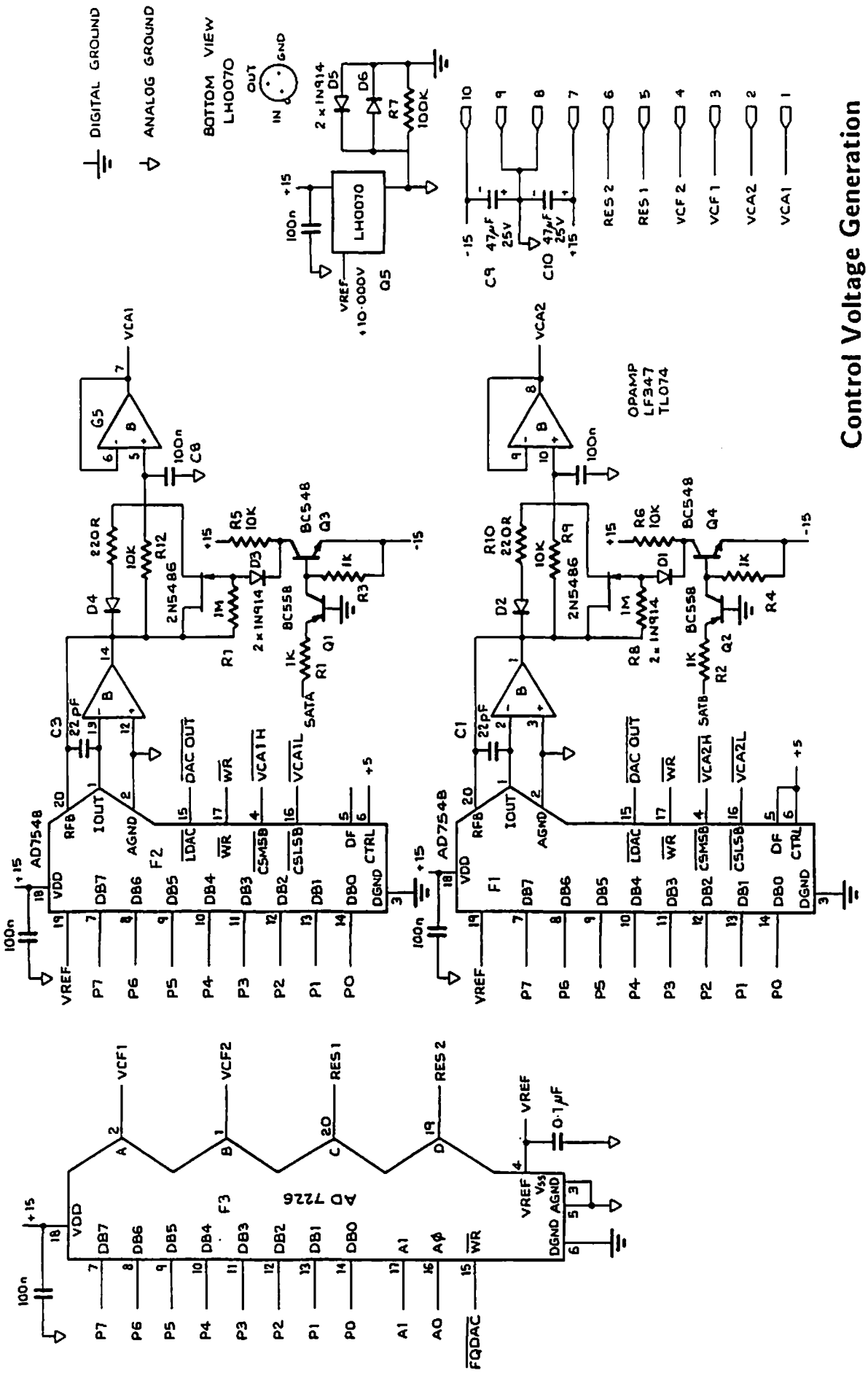
	Channel A		Channel B
0	start loop1 lsb	20	start loop1 lsb
1	" mb	21	" mb
2	" msb	22	" msb
3	loop length 1 lsb	23	loop length 1 lsb
4	" mb	24	" mb
5	" msb	25	" msb
6	word count 1 lsb	26	word count 1 lsb
7	" mb	27	" mb
8	" msb	28	" msb
9	waddress lsb	29	waddress lsb
A	" mb	2A	" mb
B	" msb	2B	" msb
C		2C	
D		2D	
E		2E	
F		2F	
10	start loop 2 lsb	30	start loop 2 lsb
11	" mb	31	" mb
12	" msb	32	" msb
13	loop length 2 lsb	33	loop length 2 lsb
14	" mb	34	" lsb
15	" msb	35	" msb
16		36	
17		37	
18		38	
19		39	
1A		3A	
1B		3B	
1C		3C	
1D		3D	
1E		3E	
1F		3F	

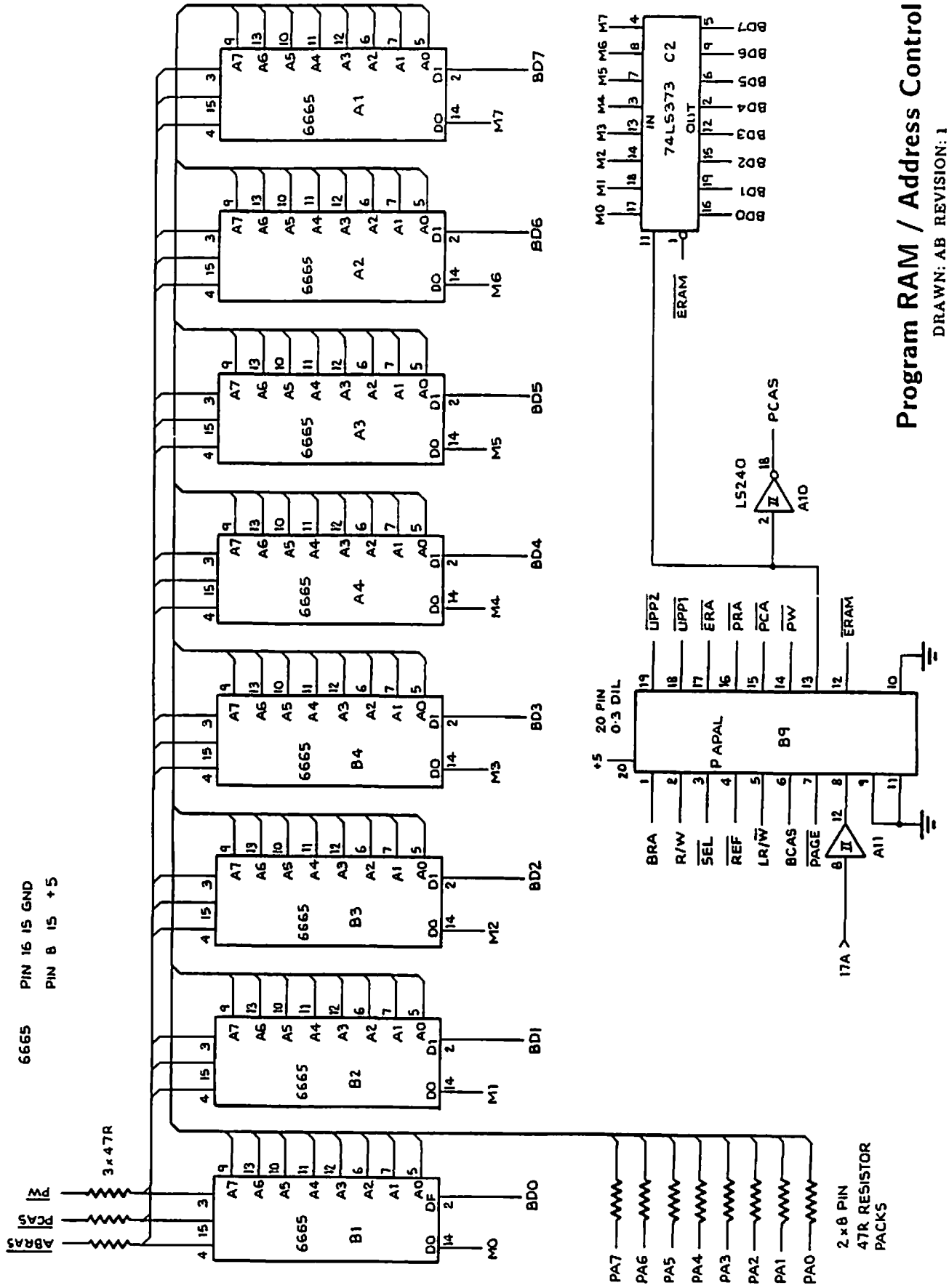

Processor, Data Buffers, and Pointer Registers
DRAWN: AB REVISION: 1
68 B09E CD 78
LS640 A6
LS245 B6
74LS374 A7
74LS374 A8
LS244 B7
LS244 B6
LS244 B5
LS244
LS240
LS32
LS00
LS08
4K7
68OR
RAMEN
R/W
LR/W
SEL
BRA
REF
REFRESH 47B
DRAS 47A
RA 37A
CAS 34A
ERA
PCA
PRA
UPP1
UPP2
UP1
UP2

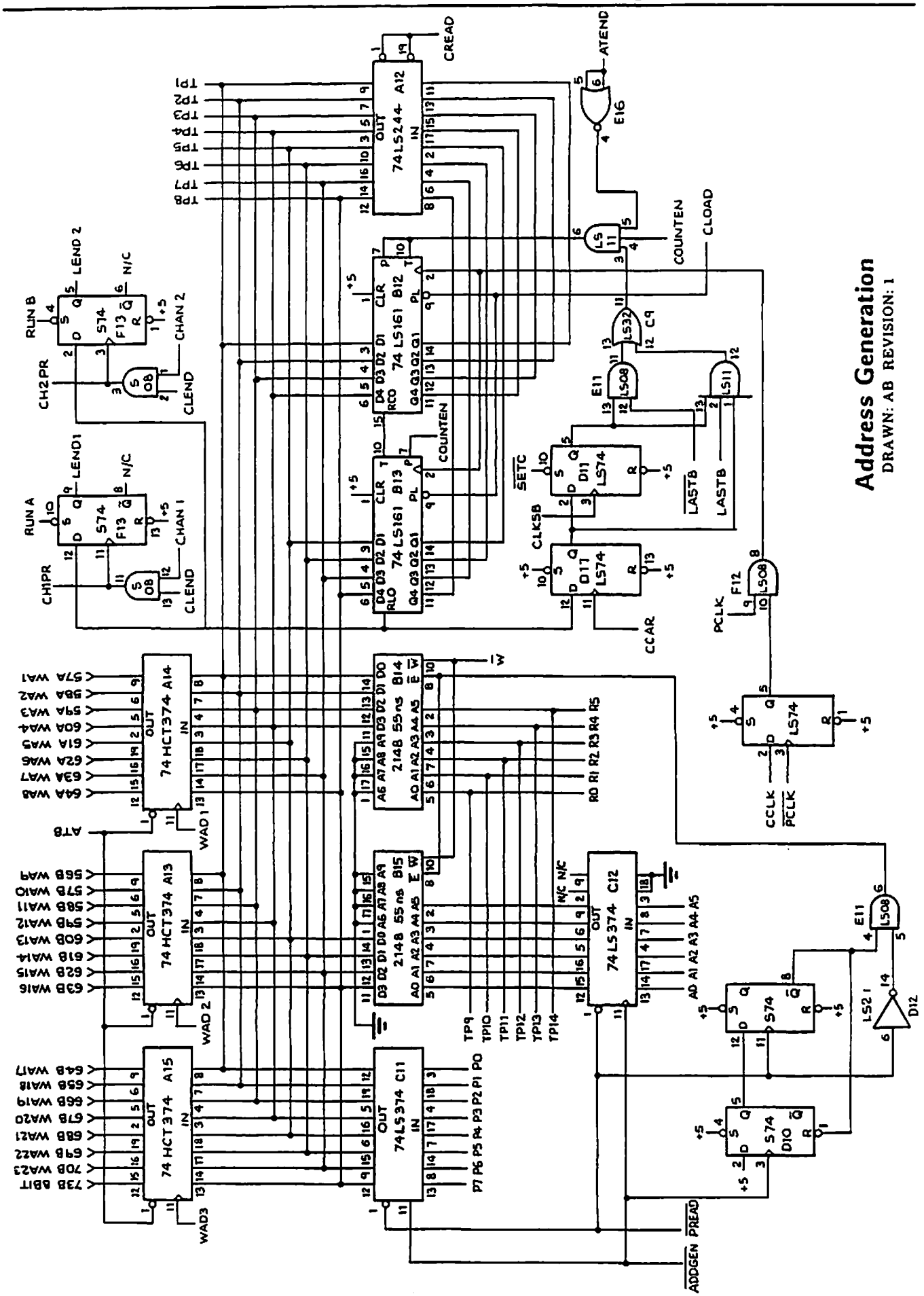

I/O Decode for onboard uP
and Channel Card Selection
DRAWN: AB REVISION: 1
74LS138
D4
E2
C3
C4
A5
C6
D3
D12
E15
E16
F12
LS174
LS240
LS244
LS133
LS32
LS08
LS02
09 STAT
ACK
N/C
RTC
PL4
PL3
PL2
PL1
PL6
DACOUT
VCA2L
VCA2H
VCA1L
VCA1H
FQDAC
ATT
UP1
UP2
CNTRL
STATUS
ADDGEN
RAMEN
QP
EP
WR
PAGE
SEL
BCAS
BRA
LR/W
BMA3
BMA2
BMA1
CMIINT
BLOCK 67A
CMI RW 8A
MA3 22A
MA2 23A
MA1 24A
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A4
A3
A2
A1
A0
+5

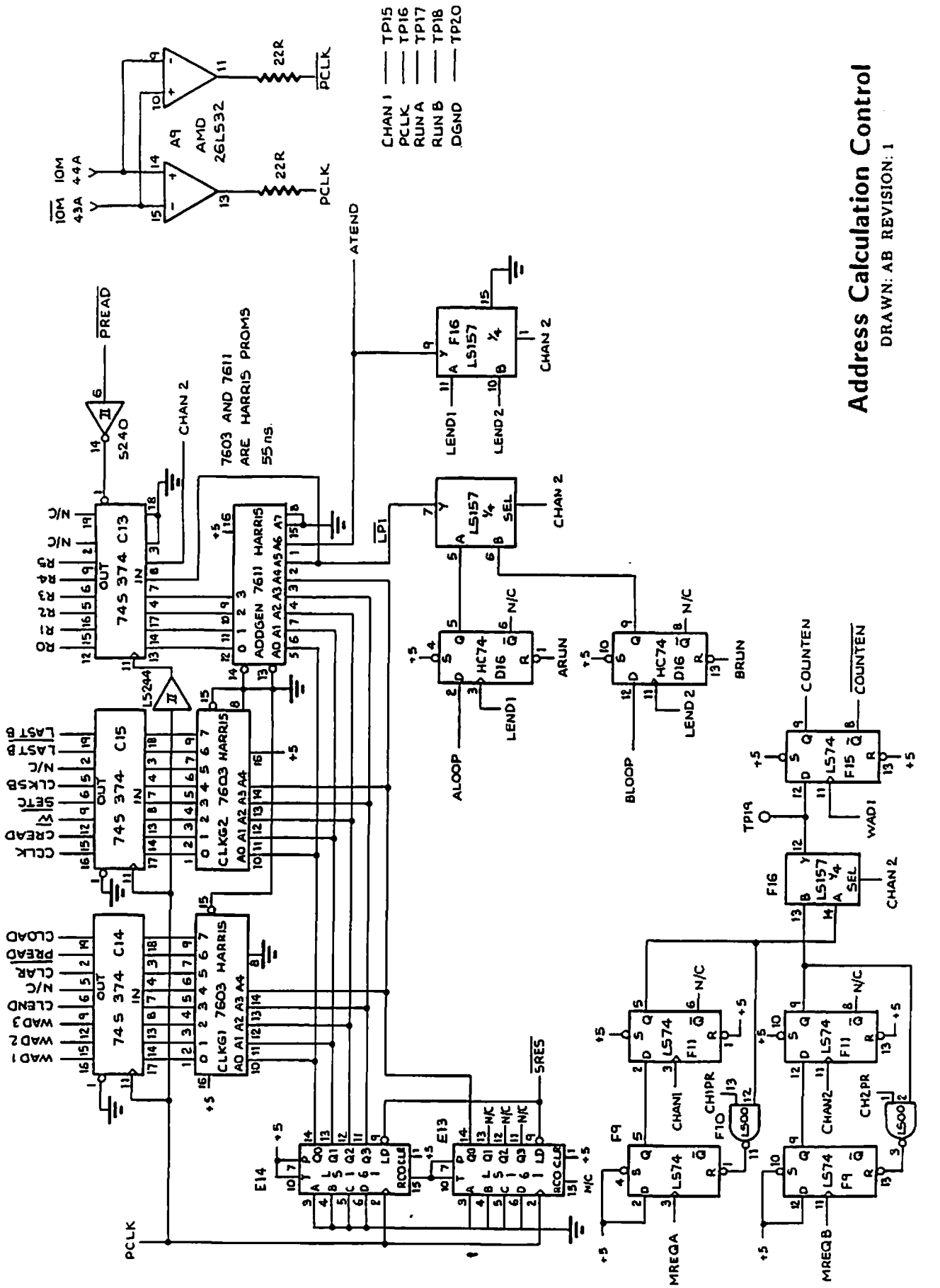

Channel B Pitch Generation
DRAWN: AB REVISION: 1
ACK
STATUS
RUN B
RUN A
RES
ACTINT
P0
P1
P2
P3
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
74 HC173
74 HC244
LED 2
DIALIGHT 555 - 4004
+5
E16
74LS02
E10
LS08
MREQB
RUN B
E9
LS393
E4
74LS151
E3
LS393
E5
74 HC273
N/C
E6
74 HC273
F5
7497
F6
7497
PL3
PL4
MOSC

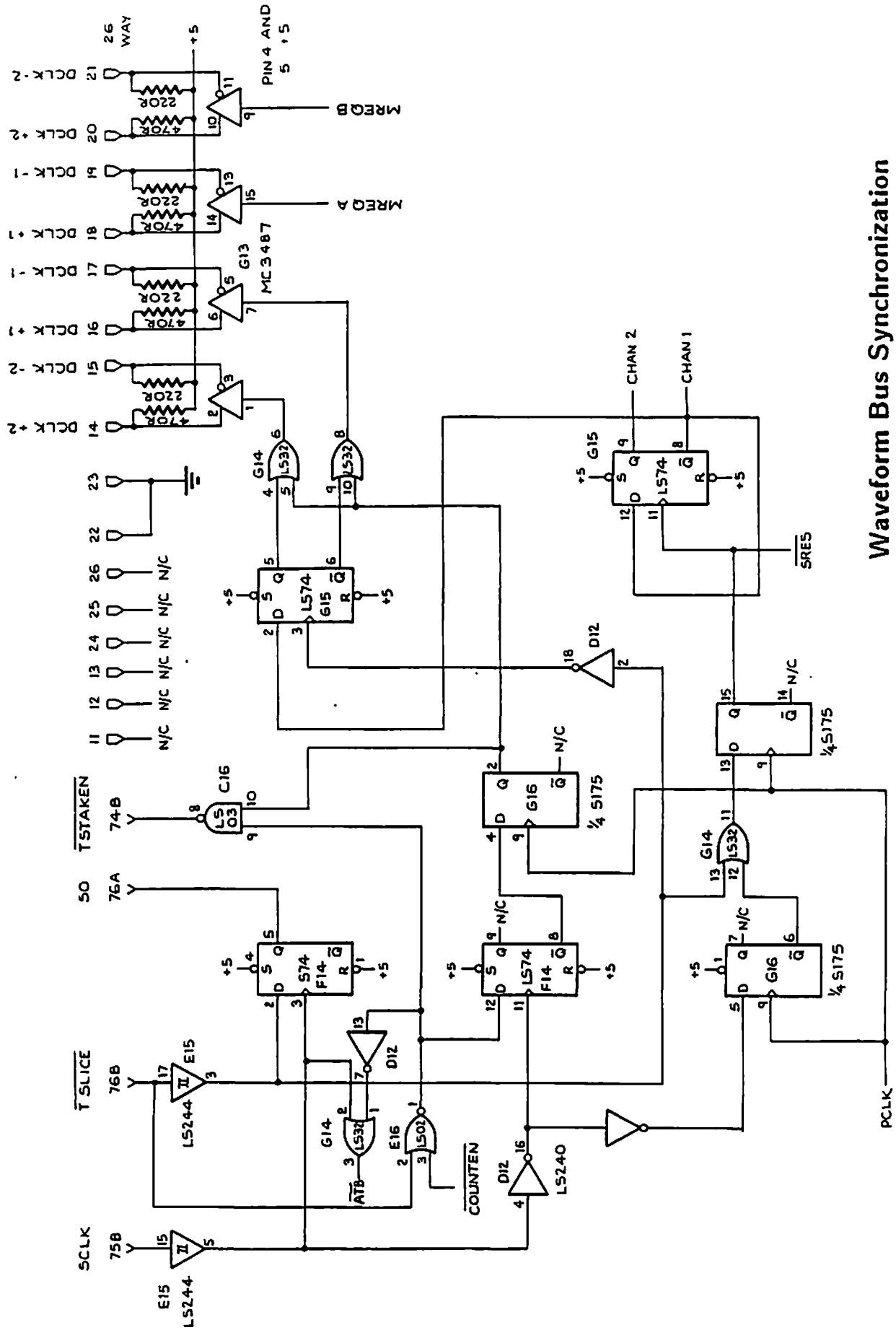

Channel A Pitch / Control Latch
DRAWN: AB REVISION: 1
SYRES
C10
74HC273
IN
OUT
P0 P1 P2 P3 P4 P5 P6 P7
BLOOP
ALOOP
DEOL B
DEOL A
SAT B
SAT A
RUN B
RUN A
PL6
ADCLK
77B
D12
LS240
MREQA
E10
LS08
DIALIGHT
LED 1
C16
RUN A
E9
LS393
QD
BR
PL1
E7
N/C
PL2
E8
F7
F8
7497
D6
LS151
D5
LS393
MOSC
A9
26LS32
AMD ONLY
39A
38A
SYRES
42A
LS74
RES
DIALIGHT
LED3
CNTRL
BD7

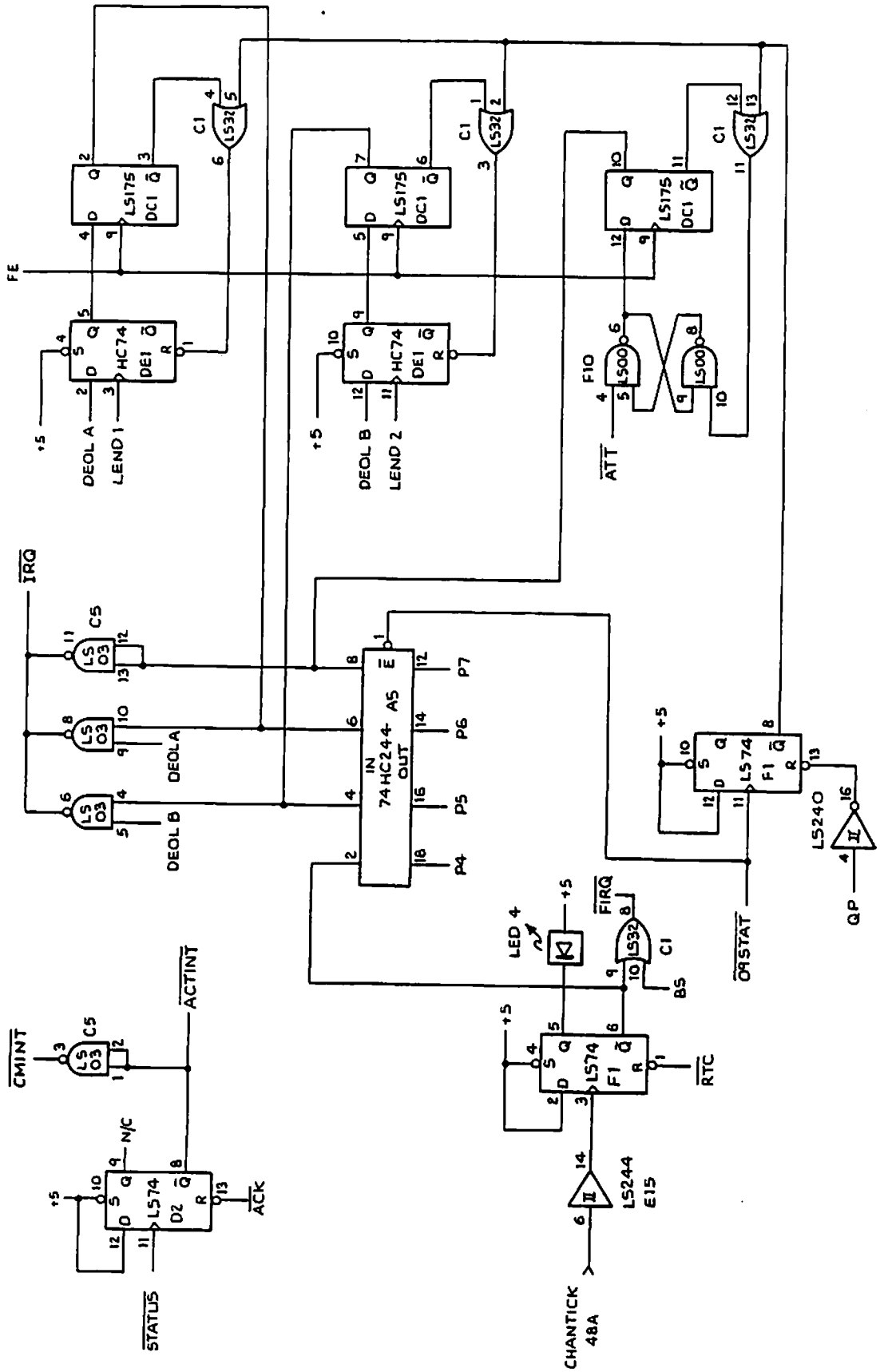

Control Voltage Generation
DRAWN: AB REVISION: 1
DIGITAL GROUND
ANALOG GROUND
BOTTOM VIEW
LH0070
OUT
IN
GND
2 x 1N914
D5
D6
R7
100K
100n
+15
LH0070
VREF
+10.000V
Q5
-15
C9
47µF
25V
C10
47µF
25V
+15
RES 2
RES 1
VCF 2
VCF 1
VCA2
VCA1
OPAMP
LF347
TL074
VCA1
VCA2
G5
100n
C8
220R
10K
R12
R10
R9
R5
10K
R6
10K
BC548
Q3
Q4
1K
R3
R4
D4
D2
2N5486
1M
D3
D1
2 x 1N914
R1
R8
BC558
Q1
Q2
SATA
SATB
C3
22pF
C1
AD7548
F2
F1
RFB
IOUT
AGND
VDD
DGND
CTRL
DF
LDAC
WR
CSMSB
CSLSB
DAC OUT
VCA1H
VCA1L
VCA2H
VCA2L
+5
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
VREF
P7
P6
P5
P4
P3
P2
P1
P0
VCF1
VCF2
RES1
RES2
0.1µF
AD 7226
F3
A1
A0
FQDAC


Program RAM / Address Control
DRAWN: AB REVISION: 1
6665 PIN 16 IS GND
PIN 8 IS +5
3x47R
ABRAS
PCAS
PZ
B1
B2
B3
B4
A1
A2
A3
A4
BD0
BD1
BD2
BD3
BD4
BD5
BD6
BD7
M0
M1
M2
M3
M4
M5
M6
M7
74LS373 C2
IN
OUT
ERAM
LS240
A10
PCAS
B9
P APAL
20 PIN 0·3 DIL
+5
BRA
R/W
SEL
REF
LR/W
BCAS
PAGE
A11
17A
PA7
PA6
PA5
PA4
PA3
PA2
PA1
PA0
2x8 PIN 47R RESISTOR PACKS

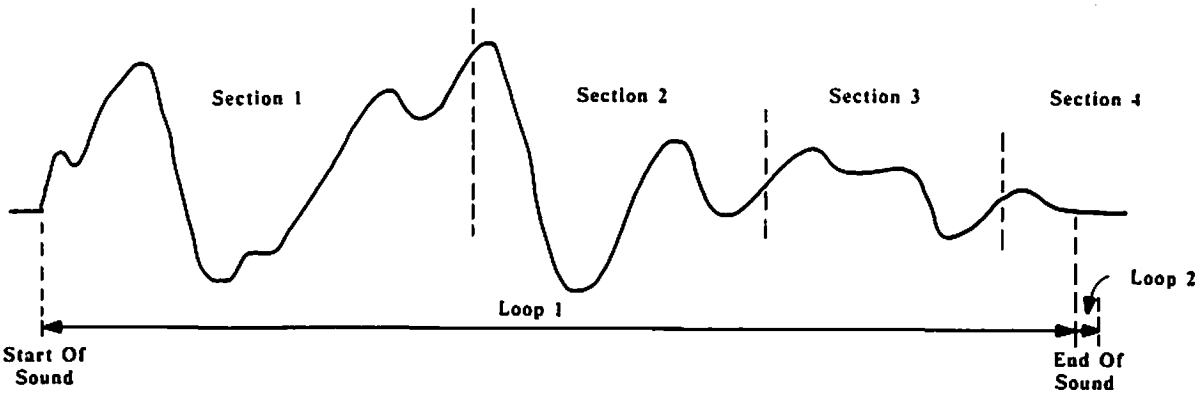

Address Generation
DRAWN: AB REVISION: 1
CREAD
ATEND
E16
74LS244 A12
TP1
TP2
TP3
TP4
TP5
TP6
TP7
TP8
COUNTEN
CLOAD
74LS161 B12
74LS161 B13
RUN B
LEND 2
N/C
CHAN 2
CH2PR
CLEND
RUN A
LEND1
CHAN 1
CHIPR
SETC
LASTB
CLKSB
CCAR
PCLK
CCLK
PCLK
74 HCT374 A14
74 HCT374 A13
74 HCT374 A15
74LS374 C11
74LS374 C12
2148 55ns B14
2148 55ns B15
WAD1
WAD2
WAD3
ATB
TP9
TP10
TP11
TP12
TP13
TP14
PREAD
ADDGEN

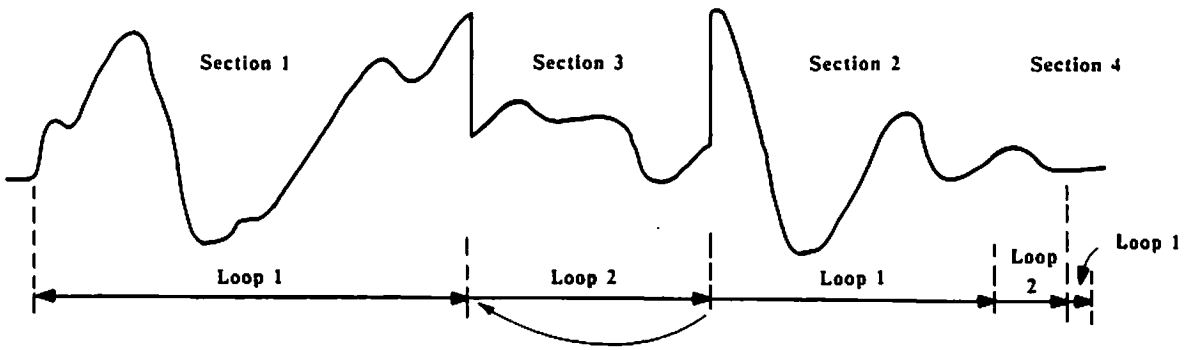

Address Calculation Control
DRAWN: AB REVISION: 1
CHAN 1 — TP15
PCLK — TP16
RUN A — TP17
RUN B — TP18
DGND — TP20
AMD 26LS32
A9
22R
PCLK
IOM 43A
IOM 44A
ATEND
PREAD
5240
CHAN 2
7603 AND 7611 ARE HARRIS PROMS 55 ns.
745 374 C13
745 374 C15
745 374 C14
ADDGEN 7611 HARRIS
CLKG2 7603 HARRIS
CLKG1 7603 HARRIS
LS244
LS157
F16
LEND1
LEND2
LP1
HC74
D16
ARUN
BRUN
ALOOP
BLOOP
LS74
F15
F11
F9
F10
COUNTEN
WAD1
TP19
CHAN1
CHAN2
CH1PR
CH2PR
LS00
MREQA
MREQB
SRES
PCLK
N/C
+5

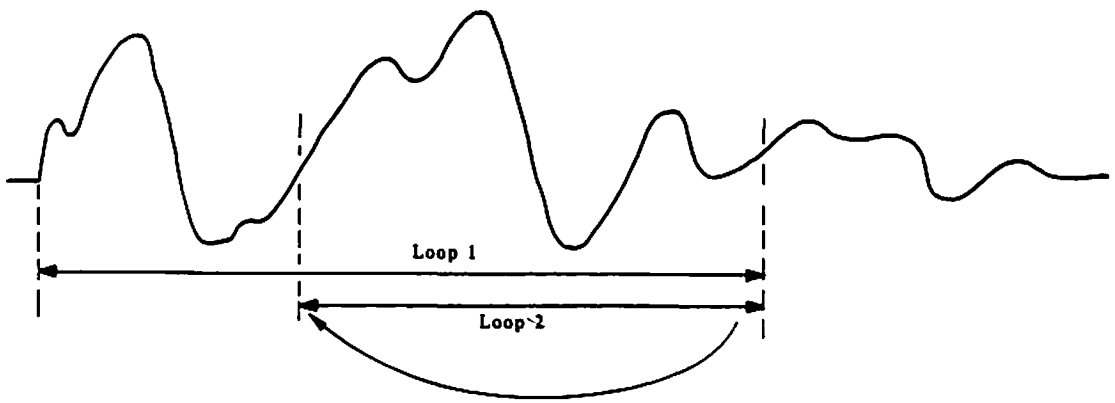

Waveform Bus Synchronization
DRAWN: AB REVISION: 1
26 WAY
+5
PIN 4 AND 5 +5
DCLK -2
DCLK +2
DCLK -1
DCLK +1
DCLK -1
DCLK +1
DCLK -2
DCLK +2
220R
470R
MREQ B
MREQ A
G13
MC3487
G14
LS32
CHAN 2
CHAN 1
G15
LS74
SRES
G15
LS74
N/C
D12
LS240
LS175
C16
LS 03
TSTAKEN
74B
50
76A
F14
LS74
G16
LS175
TSLICE
76B
E15
LS244
G14
LS32
E16
LS02
ATB
COUNTEN
SCLK
75B
PCLK


Interrupts
DRAWN: AB REVISION: 1
FE
DEOL A
LEND 1
DEOL B
LEND 2
ATT
F10
IRQ
DEOL A
DEOL B
74HC244
P7
P6
P5
P4
LED 4
FIRQ
BS
OPSTAT
QP
LS240
RTC
CHANTICK
48A
LS244
E15
CMINT
ACTINT
STATUS
ACK
N/C

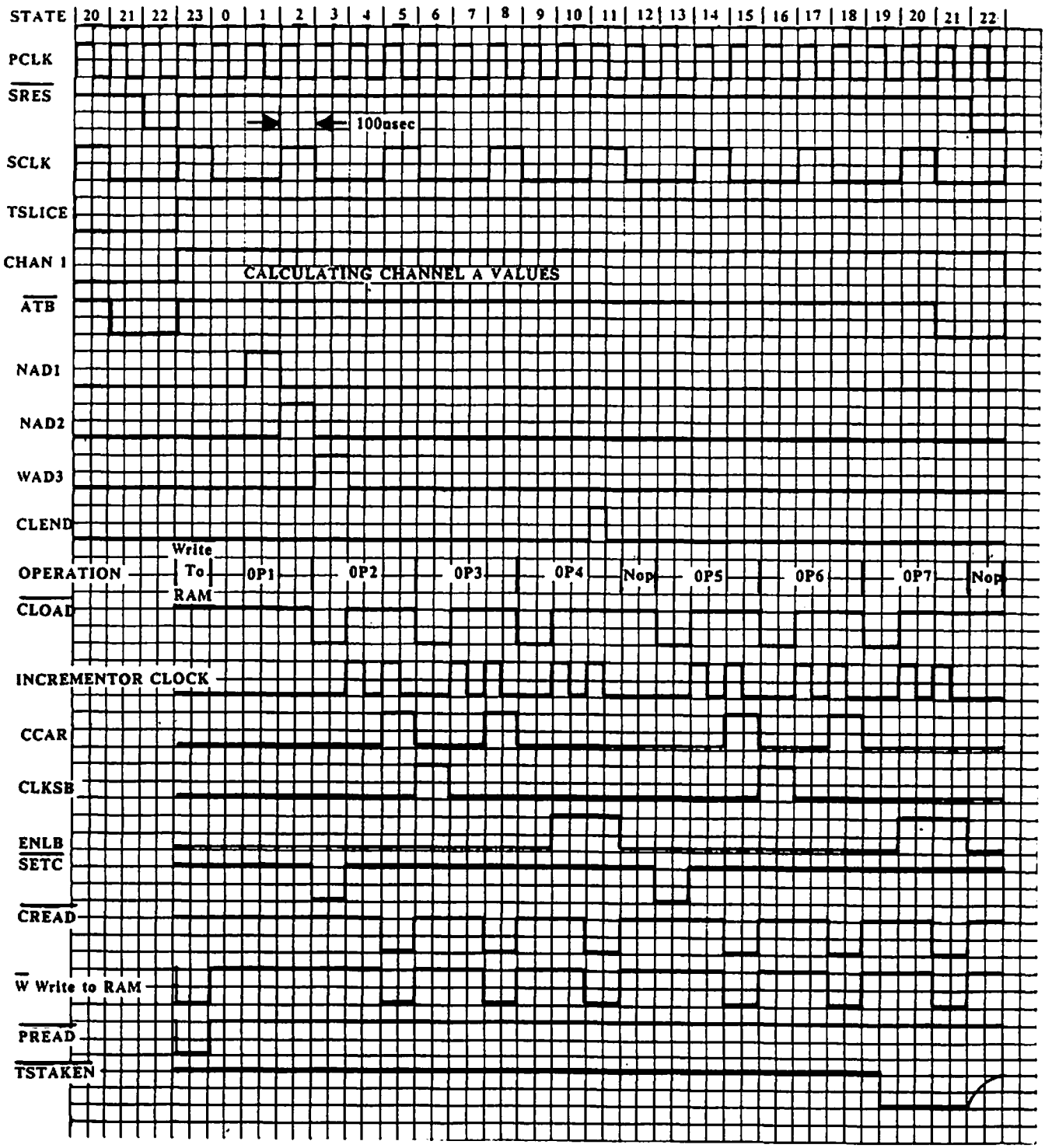
Address Generator Loops



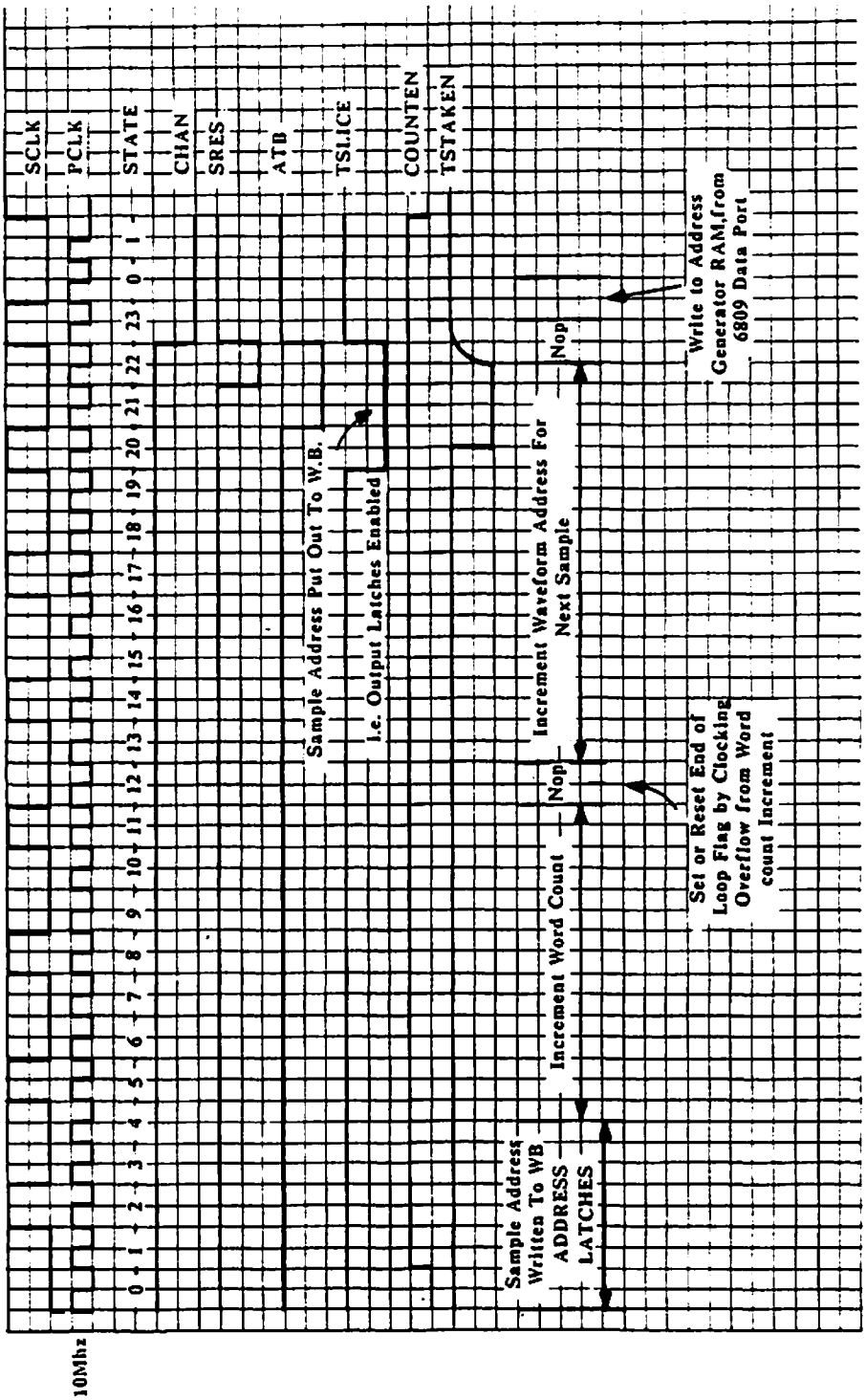


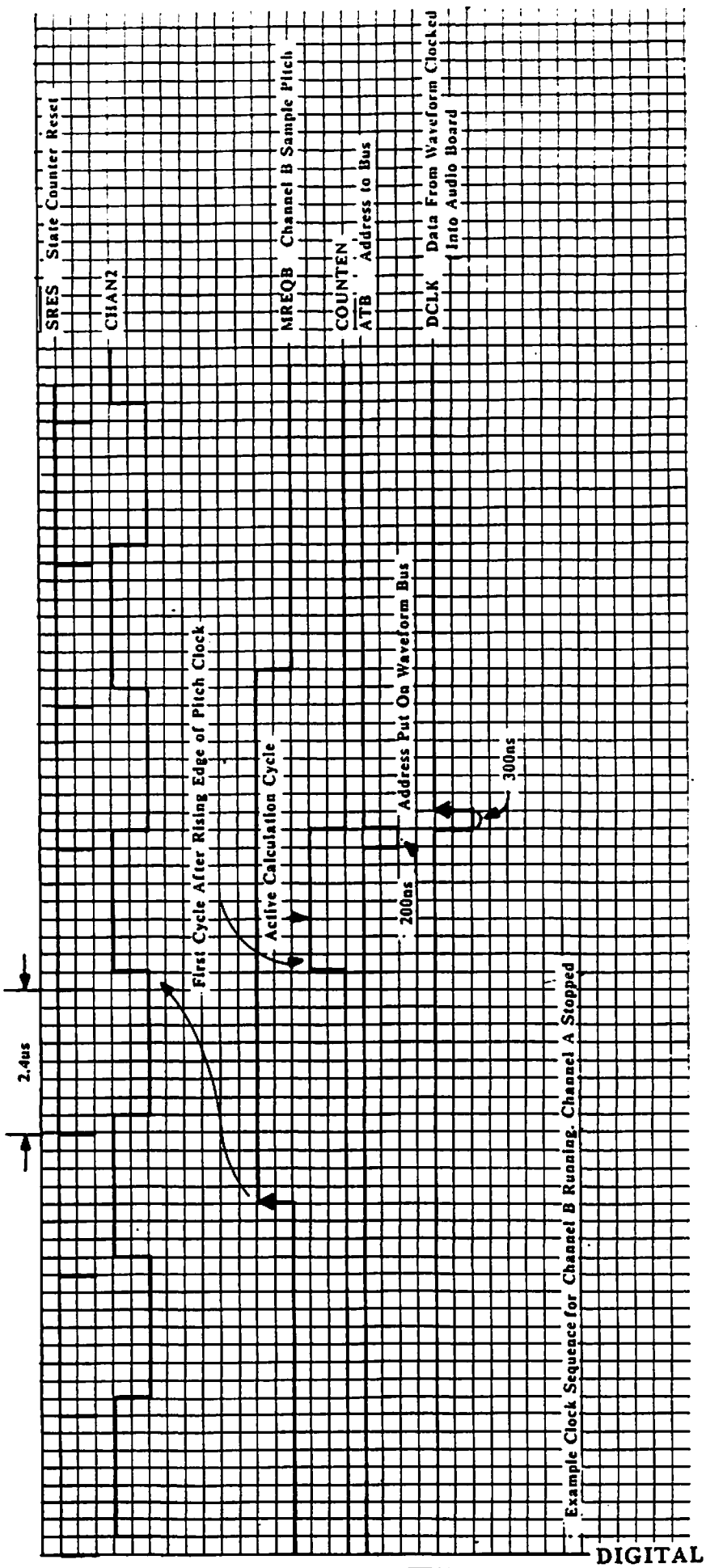


Address Generator Control Signals

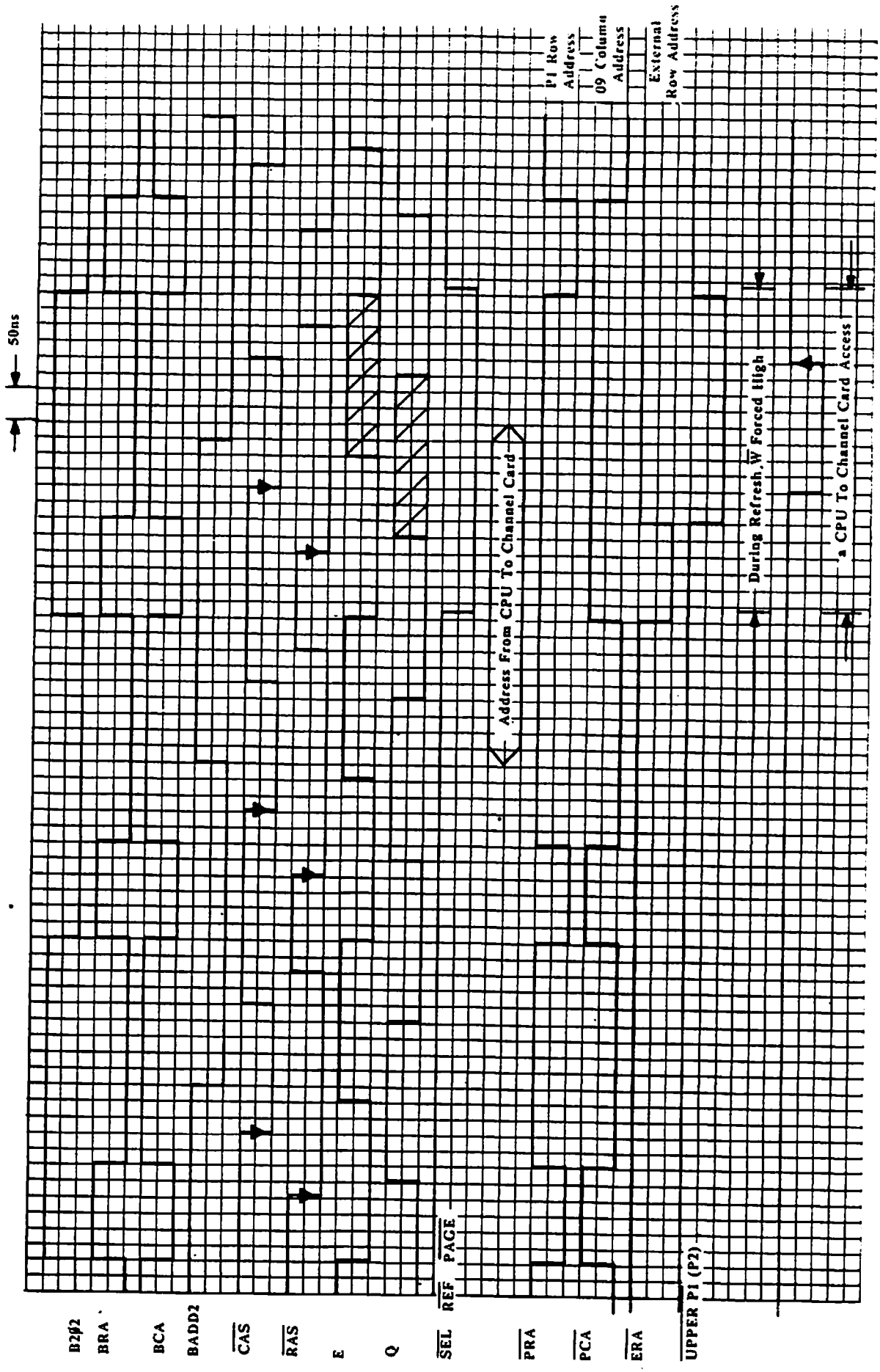


An active address generator cycle





The operation of count enable

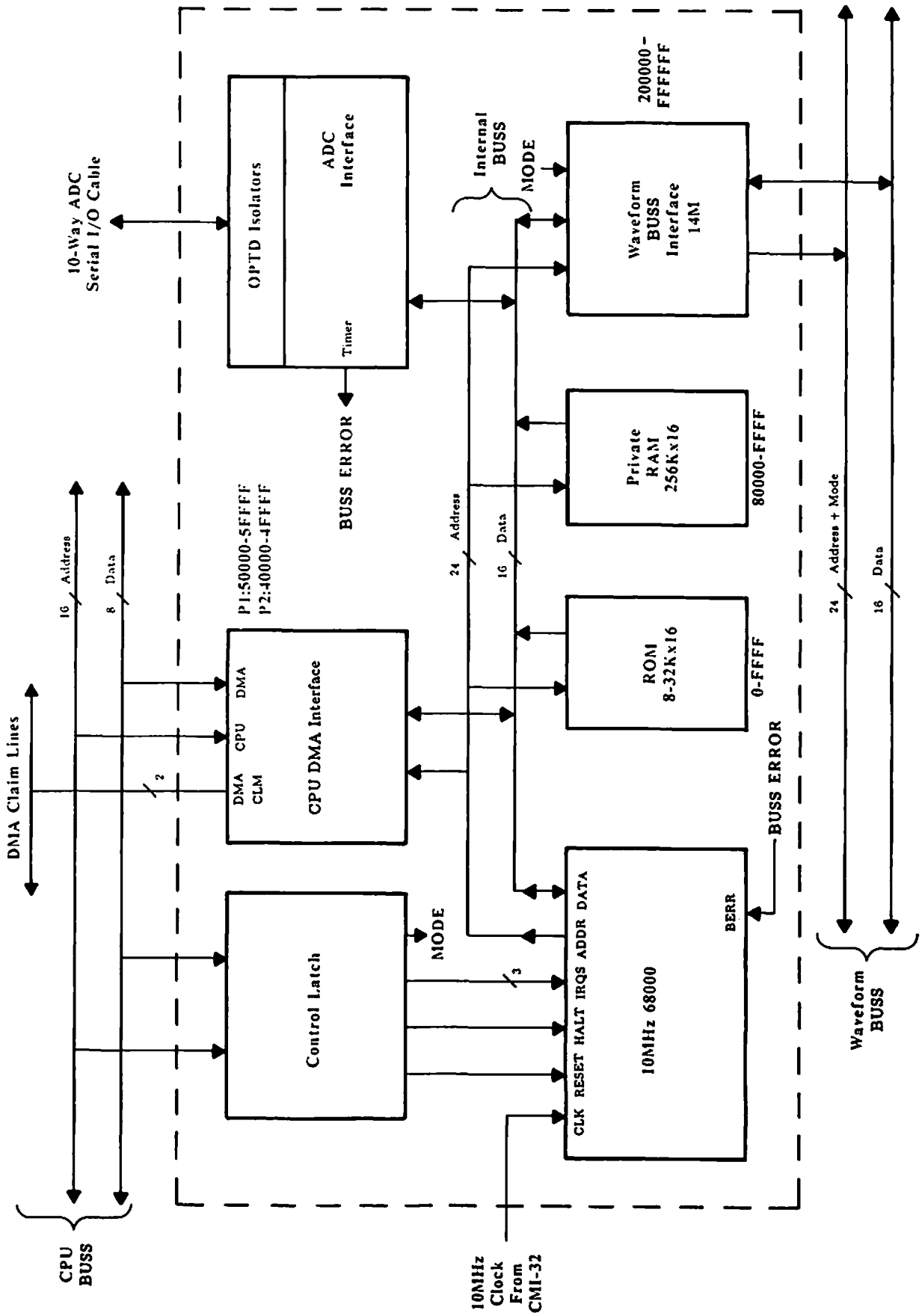


CMI Bus Interface

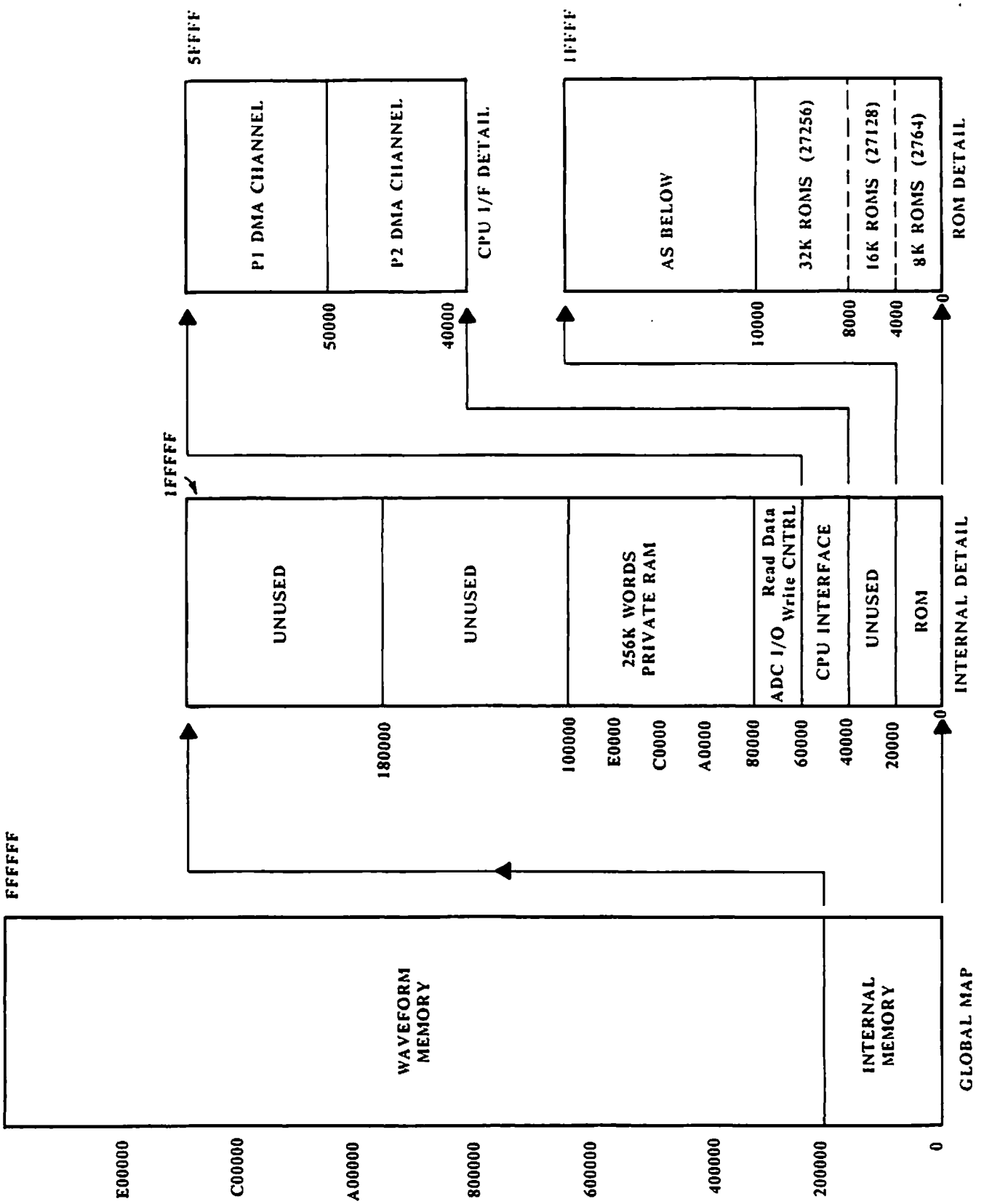
CMI-33 2.12

Waveform Processor

Block diagram



Hardware Memory Map



Terminology
WP: Waveform Processor
WRAM: Waveform RAM (2M - 14M bytes)
CPU: Dual 6809 processor system usually running OS9.
System RAM: Q256 256K memory with MMU used by the CPU.
Hexidecimal numbers are in the form nnnnnnH.
Active low signal names are preceded by a "/" character.

Introduction

The Waveform Processor (WP) is the only intelligent device which has access to the Waveform Buss and is the only device which can write to WRAM. Its primary job therefore is loading waveform memory with data for the channel cards to play, and processing that data if required. The sound data comes either from the Analog to Digital Convertor (ADC) interface or from the CPU, or the WP can construct its own waveforms . The WP can access the entire CPU system memory so in the latter case, the CPU usually gets data from disk and places it in system RAM then the WP moves it from there to the Waveform RAM. The ability of the WP to access the CPU buss means that it can also directly drive any hardware on that buss if it so desires (ACIAs, disk controllers, channels etc. etc.). The system RAM is also the means of communication between the WP and the CPU.

The facilities built in to the WP card are as follows:

- 10MHz 68000 processor
- ROM 8K/16K/32K x 16 bits.
- Private RAM 256K x 16 bits.
- DMA interface to the CPU bus
- Waveform Buss interface
- ADC serial I/O interface
- 6 bit control latch

ROMs

The pair of on-board 8-bit ROMs are arranged in parallel to provide 16 bit code. Presently 2764 ROMs are installed, occupying the bottom 8K words of 68000 address space (see memory map). The bottom 400H bytes are reserved for 68000 exception vectors including reset PC and Supervisor Stack Pointer which are loaded when the processor comes out of reset. The rest of the ROMs are occupied by the 68000 monitor, 68000/CPU IO routines, and self tests. Refer to software documentation for further information on these items.

RAM

The 256K words of Private RAM on the CMI-33 is accessible only by the 68000. Its base address is 80000H (see memory map). 64K RAM chips are compatible but it is intended that all production boards will be equipped with 256K RAMs.

Since the memory consists of dynamic RAM, refresh must be performed. Refresh requests are generated by a pulse which comes from the Channel Support Card CMI-32 every 16uS.

If the on-board buss is available due to the 68000 performing internal operations, refresh will take place without affecting it. If the 68000 is occupying the bus, refresh will be granted at the completion of the buss cycle. If a refresh is in progress when the 68000 comes to require the bus, the processor will wait until the completion of the refresh cycle.

CPU DMA Interface

The two channels of the DMA interface to the CPU buss appear as two 64K slices of the 68000's 16M memory space. Accesses in the range 40000H to 4FFFFH steal P2 cycles, and select the P2 DMA memory map. Accesses in the range 50000H-5FFFFH steal P1 cycles, and select the P1 DMA memory map. Appropriate initialization of the Q256 memory maps thus allow the 68000 to access any physical area of system memory or peripherals whether or not they are mapped into the CPUs' logical spaces. Refer to Q256 documentation for details about system memory management.

Data size mismatch between the 68000 and the CPU buss is handled by the hardware. A 16-bit access by the 68000 is handled by two separate 8-bit DMA transfers across the CPU buss and the 68000 receives one Data Transfer Acknowledge (DTACK) when both transfers have been completed. Which byte goes first is indeterminate; in any case each byte is always transferred to or from the right place in CPU space.

Access by various devices to the CPU buss is arbitrated by two daisy chains, one for each 6809. Higher priority devices may prevent access by the WP to the CPU buss indefinitely. Refer to a separate document which describes the daisy chain allocation for Series III.

Waveform Buss Interface

Any 68000 access above 200000H goes through this interface which provides all the arbitration control for the waveform buss and matches the asychronous internal 68000 buss to the 3.3Mhz synchronous waveform buss on the backplane.

The three types of device which can access the waveform buss are prioritised in the following order:

1. Channel card
2. Refresh
3. Waveform Processor

While a round-robin allocation system allows each channel only one particular 300nS time slice out of every 8, the WP can use any time slice not required by its channel or refresh. Refresh is lower priority than channel access on the basis that if the channels are running fast enough to block out refresh for a considerable period, the WRAM will be self-refreshing anyway. Normal refresh is requested by the same pulse as the Private RAM refresh on average once every 16uS but the actual refresh operation of WRAM is performed independently of PRAM refresh. Unless the WP wants to access WRAM at the same time, WRAM refresh takes place without affecting the 68000 at all.

The function of 8-bit mode waveform access is fully explained in the WRAM documentation. When the 16-bit mode is selected, WRAM appears as just normal memory which supports byte and word accesses by the 68000. Code can be executed in WRAM by the 68000 provided 16-bit mode is selected. Mode selection (for the WP only) is achieved via the WP control latch, bit 5. 8-bit mode is selected by setting this bit low. Either the CPU can set this bit directly or the WP can do it via the CPU DMA interface at 4FC5CH or 5FC5CH. The control latch bit does not affect the mode in which the channel cards access WRAM. Refer to channel card documentation for further details.

ADC Serial Interface

Communication with the CMI-337 ADC board is achieved with an optically isolated serial IO link. Signals in this link consist of a Start Conversion and Control Data outputs and Clock, End of Conversion (EOC) and Sample Data inputs.

The Clock, generated by the ADC board, clocks each bit of Sample Data in and Control Data out.

The Start Conversion signal must be generated by running Channel Card 0 at the required sampling frequency. A signal from that channel goes to the WP and is output as the Start Conversion command.

Miscellaneous Items

The indivisible Read-Modify-Write instruction TAS is not supported by any of the memory interfaces on the WP. The address strobe signal AS is used as the buss cycle terminator and as this is not negated in between the read and write cycles of the RMW instruction, the 68000 would hang.

A no-wait state memory access is executed by the 68000 in 8 states, or 400nS at 10MHz.

Processor, ROMs, and Decoding

(refer schematic CMI-33-00)

The 10MHz processor clock is generated on the Channel Support Card (CMI-32) and driven along the motherboard as two differential lines. These are received by two Am26LS32 differential line receivers, one with its inputs reversed to generate the processor clock (PCLK) and its inverse $\overline{PCLK}$ without any skew between the complementary signals. Series 22R resistors prevent excessive undershoot and ringing on the PCLK signals

Detailed description of the operation of the 68000 can be obtained from the Motorola literature but the following is a very brief indication of how 68000 buss cycles proceed. A cycle is initiated by driving the address onto the 23 address lines A1 to A23 following the falling edge of the clock and asserting the address strobe $\overline{AS}$ 50nS later, by which time the address lines should be stable. There is no A0 address line signal. A1 to A23 determine which 16-bit word in memory is to be accessed. Whether the high byte or the low byte or both is to be accessed is determined by the Upper and Lower Data Strobes $\overline{UDS}$ and $\overline{LDS}$, so the least significant address bit A0 is implied by these two data strobes. With a 24 bit effective address width, the 68000 can access 16 megabytes of memory.

In a read cycle, $\overline{UDS}$ and/or $\overline{LDS}$ are asserted at the same time as the address strobe. For a write cycle, the processor puts its data on the buss 50nS after $\overline{AS}$ and asserts the data strobe(s) 50nS later still, by which time the data should be stable.

Nothing happens now until the accessed device returns a Data Transfer Acknowledge ($\overline{DTACK}$) signal. The processor samples $\overline{DTACK}$ and recognises that it has been asserted on the falling edge of the clock. One clock cycle later, on the next falling edge, the buss cycle is terminated. In the read case, the data is latched into the processor, and address and data strobes negated. In the write case, the strobes are negated then the data buss tri-stated 50nS later. Each half clock cycle which the processor spends waiting for the return of $\overline{DTACK}$ is called a wait state.

The upper seven address lines are used in the decoding circuitry which divides the address space as shown on the memory map diagram. All decoder outputs depend upon AS being asserted so the entire address is stable whenever any output is active. Any address above 200000H (hex) will have one of the top 3 address signals high so will generate WMEM to access Waveform Memory. Within the 2M bytes below 200000H, decoding is performed by the S139 at D10. The top 1M is unused. The next 512K bytes down is occupied by the private RAM, accessed by the DAS signal. The bottom 512K bytes is further divided into four, occupied by the ADC serial interface (/IO), the CPU DMA interface (/CMI), a spare slot, and the ROMs (/EPROM) at the bottom.

The ROMs must reside at the bottom of the memory space because after reset the 68000 boots up by fetching its supervisor stack vector and restart program counter from the locations 0 and 4 respectively (in contrast to the 6809 which restarts from the top of memory). 450nS ROMs are currently used requiring the insertion of 6 wait states (each wait state is 50nS long) into each ROM access cycle. The counting of the wait states is performed by the LS161 counter at C13. While EPROM is not asserted, the counter is held preset at 13 but when EPROM comes active, the counter is released while the ROM(s) are enabled. On the second rising edge of the clock after this, the counter reaches 15 and generates the ripple carry out which clocks the flip-flop at C14 to assert DTACK1. When the processor terminates the cycle, AS is negated so BAS goes low and clears the flip flop to remove DTACK. EPROM is negated so the LS161 is put back into preset ready for the next ROM cycle. The delay of EPROM through the decoding circuitry plus the two counted clock cycles, plus the delay between DTACK and the processor latching the data in provides the required 450nS access time. The complete cycle takes 700nS from buss inactive to buss inactive again.

If 27256 or other 32K x 8bit ROMs are used the W1 link must be set to position "1".

Private RAM

(refer schematics CMI-33-01,02 and timing diagram)

The 68000's Private RAM consists of 256K by 16 bits of dynamic MOS memory which requires regular refreshing of the contents. Within each RAM IC memory is divided two dimensionally into rows and columns. Which row is accessed is determined by the 9-bit row address, clocked into the RAM chips by the falling edge of /RAS. The column is selected by the column address, clocked by the falling edge of /CAS. In this manner the 18 bit address required to access 256K is multiplexed onto 9 address pins of each RAM.

To consider a RAM access cycle, assume for a moment that refresh is inactive, i.e. /BG, and /BGACK are negated. The RAM access cycle begins when the /DAS (Dynamic RAM Address Strobe) signal is asserted by the decoding. At this time the three flip-flops F8 and E8 at the top of drwg. CMI-33-01 will be set. This means that RA will be asserted so the Row Address (A1 to A8 and A18) will be driven onto the RAM address lines DA0 - DA8 via LS244 buffer B4 and part of

LS125 B5. The next rising edge of PCLK will clock the first F8 flip-flop and thus assert RAS. 50nS later, which is sufficient row address hold time, the second flip-flop is clocked by rising edge of PCLK so RA is negated and CA is asserted to switch the address multiplexer over to the column address A9 - A17 via LS244 B3 and the other LS125 gate. Another 50nS later and the rising of PCLK again clocks the third flip-flop to generate DTACK2 while the other side of the flip-flop goes through the AND gate G9 and ALS00 gates at E9 to assert UCAS or LCAS or both. Having two seperate CAS signals allows the RAM to be accessed a byte at a time or a word at a time. The 100nS between the recognition of DTACK by the processor and the time it terminates the buss cycle guarantees that read data from RAM will be valid, since the RAMs access time is 75nS from the falling edge of CAS. This is also ample data in hold time for write cycles. The processor terminates the cycle by negating the address and data strobes. The negated AS sets the three flip-flops again ready for the next access.

The above timing means that two wait states are inserted into PRAM cycles, which take 500nS altogether.

Data in and out of the RAM is controlled by the pair of LS245 bi-directional buffers D5 and E5 which are enabled by the DAS signal and the direction is determined by the processor R/W line.

Refresh is performed a row at a time by placing an 8-bit refresh count on the address lines (which nominates the particular row to be refreshed) and driving RAS low for a specified period. CAS is not asserted at all so data in is ignored and data out remains in the high impedance state. Refer to RAM data for further information on the internal aspects of refresh.

The refresh operation is achieved using the 68000 buss arbitration protocol to force the processor off the buss while the hardware performs the refresh operation. The three processor signals which provide this protocol are as follows:-

Bus Request (/BR): asynchronous input to the 68000 which tells it that an alternate buss controller (in this case the refresh hardware) wishes to use the bus. ("I want the bus").

Buss Grant (/BG): output from the 68000 indicating that BR has been recognised and it will relinquish the buss at the end of the current cycle if it is not available already. ("You can have the bus").

Buss Grant Acknowledge (/BGACK): Input to the 68000 to indicate control of the buss has been taken by the alternate buss controller (refresh). Once BG has been asserted and the current cycle completed (indicated by AS negated), the 68000 will wait indefinitely with all its buss outputs tri-state until this signal is asserted and negated again. ("I've got the bus").

Refresh request pulses arrive approximately every 16uS on pin 47B from the channel support card. Each pulse sets the LS74 G14 to generate the buss request to the 68000. As soon as the processor recognizes BR has been asserted it asserts Buss Grant (/BG). The buss becomes available when BG is asserted and the address strobe (/AS) is negated. Shortly after it is negated AS actually goes tri-state but its pull-up resistor maintains the high state.

When $\overline{AS}$ is high and $\overline{BG}$ is low the S-R flip flop formed by NANDS F9 is set to assert the acknowledge signal $\overline{BGACK}$. The refresh hardware has control of the buss for as long as $\overline{BGACK}$ remains low. The row/column address drivers are inhibited by NANDS at E9 and the LS461 refresh address counter/driver is enabled onto the address buss instead. The complement of $\overline{BGACK}$ is input to the LS175 which is wired up as a shift register clocked by the 10MHz inverted processor clock. This simply provides a 2 clock cycle delay. On the first rising edge of $\overline{PCLK}$ after $\overline{BGACK}$ is asserted, a high is shifted into Q1. On the third rising edge the pulse is shifted into Q3 and the corresponding low on $\overline{Q3}$ resets the S-R flip flop to negate $\overline{BGACK}$. The low on BGACK (F9-6) then clears all stages of the LS175 readv for the next refresh cycle. The rising edge of $\overline{BGACK}$ clocks the LS461 to increment the refresh counter at the same time as the drivers are inhibited.

The LS175 also performs one other function. The high shifted into Q1 corresponds to a low in $\overline{Q1}$. This results in a low input to LS74 flip flop F8-12 so that on the rising edge of PCLK, after the refresh count has been enabled for 50nS (less propagation delays), $\overline{RAS}$ is sent out to all the RAM chips. $\overline{RAS}$ is negated at the end of the refresh interval when the LS74 is set by the high on $\overline{BGACK}$.

The RA/CA and $\overline{DTACK2}$ flip flops are both clocked as if a normal memory cycle was occurring but this is not significant to the refresh operation. $\overline{UCAS}$ and $\overline{LCAS}$ are prevented from being asserted by the $\overline{BGACK}$ input to AND gate G9 being low.

The buss request is removed from LS74 G14 at the beginning of the refresh cycle by the low on $\overline{BGACK}$ resetting this flip flop. The result of the above timing is that refresh cycles take 350 nS. As mentioned above, if refresh request happens to occur when the 68000 is not using the bus, it will take place without affecting 68000 execution. Requests are ignored during refresh cycles and while the processor is held reset. If the processor is halted, buss arbitration continues normally (although obviously the buss is permanently available) so that refresh occurs as required.

Interrupt and Reset Control
(refer schematic CMI-33-03)

The 68000 can be reset, halted and interrupted via a 6-bit write only latch at FC5CH in the CPU address space. This address is decoded by the large NAND gate B1. The decoder output is latched by flip-flop A5 on the rising of BRA and data is written into the LS174 latch B6 at the end of the system CAS pulse. LS240 A6 permanently buffers all incoming data to the latch.

The bit assignments are as follows:

Bit	Function	Active
0-2	Interrupt level input to the 68K	LO
3	Halt	LO
4	Reset	LO
5	Waveform 8-bit mode select	LO

The interrupt bits form the priority encoded interrupt level and inputs directly to the processor. Level 7 (all bits low) is the highest priority while level 0 (all bits high) means no interrupt. Interrupts are cleared by the 68000 writing to the control latch itself via the CPU buss (see Sec 1.4).
To reset the processor, both $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ must be asserted. Note that refresh of on-board RAM is prevented while the processor is held reset. The minimum reset period required by the 68000 is 10 clock cycles i.e. 1uS.

If only HALT is driven low, the processor will halt but refresh operates normally.

The waveform 8-bit mode select $\overline{\text{W8BIT}}$ will be explained along with the waveform interface. The state of the $\overline{\text{HALT}}$ and $\overline{\text{RESET}}$ signals are indicated by the LEDS on the front of the Waveform Processor card. Both off means that the processor is runnng.

System Reset (SYRES) clears the latch, thus putting the 68000 into reset. The level 7 interrupt also generated will be ignored while the processor is reset. A manual control panel may be plugged into the WP board with a HALT/RUN toggle switch and RESET momentary switch.

The 68000 has a very powerful interrupt vectoring system which permits the interrupt service routine vector to be provided by the interrupting device. Many such devices may therefore generate interrupts without the need for extensive polling procedures to find out which device is requesting. This facility is not required on the Waveform Processor and the only interrupts used are the seven Auto-vector interrupts determined by the interrupt level on $\overline{\text{IPL0}}$-$\overline{\text{IPL2}}$. The function code outputs FC0-FC2 are all high during an interrupt vector fetch and this is detected by the LS10 NAND D12, on drwg. CMI-33-00. This results in $\overline{\text{VPA}}$ being asserted which tells the processor to use the autovector instead of an interrupting device-supplied one.

68000/6809 DMA Buss Interface
(refer schematic CMI-33-04 and timing diagram)

Communication between the 68000 processor and the 6809 CPU is achieved by DMA (Direct Memory Access) on the system bus.

DMA is initiated by the 68000 when it accesses any address in the range 040000H to 05FFFFH. These addresses are decoded by the S139s on drawing CMI-33-00 and result in the $\overline{\text{CMI}}$ signal being asserted (low). Since the rest of the interface circuitry is not activated yet, PACK (to be explained later) will be low and a low will be presented at the data input of flip flop F11 whose function is to synchronise the transfer with the CMI bus.

Address line A16 is used to select which 6809 processor's buss cycle(s) are to be used for the transfer. The timing signals for both processors are input to LS241 buffer A8 which is wired as a muliplexer. If A16 is low, P2ø2 is enabled through to become Pø2, ADD2 becomes PADD and so on. If A16 is high, P1's timing signals are enabled instead. By this means, the address range specified above is split in two: from 040000H to 04FFFFH the transfer automatically occurs on P2 buss cycles, while from 050000H to 05FFFFH it occurs on P1 cycles. Refer to the 6809 CPU documentation for more information on the interleaved P1/P2 CMI buss cycles.

Thus at the beginning of the data cycle of whichever processor is selected, the Pø2 signal clocks the LS74, recording the fact that a DMA cycle is required.

All DMA devices are interconnected on the motherboard in a "daisy chain". Each device is assigned a given priority in the chain and must wait until no higher priority device is already using the bus. The 6809 CPU is the always the last device in the chain . There are two separate daisy chains in the CMI system, one for each 6809 CPU. Since the 68000 can perform DMA on either CPU's cycles, it is a member of both chains. $\overline{ETL1}$, $\overline{ENL1}$ and $\overline{RDMA1}$ are the chain signals for P1, $\overline{ETL2}$, $\overline{ENL2}$, $\overline{RDMA2}$ are for P2. Which set are used is again selected by the state of A16 at the time of transfer.

The selected $\overline{ETL}$ (Enable This Level) signal is low when no higher priority device is occupying the bus. After the $\overline{CMI}$ signal has been latched, nothing happens until this signal is low, whereupon the $\overline{RDMA}$ (Request DMA) is driven low through the selected LS12 gate. Any DMA device pulls this open collector line low to to request buss access to the CPU. At the same time, the selected $\overline{ENL}$ (Enable Next Level) signal is inhibited. Normally, the low on $\overline{ETL}$ comes in and goes out again on $\overline{ENL}$ to indicate to lower priority devices that the buss is available but when the 68000 requires a transfer $\overline{ENL}$ is held high to stop the lower devices accessing the bus.

The CPU acknowledges that it will hang and release the buss for the next cycle by asserting ACK1 or ACK2; the selected ACK signal becomes PACK. When a request has been generated (F11 $\overline{Q}$ hi) and this level is enabled (/ENL lo), the rising edge of PACK clocks a low into flip flop G13 to generate $\overline{DCYCLE}$. This signal indicates that the next buss cycle is definitely going to be a 68000 DMA transfer and remains asserted until the end of the address phase of the actual DMA cycle.

The other half of G13 is also clocked by PACK to generate the P1 or P2 $\overline{DMAC}$ (DMA Claim) signal as selected by A16. This signal goes to the Q256 RAM card to select the memory mapping which has been set up specifically for the 68000. In this way the 68000 may have access to part or all of the same physical memory space as the 6809 CPU or it may have access to an entirely different part of physical memory as required by software. The $\overline{DMAC}$ signal is asserted during the data cycle preceding the actual transfer.

The address phase of the DMA cycle is indicated when $\overline{ATB}$ (Address To Buss) is asserted by the LS10 F13. At this time the lower 15 bits of the 68000 address buss are enabled on to the CMI buss through the two LS244s A3 and A4 to select the required location within the 6809 address space. VMA is driven high through LS125 B5 to indicate a Valid Memory Address and the 68000 R/W line is driven through the same buffer to indicate a read or write cycle. When the 68000 performs 8-bit memory accesses, the $\overline{UDS}$ and $\overline{LDS}$ signals (upper and lower address strobes) indicate whether an even or odd address is being accessed. The sense of these signals are clocked into JK flip flop D13 at the beginning of $\overline{DCYCLE}$ to generate HIBYTE and LOBYTE. The latter signal becomes the least significant address line driven onto MA0 through A3.

In the case of 16-bit accesses, the hardware automatically requests two successive DMA accesses across the 8-bit CMI bus. Both $\overline{UDS}$ and $\overline{LDS}$ are asserted so that the JK outputs HIBYTE and LOBYTE simply

toggle on each access. It does not matter which byte transfers first and in fact this depends on the initial state of D13. LOBYTE directs the data to or from the odd or even address and both signals control whether the higher or lower 8 data lines are directed to the data bus.

The data buss interface consists of Schmitt bidirectional buss transceiver LS640 B7 and bidirectional driver/latches B8 and B9 (LS646s). The data phase of the DMA transfer is indicated by the assertion of $\overline{\text{DTB}}$ (Data To Buss) at the rising edge of BRA when a DMA cycle is in progress. This is performed by flip flop G14. $\overline{\text{DTB}}$ enables the buss transceiver B7 and the direction is determined by the 68000 R/W signal.

If the 68000 is writing to the CMI bus, B8 or B9 simply act as buffers to transfer the high or low 68000 data signals (PD0-15) through to B7. HIBYTE or LOBYTE plus $\overline{\text{CMI}}$ being asserted will drive the $\overline{\text{G}}$ input of the appropriate LS646 for the duration of the DMA cycle (LS02 and LS32 gates E14 and E15).

When the 68000 reads from the CMI bus, B8 or B9 must latch the data in from the buss to hold it until the 68000 terminates its own cycle and latches the data internally, about 50nS after the end of the DMA cycle. 100nS before the end of the data phase, the CMI timing signal CAS goes low, resulting in a rising edge on $\overline{\text{BCAS}}$. Data from memory is guaranteed to be valid at this time. The LS11 gate L5 generates the LDATA (Latch Data) signal which is ANDed with either HIBYTE or LOBYTE to latch the data coming into the A side of B8 or B9. The output of the latch (B side of the selected LS646) is driven onto the PD lines until the 68000 completes its cycle and negates $\overline{\text{CMI}}$.

Termination of the transfer after single or double DMA cycles is controlled by the two flip flops in LS74 E11. In the single (8-bit) transfer case, either UDS or LDS will be low. This will cause the LS10 M5 to output a high, and $\overline{\text{DTACK4}}$ will be generated as soon as LDATA occurs. The 68000 will then terminate its cycle immediately, after only one DMA cycle.

In the double DMA cycle (16-bit) case, both UDS and LDS are high so $\overline{\text{DTACK4}}$ will not be generated until the first flip flop in E11 is set. Initially this flip flop is reset. At the first LDATA pulse a high is clocked in but $\overline{\text{DTACK4}}$ is not generated because of the propagation delay through to the next flip flop. Since $\overline{\text{DTACK4}}$ is not asserted, the 68000 still waits with address and address/data strobes asserted. If writing, the data remains asserted by the 68000 but both address and data are removed from the CMI buss when $\overline{\text{ATB}}$ and $\overline{\text{DTB}}$ are negated respectively. If reading, the first byte read in is latched and held by C6 or C7. Since $\overline{\text{CMI}}$ will still be asserted and PACK will have been negated, the whole process of waiting for daisy chain priority and DMA requesting begins again in order to perform a second DMA cycle. The second cycle can be held up indefinitely by higher priority devices using the buss after the first cycle. When the second LDATA edge comes along the high on the LS10 output is clocked into the second E11 flip flop and /DTACK4 is asserted.

On the next falling edge of PCLK, the 68000 recognises that /DTACK has been asserted. On the second falling edge of PCLK the data is latched internally for a read, and the address and strobes are released. The low on BAS resets the flip flops at E11.

Timing Summary

The maximum DMA data rate permitted by the CPU is 500kHz since each 6809 accesses the buss at 1MHz and only every second cycle is permitted DMA.

The best case for a single byte transfer across the CPU interface is 1.8uS assuming the decoding circuitry (driven by the 68000 address buss and address strobe) activates the DMA hardware just as a DMA enable pulse (ENL) arrives from the required DMA daisy chain (there is a separate daisy chain for each 6809 processor). Each higher priority device on the daisy chain which prevents the WP from gaining the CPU buss will add 2uS to the access time. Since even without other devices using the buss synchronization with the 500kHz ENL signal is necessary, the worst case for a one byte transfer not counting the daisy chain is just under 3.8uS.

Similarly double byte accesses will take at best 3.8uS, and at worst just under 5.8uS, not counting other DMA devices crashing in. If a higher priority device takes the buss in between the two bytes, the hardware waits for the next available cycle to transfer the second byte.

Debugging Notes

If the timing circuitry of the DMA interface is faulty, the most likely result is that $\overline{\text{DTACK4}}$ will never be generated and the 68000 will simply hang which makes debugging easy. In this case, check first that the address decoding is generating $\overline{\text{CMI}}$, then that the daisy chain signals are present. Then look for an 800nS pulse on $\overline{\text{DCYCLE}}$, indicating that DMA cycles are actually ocurring. Continue through to the $\overline{\text{ATB}}$, $\overline{\text{DTB}}$ and $\overline{\text{LDATA}}$ signals, checking not only that they are generated but also that they get to their respective destinations in the circuitry.

If the DMA cycles are being synchronised and timed correctly check that the address buffers and data buffer/latches are being enabled and clocked at the correct times.

If all timing circuitry is correct, the last possibilty is data or address buss shorts, open circuits or faulty drivers. Special test ROMs are available which cause the 68000 to repetitively copy bytes and words from one location to another in CMI memory. The 6809 monitor can then be used to deduce which data or addresses cause problems.

Waveform Buss Arbitration and Interface

(refer schematic CMI-33-05 and timing diagram)

The most complex part of this circuitry is the arbitration logic which is contained within the PAL16R4 programmable logic array. The function of this logic, along with the round-robin allocation circuitry on each channel card, is to determine which of the channels, waveform processor, or refresh is to be allowed access to the waveform bus.

The arbitration decision is made on the falling edge of SCLK, the 3.3MHz clock which synchronizes the waveform bus. Thus it is $\overline{\text{SCLK}}$ which clocks the latched outputs of the PAL. Feedback is provided in the PAL so that the new outputs can be any combinations of the inputs and the old outputs. The inputs are as follows:

/SCLK	is used as data as well as a clock.
PCLK	the 10MHz processor clock
RFSH	refresh request
/TSTAKEN	"Time Slice Taken" means the channel card allocated to the next 300nS cycle actually wishes to use it. All channels have an open collector output connected to this signal on the buss. Timing is critical, and the termination network on the WP input holds the line at 2.5V from which a channel can rapidly pull it down.
WMEM	is the decoder output indicating 68000 access to the Waveform Buss.
UDS and LDS	are the 68000 data strobes. pin 9 of the PAL is a feedback signal /WDTB indicating the data phase of a waveform buss access is in progress.

The PAL generates outputs as follows:

/WAS	is asserted if either a channel wants the buss or the waveform processor wants it.
/WUDS	is asserted if a channel is accessing or if the WP is accessing the upper byte.
/WLDS	is asserted if a channel is accessing or if the WP is.
/RPEND	(Refresh Pending) is asserted if RFSH is asserted but the last cycle was not a refresh cycle, or if refresh was already pending but the last cycle was not refresh. The latter happens if a channel preempted
/	the refresh cycle before.
/WREF	(Waveform Refresh) is generated if /RPEND is asserted and /TSTAKEN is not (refresh pending and channel doesn't require bus).
/WATBL	is asserted if the 68000 is to have the cycle.
pin 12	is the same as /WATBL except unlatched, so it can be clocked into C15 by the falling edge of SCLK.
pin 13	is a short pulse (combination of /SCLK and PCLK) near the end of the 68000 access cycle which could be used to set C15 and hence shorten the address assertion time but this has not proven to be necessary.

Once the arbitration is achieved, the actual timing of WP accesses to the Waveform Buss is fairly simple. A 100nS-delayed and inverted SCLK signal is produced by flip-flop A5 by clocking it with PCLK. Although SCLK is generated on the Channel Support Card by PCLK their delays through the buss and buffers result in some uncertainty as to which transition will occur first at the receiving end. Delaying SCLK through two LS244 buffers ensures that PCLK will be slightly ahead of SCLK and thus produce the correct 100nS delay.

The decoding can generate WMEM any time before the falling edge of SCLK. If the WP is to have the cycle, the address will be driven onto the buss from that edge to the same edge of the next cycle, while WATB (Waveform Address to Buss) is low (300nS). The data buss is enabled for 300nSf rom the rising of DSCLK, which is dead in the middle of the SCLK low period. Thus the data cycle is skewed from the address cycle by 100nS. The direction of data flow through the bi-directional buffers A9 and A10 is determined by the R/W line. WDTB is fed back to the PAL because at the beginning of the next address cycle WMEM will still be asserted, although a second WP access is not desired. WDTB indicates that the last cycle was a WP access, so the PAL ignores WMEM.

DTACK3 is generated at the rising edge of SCLK which is also on a rising edge of PCLK, so approx 150 nS later (i.e. on the second falling edge of PCLK) the 68000 cycle will be terminated and data, if a read cycle, latched in. This coincides with the time that data from the Waveform RAM is valid.

The Waveform Buss write line WR/W is driven at the same time as the address and is open collector to allow future devices to write to the WRAM during unused channel time slices.

The Waveform Processor may read the WRAM in 8-bit mode if it wishes (this is not the same as reading or writing bytes according to UDS or LDS - see WRAM documentation) and this is determined by the W8BIT mode signal from the control latch. There is no need to do this except as a means of testing the WRAM cards. The 8BIT signal goes out to the buss as a normal address line.

Timing Summary

Best case waveform accesses will take 500nS if the decoding circuitry requests the buss just as the edge which synchronises the 68000 to the waveform buss occurs, and no other device wants the bus. This implies 2 wait states. Worst case not counting other devices on the buss is just under 800nS. Each higher priority device which preempts the WP will cause a further 300nS delay.

ADC I/O Interface

(refer schematic CMI-33-06 and timing diagram)

The 68000 connects to the CMI-337 Stereo Analogue to Digital Converter (ADC) via a high speed optically isolated serial link along a ten-way cable. The sample rate signal comes in from one of the channel cards as ADCLK and is used sent out as the Start Conversion command. Early versions of the ADC required a 5uS pulse which was generated by half of the LS123 one-shot G12. This is normally bypassed by link LK1. The ADC responds by generating 16 clock pulses, each of which clocks in one bit of the converter data followed by an End of Conversion (/EOC) flag. At the same time as converter data is clocked in, control data is clocked out from the WP. This activity happens continuously while ADCLK is present and requires no intervention from the 68000 except to actually read the data coming in if it wishes.

Sample Data is input to LS595 shift registers with output latches D6 and E6 which can be read by the 68000. Data is clocked from the shift registers to the latches at the beginning of the next conversion, signified by a rising edge on the $\overline{EOC}$ signal. If a 68000 read of Sample Data has been initiated, this edge also generates the DTACK signal to terminate the cycle. In this manner no status checking by the processor is required; successive reads will always get successive data. To put it another way, a 68000 read of Sample Data will hang until the next $\overline{EOC}$ edge. Thus an ADC acquisition loop simply has to be shorter than the sample period, and any spare time is killed by the hang-up mechanism.

A hardware timeout is provided by one-shot G12 in case the ADC is not working or the cable is disconnected since without the $\overline{EOC}$ from the ADC the 68000 would hang for ever. If there is no $\overline{EOC}$ within 1.5mS of the beginning of the read, a buss error exception (BERR) is generated and the 68000 will divert to the appropriate vector in ROM. This is the only source of buss error exception on the WP card. The 1.5mS timeout implies a lower sampling frequency limit of around 670Hz.

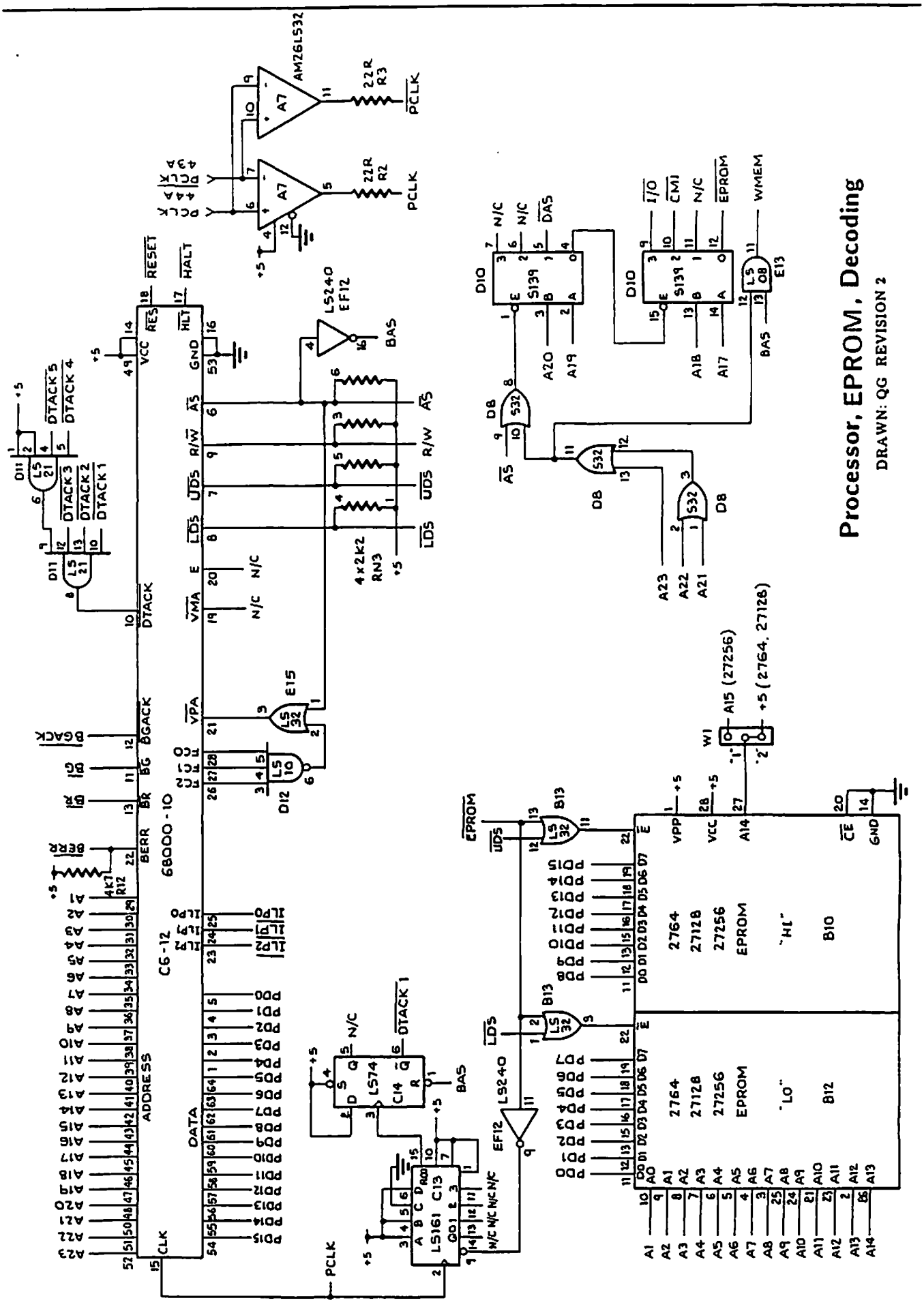
Control data is written from the 68000 to parallel inputs of LS597s D7 and E7, latched on the falling edge of either or both data strobes. The shift register is loaded with the contents of the latch when $\overline{EOC}$ goes low at the end of the conversion Termination of the write cycle is the same as for sample reads. The function of the control data is described in CMI-337 documentation.

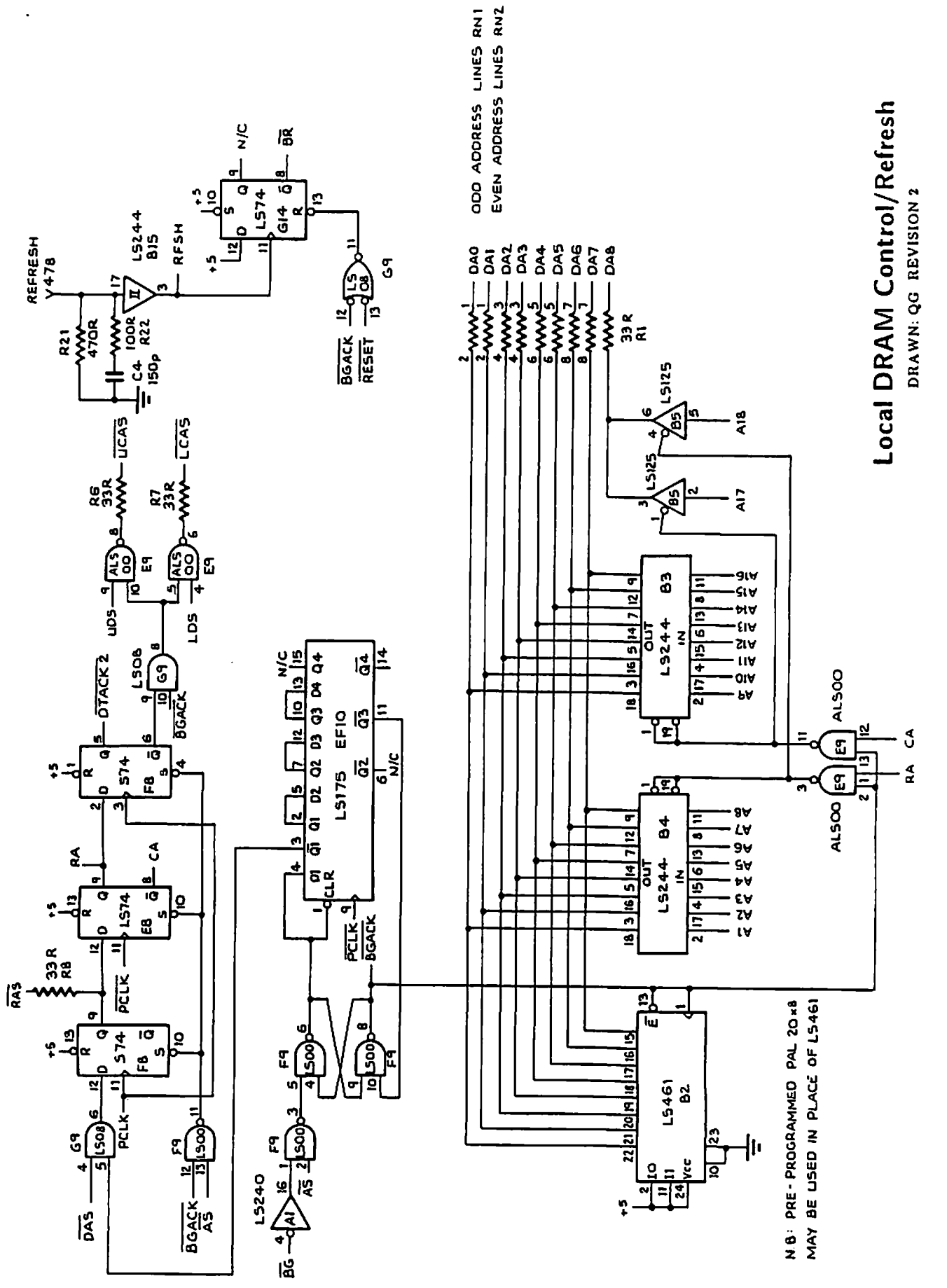
Any address in the range 60000H to 7FFFFH may be used for reading Sample Data or writing Control Data.

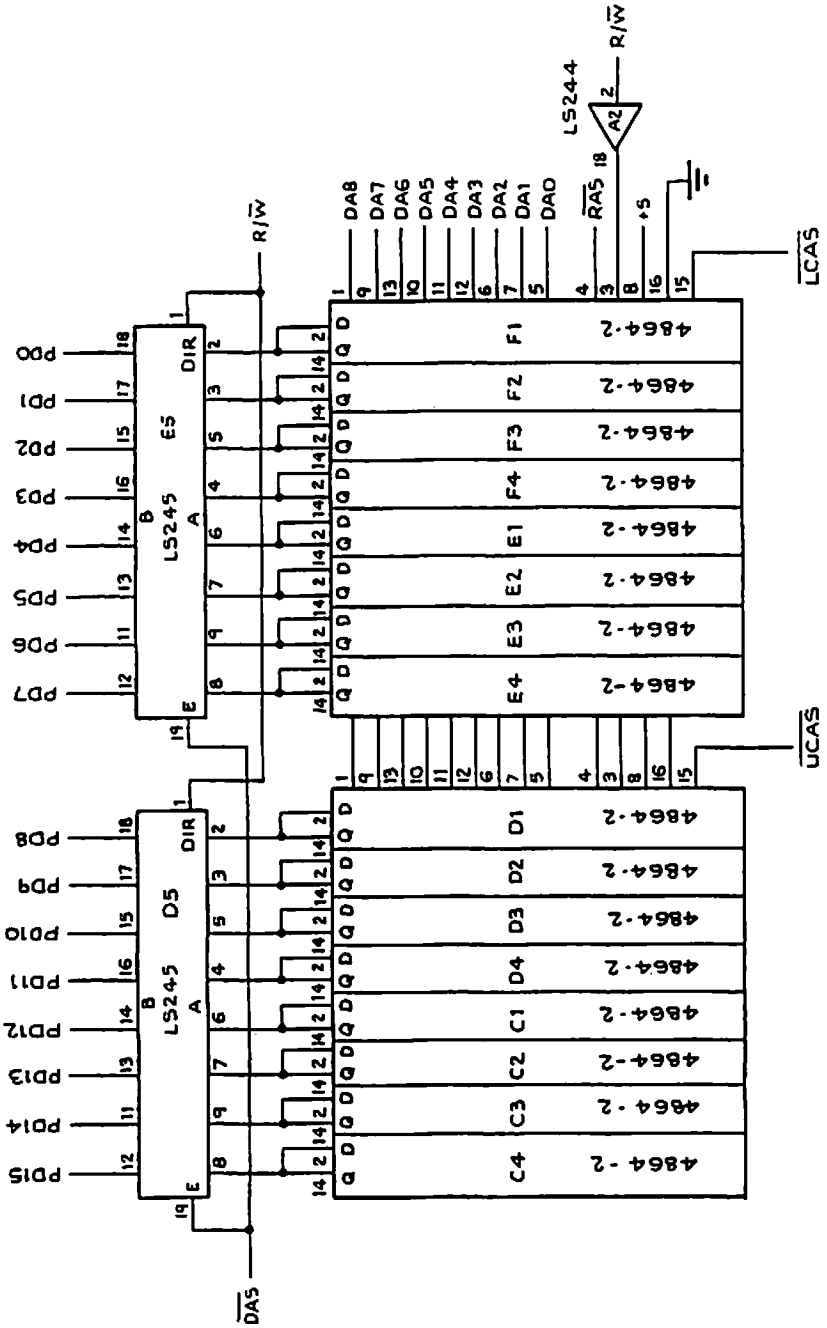
Timing Summary

For an ADC I/O transfer to occur without wait states, it must commence more than 100nS and less than 250nS before the EOC edge.

Note that ADC accesses "hang" the processor, with the buss active for up to 22.6uS (assuming a 44kHz mono sample rate) waiting for the next $\overline{EOC}$ edge. Since refresh requests arrive every 16uS, this means that the next cycle after the completion of an ADC access will always be a refresh cycle. Further, if a refresh cycle has been delayed by the ADC access so that a second request arrives while the first cycle is actually in progress (/BGACK asserted), or before it even starts, the second request will be missed. The number of missed accesses increases with longer ADC accesses so it is possible for slow sample rates to result in the RAM not being refreshed according to spec. Fortunately there is normally a considerable degree of safety margin on the 4mS refresh 256-cycle (or 2mS 128-cycle) requirement specified by most RAM manufacturers. Nevertheless, slow sample rates should be used with caution.


Processor, EPROM, Decoding
DRAWN: QG REVISION 2
68000-10
C6-12
ADDRESS
DATA
AM26LS32
LS240
EF12
LS74
LS161
LS32
LS10
LS21
LS08
S139
S32
2764
27128
27256
EPROM
RESET
HALT
BERR
BGACK
BG
BR
AS
R/W
UDS
LDS
VMA
VPA
DTACK
DTACK5
DTACK4
DTACK3
DTACK2
DTACK1
PCLK
BAS
4x2k2
RN3
4K7
R12
22R
R2
R3
A15 (27256)
+5 (2764, 27128)
WMEM
EPROM
CMI
I/O
N/C
A23
A22
A21
A20
A19
A18
A17
A14
A15
A16
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
D0
D1
D2
D3
D4
D5
D6
D7
"HI"
"LO"
B10
B12
B13
D10
D11
D12
D8
E13
E15


Local DRAM Control/Refresh
DRAWN: QG REVISION 2
ODD ADDRESS LINES RN1
EVEN ADDRESS LINES RN2
REFRESH
LS244 B15
RFSH
LS74 G14
N/C
BR
R21 470R
R22 100R
C4 150p
BGACK
RESET
LS 08 G9
DA0
DA1
DA2
DA3
DA4
DA5
DA6
DA7
DA8
33 R R1
LS125
A18
A17
UCAS
LCAS
R6 33R
R7 33R
ALS 00 E9
UDS
LDS
LS08 G9
DTACK 2
BGACK
S74 F8
RA
CA
LS74 E8
33 R R8
RAS
PCLK
LS175 EF10
N/C
CLR
LS244 B3
LS244 B4
OUT
IN
A16
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
ALS00 E9
LS00 F9
LS240
AS
DAS
BG
LS461 B2
Vcc
NB: PRE-PROGRAMMED PAL 20x8
MAY BE USED IN PLACE OF LS461



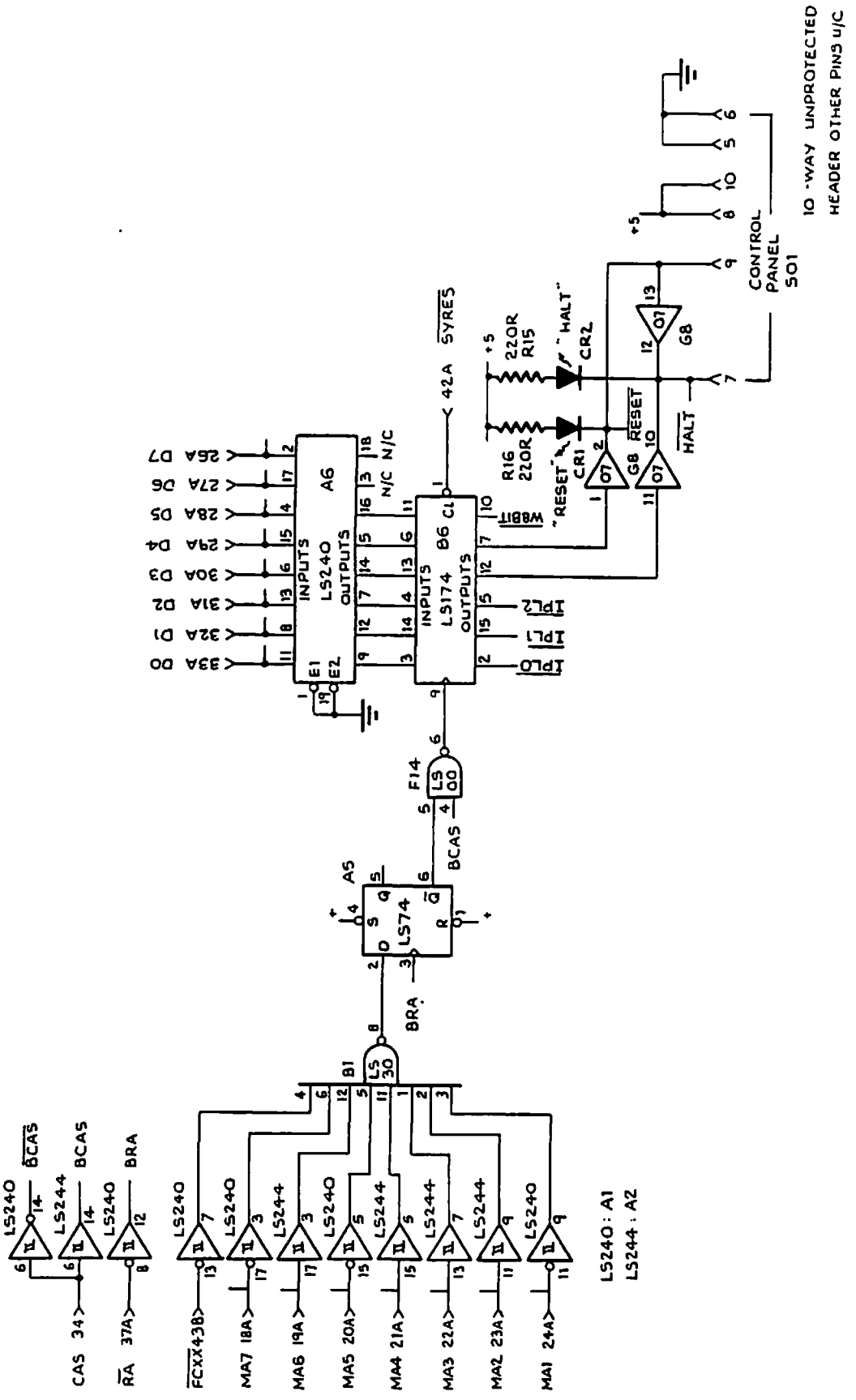
Local DRAM 64k/256k Words

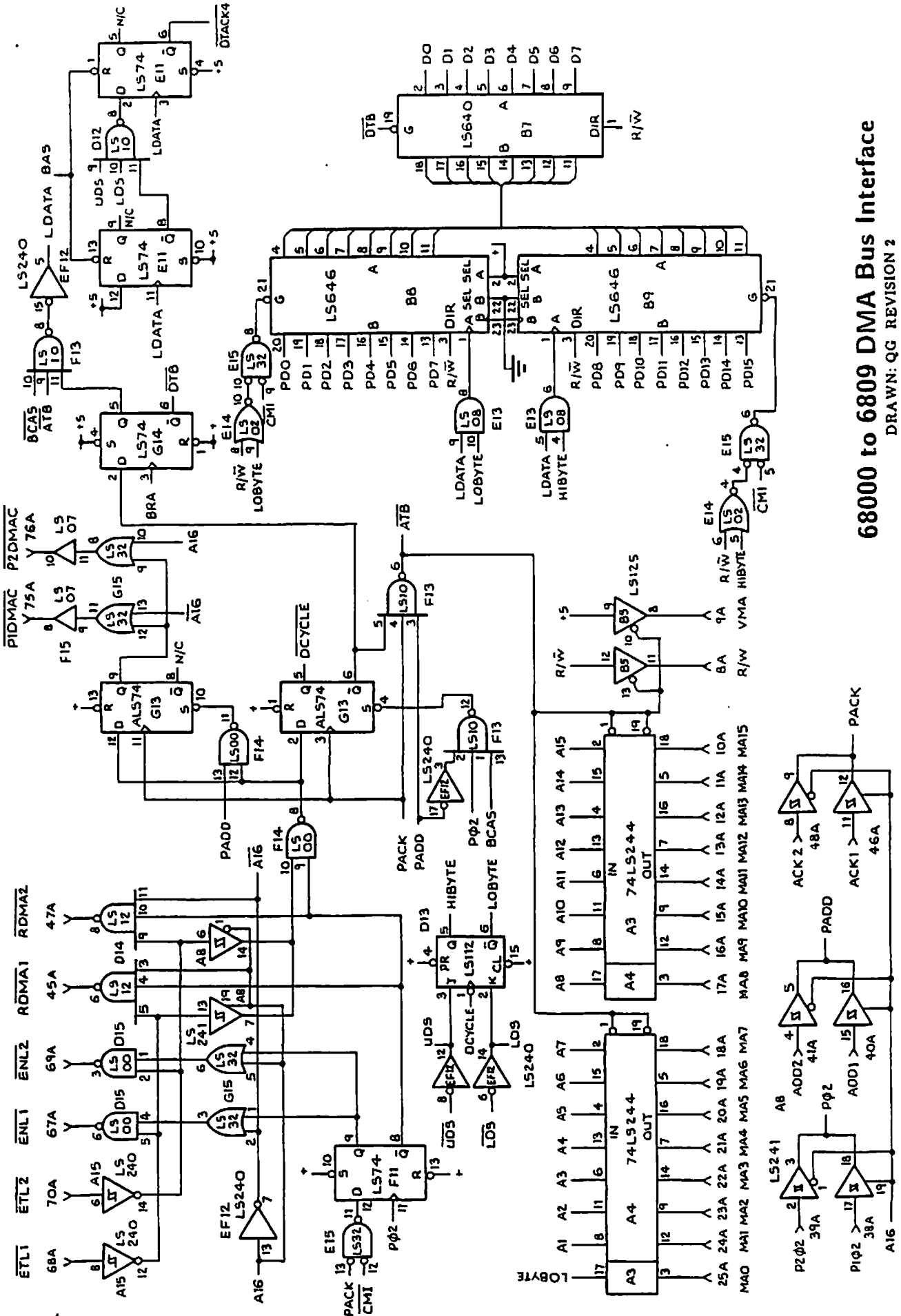
DRAWN: QG REVISION 2

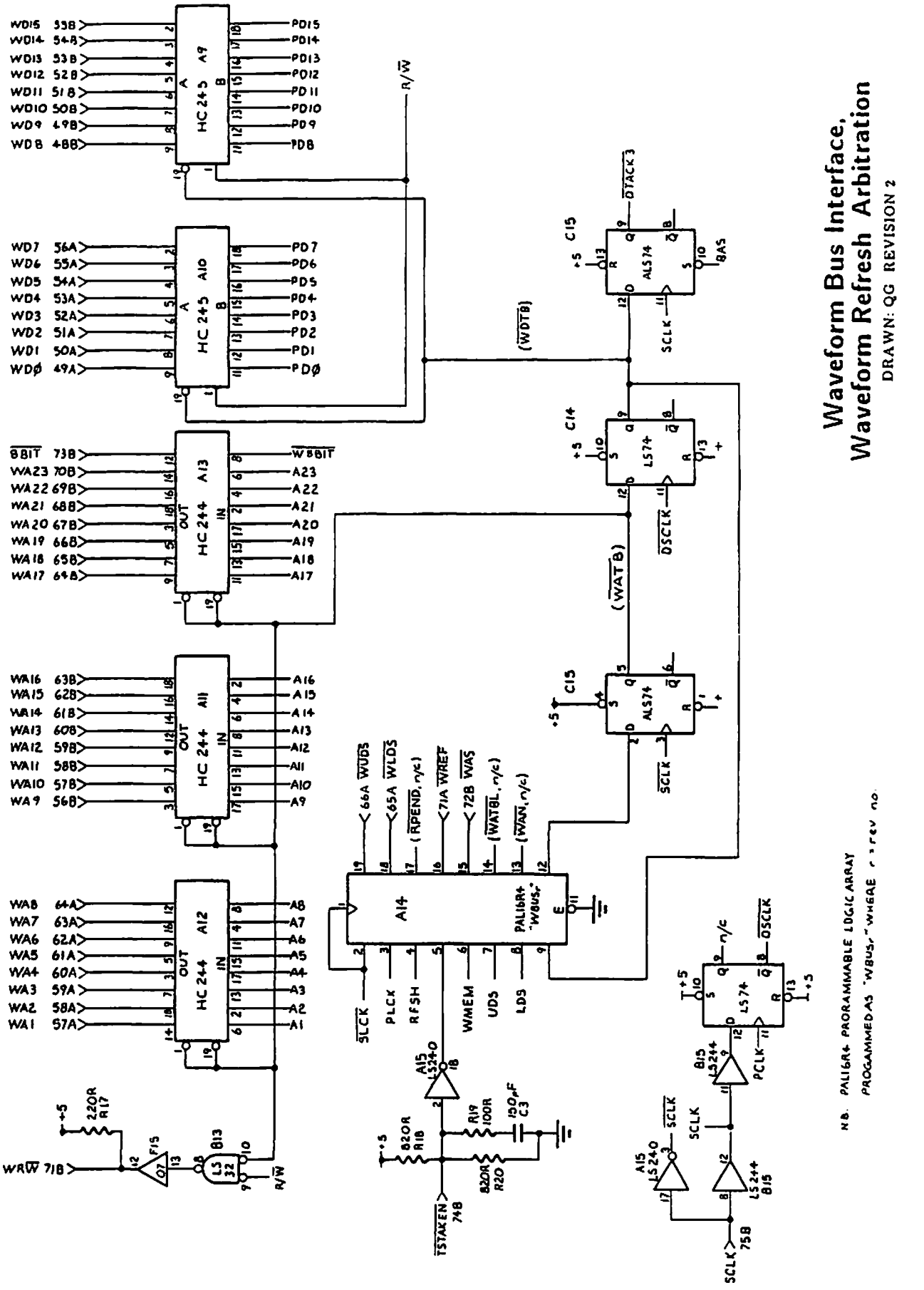
DRAMS 150 nS ACCESS MAX. 256K x 1BIT NORMALLY INSTALLED

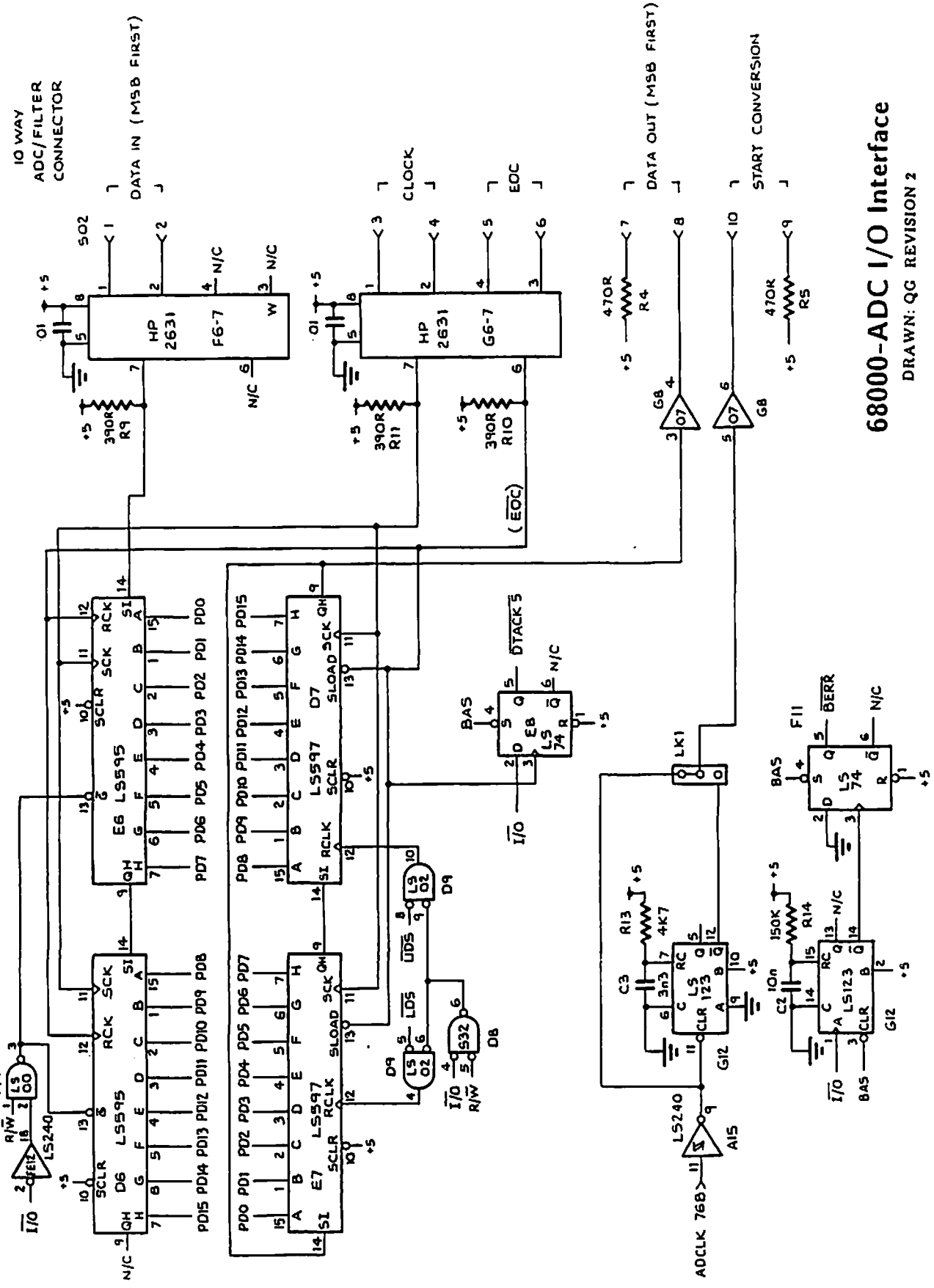
64K EQUIVALENTS - 6665 - 15 (MOT)
4164 - 15 (FAIRCHILD)
4564 - 15 (MOSTEK)
4164 - 15 (MITSUB)
8264 - 15 (FUJITSU)

256K COMPATIBLE TYPE:
6256 - 15 (MOT)
TMS 4256 (TEXAS)

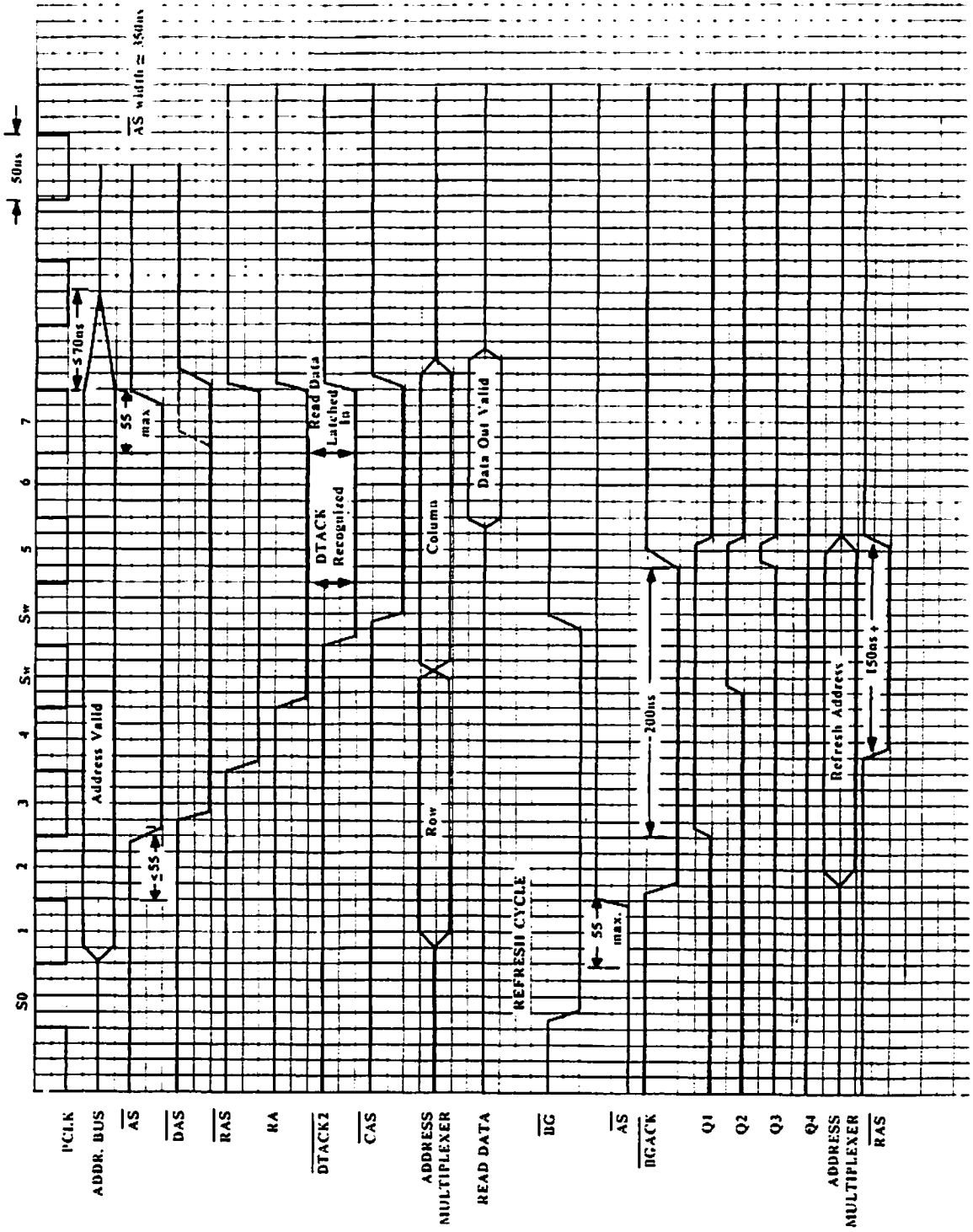

Interrupt and Reset Control
DRAWN: QG REVISION 2
10 -WAY UNPROTECTED
HEADER OTHER PINS U/C
CONTROL PANEL SO1
+5
220R R15
CR2
"HALT"
R16 220R
CR1
"RESET"
RESET
HALT
G8
07
42A SYRES
W88IT
IPL2
IPL1
IPL0
A6
LS240
INPUTS
OUTPUTS
N/C N/C
B6
LS174
CL
F14
LS 00
BCAS
A5
LS74
BRA
B1
LS 30
LS240 : A1
LS244 : A2
CAS 34
RA 37A
FCXX438
MA7 18A
MA6 19A
MA5 20A
MA4 21A
MA3 22A
MA2 23A
MA1 24A


68000 to 6809 DMA Bus Interface
DRAWN: QG REVISION 2

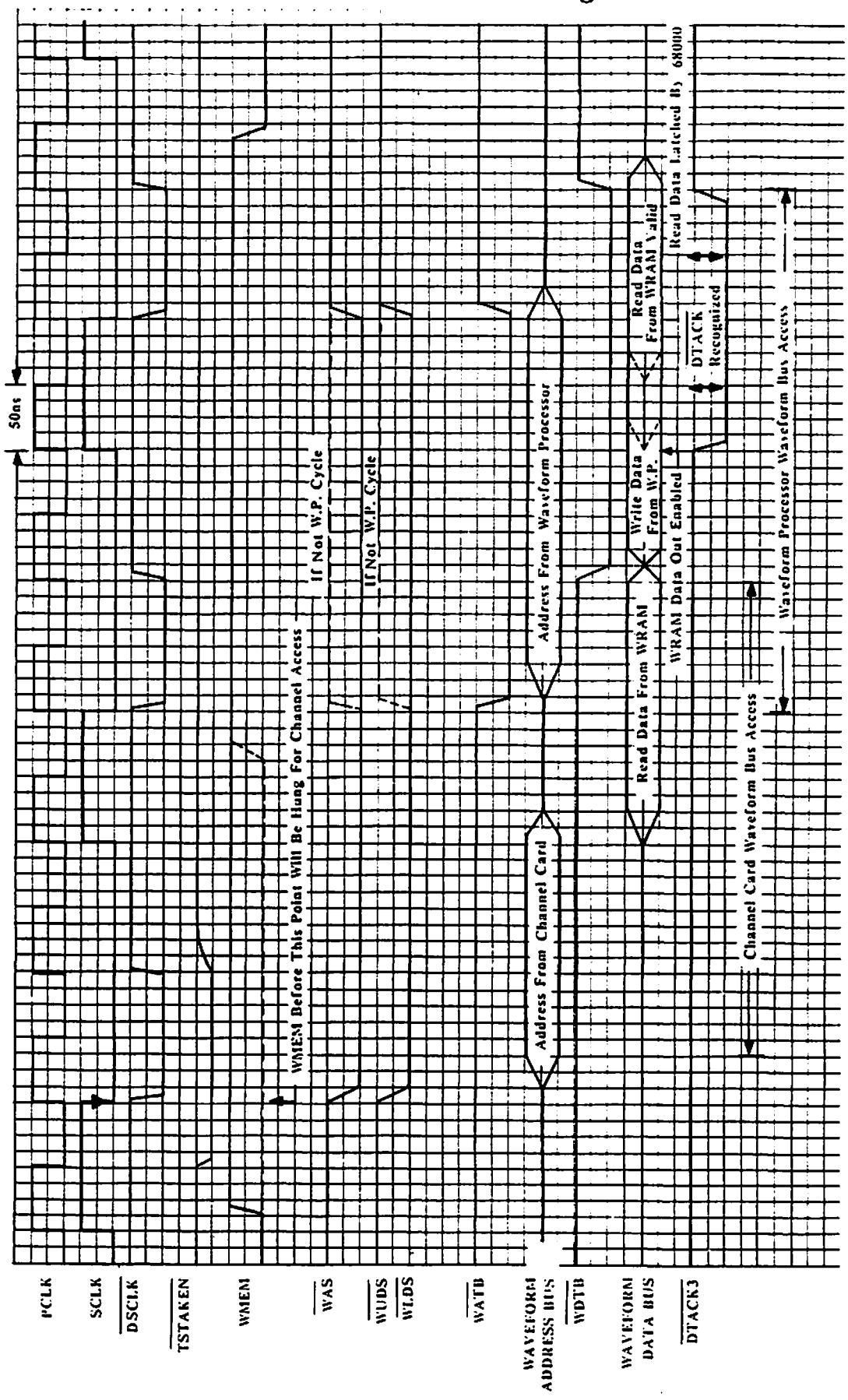

Waveform Bus Interface,
Waveform Refresh Arbitration
DRAWN: QG REVISION 2
WD15 53B
WD14 54B
WD13 53B
WD12 52B
WD11 51B
WD10 50B
WD9 49B
WD8 48B
HC 245 A9
PD15
PD14
PD13
PD12
PD11
PD10
PD9
PD8
R/W
WD7 56A
WD6 55A
WD5 54A
WD4 53A
WD3 52A
WD2 51A
WD1 50A
WDØ 49A
HC 245 A10
PD7
PD6
PD5
PD4
PD3
PD2
PD1
PDØ
(WDTB)
+5 C15
AL574
DTACK3
BAS
SCLK
+5 C14
LS74
DSCLK
(WATB)
+5 C15
AL574
SCLK
8BIT 73B
WA23 70B
WA22 69B
WA21 68B
WA20 67B
WA19 66B
WA18 65B
WA17 64B
HC 244 A13
W8BIT
A23
A22
A21
A20
A19
A18
A17
WA16 63B
WA15 62B
WA14 61B
WA13 60B
WA12 59B
WA11 58B
WA10 57B
WA9 56B
HC 244 A11
A16
A15
A14
A13
A12
A11
A10
A9
WA8 64A
WA7 63A
WA6 62A
WA5 61A
WA4 60A
WA3 59A
WA2 58A
WA1 57A
HC 244 A12
A8
A7
A6
A5
A4
A3
A2
A1
66A WUDS
65A WLDS
(RPEND, n/c)
71A WREF
72B WAS
(WATBL, n/c)
(WAN, n/c)
A14
PAL16R4 "WBUS"
SLCK
PLCK
RFSH
WMEM
UDS
LDS
A15 LS240
+5 820R R18
R19 100R
150pF C3
820R R20
TSTAKEN 74B
+5 220R R17
WRW 71B
F15
B13 LS 32
R/W
+5
n/c
LS74
OSCLK
PCLK
B15 LS244
A15 LS240
SCLK
B15 LS244
SCLK 75B
NB. PAL16R4 PROGRAMMABLE LOGIC ARRAY
PROGAMMED AS "WBUS" WHERE [illegible] REV 00


68000-ADC I/O Interface
DRAWN: QG REVISION 2
10 WAY ADC/FILTER CONNECTOR
DATA IN (MSB FIRST)
CLOCK
EOC
DATA OUT (MSB FIRST)
START CONVERSION
HP 2631
F6-7
G6-7
390R R9
390R R11
390R R10
470R R4
470R R5
LS595
LS597
LS240
LS74
LS123
LS00
LS02
LS32
4K7
150K R14
3n3
ADCLK 768
I/O
R/W
BAS
LK1
DTACK
BERR

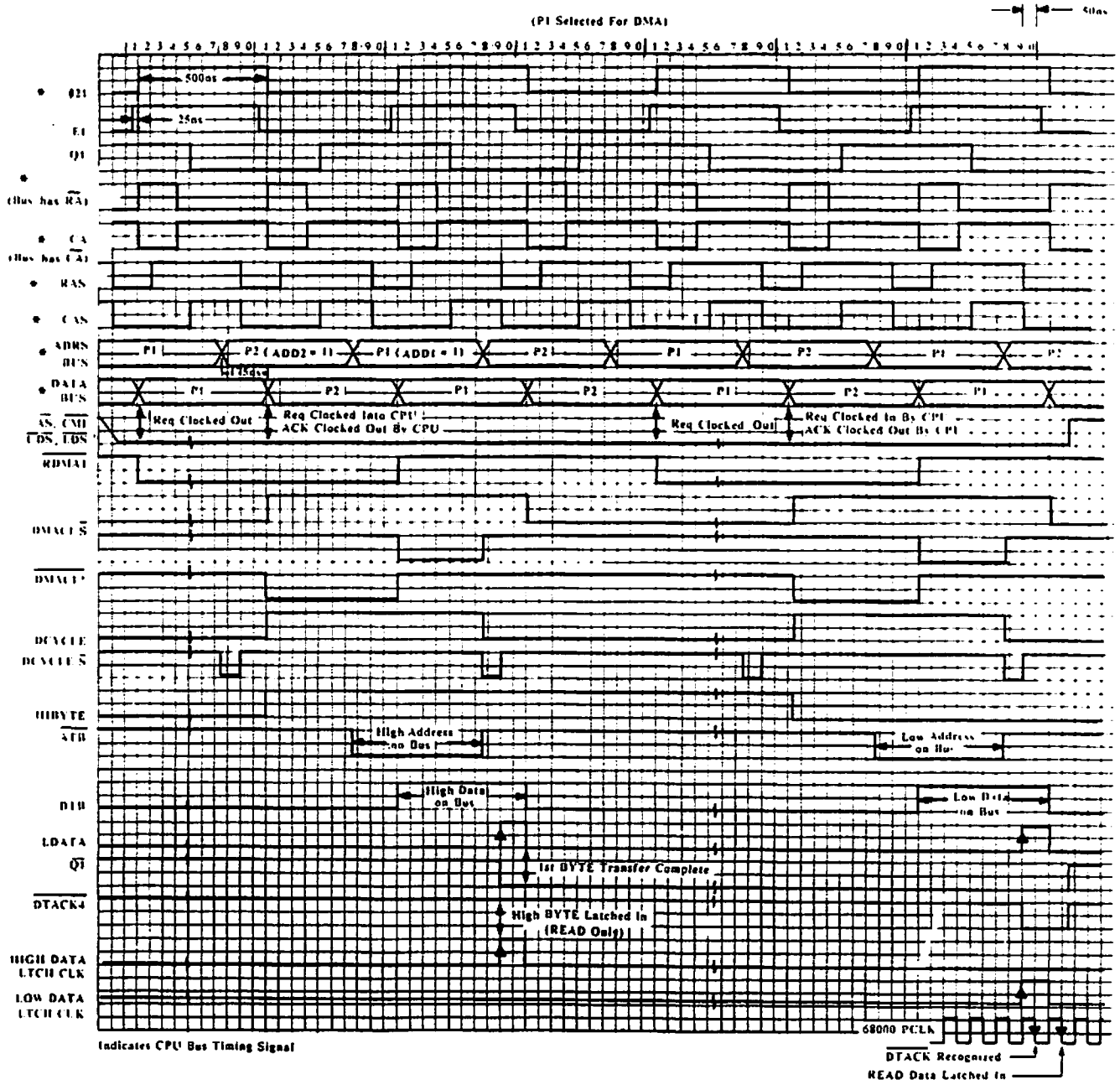
68K Timing Dynamic RAM Access



Waveform Bus Interface Timing



CMI-33 Waveform Processor / General Interface Card 68000 To CPU (6809) DMA Interface



CMI-39 2.13

Waveform Ram

Introduction

The Waveform RAM (WRAM) provides the bulk fast memory required for storage of multi-sampled sounds, although it may be used for any purpose. It resides on a 24 bit address buss which is controlled by the Waveform Processor (WP) and Channel cards and the data path is 16 bits wide. When accessed by a Channel card, the data ouput from the WRAM is received not by the Channel card but the Channel Support Card, for transmission to the audio board DACs. Currently the only device which can write to the WRAM is the WP.

Memory Configuration

Each card contains 2M bytes of RAM using 256K x 1 bit chips. Up to and including Revision 2, 64K chips were supported using a PC link block to adjust the memory configuration and yielding 512K bytes per card. Revision 3 and above only accommodate 256K bit chips. Up to seven cards may be installed in the Series III motherboard, DIP-switched to different address ranges. 7 cards containing 256K chips will provide 14M bytes of RAM, filling the available address space (the bottom 2M is used by internal decoding on the WP).

The WP and Channel cards differ slightly in their respective "views" of WRAM:

Channel Card - generates the upper 23 address lines plus the mode bit. The bottom address line A0 does not exist physically but is constituted by the two buss lines Upper and Lower Data Strobes (UDS, LDS). When a Channel accesses WRAM the WP buss arbitration logic automatically asserts both strobes, since all channel accesses yield 16-bit outputs.

WP - also generates 23 address lines and the mode bit, plus either or both of UDS and LDS according to whether a byte or word access is required by the 68000. The operation of UDS and LDS is independent from and should not be confused with the mode bit explained below.

The main difference between the Channel and WP views of the WRAM is that the 68000 has an internal concept of bytes and words and generates, internally at least, A0. The address generated by the channel card has no internal A0 and an implicit zero is tacked on in the 16-bit mode. In the 8-bit case the address is shifted right to use A1 as the odd/even byte selector.

As indicated by the memory maps, the lowest number 2M card which can be installed is card 1. There is no memory mapping of WRAM contents.

8/16-bit Modes

The WRAM may be read in either full 16-bit or left-justified 8-bit mode. 16-bit mode gives optimum audio quality, while 8-bit mode doubles the length of sounds which can be stored, with reduced audio performance. Mode selection is on a cycle-by-cycle basis so mixed 8- and 16-bit sounds can be in WRAM simultaneously. The mode is selected by an auxiliary buss line which must be driven appropriately by the accessing device.

In the 8-bit case the data byte is output on the most significant data lines whether or not it was written at an odd or even address, and the least significant 8 data lines are driven with zero. As mentioned above the waveform address is shifted right once by the WRAM so a byte written at a given address in 16-bit mode will be read back at twice the address in 8-bit mode. This has the side effect of doubling the apparent size of each WRAM card to 1M or 4M bytes. The "shifted out" address line, A1, is used to select the odd or even byte and control the left-justification circuitry. See WRAM memory maps for further explanation.

The purpose of the 8-bit mode is for 8-bit sounds to be played by the channel cards. There is no need for the WP to access the WRAM in 8-bit mode except to test the WRAM. When the 16-bit mode is selected, WRAM appears as just normal memory which supports byte and word accesses by the 68000 as described in Section 1.2. Code can be executed in WRAM by the 68000 provided it accesses WRAM in 16-bit mode.

Timing Generation

(refer schematic CMI-39-00 and timing diagram)

All timing is generated from one bus signal SCLK, which originates from the Channel Support card, and control signals from the Waveform Bus arbitration logic on the Waveform Processor. The whole Waveform Bus is synchronous to this signal. SCLK has a 300nS period, which sets the bus cycle time at 300nS, and a 1:3 duty cycle. SCLK is inverter-buffered and fed into a ten tap, 25nS/tap digital delay line at A6. Four delayed versions of SCLK are then combined in various ways to produce the complex waveforms required to drive the RAM array at the highest possible speed.

A valid access cycle is indicated by a low on the control signal WAS (Waveform Address Strobe), and a refresh cycle by a low on WREF (Waveform Refresh). In either case, RAS (Row Address Strobe) is clocked on the rising edge of the 4th delay line tap and is cleared again by the next low on the 3rd tap. The RAS pulse is driven through the HC244 buffer C18 (CMI-39-01) to all RAM chips.

The WR/W line is buffered by A18 and latched at A4 by rising edge of tap 3. Since this flip-flop is not otherwise set or reset, LR/W (Latched Read/Write) is updated every 300nS.

RA and CA are the Row and Column Address drive lines which control the address multiplexor (CMI-39-02). While these two signals are essentially the inverse of each other, RA is generated by the LS00 at A9 and CA by the ALS08 at A7. This ensures that they are as close as possible to non-overlapping so that contention is not caused on the multiplexed address lines. CA and RA both are inverted to produce CA and RA which will be similarly non-overlapping. Thus the Column address is enabled when both taps 4 and 5 are high, and the Row address is enabled on the opposite condition.

CAS is just a timing signal generated continuously which is later qualified to produce the actual CAS (Column Address Strobe) signals to the RAM array. It is 100nS long, from the rising edge of tap 7 to the falling edge of tap 3.

The Output Enable signals $\overline{OE1}$ - $\overline{OE4}$ drive the latches and buffer on CMI-39-03. The logic preceding the data inputs to the LS175 at B9 will be explained along with output section. The timing of these signals are that they are clocked out on the rising edge of tap 7, and cleared when tap 3 is high but tap 5 is low. This results in a 200 output drive pulse during valid read cycles. If READ is low, the latch will be held cleared across the clock edge so no outputs will be enabled.

Address Decoding
refer schematic CMI-39-01)

As described in Sect. 1.2, 256K RAM chips yield 2M bytes on each card, and 7 cards can be plugged into the Series III system. Therefore the top 3 address lines are used as the card select bits. In 16-bit mode, B8BIT is low so WA23-WA21 are buffered through A17, through the link block at A15 to become CS2-CS0 and into the LS85 comparator at A14. If the data on these three lines agree with the setting of the DIP switch, a high is output from the LS85. If a low on $\overline{WAS}$ indicates a valid access cycle, a low is generated by A9 pin 8 and this is latched on the rising edge of tap 3 (SCLKD125 = SCLK inverted and delayed by 125nS) to produce the enable for the memory rank decoder A3.

In 8-bit mode the one-bit right shift of the address is effected by enabling buffer A16 instead of A17 when $\overline{B8BIT}$ is low. Then only WA23 and WA22 are used as card select lines because four cards fill up the address space (see memory maps).

The next two address bits (WA20 and 19 in 16-bit mode, WA21 and 20 in 8-bit mode) come out of the link block as BLK1 and BLK0. These are used as the block number, to determine which rank of RAM chips is accessed. BLK1 and BLK0 are again latched by SCLKD125 and input to the 1-of-4 decoder A3. The decoded rank select goes to both halves of A2. One of the Upper and/or Lower $\overline{CAS}$ outputs are then driven, depending on the CAS timing pulse and either or both buffered data strobes (BUDS and BLDS).

Being a TTL device, A2 has to drive the RAM array through series resistors. At the end of the CAS pulse both halves of A2 are desabled and the $\overline{CAS}$ lines are pulled up quickly by 330R pullups.

The valid card select signal (A9 pin 8) is also combined with the BR/W signal (Buffered read/write) and latched to produce the READ signal. READ feeds back to the logic on CMI-39-00 which inhibits the output enable signals if a cycle is not a valid read.

The next two address lines (WA18, 17 in 16-bit mode, WA19, 18 in 8-bit mode) are used by the top ninth of the address multiplexor formed by the LS125 B18. Either of these two gates are enabled by the active low Row and Column Address lines ($\overline{RA}$ and $\overline{CA}$) and drive A8 through the HC244 buffer. The lower of the two address lines is latched at B3 first.

Revision 2 and earlier cards support 64K RAM chips. If these are used each card only provides one quarter as much memory. To keep successive cards contiguous in the memory space, all the card select and rank select lines must be shifted down 2 bits with the respect to the address bus. This is performed by changing the link block A15 to connect pins 3-16, 4-15, 5-14, 6-13, 7-12 and 8-11. Then in 16-bit mode WA23 is ignored, WA22-WA19 become CS3-CS0, and WA18 & 17 become BLK1 & 0. 64K chips do not have an A8 address line so LS125 multiplexor inputs do not have to be adjusted. Again, 8-bit mode shifts the card and rank select bits along once.

The HC244 buffer C18 drives the write (/W) lines and the $\overline{RAS}$ lines of the four ranks of RAM.

Address/Refresh Multiplexor

(refer schematic CMI-39-02)

Dynamic RAMs have multiplexed address inputs which allow, in this case, 256K bits per chip to be addressed with only 9 address pins compared to 18 pins which would be required by a non-multiplexed addressing arrangement.

During a valid memory access the first half of the address is enabled by a high on RA and strobed into the RAMs on the falling edge of $\overline{RAS}$. In 16-bit mode ($\overline{B8BIT}$ high) this is WA1-WA8 through B14, and in 8-bit mode, WA2-WA9 are driven through B15. The waveform address buss is stable during the critical setup and hold times before and after $\overline{RAS}$ so the buffers can drive the RAMs directly.

It can be seen from the timing diagram that the channel card does not assert its address for very long after the falling edge of $\overline{CAS}$, so it is necessary to latch the column address. WA9-WA16 are enabled in 16-bit mode, and WA10-WA17 in 8-bit mode, onto the HC373 inputs at B13. These are both latched and driven to the RAMs by B13 when CA goes high. The falling edge of $\overline{CAS}$ clocks the column address into the RAM chips.

During a refresh cycle both the row address buffers and the column address latch are disabled by the low on $\overline{BWREF}$. The same signal increments the LS393 refresh counter and drives the refresh count onto the RAM address lines.

Data Bus Interface

(refer schematic CMI-39-03)

HC244s B1 and B17 are permanently enabled to buffer data off the waveform data bus to the RAM D inputs. During write cycles the data is written in to the RAM on the falling edge of $\overline{CAS}$. During read cycles, input data is ignored.

The remaining three latches and one buffer are used during read cycles to output 16-bit data and left-justified 8-bit data. Left-justified means that whether the data byte is in the low or high 8 bits of the addressed memory word, it appears on the upper 8 bits of the data buss and the lower 8 bits are driven to zero. All latches are clocked by the falling edge of the CAS timing signal which is near the end of the low-going $\overline{CAS}$ pulse on the RAM chips.

The latch outputs are enabled by the LS175 outputs $\overline{OE1}$-$\overline{OE4}$ on CMI-39-00. The three cases are: 16-bit read, 8-bit even read, and 8-bit odd read. The high RAM byte must drive the high buss lines in the 16-bit read case (B8BIT low) or in the 8-bit even case (ODD low). This gives the gating for $\overline{OE4}$.

The low RAM byte must drive the low buss lines only in the 16-bit case, so $\overline{B8BIT}$ high is the input for $\overline{OE3}$.

The low RAM byte must drive the high buss lines only in the 8-bit odd read case when B8BIT and ODD are both high. This gating produces $\overline{OE2}$.

$\overline{OE1}$ enables the buffer with all inputs to ground onto the low data lines, which is required for any 8-bit cycle (B8BIT low). The ODD signal is simply buffered WA1, which is ignored by the decoding and multiplexor during 8-bit cycles. To put it another way, WA1 is "shifted in" to the WA0 position which is the odd/even byte selector and does not explicitly exist on the buss.

RAM Array

(refer schematic CMI-39-04.05)

The CMI-39 memory is composed of 256K or 64K by 1-bit chips. The 64-chip ram array is divided into four columns, or ranks, of 256K or 64K by 16 bits each. Rank 0 which occupies the bottom quarter of the address space, is in F column nearest the front edge of the board. Each rank is divided into an upper and a lower byte distinguished by upper and lower $\overline{CAS}$ signals. It is the $\overline{CAS}$ signal alone which determines whether a given group of 8 chips are accessed: data in, $\overline{RAS}$ and $\overline{W}$ signals are distributed to all chips but it is the presence of $\overline{CAS}$ which determines whether data is written into or read out from a RAM chip.

Refresh cycles are performed by asserting $\overline{RAS}$ while the refresh count is driven on the address lines, without a subsequent $\overline{CAS}$ pulse. During an access cycle of one group with $\overline{CAS}$ asserted, the presence of $\overline{RAS}$ without $\overline{CAS}$ on another group will refresh the row of memory cells in the non-accessed group corresponding to the accessed row address. The accessed group is also refreshed in the process of reading or writing to it. It is for this reason that the Waveform Processor Bus Arbitration logic gives higher priority to channel card accesses than to refresh: if channel cards are hogging the buss to the extent that refresh cannot be serviced it can be guaranteed that the RAM array will be refreshed by the channel accesses alone. This is true providing at least one of the channels is playing a loop of 256 or more words, since an 8-bit refresh count must be fully cycled to refresh the whole of memory. This is also why the least significant 8 waveform address bits, which change the most rapidly when a channel is looping, are used as the row address.

Timing Tolerances

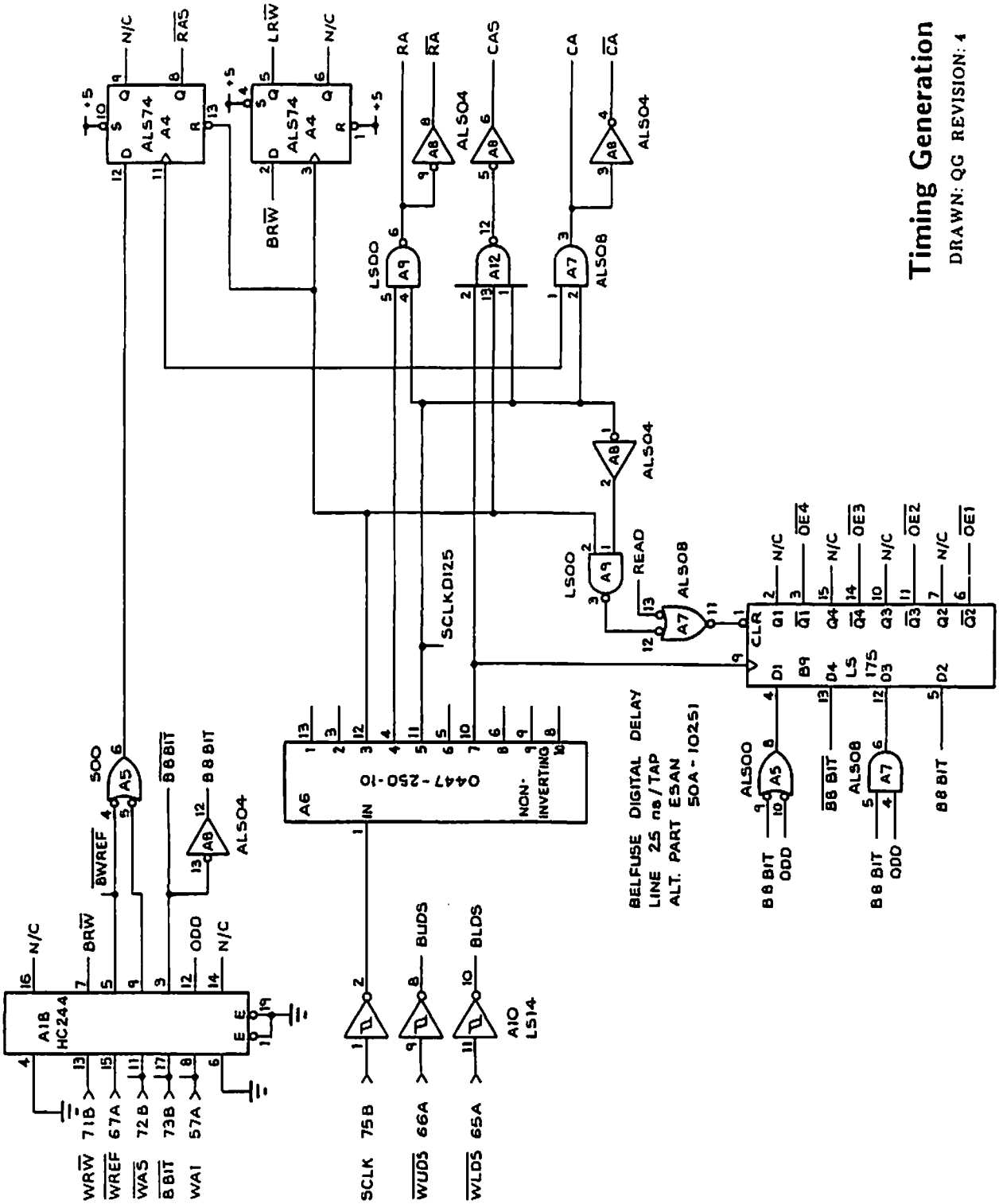
The CMI-39 Waveform RAM operates at a continuous cycle time of 300nS and is designed to use 150nS access time RAM chips which usually have a minimum cycle time of 300nS. As a result, some timing tolerances are extremely tight and it was not possible to design the board to cope with all components operating at their worst-case level. In practice this should not be a problem since RAM timing requirements such as the 300nS cycle time are specified for worst-case conditions of temperature, radiation level etc. and most RAMs will still work when being driven considerably out of spec. However, to provide a guide in case future component variations are suspected to be causing problems, the following is a list of tight parameter specs and actual measured timings from typical CMI-39 boards. Various RAMs have slightly different timing requirements: the list specifies the most difficult to meet of several brands analyzed. In most cases, 256K chips have easier specs than 64K chips.

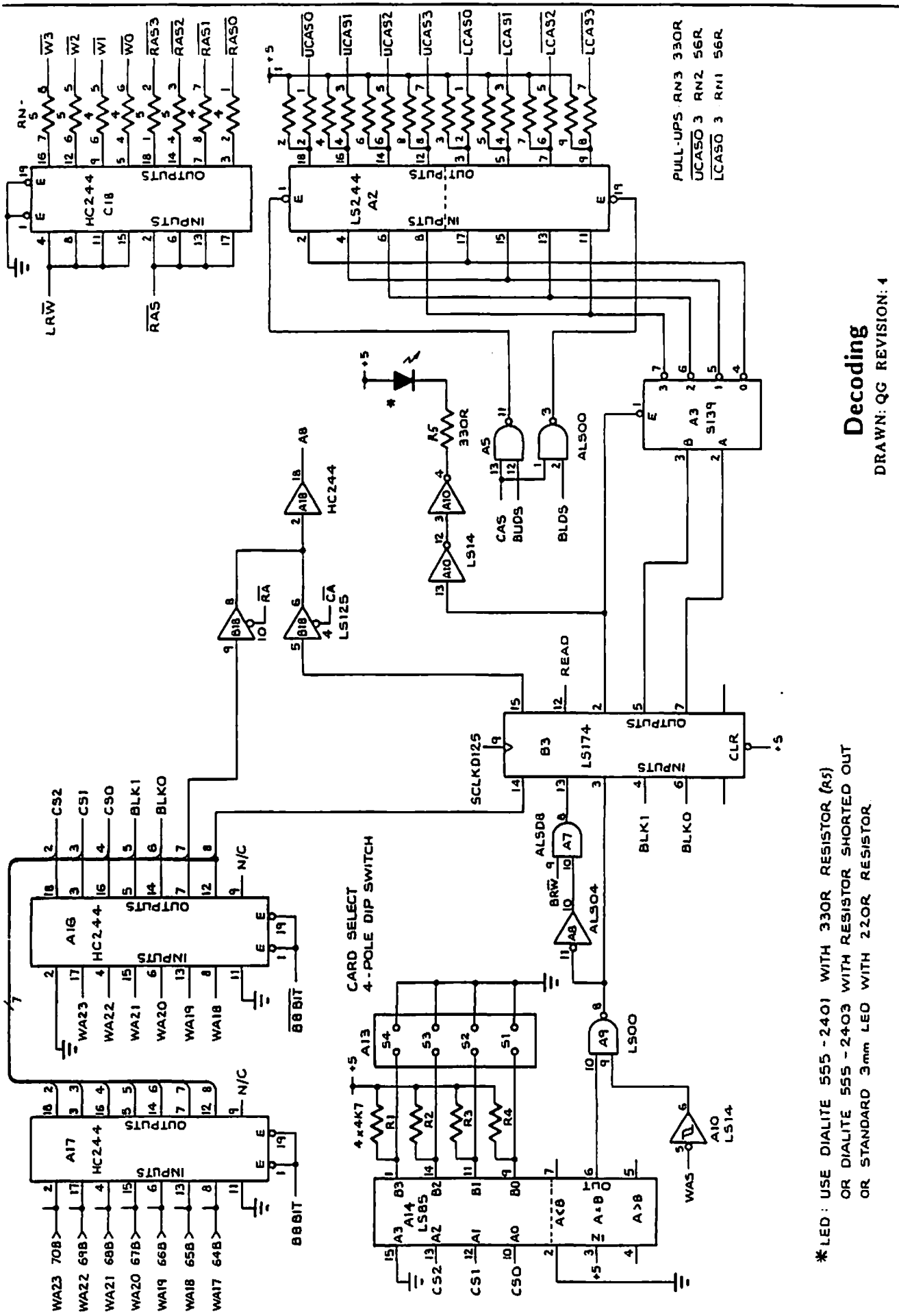
Symbol	Parameter	Spec.	Measured
t_{RC}	Cycle time	300	300
t_{RP}	RAS precharge	120	120
t_{RAS}	RAS width	150	150
t_{CAS}	CAS width	75	80
t_{RCD}	RAS to CAS lead time	75	75
t_{RSH}	RAS hold	75	80
t_{CSH}	CAS hold	150	150+
t_{ASR}	Row address set up	0	40
t_{RAH}	Row address hold	20	20
t_{ASC}	Column address set up	0	30
t_{CAH}	Column address hold	45	140
t_{DS}	Data in set up	0	50

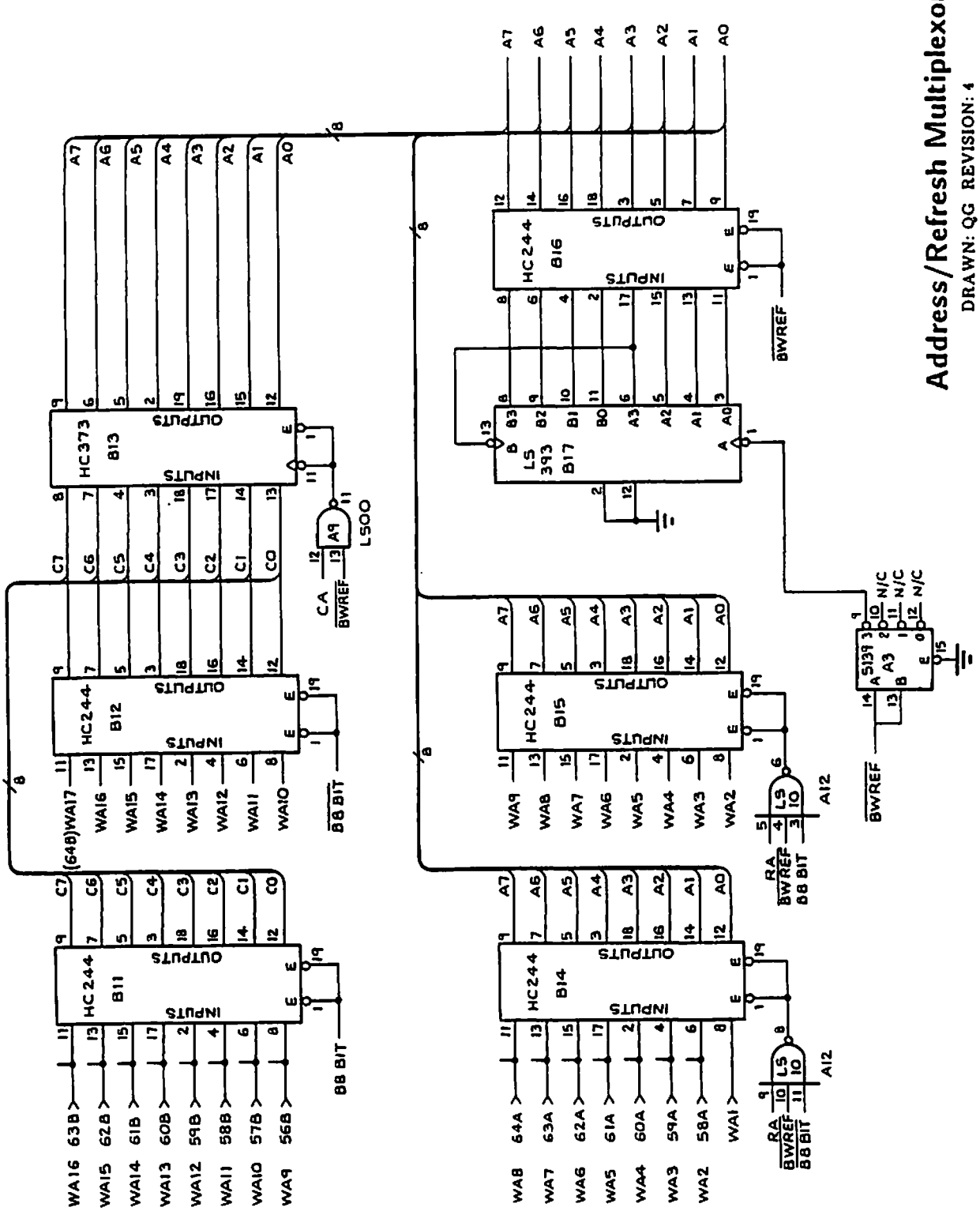
The above specs are all minimum, and all entries are in nS. All measurements are made on the RAM chips themselves.

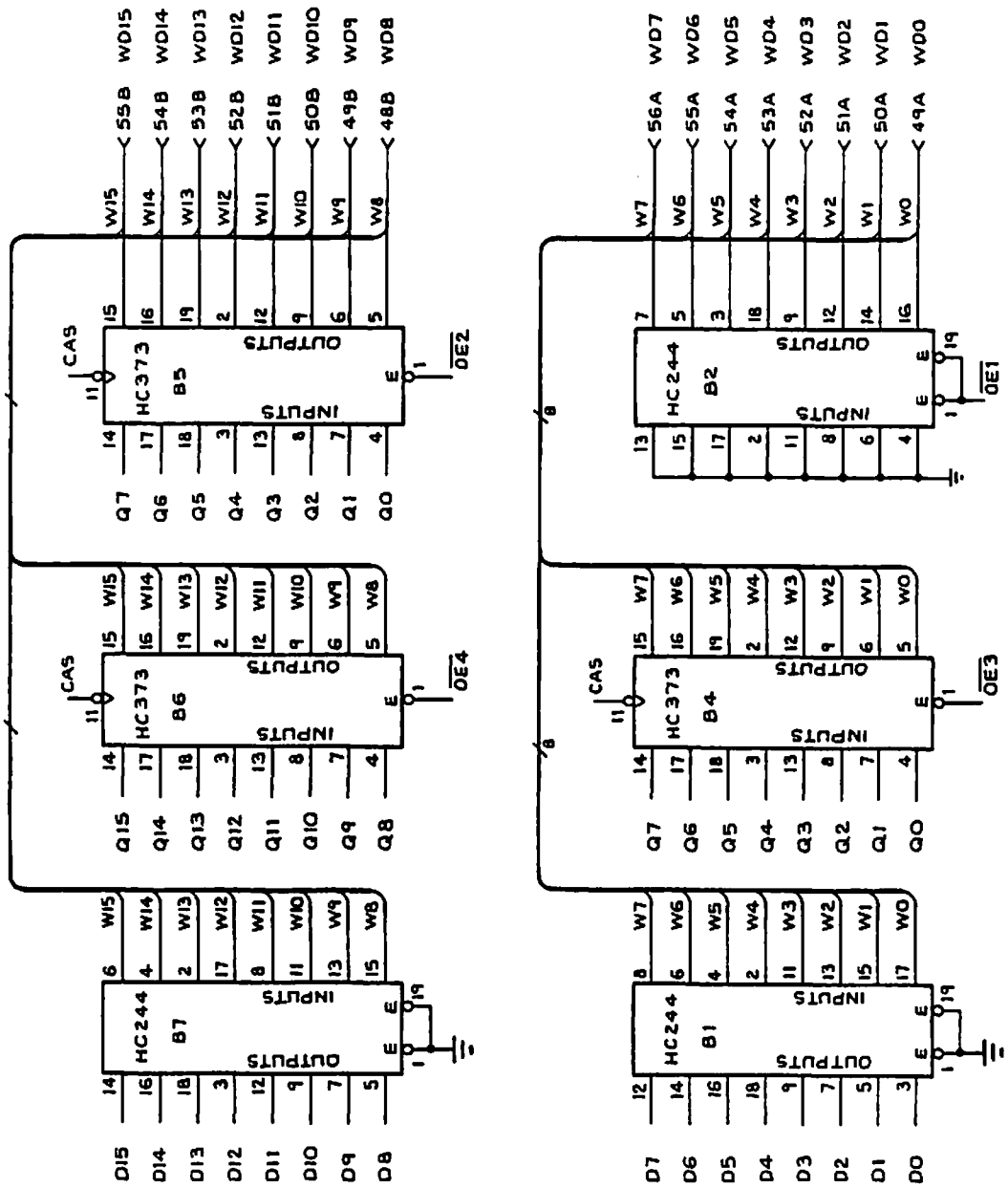
The following measurements may be of interest:-

Buffered $\overline{SCLK}$ rising edge to $\overline{RAS}$ falling edge	120
Buffered $\overline{WAS}$ falling edge to $\overline{RAS}$ falling edge	100

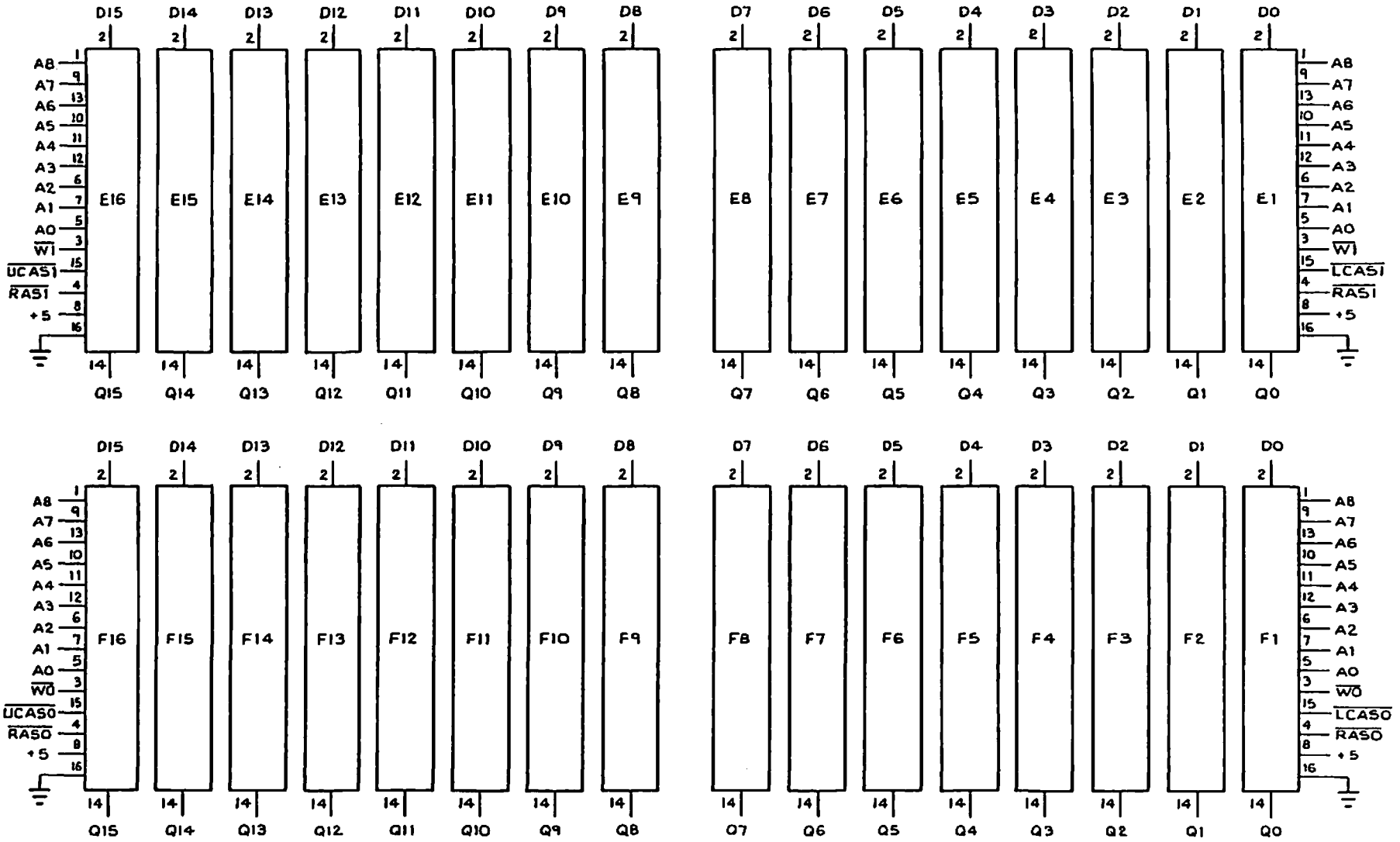

Timing Generation
DRAWN: QG REVISION: 4
BELFUSE DIGITAL DELAY
LINE 25 ns/TAP
ALT. PART ESAN
50A-10251
A18 HC244
A10 LS14
A6 0447-250-10 NON-INVERTING
AL574 A4
ALS04
ALS08
LS00
LS 175
SCLKD125
WRW 71B
WREF 67A
WAS 72B
B BIT 73B
WAI 57A
SCLK 75B
WUDS 66A
WLDS 65A
B8 BIT ODD


Decoding
DRAWN: QG REVISION: 4
PULL-UPS RN3 330R
UCAS0 3 RN2 56R
LCAS0 3 RN1 56R
*LED: USE DIALITE 555-2401 WITH 330R RESISTOR (R5)
OR DIALITE 555-2403 WITH RESISTOR SHORTED OUT
OR STANDARD 3mm LED WITH 220R RESISTOR
CARD SELECT
4-POLE DIP SWITCH
HC244
LS244
LS139
LS174
LS125
LS14
LS85
LS00
ALS08
ALS04
88 BIT


Address/Refresh Multiplexor
DRAWN: QG REVISION: 4
HC373 B13
HC244 B12
HC244 B11
HC244 B16
LS 393 B17
HC244 B15
HC244 B14
OUTPUTS
INPUTS
LS00
LS10
LS139
BWREF
88 BIT
RA
CA
N/C
A12


Data Bus Interface
DRAWN: QG REVISION: 4
B1
B2
B4
B5
B6
B7
HC244
HC373
INPUTS
OUTPUTS
CAS
OE1
OE2
OE3
OE4
WD0
WD1
WD2
WD3
WD4
WD5
WD6
WD7
WD8
WD9
WD10
WD11
WD12
WD13
WD14
WD15
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15

fairlight

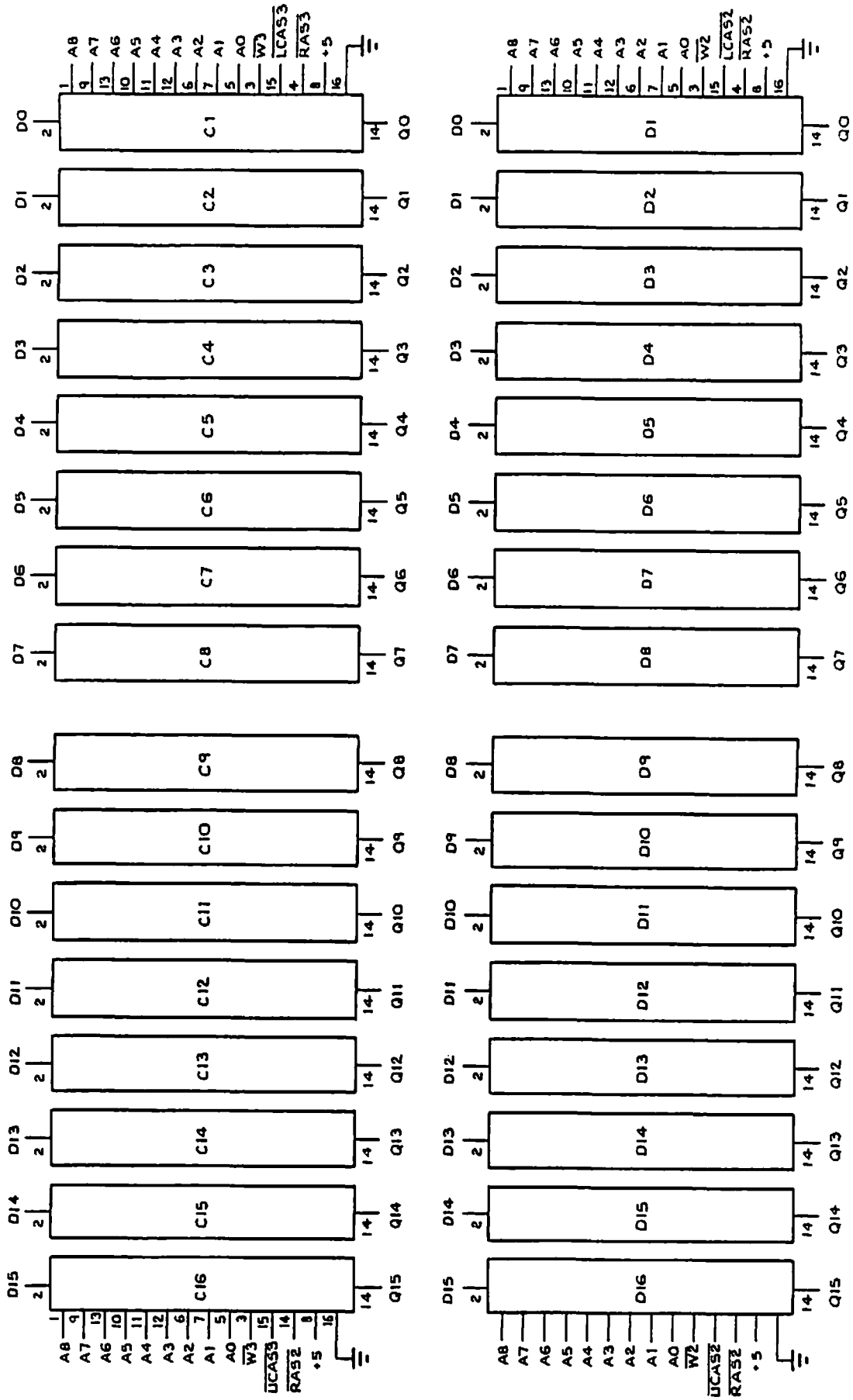


COMPATIBLE 256K RAMS:-
50256-15 (HITACHI)
MN41256 (MATSUSHITA)
6256-15 (MOT.)
41256-15 (NEC)

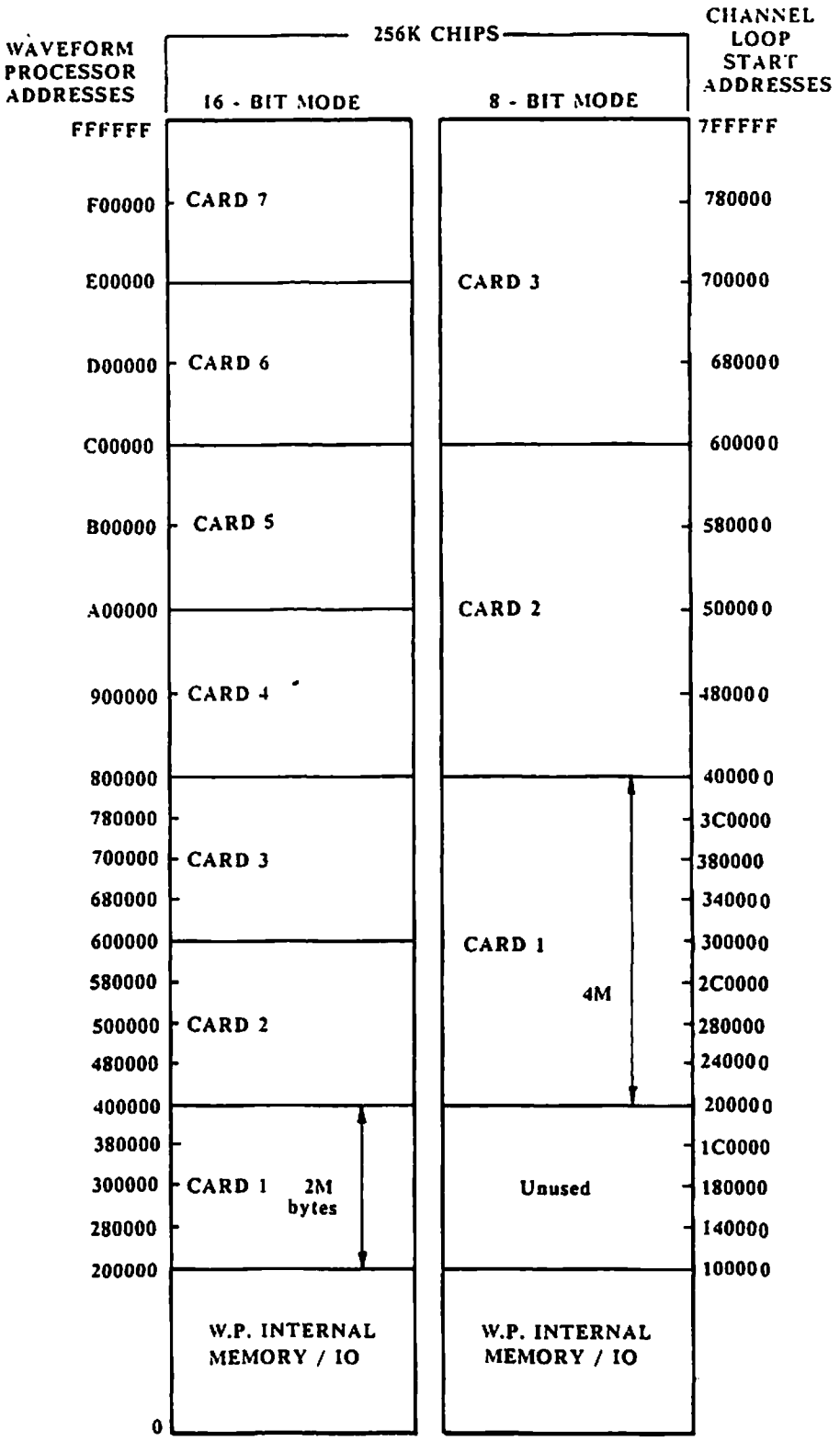
RAMS MUST BE 150 nS MAX. ACCESS TIME OR FASTER

Dynamic RAMS

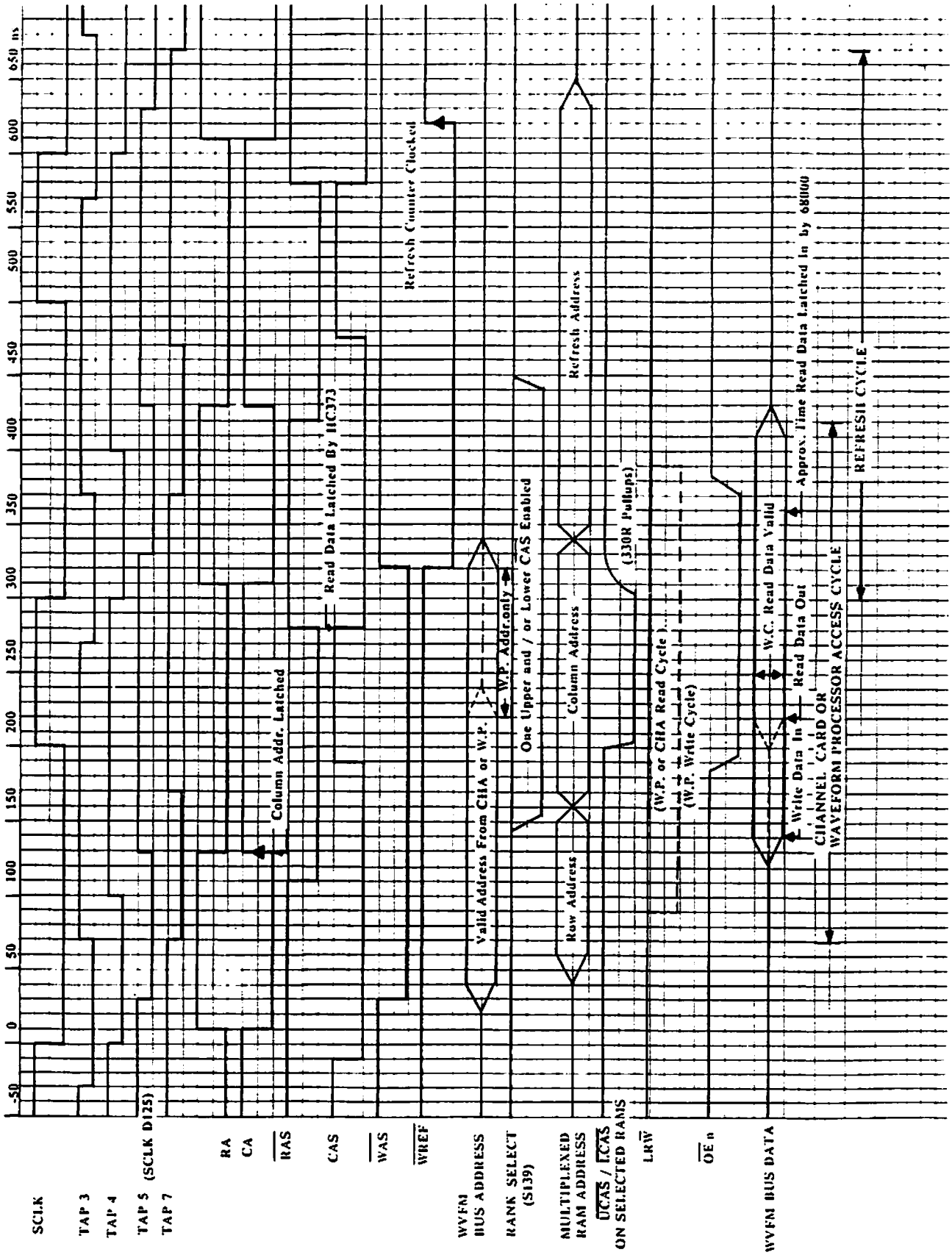
DRAWN: QG REVISION: 4


Dynamic RAMS
DRAWN: QG REVISION: 4
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C13
C14
C15
C16
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
A8
A7
A6
A5
A4
A3
A2
A1
A0
W3
W2
LCAS3
LCAS2
RAS3
RAS2
+5

CMI-39 Waveform Memory Map



Waveform Ram Timing



CMI-35 2.14

Digital Mother Board

Terminology
WP - Waveform Processor CMI-33
CS - Channel Support Card CMI-32
WRAM - Waveform RAM CMI-39
RAM - System RAM Q256

Introduction

All of the digital CMI modules described in chapters 2.2 to 2.13 of this manual, plug directly into 78 pin edge connectors mounted on the CMI digital motherboard CMI-35. This is in turn mounted on the rear of the CMI card cage. The motherboard is the means by which all logic signals and power supplies are routed between the plug-in modules. This section specifies each of these signals for each module, starting from the left.

All modules are "double sided" so require two columns of pins on each connector. "Side A" refers to the wiring side of a plug-in module which corresponds to the left hand column of pins on the motherboard when viewed from the front of the card cage. Conversely, "Side B" refers to the component side of the module and connects to the right hand column of pins on the edge connector.

Pin numbers not included in the following lists are not used, and have no connection on the motherboard. Signals which are listed but have "N/C" entered as the source or destination are those which have been connected to edge connector fingers but have no connections leading to or from the corresponding pins on the motherboard.

A pin specified as an input ("I/P") will have the Source module which drives that input entered in the Source/Destinations column. Conversely an output ("O/P") will have the Destination or driven module entered in the Source/Destinations column.

Active-low signals are indicated by overlining. All other signals are active-high. Where different names have been used for one signal going between various modules, the Signal Name column contains the name for the module of the current section, and the alternative name is enclosed in brackets in the Source/Destination column.

This document refers to the CMI-35 Rev. 3 motherboard.

General Description

The Series III CMI contains two entirely independent busses, and this is reflected in the layout of the CMI-35 motherboard. The first buss is the 2MHz dual-6809 processor buss, or CPU buss. It extends from the slot-27 end of the board on which the Q-777 SCSI controller resides up to slot 16, the Channel Support slot. Part of the CPU buss then extends further to slot 8 which is occupied by Channel 8. The only cards which have no connections at all to the CPU buss are the Waveform RAM cards. The CPU buss comprises 16 bits of address, 8 bits of data plus the system reset signal, CPU timing signals, DMA arbitration, daisy chain, and mapping signals, Peripheral Enable, Interrupts, and a few control signals.

The second buss in the system is the 3.3MHz Waveform Bus which extends from the slot-1 end of the CMI-35 board up to the Waveform Processor on slot 17. It comprises 23 address bits, 16 data bits, two data strobes, Channel card timing signals, a 17-MHz Master Oscillator differential pair, a 10MHz Processor Clock differential pair, a 3.3MHz buss clock, Channel Select lines, refresh controls, and Channel arbitration signals.

CPU Bus Signals

The following connections are common to all slots on the CPU buss and travel from slot 27 to slot 8, except that the signals on pins 9A to 17A stop at slot 15. The buss lines are arranged in a ground shield interleave pattern which provides protection against electromagnetic noise pickup and cross-talk between adjacent signal tracks. Not all modules use all the busssed signals.

Side A

Pin	Signal Name	Function	Source
42	SYRES	System reset	Q133
41	ADD2	P2 address on buss[1]	Q209
40	ADD1	P1 address on buss[1]	Q209
39	P2ø2	P2 phase 2 reference	Q209
38	P1ø2	P1 phase 1 reference	Q209
37	$\overline{RA}$	Row address mux signal	Q209
36	$\overline{CA}$	Column address mux sig.	Q209
35	$\overline{RAS}$	Row address strobe	Q209
34	$\overline{CAS}$	Column address strobe	Q209
33-26	D0-D7	Data Buss	Various
25-18	MA0-MA7	Address Buss, upper half	Various
17-10	MA8-MA15	Address Buss, lower half[2]	Various
17	ADD2	P2 address on buss[1]	Q209
16	ADD1	P1 address on buss[1]	Q209
9	VMA	Valid Memory Address[2]	Various
8	R/W	Read/write	Various
7	Key	Socket index key	
6	+12V	+12V supply rail	
3	-12V	-12V supply rail	
2,1	GND	Ground rail	

Side B

Pin	Signal Name	Function	Source
44-46	GND	Ground rail	
42-1	GND	Ground shielding pattern	
5,4,2,1	GND	Ground rail	

Notes
1. The P2ø2 and P1ø2 signals are on pins 39A and 38A respectively of slots 27 to 16, then pins 17A and 16A of slots 15 to 8.
2. The lower half of the address bus and the VMA signal only go from slots 26 to 16, and do not go to the Channels card slots.

Waveform Bus Signals

The following signals are common to all slots on the Waveform Bus and travel from slot 1 to slot 15. The Channel Support Card on slot 16 also connects to the Waveform Bus but some connections are different Not all the modules use all the signals.

Side A

Pin	Signal Name	Function	Source
67	$\overline{\text{BLOCKSEL}}$	All-channel select[1]	CS
66	$\overline{\text{WUDS}}$	Waveform Upper Data Strobe	WP
65	WLDS	Waveform Lower Data Strobe	WP
64-57	WA8-WA1	Waveform Address lower byte	Various
56-49	WD7-WD0	Waveform Data lower byte	Various
48	CHANTICK	Chan. processor interrupt clock[1]	CS
47	DRAS	Delayed RAS for channel RAM[1]	CS
46	2E	2MHz E clock for chan. proc.[1]	CS
45	2Q	2MHz Q clock for chan. proc.[1]	CS
44	PCLK	10MHz clock for WP and chans[2]	CS
43	$\overline{\text{PCLK}}$	10MHz clock for WP and chans[2]	CS
42	$\overline{\text{SYRES}}$	System Reset from CPU	CCC
39	MOSC	Master Oscillator[3]	CS
38	$\overline{\text{MOSC}}$	Master Oscillator[3]	CS

Side B

Signal Pin	Name	Function	Source
75	SCLK	3.3MHz Waveform Bus clock[4]	CS
74	TSTAKEN	Time Slice Taken	Channels
73	8BIT	8-bit mode select	Channels, WP
72	WAS	Waveform Address Strobe	WP
71	WRW	Waveform Read/Write	WP
70-56	WA23-WA9	Waveform Address high bits	Channels, WP
55-48	WD15-WD8	Waveform Data high byte	WP, WRAM
47	REFRESH	Channel refresh strobe	CS
46-44	GND	Ground Rail	
5,4,2,1	GND	Ground Rail	

Notes
1. Slots 16 to 8 only
2. Differential pair, slots 17 to 8 only
3. MOSC and $\overline{\text{MOSC}}$ are the differential pair which drive the 17Mhz Master Oscillator used as the pitch reference by all channel cards. It is generated on the Channel Support Card, slot 16 onto pins 78B and 77B. These pins connect to link connector holes "A" and "B" near the top of slot 16 the pair of signals then goes via a twisted pair flying connection to another two link connector holes marked "A" and "B" near the middle of slot 15. The latter holes connect to pins 39A and 38A of slot 15 and the MOSC signal are then bussed along all channel card slots.
4. SCLK is a rectangular wave clock: 100nS high, 200nS low.

Slots 1 to 7: Waveform RAM

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
67	WREF	Waveform Refresh	I/P	WP
66-49	see Sec. X.3			

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	TSLICE	Last Time Slice out	I/P	Ch. 7
75-48	see Sec. X.3			

Slot 8: Channel 8
Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	SO	Time Slice Out	O/P	WRAMs (TSLICE)
68	$\overline{\text{CHSEL8}}$	Channel 8 select	I/P	CS

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	TSLICE	Time Slice In	I/P	Ch. 7 (SO)
17	$\overline{\text{CHINT}}$	Channel interrupt	O/P	Q133 (IL21)

Slot 9: Channel 7
Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	SO	Time Slice Out	O/P	Ch. 8 (TSLICE)
69	$\overline{\text{CHSEL7}}$	Channel 7 select	I/P	CS

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	TSLICE	Time Slice In	I/P	Ch. 6 (SO)
17	$\overline{\text{CHINT}}$	Channel interrupt	O/P	Q133 (IL21)

Slot 10: Channel 6
Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	SO	Time Slice Out	O/P	Ch. 7 (TSLICE)
68	CHSEL6	Channel 6 select	I/P	CS

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	TSLICE	Time Slice In	I/P	Ch. 5 (SO)
17	CHINT	Channel interrupt	O/P	Q133 (IL21)

Slot 11: Channel 5
Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	SO	Time Slice Out	O/P	Ch. 6 (TSLICE)
68	CHSEL5	Channel 5 select	I/P	CS

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	TSLICE	Time Slice In	I/P	Ch. 4 (SO)
17	CHINT	Channel interrupt	O/P	Q133 (IL21)

Slot 12: Channel 4
Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	SO	Time Slice Out	O/P	Ch. 5 (TSLICE)
68	CHSEL4	Channel 4 select	I/P	CS

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	TSLICE	Time Slice In	I/P	Ch. 3 (SO)
17	CHINT	Channel interrupt	O/P	Q133 (IL21)

Slot 13: Channel 3
Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	SO	Time Slice Out	O/P	Ch. 4 (TSLICE)
68	CHSEL3	Channel 3 select	I/P	CS

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	TSLICE	Time Slice In	I/P	Ch. 2 (SO)
17	CHINT	Channel interrupt	O/P	Q133 (IL21)

Slot 14: Channel 2
Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	SO	Time Slice Out	O/P	Ch. 3 (TSLICE)
68	CHSEL2	Channel 2 select	I/P	CS

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	TSLICE	Time Slice In	I/P	Ch. 1 (SO)
17	CHINT	Channel interrupt	O/P	Q133 (IL21)

Slot 15: Channel 1
Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	SO	Time Slice Out	O/P	Ch. 2 (TSLICE)
68	CHSEL1	Channel 1 select	I/P	CS

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
77	ADCLK	A/D Convertor clock	O/P	WP
76	TSLICE	Time Slice In	I/P	CS
17	CHINT	Channel interrupt	O/P	Q133 (IL21)

Slot 16: Channel Support Card
Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	TSLICE	First Time Slice	O/P	Ch. 1
75	CHSEL1	Channel 1 select	O/P	Ch. 1
74	CHSEL2	Channel 2 select	O/P	Ch. 2
73	CHSEL3	Channel 3 select	O/P	Ch. 3
72	CHSEL4	Channel 4 select	O/P	Ch. 4
71	CHSEL5	Channel 5 select	O/P	Ch. 5
70	CHSEL6	Channel 6 select	O/P	Ch. 6
69	CHSEL7	Channel 7 select	O/P	Ch. 7
68	CHSEL8	Channel 8 select	O/P	Ch. 8
66	REFEN	Refresh enable	I/P	Q133 ($\overline{\text{ENL1}}$)
65	ACK1	P1 DMA Acknowledge	I/P	Q209

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
78	MOSC	Master Oscillator[1]	O/P	Channels
77	$\overline{\text{MOSC}}$	Master Oscillator[1]	O/P	Channels
76	TIMINT	Timer Interrupt	O/P	Q133 (IL11)
43	PENB	Peripheral Enable	I/P	RAM, Card 0

Notes
1. MOSC and $\overline{\text{MOSC}}$ are the differential pair which drive the 17MHz Master Oscillator used as the pitch reference by all channel cards. It is generated on the Channel Support Card, slot 16 onto pins 78B and 77B. These pins connect to link connector holes "A" and "B" near the top of slot 16 the pair of signals then goes via a twisted pair flying connection to another two link connector holes marked "A" and "B" near the middle of slot 15. The latter holes connect to pins 39A and 38A of slot 15 and the MOSC signal are then bussed along all channel card slots.

Slot 17: Waveform Processor

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	P2DMAC	P2 DMA Claim	O/P	RAMs (DMAC24)
75	P1DMAC	P1 DMA Claim	O/P	RAMs (DMAC12)
71	WREF	Waveform Refresh	O/P	WRAMs
70	ETL2	P2 Enable This Level	I/P	SMIDI (ENL2)
69	ENL2	P2 Enable Next Level	O/P	SCSI (ETL)
68	ETL1	P1 Enable This Level	I/P	SMIDI (ENL1)
67	ENL1	P1 Enable Next Level	O/P	N/C
48	ACK2	P2 DMA Acknowledge	I/P	Q209
47	RDMA2	P2 Request DMA	O/P	Q209
46	ACK1	P1 DMA Ackknowledge	I/P	Q209
45	RDMA1	P1 Request DMA	O/P	Q209

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	ADCLK	AD Convertor Clock	I/P	Ch. 1
43	FCXX	Control latch select	I/P	Q133

Slot 18: General Interface Card CMI-28

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
78,77	+5V	+5V Logic supply		
76	P2DMAC	P2 DMA claim	O/P	RAMs (DMAC24)
75	P1DMAC	P1 DMA claim	O/P	RAMs (DMAC11)
70	ETL2	P2 DMA Enable This Level	I/P	QFC9 (ENL)
69	ENL2	P2 DMA Enable Next Level	O/P	WP (ETL2)
68	ETL1	P1 DMA Enable This Level	I/P	Q133 (ENL1)
67	ENL1	P1 DMA Enable Next Level	O/P	WP (ETL1)
66	MIDINT	MIDI IRQ	O/P	Q133 (IL01)
65	SMPTEINT	SMPTE IRQ	O/P	Q133 (IL31)
64	FCXX	Control latch select	I/P	Q133
48	ACK2	P2 DMA Acknowledge	I/P	Q209
47	RDMA2	P2 DMA Request	O/P	Q209
46	ACK1	P1 DMA Acknowledge	I/P	Q209
45	RDMA1	P1 DMA Request	O/P	Q209

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
69	MIDINT	MIDI IRQ	O/P	N/C
68	SMIDINT	SMPTE IRQ	O/P	N/C
67	FCXX	Port select	I/P	N/C

The B-side signals on pins 67-69 are duplicates of the corresponding signals on side A, and are only used when a CMI-28 module is installed in a Series I machine upgraded to Series IIX.

Slot 19: Spare

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
78,77	+5V	+5V Logic supply		
54	$\overline{FCXX}$	Port select	I/P	Q133
46	$\overline{IRQ}$	Interrupt	O/P	Q133(IL41)

Slots 20,21: System RAM Q256

One Q256 module must be installed in slot 21. This is referred to as Card 0 and must have zero set up on the card select DIP switch on the board (see Q256 description). A second Q256 module may be installed in slot 20, and should be switched as Card 1.

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
78-77	+5V	Logic power supply	I/P	
76	$\overline{BE1}$	P1 E signal (unused)	I/P	Q209
75	$\overline{DMAC11}$	P1 DMA claim level 1	I/P	Gen. I/F ($\overline{P1DMAC}$)
74	$\overline{DMAC12}$	P1 DMA claim level 2	I/P	WP ($\overline{P1DMAC}$)
73	$\overline{DMAC13}$	P1 DMA claim level 3	I/P	N/C
72	$\overline{DMAC14}$	P1 DMA claim level 4	I/P	N/C
71	$\overline{DMAC21}$	P2 DMA claim level 1	I/P	Floppy ($\overline{DMACLM}$)
70	$\overline{DMAC22}$	P2 DMA claim level 2	I/P	H. Disk ($\overline{DMACLM}$)
69	$\overline{DMAC23}$	P2 DMA claim level 3	I/P	WP ($\overline{P2DMAC}$)
68	$\overline{DMAC24}$	P2 DMA claim level 4	I/P	Gen. I/F ($\overline{P2DMAC}$)
65	$\overline{RAMINH}$	System RAM Inhibit	I/P	Graphics ($\overline{VRAMSEL}$)
64	$\overline{FCXX}$	Port select (Mapsel RAM)	I/P	Q133
63	A/$\overline{B2}$	P2 System/User State	I/P	Q209
62	A/$\overline{B1}$	P1 System/User State	I/P	Q209
52	PENBIN	Peripheral Enable In[1]	I/P	RAM, Card 0 (PENBOUT)
51	PENBOUT	Peripheral Enable Out[1]	O/P	Various
50	VRAMEN	Video RAM Enable[2]	O/P	Graphics
48	REF	Refresh cycle	I/P	Q133
46	PERRINT	Parity Error IRQ	O/P	Q133 (IL02)

Notes

1. The PENB (Peripheral Enable) signal is driven by the PENBOUT pin 52A of RAM card 0 only, in slot 21. PENB goes to all cards which act as peripherals on the CPU bus, including PENBIN of both RAM cards on pin 51A. The main RAM is not a peripheral, but the MAPSEL and MAPRAM are (see Q256 documentation). PENBOUT of card 1 is N/C.
2. The VRAMEN output from RAM card 0 only goes to the Graphics card. VRAMEN of card 1 is N/C.

Slot 22: Q014 4-Port ACIA Module (Optional)

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
78-77	+5V	Logic power supply	I/P	
66	ENB	Peripheral Enable	I/P	RAM (PENBOUT)
65-58	IRQ0-7	Individual ACIA IRQs	O/P	N/C
57	IRQ8	Combined IRQ	O/P	Q133 (IL62)
56-53	AS6-AS9	Address select	I/P	N/C

Slot 23: Processor Control Module Q133

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
78-77	+5V	Logic power supply	I/P	
76	IL32	P2 level 3 IRQ	I/P	Q209 (IPI2)
75	IL22	P2 level 2 IRQ	I/P	N/C
74	IL12	P2 level 1 IRQ	I/P	Graphics (RINT)
73	IL02	P2 level 0 IRQ	I/P	Q133 (RTCINT) & RAM (PERRINT)
72	IL31	P1 level 3 IRQ	I/P	Q209 (IPI1) & GIF (SMPTEINT)
71	IL21	P1 level 2 IRQ	I/P	All Chs (CHINTX)
70	IL11	P1 level 1 IRQ	I/P	Ch. Supp (TIMINT)
69	IL01	P1 level 0 IRQ	I/P	GIF (MIDINT)
68	ILS2	P2 PICU latch strobe	I/P	Q209
67-64	IA42-IA12	P2 intrpt vector addr.	O/P	Q209
63	IRQ2	P2 interrupt request	O/P	Q209
62	ILS1	P1 PICU latch strobe	I/P	Q209
61-58	IA41-IA11	P1 intrpt vector addr.	O/P	Q209
57	IRQ1	P1 interrupt request	O/P	Q209
56	NMI2	P2 NMI request	O/P	Q209
55	NMI1	P1 NMI request	O/P	Q209
54	RES2	P2 restart	O/P	Q209
53	HLT2	P2 HALT	O/P	Q209
52	RES1	P1 restart	O/P	Q209
51	HLT1	P1 HALT	O/P	Q209
50	W2	P2 halt acknowledge	I/P	Q209
49	W1	P1 halt acknowledge	I/P	Q209
48	REF	Refresh cycle	O/P	RAM
47	PENB	Peripheral enable	I/P	RAM (PENBOUT)
46	ACK1	Refresh acknowledge	I/P	Q209
45	REQ1	Refresh cycle request	O/P	Q209
44	ROMEN	Restart ROM enable	I/P	Q209

Side B

Pin	Signal Name	Function	Input or Output	Source/ Destination
76	IL72	P2 level 7 IRQ	I/P	QFC9, SCSI ($\overline{IRQD}$)
75	IL62	P2 level 6 IRQ	I/P	Q133 (ACINT) & Q014 ($\overline{IRQ8}$)
74	IL52	P2 level 5 IRQ	I/P	Graphics ($\overline{PENINT}$)
73	IL42	P2 level 4 IRQ	I/P	Graphics ($\overline{TOUCHINT}$)
72	IL71	P1 level 7 IRQ	I/P	N/C
71	IL61	P1 level 6 IRQ	I/P	N/C
70	IL51	P1 level 5 IRQ	I/P	N/C
69	IL41	P1 level 4 IRQ	I/P	Spare ($\overline{IRQ}$)
68	$\overline{ACINT}$	Keyboard ACIA IRQ	O/P	Q133 (IL62)
67	$\overline{IRQPA}$	User PIA IRQ A	O/P	N/C
66	$\overline{IRQPB}$	User PIA IRQ B	O/P	N/C
65	$\overline{ENL1}$	P1 DMA Enable Next Level	O/P	GIF ($\overline{ETL1}$) & Ch Supp (REFEN)
64	$\overline{FCXX}$	Ports access	O/P	Various
63	$\overline{RTCINT}$	Real time clock intrpt	O/P	Q133 (IL02)
62,61		Current loop TX	O/P	N/C
60	$\overline{FXXX}$	FXXX address range decode	O/P	N/C

Slot 24: Central Processor Module Q209

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
78-77	+5V	Logic power supply	I/P	
76	BE1	P1 "E" signal	O/P	RAM, Spare
75	FIRQ2	P2 Fast IRQ	I/P	N/C
74	FIRQ1	P1 Fast IRQ	I/P	N/C
73	IPI2	P2 Interprocessor Intrpt	O/P	Q133 (IL32)
72	IPI1	P1 Interprocessor Intrpt	O/P	Q133 (IL31)
71	FCXX	Port select	I/P	Q133
70	A/B2	P2 System/User state	O/P	RAM
69	A/B1	P1 System/User state	O/P	RAM
68	ILS2	P2 PICU latch strobe	O/P	Q133
67-64	IA42-IA12	P2 intrpt vector addr.	I/P	Q133
63	IRQ2	P2 interrupt request	I/P	Q133
62	ILS1	P1 PICU latch strobe	O/P	Q133
61-58	IA41-IA11	P1 intrpt vector addr.	I/P	Q133
57	IRQ1	P1 interrupt request	I/P	Q133
56	NMI2	P2 NMI request	I/P	Q133
55	NMI1	P1 NMI request	I/P	Q133
54	RES2	P2 restart	I/P	Q133
53	HLT2	P2 HALT	I/P	Q133
52	RES1	P1 restart	I/P	Q133
51	HLT1	P1 HALT	I/P	Q133
50	W2	P2 in wait state (BA)	O/P	Q133
49	W1	P1 in wait state (BA)	O/P	Q133
48	ACK2	P2 DMA acknowledge	O/P	QFC9, SCSI, WP, GIF
47	REQ2	P2 DMA request	I/P	QFC9, SCSI, WP, GIF
46	ACK1	Refresh cycle grant	O/P	Q133, WP, GIF, CS
45	REQ1	Refresh cycle request	I/P	Q133, WP, GIF
44	ROMEN	Restart ROM enable	O/P	Q133
43	OSC	Master proc. clock	O/P	N/C

Slot 25: Lightpen/Graphics Interface Q219

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
78-77	+5V	Logic power supply	I/P	
68	RINT	Timer IRQ	O/P	Q133 (IL12)
67	PENINT	Light pen hit IRQ	O/P	Q133 (IL52)
66	TOUCHINT	Light pen touch IRQ	O/P	Q133 (IL42)
65	VRAMEN	Video RAM Enable	I/P	RAM Card 0
64	FCXX	Port select	I/P	Q133
63	VRAMSEL	Video Ram Select	O/P	RAM (RAMINH)

Slot 26: Floppy Disc Controller QFC-9

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
78-77	+5V	Logic power supply	I/P	
71	EDL	Enable Daisy Links	O/P	Floppy ETL[1]
70	ETL	Enable This Level	I/P	Floppy & SCSI EDL[1]
69	ENL	Enable Next Level	O/P	GIF (ETL)
68	DMACLM	DMA claim	O/P	RAM (DMAC21)
65	PENB	Peripheral Enable	I/P	RAM card 0
63	IRQD	Floppy Controller IRQ	O/P	Q133 (IL72)
48	ACK2	DMA cycle grant	I/P	Q209
47	RDMA	DMA request	O/P	Q209 (REQ2)

Slot 27: SCSI Controller Q777

Side A

Pin	Signal Name	Function	Input or Output	Source/ Destination
78-77	+5V	Logic power supply	I/P	
72	EDL	Enable Daisy Links	O/P	GIF (ETL)[2]
71	EDL	Enable Daisy Links	O/P	Floppy ETL[1]
70	ETL	Enable This Level	I/P	WP (ENL2)
69	ENL	Enable Next Level	O/P	N/C
68	DMACLM	DMA claim	O/P	RAM (DMAC22)
65	PENB	Peripheral Enable	I/P	RAM card 0
63	IRQD	SCSI Controller IRQ	O/P	Q133 (IL72)
48	ACK2	DMA cycle grant	I/P	Q209
47	RDMA	DMA request	O/P	Q209 (REQ2)

Notes
1. $\overline{EDL}$ outputs of both QFC-9 and Q777 go to $\overline{ETL}$ input of QCF-9. Link optioning on the two controller boards determines which output is actually used. QFC-9 output should only be used if no Q777 is installed. See section X.25.
2. Extra $\overline{EDL}$ output from Q777 bypasses the QFC-9 and goes directly to $\overline{ETL}$ input of GIF card to allow system to be run without QFC-9 card. Currently pin 72A of Q777 is not driven. See section X.25.

Series III CPU Interrupt Level Assignments

Processor 1

Level	Source(s)		o/c	IRQ input
0	MIDIINT: CMI-28 66A	*	y	IL01: DBG 69A
1	TIMINT: Ch. Supp. 76B	*	y	IL11: DBG 70A
2	CHINT: All channels 10B-17B		y	IL21: DBG 71A
3	P1 IPI: CPU 72A	*	y	IL31: DBG 72A
	SMPTE: CMI-28 65A		y	
4	AIC: AIC 46A	*	y	IL41: DBG 69B
5	-			IL51: DBG 70B
6	-			IL61: DBG 71B
7	-			IL71: DBG 72B

Processor 2

Level	Source(s)		o/c	IRQ input
0	RTCINT: DBG 63B		y	IL02: DBG 73A
	PERRINT: Q256 46A		y	
1	LPEN timer: Q219 68A		y	IL12: DBG 74A
2	-			IL22: DBG 75A
3	P2 IPI: CPU 73A		y	IL32: DBG 76A
4	TOUCHINT: Q219 66A		y	IL42: DBG 73B
5	PEN INT: Q219 67A		y	IL52: DBG 74B
6	Keyboard ACIA: DBG 68B	*	y	IL62: DBG 75B
	ACIAs: Q014 57A		y	
7	DISKS: QFC-9, Q077, 63A		y	IL72: DBG 76B

Notes
1) * indicates departure from or addition to Series II assignments
2) "y" in the o/c column indicates IRQ outputs which are open collector and can therefore be wired in paralled with other IRQs on the same level if desired.
3) Source and IRQ input columns give the name of the interrupt followed by the card and edge connector pin number.
4) There is no second P1 interrupt control unit in Series III as there was on the Master Card in Series II. Hence the wiring together of all Channel interrupts.

Series III DMA Daisy Chain Assignments

Modification history
Rev 3: Provision for $\overline{EDL}$ from either Q777 or QFC-9 to QFC -9 $\overline{ETL}$, and provision for $\overline{EDL}$ direct from Q777 to CMI-28.

Arbitration between the various devices which can access the CPU buss via DMA (cycle-stealing Direct Memory Access) is achieved by a daisy chain. At present the Series III daisy chains are prioritised as follows:

P1: 1. System RAM Refresh (Q133) ($\overline{EDL}$ generated internally)
2. General Interface Card (CMI-28)
3. Waveform Processor (CMI-33)

P2: 0. EDL from either Q777 or QFC-9
1. Floppy controller (QFC-9)
2. General Interface Card (CMI-28)
3. Waveform Processor (CMI-33)
4. SCSI Controller (Q777)

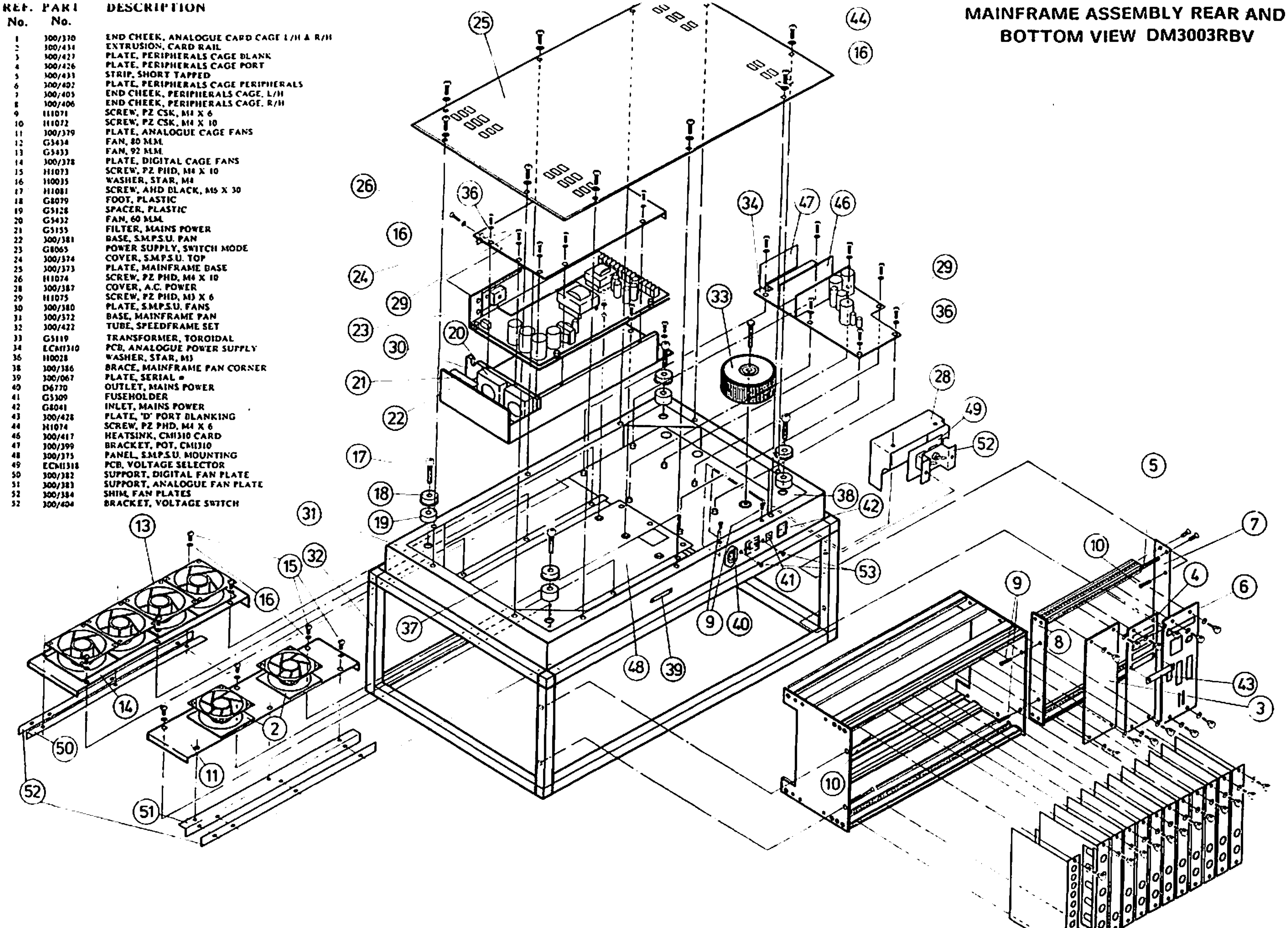
Note that for a given DMA device to work, all higher priority devices must be present in the system or the daisy chain signals for an absent device jumpered on the motherboard. The only exception to this is the bypass link from 72A of the Q777 slot ($\overline{EDL}$) to the GIF $\overline{ETL}$ which allows the system to be run without a QFC-9 simply by linking 71A and 72A on the Q777 board.

The Q256 DMA Channels are assigned as follows:

P1:	Ch 1 SMPTE/MIDI Processor	FC4C
	Ch 2 Waveform Processor	FC4D
	Ch 3 Unused	FC4E
	Ch 4 Unused	FC4F
P2:	Ch 1 Floppy controller	FC44
	Ch 2 SCSI controller	FC45
	Ch 3 Waveform Processor	FC46
	Ch 4 SMPTE/MIDI Processor	FC47

MAINFRAME ASSEMBLY REAR AND BOTTOM VIEW DM3003RBV

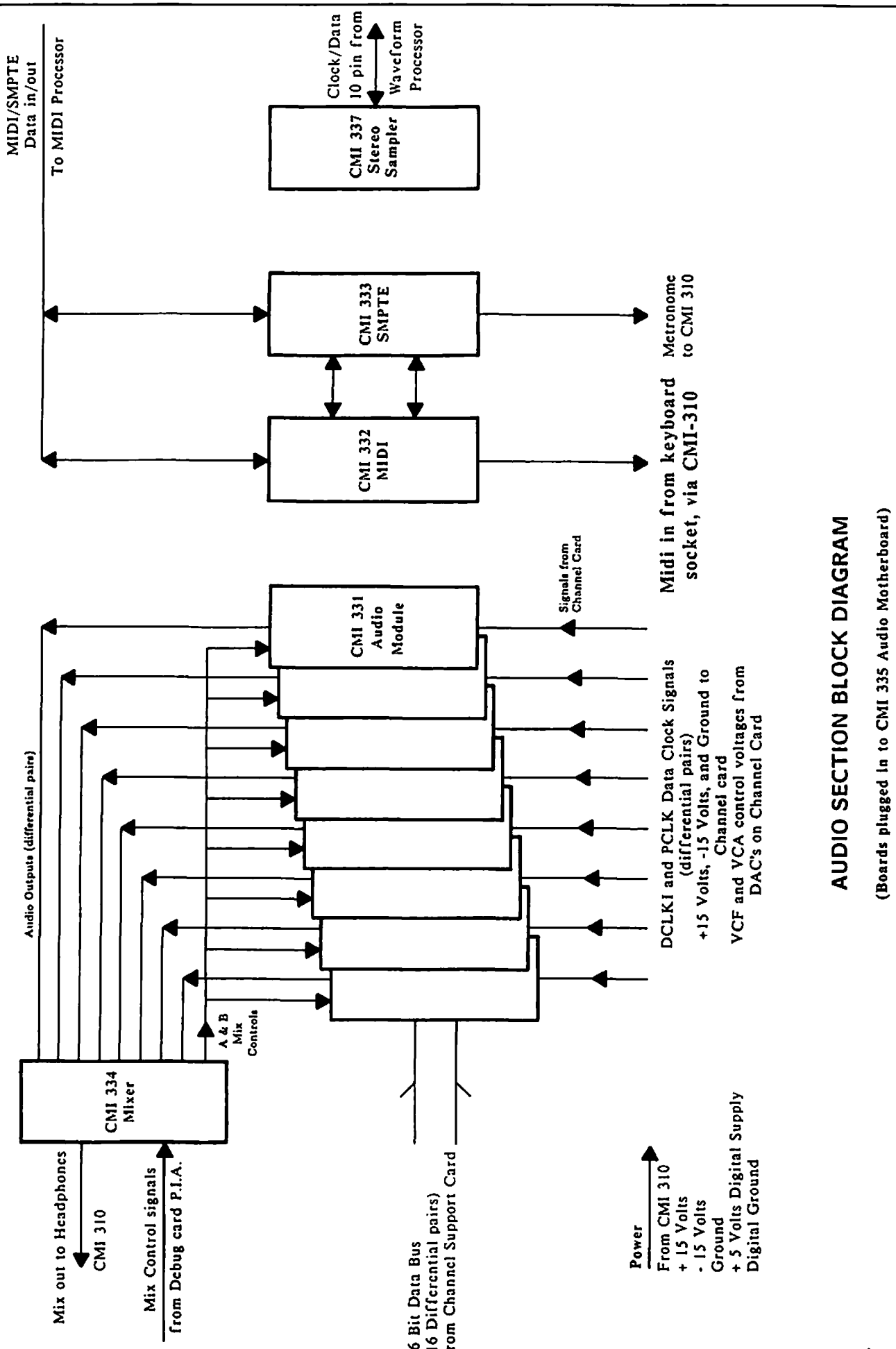
REF. No.	PART No.	DESCRIPTION
1	300/370	END CHEEK, ANALOGUE CARD CAGE L/H & R/H
2	300/434	EXTRUSION, CARD RAIL
3	300/427	PLATE, PERIPHERALS CAGE BLANK
4	300/426	PLATE, PERIPHERALS CAGE PORT
5	300/433	STRIP, SHORT TAPPED
6	300/402	PLATE, PERIPHERALS CAGE PERIPHERALS
7	300/405	END CHEEK, PERIPHERALS CAGE, L/H
8	300/406	END CHEEK, PERIPHERALS CAGE, R/H
9	H1071	SCREW, PZ CSK, M4 X 6
10	H1072	SCREW, PZ CSK, M4 X 10
11	300/379	PLATE, ANALOGUE CAGE FANS
12	G5434	FAN, 80 M.M.
13	G5433	FAN, 92 M.M.
14	300/378	PLATE, DIGITAL CAGE FANS
15	H1073	SCREW, PZ PHD, M4 X 10
16	H0035	WASHER, STAR, M4
17	H1081	SCREW, AHD BLACK, M5 X 30
18	G8079	FOOT, PLASTIC
19	G5128	SPACER, PLASTIC
20	G5432	FAN, 60 M.M.
21	G5155	FILTER, MAINS POWER
22	300/381	BASE, S.M.P.S.U. PAN
23	G8065	POWER SUPPLY, SWITCH MODE
24	300/374	COVER, S.M.P.S.U. TOP
25	300/373	PLATE, MAINFRAME BASE
26	H1074	SCREW, PZ PHD, M4 X 10
28	300/387	COVER, A.C. POWER
29	H1075	SCREW, PZ PHD, M3 X 6
30	300/380	PLATE, S.M.P.S.U. FANS
31	300/372	BASE, MAINFRAME PAN
32	300/422	TUBE, SPEEDFRAME SET
33	G5119	TRANSFORMER, TOROIDAL
34	ECM1310	PCB, ANALOGUE POWER SUPPLY
36	H0028	WASHER, STAR, M3
38	300/386	BRACE, MAINFRAME PAN CORNER
39	300/067	PLATE, SERIAL #
40	D6770	OUTLET, MAINS POWER
41	G5309	FUSEHOLDER
42	G8041	INLET, MAINS POWER
43	300/428	PLATE, 'D' PORT BLANKING
44	H1074	SCREW, PZ PHD, M4 X 6
46	300/417	HEATSINK, CM1310 CARD
47	300/399	BRACKET, POT, CM1310
48	300/375	PANEL, S.M.P.S.U. MOUNTING
49	ECM1318	PCB, VOLTAGE SELECTOR
50	300/382	SUPPORT, DIGITAL FAN PLATE
51	300/383	SUPPORT, ANALOGUE FAN PLATE
52	300/384	SHIM, FAN PLATES
52	300/404	BRACKET, VOLTAGE SWITCH



DIGITAL CARD CAGE - 2.14.19

Audio Card Cage 3

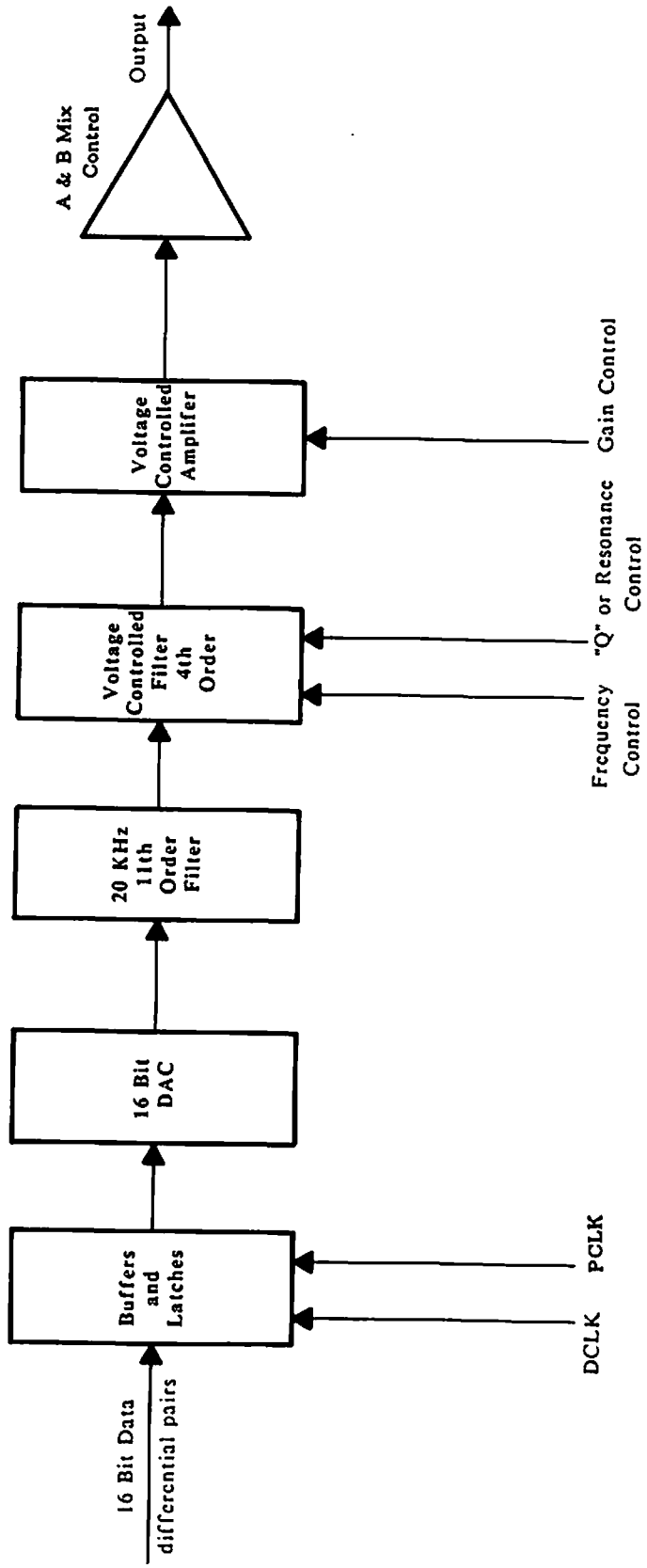
Contents



AUDIO SECTION BLOCK DIAGRAM

(Boards plugged in to CMI 335 Audio Motherboard)

CMI-331 3.2
Audio Module



AUDIO MODULE BLOCK DIAGRAM

1/2 of CMI 331

Terminology
WP: Waveform Processor
WRAM: Waveform Ram (2M-14Mbytes)
CC: Channel Card
CCP: Channel Card Processor
CSC: Channel support Card
CPU: Dual 6809 processor system running OS/9
AM: Audio Module CMI331
AMM: Audio Mixer Module CMI334
AR: Audio Rack Assembly
AMB: Audio MotherBoard CMI335
DAC: Digital to Analog Convertor
WB: Waveform Bus
VCF: Voltage Controlled Filter
Active low signals are preceded by a '/' character.

Introduction

The Audio Module (AM) contains the analog circuitry necessary to filter and control the output from a 16 bit Digital to Analog Convertor (DAC). A typical CMI system contains eight AMs which are located in the rear Audio Rack (AR) slots two through nine. (slots are numbered in Chinese fashion, from right to left looking at the rear of the CMI).
The AM is supplied with all inputs by the Audio MotherBoard (AMB). These include:

Power, which is derived from the CMI310;
Data, which is supplied in differential form from the Channel Support Card (CSC);
Clocks, these are supplied originally from the appropriate Channel Card (CC);
Control Voltages, these also are supplied by the CC;
Control Signals, these are NON-essential controls which may be supplied from the Audio Mixer Module (AMM), see later discussion of control functions for further details.

The only output from the AM is the two audio channels which appear on the rear XLRs and also go to the AMM via the AMB.

Audio Module Structure
(refer Audio Module Block Diagram)

As can be seen from the above referenced drawing, sixteen bit data is buffered and latched (by DCLK) on the AM then clocked (by PCLK) into the DAC. The analog output is then fed through an 11th order 20 kHz low pass filter to remove all traces of sampling noise and aliassing. Note that this filter is ineffective for playback rates below approximately 44 kHz, ie when samples are played down pitch. To provide this function for down pitch sounds a fourth order tracking filter is used. The tracking filter also allows some sound effects to be created by altering the cut-off frequency as well as the resonance or 'Q' of the filter. From the tracking filter the audio signal is passed through a voltage controlled amplifier which allows control of the output level, attack and decay of the sound. Finally the signal passes through a switched mixer (under control of the AMM) which is capable of mixing the two audio signals present on the AM to form

form one. This or the separate unmixed signals are buffered and output to the rear panel sockets where the signal is compatible with a 600 ohm balanced line.

Control Structure

Control signals for the AM are supplied from two sources.

1) The CC: this supplies the clocks for latching data from the common data bus and for clocking data into the DAC at the correct rate. Analog signals to drive the Voltage Controlled Filters (VCF) and Voltage Controlled Amplifiers (VCA) are derived from control voltage DACs on the CC.

2) The AMM: this supplies the digital signal to control the two channel mix function as described in section 1.2

The double latching of data into the DAC is necessary to ensure that minimum pitch jitter occurs when the data is strobed in. The first clock is synchronous with the Waveform Bus, giving 4.8uS resolution, while the second is generated by the CC pitch circuitry whose resolution is approx 60nS.

Data Receivers and Power Supplies

(refer schematic CMI331-00)

Digital differential data coming in from the edge connector are buffered and converted into unipolar signals by four MC3486 differential receivers (ICs A-C 13 through 16).

The power supplies are pre-regulated by the CMI310 card which is connected to the Audio Motherboard. Capacitors C1 through C6 decouple these on the AM.

Reference voltages for the DACS and biasing of the analog switch is provided by Q1 & 2 and IC B9, R4 & R5 protect IC B9 in case of short circuit. Note that there is no connection between the digital and analog grounds on this card, therefore if it is powered up while not plugged into the AMB the result may be damage to the DACs. R1, D1&2 help prevent this by limiting drift between these supplies. The AMB has a link between the grounds on the component side between slots 5 and 6.

Output Buffers and Channel Card Inputs

(refer schematic CMI331-01)

This drawing shows the two output driver ICs and the twenty connections from the CC. Note the 6 resistors (100R) and capacitors (1nF) for filtering the analog controls from the CC, these replace six chokes and 0.1uF capacitors from the original rev 1.0 card. If these capacitors are still 0.1uF the Field Change Note upgrade to replace these must be performed since filter oscillation and very high distortion may occur.

Audio output signals are buffered by ICs L2 and L5 (5532) which are capable of directly driving a 600 ohm balanced line. The 33 ohm standoff resistors provide higher tolerance of capacitive loads.

Control Voltages

(refer schematic CMI331-02)

The two filter control sections are shown on the left of the page while on the right the VCA controls are shown.

VCA gain controls are calibrated by RV9 and 10 while RV13 and 14 provide calibration for the +4dBm output level. R121 and R123 are thermistors to compensate for thermal drift of the VCAs. C52 through C55 limit bandwidth and assist with stability as the op amp has a gain of less than 1.

VCF control for the 'A' half of the AM centres around ICs J1,2 and L1. Working backwards, J1,2 pins 1 and 7 supply the control voltages for the four VCAs within the VCF and RN3 (100 ohm) provides isolation between VCA control inputs. R62 and R63 give this stage (pins 1,2 & 3) a gain of 1 while RN4 (1Meg ohm pins 3 and 4) gives a dc offset of approx 460mV so that the filter is set to 20Hz 3dB point with a control voltage of 0V applied.

IC J1,2 (pins 12,13 & 14) has a gain of 6 with R64 giving thermal compenstaion of cut-off frequency. R34 and 43 divide the 0 to 10volt control down to approx 0 to 60 mV.

IC J1,2 (pins 8,9 & 10) is used to cause a variation in control voltage between the 1st and 2nd VCA in each filter stage. This causes an increase in resonance while not changing the passband cut-off point.

The 50k ohm trimmer is used to set the resonance so that flat response is obtained when the control input is set to 0V.

Operation of both halves of the control circuits is identical.

Digital to Analog Convertor and Voltage Controlled Amplifier
(refer schematic CMI331-03 and 04)

Starting from the left, the sixteen bit data is latched from the WB by DCLK into the first set of latches, PCLK clocks the data into the DAC through the second set of latches, the Two current outputs are added and fed via a buffer stage to the 11th order elliptical filter which attenuates unwanted sampling noise to -85dB.

Data is clocked into the first stage of latches (ICs D13,14 &D15,17) by /DCLK through IC A5,6. The pitch clock (/PCLK) (IC A5,6) clocks the data into the second latch stage (ICs E13,14 and E15,17).

The DAC converts this data into 2 current outputs which are converted by IC G12,13 into a voltage of 14Vpeak to peak max at pin 1 of IC G12,13. IC I12 reduces this signal and provides some filtering before feeding into the 11th order elliptical filter.

The VCA is a current mode device and should be fed with a signal with 0 dc offset via C45 and R111. Distortion is trimmed with RV12.

The B Half of the circuitry is identical in operation to that just described.

Voltage Controlled Filter
(refer schematic CMI331-05, 06, 07, 08)

The VCF is a state variable type utilising two voltage controlled integrators per stage with feedback. Two stages are used to give a 4th order characteristic. One stage only will be described as all four stages are identical.

C51 isolates any DC component, L15 pin 1 is a high pass output. IC L15 pin 7 is a band pass output and J11 pin 1 is the low pass output.

R115 and IC J15 (VCA) operate as a variable feed in resistor to the integrator formed by IC L15 (pins 5,6 & 7) and C47. Similarly, R114, IC K15, IC J11 (pins 1,2 & 3) and C43 operate as a voltage controlled integrator.

RV7 and RV8 adjust the symmetry of each VCA for minimal distortion but have only a minor effect at low frequencies, see the diagnostic section for the correct adjustment procedure.

Each op amp stage of the VCF feeds back into a common node at pin 2 of IC L15. Varying the control voltages to the two integrators so their frequency response changes alters the output frequency response.

Mixing and VCA Buffers
(ref schematic CMI331-09)

The VCAs are a current mode device, therefore their outputs may be mixed before buffering. For this purpose a 4053 analog switch is used (note that its power supply is not +-V but +vref and -vref)

The current outputs from the VCAs are fed into analog switch IC M9,10 (4053) this operates in current mode to switch the signals to IC L8 (5532) or to mix them into pin 15 of IC M9,10. C57 limits the bandwidth of the mixed signal to reduce noise power and assist with the stability when mixing is switched out.

When mixing is not used the currents from the VCAs pass directly through IC M9,10 and are converted back to voltage signals by IC L8. For mixing the A and B signals the currents from the VCAs are switched into a summing node at pin 3 and 13 of IC M9,10 where the signal is converted back to voltage by R133 and again to current by R107 and R108. R134 and R135 help improve stability and reduce noise.

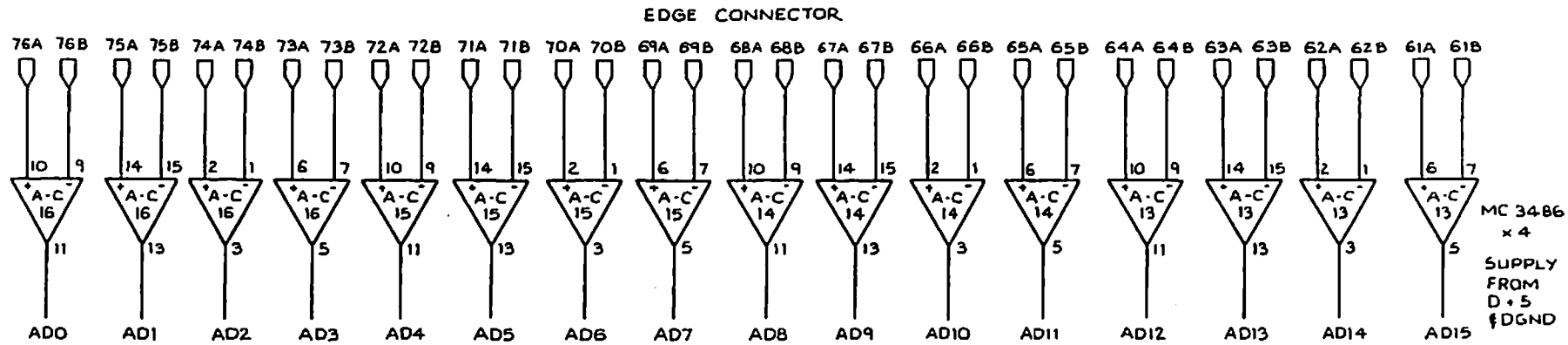
R136 ensures that mixing is off when no control is present.

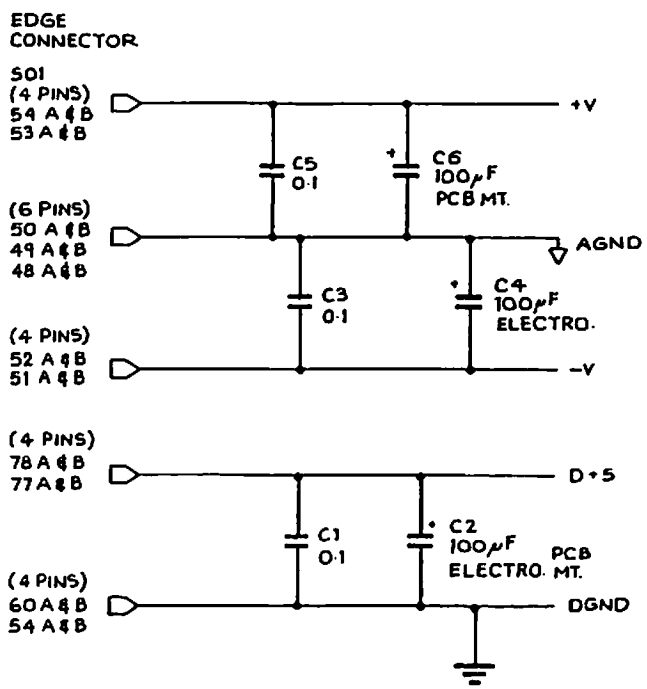
IC L8 converts the current signals to voltage and drives the output buffers.

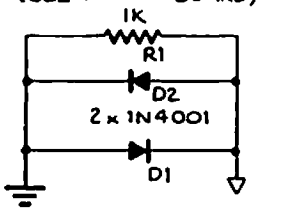
Calibration Procedure

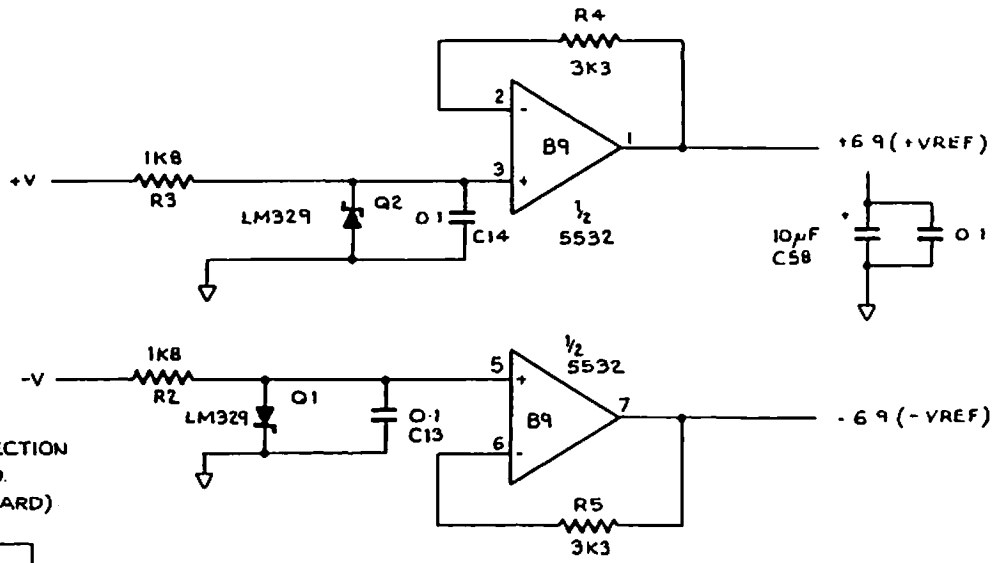
The calibration procedure is fully described in the diagnostic section of this manual.

This procedure requires a minimum of a 20MHz bandwidth CRO, an accurate noise,distortion and power meter and a digital multimeter in addition to other normal tools.



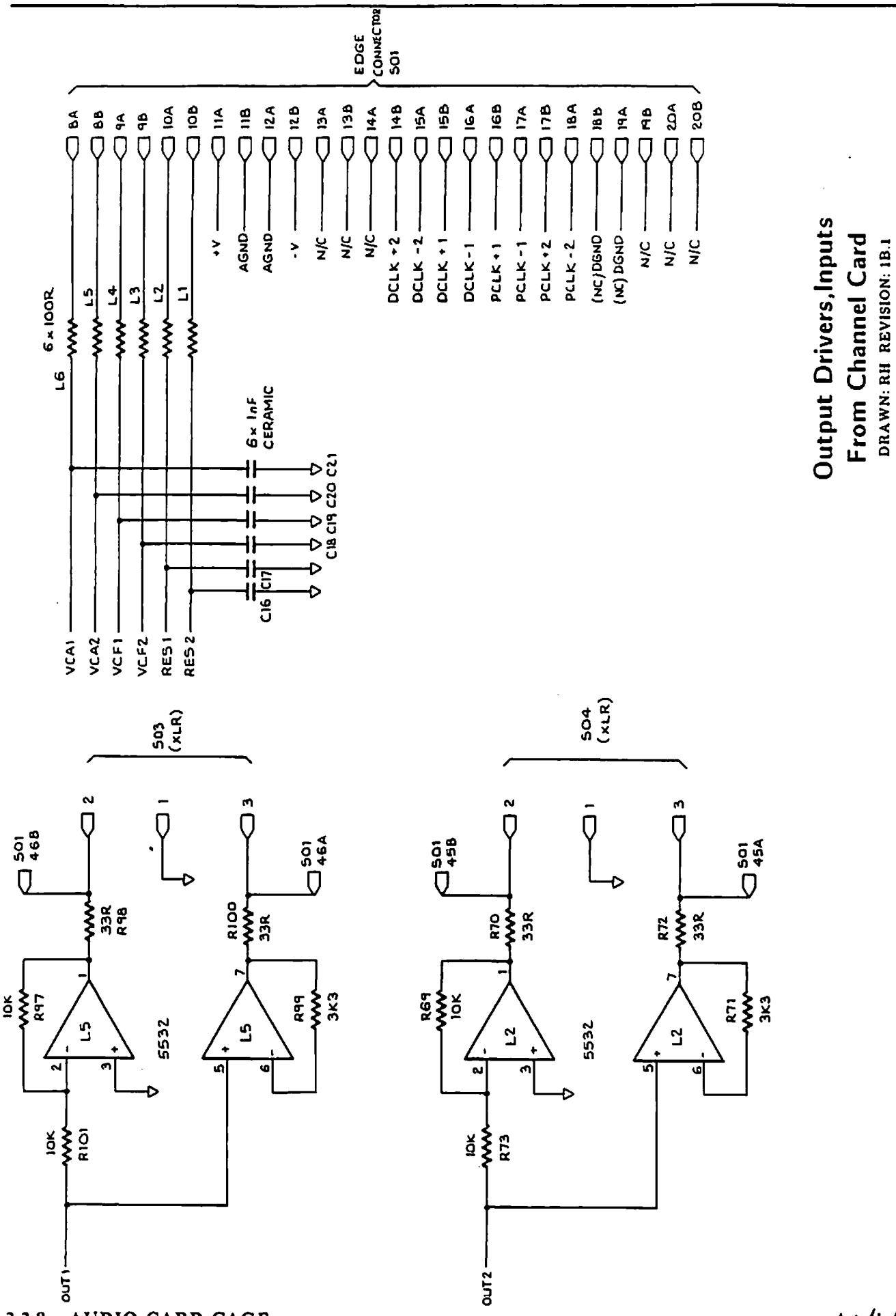


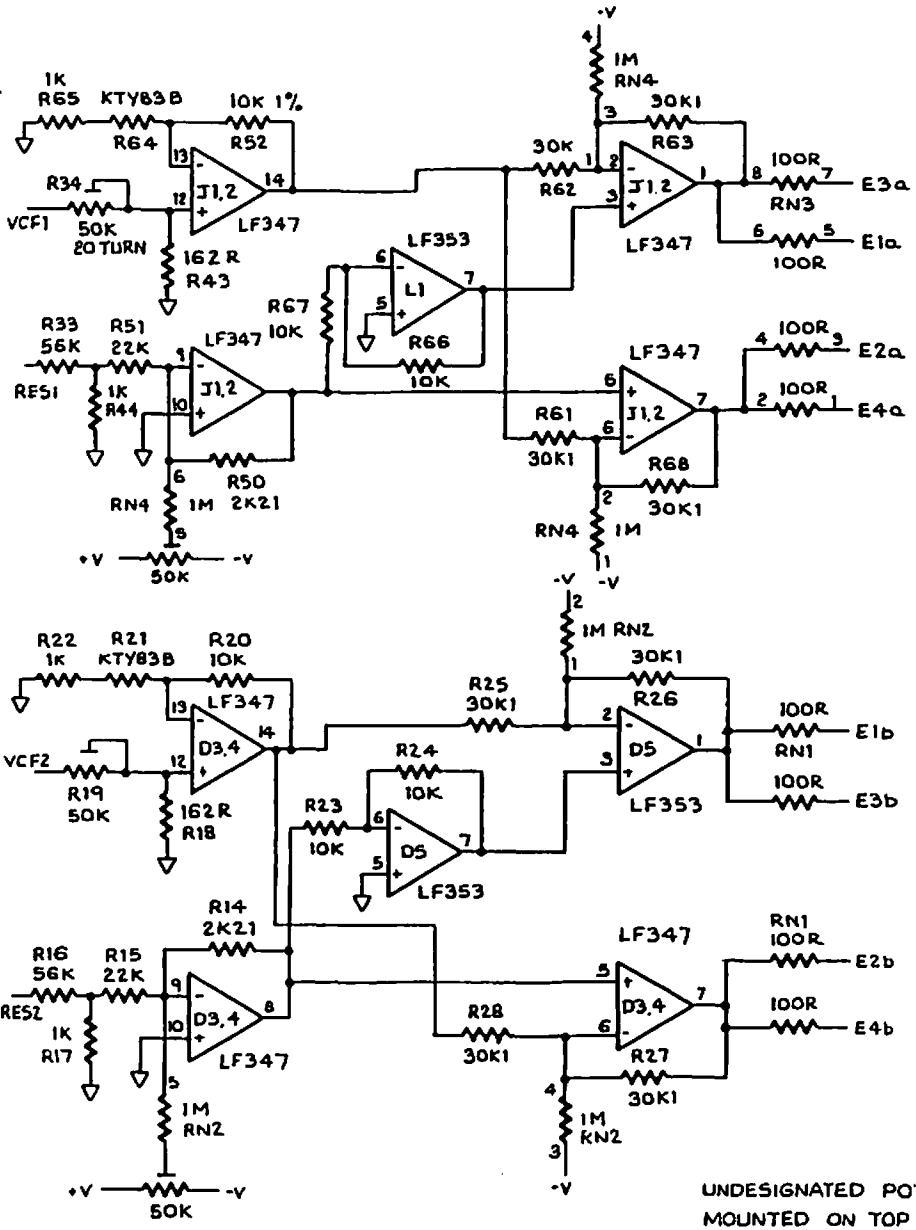




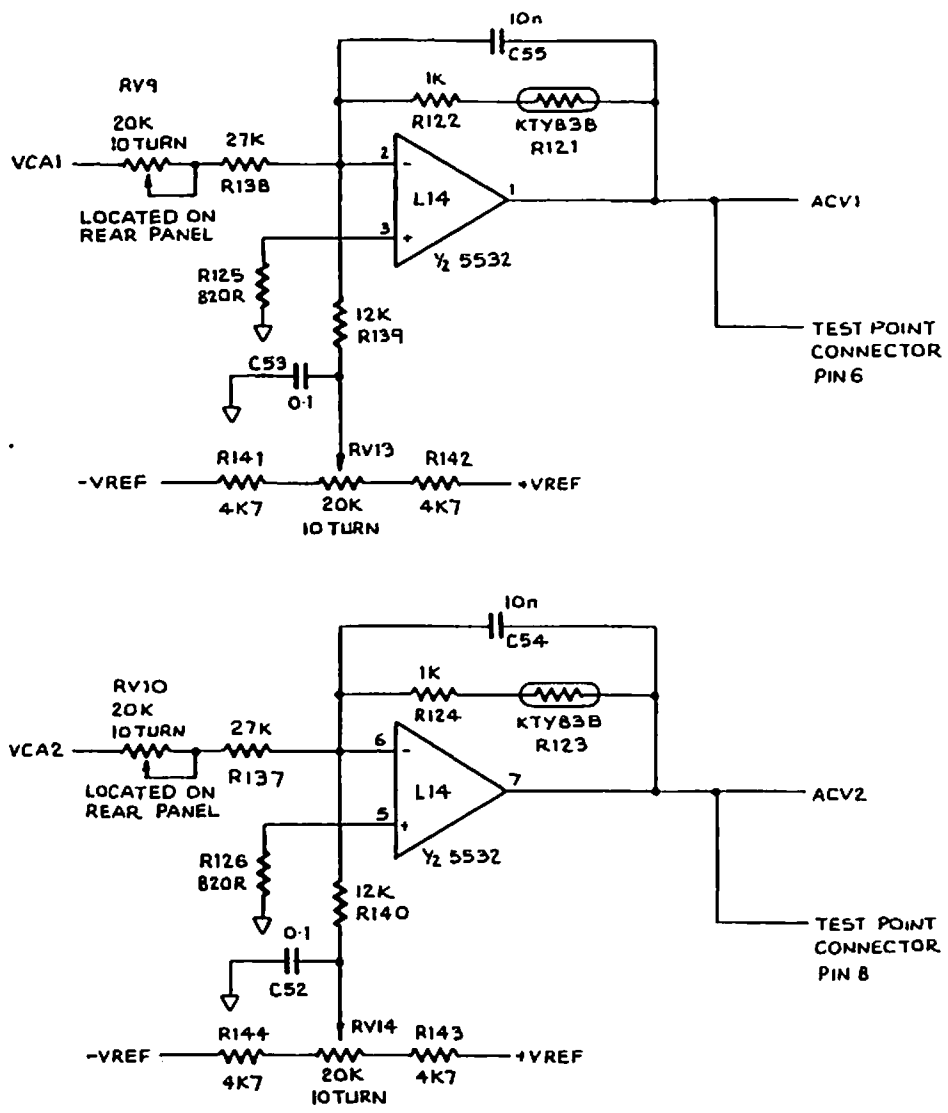
Data Buffers, Power Supplies and Voltage References

DRAWN: RH REVISION: 1B.1


EDGE CONNECTOR SO1
8A 8B 9A 9B 10A 10B 11A 11B 12A 12B 13A 13B 14A 14B 15A 15B 16A 16B 17A 17B 18A 18B 19A 19B 20A 20B
+V AGND AGND -V N/C N/C N/C DCLK +2 DCLK -2 DCLK +1 DCLK -1 PCLK +1 PCLK -1 PCLK +2 PCLK -2 (NC)DGND (NC)DGND N/C N/C N/C
6 x 100R
L6 L5 L4 L3 L2 L1
6 x 1nF CERAMIC
C16 C17 C18 C19 C20 C21
VCA1 VCA2 VCF1 VCF2 RES1 RES2
Output Drivers, Inputs From Channel Card
DRAWN: RH REVISION: 1B.1
SO3 (XLR)
SO4 (XLR)
SO1 46B
SO1 46A
SO1 45B
SO1 45A
33R R98
33R R100
33R R70
33R R72
10K R97
10K R101
10K R69
10K R73
3K3 R99
3K3 R71
L5
L2
5532
OUT1
OUT2

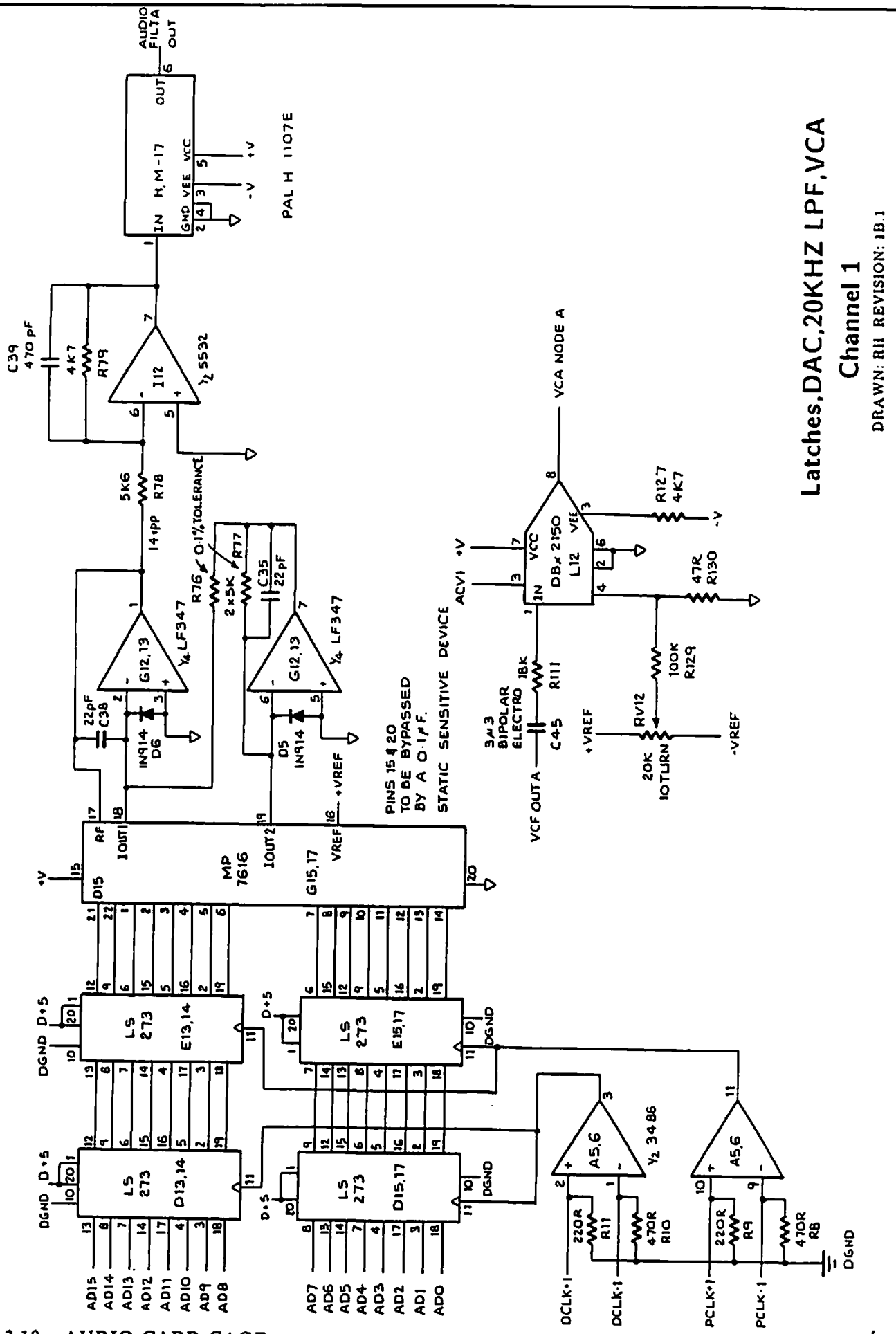


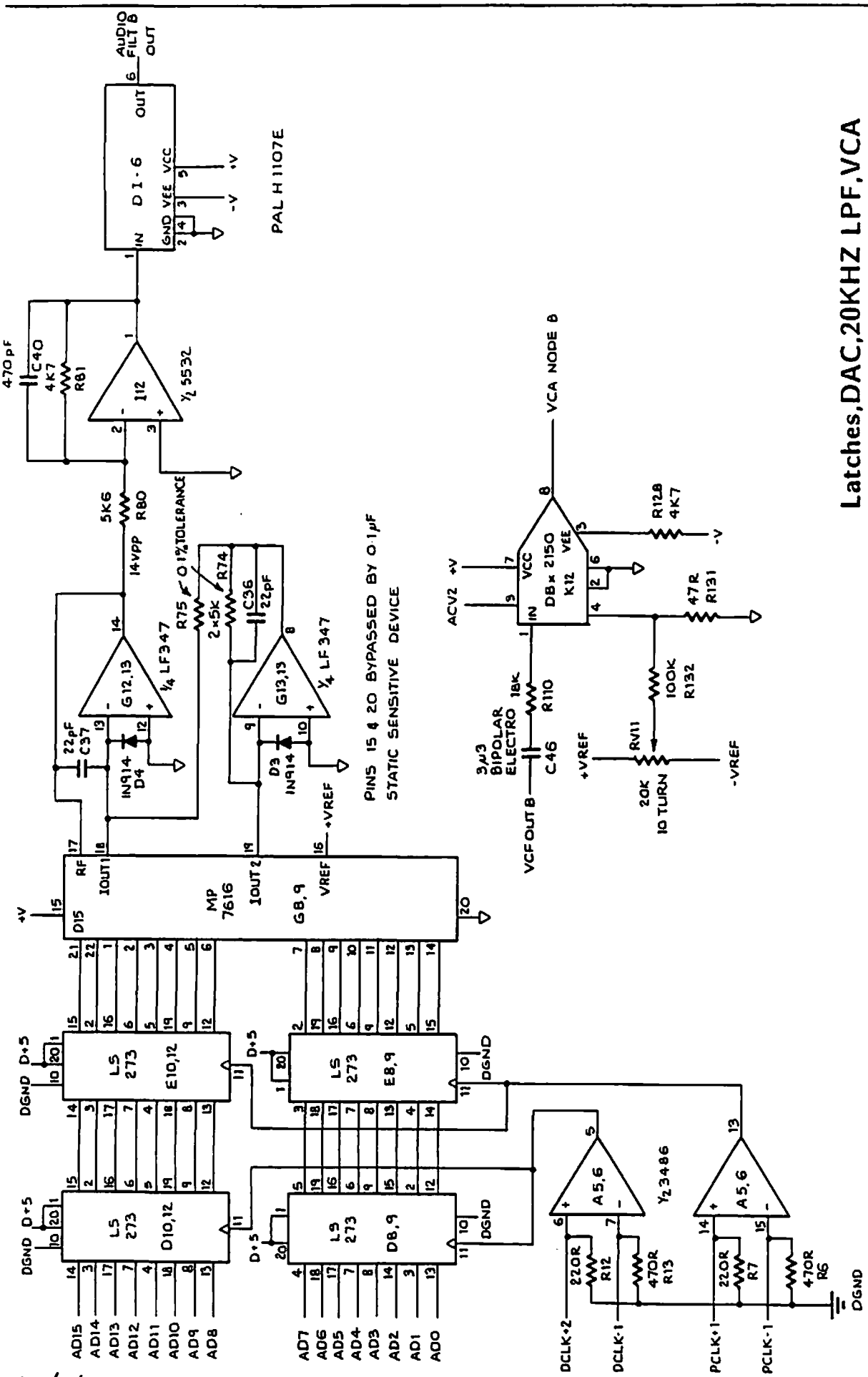
UNDESIGNATED POTS ARE MOUNTED ON TOP OF IC S J1, J2, F1, D3, D4 & D5

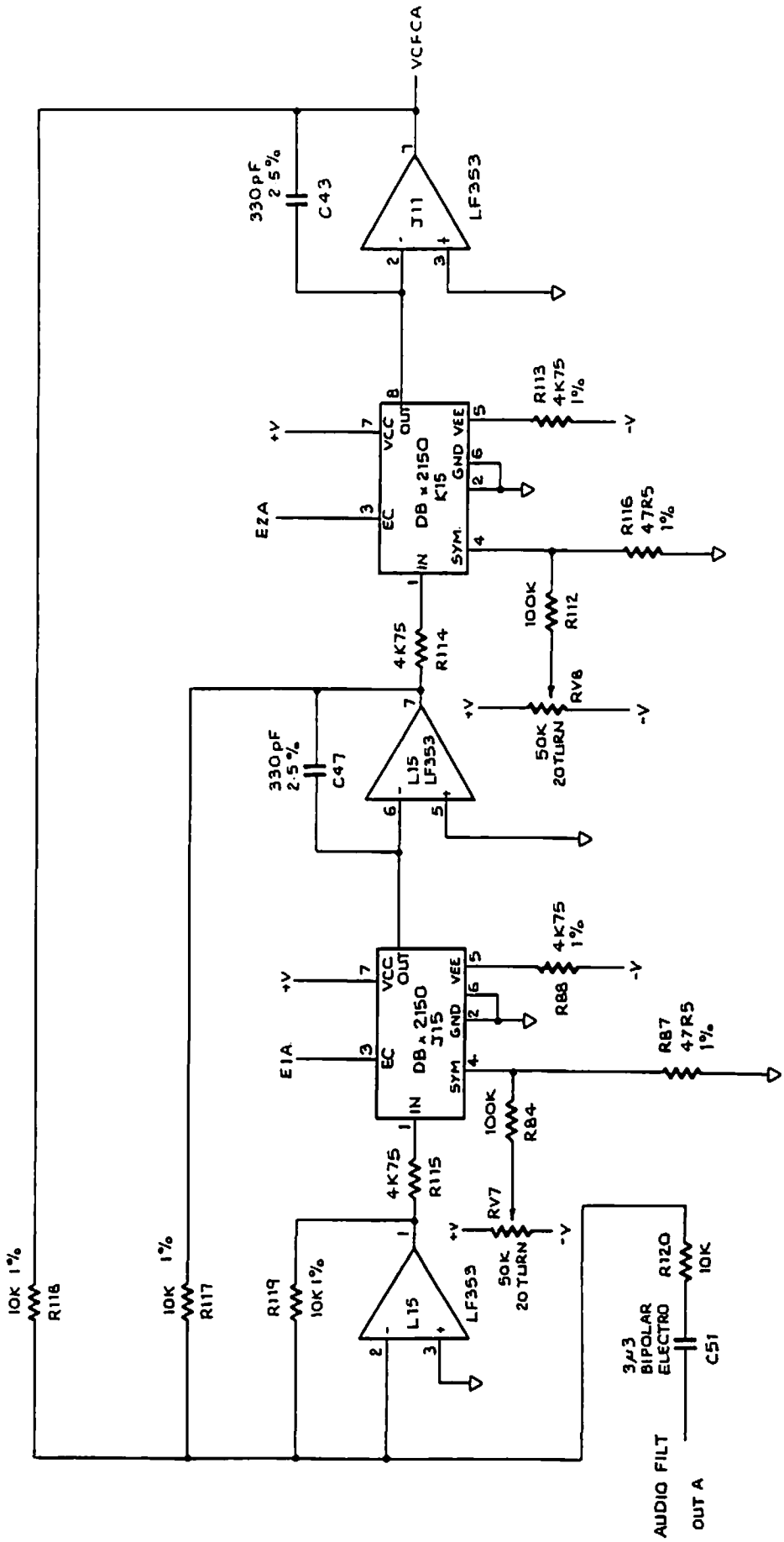


Voltage Controls For VCA'S &VCF'S

DRAWN: RH REVISION: 1B.1

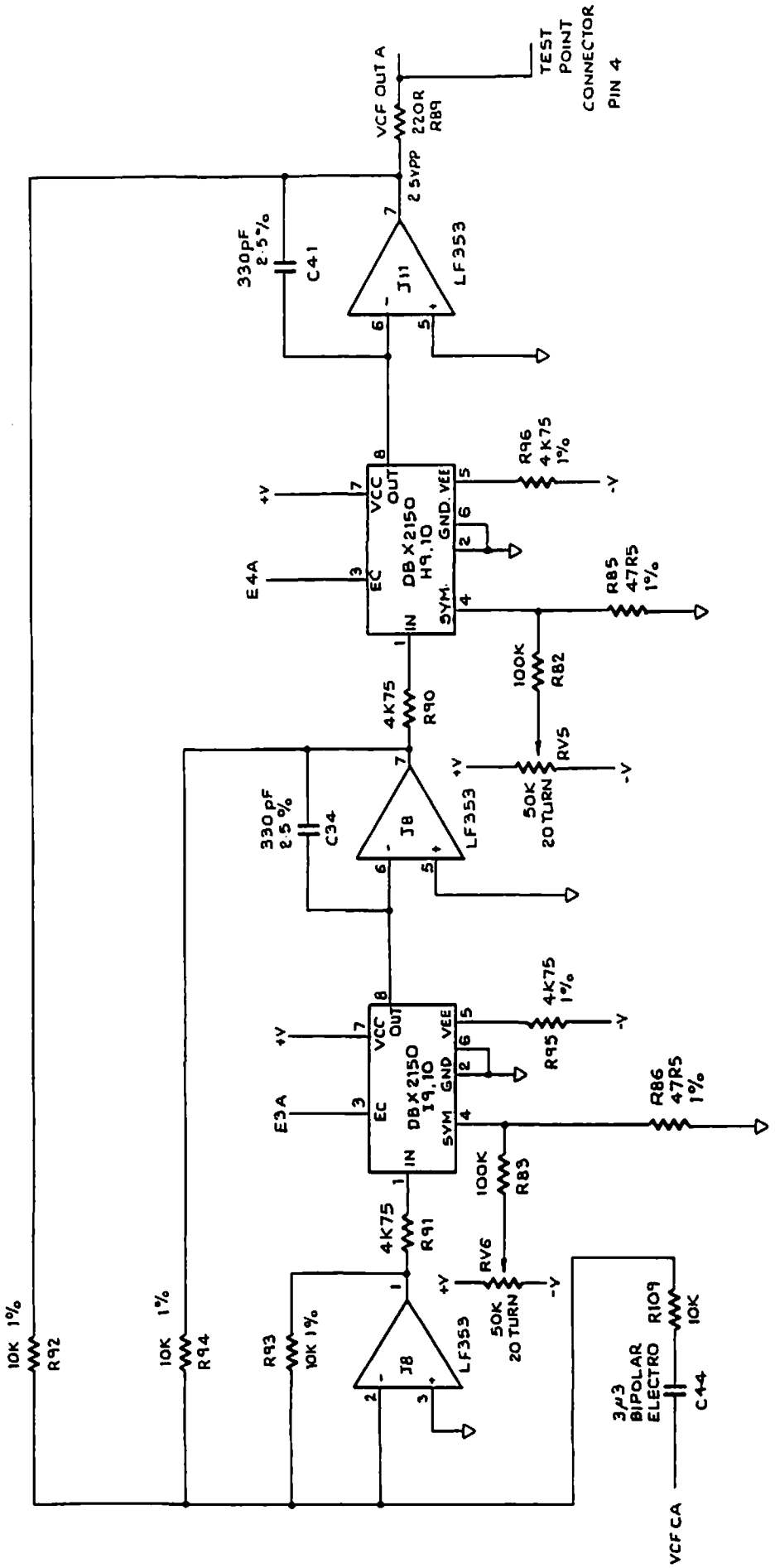

Latches,DAC,20KHZ LPF,VCA
Channel 1
DRAWN: RH REVISION: 1B.1
PAL H 1107E
AUDIO FILTA OUT
H.M-17
I12
½ 5532
C39 470 pF
4K7 R79
5K6 R78
14pp
R76 0.1% TOLERANCE
2x5K R77
C35 22pF
G12.13 ¼ LF347
22pF C38
IN914 D6
D5 IN914
PINS 15 & 20 TO BE BYPASSED BY A 0.1µF.
STATIC SENSITIVE DEVICE
MP 7616
RF IOUT1 IOUT2 VREF
D15 G15,17
+VREF
VCA NODE A
DBx 2150 L12
ACV1
VCF OUT A
3µ3 BIPOLAR ELECTRO C45
18k R111
100K R129
47R R130
R127 4K7
RV12 20K 10TURN
LS 273 E13,14
LS 273 E15,17
LS 273 D13,14
LS 273 D15,17
DGND D+5
A5,6 ½ 3486
220R R11
470R R10
220R R9
470R R8
AD15 AD14 AD13 AD12 AD11 AD10 AD9 AD8
AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0
DCLK-1 PCLK-1


Latches,DAC,20KHZ LPF,VCA
Channel 2 (B)
DRAWN: RH REVISION: 1B.1
PAL H 1107E
PINS 15 & 20 BYPASSED BY 0.1µF
STATIC SENSITIVE DEVICE
AUDIO FILT B OUT
D1-6
112
½ 5532
470pF C40
4K7 R81
5K6 R80
14VPP
R75 ≠ 0.1% TOLERANCE
2×5K R74
C36 22pF
¼ LF347
G12,13
G13,13
22pF C37
IN914 D4
D3 IN914
MP 7616
IOUT1
IOUT2
VREF
RF
D15
G8,9
+V
+VREF
VCA NODE B
DB× 2150 K12
VCC
VEE
IN
ACV2 +V
R128 4K7
47R R131
100K R132
3µ3 BIPOLAR ELECTRO C46
18K R110
VCF OUT B
RV11 20K 10 TURN
-VREF
LS 273 E10,12
LS 273 E8,9
LS 273 D10,12
LS 273 D8,9
DGND D+5
½ 3486
A5,6
22OR R12
47OR R13
22OR R7
47OR R6
AD15 AD14 AD13 AD12 AD11 AD10 AD9 AD8
AD7 AD6 AD5 AD4 AD3 AD2 AD1 AD0
DCLK+2 DCLK-1 PCLK+1 PCLK-1



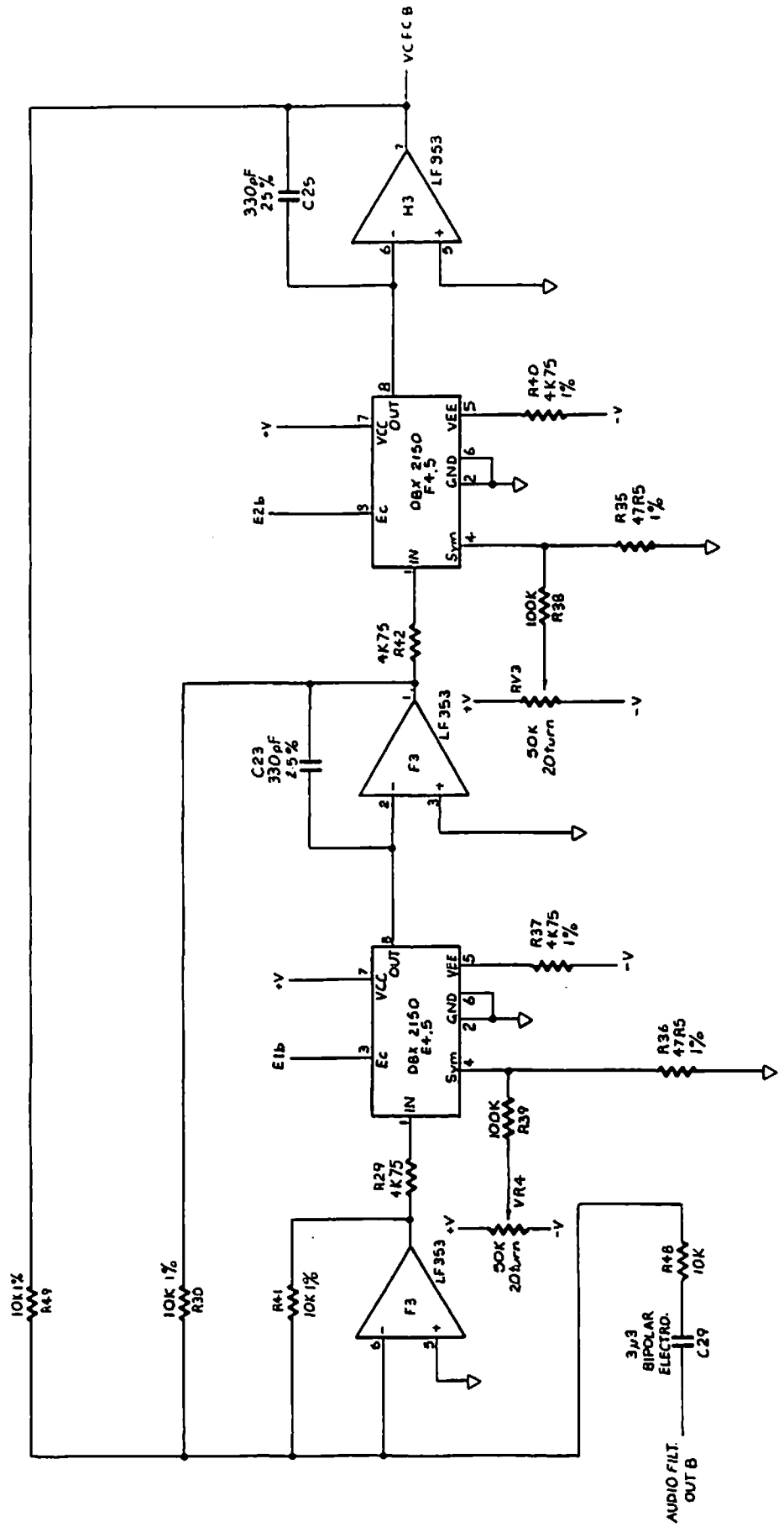
Voltage Controlled Filter Channel A Stage 1

DRAWN: RH REVISION: 1B.1



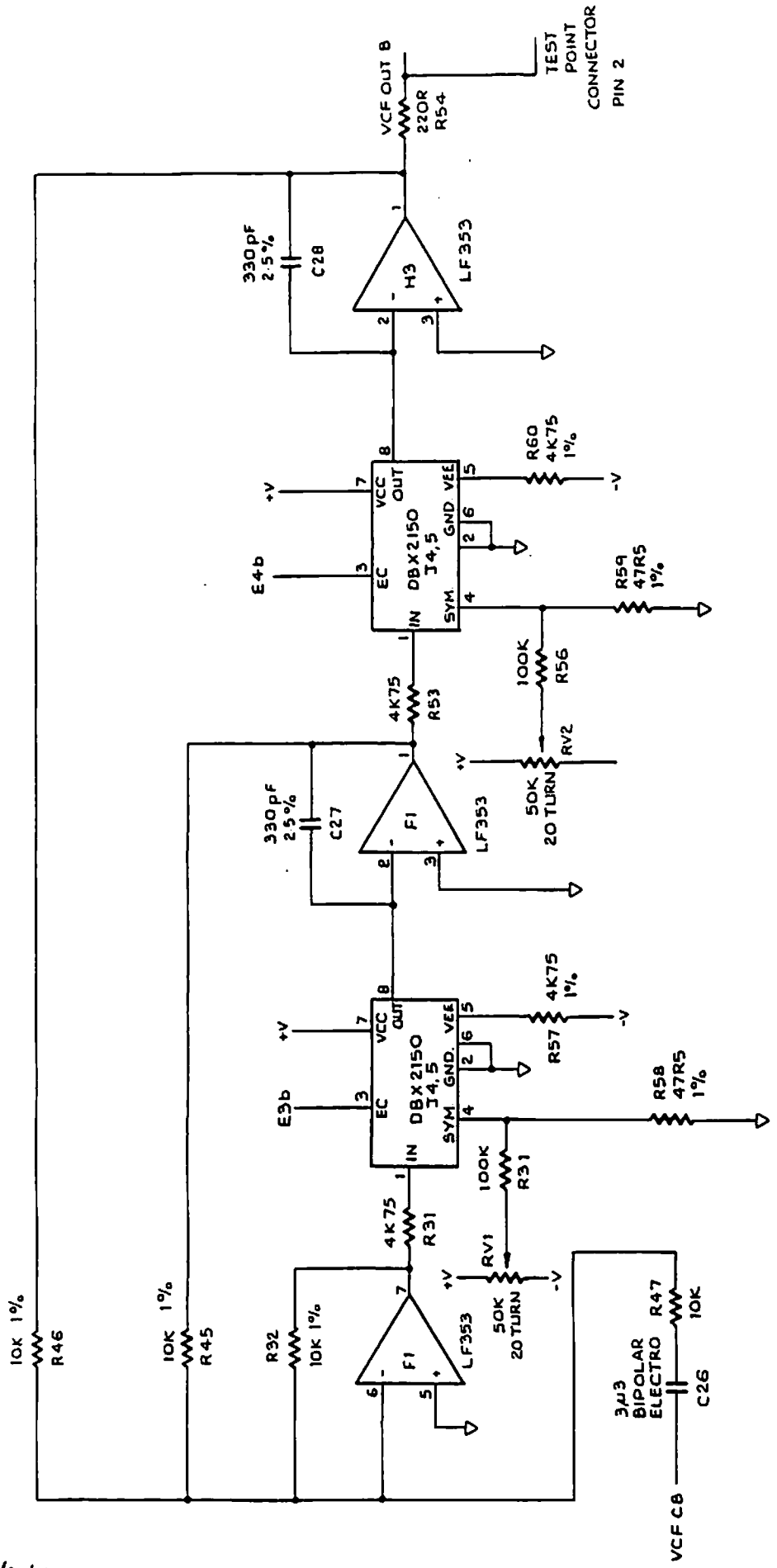
Voltage Controlled Filter
Channel A Stage 2

DRAWN: RH REVISION: 1B.1



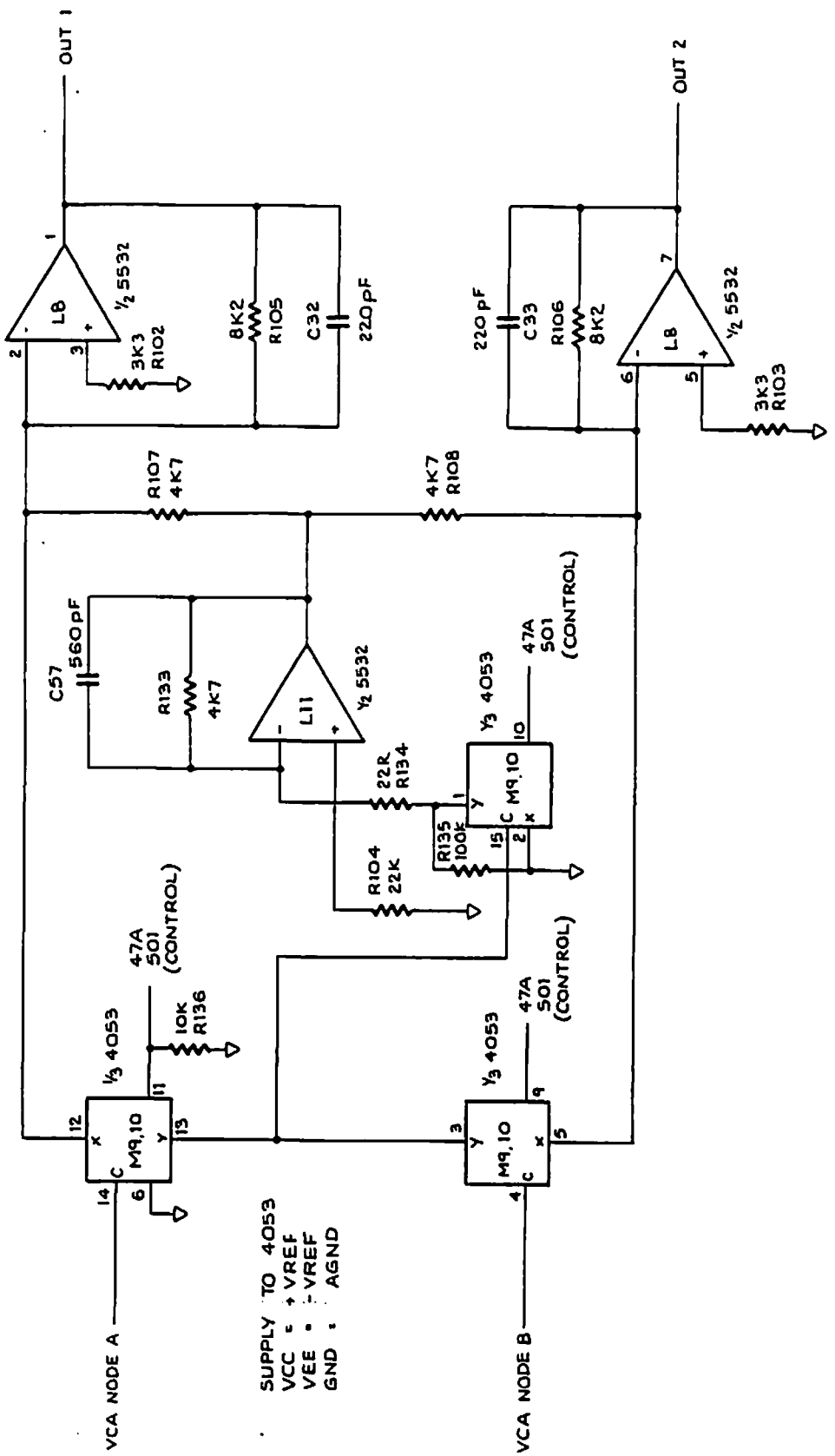
Voltage Controlled Filter Channel B Stage 1

DRAWN: RH REVISION: 1B.1



Voltage Controlled Filter
Channel B Stage 2

DRAWN: RH REVISION: 1B.1



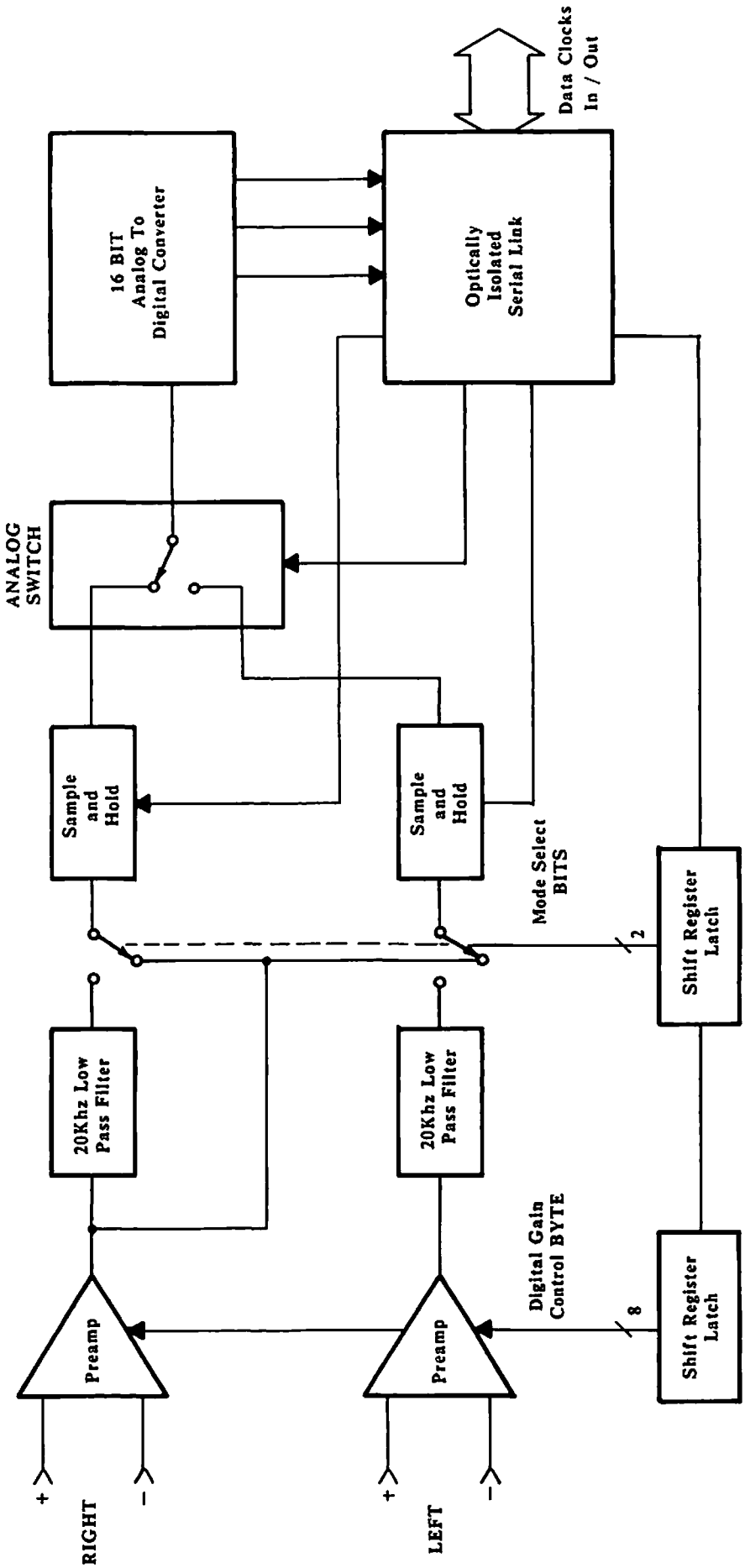
Panning Facility

DRAWN: RH REVISION: 1B.1

CMI-337 3.3

Stereo ADC

Stereo ADC Block Diagram



Terminology
WP: Waveform Processor
CC: Channel Card
ADC: Analog to Digital Converter
TH: Track and Hold
MDAC: Multiplying Digital to Analog Converter
LPF: lowpass filter

Introduction

The Stereo ADC is the means by which real sounds are digitized to 16 bits into the Series III CMI. It connects via an optically isolated serial link to the WP and is situated in the Audio rack. The sampling rate is determined by the pitch set on side A of CC 1. The ADC operates in mono, stereo and in a filter bypassed high speed mono mode. Mono and stereo allow sample rates to approximately 50k and the unfiltered mode allow rates to approximately 100k. The mode of operation and the input level is controlled by a 16 bit word in the WP address space. Data from the ADC is read from a 16 bit location in the WP address space.

Analog signal chain

The stereo balanced inputs are converted to single-sided signals in the first gain stage and then fed through MDACs, for software gain control of the sampling level. For best performance, the user should always select the maximum level which will not cause clipping. Anti-aliasing low-pass filters are fixed at 20kHz. The stereo channels are then fed through two Track and Holds running out of phase with each other. That is, when the right is in hold mode, the left is in track mode. The output of the track and hold that is in hold mode at any one time is feed to the input of the ADC via the an analog switch. The switching between track-and-holds occurs at the sampling frequency.

Mode selection

A 16 bit control word on the CMI-337 is used to control the selection of mono and stereo sampling, the input level, and the bypassing of the fixed lowpass filters. The mono/stereo mode bit is also used to synchronize the receiving of the right and left samples by the WP software when in stereo mode. This occurs before a sample actually starts being read by the WP. It is occurring continuously when the level meters on the sampling page are being processed.

Clocks and control

The ADC needs a 1.78MHz clock to operate. The other timing signals come into and out of the CMI-337 via the optically isolated interface. This interface consists of current drivers on output signals and opto isolators on input signals. The ADC chip itself generates the clock signals to load the 16 bit data into and out of the WP. These clocks occur in 16-pulse bursts when the ADC is given a start conversion command. The start conversion command is generated from the rising edges of the clock originating from Channel Card 1, side A which is transmitted to the CMI-337 through the optical link from the Waveform Processor.

Input preamps and low pass filters
(ref schematic CMI-337-01)

Low noise NE5534 amplifiers are used to receive the balanced inputs and make them single ended. These input amplifiers have a gain of 3.8 and 18 volt supply rails to allow them to handle inputs with levels ranging from +4dbm to -10dbm.

The input stages then drive the reference inputs of AD7524 MDACs via large capacitors to remove and DC offset. The MDACs allow software control of input signal attenuation by multiplying the input signal by the number from the WP. The Digital attenuation value comes from the most significant 8 bits of the CMI-337's control word at A7 via isolating RC networks.

The output of the right channel MDAC goes to both a fixed low pass filter and a resistive attenuator. One of the two outputs are selected by a relay whose function is to allow the fixed filter to be bypassed in high speed mono sampling mode.

The output of the left channel MDAC goes only to a fixed lowpass filter.

The fixed 11th order lowpass filters have a corner frequency of 20khz and passband gain of one half. To get the best noise and distortion performance, the input levels should be about 3 volts peak to peak.

In high speed mono sampling, the output of the right channel's MDAC is fed into both channels of track-and-hold and the fixed low pass filter is bypassed.

The last stage in each channel is an amplifier with a gain of 4.7 to bring the level up to that needed by the ADC of 10 volts peak to peak. This stage also has a DC servo feeding from the outputs of the track and holds to null DC offsets in each channel. The DC servo is basically an inverting integrator with a long time constant that generates a DC offset error voltage that is fed back into the gain stage. The cancelling of DC between the two channels is important in the high speed mono sampling mode. In this mode, alternate samples come from alternate track-and-holds, so that any DC offset between the two will result in severe distortion as every second sample would be displaced from its correct level by the offset.

Track and Holds
(refer schematic CMI-337-02)

The track-and-holds are constructed around high current drive voltage followers, an analog switch, a polystyrene capacitor and a JFET input opamp buffer. When the switch is closed, the voltage on the 2n2 capacitor follows the input signal. When the switch is open the voltage on the capacitor is held.

The hold control inputs are clocked on opposite phases of the clock from the WP, divided by two by D10 so that when the right track-and-hold is holding, the left TH is tracking.

Channel Multiplexing and the Analog to Digital Converter

(refer schematic CMI-337-03)

The same signal that switches the track-and-holds also switches the analog switch at F8. The track-and-holds are continuously multiplexed into the JFET buffer at H10. The Track and Hold that is currently in Hold mode is always multiplexed to the input of the ADC.

Even in Mono mode, when only the right channel input is converted, the multiplexing still takes place.

The ADC is the Matsushita MA6206. On receipt of a start pulse, the ADC does a succesive approximation conversion on its input signal. This results in 16 clock pulses and data bits being generated and shifted out of the CMI-337 to the WP.

The power supply for the CMI-337 is completely isolated from the rest of the CMI with only a ground tie resistor between the two . On card regulators couple the three DC supplies to the on board components.

Control

(refer schematic CMI-337-04)

The same clock used to clock the ADC data into the WP is used to shift data from the WP into the control latch on the CMI-337. The ADC clock output consists of 15 normal pulses followed by one truncated pulse of only 100nS, for each conversion cycle. This last pulse is too short to send over the opto-isolated link so the ADC clock out is used to fire a monostable, half of C11, whose output is cabled to the WP as the data clock.

This clocking occurs each time a conversion is done. The data from the WP is latched into output latches in the 74LS595s by the end of conversion signal generated by the ADC, (the same occurs at the WP with received ADC data).

The most significant 8 bits of the control latch are used to control the Audio level. The next two bits are used to select filter bypass/mono mode and mono/stereo mode.

The conversion rate clock and the serial data from the WP are received at B10 by a high speed optoisolator. The data goes into the data input of the 16 bit control register, at A7 and B9.

The Start Conversion clock goes into a divide-by-two stage at D10 and a multiplexer formed from 3 gates of the NAND package at D11. The multiplexer is used to send either the direct conversion clock or the divided by two conversion clock to the ADC start pulse generator, formed around E11 and F11. This generates a start conversion pulse on every rising edge presented to it.

The half of D10 that feeds pin 5 of D11 is used to synchronize mode changing with conversions.

In mono mode, the divided by two signal is selected to make the ADC do a conversion only when the right channel input is present at the ADC input.

In stereo mode the direct clock is used, so that the right and left channel inputs are converted in turn.

In high speed mono, both clocks are used as in stereo but the right input is connected to both track-and-holds.

Remember that the conversion clock is a square wave that drives the track-and-holds in anti phase, and selects the Sample and Hold that is currently in hold mode to the ADC. The ADC always has a signal at its input that is being held by one of the Sample and Holds.

Before the selected conversion clock is fed to the start circuit, it is fed into a delay to allow the analog circuits to settle before starting the conversion. This is most significant on low level signals, as it is the most significant bit that will be corrupted on conversion. The result being something resembling cross over distortion. The effect is most significant on small amplitude signals, as the overshoot generated when the analog multiplexer swithces is small, but of greater amplitude than the signal being converted.

The master oscillator is made around inverters G11 and the 3.57MHz crystal. This is divided by two to clock the MA6206 and the start pulse generator.

The TTL control signals from the MA6206 are converted to currents via the diode and resistor networks coupling them to the edge connector and socket.

Control Word Bit Definitions

The control word has the following bit definitions

Bit	Function
15	most significant bit of preamp attenuation
14	
13	
12	
11	
10	
9	
8	least significant bit of preamp attenuation
7	bypass Low Pass Filters
6	sample mode control
5	n/c
4	n/c
3	n/c
2	n/c
1	n/c
0	n/c

Bit 7 Bypass Low pass filters

This bit does 2 things. It bypasses the 20 KHz LPF and connects the output of the right channel preamp into both sample and holds, to allow mono sampling at greater than 50 KHz. Sampling at greater than 50KHz in mono can be done without switching this bit, but at the expense of reduced capture time for the track-and-hold and hence increased distortion.

Bit 6 Sample Control

This bit selects whether the ADC does every sample or every second sample.

Every Sample is used for mono at greater than 50KHz with the filters switched out or stereo.

Every second sample is used for mono sampling at less than 50KHz.

With the Channel card pitch set at P, then the sampling frequencies and modes will be:-

Bit 7	Bit 6	Result
0	0	Stereo sampling at P/2 per channel.
0	1	Mono sampling at less than 50KHz at P/2 .
1	0	Mono sampling at greater than 50KHz at P.
1	1	Mono sampling at greater than 50KHz at P/2. (Invalid)

CALIBRATING THE CMI-337 ADC CARD (Rev 3)

CH1 means channel 1 of the CRO
CH2 means channel 2 of the CRO
all triggering is done on channel 1.

Connect the card to either a waveform processor or test card. This will generate the square wave that is used to start conversions.

Before data can be read from the ADC, 2 trim pots on the 74LS123 at C11 must be set.

1. Start-Conversion Pulse Delay

Put CH1 on C11 pin 10. Put CH2 on C11 pin 5 and adjust RV2 so that there is a high going pulse approximately 1.5 microsecond long. This will now allow the ADC to receive convert commands via F11 pin 9.

2. Data Clock Pulse Length

To allow data to be transmitted from the ADC card, Trimpot RV1 must be calibrated.

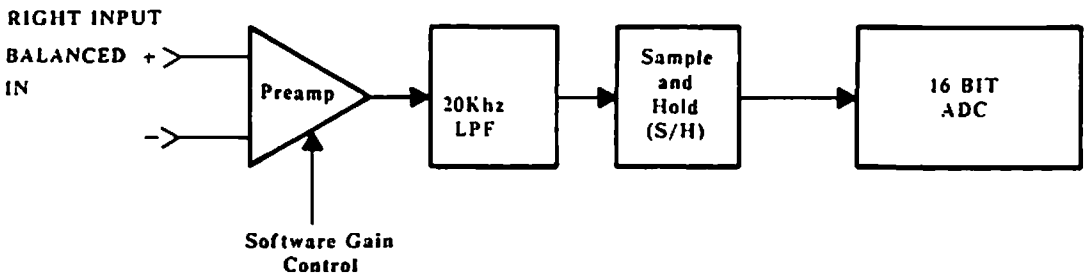
Put CH1 on F11 pin 9. Put CH2 on C11 pin 4. Adjust RV1 so that the pulses on CH2 are 50% duty cycle.

This adjustment is needed to stretch the last pulse from the ADC to allow it to be sent down the current loop.

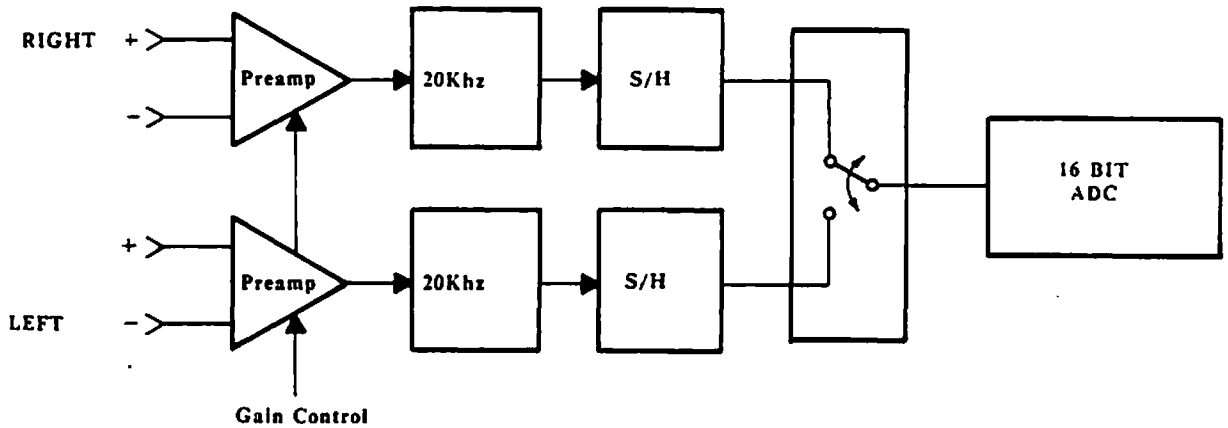
At this point the CMI-337 should be able to run with the CMI software. Messages such as "Sample fault" mean that the above adjustments haven't been made, that there is no power to the CMI-337 or that the CMI-337 is not plugged in correctly.

The 3 Modes of Operation of The ADC

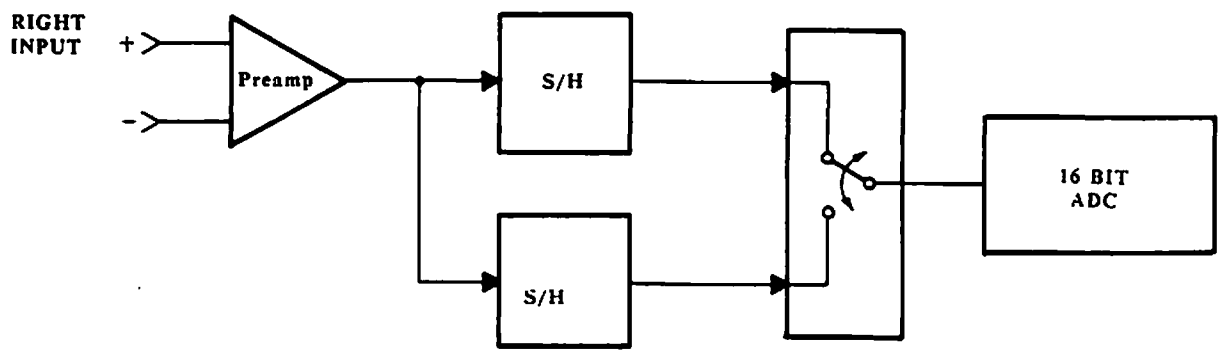
MONO MODE (50Khz or Less)

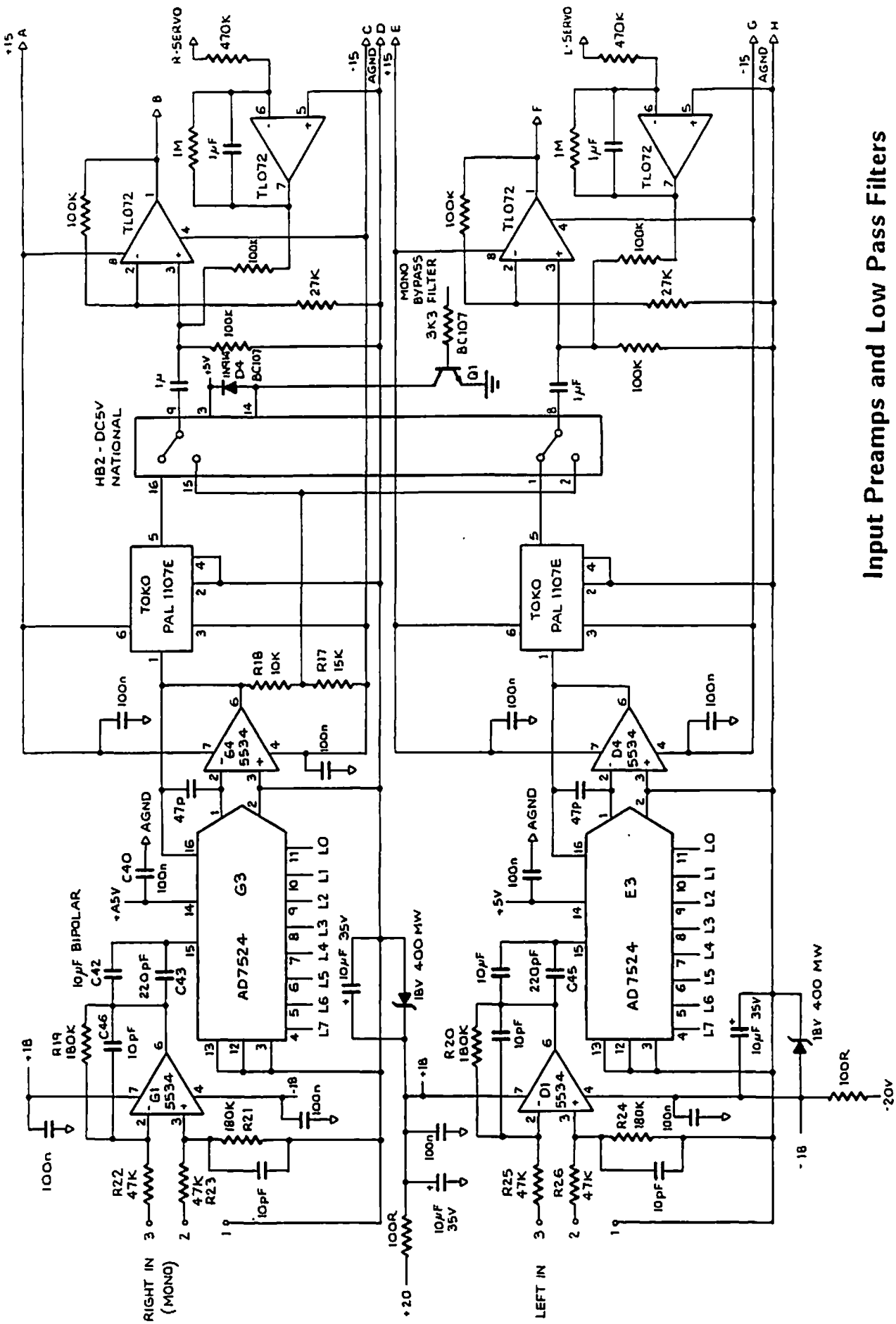


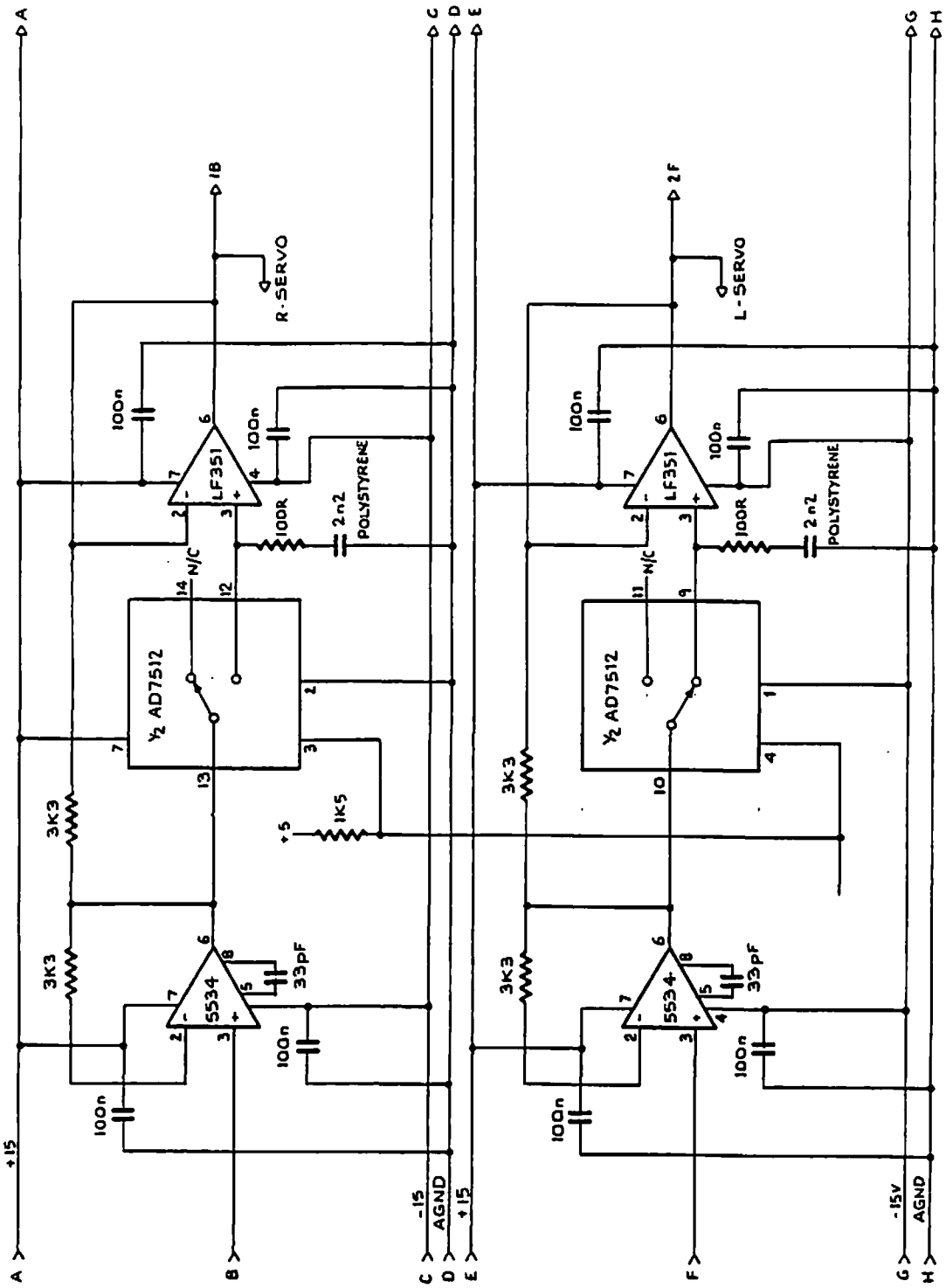
STEREO MODE (UP TO 50Khz A Channel)



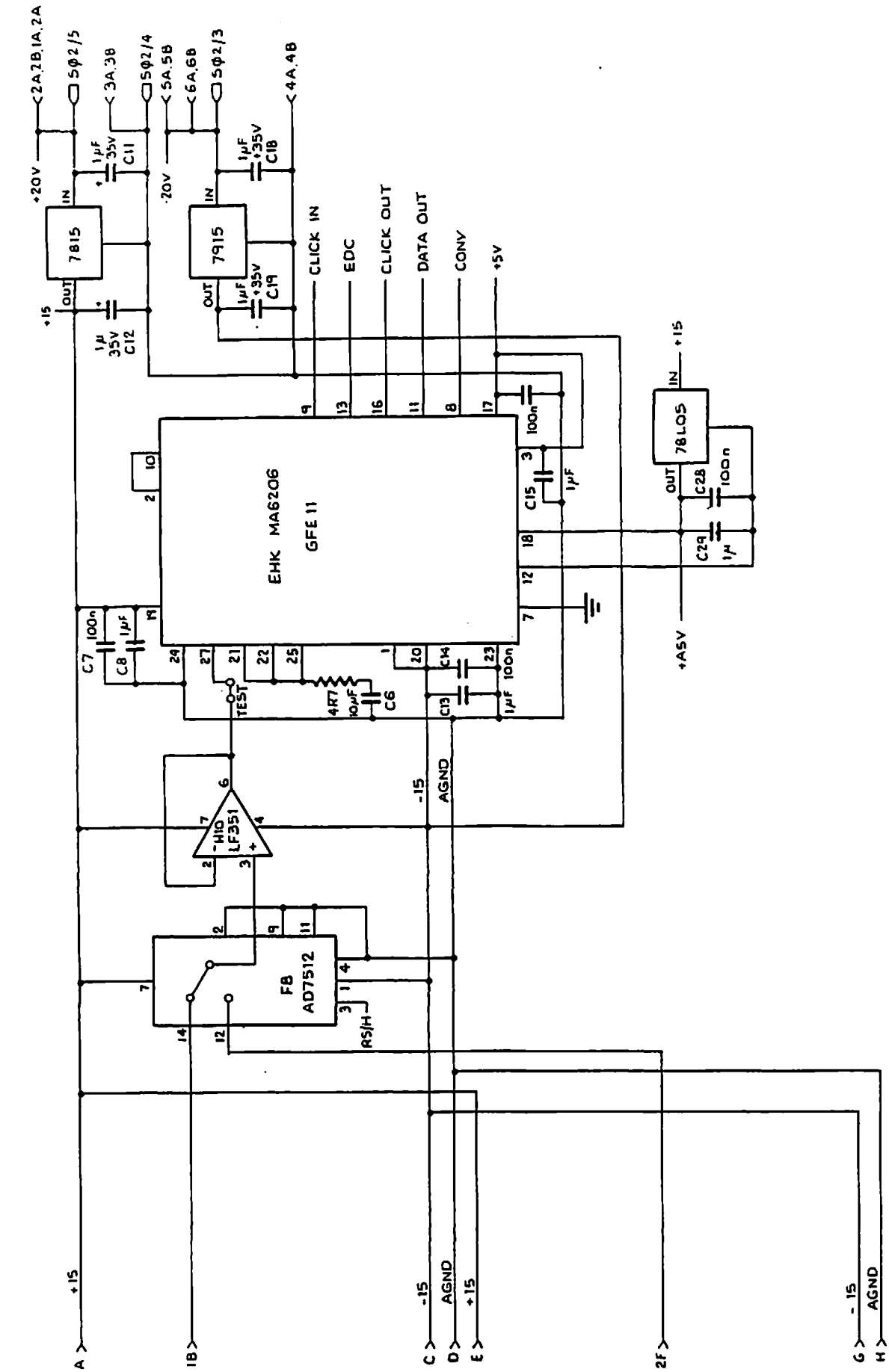
MONO MODE (Greater Than 50Khz)

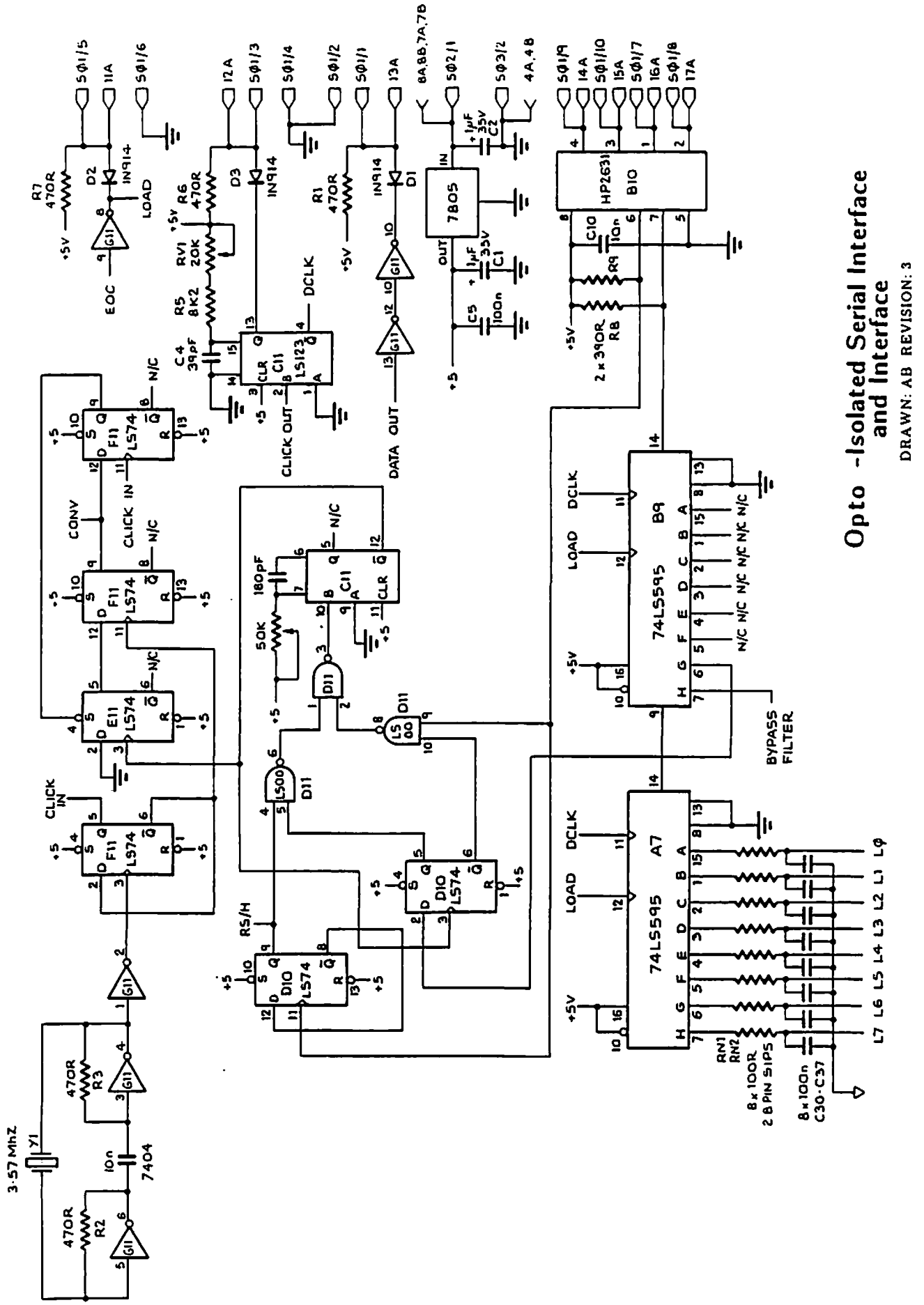


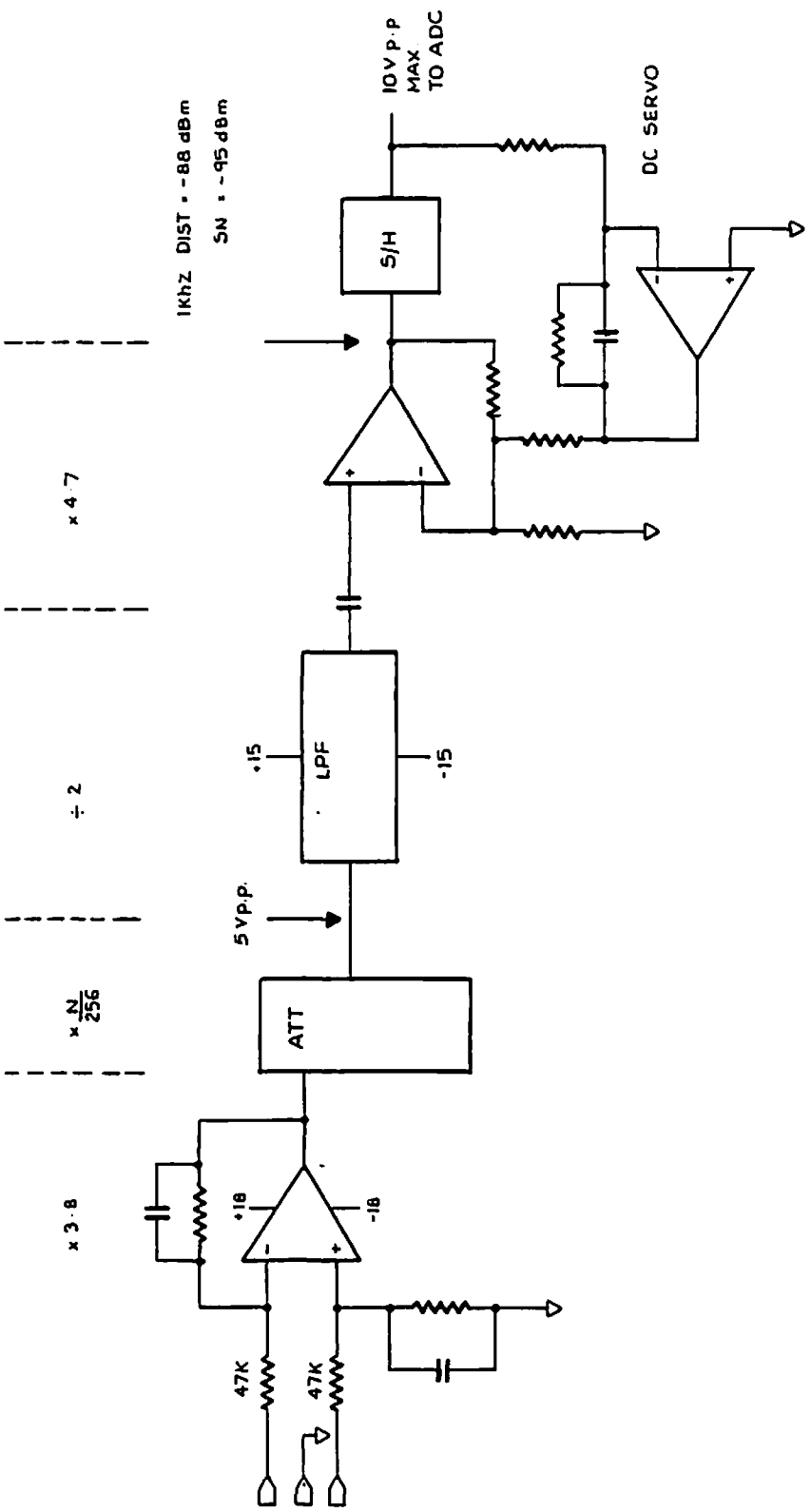

Input Preamps and Low Pass Filters
DRAWN: AB REVISION: 3
R-SERVO
L-SERVO
470K
TL072
1M
1µF
100K
27K
MONO BYPASS FILTER
3K3
BC107
Q1
HB2 - DC5V NATIONAL
TOKO PAL 1107E
R18 10K
R17 15K
100n
47P
AD7524
G3
E3
L7 L6 L5 L4 L3 L2 L1 LO
10µF BIPOLAR
C42
C43
C45
C46
220pF
10pF
R19 180K
R20 180K
R21
R22 47K
R23 47K
R24 180K
R25 47K
R26 47K
10µF 35V
18V 400 MW
100R
RIGHT IN (MONO)
LEFT IN
+15
-15
+18
-18
+20
-20V
AGND
+5V



Track-and-Holds
DRAWN: AB REVISION: 3

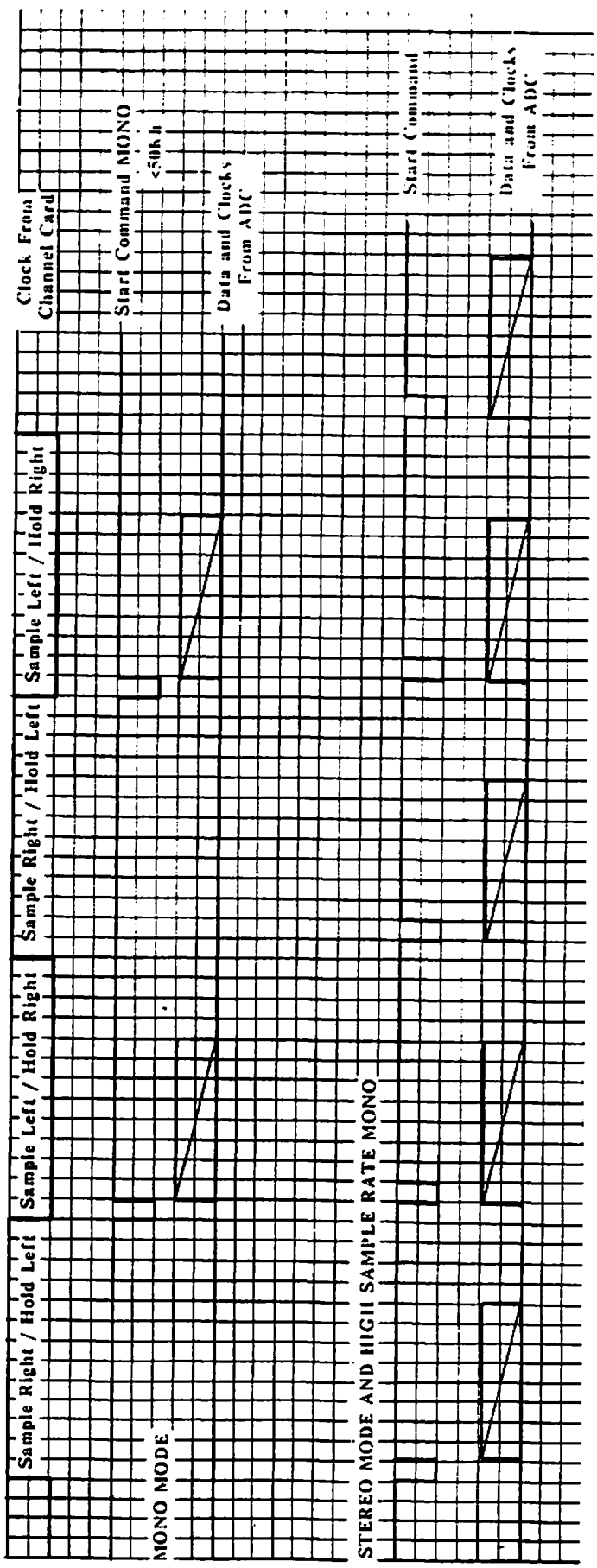

Sample and Holds and ADC
DRAWN: AB REVISION: 3
EHK MA6206
GFE 11
7815
7915
78L05
LF351
AD7512
CLICK IN
EDC
CLICK OUT
DATA OUT
CONV
+5V
+20V
-20V
+15
-15
AGND
+45V
TEST
RS/H
A
1B
C
D
E
2F
G
H


Opto -Isolated Serial Interface and Interface
DRAWN: AB REVISION: 3
CLICK OUT
DATA OUT
CLICK IN
CONV
EOC
LOAD
DCLK
RS/H
BYPASS FILTER
3.57 MHz
74LS595
HP2631
7805
LS123
LS74
LS00
7404
1N914
8 x 100R 2 8 PIN SIPS
8 x 100n C30-C37



Sampling Card Gain Structure

DRAWN: AB REVISION: 3



3 Modes of Operation of ADC Timing

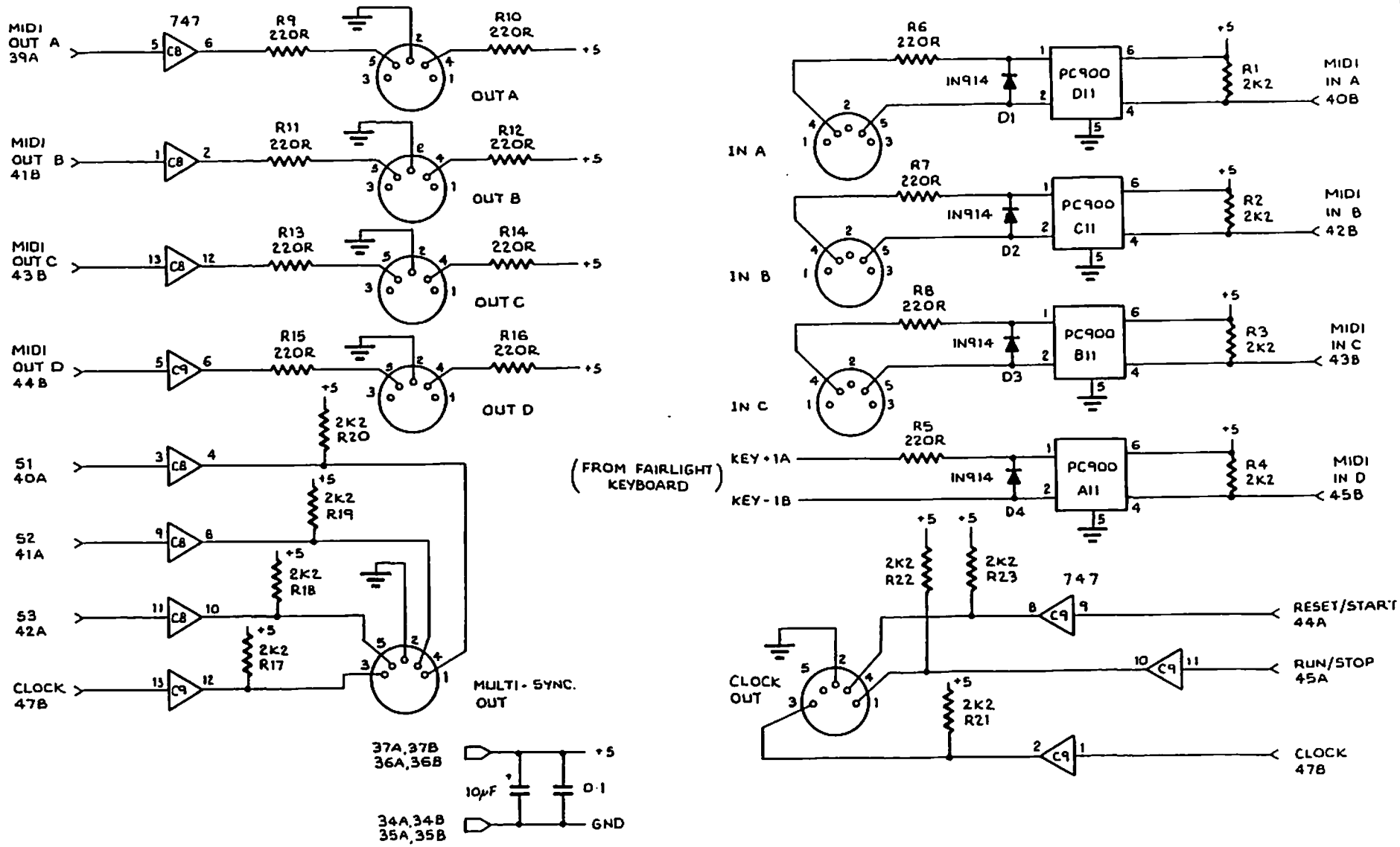
CMI-332 3.4

MIDI Module

CMI-332 MIDI Support Card

This circuit board contains the analog circuitry required for the I/O for MIDI. There are 3 MIDI inputs (A, B & C) and 4 MIDI outputs (A, B, C & D) all through DIN sockets. There is provision for a fourth MIDI input (D), this connects to the 9-way cable to the Fairlight keyboard. The MIDI I/O circuitry is the standard current loop drivers (open-collector buffers 7407 (C8,C9) and receivers (fast opto-couplers PC900 (A11,B11,C11,D11)).

There are two other output (5-pin DIN) sockets. One is the CLOCK output, containing the CLOCK, RESET/START and RUN/STOP TTL compatible signals. This CLOCK output is designed to control Roland drum machines, etc. The other is the multiple SYNC output. A click or sync signal received through the CLICK input is fed to the 68B40 Timers (see above). The outputs are connected to the DIN socket driven by open-collector buffers. You will notice that the SYNC out 4 signal is the same as that of the CLOCK and of CLICK out.



NOTES: INPUT SOCKETS ARE SHOWN LOOKING AT FACE OF THE CONNECTOR BODY
OUTPUT SOCKETS ARE SHOWN LOOKING AT REAR OF CONNECTOR BODY.

0·1 DECOUPLERS TO BE PLACED NEAR EACH I.C.

Drivers and Sockets
MIDI

DRAWN: PF REVISION: 1

Pin Connections for the 26-way Connector

(between the CMI-28 and CMI-332 and CMI-333)

Pin 1 MIDI out A.
Pin 2 +5 volts.
Pin 3 MIDI in A.
Pin 4 SYNC out 1.
Pin 5 MIDI out B.
Pin 6 SYNC out 2.
Pin 7 MIDI in B.
Pin 8 SYNC out 3.
Pin 9 MIDI out C.
Pin 10 Digital Ground.
Pin 11 MIDI in C.
Pin 12 Digital Ground.
Pin 13 MIDI out D.
Pin 14 RESET/START.
Pin 15 MIDI in D.
Pin 16 RUN/STOP.
Pin 17 SMPTE code in.
Pin 18 Digital Ground.
Pin 19 SMPTE code out.
Pin 20 CLICK out; SYNC out 4.
Pin 21 CLICK in.
Pin 22 (CMI332-3) Analog Ground.# (CMI28) n/c.*
Pin 23 (CMI332-3) +15 volts.# (CMI28) CPU Halt switch.*
Pin 24 (CMI332-3) -15 volts.# (CMI28) Digital Ground.*
Pin 25 (CMI332-3) n/c.# (CMI28) CPU Reset switch.*
Pin 26 (CMI332-3) n/c.# (CMI28) Digital Ground.*

Notes:

 # - these connections are on Audio Rack only.

*** - these connections (from the CMI28 board only) are for debugging purposes only. If two push-button switches are connected between pins 23 & 24 and pins 25 & 26, they can be used to manually halt and reset the 68K processor, respectively.**

CMI-333 3.5

SMPTE Module

Introduction

The SMPTE input has a balanced line receiver. The signal is then filtered and converted to TTL compatible signals through the LM311 comparator. The SMPTE out signal is converted from a TTL to a balanced line signal. The SMPTE in and out signals are received and transmitted via two 3-pin XLR sockets.

Pin Connections for the 26-way Connector

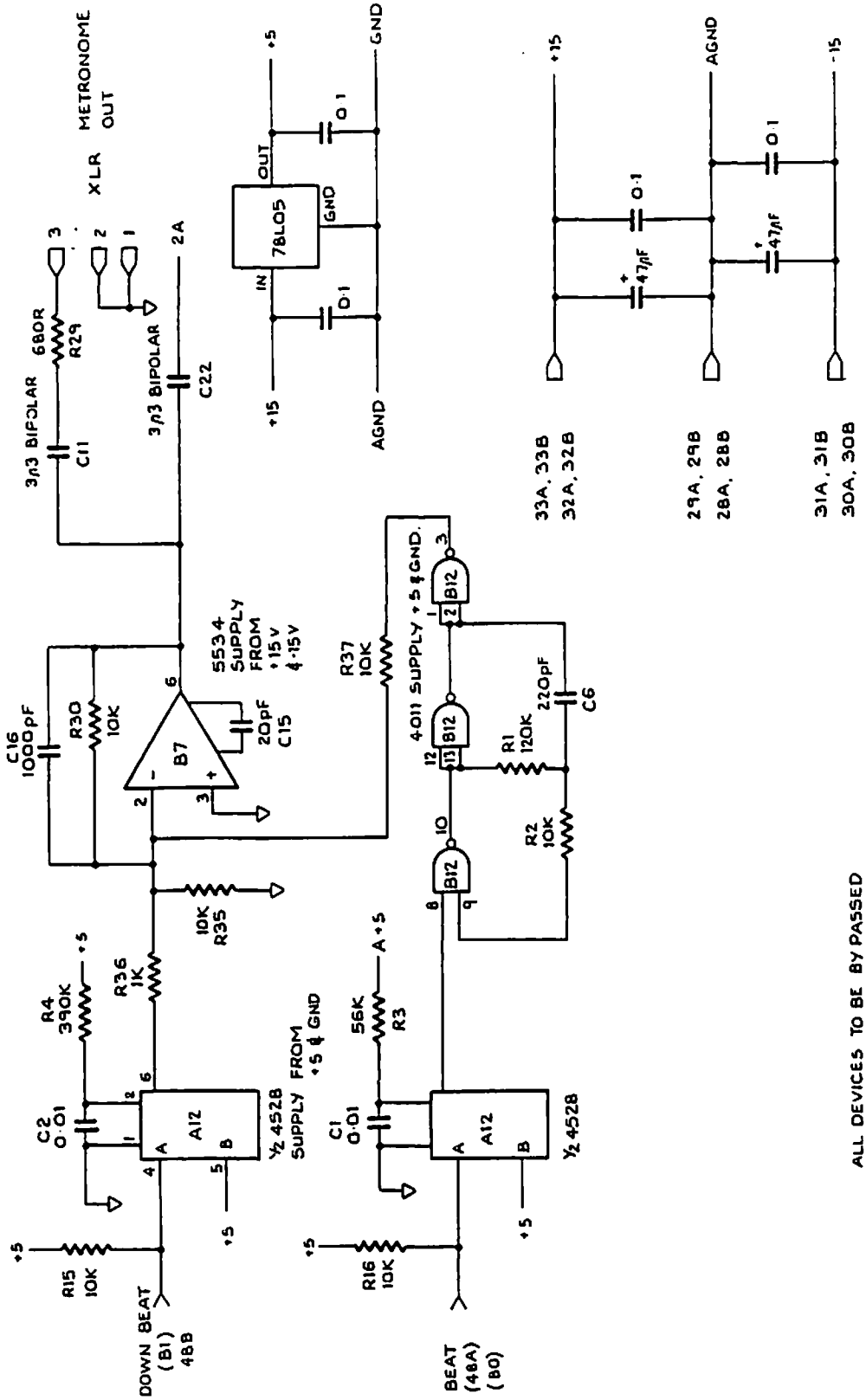
(between the CMI-28 and CMI-332 and CMI-333)

Pin 1 MIDI out A.
Pin 2 +5 volts.
Pin 3 MIDI in A.
Pin 4 SYNC out 1.
Pin 5 MIDI out B.
Pin 6 SYNC out 2.
Pin 7 MIDI in B.
Pin 8 SYNC out 3.
Pin 9 MIDI out C.
Pin 10 Digital Ground.
Pin 11 MIDI in C.
Pin 12 Digital Ground.
Pin 13 MIDI out D.
Pin 14 RESET/START.
Pin 15 MIDI in D.
Pin 16 RUN/STOP.
Pin 17 SMPTE code in.
Pin 18 Digital Ground.
Pin 19 SMPTE code out.
Pin 20 CLICK out; SYNC out 4.
Pin 21 CLICK in.
Pin 22 (CMI332-3) Analog Ground.# (CMI28) n/c.*
Pin 23 (CMI332-3) +15 volts.# (CMI28) CPU Halt switch.*
Pin 24 (CMI332-3) -15 volts.# (CMI28) Digital Ground.*
Pin 25 (CMI332-3) n/c.# (CMI28) CPU Reset switch.*
Pin 26 (CMI332-3) n/c.# (CMI28) Digital Ground.*

Notes:

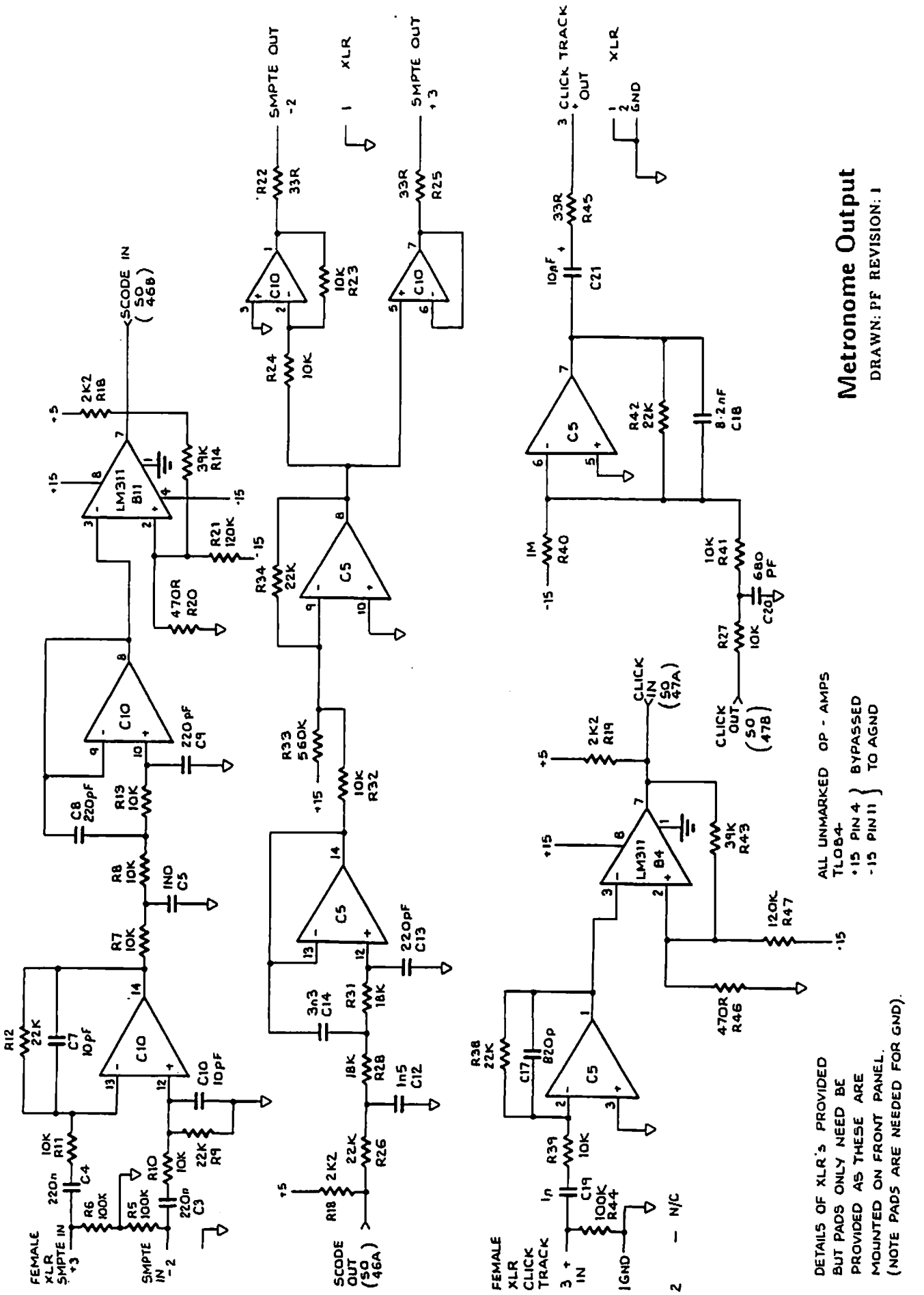
 # - these connections are on Audio Rack only.

*** - these connections (from the CMI28 board only) are for debugging purposes only. If two push-button switches are connected between pins 23 & 24 and pins 25 & 26, they can be used to manually halt and reset the 68K processor, respectively.**



SMPTE Drivers and Sockets

DRAWN: PF REVISION: 1


Metronome Output
DRAWN: PF REVISION: 1
SCODE IN (50 46B)
SMPTE OUT -2
SMPTE OUT +3
CLICK TRACK OUT
CLICK IN (50 47A)
CLICK OUT (50 47B)
SCODE OUT (50 46A)
FEMALE XLR SMPTE IN +3
SMPTE IN -2
FEMALE XLR CLICK TRACK
ALL UNMARKED OP - AMPS TL084
+15 PIN 4 } BYPASSED
-15 PIN 11 } TO AGND
DETAILS OF XLR's PROVIDED BUT PADS ONLY NEED BE PROVIDED AS THESE ARE MOUNTED ON FRONT PANEL. (NOTE PADS ARE NEEDED FOR GND).

CMI-334 3.6
Audio Mixer Module

Terminology
AM: Audio Module
AMM: Audio Mixer Module
AR: Audio Rack
AMB: Audio MotherBoard

Introduction
The Audio Mixer Module (AMM) provides a fixed 16 channel into 1 mixing facility for the Audio Rack (AR) assembly. It also provides the latches for controlling the facility to mix the A and B halves of the Audio Modules (AM). It is not essential to the operation of the CMI and may be left out if desired.

Mixing
The signals to be mixed are received as differential analog inputs from each channel. They are buffered and converted into unipolar form before being mixed and again buffered for output on an XLR type connector. This mixed signal is also sent to the CMI310 power supply and headphone amplifier where it is used to drive the headphone output. The output from the mixer to the CMI310 is connected electrically to both slots 10 and 11 of the AMB. If two mixer modules are present then the signal level to the headphones will be mixed but reduced in level. Note that, since the CMI334 is mono, both left and right signals to the headphones are connected together.

Controls
The AMM contains two eight bit latches, level shifters and address decoding logic to provide control signals to each AM. These signals are used to mix the A & B half of the module so that the channel card may pan between each voice.

Buffer Circuitry and Mixing
(refer schematic CMI334-00. 01,02)
The incoming differential signals from each voice are buffered by op-amp D1, D3, D4, D6, D7, D9, D10 and D12. These operate in a standard differential receiver format with some high frequency roll-off and stability provided by the 220pF capacitors.

Mixing is accomplished by the 16 10K ohm resistors feeding into the virtual earth current node of IC B1 pin 2. The 220 ohm standoff resistor provides some isolation from the relatively large capacitive load seen by pin 2. IC B1 is bypassed from pin 1 to 2 by a 470pF ceramic cap which reduces the bandwidth (and hence noise power) and also slugs the amplifier to a gain of 0 at high frequencies. The 1K ohm resistor in series with the 10K give a gain of just over 1 for normal mixing conditions. Switch 1 bypasses the 10K ohm for large signals so that output clipping may be avoided. It can also be used to reduce signal level into mixing consoles or amplifiers with insufficient headroom.

IC B2 is a 5532 type which may directly drive a 600 ohm balanced line in differential form. Note that 33 ohm standoff resistors are included in the feedback path reducing the output impedence while improving the stability with a capacitive load.

Two 1K ohm resistors are fitted to Rev 3 boards and also to Rev 2.1A to allow two mixer modules to be inserted into a CMI without damage. If these resistors are missing then the appropriate FCN should be performed. They are installed in series with the output signal to edge connectors 2 and 3 sides A and B.

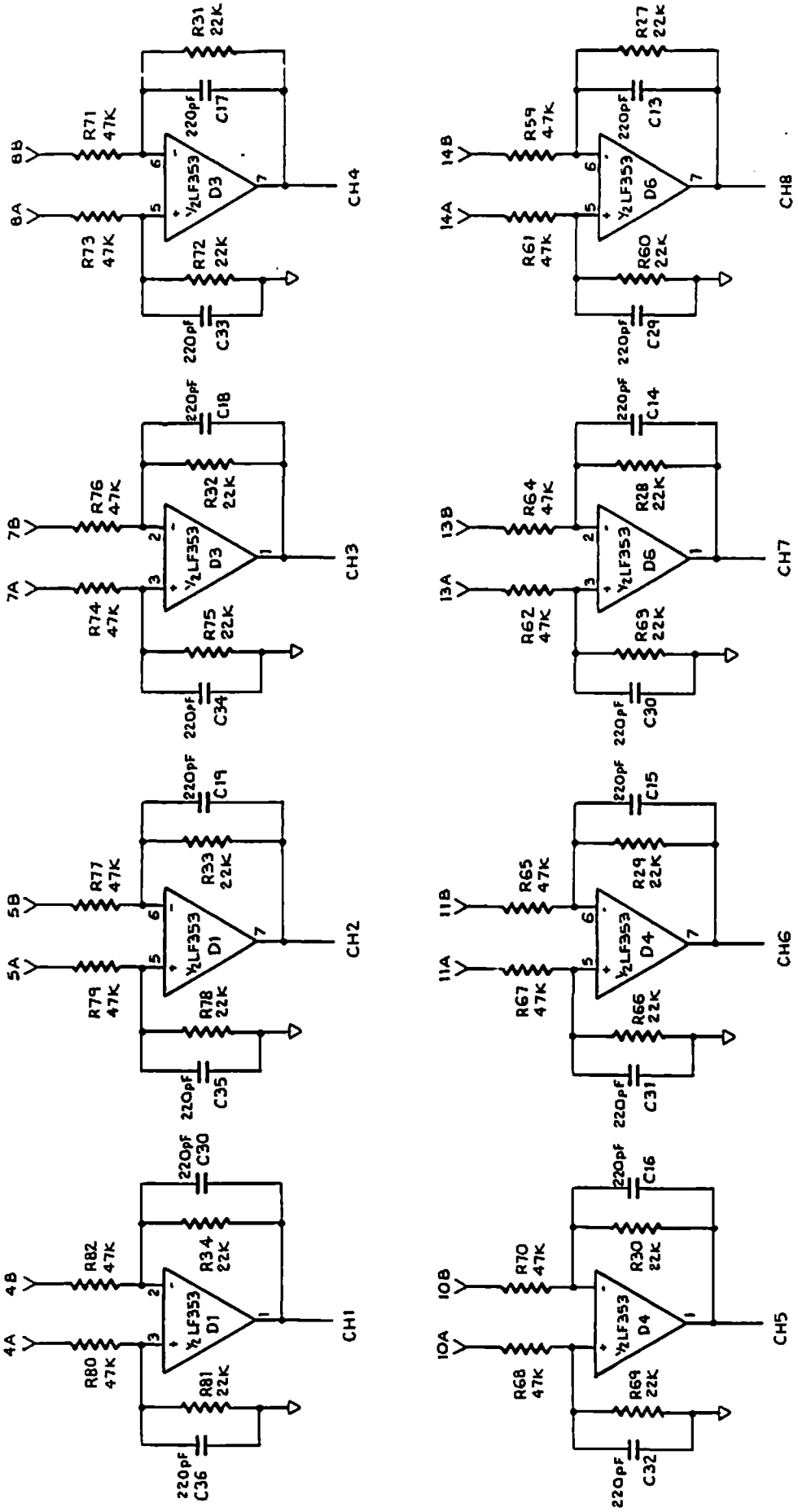
Control Circuitry
(refer schematic CMI334-03)

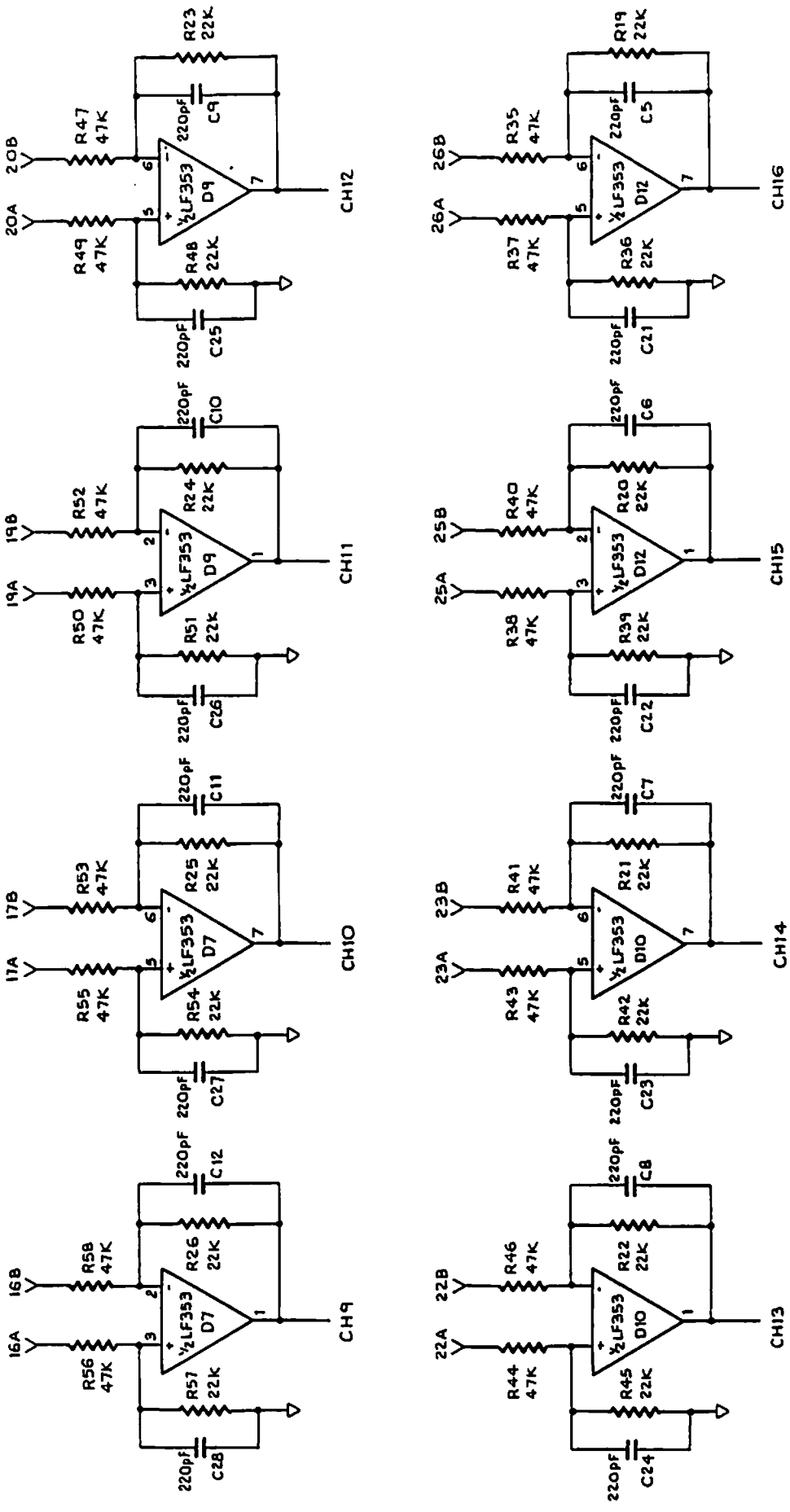
IC A9 (74HC138) decodes the lower address bits from the debug card PIA providing outputs so as to enable the latches (IC's A7 and B7 4099's) when 0 or 8 is presented. These signals and the higher bits are level shifted by IC B9 (4504) and fed to the latches. These further decode the higher bits and latch the state of bit 7 for output to the Audio Modules.

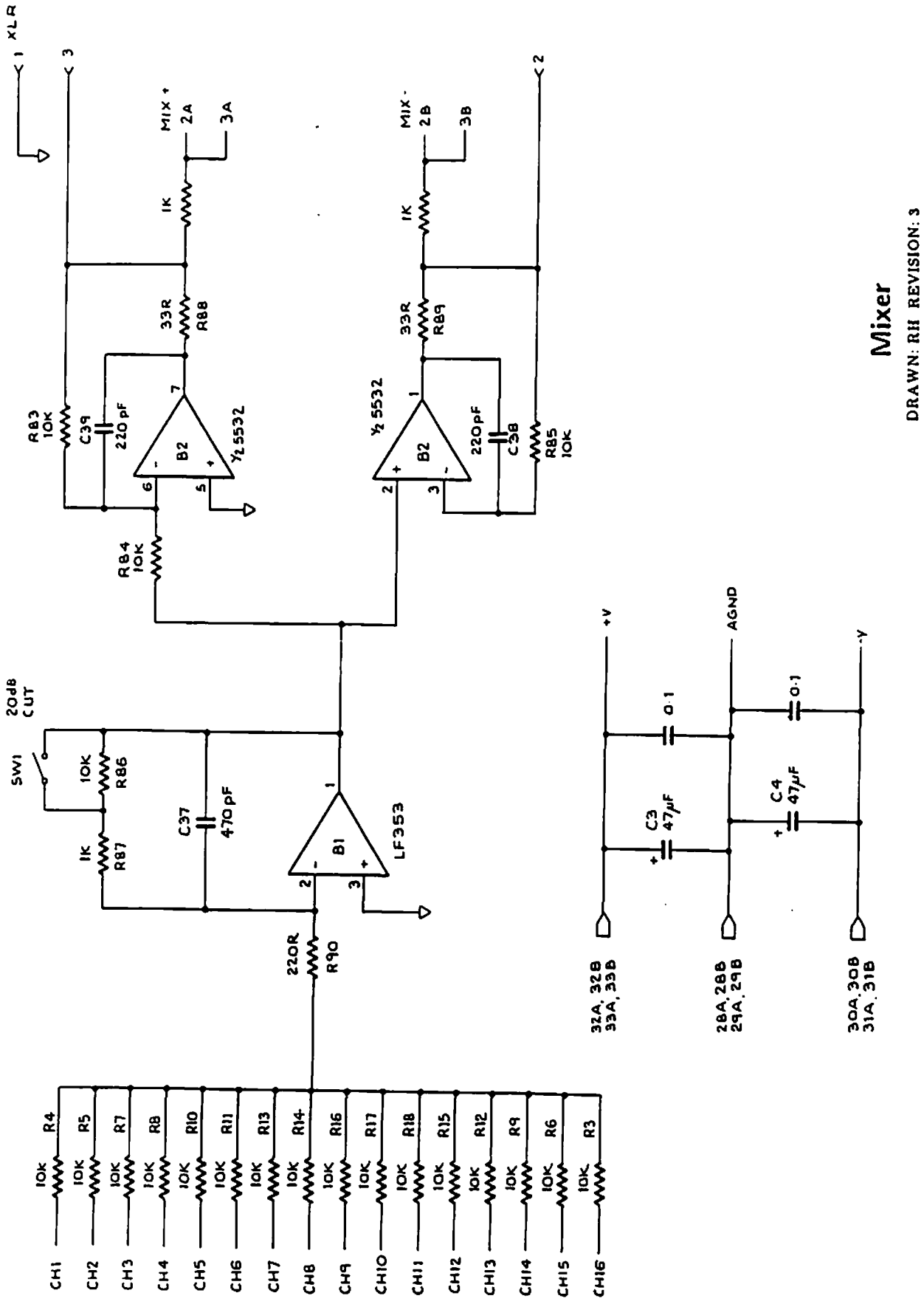
ICs A7 and B7 are bit addressable latches in which bit settings 'map' the PIA data to channel selections for mix control signals.

The LM317T provides the regulated 7 volts for driving the level shifted controls.

The rev 3 circuit card differs only in that it has sixteen LEDs added and a 16 pin socket for ease of testing the control circuitry. The LEDs indicate a control line is active.

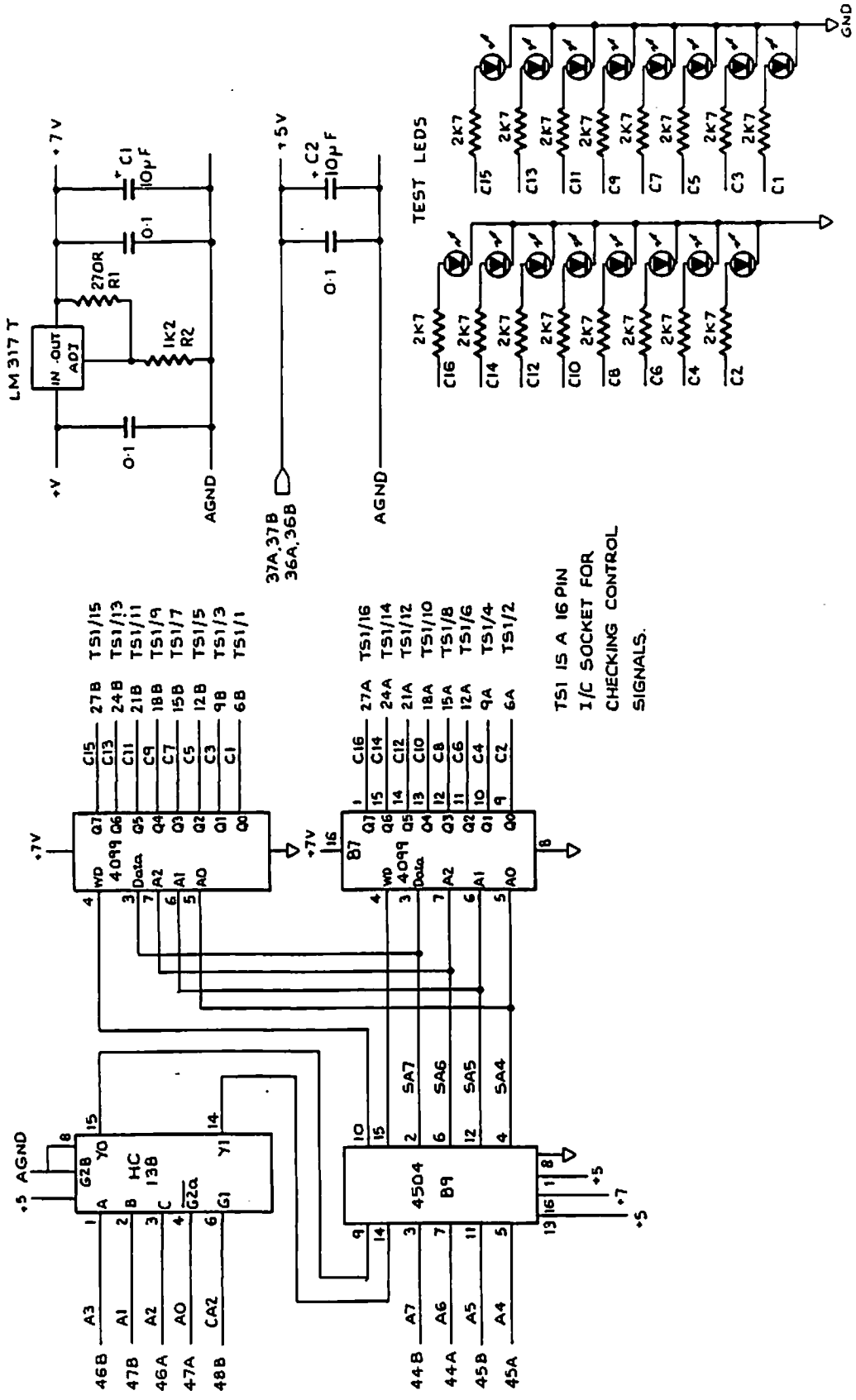

Differential Receivers
DRAWN: RH REVISION: 3
CH1
CH2
CH3
CH4
CH5
CH6
CH7
CH8
D1
D3
D4
D6
½LF353
4A
4B
5A
5B
7A
7B
10A
10B
11A
11B
13A
13B
14A
14B
R80 47K
R82 47K
R79 47K
R77 47K
R74 47K
R76 47K
R73 47K
R71 47K
R68 47K
R70 47K
R67 47K
R65 47K
R62 47K
R64 47K
R61 47K
R59 47K
R81 22K
R34 22K
R78 22K
R33 22K
R75 22K
R32 22K
R72 22K
R31 22K
R69 22K
R30 22K
R66 22K
R29 22K
R63 22K
R28 22K
R60 22K
R27 22K
220pF C36
220pF C30
220pF C35
220pF C19
220pF C34
220pF C18
220pF C32
220pF C16
220pF C31
220pF C15
220pF C30
220pF C14
220pF C29
220pF C13
220pF C33
220pF C17


Differential Receivers
DRAWN: RH REVISION: 3
CH12
CH11
CH10
CH9
CH16
CH15
CH14
CH13
½LF353



Mixer

DRAWN: RH REVISION: 3


Control Circuitry
DRAWN: RII REVISION: 3
TEST LEDS
GND
+7V
+5V
LM317T
IN OUT ADJ
270R R1
1K2 R2
C1 10µF
0·1
+V
AGND
C2 10µF
37A,37B
36A,36B
TS1 IS A 16 PIN I/C SOCKET FOR CHECKING CONTROL SIGNALS.
4099 B7
4504 B9
HC 138
+5 AGND
TS1/15 TS1/13 TS1/11 TS1/9 TS1/7 TS1/5 TS1/3 TS1/1
TS1/16 TS1/14 TS1/12 TS1/10 TS1/8 TS1/6 TS1/4 TS1/2
SA7 SA6 SA5 SA4
A3 A1 A2 A0 CA2
A7 A6 A5 A4
46B 47B 46A 47A 48B
44B 44A 45B 45A

CMI-335 3.7
Audio Mother Board

Terminology
WB: Waveform Buss
WP: Waveform Processor CMI-33
WRAM: Waveform RAM (1 to 7 cards each 2Mbytes) CMI39
CC: Channel Card (8 used per system) CMI31
CSC: Channel Support Card CMI32
AM: Audio Module (8 cards used per system) CMI-331
AMM: Audio Mixer Module CMI-334
MIDI CARD: CMI-332
SMPTE CARD: CMI-333
SAMPLER: CMI-337
SHIELD: CMI-336
DEBUG CARD: Q133
SMIDI BOARD: CMI28
AMB: Audio MotherBoard

Introduction

The Audio MotherBoard (AMB) is the central interface between the digital part or computer of the CMI and the audio output stage.

The audio circuitry receives all its power supplies and signals from this card. The exception is, of course, the Audio outputs and the MIDI, SMPTE and Sampler interfaces to the outside world.

The AMB is located in the centre of the CMI facing in the opposite direction to the Digital motherboard.

Structure of the Motherboard

The AMB is structured so that most connections of audio signals are accomplished without the need for hand wiring as in the Series II machines. The board is set up with thirteen slots into which plug the various modules for interfacing to the outside world. A modular arrangement was used to allow the required equipment to fit while still allowing ease of servicing and upgradeability. The power supply from CMI310 is via a multi-way power cable soldered directly to the board. The digital five volts is connected directly from the digital motherboard to the five volt buss on rev 0, 1 and 2 boards while rev 3 and 4 have a five amp fuse mounted on the AMB and solder pads for the cables.

Starting at the right hand end of the AMB is slot 1 which holds the Sampler card for stereo input. The next eight slots are for the eight Audio Modules (AM) followed by two slots (10 & 11) for mixer modules and two for the MIDI and SMPTE modules (12 & 13).

Cables to supply these cards are connected via IDC type mass termination connectors at either end of the AMB.

The digital and analog grounds have their only direct connection on the CMI-335 between slots 5 and 6. Both grounds are separately bussed to each slot.

Interconnection Structure

Slot 1 (Sampler CMI-337) is connected to its own power supply from CMI310 via the power cable, signals are connected via a ten way cable to the Waveform Processor (WP).

Slots 2 through 9 are for Audio Modules. The Audio Modules connect to the motherboard via two edge connectors, one a double

sided thirty-four way device (68 pins total) for those signals which are bussed along the motherboard and the second a double sided thirteen way connector (26 pins total). This connector is an edge to cable IDE type connector which allows the signals from the CC to connect directly to the AM without going via the AMB first. Since the two connectors are in line with one another they are considered to be one edge connector split into two sections. This allows unambiguous labelling of the AM edge fingers. To satisfy the physical positioning of the two edge connectors the same numbering pattern used in the digital card cage is used here, i.e. The edge connector is numbered as if it were a 78 way double sided device with the solder side labelled as the A side and the component side as the B side, the IDE type connector occupying fingers 8 A and B through to 20 A and B, the 34 way connector then occupies fingers 45 A and B through to 78 A and B.

The above numbering system is used everywhere with any exception specifically noted.

The AM slots receive data, power and digital control lines and output audio for mixing purposes via the 34 way connector. The flat ribbon cable connected to the IDE connector carries power from the AM to the CC, data clocks, and analog control voltages in the reverse direction.

Slots 10 and 11 are for mixer modules which are optional devices. Each CMI is shipped with a CMI-334 basic mixer module which is primarily for headphone and monitoring purposes. Slots 10 and 11 differ in that the control function outputs from the mixer modules are connected on slot 11 but not on slot 10. Two reasons exist for this, one is that if two modules are inserted then both would conflict on driving the mixer lines, the second reason is that by positioning the mixer in slot 10 the AMs are set so that no digital control signals (i.e. mix) are active.

Slots 12 and 13 are for the MIDI and SMPTE support modules. These may be inserted in any order since these slots have no electrical difference. The MIDI and SMPTE cards, and the mixers, connect via a double sided 48 way edge connector (96 pins). The main connections are to the CMI-28 SMIDI card and to the power supply. Note that the MIDI connection to the FAIRLIGHT music keyboard is routed in an unusual fashion, this is to avoid additional cabling within the CMI.

Circuit Description

The AMB is mainly an interconnection device for various signals used in the audio section.

Looking from the component side, in the top left corner the 34 way flat cable connector which carries the 16 bit data from the waveform buss in differential form is located. This data is bussed along slots 2 through 9 for the AMs. The 10 way ribbon cable connector, located at the bottom right corner of the AMB near slot 1, is for the Sampler signals to the waveform processor.

On the left hand end of the AMB are 3 ribbon cable connectors, the Debug cable (J1) , the MIDI cable (J2) and the headphone output cable (J3). These connect to the Debug card, the MIDI card and the CMI310 power supply board respectively.

Slot Schedule

Each slot is listed below showing the connections available from the AMB.

Slot 1: Stereo ADC.

Side A

Pin	Signal Name	Function	Source
20	MUTE	power on mute	various
17	-DATA	data from WP	CMI33
16	+DATA	data from WP	CMI33
15	-SAMPCLK	start conversion pulse	CMI33
14	+SAMPCLK	start conversion pulse	CMI33
13	DATAOUT	data to WP	CMI337
12	CLKOUT	bit clock to WP	CMI337
11	EOC	end of conversion	CMI337
9,10	DSGND	digital gnd	CMI337
7,8	+9V	+9V sampler supply	CMI310
5,6	-20V	-20V sampler supply	CMI310
3,4	ASGND	sampler analog gnd	CMI310
1,2	+20V	+20V sampler supply	CMI310

Side B

Pin	Signal Name	Function	Source
9,10	DSGND	digital gnd	CMI33
7,8	+9V	+9 sampler supply	CMI310
5,6	-20V	-20V sampler supply	CMI310
3,4	ASGND	sampler analog gnd	CMI310
1,2	+20V	+20 sampler supply	CMI310

A polarising key is fitted in place of fingers 39A and B

Slots 2 to 9
These signals are bussed.
Please see text for description of numbering system used for edge fingers.

Side A

Pin	Signal Name	Function	Source
77,78	D+5	digital 5 volts	CMI35
76-61	D0+ to D15+[1]	16 bit WB data +	CMI32
59,60	DGND	digital gnd	CMI35
58		polarizing key	
56,57		spare	
55	MUTE	mute control	various
53,54	+15V	regulated +15volts	CMI310
51,52	-15V	regulated -15volts	CMI310
48-50	AGND	analog gnd	CMI310

Note 1: data to the audio modules is differential with + and - on opposite sides of the edge connector.

The following are signals which arrive at slots 2 through 9 via IDE type cable to edge connectors, each CC connecting to its corresponding AM (slot 2 from CC 1,.. slot 9 from CC 8).

20	n/c	no connection	CMI31
19	DGND	digital gnd	CMI31
18	PCLK-2[2]	pitch clock 2	CMI31
17	PCLK-1[2]	pitch clock 1	CMI31
16	DCLK-1[2]	data clock 1	CMI31
15	DCLK-2[2]	data clock 2	CMI31
13,14	n/c	no connection	CMI31
12	AGND	analog gnd	CMI331
11	+15V	+15volts to CMI31	CMI331
10	RES1[3]	resonance control	CMI31
9	VCF1[3]	filter control	CMI31
8	VCA1[3]	VCA control	CMI31

Note 2: PCLKs and DCLKs are differential with + and - on opposite sides of the connector. Two differential pairs of each for two channels on the AM.
Note 3: Two sets of RES, VCF and VCA on opposite sides of the connector for the Two channels of the AM.

Side B

Pin	Signal Name	Function	Source
77,78	D+5	digital +5 volts	CMI35
76-61	D0- to D15-[1]	data 0 to data 15 -	CMI32
59,60	DGND	digital gnd	CMI35
58		polarizing key	
55-57		spare	
53,54	+15V	regulated +15 volts	CMI310
51,52	-15V	regulated -15 volts	CMI310
48-50	AGND	analog gnd	CMI310

The following are signals which arrive at slots 2 through 9 via IDE cable to edge connectors, each CC connecting to its corresponding AM (slot 2 from CC 1,.. slot 9 from CC 8).

20	n/c	no connection	CMI31
19	n/c	no connection	CMI31
18	DGND	digital gnd	CMI31
17	PCLK+2[2]	pitch clock	CMI31
16	PCLK+1[2]	pitch clock	CMI31
15	DCLK+1[2]	data clock	CMI31
14	DCLK+2[2]	data clock	CMI31
13	n/c	no connection	CMI31
12	-15V	-15volts to CMI31	CMI331
11	AGND	analog gnd to CMI31	CMI331
10	RES2[3]	resonance control	CMI31
9	VCF2[3]	filter control	CMI31
8	VCA2[3]	VCA control	CMI31

Signals which are not common or bussed are described below.

Slot 2 **Audio Module 1**

Side A

Pin	Signal Name	Function	Source
47	CONTROL1	audio module control 1	slot 11
46	AUDIO1+	channel 1 audio +	CMI331
45	AUDIO2+	channel 2 audio +	CMI331

Side B

Pin	Signal Name	Function	Source
47	CONTROL2	audio module control 2	slot 11
46	AUDIO1-	channel 1 audio -	CMI331
45	AUDIO2-	channel 2 audio -	CMI331

Slot 3 **Audio Module 2**

Side A

Pin	Signal Name	Function	Source
47	CONTROL3	audio module control 3	slot 11
46	AUDIO3+	channel 3 audio +	CMI331
45	AUDIO4+	channel 4 audio +	CMI331

Side B

Pin	Signal Name	Function	Source
47	CONTROL4	audio module control 4	slot 11
46	AUDIO3-	channel 3 audio -	CMI331
45	AUDIO4-	channel 4 audio -	CMI331

Slot 4 **Audio Module 3**

Side A

Pin	Signal Name	Function	Source
47	CONTROL5	audio module control 5	slot 11
46	AUDIO5+	channel 5 audio +	CMI331
45	AUDIO6+	channel 6 audio +	CMI331

Side B

Pin	Signal Name	Function	Source
47	CONTROL6	audio module control 6	slot 11
46	AUDIO5-	channel 5 audio -	CMI331
45	AUDIO6-	channel 6 audio -	CMI331

Slot 5 **Audio Module 4**

Side A

Pin	Signal Name	Function	Source
47	CONTROL7	audio module control 7	slot 11
46	AUDIO7+	channel 7 audio +	CMI331
45	AUDIO8+	channel 8 audio +	CMI331

Side B

Pin	Signal Name	Function	Source
47	CONTROL8	audio module control 8	slot 11
46	AUDIO7-	channel 7 audio -	CMI331
45	AUDIO8-	channel 8 audio -	CMI331

Slot 6 **Audio Module 5**

Side A

Pin	Signal Name	Function	Source
47	CONTROL9	audio module control 9	slot 11
46	AUDIO9+	channel 9 audio +	CMI331
45	AUDIO10+	channel 10 audio +	CMI331

Side B

Pin	Signal Name	Function	Source
47	CONTROL10	audio module control 10	slot 11
46	AUDIO9-	channel 9 audio -	CMI331
45	AUDIO10-	channel 10 audio -	CMI331

Slot 7 **Audio Module 6**

Side A

Pin	Signal Name	Function	Source
47	CONTROL11	audio module control 11	slot 11
46	AUDIO11+	channel 11 audio +	CMI331
45	AUDIO12+	channel 12 audio +	CMI331

Side B

Pin	Signal Name	Function	Source
47	CONTROL12	audio module control 12	slot 11
46	AUDIO11-	channel 11 audio -	CMI331
45	AUDIO12-	channel 12 audio -	CMI331

Slot 8 **Audio Module 7**

Side A

Pin	Signal Name	Function	Source
47	CONTROL13	audio module control 13	slot 11
46	AUDIO13+	channel 13 audio +	CMI331
45	AUDIO14+	channel 14 audio +	CMI331

Side B

Pin	Signal Name	Function	Source
47	CONTROL14	audio module control 14	slot 11
46	AUDIO13-	channel 13 audio -	CMI331
45	AUDIO14-	channel 14 audio -	CMI331

Slot 9 **Audio Module 8**

Side A

Pin	Signal Name	Function	Source
47	CONTROL15	audio module control 15	slot 11
46	AUDIO15+	channel 15 audio +	CMI331
45	AUDIO16+	channel 16 audio +	CMI331

Side B

Pin	Signal Name	Function	Source
47	CONTROL16	audio module control 16	slot 11
46	AUDIO15-	channel 15 audio -	CMI331
45	AUDIO16-	channel 16 audio -	CMI331

Slot 10 and 11 Mixers
The following signals are common to both slots.

Side A

Pin	Signal Name	Function	Source
48	CA1	Debug PIA line	Q133
47	A0	Debug PIA line	Q133
46	A2	Debug PIA line	Q133
45	A4	Debug PIA line	Q133
44	A6	Debug PIA line	Q133
43	B0	Debug PIA line	Q133
42	B2	Debug PIA line	Q133
41	B4	Debug PIA line	Q133
40	B6	Debug PIA line	Q133
39		Polarizing key	
38	CB1	Debug PIA line	Q133
36,37	D+5	digital +5 volts	CMI35
34,35	DGND	digital gnd	CMI35
32,33	+15V	regulated +15 volts	CMI310
30,31	-15V	regulated -15 volts	CMI310
28,29	AGND	analog gnd	CMI310
27	CONTROL16	control line 16	CMI334
26	AUDIO16+	audio output 16	slot 9
25	AUDIO15+	audio output 15	slot 9
24	CONTROL14	control line 14	CMI334
23	AUDIO14+	audio output 14	slot 8
22	AUDIO13+	audio output 13	slot 8
21	CONTROL12	control line 12	CMI334
20	AUDIO12+	audio output 12	slot 7
19	AUDIO11+	audio output 11	slot 7
18	CONTROL10	control line 10	CMI334
17	AUDIO10+	audio output 10	slot 6
16	AUDIO9+	audio output 9	slot 6
15	CONTROL8	control line 8	CMI334
14	AUDIO8+	audio output 8	slot 5
13	AUDIO7+	audio output 7	slot 5
12	CONTROL6	control line 6	CMI334
11	AUDIO6+	audio output 6	slot 4
10	AUDIO5+	audio output 5	slot 4
9	CONTROL4	control line 4	CMI334
8	AUDIO4+	audio output 4	slot 3
7	AUDIO3+	audio output 3	slot 3
6	CONTROL2	control line 2	CMI334
5	AUDIO2+	audio output 2	slot 2
4	AUDIO1+	audio output 1	slot 2
3	MIXRT+	right mixed audio	CMI334
2	MIXLFT+	left mixed audio	CMI334

Side B

Pin	Signal Name	Function	Source
48	CA2	debug PIA line	Q133
47	A1	debug PIA line	Q133
46	A3	debug PIA line	Q133
45	A5	debug PIA line	Q133
44	A7	debug PIA line	Q133
43	B1	debug PIA line	Q133
42	B3	debug PIA line	Q133
41	B5	debug PIA line	Q133
40	B7	debug PIA line	Q133
39		polarizing key	
38	CB2	debug PIA line	Q133
36,37	D+5	digital 5 volts	CMI35
34,35	DGND	digital GND	CMI35
32,33	+15V	regulated supply	CMI310
30,31	-15V	regulated supply	CMI310
28,29	AGND	analog GND	CMI35
27	CONTROL15	control line 15	CMI334
26	AUDIO16-	audio output 16	slot 9
25	AUDIO15-	audio output 15	slot 9
24	CONTROL13	control line 13	CMI334
23	AUDIO14-	audio output 14	slot 8
22	AUDIO13-	audio output 13	slot 8
21	CONTROL11	control line 11	CMI334
20	AUDIO12-	audio output 12	slot 7
19	AUDIO11-	audio output 11	slot 7
18	CONTROL9	control line 9	CMI334
17	AUDIO10-	audio output 10	slot 6
16	AUDIO9-	audio output 9	slot 6
15	CONTROL7	control line 7	CMI334
14	AUDIO8-	audio output 8	slot 5
13	AUDIO7-	audio output 7	slot 5
12	CONTROL5	control line 5	CMI334
11	AUDIO6-	audio output 6	slot 4
10	AUDIO5-	audio output 5	slot 4
9	CONTROL3	control line 3	CMI334
8	AUDIO4-	audio output 4	slot 3
7	AUDIO3-	audio output 3	slot 3
6	CONTROL1	control line 1	CMI334
5	AUDIO2-	audio output 2	slot 2
4	AUDIO1-	audio output 1	slot 2
3	MIXRT-	right mixed audio	CMI334
2	MIXLFT-	left mixed audio	CMI334
1	MUTE	mute control line	various

The following signals are unique to each slot.

Slot 10
Side A

Pin	Signal Name	Function	Source
1	D+5	module select 2	CMI35

Slot 11

Pin	Signal Name	Function	Source
1	DGND	module select 1	CMI35

Slot 12 and 13 MIDI and SMPTE
Both slots are identical.

Side A

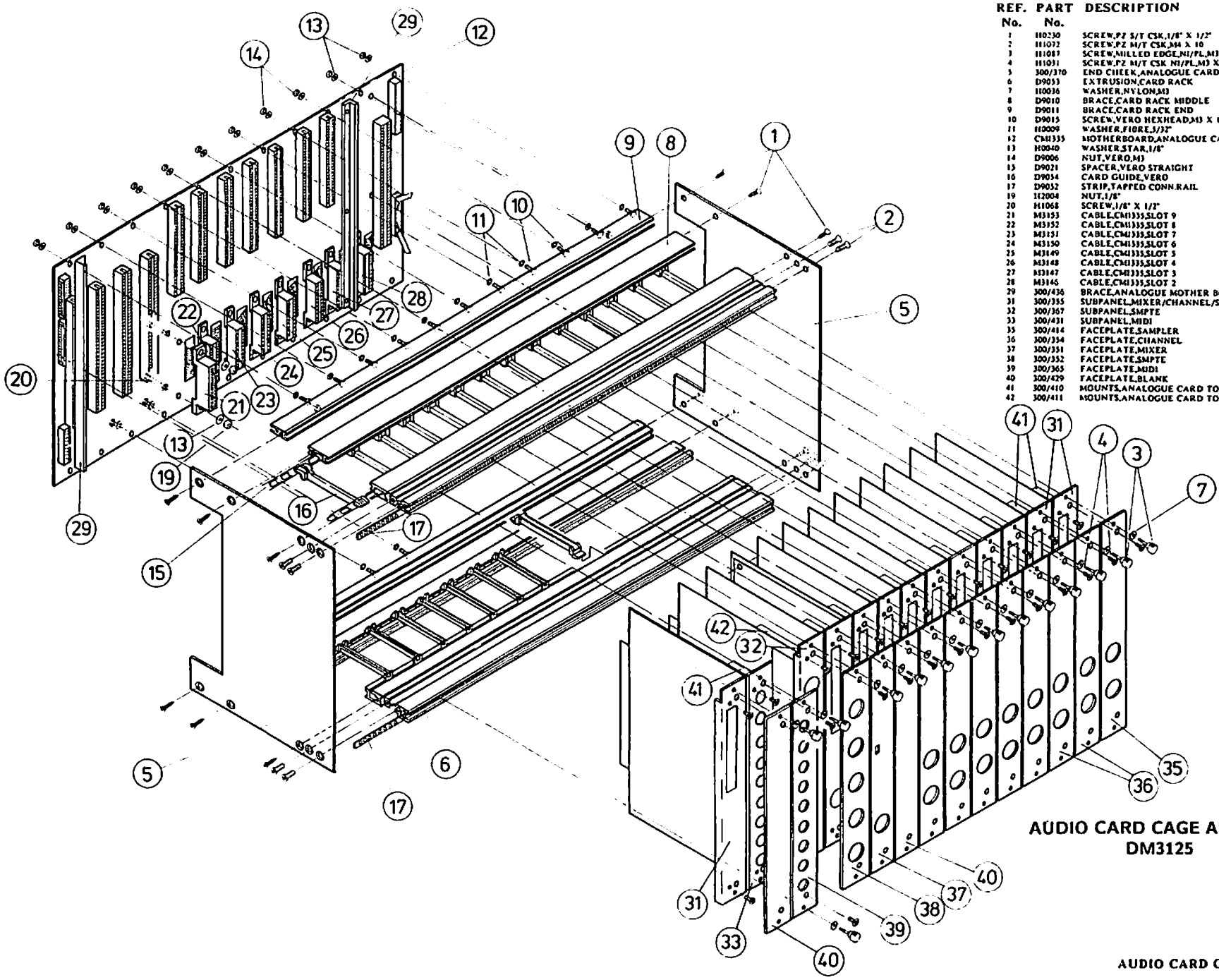
Pin	Signal Name	Function	Source
47	CLICK IN	click track input	CMI333
46	SMPTE OUT	SMPTE signal output	CMI28
45	RUN/STOP	drum machine control	CMI28
44	RESET/START	drum machine control	CMI28
43	MIDI IN C	MIDI input 3	CMI332
42	SYNC OUT 3	sync output	CMI28
41	SYNC OUT 2	sync output	CMI28
40	SYNC OUT 1	sync output	CMI28
39	MIDI OUT A	MIDI output 1	CMI28
36,37	D+5	digital 5 volts	CMI35
34,35	DGND	digital GND	CMI35
32,33	+15V	regulated supply	CMI310
30,31	-15V	regulated supply	CMI310
28,29	AGND	analog GND	CMI310
21	LK19	link between slots	future
20	LK17	link between slots	future
19	LK15	link between slots	future
18	LK13	link between slots	future
17	LK11	link between slots	future
16	LK9	link between slots	future
15	LK7	link between slots	future
14	LK5	link between slots	future
13	LK3	link between slots	future
12	LK1	link between slots	future
10		polarizing key	
3	MUTE	mute control line	various
2	METOUT	metronome out	CMI333
1	KEY+	MIDI from keyboard	CMI310

Side B

Pin	Signal Name	Function	Source
48	PB1	PIA control line	Q133
47	CLICK OUT	sync output	CMI28
46	SMPTE IN	SMPTE input	CMI333
45	MIDI IN D	MIDI input 4 (KBD)	CMI332
44	MIDI OUT D	MIDI output 4	CMI28
43	MIDI OUT C	MIDI output 3	CMI28
42	MIDI IN B	MIDI input 2	CMI332
41	MIDI OUT B	MIDI ouput 2	CMI28
40	MIDI IN A	MIDI input 1	CMI332
36,37	D+5	digital 5 volts	CMI35
34,35	DGND	digital GND	CMI35
32,33	+15V	regulated supply	CMI310
30,31	-15V	regulated supply	CMI310
28,29	AGND	analog GND	CMI310
21	LK20	link between slots	future
20	LK18	link between slots	future
19	LK16	link between slots	future
18	LK14	link between slots	future
17	LK12	link between slots	future
16	LK10	link between slots	future
15	LK8	link between slots	future
14	LK6	link between slots	future
13	LK4	link between slots	future
12	LK2	link between slots	future
10		polarizing key	
2	AGND	analog GND	CMI310
1	KEY-	MIDI from keyboard	CMI310

Cable Schedule

Socket J1 is connected as per table 8 in the CMI wiring schedule.
Socket J2 is as per table 7 in the CMI wiring schedule.
Socket J3 is as per table 5, Socket J4 is as per table 11 and socket J5 is as per table 9.



REF. No.	PART No.	DESCRIPTION
1	H0230	SCREW,PZ S/T CSK,1/8" X 1/2"
2	H1072	SCREW,PZ M/T CSK,M4 X 10
3	H1087	SCREW,MILLED EDGE,NI/PL,M3 X 11
4	H1031	SCREW,PZ M/T CSK NI/PL,M3 X 6
5	300/370	END CHEEK,ANALOGUE CARD CAGE
6	D9033	EXTRUSION,CARD RACK
7	H0036	WASHER,NYLON,M3
8	D9010	BRACE,CARD RACK MIDDLE
9	D9011	BRACE,CARD RACK END
10	D9015	SCREW,VERO HEXHEAD,M3 X 10
11	H0009	WASHER,FIBRE,3/32"
12	CMI335	MOTHERBOARD,ANALOGUE CARD CAGE
13	H0040	WASHER,STAR,1/8"
14	D9006	NUT,VERO,M3
15	D9021	SPACER,VERO STRAIGHT
16	D9034	CARD GUIDE,VERO
17	D9032	STRIP,TAPPED CONN.RAIL
19	H2004	NUT,1/8"
20	H1068	SCREW,1/8" X 1/2"
21	M3153	CABLE,CMI335,SLOT 9
22	M3152	CABLE,CMI335,SLOT 8
23	M3151	CABLE,CMI335,SLOT 7
24	M3150	CABLE,CMI335,SLOT 6
25	M3149	CABLE,CMI335,SLOT 5
26	M3148	CABLE,CMI335,SLOT 4
27	M3147	CABLE,CMI335,SLOT 3
28	M3146	CABLE,CMI335,SLOT 2
29	300/436	BRACE,ANALOGUE MOTHER BOARD
31	300/355	SUBPANEL,MIXER/CHANNEL/SAMPLER
32	300/367	SUBPANEL,SMPTE
33	300/431	SUBPANEL,MIDI
35	300/414	FACEPLATE,SAMPLER
36	300/354	FACEPLATE,CHANNEL
37	300/351	FACEPLATE,MIXER
38	300/352	FACEPLATE,SMPTE
39	300/365	FACEPLATE,MIDI
40	300/429	FACEPLATE,BLANK
41	300/410	MOUNTS,ANALOGUE CARD TO SUBPANEL,SHORT
42	300/411	MOUNTS,ANALOGUE CARD TO SUBPANEL,LONG

AUDIO CARD CAGE ASSEMBLY
DM3125

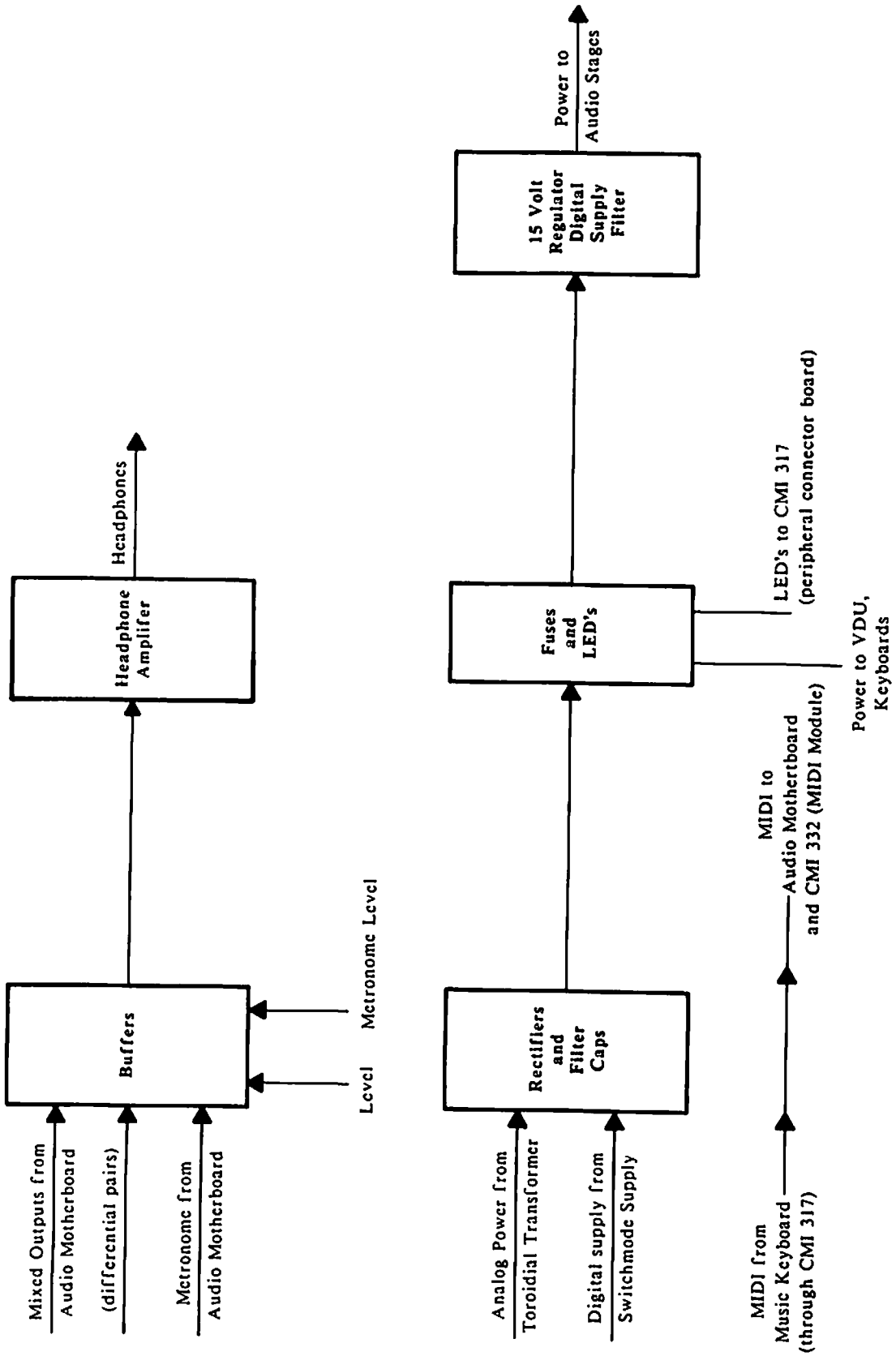
Ancillary Boards 4

Contents

CMI-310 4.2

Audio PSU and Headphone amp

Block Diagram



Terminology
APSU: Analog Power Supply Unit
AR: Audio Rack assembly

Introduction
The Analog Power Supply Unit (APSU) provides the rectified and filtered power to the VDU, Keyboards and Audio Rack components of the CMI. In addition it contains a tracking regulated +/-15 volt supply and a stereo headphone amplifier which gives the user a mixed output for non studio use.

Voltage Regulator
The Audio Rack (AR) unit is supplied with 3 amp regulated +/-15 volt supply by a tracking regulator IC (LM325) and associated bypass transistors. This regulator also supplies the Op-amps on the CMI-310.

Headphone Amplifier
The headphone amplifier is for amplifying the stereo outputs from both mixer modules which may be inserted in the audio rack. It consists of two LM380 IC amplifiers which are fed buffered signals from the audio motherboard. It can also mix the metronome output if desired.

Rectification and Filtering
(*refer schematic CMI310-01*)
The Alternating Current from the toroidal transformer comes in on connector SO1 (15 way). Various windings are used to generate the voltages for the CMI.

Connections 1 and 2 (16vac) are rectified by BR5, filtered by C34 and fused by F8 to supply the music keyboard +18 volts. LED 8, if inserted, confirms the fuse condition. LEDs 1 through 10 are not normally inserted as the CMI has these leds mounted on the rear panel circuit board CMI-317.

Connections 3 and 4 (16vac) are rectified by BR4 filtered by C33 and fused by F7 to supply the music keyboard -18 volts. LED 7, if fitted, shows the fuse state.

Connections 5 and 6 (9vac) are rectified by BR2, filtered by C30 , fused by F6 to feed the music keyboard +10 volts. LED 6 shows the fuse state.

Connections 7, 8 and 9 (+/-18vac) are rectified by BR3, filtered by C31 and C32 , fused by F4 and F5 to supply the Audio Rack 15 volt regulator (see later circuit description). LEDs 4 and 5 indicate fuse condition.

Connections 10, 11, 12 and 13 (+18,+8,common and -18vac respectively) are for the sampling card supply.

Connections 10 and 13 (+18, -18vac with reference to pin 12) are rectified by BR1 filtered by C27 and C28 and fused by F2 and F3 to supply the sampling card +20 and -20vdc supplies. LEDs 2 and 3 show the fuse states.

Connection 11 (8vac with reference to pin 12) is rectified by D5, filtered by F1 to feed the +10 volts DC to the sampler.

Connections 14 and 15 are the 16vac for the VDU which is fused by F9. LED 9 indicates the fuse condition.

Voltage Regulator
(refer schematic CMI310-02)

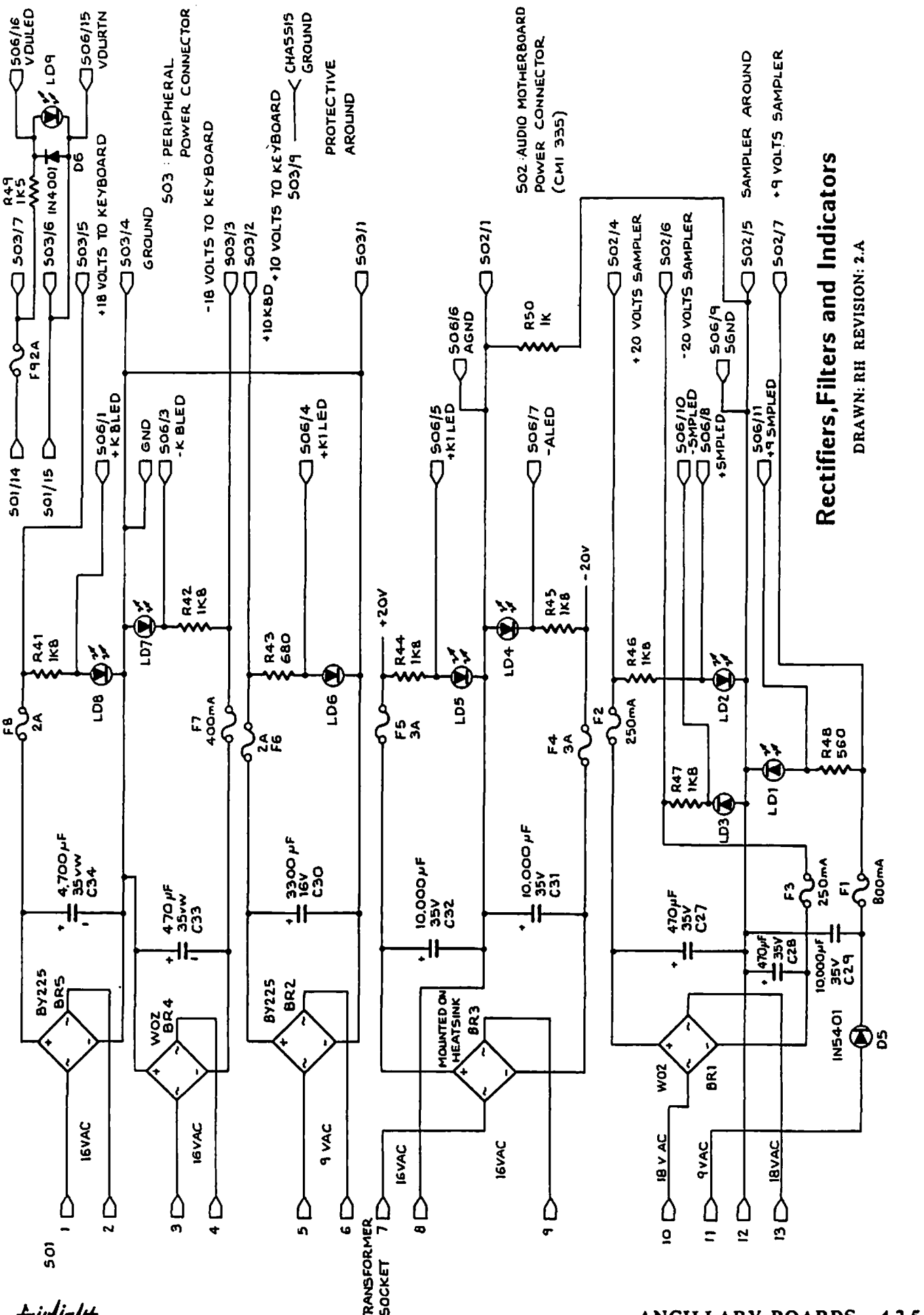
The +/- 15 volt regulation for the Audio modules is provided by the LM325 IC and bypass transistors T1 and T2. Resistor R36 provides the current limit of just over 3 amps for the +15 volt while R39 provides the -15 volt limit. Diodes D7 and D2 protect transistors T1 and T2 from excessive reverse voltage during transient conditions at power up and power down times.

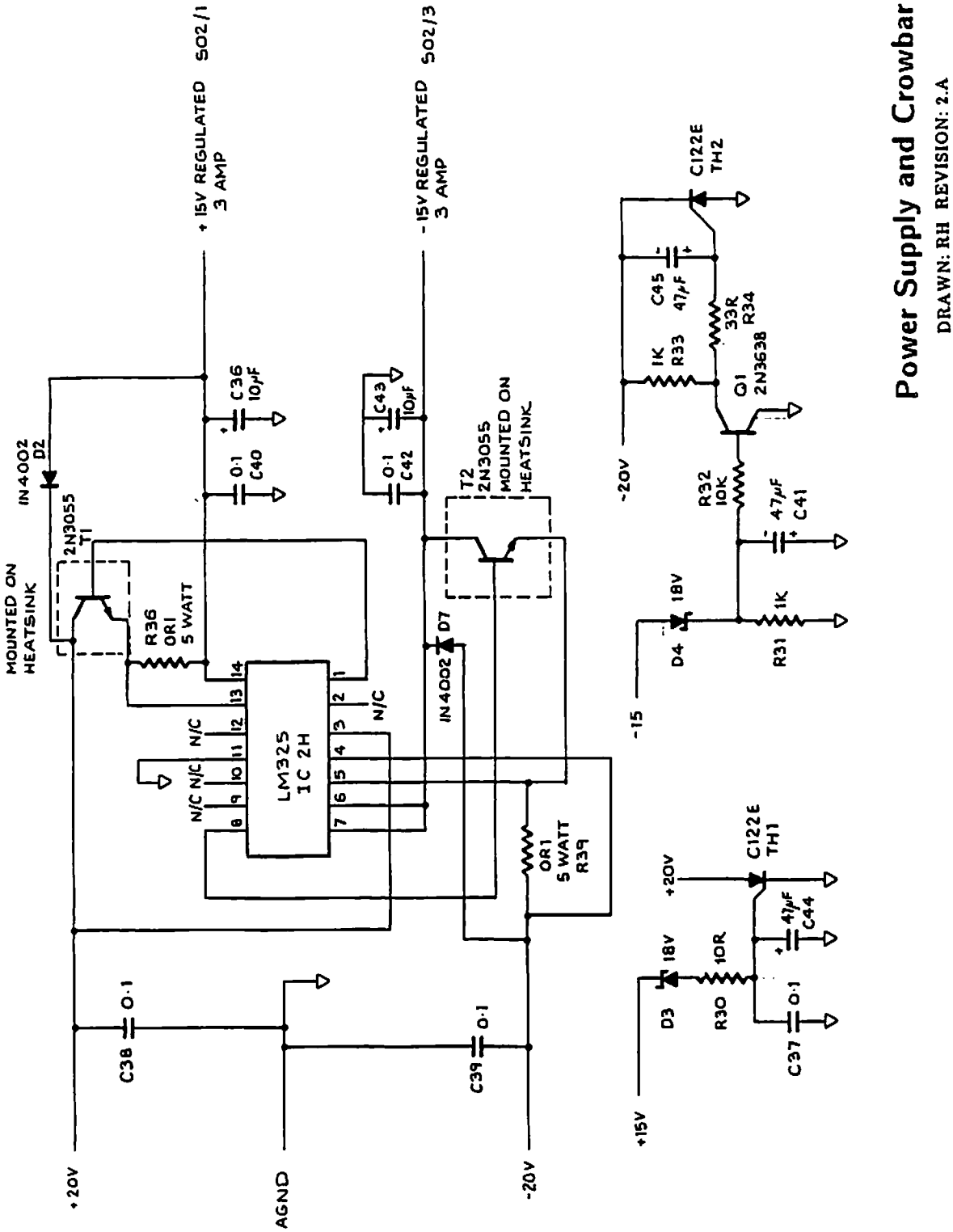
Diode D3 and Thyristor TH1 (C122E) are a crowbar protection circuit for the +15 volt supply to help prevent faulty transistors from damaging audio circuitry. Similarly D4, Q1 and TH2 protect the -15 volt supply from over voltage. These operate by sensing an over-voltage condition and firing the thyristor to blow the fuse. This saves the audio circuitry from damage.

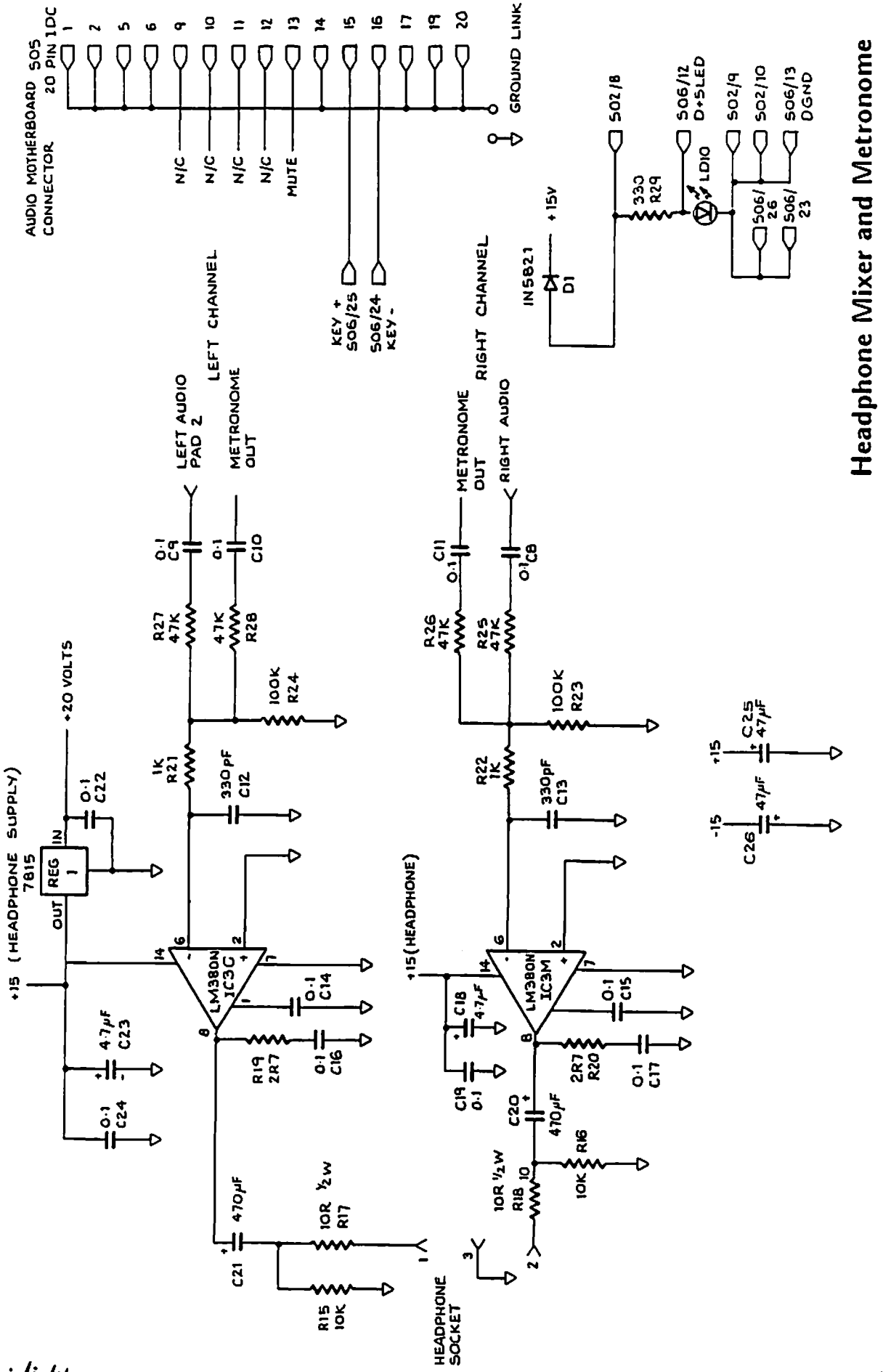
Headphone Amplifier
(refer schematic CMI310-03 -05)

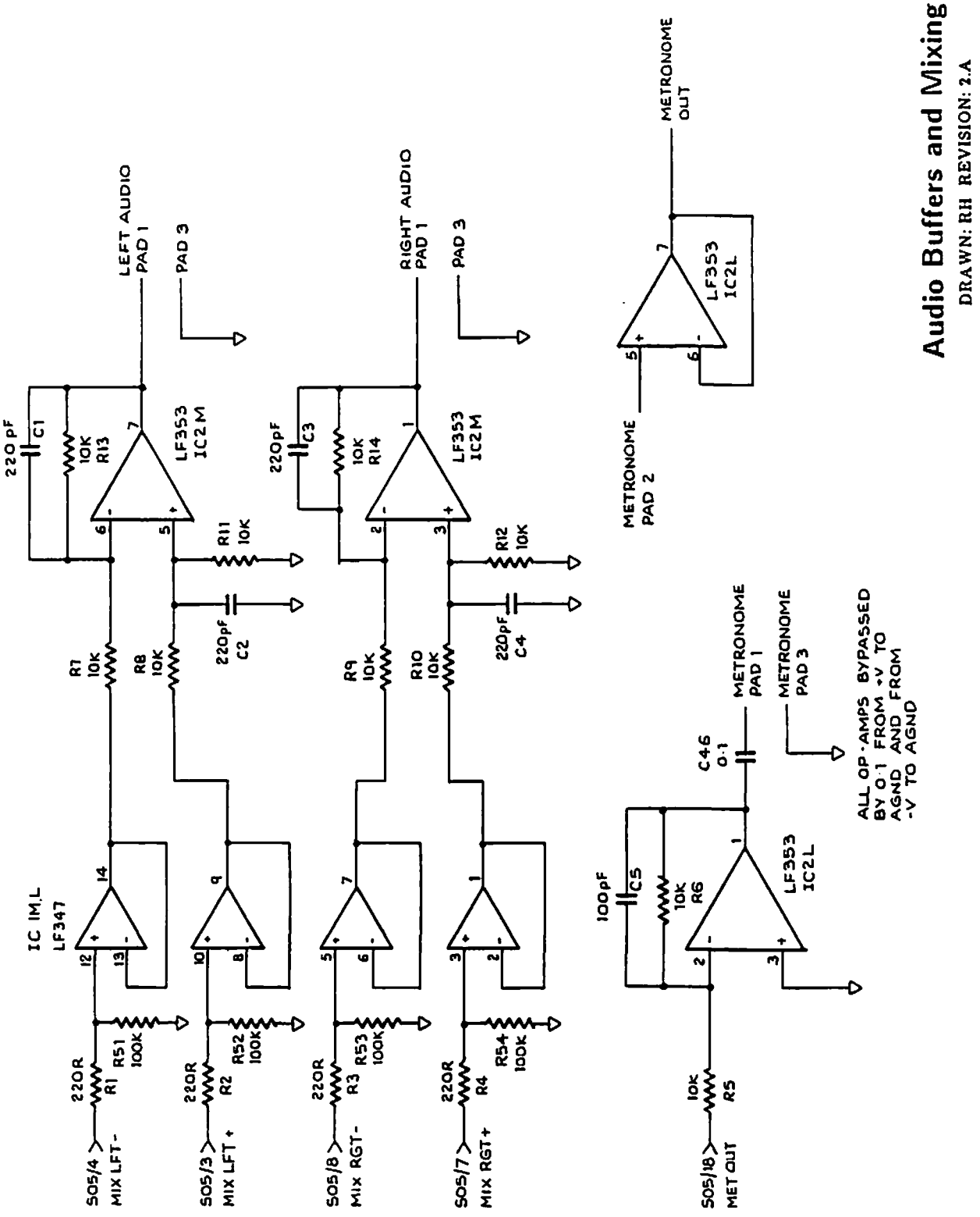
The incoming audio signals are buffered by IC 1M,L (LF347) then converted to unipolar signals and filtered to remove high frequency noise (induced from various sources into the Audio Motherboard and cabling) by IC 2M (LF353). The metronome is buffered by IC 2L then fed to the metronome level pot and again buffered before being mixed into the headphone amplifiers.

The audio signal from the level pot is mixed with the metronome output and passed through a simple filter then amplified by the LM380s (one for right and one for left). The inverting input is used on the LM380 for improved stability with a high impedance source. A 470 uF cap provides DC isolation for the output and a 10R protects from short circuit loads. The 7815 regulator provides a direct low impedance supply for the LM380s to improve isolation and stability.


Rectifiers, Filters and Indicators
DRAWN: RH REVISION: 2.A
503 : PERIPHERAL POWER CONNECTOR
502 :AUDIO MOTHERBOARD POWER CONNECTOR (CMI 335)
PROTECTIVE AROUND
SAMPLER AROUND
TRANSFORMER SOCKET
MOUNTED ON HEATSINK
+18 VOLTS TO KEYBOARD
-18 VOLTS TO KEYBOARD
+10 VOLTS TO KEYBOARD
GROUND
CHASSIS GROUND
+20 VOLTS SAMPLER
-20 VOLTS SAMPLER
+9 VOLTS SAMPLER
16VAC
9 VAC
18 V AC
BY225 BR5
WOZ BR4
BY225 BR2
BR3
W02 BR1
IN5401 D5
F8 2A
F7 400mA
F6 2A
F5 3A
F4 3A
F2 250mA
F3 250mA
F1 800mA
R41 1K8
R42 1K8
R43 680
R44 1K8
R45 1K8
R46 1K8
R47 1K8
R48 560
R49 1K5
R50 1K
4,700µF 35VW C34
470µF 35VW C33
3300µF 16V C30
10,000µF 35V C32
10,000µF 35V C31
470µF 35V C27
470µF 35V C28
10,000µF 35V C29
LD1
LD2
LD3
LD4
LD5
LD6
LD7
LD8
LD9
D6 IN4001
+K8LED
GND
-K8LED
+K1LED
-ALED
AGND
-SMPLED
+SMPLED
SGND
+9 SMPLED
VDULED
VDURTN
501/14
501/15
503/7
503/6
503/5
503/4
503/3
503/2
503/1
502/1
502/4
502/6
502/5
502/7
506/1
506/3
506/4
506/5
506/7
506/10
506/8
506/9
506/11
506/16
506/15
501
1
2
3
4
5
6
7
8
9
10
11
12
13


+15V REGULATED 3 AMP
SO2/1
-15V REGULATED 3 AMP
SO2/3
IN4002 D2
2N3055 T1
MOUNTED ON HEATSINK
R36 OR1 5 WATT
C36 10µF
C40 0.1
C43 10µF
C42 0.1
T2 2N3055 MOUNTED ON HEATSINK
IN4002 D7
LM325 IC 2H
N/C
OR1 5 WATT R39
C38 0.1
C39 0.1
+20V
AGND
-20V
C122E TH2
C45 47µF
1K R33
33R R34
Q1 2N3638
R32 10K
C41 47µF
D4 18V
R31 1K
-15
C122E TH1
+20V
C44 47µF
D3 18V
R30 10R
C37 0.1
+15V
Power Supply and Crowbar
DRAWN: RH REVISION: 2.A


Headphone Mixer and Metronome
DRAWN: RH REVISION: 2.A
AUDIO MOTHERBOARD 505
CONNECTOR
20 PIN IDC
1
2
5
6
9
10
11
12
13
14
15
16
17
19
20
N/C
N/C
N/C
N/C
MUTE
KEY +
506/25
506/24
KEY -
GROUND LINK
502/8
506/12
D+5LED
502/9
502/10
506/13
DGND
330
R29
LD10
506/
26
506/
23
+15V
IN5821
D1
LEFT CHANNEL
LEFT AUDIO
PAD 2
METRONOME
OUT
RIGHT CHANNEL
METRONOME
OUT
RIGHT AUDIO
0.1
C9
0.1
C10
C11
0.1
0.1
C8
R27
47K
47K
R28
R26
47K
R25
47K
100K
R24
100K
R23
1K
R21
330pF
C12
R22
1K
330pF
C13
+20 VOLTS
0.1
C22
+15 (HEADPHONE SUPPLY)
7815
REG
1
IN
OUT
+15
+15(HEADPHONE)
LM380N
IC3G
LM380N
IC3M
0.1
C14
0.1
C15
4.7µF
C23
4.7µF
C18
0.1
C24
0.1
C19
R19
2R7
0.1
C16
2R7
R20
0.1
C17
C20
470µF
10R ½W
R18
10
10K
R16
470µF
C21
10R ½W
R17
10K
R15
HEADPHONE
SOCKET
+15
C25
47µF
-15
47µF
C26


Audio Buffers and Mixing
DRAWN: RH REVISION: 2.A
LEFT AUDIO PAD 1
PAD 3
RIGHT AUDIO PAD 1
PAD 3
METRONOME OUT
METRONOME PAD 2
METRONOME PAD 1
METRONOME PAD 3
IC 1M.L LF347
LF353 IC2M
LF353 IC2L
220pF C1
220pF C3
100pF C5
C46 0.1
10K R13
10K R14
10K R6
10K R5
R11 10K
R12 10K
220pF C2
220pF C4
R7 10K
R8 10K
R9 10K
R10 10K
220R R1
220R R2
220R R3
220R R4
R51 100K
R52 100K
R53 100K
R54 100K
505/4 MIX LFT-
505/3 MIX LFT+
505/8 MIX RGT-
505/7 MIX RGT+
505/18 MET OUT
ALL OP-AMPS BYPASSED BY 0.1 FROM +V TO AGND AND FROM -V TO AGND

CMI-317 4.3
Peripheral Connector Board

Introduction

The Peripheral Connector Card is located on the rear panel of the Series III mainframe assembly at the left hand end. Its function is to interface the power, signals and video input and output from the various parts of the CMI to the external components of a complete system. The card exists only as an interconnection medium and display device for the power supply fuse state (The leds represent the fuse state on card CMI-10)

Detailed Circuit Description

(refer schematic CMI-17-01,-02,-03)

The circuit diagram consists mainly of a schedule of pinouts of the various connectors involved. The input connectors are shown on sheet CMI-17-01, P1 is a 9 pin power connector for the cable from the C.M.I 310 power supply, P2 is an IDC ribbon cable type connector from the C.M.I 310 and carries the LED signals to indicate fuse condition and also MIDI signals from the music keyboard (from CMI-10 MIDI is re-directed up to CMI-32 MIDI module in the audio rack, via CMI-35, and then to CMI28 in the digital card cage for processing), P3 is a 10 pin IDC ribbon cable type connector which carries the serial data to the printer ports (S6 and S7), P4 is a MOLEX type wafer connector which receives the composite video signal to be sent to the monitor.

Outputs are shown on sheet CMI-17-02: S4 connects the keyboards to the CMI (note the connection between pins 5 and 7, this is the only change between rev 1 and 2 however all rev 1 boards were fitted with this as a modification before shipment. S5 is the music keyboard power connector. S6 & S7 are serial printer sockets. S8 is the video out and monitor power connector.

The display LEDs are shown on sheet CMI-17-03, their functions are labelled. Note that the series dropping resistors are located on CMI-310 power supply.

(UTILUX)
P1 POWER

1	10V RETURN
2	+10V
3	-20V
4	±20 RETURN
5	+20V
6	16VAC
7	16VAC
8	N/C
9	PGND

P4 VIDEO

1	VIDEO
2	VGND
3	N/C
4	N/C
5	N/C

(IDC)
P2 INDICATOR

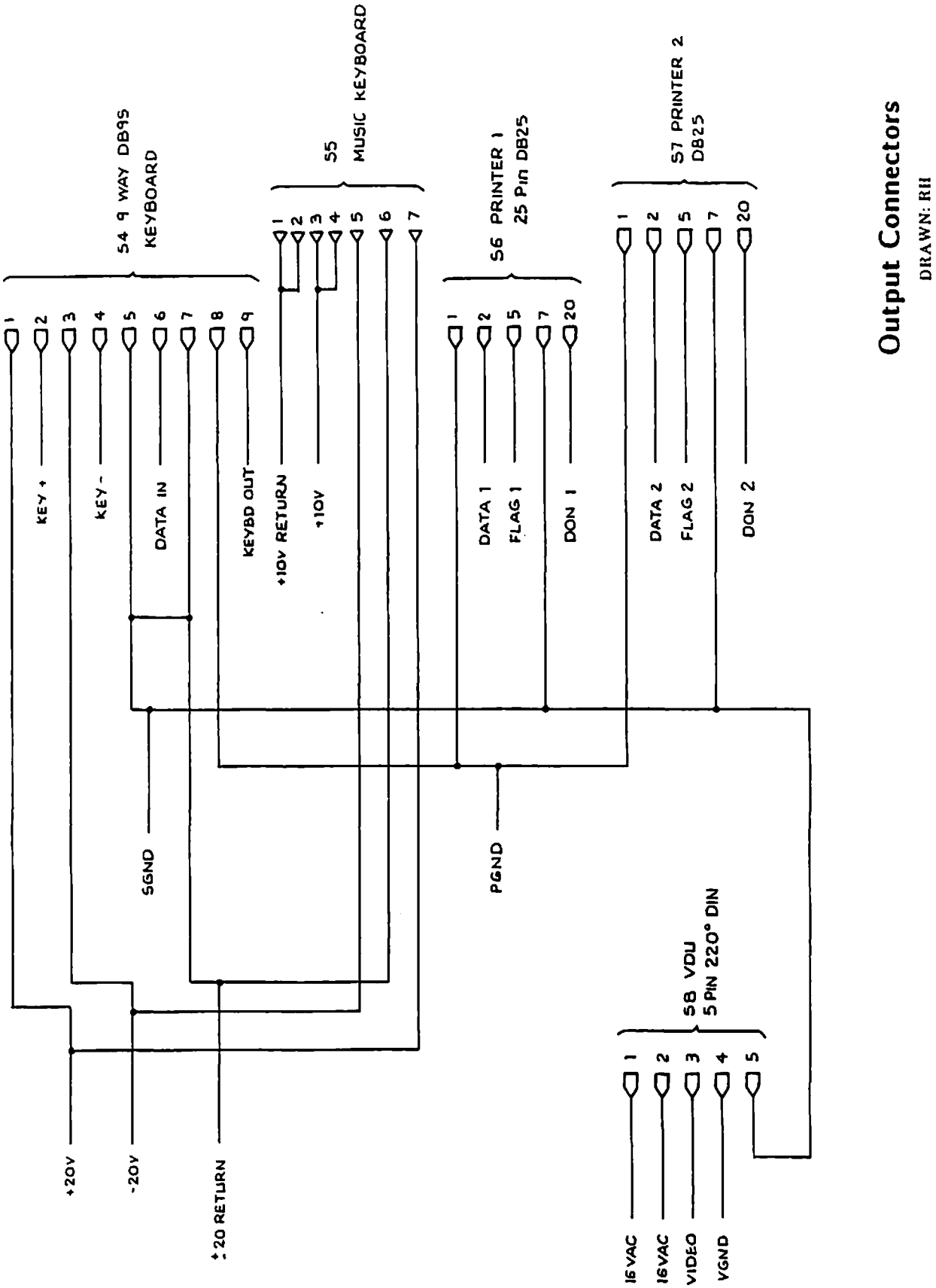
1	+KBLED
2	GND
3	-KBLED
4	+KILED
5	+ALED
6	AGND
7	-ALED
8	+SMPLED
9	SGND
10	-SMPLED
11	+9SMPLED
12	D+5LED
13	DGND
14	N/C
15	VDURTN
16	VDULED
17	N/C
18	N/C
19	N/C
20	N/C
21	N/C
22	N/C
23	DGND
24	KEY-
25	KEY+
26	DGND

(IDC)
P3 SIGNAL

1	DATA 1
2	DON 2
3	DON 1
4	FLAG 2
5	SGND
6	DATA IN
7	KEYBD OUT
8	SGND
9	FLAG 1
10	DATA 2

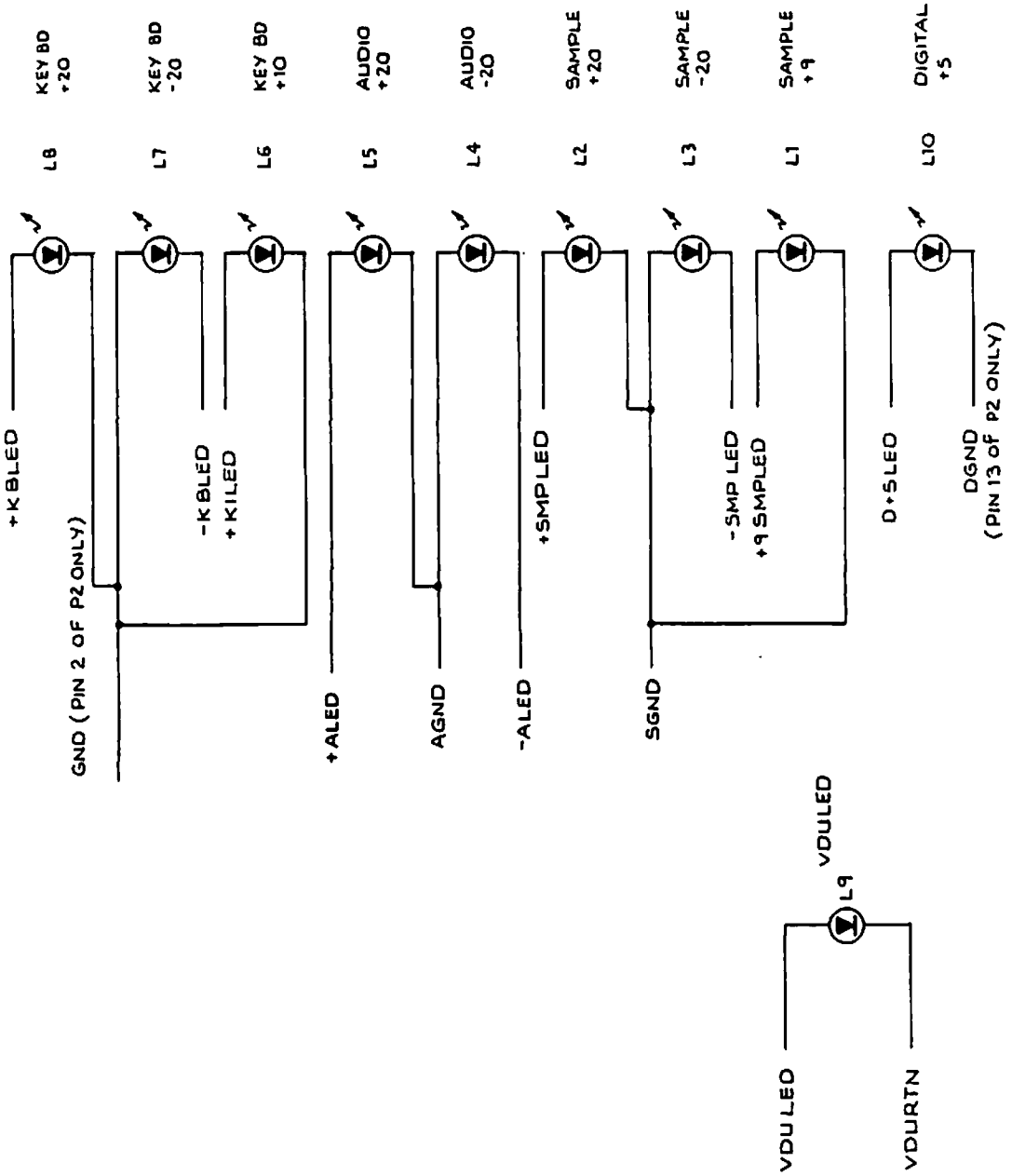
Data Connectors etc., Input

DRAWN: RH


Output Connectors
DRAWN: RH
54 9 WAY DB9S KEYBOARD
1 KEY +
2
3
4 KEY -
5 DATA IN
6
7
8 KEYBD OUT
9
55 MUSIC KEYBOARD
1 +10V RETURN
2
3 +10V
4
5
6
7
56 PRINTER 1 25 Pin DB25
1 DATA 1
2
5 FLAG 1
7
20 DON 1
57 PRINTER 2 DB25
1 DATA 2
2
5 FLAG 2
7
20 DON 2
SGND
PGND
+20V
-20V
±20 RETURN
58 VDU 5 PIN 220° DIN
1 16VAC
2 16VAC
3 VIDEO
4 VGND
5

Led Indicators

DRAWN: RH



Internal Cabling 4.4

Introduction

This cable schedule is for use in reference to the Cable Wiring diagram for CMI Series III Mainframe. The Series III uses IDC type ribbon cabling where possible to provide high reliability and ease of manufacture. This collection of tables gives the signal names of each conductor within the ribbon cable and source or destination of each cable.

Table 1
SCSI Interface Cable
Source: Q777 SCSI Interface Adapter
Destination: ADAPTEC Disk Controller,EMULEX Tape Controller, External SCSI Connector.

Pin	Signal Name	Signal Function	Origin
1	GND	signal ground	Q777
2	DB0	data bus 0	Q777
3	GND		Q777
4	DB1	data bus 1	Q777
5	GND		Q777
6	DB2	data bus 2	Q777
7	GND		Q777
8	DB3	data bus 3	Q777
9	GND		Q777
10	DB4	data bus 4	Q777
11	GND		Q777
12	DB5	data bus 5	Q777
13	GND		Q777
14	DB6	data bus 6	Q777
15	GND		Q777
16	DB7	data bus 7	Q777
17	GND		Q777
18	unused		
19	GND		Q777
20	unused		
21	GND		Q777
22	unused		
23	GND		Q777
24	unused		

25	GND		Q777
26	unused		
27	GND		Q777
28	unused		
29	GND		Q777
30	unused		
31	GND		Q777
32	ATN	attention	Q777
33	GND		Q777
34	unused		
35	GND		Q777
36	BSY	busy	Q777
37	GND		Q777
38	ACK	acknowledge	Q777
39	GND		Q777
40	RST	reset	Q777
41	GND		Q777
42	MSG	message	Q777
43	GND		Q777
44	SEL	select	Q777
45	GND		Q777
46	C/D	control/data	Q777
47	GND		Q777
48	REQ	request	Q777
49	GND		Q777
50	I/O	input/output	Q777

Table 2
Graphics Cable
Source: Q219 Graphics Card (10 pin IDC)
Destination: CMI317 peripheral connector panel (5 pin Molex type)

Q219 Pin	Signal Name	CMI317 Function	Pin
1-6	unused		
7	VIDEO	composite video out	1
8	GND	signal ground	2

Table 3
Serial Port Cable
Source: Q133 CPU Control Card
Destination: CMI317 peripheral connector panel

Pin	Signal Name	Function	Signal Origin
1	DOUT0	Printer 1 data out	Q133
2	DON1	Printer 2 Device On (DTR) output	Q133
3	DIN0	Printer 1 Device On (DTR) output	Q133
4	FLG1	Printer 2 Flag (DSR) input	CMI317
5	DGND	Digital Ground	Q133
6	DIN	Keyboard (RS232) data in	CMI317
7	KBDOUT	Keyboard (RS232) data out	Q133
8	DGND	Digital Ground	Q133
9	FLG0	Printer 1 Flag (DSR) input	CMI317
10	DOUT1	Printer 2 data out	Q133

Table 4
MIDI and LED cable
Source: CMI310 power supply
Destination: CMI317 peripheral connector panel

Pin	Signal Name	Function	Signal Origin
1	+KBLED	keyboard power +20 volts LED	CMI310
2	GND	keyboard power return LED	CMI310
3	-KBLED	keyboard power -20voltsLED	CMI310
4	+K1LED	keyboard power +10 volts LED	CMI310
5	+ALED	audio +20 volts LED	CMI310
6	AGND	audio LED return	CMI310
7	-ALED	audio -20 volts LED	CMI310
8	+SMPLED	sampler +20 volts LED	CMI310
9	SGND	sampler LED return	CMI310
10	-SMPLED	sampler -20 volts LED	CMI310
11	+9SMPLED	sampler +9 volts LED	CMI310
12	D+5LED	digital 5 volts LED	CMI310
13	DGND	digital GND	CMI310
14	n/c		
15	VDURTN	VDU LED return	CMI310
16	VDULED	VDU LED signal	CMI310
17-22	n/c		
23	DGND	digital GND	CMI310
24	KEY-[1]	MIDI from music KBD	CMI317
25	KEY+[1]	MIDI from music KBD	CMI317
26	DGND	digital GND	CMI310

Note 1:The MIDI from the music keyboard is routed from the CMI317 through to CMI310 then to CMI335 and CMI332 where it is transmitted to the MIDI processor.

Table 5
Headphone signals and MIDI cable
Source: CMI335 Audio Motherboard
Destination: CMI310 Power Supply
This cable connects the various signals from plug in cards connected to CMI335 and routes them to CMI310. CMI335 has no electronics self contained.

Pin	Signal Name	Function	Signal Origin
1	AGND	analog GND	CMI335
2	AGND	analog GND	CMI335
3	MIXLFT+	mixed output left channel	CMI335
4	MIXLFT-	mixed output left channel	CMI335
5	AGND	analog GND	CMI335
6	AGND	analog GND	CMI335
7	MIXRT+	mixed output right channel	CMI335
8	MIXRT-	mixed output right channel	CMI335
9-12	n/c		
13	MUTE	mute control line	various
14	AGND	analog GND	CMI335
15	KEY+	MIDI from music keyboard	CMI310
16	KEY-	MIDI from music keyboard	CMI310
17	AGND	analog GND	CMI335
18	METOUT	metronome output to headphone	CMI335
19	AGND	analog GND	CMI335
20	AGND	analog GND	CMI335

Table 6
Floppy disc cable
Source: QFC9 Floppy disc controller
Destination: Floppy disc drive

Pin	Signal Name	Signal Function	Origin
1	unused		
2	GND	Ground	QFC9
3	unused		
4	HDL3	unused	QFC9
5	READ DATA		drive
6	HDL2	unused	
7	WRITE PROTECT		drive
8	unused		
9	TRACK 0		drive
10	unused		
11	WRITE GATE		QFC9
12	unused		
13	WRITE DATA		QFC9
14	unused		
15	STEP		QFC9
16	unused		
17	DIRECTION		QFC9
18	unused		
19	DS4	Drive select	QFC9
20	unused		
21	DS3	Drive select	QFC9
22	unused		
23	DS2	Drive select	QFC9
24	unused		
25	DS1	Drive select	QFC9
26	unused		
27	unused		
28	HDL1	unused	
29	READY		drive
30	unused		
31	INDEX		drive
32	unused		
33	HEAD LOAD		QFC9
34	HDL0	unused	
35	ALT	unused	
36	IN USE	unused	
37	SIDE SELECT		QFC9
38	unused		
39	unused		
40	DISK CHANGE		drive
41	unused		
42	TWO SIDE		drive
43-48	unused		
49	LOW CURRENT	write current control	QFC9

Table 7
MIDI and SMPTE cable
Source: CMI28 General Interface Card
Destination: CMI335 Audio Motherboard
The signals connecting to CMI335 are actually routed to CMI332 and CMI333 which plug in to CMI335.

Pin	Signal Name	Function	Signal Origin
1	MIDI OUT A	MIDI output A	CMI28
2	D+5	digital 5 volts	CMI28
3	MIDI IN A	MIDI input A	CMI335
4	SYNC OUT 1	sync output	CMI28
5	MIDI OUT B	MIDI output B	CMI28
6	SYNC OUT 2	sync output	CMI28
7	MIDI IN B	MIDI input B	CMI335
8	SYNC OUT 3	sync output	CMI28
9	MIDI OUT C	MIDI output C	CMI28
10	DGND	digital GND	CMI28
11	MIDI IN C	MIDI input C	CMI335
12	DGND	digital GND	CMI28
13	MIDI OUT D	MIDI output D	CMI28
14	RESET/START	drum machine control	CMI28
15	MIDI IN D	music keyboard MIDI input	CMI335
16	RUN/STOP	drum machine control	CMI28
17	SMPTE IN	SMPTE time code input	CMI335
18	DGND	digital GND	CMI28
19	SMPTE OUT	SMPTE time code output	CMI28
20	CLICK OUT	click track output	CMI28
21	CLICK IN	click track in	CMI335
22	n/c		
23	CPU HALT	MIDI processor halt	CMI28
24	DGND	digital GND	CMI28
25	CPU RESET	MIDI processor reset	CMI28
26	DGND	digital GND	CMI28

Table 8
Debug cable
Source: Q133 CPU Control Card (Debug Card)
Destination: CMI335 Audio Motherboard.
The signals connected to the CMI335 card are actually routed to CMI334 or other cards plugged into CMI335.

Pin	Signal Name	Function	Signal Origin
1	CB2	Debug PIA line	Q133
2	DGND	digital GND	Q133
3	CB1	Debug PIA line	Q133
4	DGND	digital GND	Q133
5	B7	Debug PIA line	Q133
6	B6	Debug PIA line	Q133
7	B5	Debug PIA line	Q133
8	B4	Debug PIA line	Q133
9	B3	Debug PIA line	Q133
10	B2	Debug PIA line	Q133
11	B1	Debug PIA line	Q133
12	B0	Debug PIA line	Q133
13	DGND	digital GND	Q133
14	DGND	digital GND	Q133
15	A7	Debug PIA line	Q133
16	A6	Debug PIA line	Q133
17	A5	Debug PIA line	Q133
18	A4	Debug PIA line	Q133
19	A3	Debug PIA line	Q133
20	A2	Debug PIA line	Q133
21	A1	Debug PIA line	Q133
22	A0	Debug PIA line	Q133
23	DGND	digital GND	Q133
24	CA2	Debug PIA line	Q133
25	DGND	digital GND	Q133
26	CA1	Debug PIA line	Q133

Table 9
Waveform Data Cable
Source: CMI32 Channel Support Card
Destination: CMI335 Audio Motherboard
The signals connected to the audio motherboard are routed to slots 2 through 9 where they connect to the 8 audio modules.

Pin	Signal Name	Function
1	D0+	bit 0 Least significant bit
2	D0-	bit 0
3	D1+	bit 1
4	D1-	
5	D2+	bit 2
6	D2-	
7	D3+	bit 3
8	D3-	
9	D4+	bit 4
10	D4-	
11	D5+	bit 5
12	D5-	
13	D6+	bit 6
14	D6-	
15	D7+	bit 7
16	D7-	
17	D8+	bit 8
18	D8-	
19	D9+	bit 9
20	D9-	
21	D10+	bit 10
22	D10-	
23	D11+	bit 11
24	D11-	
25	D12+	bit 12
26	D12-	
27	D13+	bit 13
28	D13-	
29	D14+	bit 14
30	D14-	
31	D15+	bit 15
32	D15-	
33	DGND	digital GND
34	DGND	digital GND

Table 10
Data Control Cable
Source: Channel Cards CMI31
Destination: CMI335/CMI331 Audio Modules

Pin	Signal Name	Function	Signal Origin
1	VCA1	VCA control channel 1	CMI31
2	VCA2	VCA control channel 2	CMI31
3	VCF1	VCF control channel 1	CMI31
4	VCF2	VCF control channel 2	CMI31
5	RES1	resonance control channel 1	CMI31
6	RES2	resonance control channel 2	CMI31
7	+15V	regulated supply from audio	CMI331
8	AGND	analog GND	CMI331
9	AGND	analog GND	CMI331
10	-15V	regulated supply from audio	CMI331
11	n/c		
12	n/c		
13	n/c		
14	DCLK+2	data latch signal channel 2	CMI31
15	DCLK-2		
16	DCLK+1	data latch signal channel 1	CMI31
17	DCLK-1		
18	PCLK+1	pitch clock signal channel 1	CMI31
19	PCLK-1		
20	PCLK+2	pitch clock signal channel 2	CMI31
21	PCLK-2		
22	DGND	digital GND	CMI31
23	DGND	digital GND	CMI31
24	n/c		
25	n/c		
26	n/c		

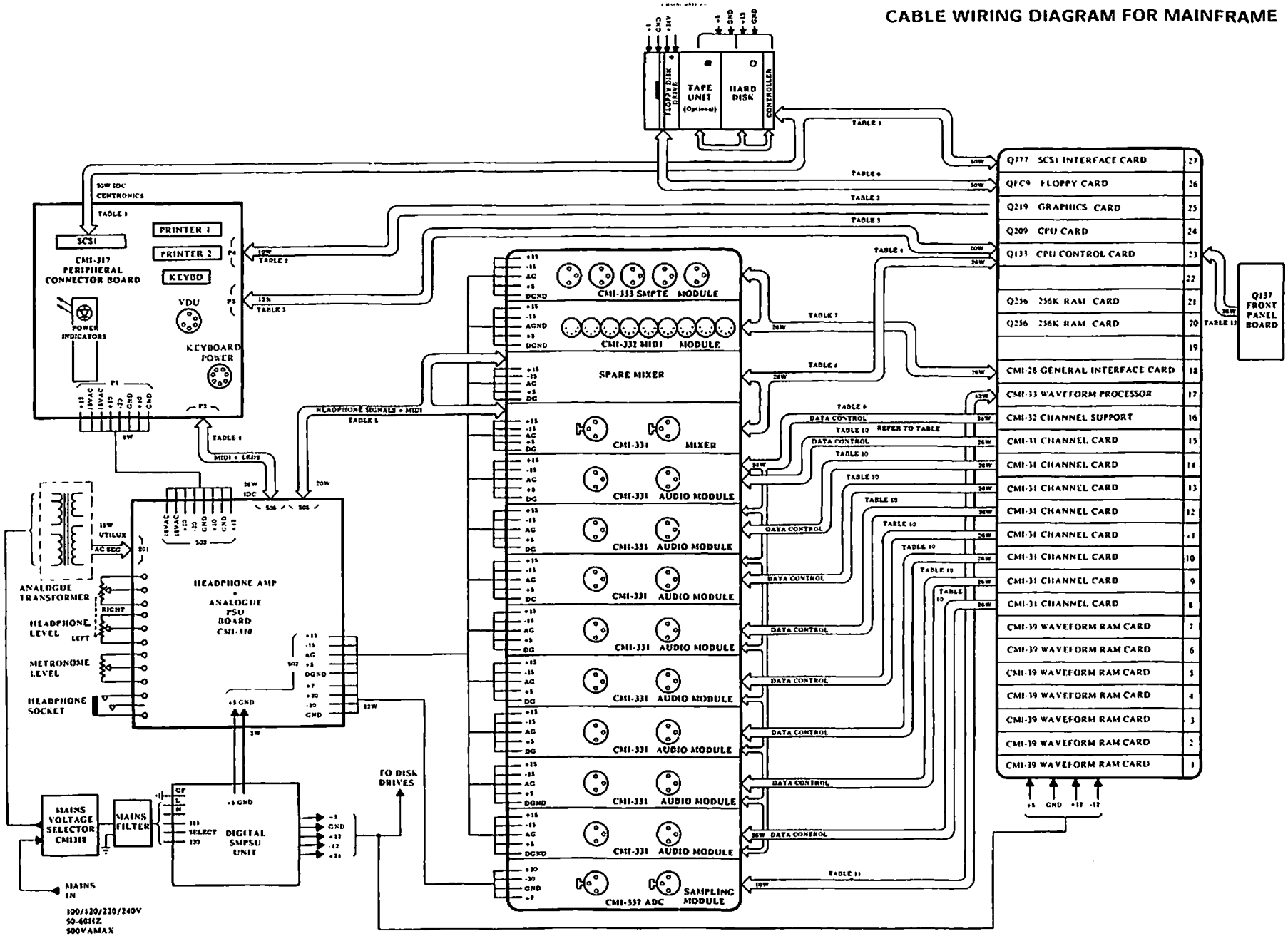
Table 11
Sampling Cable
Source: CMI33 Waveform Processor
Destination: CMI335 Audio Motherboard socket J4, then signals are routed to the CMI337 Sampling Card (ADC) edge connector (slot 1).

Pin	Signal Name	Function	Signal Origin
1	DATA IN+	ADC data current loop	CMI337
2	DATA IN-		
3	CLOCK+	Bit clock current loop	CMI337
4	CLOCK-		
5	EOC+	End of Conversion current loop	CMI337
6	EOC-		
7	DATA OUT+	Control data current loop	CMI33
8	DATA OUT-		
9	START+	Start conversion command loop	CMI33
10	START-		

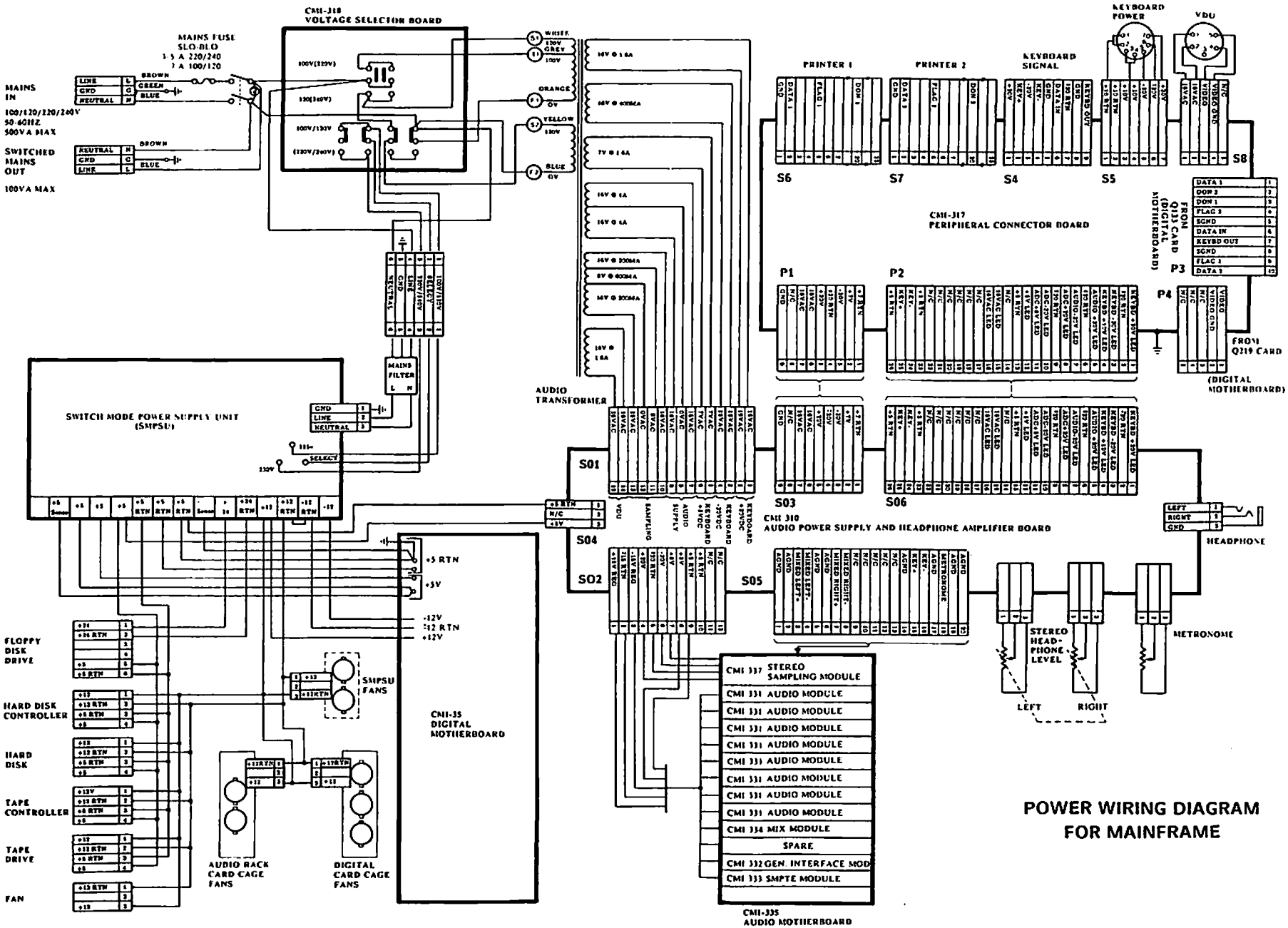
Table 12
Front Panel Cable
Source: Q133 CPU Control Card
Destination: Q137 Front Panel

Pin	Signal Name	Function	Signal Origin
1	GND	Digital ground	Q133
2	RxD	Comms data in	Q137
3	TxD	Comms data out	Q133
4	CTS	Clear To Send to CMI	Q137
5	DCD	Data Carrier Detect to CMI	Q137
6	RTS	Request To Send from CMI	Q133
7	RX+	RS422 data in	Q137
8	RX-		
9	TX+	RS422 data out	Q133
10	TX-		
11	GND	Digital ground	Q133
12	RES1	P1 reset	Q137
13	RES2	P2 reset	Q137
14	NMI1	P1 console interrupt	Q137
15	CMI2	P2 console interrupt	Q137
16	HLT1	P1 halt	Q137
17	HLT2	P2 halt	Q137
18	-12V	-12V to power-on LED	Q133
19	+12V	-12V to power-on LED	Q133
20	+5V	+5V to power-on LED	Q133

CABLE WIRING DIAGRAM FOR MAINFRAME



INTERNAL CABLE SCHEDULE - 4.4.13


CMI-318
VOLTAGE SELECTION BOARD
MAINS FUSE
SLO-BLO
3.5 A 220/240
7 A 100/120
MAINS IN
100/120/220/240V
50-60HZ
500VA MAX
SWITCHED MAINS OUT
100VA MAX
AUDIO TRANSFORMER
SWITCH MODE POWER SUPPLY UNIT (SMPSU)
MAINS FILTER
SELECT
PRINTER 1
PRINTER 2
KEYBOARD SIGNAL
KEYBOARD POWER
VDU
CMI-317
PERIPHERAL CONNECTOR BOARD
FROM Q219 CARD (DIGITAL MOTHERBOARD)
FROM Q219 CARD
(DIGITAL MOTHERBOARD)
S1
S2
S3
S4
S5
S6
S7
S8
P1
P2
P3
P4
CMI 310
AUDIO POWER SUPPLY AND HEADPHONE AMPLIFIER BOARD
HEADPHONE
STEREO HEAD-PHONE LEVEL
LEFT
RIGHT
METRONOME
+5 RTN
+5V
-12V
-12 RTN
+12V
SMPSU FANS
CMI-35
DIGITAL MOTHERBOARD
FLOPPY DISK DRIVE
HARD DISK CONTROLLER
HARD DISK
TAPE CONTROLLER
TAPE DRIVE
FAN
AUDIO RACK CARD CAGE FANS
DIGITAL CARD CAGE FANS
CMI 337 STEREO SAMPLING MODULE
CMI 331 AUDIO MODULE
CMI 331 AUDIO MODULE
CMI 331 AUDIO MODULE
CMI 331 AUDIO MODULE
CMI 331 AUDIO MODULE
CMI 331 AUDIO MODULE
CMI 331 AUDIO MODULE
CMI 331 AUDIO MODULE
CMI 334 MIX MODULE
SPARE
CMI 332 GEN. INTERFACE MOD
CMI 333 SMPTE MODULE
CMI-335
AUDIO MOTHERBOARD
POWER WIRING DIAGRAM
FOR MAINFRAME
INTERNAL CABLE SCHEDULE - 4.4.15

Troubleshooting & Diagnostics 5

Contents

Troubleshooting 5.2

Introduction

The CMI system relies on a complex interaction of hardware and software for proper operation. It is often difficult to differentiate between hardware faults, software bugs and operator errors. Before attempting repair of the Mainframe, establish that the fault is definitely a hardware fault in that unit. If in doubt, try the same series of operations on another system of the same revision level to ensure it is not a software bug and check with the User's Manual to ensure it is not operator error.

Due to the complexity of the system, this manual does not attempt to present an exhaustive list of faults and repair procedures. Instead, the detailed descriptions of the various components of the system should provide service personnel with a thorough understanding of the hardware. This knowledge should allow the general approach to fault finding outlined below to be adapted to isolating particular problems down to the board level.

Extensive diagnostic software is provided to thoroughly test most of the hardware. Of course, in order to run the diagnostics the computer section must be running at least to the stage of loading and executing the software. Use of the diagnostic software is described in detail in section 6 (below).

From time to time, due to the imperfect nature of the world and particularly during the early stages of production of a new machine, design faults emerge which chose not to reveal themselves during the prototyping and pre-production phases. The solution to these problems and the description of the conditions under which they occur are disseminated to all distributors in the form of Field Change Notices (FCNs). In general, FCNs do not need to be implemented on a particular machine unless the fault it cures is in evidence. The exception to this is when a hardware mod or ROM change is required for a new software release or a new hardware option to work properly - this situation would always be indicated on the FCN. Service technicians should always be familiar with the FCNs already distributed so that known problems can be readily recognised and fixed without delay.

A guide to troubleshooting the major components of the Mainframe follows:

Power Supply

There are two separate power supplies supplying a wide variety of voltages to different parts of the system so power supply failure may lead to anything from total system failure down to improper operation of a single component. Refer to the Power Wiring Diagram (at the end of this section) while reading the following description.

The CPU (comprising all circuit boards in the Logic Rack including and to the right of the Q256 System RAM) requires only +5V, +12V and -12V to function. These supplies come from the Switch-Mode Power Supply Unit (SMPSU) via the Logic Motherboard (CMI-35) and three LEDs are provided on the front panel of the CMI to indicate the state of each. The LEDs will not light if the relevant supply drops below two-thirds of the nominal voltage, however 4.8V is the absolute

minimum supply for correct operation of the CPU, so the LEDs are only indicators of gross failure. The best place to measure the +5V supply to the actual boards in the rack is across the 47uF electrolytic capacitors on the Waveform RAM cards, accessible with the lid removed from the machine. Note that the LEDs pick up their supplies via the ribbon cable connected to the Q133 card, so it is possible that a fault in the Q133 could cause the LEDs to malfunction.

Any other form of power supply failure will result in malfunction of specific parts of the system rather than the total system, as follows:

The Floppy Disc drive requires +5V and +24V from the SMPSU. There is no LED indicator for +24V, so a meter should be used to check this supply at the drive.

The Hard Disc(s), Streaming Tape and their associated controller boards are all supplied with +5V and +12V from the SMPSU, with supply conditions indicated by the front panel LEDs. Note that the LEDs only indicate the presence of the supplies on the Logic Rack, so meter checking is necessary to eliminate supply wiring faults between the SMPSU and the mass storage peripherals.

The VDU will only operate if it is receiving its 16VAC supply signal from the Toroidal Transformer, which is the first stage in the Audio Power Supply Unit. The transformer and CMI-310 Audio Power Supply board are mounted in the bottom of the machine to the right of the card cages and SMPSU, and are accessed by removing the bottom panel. One secondary winding of the Transformer is dedicated to the VDU supply which is routed through a fuse on the CMI-310 board then to the CMI317 Peripheral Connector board where a LED on the rear panel indicates the condition of the supply. This 16VAC supply is floating with respect to the rest of the system.

All fans in the system are supplied with 12V from the SMPSU.

The Music Keyboard and Alpha-numeric keyboard receive +20V and -20V unregulated from the CMI-310 Audio Power Supply. These supplies are fuse protected on the CMI-310 and go to the Keyboard socket via the CMI-317 Peripheral Connector where a LED on the rear panel indicates supply conditions.

The Audio Rack receives all its supplies from the CMI-310 except for digital +5V which comes from the SMPSU. Rev 3 CMI-310 modules include fuses for all audio supplies. Rev 2 and below include fuses for all audio supplies except digital +5V. A cable from the CMI-310 carries all audio supplies to LED indicators on the rear panel, mounted on the CMI-317 board. The Audio Modules, General Interface Module, SMPTE Module, and Mixer Module(s) depend upon the regulated +15V and -15V supplies and digital +5V. **The Stereo Sampling Module** has its own independent floating supplies: +20V, -20V and +8V all unregulated, so it is quite possible for a supply problem to cause only sampling failure. "Sample Fault" is the most likely system error message in this case.

The Headphone Amplifier uses +20V and -20V unregulated but is driven by audio circuitry which depends on the +15V and -15V regulated audio supplies.

In the event of a power supply failure, be suspicious that the failure may have been caused by a malfunction elsewhere, or incorrect setting of the mains voltage selector (on the Mainframe rear panel).

CAUTION

High voltages exist within the SMPSU (up to 700V) so servicing these units can be very hazardous.
Fan cooling of the SMPSU is vital and becomes ineffective if the cover over the unit is removed. For safety also, do not operate the CMI with the SMPSU cover removed.

CAUTION

Applying 220V or 240V when the mains voltage select switches are set to 110V will usually destroy the SMPSU.

If power supply failure is traced to the Switched Mode Power Supply Unit, replace the complete unit and return to Fairlight Instruments.

Computer (CPU) Section

A failure in the computer section may result in permanent or intermittent malfunction of the system. Normally a Boot (floppy) Disc is used to start the system which transfers immediately to the Hard disk for loading of the rest of the system software. If this does not complete successfully, try first to boot from a Floppy-only CMI System Disc. This will eliminate the Hard Disc system (both the software on the Hard Disc and the actual hardware) as source of the problem. If you are used to the speed of Hard Disc operations, do not be alarmed by the slow operation of the Floppy-only System.

Failing this, try to load a Diagnostics system floppy disk. This uses the QDOS operating system (same as Series IIX) which is much simpler than the Series III OS-9 System and less demanding on the hardware. If this is successful, refer to section 5.7 which describes how to approach non-fatal digital faults.

Fatal Digital Faults

If the system will not run at all (will not load the diagnostic disk) start by checking the following:

1) ENABLE/SAFE switches on the front panel must be in the Down (RUN) position.
2) Mains voltage selector on the rear panel must be set correctly.
3) Power must be applied, as indicated by the three LEDs on the front panel, and ten LEDs on the rear panel.
4) The System Disk must be in good order (as indicated by testing in another system) and inserted correctly.
5) The floppy drive containing the System Disk should be configured as Drive 0, and the switch on the front of the QFC-9 card in the mainframe should be set in the raised position. (It is easy to accidentally flip the switch when removing the floppy disk ribbon cable.)

If these items are all in order, but the system disk will not load, confirm that the fault lies in the Mainframe by disconnecting everything except the AC supply from it and trying to load the disk again. If successful, find out which connected peripheral is interfering with system operation by trying one at a time. Otherwise, attempt to ascertain whether the CPU is running but unable to access the disk, or the CPU is failing to execute at all. The LOAD SYSTEM DISK IN DRIVE message on the VDU indicates immediately that the CPU does get as far as executing the startup software contained in the CPU Control Card ROMs, so the fault probably lies somewhere in the floppy disk sub-system (see Sec. 5.4). The absence of the message means either the graphics sub-system is not working or the CPU is dead. Even if neither the Floppy disc nor the VDU is operational, a clue to the operation of the CPU can be gained if the Parity Error LED on the front of the Q256 RAM turns on at power up then off again after about 2 seconds, since the clearing of the parity error register is under software control of the Q133 boot ROMs.

If the disk will still not load, the system should be "stripped down" until operation is restored. Remove cards in the following order, attempting to load the disk after each stage:

1) Remove all Waveform RAM cards (CMI-39), all channel cards (CMI-31), Channel Support card (CMI-32),.Waveform Processor (CMI-33), General Interface (CMI-28). With this configuration, the Hard Disk will not operate and SCSI diagnostics will fail, but it is still possible to boot from the floppy.
2) Remove the second System RAM card, if installed, from slot 20.
3) Remove the SCSI Controller (Q777) from Slot 27. Close the PCB link leading to edge connector pin 71 on the wiring side of the QFC-9 board (see "potential red herrings" below).
4) Remove the Graphics card (Q219) from slot 25.

5) Replace the Graphics card and remove the Floppy controller (QFC9, slot 26). Obviously it will not be possible to load the system disc but if the CPU and graphics sub-systems are functional, a "* READY *" message followed by the Processor 2 6809 monitor prompt ":" should appear.

If the fault still appears to be in the CPU itself, substitute the CPU cards with known good spares, one at a time until all have been replaced. If the fault persists, the problem must be in the power supply, floppy disk system, motherboard or cabling. Refer to Section 7 (Motherboard Signals) and Section 8 (External Connections) below, for details of motherboard and external signals.

CAUTION

Potential "red herrings" to fault finding by board elimination

Daisy chaining of DMA (Direct Memory Access) signals means that some boards will not work unless certain other boards are present in the system. Up to Rev. 3 of the CMI-35 motherboards, the order of dependency is:

0. SCSI Controller (Q777)
1. Floppy Controller (QFC9)
2. General Interface (CMI-28)
3. Waveform Processor (CMI-33)

Each item in the list must have all higher items present in the system in order to work. For example if the CMI-28 is absent, the CMI system will abort its startup, and diagnostics involving the Waveform Processor will output "WP not present or won't load" messages because the WP is unable to communicate with the CPU unless it gets its daisy chain signals from the CMI-28. The system can be run without the SCSI Controller by using the $\overline{\text{EDL}}$ output from the QFC9. On Rev 6A and below, this involves repairing the cut in the track going to pin 71A of the edge connector. On later revisions a link option connecting to the same place must be closed, but this link must be reopened or the track recut to install the Q777.

Provision has been made for future systems with SCSI-compatible floppy disk drive or no floppy at all. Linking pins 71A and 72A on the Q777 will result in a new order of dependency:

1. SCSI Controller (Q777)
2. General Interface (CMI-28)
3. Waveform Processor (CMI-33)

Refer to the Digital Motherboard Section 2.15 for further detail.

Floppy Disk System

The Floppy Disk system consists of three sections - the floppy-disk controller card QFC9, interconnecting cables, and the floppy-disk drive. More than one drive may be installed if required.

A fault in the controller card or cabling will generally result in hard disk errors or total failure to access disks. Soft or intermittent disk errors will usually be caused by a faulty or misaligned drive.

A toggle switch mounted on the front edge of the controller card reverses the drive select signals to the drives so that the logical drive numbers can be swapped for diagnostic purposes, although this is obviously useful only if more than one drive is installed. For example, if a disk error is reported during boot-load, the drives can be swapped (switch down) and the system disk inserted in the second drive. If it then loads sucessfully, the first drive is faulty. If the fault persists, the fault is in the controller, cabling or power supply, unless the system diskette itself is faulty.

Refer to the Floppy Disk maitenance section for full details of disk drive maintenance.

Hard Disk System

Refer to the Mass Storage Devices Section 5.

Non-Fatal Digital Faults

Non-fatal faults encompass those which do not prevent the CMI from loading and running at least the Diagnostic System disk. Such faults therefore include minor faults in the CPU section and nearly all faults in non-CPU cards except those which cause the CPU bus to be corrupted or by some other means crash the CPU. Once the Diagnostic System disk is able to be loaded, faults may be isolated by the intelligent use of diagnostic software, board swapping, and observation of the behaviour of the machine under the full CMI System software if it is possible to load that as well.

Troubleshooting using the CMI System Software

When the CPU is starting up the CMI System software it checks to ensure that all peripheral processors (General Interface processor, Waveform Processor, Channel processors) are working. If either the GIF or Waveform processors do not respond correctly the startup procedure is aborted with a message on the console, so these problems are easily detected. Both LEDs on the GIF and WP cards are on after power-up, turn off when the processors are started by the CPU, and one or both come on again if the System load aborts. LED functions are explained later.

When starting up Channel Processors, the CPU first does a quick test on each Channel's on-board memory to see if the Channel is installed (and working). It then loads Channel driver software into the memories of the channels present, releases the Channel Processors

from reset (RUN LED switches on), then checks for sensible communication with them. If a Channel Processor, whose memory test worked, does not respond, the startup procedure is aborted with a message. The usefulness of abort messages is a function of software release level and will become more informative in later releases. Messages mentioning "mptask" and "wptask" refer to the General Interface and Waveform processors respectively.

If the CMI System software loads successfully it can be further used to investigate digital faults. For example, if the sequencers play music successfully but notes played on the keyboard have no effect, the General Interface card must be 95% working since it has a major role in running sequences. The fault may lie somewhere in the path from the keyboard to the GIF card - the General Interface Support module in the Audio Rack, the I/O interface of the GIF card, interconnecting cables, or the keyboard itself (see MIDI Frame Display section below). Another example is the case of distorted or crackly waveforms - a problem which is common because there are so many possible causes in both the analog and digital sections of the machine. The following clues can be used to isolate the problem:

i) If the clicks or crackles always sound the same between successive key presses or between successive loads of the same sound then the problem may be in the sound sample data (several sounds in the first release of the Fairlight Library were prone to this problem).

ii) If it is observed that placing the sound in different places in Waveform Memory (by sampling or loading other sounds first) causes the problem in only one place in memory, then it is highly likely that the Waveform RAM card corresponding to that area is at fault. Do not jump to conclusions too soon though, as the Waveform Processor or a Channel card may be having problems addressing that area.

iii) If it is found that sampled sounds are OK but sounds from disk are bad, then the Waveform RAMs and Channels are innocent but suspect the Waveform Processor, the SCSI controller, the disk drive, and the voice data itself on disk.

iv) If a defective voice is overwritten with an internally generated sine wave which then plays without problem, then the SCSI controller, disk drive, or voice data are suspect for sounds from disk, or the Sampling Module is faulty in the case of sampled sounds. The Waveform Editor is very useful in examining whether waveform errors are in Waveform RAM or only occur when the sound is being played.

MIDI Frame Display

It is possible to generate a display of raw MIDI data received by the CMI. If, for example, sounds can be played from the sequencer but playing on the music keyboard has no effect, the display can verify whether any data is being received from the keyboard. The operation

of controls such as modulation wheels and switches can also be checked. Future software releases may have a built-in command which produces a MIDI frame display without the complicated procedure above. Refer to the user manual to find out if this command is available.

To obtain the display without a special command, boot the CMI system normally then follow the procedure below. System prompts and messages are in italics, you type the text in bold. Explanatory notes are on the right in normal print.

$	Call up an OS-9 Shell
Shell	
# com68 -r -i -m	Run program to talk directly with GIF processor
68k Communications Program v2.8 - type '?<cr>' for help	
<cntrl-N>	Interrupt GIF processor
.	
.	3 lines of 68000 internal register dump
.	
K: **80200/*00* ff<cr>**	Set software flag for screen output
K: **p**	Instruct GIF processor to proceed

From this point each incoming MIDI data frame relating to a key or control movement etc. will be printed on the screen and will look something like

pQ:01903c4f

The data above is what you get if you depress middle C with a moderate key velocity. If anything at all gets printed then the music keyboard and all the connections between it and the GIF card are obviously working so the actual data does not need careful attention. Just in case you are interested however, the "pQ:" means the data is in the "Input Play Queue", the following "01" means it is going to operate on instrument number 1, and the last six characters are the hexadecimal values of the 3-byte MIDI frame.

When you have finished looking at the MIDI frame display, return to normal operation as follows -

<cntrl-N>	Interrupt GIF processor again
.	
.	3 lines of 68000 internal register dump
.	
K: **80200/*ff* 0<cr>**	Set flag back to normal
K: **p**	Proceed
<cntrl-C>	
command: **x<cr>**	Exit communications program
# <ESC>	Exit shell, return to CMI system.

Channel Allocation Scheme

In the process of testing using the System software, it is often desirable to play a certain channel or channels. This requires an understanding of the channel allocation scheme. When creating voices, the lowest unused channels are used. When playing a voice with Nphony greater than one, the longest unused channel is played (note that this is very different from the Series I/IIX scheme). Channels equally unused are cycled through numerically.

Thus to test all channels in succession, create a voice (the in-built sine voice will do for go/no-go tests) with an Nphony of 16. Playing a single note repetitively, careful to release each note before beginning the next, will then cycle through in numerical order. Playing a chord will permanently upset the order, although all channels will still be cycled.

To test a single channel continuously, create a voice with sufficient Nphony to occupy all lower channels, then create a voice with Nphony 1 for the test channel and select that voice to be played by the keyboard.

Avoid using sounds with velocity sensitive level controls for this kind of testing, or turn Keyvel control off, otherwise unconscious variations in key velocity will cause confusing differences between channel levels.

Using the CMI System software for troubleshooting is not as exhaustive as running diagnostic chain tests but is often much quicker, and the shrewd technician with a full understanding of the function of each component in the system can use it to narrow down the fault before turning to the specific diagnostics. LED indicators on the front of many boards have been included also for this purpose. The rest of this section briefly describes the kinds of digital faults each non-CPU card can produce and the function of the LED indicators.

General Interface Card - CMI System load abort, failure to play notes from the keyboard, failure to play sequencers, improper operation of SMPTE and metronome.

Note: if the 26-way MIDI/SMPTE ribbon cable which connects from the Audio Motherboard to the front of the General Interface card is unplugged or the CMI-332 MIDI module, excessive spurious interrupts to the GIF processor may cause the sequencer to slow down, operate intermittently, or stop completely. It is therefore advisable to always have the CMI-332 plugged in and the cable securely attached.

LEDs: Processor HALT and RESET. HALT is nearer the front. Processor is halted when HALT light is on; reset when both are on.

Waveform Processor - CMI System load abort, bad waveforms in memory, sampling problems.
LEDs: Processor HALT and RESET. HALT is top LED. Processor is halted when HALT light is on; reset when both are on.

Channel Support Card - Bad waveforms in memory, correct waveform in memory but bad waveform data reaching audio modules, failure of one or more Channel card to operate properly or at all, incorrect or absent control voltages from all Channel cards to Audio Modules.

If Channel Card operation faults remain the same when Channel cards are swapped around or replaced with known good ones, then the Channel Support is most likely to be at fault, although faulty edge connector or motherboard is possible.

Channel Cards - CMI System load abort, failure to play certain notes, bad waveforms from one channel, corrupted waveforms in memory.

LEDs: Four, from top, "A" side playing, "B" side playing, Channel processor RUN (ON = not reset), FIRQ (Fast Interrupt Request from Channel Support; ON = requests being serviced). The power-on state of the FIRQ LED is undefined but it should always be on when the RUN LED is on.

Note that a faulty channel may cause a fault which appears to be a particular Waveform RAM card unless different sounds are loaded to make the channel play in more than one 2 megabyte space.

A faulty Channel card may cause other channels, especially those to the left of it, to also appear faulty due to daisy chaining of Waveform Bus allocation signals from one channel to the next. Any channels to the left of a gap will be unable to generate sounds.

Failure of both "sides" of a channel card to play may be due to the Channel, the Channel Support card, or the GIF but failure of only one side is almost certainly the Channel card itself, an Audio module problem or a faulty cable between the two.

Waveform RAMs - Bad waveforms in memory.

If it is suspected that a particular RAM card is at fault, swap with a known good card or change the assignment of RAM card numbers using the on-card DIP switches. If the bad waveforms still occur on the same card but with different sounds and different addresses, the fault is in the RAM card itself or the motherboard slot. Move the RAM cards around in the last 7 slots of the rack to eliminate the latter. The physical position of Waveform RAM cards is unimportant to the system since space assignment is achieved with the 4-pole DIP switch. By convention, however, cards are placed consecutively, starting with Card 1 nearest the channel cards.

Trouble Shooting using the Diagnostic Software

The diagnostics and their use are fully covered in a separate section. It suffices to state here that the diagnostics can be used individually to find more information about a well-localised fault or chain tests can be run to test the whole system automatically. Chain tests are also available to facilitate continuous testing of specific parts of the system.

The Transition from Digital to Audio

If all diagnostics pass and the CMI System software appears to operate correctly yet sounds are not being played correctly it would seem reasonable to proceed to the Audio Rack to investigate the fault. But first it is necessary to ensure that all signals flowing from the Digital Rack are reaching the Audio Rack, since it is not possible for the CPU to internally test the interface components or the cables between the two racks.

Audio Module Inputs

First attempt to isolate the fault as a Channel card or Audio Module problem by systematic swapping, preferably with known good boards or with other boards in the system. If the Channel card is eliminated, the problem must be either in the Audio module, power supply, or the interconnections.

Each Audio Module receives digital waveform data from the Channel Support Card, and Data Clocks, Pitch Clocks, and Control Voltages from its corresponding Channel Card. These signals are most easily checked by placing the Audio Module on an extender card and using an oscilloscope. Refer to the Audio Module schematics for where to look for each signal. Waveform data is bussed along all modules so check it first if no modules work, by playing any note and examining the changing data on the outputs of the differential receivers. The clocks and control voltages appear only when the corresponding Channel Card is playing (make sure you are looking at the correct half of the Audio Module).

If any input to the Module is missing or incorrect, check back along the Audio Motherboard, ribbon cables, to Channel Card or Channel Support output buffers.

Mixer Controls

The digital control signals for the Mixers can be checked automatically by replacing the cable from the Q133 to the Audio Motherboard with a PIA shorting plug and running the PIA test in the DBTST diagnostic (see Diagnostics section). If this passes, the problem must be in the cable, the Audio Motherboard, or the Mixer itself. The latter can be eliminated by board swapping.

Stereo Sampling I/O

The Sampling Module receives control data and sends digital waveform data and clocks via an optically isolated serial link to the Waveform Processor. The "ADC timeout" message from the CMI System software indicates that a Start Conversion pulse was sent to the Sampler but no End-Of-Conversion pulse came back.

The opto-isolated link is a current loop system so it is difficult to see any signals on the cable. Use an oscilloscope to check output signals on the open collector gate inputs, and input signals on the opto-isolator outputs on both the CMI-337 and CMI-33. Refer to the schematics in both cases.

Waveform Diagnostic Card

Fairlight Instruments will make available to Service Centers a Waveform Diagnostic Card which will intercept all outputs from the Digital Rack to the Audio Rack and interface them back to the CPU. This will allow automatic testing of all the above signals and hence positive isolation of any fault between the two racks. Full documentation of the card plus diagnostic software will be available when deliveries begin.

Audio Faults

In general, if a fault occurs in the Audio rack, it is easy to isolate which module is responsible. This section briefly describes some of the more common audio problems which can be fixed without resort to circuit board exchange.

CMI-331 Audio Module

Distorted or "crackly " waveform from DAC output - may be caused by damaged DAC (MP7616 DACs are severely static sensitive) or faulty logic circuitry preceding the DAC. (The DAC output is current mode, so its output must be observed at the output of the current-to-voltage op-amp circuitry). Errors in the most significant bits of the DAC or its driving logic can easily be seen on an oscilloscope, while errors in the least significant bits appear as distortion components on the output of a distortion meter.

If the DAC output is correct but the VCF produces nothing but a buzz, recalibration of the VCF is required. Too high a Q setting can result in latch-up of the VCF with occasional negative spikes. Incorrect frequency adustment of the VCF will cause sounds to be duller than they should be, or allow digital "grit" to get through to the output. Refer to the diagnostics description for the calibration procedure.

VCA miscalibration can result in incorrect output level and clipping. Serious miscalibration can even cause an apparently square-wave output from a sine waveform. Again, refer to the diagnostics description for the calibration procedure.

Cross-talk between the two channels on each audio module or strange output levels can be caused by incorrect connections to the audio outputs of the machine. All audio channel and mixed outputs are balanced, with a low output impedance (typically 33 ohms). If driving single-sided lines, leave the unused outputs open.

CMI-334 Mixer Module

Clipping of the mixer output can occur if multiple channels are played with highly coherent waveforms. The gain of the mixer is set so that a reasonable level is produced when only one channel is playing. However this means that clipping will occur if all 16 channels play, for example, in-phase sine waves. A -20dB cut switch is mounted on the mixer module, accessible from the rear panel, to avoid this problem. The switch has no effect on any other outputs.

The mixer output may appear to be rather noisy when referenced to one channel driving it - a typical SNR in this situation is -67dB. However with reference to the mixer's full output of +4dBm, achieved by having several channels driving it in phase, just below the onset of

clipping, the SNR should be more like -90dB which will be the sum of all the individual audio module noise levels. The -20dB cut switch will also reduce the absolute noise level by 20dB so that the SNR will remain the same.

CMI-337 Stereo Samping Module

Poor performance of the ADC at high sampling frequencies (above 50kHz) can result if the 20kHz brickwall filter is not switched out. Software versions 2.03 and beyond take care of this automatically by preventing too high a sample rate.

Revisions 1 and 2 of the CMI-337 required trimming of the DC offset to the ADC input. Refer to the detailed information on the CMI-337 for the calibration procedure.

Glitches around the zero crossings of very low level signals, and partial or complete failure of the digital timing circuitry can also result from miscalibration of the module.

Since the Sampling Module has its own independent (and floating) power supplies, complete failure may also be the result of a power supply problem. Even if the LED indicators on the rear panel are lit, check that the supplies are actually getting to the module.

Several known problems which occured on early revisions of Audio Rack modules have been the subject of Field Change Notices. To avoid re-inventing the wheel at best, and at worst, much frustration and customer grief, always check the FCNs as soon as the basic symptoms of a fault have been established.

SCSI Interface

All mass storage (excluding the floppy system) is accessed through the Q777 SCSI adapter which connects to various controllers via a 50 way ribbon cable. A failure in this card will render all SCSI devices inoperative. Normally a system will have a hard disc and a tape unit attached to the SCSI buss, so after booting from a floppy disc try to access both these devices. TP DIR with a good tape will soon show whether the tape system is working. If no SCSI device appears to work then check the following.

1. Thoroughly check SCSI cable.
2. Check all attached controllers have power.
3. Check external SCSI conection.
4. Replace or run diagnostics on Q777.

Hard Disc System

If the Hard Disc system is faulty, it may not be possible to access any files through OS9, or there may be a fatal error during operation. The problem could be a controller fault, drive fault or simply a cable problem.

First try to eliminate simple faults by checking all cables associated with the disc unit :-

* power to drive
* power to Adaptec
* SCSI to Adaptec
* both cables from adaptec to drive

Next, replace the Adaptec controller (if possible) and retry.If the fault persists then the Hard disk unit is faulty and must be replaced.

Streaming Tape

As with the Hard disk check all cables and power supplies. The tape controller card has a green LED visable from the rear and if the controller is functioning this LED will flash continuously. The tape transport operation may be tested by inserting a tape into the drive and closing the door. The tape should reposition, the heads should move and finally the red LED at the front of the tape unit should come on. Any radical departure from this behaviour suggests a faulty tape unit.

If the Tape system operates but returns "TAPE ERROR 3", then check the following :-

* faulty tapes
* dirty heads

If "TAPE ERROR 3" persists then the tape drive requires replacement.

Diagnostics 5.3

Introduction

This section describes the set of diagnostic test programs available on the Diagnostics Disk for testing and debugging all the plug-in circuit modules of the Fairlight CMI (Series III).
Conventions used in this documentation:
$nnnn = hexadecimal notation (where n = 0 to F)
[ret] = return key, used in terminating most commands.

Running the Diagnostic Disk

The diagnostics run under the QDOS operating system. They are therefore supplied on a QDOS system disk which should be loaded into the CMI instead of the usual CMI system disk immediately after power-up or RESTART.

As soon as the disk drive door is closed, the CMI will load QDOS automatically and display a sign-on message giving version and revision numbers.

The QDOS prompt will then be displayed. This is just an equals sign "=" on a line of its own. This indicates that the computer is ready to accept a command.

For further details of the QDOS operating system features, refer to the QDOS User's Guide.

Chain File Diagnostics

Complete diagnostic tests may be run on basic CMI hardware by using chain files. Chain files effectively combine all the individual test programs relevant to a major test. This eliminates typing in each individual test.

Chain files are invoked by typing CHAIN followed by the name of the chain. For example, to test the whole system, type CHAIN TEST [ret].

The following is a list of the chain files currently in use.

CHAIN	FUNCTION	NOTES
TEST	checks most of the system firstly with overall interrupt tests then tests the Q133 debug card, the Q219 graphics card, the Q256 RAM cards, the CMI33 waveform processor, the seven CMI39 waveform RAM cards, and the CMI28 general interface card. Some general CMI31 channel card tests and finally some disc drive tests.	
WAVERAM	checks the waveform RAM cards for memory errors with channels looping	

See Chain Tests for more details about theses chain files.

Running the individual Test Programs

Most of the diagnostic programs make use of a common command interpreter for operator control, so there is a uniform command syntax employed throughout, and several test options available as standard. Tests may be run by typing

<test name>[,<option1>,<option2>...,<optionN>] [ret]

where the <test name> is as described in each section. Options are of the form

<O>=n where <O> is a single character and n is an integer.

Standard options are;

P=n Repeat test command n times. Using "C" instead of an integer initiates continuous testing.

N=n Select test number n. There are usually several test commands with the same name. By default, all tests are executed sequentially but single tests or subsets of the available tests can be specified.

For example	N=1	Test no. 1 only
	N=1,3,5	Tests 1, 3 and 5
	N=4-7	Tests 4 to 7 inclusive

Some tests, which require a waveform to be observed, wait for the spacebar to be pressed before terminating or proceeding to the next test.

To obtain a reminder of what tests are available from the current test program being run, type

LIST [ret]

To repeat the last test, just type

R [ret]

If an error condition occurs, a moderately helpful message is printed on the console and the program returns to the command interpreter or continues testing depending on the setting of the errors option.

Some tests have error handling options which allow testing to continue with or without error messages being displayed.

Successful tests terminate without comment and return to the interpreter or proceed to the next test as soon as completed. Certain tests require the user to check waveforms with an oscilloscope and will not terminate or proceed to the next test until the spacebar is pressed.

Few diagnostics are available for testing this card (Q-133) since it is not possible to load the DOS and run a test without most of the card functioning correctly, some peripheral functions are tested by the program DBTST.CM, which may be run by typing

DBTST [ret]

with the CMI diagnostics disk in drive 0.

The standard command interpreter is used so the LIST and R commands, and P and N options are available as described in the general introduction. Tests are run by typing

<test name>[,<option1>,<option2>...,<optionN>] [ret]

User Peripheral Interface Adapter

The PIA tests require a special test plug to be inserted in the user PIA socket (the one nearest the top of the board) which has the effect of connecting the A side of the PIA to the B side. Connections are as follows:

1 - 24, 2 - 25, 3 - 26,5 - 22, 6 - 21, 7 - 20,
8 - 19, 9 - 18, 10 - 17,11 - 16, 12 - 15.

Test Name: PIA
No. tests: 6

Test No.1 *PIA B/Send A/Receive*
Test No.2 *PIA A/Send B/Receive*
Purpose: With the test plug connecting the two sides of the PIA, A should be able to write to B and vice-versa.

Both tests check each side of the PIA individually first by defining the side as all bits inputs, changing them to outputs, then writing 0, $FF, $FE etc. down to 0 again to the data register and reading back to verify each write.

Test no. 1 then sets side A as all inputs and side B as all outputs and writes all values from 0 decrementing back to 0 to side B, AND reading from side A. Test no.2 does the same in the opposite direction.

Test No.3 *CB2/Send CA2/Receive -ve*
Test No.4 *CB2/Send CA2/Receive +ve*
Test No.5 *CA2/Send CB2/Receive -ve*
Test No.6 *CA2/Send CB2/Receive +ve*
Purpose: Interrupt inputs/control outputs check.

The signal specified by "Send CX2" is configured as a control output whose state is determined by CRB-3 in the PIA control register. The other signal is configured as an interrupt input which will set the interrupt flag in the control register on an edge whose direction is indicated by the "Receive +ve or -ve".

The transmit end is first set to the state which will allow the wrong transition to cause an interrupt, (i.e. if the interrupt receiver is +ve edge triggered, the transmit end is set high) and the flag is checked as clear. Then the transmit state is toggled and checked again as still clear. A second toggle should trigger the interrupt flag. The actual IRQ output is disabled during the test.

System Timer

Test Name: TIM
No. tests: 2

Test No.1 *System Timer Read/Write Latch*
Purpose: Check ability to communicate with timer.

The 3 timers in the 6840 timer are put into the preset state and all numbers from zero to $FFFF are written to the timer 1 latch Each write is followed by a timer read for verification. Timers 2 and 3 are tested in the same way. Each write/read is a double byte transfer through the 8-bit bus.

Test No.2 *System Timer Internal Clock Timeout*
Purpose: Check timeout operation under internal clock.

The timers are clocked by the internal clock. All three timers are initialised to $FFFF then started. A software timing loop is used as reference, and the timer status is continually polled for premature timeout. Timeout must occur within a certain tolerance before or after reference timeout. Clock outputs are enabled during the tests.

Interrupts

Test Name: INT
No. tests: 1

Test No.1 *Interactive Interrupt Test*
Purpose: Test interrupts.

Enable interrupts, then display any which occur. Exit when <ESC> typed. Interrupts may be induced by user poking interrupt lines with an earthed probe.

Asynchronous Communications Interface Adapter

Test Name: ACIA
No. tests: 1

Test No.1 *Comms ACIA Send/Receive 19200 Baud*
Purpose: Tests asynchronous transmit/receive.

Checks that DCD and DSR status bits follow RTS, character is sent and received, interrupt flag set after character is sent, parity error and framing error. This test requires a 10-way loop back plug to be inserted.

Test Name: LATCH
No. tests: 1

Test No.1 *Comms Latchup Protect*
Purpose: Test if ACIA latched up (usually on power up).

Transmits a character and goes into a software timeout loop. Error message displayed if character not received.

Lightpen/Graphics Card Diagnostics
Use Light Pen/Graphics test, LGTST.CM by typing

LGTST [ret]

with the CMI diagnostics disk in drive 0.

The standard command interpreter is used so the LIST and R commands, and P and N options are available as described in the general introduction. Tests are run by typing

<test name>[,<option1>,<option2>...,<optionN>] [ret]

Light Pen Timers

Note: Although the Series III does not use a lightpen as a graphics input device, the circuitry is still present and facilities such as the 6840 timer may still be used for general software purposes. Therefore the following diagnostics should be used. Either processor can run the TIM tests, as selected by the SEL command (see below).

Test name: TIM
No. tests: 4

Test No.1 *Light Pen Timer Read/Write Latches*
Purpose: 6840 timer preset latches can be written to and the counters read.

Timers are put into preset state in which the counters always reflect the contents of the preset latches. Then each timer is write/read tested with all numbers from 0 to $FFFF. Each write/read is a 16-bit transfer through the 8-bit bus.

Test No.2 *Light Pen Timer Internal Clock Timeout*
Purpose: Correct timeout from timers under internal clock.

The internal clock is provided by the BCA signal. Timer outputs are enabled and latches preset to $FFFF. Then all three counters are released and their timeouts compared to a software status-polling (to sense timeout) timing reference loop.

Test No.3 *Light Pen Timer 2 External Clock*
Test No.4 *Light Pen Timer 3 External Clock*
Purpose: Timers under external clock

Timer 2 counts frames, thus gets a 20mS clock cycle. The test is not synchronised to the frame pulses so a +/-10mS jitter is permissible. The timer is preset to count 200 clocks (4 secs), then released and compared to the software reference with the required tolerance.

Timer 3 is clocked by processor 2 phase 2 (1MHz). It is preset to $FFFF then released and its timeout compared to the software reference.

Both timers run in single shot mode with outputs enabled.

Light Pen PIA

Test name: PIA
No. tests: 1

Test No.1 *PIA Test*
Purpose: The PIA is used to control the scroll register as well as the light pen circuitry.

Each side of the PIA is configured as all outputs then all numbers from zero backwards down to zero are written to the data latches and verified.

Processor Access Selection

Test name: SEL
Purpose: Allows user to specify which processor runs the TIM tests.
Options: C=n CPU selection
Range 1-2, default 2

In addition to selecting which processor runs the TIM tests, the SEL command adjusts the timing parameters used to compare the hardware timer to software loops. This is necessary because refresh occurs on P1 cycles, resulting in P1 loops running slightly slower.

Video RAM Testing

Test name: VRAM
No. tests: 8
The VRAM tests do not use the SEL command to select processors. Which processor runs these tests is determined by the default graphics processor select byte contained in the Q133 boot ROM (normally P2 for the CMI).

Test No.1 *VRAM Processor*
Purpose: This determines which processor has access to the video RAM. To actually change processors, use the SEL command.

Test No.2 *Random VRAM Test*
Purpose: Random numbers are written to video RAM, then compared with original. Error if not the same.

Test No.3 *AA VRAM Test*
Purpose: Video RAM filled with hex number AA.

Test No.4 *55 VRAM Test*
Purpose: Video RAM filled with hex number 55.

Test No.5 *XY Byte/Inc*
Purpose: Test XY vector drawing hardware by writing random numbers and auto-incrementing. Numbers are then compared with actual video RAM contents.

Test No.6 *XY Byte/Dec*
Purpose: Same as XY/Inc except vector drawing hardware auto-decrements.

Test No.7 *XY Draw*
Purpose: Test the XY vector hardware.

This test draws a 2 byte wide by 16 byte high pattern into video RAM using the XY vector hardware, then checks that it was correctly drawn in the correct absolute locations in memory.

Test No.8 *Random XY, Byte Read*
Purpose: Video RAM is directly filled with random numbers. Random numbers are then regenerated and compared with video RAM via XY hardware.

Q256 memory card memory tests

When testing system memory it is desirable to have as little memory space as possible taken up by the operating system and the memory test itself. In order to restrict the memory diagnostics within a 16K block, they are split into three separate programs: MEM256, MAP256 and DMA256. The first performs bulk memory testing in 16K chunks and exercises the parity generation/checking system. MAP256 tests the full capability of the memory management system, except for the automatic DMA map selection, which is tested in DMA256.

A fourth program, MEMDBG is a small and simple program which exercises specific sections of the Q256 circuitry in very tight loops and thus generates the stable CRO traces needed to find circuit malfunctions. It is only appropriate to use MEMDBG once MEM256 or the other diagnostics have been used to obtain an approximate area of the fault. See Section MEMDBG

The first three tests, MEM256, MAP256 and DMA256, are run in the same way. To run MEM256, for example, type

MEM256 [ret]

with the CMI diagnostics disk in the left hand drive.

The standard command interpreter is used so the LIST and R commands, and P and N options are available as described in Chain file diagnostics. Tests are run by typing

<test name>[,<option1>,<option2>...,<optionN>] [ret]

To understand the information below, three definitions are required:

Logical or Processor Space: is the numerical range of addresses actually put out on the processor address bus. The logical address may also come from a DMA device such as the Floppy or Hard Disk controllers, MIDI card or Waveform Processor.

Physical Space: is the area of RAM which is actually accessed in response to the logical address.

Mapping: is the translation of any given logical address to any physical address. The Q256 does this on 2K "pages". Data which is contiguous within a single page of logical space will be contiguous within a page of physical memory. However all the pages that are contiguous within logical space can be arbitrarily shuffled in physical memory without the processor being aware of this when running programs etc. Also, an area of physical memory can appear in zero, one, two or more different places in the logical space.

Common Initialization, Options and Test Procedures

The three test programs share common initialization and option selection routines, and use common move and mapping routines to test the area of memory occupied by the operating system and test program.

Initialization: When the test program is first entered, the PIA setup of the graphics card is read and saved, for later restoration after tests which change this setup.
When the user types a command, one or all of a group of tests with the same name are executed sequentially. Before each group is run, the error counter is zeroed. Before each individual test is executed, a standard test mapping is set up in map 28, then all system states are switched to map 28. The standard test mapping is as follows:

Processor Space	Physical Memory
$0000 - $7FFF	Blocks 0.1 (Block 0 includes QDOS and test prog)
$8000 - $FFFF	Deselected

At the end of each test or during an abort caused by errors, the graphics card PIA setup is restored to the state which was saved on program entry. Then a check for any parity errors which occured during testing is performed, and a message printed if an error is found. The operating system and test program are moved back to the bottom of physical memory if necessary, and QDOS mapping reinitialized. Lastly the disk driver routines are reloaded to the top of processor RAM space from the floppy disk controller ROM and hard disk controller ROM if installed. The reloading of disk drivers is necessary because they are not part of the protected operating system and may have been overwritten by the memory tests.

Options: A number of options which apply to all tests can be set using the SEL command, rather than having to specify them each time a command is typed. To obtain a display of current option settings just type

SEL [ret]

To change an option, type

SEL,<option>=n [ret] where n is a number.

The options available are -

	Range	Default
C - Card select.	0-15	0
B - Block(s) select.	0-15	0-15.
E - Error limit.	0-127	0
D - Display	0-1	0
V - Video ouput	0-1	1
L - Line printer output	0-1	0

The Card select option allows multiple boards to be tested. This is useful in testing a board which is not working sufficiently to be able to load the operating system and test diagnostics. The default of zero tests the system card. Note that although the test can handle up to 16 cards, the standard CMI motherboard has only two Q256 slots.

Each card contains 256K bytes of memory consisting of four columns of 64K chips. The memory management hardware divides the 256K physical space into 128 pages of 2K each, but the bulk memory tests deal with sixteen blocks of 16K each (each 16K block therefore consists of 8 2K pages). The B option allows selection of which blocks are to be tested. The default option tests all blocks, from block F (15) downwards, which means the operating system block (Block 0) gets tested last.

The error limit determines how many errors can be logged before the test aborts.

The display option enables the printing of each card and block number as it is tested. Since several of the tests take a fairly long time to execute, particularly in a multi-card test rig, this option reassures the operator that the test is still running and has not crashed due to errors in the system memory.

The video option allows the above display to be printed on the system video console, while the L option prints it on a line printer if this is connected to the back panel of the CMI.

Error Messages

For most tests, if an error occurs a common error message routine is called which indicates where in memory the error occurred, the data which was expected, and the data which was actually read. It then reads the error location again. If on the second read the data is correct, the error is indicated as "SOFT". If it is still wrong and the same as it was the first time, it is indicated as "HARD". If the second read is wrong but different from the first read, the error is "RANDOM".

Testing Physical Block 0

The operating system (OS) normally resides in the bottom 16K of both logical and physical space, within processor addresses 0 to $3FFF. All testing takes place in the processor address range $4000 to $7FFF. When Block 0 is to be tested, the OS must be moved to a different block. The new block is first mapped into the address range $4000 to $7FFF then Block 0, residing in 0 to $3FFF is copied up to the new block. This moves QDOS, the test program, and Processor 2's stack into the new block. Processor 1's stack is in separate RAM on the Q133 and so is not affected by the Q256 diagnostics. At this point two copies of the OS exist, with P2 actually executing the one in Block 0. The new block is mapped into the address range 0-$3FFF so that now P2 is executing the new copy. Block 0 is then free to be mapped by the normal mapping routine into the $4000-$7FFF address range for testing. When the test on Block 0 is complete, the OS is again copied into $4000-$7FFF, then Block 0 is mapped back into 0-$3FFF.

MEM256
This section describes MEM256 V1.5.

MEM256 performs the bulk memory data and addressing tests, refresh and parity system tests.

Test name: 29
No. tests: 8
Purpose: Checks both processors' abilities to write rapidly varying data throughout memory.

In each test, the SELected blocks are mapped one at a time into $4000 to $7FFF and a semi-random data sequence written by either or both processors in an order as specified below. Each processor then checks its own data.

Test no.1: *P1 ODD UP, P2 EVEN UP*
P1 writes to all odd locations starting from $4001 and proceeds upwards, while P2 writes to even locations starting from $4000 and proceeds upwards.

Test no.2: *P1 ODD DOWN, P2 EVEN UP*
P1 writes to all odd locations starting from $7FFF and proceeds downwards, while P2 writes to even locations starting from $4000 and proceeds upwards.

Test no.3: *P1 ODD DOWN, P2 EVEN DOWN*
P1 writes to all odd locations starting from $7FFF and proceeds upwards, while P2 writes to even locations starting from $7FFE and proceeds downwards.

Test no.4: *P1 ODD UP, P2 EVEN DOWN*
P1 writes to all odd locations starting from $4001 and proceeds upwards, while P2 writes to even locations starting from $7FFE and proceeds downwards.

Test no.5: *P1 ODD, P2 EVEN*
P1 writes to all odd locations starting at $4001 then $7FFF and proceeds by writing alternately to bottom and top of memory working in towards the middle. P2 writes to even locations starting from $4000 and $7FFE and proceeds in a similar fashion towards the middle. The verify phase starts from the middle and works outwards.

Test no.6: *P1 EVEN, P2 ODD*
P1 writes to all even locations starting at $4000 then $7FFE and proceeds by writing alternately to bottom and top of memory, working in towards the middle. P2 writes to odd locations starting from $4001 and $7FFF and proceeds in a similar fashion towards the middle. The verify phase starts from the middle and works outwards.

Test no.7: *P1 ALL*
P1 writes to all locations starting at S4000 then S7FFF and proceeds by writing alternately to bottom and top of memory working in towards the middle. The verify phase starts from the middle and works outwards. P2 executes a dummy routine.

Test no.8: *P2 ALL*
P2 writes to all locations starting at $4000 then S7FFF and proceeds by writing alternately to bottom and top of memory working in towards the middle. The verify phase starts from the middle and works outwards. P1 executes a dummy routine.

Test name: WA
No. tests: 2
Purpose: Checks for addressing errors.

Test No.1: *WALKING ADDRESS P1*
Test No.2: *WALKING ADDRESS P2*
The test block is first filled with random data. Then each even/odd pair of bytes is written with the address of the even byte (e.g. $4000 is written to $4000,$4001). Each byte of data is verified immediately after it is written, and when the block has been filled, all data is verified again in read-only pass.
In test No. 1, P1 performs the test, while P2 executes a dummy, and vice-versa for test No. 2.

Test name: REF
No. tests: 4
Purpose: Checks that dynamic memory refresh is working.

Options:	T=n	Time option Range 1-100, default 5
	O=n	Operating system block Range 0-3, default 0

Test No.1 *REFRESH*
P2 fills all the test blocks except the OS block with $50 plus the block number, then executes a delay loop for T seconds. It then checks the data in each block. If refresh is not working, the default delay of 5 seconds is ample for the memory contents to largely decay away. P1 executes the same delay loop but does no testing.

Test No.2 *REFRESH WITH TAS, P2 ONLY*
P2 fills all the test blocks except the OS block with $50 plus the block number, then executes a delay loop for T seconds, exercising the CPU card indivisible instruction hardware while it waits. It then checks the data in each block. P1 executes the dummy wait loop and does no testing.

Test No.3 *REFRESH WITH TAS. P1 ONLY*
This is the same as test no.2, with the processors swapped.

Test No.4 *REFRESH WITH TAS. BOTH CPUS*
P2 fills all the test blocks except the OS block with S50 plus the block number. P1 and P2 then both execute a delay loop for T seconds, exercising the CPU card indivisible instruction hardware while they wait. P2 then checks the data in each block.

Test Name: PARITY
No. tests: 2
Purpose: Verifies operation of the parity checking system.

Options: D=n Data used in test
Range 0-255, defaults 170, 168 respectively.

Test No.1: *PARITY SYSTEM - EVEN*
Test No.2: *PARITY SYSTEM - ODD*
These tests use the special parity error generation hardware on the Q256 card to predictably generate a parity error. Test no. 1 uses data with even parity (170=$AA), and test no. 2 uses data with odd parity (168=$A8).
Only P2 runs the test. Initially it is in the A, or system state and map 28 is selected. Map 29 is configured with the same standard test mapping (Common Initialization, Options and test Procedures) then P2 B state is allocated to map 29 with the Parity Error Generate (PERGEN) bit set. Then on each test block, a quick fill (every 128th byte) of the test data as indicated by the D option is performed, followed by a special quick verify with parity error generation. For each verify, the parity register of the current test card is first read to check for any prior errors which should not be there yet. Then the CPU fuse register is set up to switch to the B state, just as the test data byte is read. Since the byte was written in the A state with PERGEN not set (as is normal), and is read back in the B state with PERGEN set, a parity "error" should be generated at the moment of reading. We then switch back to the A state and interrogate the current test card's parity status register to check the presence of the parity error flag, the correct physical 64K rank number, and correct upper 5 bits of the processor address at which the "error" occured. The register is then read again to check that the first status read automatically cleared the error.

MAP256
This section describes MAP256 V1.6

Test name: MAP
No. tests: 15
Purpose: Tests the mapping functions of the Q256.

Test no. 1 *PAGE UNIQUENESS - P2*
Test no. 2 *PAGE UNIQUENESS - P1*

Options: O=n1,n2...n4 OSBLK: Operating system block
Range 0-3, default 0-3
(4 options)

These tests check that each 2K physical page (there are 128 on each Q256) can be accessed uniquely. The general approach is to write each page with its own page number, and then verify them all to ensure no page overwrote another. Getting around the operating system complicates things slightly.
The test is actually run four times. According to the default OSBLK option it is first run with the OS in Block 0 (no moving required). Block 0 occupies pages 0 to 7 so pages 8 to 127 are mapped into $4000 to $7FFF in groups of 8 and "quick-filled" (every 128th byte) with the page number. Pages 8 to 127 are then quick checked to still contain their page numbers. Then the OS is moved to the second OSBLK option, Block 1, which occupies pages 8 to 15, so pages 0-7 and 16-127 may be tested with filling and verifying page numbers. Similarly the OS is moved to Block 2 while testing pages 0-15 and 24-127 and finally the OS is moved to Block 3 while testing pages 0-23 and 32-127. Which blocks are used for the OS and in what order may be changed using the O option.
Processor 2 executes test no.1, and P1 runs test no. 2; the procedure is the same for both.

Test no. 3 *P2 MAPPING, P1 TAS*
Test no. 4 *P1 MAPPING, P2 TAS*
This test ensures that the indivisible instruction hardware on the CPU card can be rapidly accessed simultaneously with mapping operations on the Q256. The MAPPING processor (P2 in test 3, P1 in test 4) follows the same procedure as in the page uniqueness tests 1 and 2. Again the test is run four times with the OS in one block while all other physical pages are filled and verified with page numbers. Meanwhile, the other processor loops around quickly doing two Test-And-Set instructions followed by two Clear instructions on a flag byte in memory. The flag byte is not used for any other purpose. The TAS processor continues testing and doing IO services until another flag is set by the MAPPING processor to indicate it has finished.

Test No. 5 *PROCESSOR UNIQUENESS*

The capability of the two processors to be mapped independently is tested by selecting different blocks in 2 different maps and running tests with different data on both processors simultaneously. The test is run once for each card under test. P1 is left as initialized in map 28, and Block 1 of the test card is mapped into $4000-$7FFF. P2 is switched to map 29 and Block 2 of the test card is mapped into P2's $4000-$7FFF. P1 then fills its test area with $AA while P2 fills its area with $55. The two processors then verify their own areas to contain the correct data.

Test No. 6 *FAST MAPPING - P2*

Test No. 7 *FAST MAPPING - P1*

These tests ensure that rapid writing to the mapram does not interfere with normal memory accesses. One processor loops over all test blocks (as selected by the SEL command) writing to all locations starting at $4000 then $7FFF and proceeding by writing alternately semi-random data to bottom and top of memory working in towards the middle. The verify phase starts from the middle and works outwards. (This is the same as the ALL tests in MEM256, N=7,8).

Meanwhile the other processor switches to map 29 and rapidly maps blocks in and out of $4000-$7FFF. All blocks except the current OS block and the block being tested by the first processor are mapped into $4000-$7FFF and quick-filled (every 128th byte) with zeros. After the last block, a flag is set to tell the first processor to stop testing.

In test no. 6, P2 does the fast mapping while P1 tests memory, and vice-versa for test no. 7.

Test No. 8 *SELECT ALL MAPS - P1*

Test No. 9 *SELECT ALL MAPS - P2*

32 different mappings may be set up in the Q256 mapram. Each map from 0 to 31 is initialised with Block 0 containing the OS mapped into 0-$3FFF and a different physical page mapped into logical page 8. (Physical page 8 is used in map 0, page 9 in map 1, ... page 39 in map 31). After each initialization the processor switches to that map, and fills logical page 8 with the map number. We then start again at map 0 and verify that its logical page 0 contains zero, and repeat through to map 31.

P1 runs the test in no. 8, P2 runs no. 9.

Test No.10 *A/B SELECT - P2*

Test No.11 *A/B SELECT - P1*

The Q256 allows for different mappings to be selected depending on whether the processor is in the A (System) or B (User) state. Map 29 is initialised with the standard test mapping and the B state is allocated map 29. Then the following procedure is repeated on each test card: starting in A state, Block 1 of the test card is mapped for all states into $4000-$7FFF (call this area K1) and filled with $AA. Block 2 is then mapped for all states into K1 and filled with $BB. Next Block 1 is mapped to K1 for A states only, and Block 2 is mapped to K1 for B states only. Still in the A state, K1 is verified to contain all $AAs.

Then the processor switches to the B state and verifies K1 to contain all SBBs. Thus far it is proven possible to read from different areas of physical memory in the two processor states.
The second stage of the test switches back to the A state and fills K1 with $11. Then the processor switches to the B state again and fills K1 with $22. Still in the B state, Block 1 is mapped into K1 for all states and checked to still contain $11 then Block 2 is mapped into K1 for all states and checked to contain $22. This proves that data written while in the A or B state can be directed to different areas of memory. The processor switches back to the A state for the next card to be tested or to exit the test.
P2 runs test no. 10; P1 runs test no. 11.

Test No.12 *PERIPHERAL DISABLE - P1*
Test No.13 *PERIPHERAL DISABLE - P2*

These tests check the peripheral enable output from card 0 which allows the peripherals (i.e. everything on the bus except the RAM card itself) which usually reside above $E000 in the processors' logical address space, to be disabled and replaced by extra RAM. The Q256 mapping system is itself a peripheral - the mapram lives from $F000-$F7FF (one logical page) so this test works by trying to change the mapping when peripherals are disabled.
Map 29 is initialised with the standard mapping and the B state is allocated map 29 with peripherals disabled. Then for each test card the following procedure is followed:
Still in the A state (and map 28), physical pages 8 and 9 are mapped to logical pages 8 and 9 respectively. Page 8 is filled with $AA and Page 9 is filled with $55. Then for map 29, physical page 9 is mapped to both logical page 9 and logical page 30 i.e. to the same logical area as the mapram. From the B state, reading the mapram area would now get all $55s.
We switch to the B state and attempt to map logical page 9 to physical page 8 in map 29 (the B map). With peripherals disabled, this should not work, and logical page 9 is verified to still contain $55.

If all is well so far, the Peripheral Enable probably works, so we attempt to completely clobber the mapram by filling it with $AA from the B state. Since peripherals are disabled in the B state, this should merely write to physical page 9 without changing any mapping. Logical page 30 is verified to contain $AA, then we switch back to the A state and verify logical pages 8 and 9 as both containing $AA. (Since peripherals are disabled, the last state switch can not be achieved by writing to the processor card control latch - a dummy software interrupt is called instead. The A state is automatically selected on interrupts.)
P1 runs test no. 12; P2 runs test no. 13.

Note - Only Card 0 is effectively tested by this test because it is the only card whose Peripheral Enable output pin is connected on the motherboard to its own and all other RAM card's PENB inputs.

Note - also that if an error occurs while testing in the B state, the A state must be reselected in order to print the error message. The standard error message routine re-reads the incorrect location to ascertain what kind of error has ocurred, and since the mapping has changed it will no longer be reading the correct physical location. Thus errors which are in fact "Hard" will be indicated as "Soft", or possibly "Random".

Test No. 14 *GRAPHICS RAM DISABLE - P1*
Test No. 15 *GRAPHICS RAM DISABLE - P2*
Option: R=n
Range 0-1, default 0

Checks VRAMEN output from the memory management system which allows the 16K of dedicated RAM on the Q219 graphics card to be mapped in or out of either processor's address space.
The video graphics RAM (VRAM), if enabled, occupies the address range $8000 to $BFFF. The contents of VRAM are copied down to $4000 to $7FFF to be saved while testing takes place. Then VRAM is checked to be working by filling with $AA and verifying (the $AA pattern appears on the console screen). Map 29 is initialised with the standard mapping and the processor switches to map 29 with graphics access disabled. It attempts to clear VRAM then reselects map 28 with graphics enabled to verify $AA is still there. Finally the saved contents of the screen are copied back.
P1 runs test no. 14; P2 runs test no. 15. Normally only one processor has access to the VRAM and this is selected through the PIA control on the Q219 at system start-up and by some application programs. Both tests 14 and 15 can run because the PIA set-up is saved on entry to MAP256 and restored at the end of each test. The tests also run on the old graphics cards Q218 and Q045 but in the latter case, which processor has graphics card access is hardware selected, so one of the tests will exit without testing.
Note - Only Card 0 is effectively tested by this test because it is the only card whose VRAM Enable output pin is connected on the motherboard to the graphics card.

DMA256
This section describes DMA256 V1.6

Test Name: DMA
No. tests: 2

Test no.1 *P2 DMA MAP SWITCHING*
Test no.2 *P2 DMA/P1 UNIQUENESS*

DMA256 tests the automatic selection of special DMA mappings via the DMA claim lines which are input to the Q256 memory cards from all devices on the bus which use DMA (Floppy Disk Controller, Hard Disk Controller, General Interface card).

P2 initializes map 29 with Block 2 of the test card mapped into K1 (logical area $4000 to $7FFF) as a 16K DMA buffer, and switches into map 29 to fill the buffer with its own page numbers. Then P2 switches back to the standard map 28 which has Block 1 mapped into K1 and fills it with a semi-random sequence. A disk operation is then initiated which writes the DMA buffer out to disk and into a file called DMAFILE.TF. Map 29 is selected again and the DMA buffer cleared. P2 switches back to map 28 and starts a second DMA operation, this time reading the buffer in from disk. K1 in map 28 is checked first to be still containing the random sequence, then map 29 is selected to verify the page numbers read in from disk. The procedure is repeated for all test cards.

In test no. 1, only P2 runs the test. In test no. 2, P2 runs the same test while P1 continually random fills and checks its own separate block (Block 4 of the test card) in the same logical test area using map 27.

The test file DMAFILE.TF is not deleted at the end of the test so that in the event of errors, the file may be examined using the QDOS command DUMP (this is not the same as the Channel Card diagnostic DUMP command!). It is not required to exist before the DMA test runs, so the user may delete the file at any time.

MEMDBG

The diagnostic programs MEM256, MAP256 and DMA256 provide the comprehensive tests which thoroughly check the Q256 256K Ram card operation. However the size and complexity of these diagnostics mean that they are not very easy to use for chip-level debugging of known faulty boards. MEMDBG is a small and simple program which exercises specific sections of the Q256 circuitry in very tight loops and thus generates the stable CRO traces needed to find circuit malfunctions. The absence of error checking means that the processor does not end up spending most of its time printing error messages. It is only appropriate to use MEMDBG once MEM256 or the other diagnostics have been used to obtain an approximate area of the fault.

Use of MEMDBG

To start MEMDBG, type

MEMDBG [ret]

with a diagnostics disk containing MEMDBG.CM in the left hand drive. When the program is first entered, it calculates its own checksum in case memory faults have prevented a successful load. If the checksum is wrong, the program exits back to QDOS. This check can be overidden by typing

MEMDBG;-C [ret]

to start the program.

The user is presented with a menu of tests indicating which parts of circuitry are exercised by each test. Simply type the number of the required test then answer the prompts for data such as which processor is to run the test, what data is to be used, what 16K memory block is to be written/read etc.

Once all data required for the test is obtained the test loop is entered immediately. The only thing left to do then is watch the signals of interest behave as the test loop requires. In general it is best to connect the CRO before starting the program so that accidental shorts don't crash the system before you get to see anything. To keep the loops as tight as possible, there is no provision for terminating the test - just hit reset or console interrupt (NMI). The latter only works if P2 is running the test, and the front panel P2 NMI switch is enabled only. After a console interrupt the program may be re-entered from the monitor without reloading by typing

2000;G

A board which is unable to load the system and/or MEMDBG can only be debugged by optioning it as Card 1 and installing it along with a healthy board as Card 0. If the faulty board still crashes the system, disable the data driver using option W3 (see Q256 hardware documentation).

Adding Your Own Test Loops

The eight tests contained in the program should be sufficient to solve most problems on the Q256. However MEMDBG has been written such that debug technicians can easily add their own test loops tailor made to investigate particular curly problems if the need arises. To do this a disk containing the following files should be placed in the right hand drive:

MEMDBG.SA	Source file
MEMDBG.CF	Chain file for editing and assembling
QEQU.SA	QASAR system equates
MAC256.SA	Q256 mapping macros

A system disk containing the following files should be booted in the left hand drive:

RASM09.CM	6809 assembler
EDIT.CM	Cyword editor
IO.CM or QIOPACK.CM	QASAR IOpack required for editor.

A 6809 assembler manual will also be required.

The procedure for adding a new loop is contained in the chain file. Just type

CHAIN MEMDBG:1 [ret]

to start the process. The chain calls the editor followed by the assembler to generate a new MEMDBG.CM on drive 1. Options in the chain include:

C - generate an assembly listing in the console screen
P - print assembly listing on the line printer
L - save assembly listing as the file MEMDBG.AL:1

For example, type

CHAIN MEMDBG:1;C [ret] to list to the console.

The program consists of three sections: A test executive, the menu, and the collection of test loops. Only the menu needs to be modified to include the new loop, which should be typed into the source file after the other loops. Refer to the fully commented source file for further information on its structure.
After adding another test loop the version number and "last modified" message should be updated to distinguish the new program from old versions.
Once the program is modified, the checksum will no longer be correct. Use the -C option to run the program and get the checksum error message. The formula for calculating the new checksum is:-

new CHKSM = complement(old CHKSM - error CHKSM reported) + 1

Run the chain again and modify the CHKSM byte to the new value.

Waveform Processor Card Memory tests
Run file: WPMEMTST [ret]
So called; Waveform Processor MEMory TeST.
This document is current for WPMEMTST Rev 3.

The SEL command is the selection for global options; D, E and M

When D=0, which is its default value, no messages will be displayed but error messages may be displayed. D=1 means that messages to help you monitor the progress of the test will be displayed. For example messages will appear as RAM is being filled and checked.

When E=0 no error messages will be displayed but the diagnostics will continue testing. In other words errors are ignored. When E=1, which is the default value, the diagnostics will abort after finding and printing the first error. E=2 will print all errors and continue to test.

The option M is for displaying the 68k monitor functions as a 68k file is loaded into the Waveform Processor card. It need only be used (M=1) when you think a 68k file is not loading into the Waveform Processor card. M=0 is the default value.

To remind you of the SEL options the SELHLP command contains a summary.

Similarly the HELP command will summarize the rest of the commands in WPMEMTST.

Test Name: PRAM
No. of tests: 6
Purpose: To test the Waveform Processor private RAM.

PRAM tests the Waveform Processor card's private RAM between addresses $81000 and $C0000. Option N=1 zeros will be written and read to all locations in the range, N=2 $5555 and N=3 $AAAA similarly. For N=4 a random word will be written and read, while N=5 the address itself is written in and read out - so called *Walking Address Test.* For PRAM,N=6 bytes instead of words are written and read to make sure high- and low-order bytes may be accessed independently.

Test Name: P1RAM
No. of tests: 6
Purpose: To test the Waveform Processor's access to P1 memory.

P1RAM performs the first five tests as in PRAM but over P1 address space from $5000 to $5800. The waveform processor will see this address space from $55000 to $55800 and errors will have their addresses displayed as such.

Test Name: P2RAM
No. of tests: 6
Purpose: To test the Waveform Processor's access P2 memory.

P2RAM performs the first five tests as in PRAM but over P2 address space from $5800 to $6000. The waveform processor will see this

address space from $45800 to $46000 and errors will have their addresses displayed as such.

Test Name: RAMREF
No. of tests: 1
Purpose: To test the refresh function on the Waveform Processor's memory.

Test Name: DMATST
No. of tests: 2
Purpose: To test the DMA mapping function on the Waveform Processor's access to P1 and P2 memory.

DMATST,N=1 tests the P1 DMA mapping.
DMATST,N=2 tests the P2 DMA mapping.

Waveform Processor Card Interrupt tests
Run file: **WPINTTST** [ret]
So called; Waveform Processor INTerrupt TeST.
This document is current for WPINTTST Rev 1.

Test Name: INTRPT
No. of tests: 7
Purpose: To test the CMI to Waveform Processor interrupts (7 of them).

CMI-39 Waveform RAM Card tests
Run file: **WVRAMTST** [ret]
So called; WaVeform RAM TeST.
This document is current for WVRAMTST Rev 5.

The SEL command is the selection for global options; D, E, B, R and M

When D=0, which is its default value, no messages will be displayed but error messages may be displayed. D=1 means that messages to help you monitor the progress of the test will be displayed. For example messages will appear as RAM is being filled and checked.

When E=0 no error messages will be displayed but the diagnostics will continue testing. In other words, errors are ignored. When E=1, which is the default value, the diagnostics will abort after finding and printing the first error. E=2 will print all errors and continue to test.

The option M is for displaying the 68k monitor functions as a 68k file is loaded into the Waveform Processor card. It need only be used (M=1) when you think a 68k file is not loading into the WP card. M=0 is the default value.

The R option in SEL is to select the combination of RAM cards you wish to test. R=1 will select the first waveform RAM card. This option defaults to all 7 RAM cards. If a card is not present errors will be detected for all data except 0000, so be sure to select the required number first. A command CARDS will give you an indication of what cards are present.

The B option in SEL enables to select a particular 512K block within each RAM card. B defaults to all four values from B=0 to 3.

To remind you of the SEL options the SELHLP command contains a summary.

Similarly the HELP command will summarize the rest of the commands in WVRAMTST.

Test Name: CARDS
No. of tests: 1
Purpose: To see which CMI-39 Waveform RAM Cards are present.

While one can simply look at how many of the seven waveform RAM cards are present there is the possibility that they maybe selected incorrectly or may not be working at all. CARDS is a simple test to give a quick indication of any problem

Test Name: WVRAM
No. of tests: 6
Purpose: To test the RAM on the CMI-39 Waveform RAM Cards.

These tests will check the RAM you have selected in SEL with the R option.
WVRAM,N=1 fills the RAM with $0000 and checks it.
WVRAM,N=2 fills the RAM with $5555 and checks it.
WVRAM,N=3 fills the RAM with $AAAA and checks it.
WVRAM,N=4 fills the RAM with random data and checks it.
WVRAM,N=5 fills the RAM with its own address and checks it (a so called *walking address test*).
WVRAM,N=6 fills the RAM and checks it in *byte mode.*
WVRAM,N=7 fills the Ram with $FFFF0000 then fills it with $0000FFFF before checking it. This test is designed to slam the data bus up and down to check for problems with bus termination.

Note: The waveform RAM should be tested with and without the channel cards running, as the channels running exercise the waveform RAM more heavily. Refer below to the CRUN and CSTOP commands.

Due to the bulk of RAM to be tested (up to 14Mb) several hours of continuous testing is necessary to be completely sure of correct operation. You should use the continuous-running option P=C on any of the tests required for long periods. Refer also to the CHAIN WAVERAM section 6.11.2. Over an extended testing period, ERRORS gives an accumulated error count for each card but be sure to clear the error counts first with the CLEAR command. See below.

Error Messages

If an error occurs a common error message routine is called which indicates where in memory the error occurred, the data which was expected, and the data which was actually read. It then reads the error location again. If on the second read the data is correct, the error is indicated as *SOFT*. If it is still wrong and the same as it was the first time, it is indicated as *HARD*. If the second read is wrong but different from the first read, the error is *RANDOM*.

Test Name: RAMREF
No. of tests: 1
Purpose: To test the refresh function on the Waveform RAM cards.

Be sure that the channel cards are not running during this test (type CSTOP if they are). This is because if channel cards are running the RAM will be refreshed so the test will be invalid.

Channel card running and stopping

To run all channels type the CRUN command. To select individual channels, use CRUN with the C option. C will select any of the 16 channels. There are eight channel cards numbered 1 to 8 and each card has two channels a and b. Hence the first channel is 1a and the last or 16th channel is 8b. The channels selected will continually loop over the area of memory being tested at the time.

To halt the channel cards use the CSTOP command and similarly use the C option (C=1-16)to stop selected channels.

Error Logging

This test has seven error counters for each of the Waveform RAM cards. These counters are not automatically cleared when starting any tests, so use the CLEAR command to clear them. To display the number of errors that may have occurred on each card, type ERRORS.

To run memory tests continuously over a long period and accumulate the error counts you will also have to use the SEL,E=0 option so that errors are not printed (as when E=2) or that the test does not abort after the first error (E=1).

CMI-28 General Interface Card Memory tests
Run file: SMMEMTST [ret]
So called; SMIDI MEMory TeST.
This document is current for SMMEMTST Rev 1.0.

The SEL command is the selection for global options; D, E and M When D=0, which is its default value, no messages will be displayed but error messages may be displayed. D=1 means that messages to help you monitor the progress of the test will be displayed. For example messages will appear as RAM is being filled and checked.

When E=0 no error messages will be displayed but the diagnostics will continue testing. In other words, errors are ignored. When E=1, which is the default value, the diagnostics will abort after finding and printing the first error. E=2 will print all errors and continue to test.

The option M is for displaying the 68k monitor functions as a 68k file is loaded into the SMIDI card. It need only be used (M=1) when you think a 68k file is not loading into the SMIDI card. M=0 is the default value.

To remind you of the SEL options the SELHLP command contains a summary.

Similarly the HELP command will summarize the rest of the commands in SMMEMTST.

Test Name: PRAM
No. of tests: 6
Purpose: To test the General Interface Card's private RAM.

The PRAM command tests the SMIDI card's private RAM from locations $81000 to $82000. Option N=1 zeros will be written and read to all locations in the range, N=2 $5555 and N=3 $AAAA similarly. For N=4 a random word will be written and read, while N=5 the address itself is written in and read out - so called *Walking Address Test*. For PRAM,N=6 bytes instead of words are written and read to make sure LDS and UDS signals are operating properly

Test Name: P1RAM
No. of tests: 6
Purpose: To test the SMIDI Processor's access P1 memory.

P1RAM performs the first five tests as in PRAM but over P1 address space from $6000 to $6800. The waveform processor will see this address space from $56000 to $56800 and errors will have their addresses displayed as such.

Test Name: P2RAM
No. of tests: 6
Purpose: To test the SMIDI Processor's access P2 memory.

P2RAM performs the first five tests as in PRAM but over P2 address space from $6800 to $7000. The waveform processor will see this address space from $46800 to $47000 and errors will have their addresses displayed as such.

Test Name: DMATST
No. of tests: 2
Purpose: To test the DMA mapping function on the SMIDI Processor's access to P1 and P2 memory.

DMATST,N=1 tests the P1 DMA mapping.
DMATST,N=2 tests the P2 DMA mapping.

DMATST tests the DMA switching.

General Interface Card Peripheral tests

Run file: SMPERTST [ret]
So called; SMIDI PERipheral TeST.
This document is current for SMPERTST Rev 4.

Extra diagnostic tools needed: 11-LED monitor (plugs into 26-way socket), 7-LED monitor (has DIN plugs), 3 DIN to DIN leads, A DIN to 9-way connector and the XLR lead (3 plugs).

The SEL command is the selection for global options; D, E and M
When D=0, which is its default value, no messages will be displayed but error messages may be displayed. D=1 means that messages to help you monitor the progress of the test will be displayed. For example messages will appear as RAM is being filled and checked.

When E=0 no error messages will be displayed but the diagnostics will continue testing. In other words errors are ignored. When E=1, which is the default value, the diagnostics will abort after finding and printing the first error. E=2 will print all errors and continue to test.

The option M is for displaying the 68k monitor functions as a 68k file is loaded into the SMIDI card. It need only be used (M=1) when you think a 68k file is not loading into the SMIDI card. M=0 is the default value.

To remind you of the SEL options the SELHLP command contains a summary.

Similarly the HELP command will summarize the rest of the commands in SMPERTST.

The following two commands can be run on the SMIDI card without the need to use shorting plugs, LED monitors or other test equipmemnt.

Test Name: INTRPT
No. of tests: 2
Purpose: To test two CMI-SMIDI interrupts.

These interrupts are the 68k interrupts, levels 1 and 7. By refering to sheet 3 of the CMI-28 circuit, you'll see the two interrupts on the LS259 latch labelled $\overline{INT1}$ and $\overline{INT7}$.
INTRPT,N=1 will test level 1, and INTRPT,N=2 will test level 7.

Test Name: TIMERS
No. of tests: 4
Purpose: To test the two 68B40 timers on the board.
(Refer to sheet 6 of the circuit diagram of CMI-28.)

TIMERS,N=1,2 will test the latches of timer A (when N=1) and timer B (N=2) by writing a byte to the latch and reading it back.

TIMERS,N=3,4 tests the timers' interrupt to the 68k - level 2. Both timers are wire-ORed to this interrupt, hence the two tests. N=3 testing the ability of Timer A to interrupt the 68k and N=4 timer B's ability.

To run the next set of commands, a shorting plug or LED monitor is needed. If a shorting plug is placed in the 26-way socket tests ACIA,N=5 (ACIA looping) and SMPTE can be run only. An LED monitor (one with 11 LED's) can be plugged into the same socket and all of the following tests can be run.

Similarly all the tests can be run by connecting the SMIDI board up to the CMI-332 and -333 boards in the audio rack (just as it is in the Series III configuration) and by plugging in shorts from MIDI out A to MIDI in A; from MIDI out B to MIDI in B; from MIDI out C to MIDI in C; a special shorting lead from MIDI out D to the 9-way connector which has the keyboards' communications. Also a special shorting lead from SMPTE out to SMPTE in and CLICK in. Another LED monitor (one with 7 LED's) is plugged into the Sync and Drum-machine DIN sockets. With this arrangement the CMI-332 and most of the CMI-333 board can be tested quickly.

Test Name: ACIA
No. of tests: 5
Purpose: To test the 4 ACIA's and their interrupts to the 68k (level 3).

ACIA,N=1-4 will cause each ACIA in turn to transmit a byte over and over for a second or two. This can be monitored with the LED's or with a CRO placed on pin 6 of the ACIA's. The ACIA generates an interrupt on level 3 to the 68k each byte transmission. If an interrupt does not occur within a certain time limit the 68k assumes the interrupt is not working and will tell you so.

ACIA,N=5 is the ACIA looping test and will only work if the transmit signal from each ACIA is looped to its receive input. The 11-LED monitor has these loops implemented. The test cycles through all 4 ACIA's transmitting a byte then receiving the same. If you get receiving first, check the short or loop before suspecting the ACIA. If the test works with the 11-LED monitor plugged into the 26-way socket but does not work with the DIN plug loops then there is a fault on the CMI-332 board - either the open collector buffers are not transmitting or the opto-couplers are faulty. A CROw would be the best way to check this, but they tend to squawk and make an awful noise not to mention the mess they can leave on the bench or carpet.

INTIME and EXTIME are two timer tests, best monitored using the LED's (either the 11- or 7-LED monitors)

INTIME simply puts the timers into an internally clocked mode - the Sync out 1 and 2 flash fairly fast, Sync out 3 and 4 flash 8 times slower and the SMPTE LED (on 11-LED monitor) flashes somewhere between these rates.

The EXTIME command tests the cascading connections between the timers (Refer to Sheet 6 of the circuit). In order for this to work with the CMI-332 and -333 boards connected make sure you have the SMPTE shorting lead in and that it is plugged into the Click In socket, too.

For EXTIME,N=1 the Syncsw signal is low which means that Sync Out 4 = Click In, hence with the 11-LED monitor you will see the SMPTE LED and Sync Out 4 LED flash in time (very fast). Sync Out 1,2 and 3 will appear to count in binary.

For EXTIME,N=2 the Syncsw signal is high which means that Sync Out 4 will be a division of Click In. In this case the SMPTE LED will still flash at a very fast rate, Sync Out 4 at a slower rate, and Sync Out 1,2 and 3 will count at a very slow rate.

The DRUM command simply flashes the two Drum LED's on either the 11- or 7-LED monitors.

The SMPTE command displays the actual SMPTE code on the screen. This appears in the form of hours, minutes, seconds and frames each digit being separated by a 0, e.g. 15 seconds becomes 0105 seconds. Both the SMPTE generated and SMPTE read are displayed in the following form Hours Generated, Hours Read, Minutes Generated,... etc. Without the shorting lead or 11-LED monitor, in only SMPTE generated code will appear. SMPTE read will be all zeros.

If you have simply typed SMPTE with no option then about one minute of code will be displayed. By including the T-option (T=1 to 10) you can display up to ten minutes of code. By choosing T=0 it will display SMPTE code forever (until you press ESC).

This test exercises interrupt levels 4, 5 and 6. Level 4 for SMPTE generate and level 5 and 6 for SMPTE reading of zero and one, respectively. If the SMPTE generated code is grossly out of time with the minute timer then suspect the 3.84MHz crystal oscillator and associated circuitry. Small differences from the minute timer or between the SMPTE generated code and code read can be ignored.

Channel Card Memory Tests

Run file **CHMEMTST** [ret]
So called; CHannel Card MEMory TeST.
This document is current for CHMEMTST Rev 2.

Firstly select the number of channels you wish to test by choosing the C option in the SEL command. For instance to select the first channel card type SEL,C=1.

Test Name: CHRAM
No. of tests: 7
Purpose: To test the Channel Cards' memory externally (ie from P2)

The command CHRAM will run memory diagnostics on the channel card(s) from P2 (i.e. without the channel card processor being used).
CHRAM,N=1 fills RAM with $00 and checks it.
CHRAM,N=2 fills RAM with $55 and checks it.
CHRAM,N=3 fills RAM with $AA and checks it.
CHRAM,N=4 fills RAM with a random number and checks it.
CHRAM,N=5 fills RAM with $A5 waits then checks it (refresh test).
CHRAM,N=6 fills RAM with a word consisting of the page number (MSB) and the address within the page (LSB) then checks it (similar to walking address test).
CHRAM,N=7 tests for uniqueness in addressing any of the channel cards.

Test Name: LDRAM
No. of tests: 4
Purpose: To test the Channel Cards' memory internally.

The command LDRAM will load a program into the channel card(s) and the channel card will now run its own memory tests.
LDRAM,N=1 fills the available RAM with $00 and checks it.
LDRAM,N=1 fills the available RAM with $55 and checks it.
LDRAM,N=1 fills the available RAM with $AA and checks it.
LDRAM,N=1 fills the available RAM with a random number and checks it.

Channel Card Interrupt Tests

Run file **CHINTTST** [ret]
So called; CHannel Card INTerrupt TeST.
This document is current for CHINTTST Rev 2.

Test Name: CHINT
No. of tests: 4
Purpose: To test the end of loop, CMI to Channel Card interrupts and FIRQ's (from the timer on the Channel Support Card).
Firstly select the number of channels you wish to test by choosing the C option in the SEL command. For instance to select the first channel on the first card (Channel 1a) type SEL,C=1.

CHINT,N=1 will test the interrupts from the main CPU's to the channel cards.
CHINT,N=2 will test the fast interrupts (FIRQ's) on the channel cards driven by the timer on the channel support card.
CHINT,N=3 tests the end-of-loop interrupts on the channel cards. It sets a loop going and if the interrupt has not occurred when the loop is expected to finish then an error message is displayed. Beware if you have a problem with pitch generation the error may also occur. If this is suspected try running the pitch register tests with the file CHPITTST.

Run file **CHSUPTST** [ret]
So called; CHannel SUPport Card TeST.
This document is current for CHSUPTST Rev 1.

Test Name: TIMER
No. of tests: 1
Purpose: To test the 6840 timer on the Channel Support Card by writing to and reading each of the three timer latches.

Test Name: PIA
No. of tests: 1
Purpose: To test the 6821 PIA on the Channel Support Card by writing to and reading each of the two PIA latches.

Test Name: UNIQ
No. of tests: 2
Purpose: To test the uniquenss of addressing each of the channel cards from P1-MASK1 and from P2-MASK2. It necessary to have all channel present to run this test fully.

Channel Card Pitch Register Tests

Run file **CHPITTST** [ret]
So called; CHannel Card PITch Register TeST.
This document is current for CHPITTST Rev 1.
The test name SET was only used to help set up the software for this test and not necessary for the testing procedure. A four digit hex number is entered and the loop played, it then outputs eight values of the software timing of the loop. Use a frequency greater than $7FFF to exit this procedure.

Test Name: PIT
No. of tests: 12
Purpose: To test the 12 bits of the pitch register.

The channel runs a single loop for a particular pitch register value (predetermined pitch) and is software timed. If the end-of-loop interrupt occurs outside of a tolerance on the software timing an error message will appear.

Test Name: OCT
No. of tests: 8
Purpose: To test the 8 settings of the octave register.

The channel runs a single loop for a particular octave register value (predetermined pitch) and is software timed. If the end-of-loop interrupt occurs outside of a tolerance on the software timing an error message will appear.

Channel Card Filter Tests

Run file CHFILTST [ret]
So called; CHannel Card FILter TcST.
This document is current for CHFILTST Rev 1.

Test Name: 2TONE
No. of tests: 10
Purpose: To test the filter settings and all the bits in the filter DAC register.

The first test 2TONE,N=1 will simply generate a sinewave at 1kHz with a filter setting of $FF. Use an oscilloscope to monitor the signal, take note of the amplitude. Press the space bar for the next test and each successive test.
The next 9 tests (2TONE,N=2-10) will create a two tone signal, the lower frequency being at the -6dB point for a particular filter setting and the higher frequency being at a 0dB level for the same filter setting. Note that the signal level oscillates between the 0dB signal level (as measured with the 1kHz signal) and a level half of that amplitude. For tests 2TONE,N=3,4,5 and 6 the least significant 4 bits of the filter DAC are tested by pulsing each one on and off slowly during the display of the two tones. By looking very carefully at the display during the attenuated signal a tiny flicker can be perceived if the DAC is working properly.

Test Name: FILT
No. of tests: 10
Purpose: To test the filter settings for -6dB points.

The first test FILT,N=1 will simply generate a sinewave at 1kHz with a filter setting of $FF. Use a distortion meter to monitor the signal level, set it to 0dB. Press the space bar for the next test and each successive test.

Each test will generate a signal at the given frequency and filter DAC value. Adjust the appropriate pots on the Audio Module for a -6dB setting. But note that if the Audio Module has been adjusted according to the setup procedure in AUDIOCAL, this FILT test should really be used to exhibit the linearity of the filter. I.e. for each successive test in FILT the attenuated signals should be at approximately the same level for each.

CMI-333 METRONOME & AUDIO MIXER DIAGNOSTICS

Run file METMXTST [ret]
So called; METronome and MiXer TeST.
This document is current for METMXTST Rev 2.

Test Name: MET
No. of tests: 2
Purpose: Tests the metronome circuitry.

Connect an amplifier to the metronome output to hear the result. MET,N=1 will produce 8 short clicks from the metronome output, while MET,N=2 produces 8 emphasised clicks from the output.

Test Name: MIXER
No. of tests: 1
Purpose: Tests the mixer on the audio rack.

It does this by switching each mixing port on in turn until all are on then switches them off in the reverse sequence. On the latest mixer cards (CMI-334 Rev.3) there are LEDs to monitor the test. Older cards (Rev.2) will need to be monitored with an oscilloscope or logic probe. Refer to page CMI-334-03 of the service manual (Mixer Control Circuitry); place the probe on the outputs of the 4099 IC's labelled C1 to C16 (note they should have +7v outputs).

Q777 SCSI INTERFACE ADAPER DIAGNOSTICS

Run file: **SCSITST**[ret]
So called; **SCSI TeST.**
This document is current for SCSITST Rev 1.0

The SCSI diagnostic runs a series of tests to check various sections of a Q777 board (not connected to any peripherals). An error message is printed if an abnormal result is obtained. The error message will specify the function of the test and an IC location as a guide to trouble shooting. The SCSI test must be terminated by ESC.

Test Name: SCSI
No. tests: 1
Purpose: To test the Q777 SCSI Interface adapter.

General System Diagnostic Chain Test

To run the general test program for the whole CMI,
type CHAIN TEST [ret]

This document is current for TEST.CF Rev 2.

This will test the following cards in sequence;

1. The Q-133 Debug Card (without shorting plugs)
   - with shorting plugs simply type CHAIN TEST;PL [ret]

2. The Q-209 Graphics Card

3. The Q-256 Memory Cards.
   - you should have two cards but if only one type CHAIN TEST;-MM [ret]

4. The CMI-33 Waveform Processor Card
   - type CHAIN TEST;-WP [ret] if you wish to omit this test

5. The CMI-39 Waveform RAM Cards (7 of them)
   - if you wish to test less than 7 cards (or any combination) then use the R option and type, for instance;
   CHAIN TEST;R%1-4% [ret] (tests cards 1 to 4).
   - type CHAIN TEST;-WM [ret] if you wish to omit this test.

6. The CMI-28 General Interface Test
   - type CHAIN TEST;-MP [ret] if you wish to omit this test

7. The CMI-31 Channel Card Tests (for all 8 Channel Cards)
   - type CHAIN TEST;-CH [ret] if you wish to omit this test

8. You can test the two floppy disc drives by selecting the option DD
   - i.e. type CHAIN TEST;DD [ret]

9. You can test the interrupts of the system by selecting the option I
   - i.e. type CHAIN TEST;I [ret]

10. To have the chain repeat itself continuously use the R option;
   - i.e. type CHAIN TEST;R [ret]

Waveform RAM Chain Test

The chain file WAVERAM.CF automatically tests Waveform RAM only, both with and without channels running. Use the following chain test for a full 7 RAM cards;

CHAIN WAVERAM [ret]

This document is current for WAVERAM.CF Rev 1.0

If some other number of RAM cards are required to be tested use the R option (R=1-7) as follows;

CHAIN WAVERAM;R%1,3-5% [ret]

In this instant RAM cards 1, 3, 4 and 5 will be tested.

During the test, Channel Cards are set running to further exercise the RAM. By default, all channels are used. Otherwise, include the C option (C=1-16) as follows;

CHAIN WAVERAM;C%1,2,3% [ret]

In this instance Channel card 1, side A and B, and channel card 2 side A (1a, 1b and 2a) will be tested.

The test first does one pass of the WVRAM tests and one REFRESH test without channels running. Then channels are started and the WVRAM test is run continuously. To stop the test, hit the ESC key. The last command in the chain generates an error report (errors counters are cleared at the beginning of the chain).

This is the procedure to calibrate the CMI-331 Rev.1 audio modules with the diagnostics titled AUDIOCAL.

You will need a CRO, distortion and noise meter, DVOM and an (audio rack) extender card. The mixer card is also required for the mixer tests.

This document is current for AUDIOCAL Rev 1.01

Step 1.

To test the first card, firstly put it on the extender card and boot the diagnostics disc. Type **AUDIOCAL** [ret]. Now select which particular channel(s) you wish to calibrate by typing **SEL,C=1,2,5** [ret], for instance. This will select the first channel card both sides A and B and Channel 3 side A. You can make any combination of the C option from C=1 to C=16.

Step 2. VCF Cutoff Calibration

Start with a sine wave of 1kHz.

Type **TRIM1** [ret].

This will generate a sine wave of 1kHz on all channel selected with the VCA at a maximum, the filter cut-off a maximum and the filter resonance a minimum.

Step 3.

a) Take the DVOM, place the +ve probe on pin 7 of IC J1,2 and the -ve probe on pin 1 of the same IC. Set the pot on top of J1,2 to read 25 millivolts.

b) Repeat with the +ve probe on pin 7 of IC D3,4 and the -ve probe to pin 1 of IC D5. Set the pot on top of D3,4 to read 25mV.

c) Place the +ve probe on pin 1 of IC J1,2 and the -ve probe to ground (use pin 1 of one of the XLR connectors for a ground). Set the pot on top of IC F1 to read 72mV.

d). Place the +ve probe on pin 1 of IC D5 and the -ve probe to ground. Set the pot on top of IC D5 to read 72mV.

e) Re-check the following voltage measurements with respect to ground and re-adjust if necessary.
97mV on pin 7 of IC J1,2 (adjust pot J1,2).
72mV on pin 1 of IC J1,2 (adjust pot F1).
97mV on pin 7 of IC D3,4. (adjust pot D3,4).
72mV on pin 1 of IC D5. (adjust pot D5).

f) Set pots RV1, RV2, RV3, RV4, RV5, RV6, RV7 and RV8 to midway.

Step 4.
a) adjust pot RV13 to give +4dBm at XLR-S03 (Channel A).
b) adjust pot RV14 to give +4dBm at XLR-S04 (Channel B).
This step simply sets the output level for the next test.

Step 5
Press the space bar again (TRIM1,N=2). A sinewave of 15kHz is generated and the VCF is set to $F5. Adjust pot on F1 (for Chan. A) for -3dB (or +1dBm) and adjust pot on D5 similarly for Chan. B.

Step 6. Test of Overall Bandwidth (VCF open)
Now to test the module with a range of frequencies.

Type TRIM2 [ret].

This will generate a sine wave of 20Hz on all channels selected. Monitor output S03 (Channel A) first. Check that the output level is between +1dBm and +4dBm.
Now push the Space Bar and this will call the next test frequency; 100Hz (TRIM2,N=2). Continue with frequencies 200Hz, 500Hz, 1kHz, 2kHz, 5kHz, 10kHz, and 12kHz (use the Space Bar to step to each frequency) and check that their output reading is +4dBm±0.5.

At 15kHz we would expect a 1.5dB rolloff to +2.5dBm±0.5.
At 16kHz we would expect a 2.0dB rolloff to +2.0dBm±0.5.
At 17kHz we would expect a 2.25dB rolloff to +1.75dBm±0.5.
At 18kHz we would expect a 2.7dB rolloff to +1.3dBm±0.5.
At 19kHz we would expect a 3.2dB rolloff to +0.8dBm±0.5.
At 19.5kHz we would expect a 3.6dB rolloff to +0.4dBm±0.5.
At 20kHz we would expect a 4.0dB rolloff to 0dBm±3.

Repeat the test on Channel B via output S04.

Now if all of this is o.k. proceed to step 8 otherwise continue with step 7.

Step 7.
Type R [ret]

This will repeat the test TRIM2. Re-check all the voltages in step 6 above and note carefully the problem.

Case 1: The output level has rolled-off earlier than 16-17kHz. Adjust pot on IC F1 (for Channel A) or pot on IC D5 (for Channel B) as in step 3c or step 3d but adjust the voltage to slightly less than 72mV.

Case 2: The output level drops to about +1dBm at 19.5kHz but has rolled-off much earlier (say at 12kHz) or is still high (say +4dBm or more) at 19kHz. Adjust the pot on top of IC J1,2 (for Channel A) or the pot on top of D3,4 (for Channel B) until the optimum response is obtained. You may need to re-adjust the pot on top of IC F1 (for Channel A) or of IC D5 (for Channel B).

Now go to step 4 again,first typing

TRIM1 [ret]

Step 8. VCA Distortion

Now measure distortion on both channels by typing the command TRIM3 [ret].

This will generate a 1kHz sine wave again. On your distortion meter: adjust the meter to 0 on the calibration range then switch to the read setting. Adjust pot RV12 for Channel A and RV11 for Channel B until a level of approx -71dB is obtained. For Rev.1 cards only a figure better than -68dB is considered acceptable.

Distortion is introduced mainly by the DAC and careful selection of the DAC's can improve figures by several dB. Also the VCA contributes to distortion so substitution of the dBX chip can improve the results.

Step 9. VCF Distortion

Press the Space Bar for the next test (TRIM3,N=2). The Channel Card will now play audio zero. Feed in a 15kHz 14Vpp signal from a low distortion oscillator to pin 2 of IC G12,13 (LF347) for Channel A or to pin 13 of the same IC for Channel B. (There is a special clip that can be used to do this - see Note (i) below).

For Channel A adjust pots RV5, RV6, RV7 and RV8 for minimum distortion at 15kHz. This should be -65dB but a figure better than -61dB is considered acceptable. Adjust pots RV1, RV2, RV3 and RV4 similarly for Channel B. Remove the oscillator.

Step 10. VCA Control Range

Press the Space Bar again (TRIM3,N=3). The Channel card will generate a sine wave of 1kHz with the VCA control DAC's set at $FFF, i.e. maximum attenuation (no sound). Now set the distortion meter to dBm range and set the VCA attenuation controls, pot RV9 (for Channel A) or RV10 (for Channel B), for maximum attenuation. Ideally the level should be around -91dBm however the linearity of the dBx2150 may cause this to be as high as -85dBm.

Step 11. Overall Signal/Noise Measurement.

A final check on the noise performance of the channels. Press the Space Bar again (TRIM3,N=4), a 1kHz signal will again be generated. Check again for the +4dBm level then press the Space Bar again (TRIM3,N=5) this leaves the audio channel playing a sound of zeros (i.e. nothing) with the VCA and VCF set to their maximums, hence

measurement of the relative noise performance of the DAC and audio output circuitry can be made.

The absolute noise level should be approx -88dBm however a level of -85dBm is considered a pass/fail point. This is equivalent to a relative signal to noise ratio of approx 90dB.

Step 12. Total filter cut-off

Type the command TRIM4[ret]. TRIM4,N=1 will generate a sinewave at 1kHz with the filter open at $FF. After pushing the space bar (TRIM4,N=2) the filter will be cut completely to $00 and the signal level should drop to about -45dB.

Step 13. Resonance Control Test

Pressing the space bar again (TRIM4,N=3) will generate a 19.5kHz signal with the resonance on full (at $FF). The signal should be at about +19dBm. On the next step (TRIM4,N=4) the resonance is returned to its miniumum value of $00.

Step 14. Audio Module Mixer Test

Be sure to have the CMI-334 mixer card connected for this test.
Select both sides of the channel card under test and monitor both on separate channels of a Dual Trace Oscilloscope.

a) TRIM4,N=5 sets a 1kHz sinewave going on both A and B sides of a channel card with the mixer turned off. The VCA on Channel A is fully open (maximum volume) while the VCA on Channel B is closed (zero volume). Observe the signal on one trace and zero on the other. Press the space bar.

b) TRIM4,N=6 : same as before, but with the mixer turned on.
Observe the signal on both traces. Press the space bar.

c) TRIM4,N=7 sets a 1kHz sinewave going on both A and B sides of a channel card with the mixer turned off. The VCA on Channel A is closed (zero volume) while the VCA on Channel B is fully open (maximum volume). Observe zero on one trace and the signal on the other. Press the space bar.

d) TRIM4,N=8 : same as before, but with the mixer turned on.
Observe the signal on both traces. Press the space bar.

e) TRIM4,N=9 : both VCAs are set at the maximum volume and hence the signals will mix (as the mixer is still on) producing a sinewave at a level of about 10dBm.

f) TRIM4,N=10 : pressing the space bar again will turn the mixer off, reducing the levels to about 4dBm.

Note (i)

The special test plug can be made from an IC test clip. Solder a 4K7 resistor to pin 2 and feed the signal in on the other end of the resistor for channel A. A second resistor soldered to pin 13 is used for channel B.

Mass Storage Devices 6

Contents

Floppy Disc

Hard Disc

Streaming Tape

Introduction

The FAIRLIGHT Series III is available with various options for mass data storage peripherals. A typical system is provided with one floppy disc drive, one Winchester hard disc drive, and one streaming tape drive.

The floppy disc drive is either a Mitsubishi M2896-63 a YE DATA or a YE180, double-sided, double density soft-sectored 8-inch type.

The hard discs used are currently Maxtor or Newbury Data 85 or 140Mb Winchesters, and the streaming tape drive is Archive scorpion.

Floppy disc drive

Removal, installation and shipping of discs

Refer to the Page 6.19 for the Removal and Installation procedure. Insert the shipping disk that was shipped with the unit, and close the door, whenever reshipping the disk drives. A floppy disk will suffice as a shipping disk.

Disk Drive Optioning

Optioning may only need to be done if the disk drive has been returned to a Mitsubishi service centre. If returned to a Fairlight service centre, the drive will be correctly optioned to perform correctly on the C.M.I.

Mitsubishi M2896-63 - Option blocks to be shorted

JFC, PS, SE, DC, M2, S2, I, R, IT, MS,
M0, RFA, HR, A, HUN, WP, DS, 2S, RM

Additional wire link option - Y

YE180 - Option blocks to be shorted

C, Y, and the optioning block located near the terminating resistors.

YE180 - Option blocks to be open

X, Z, H

All other option blocks to be left open. See Figure 1 for option block locations.

Note - If two (2) floppy disc drives are installed drive 0 should have no termination resistors but drive 1 should. The termination resistors are located near the 50 way gold contacts.
Mitsubishi - MIGA 150ohms J x 1
YE180 - 760-3-150ohms x 2

Mitsubishi M2896-63 Option block Location

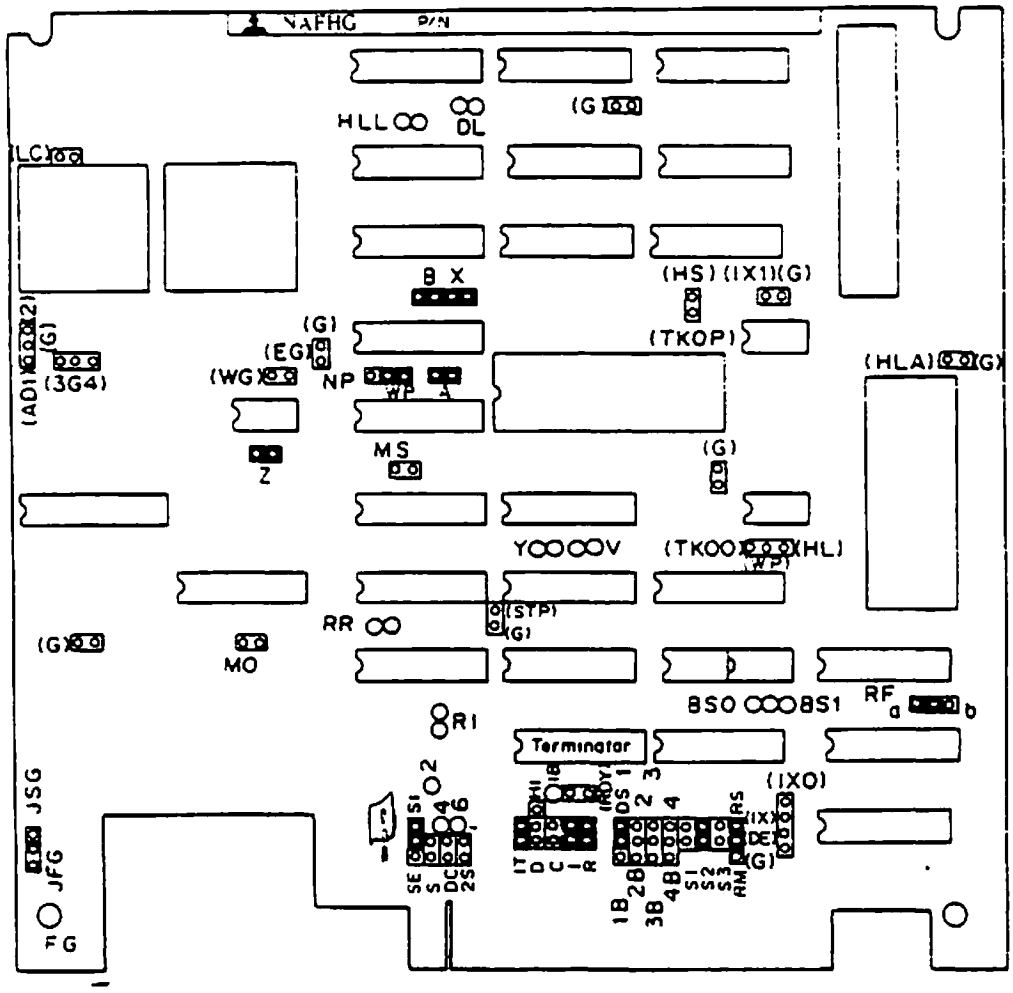


Figure 1

YE180 Option block Location

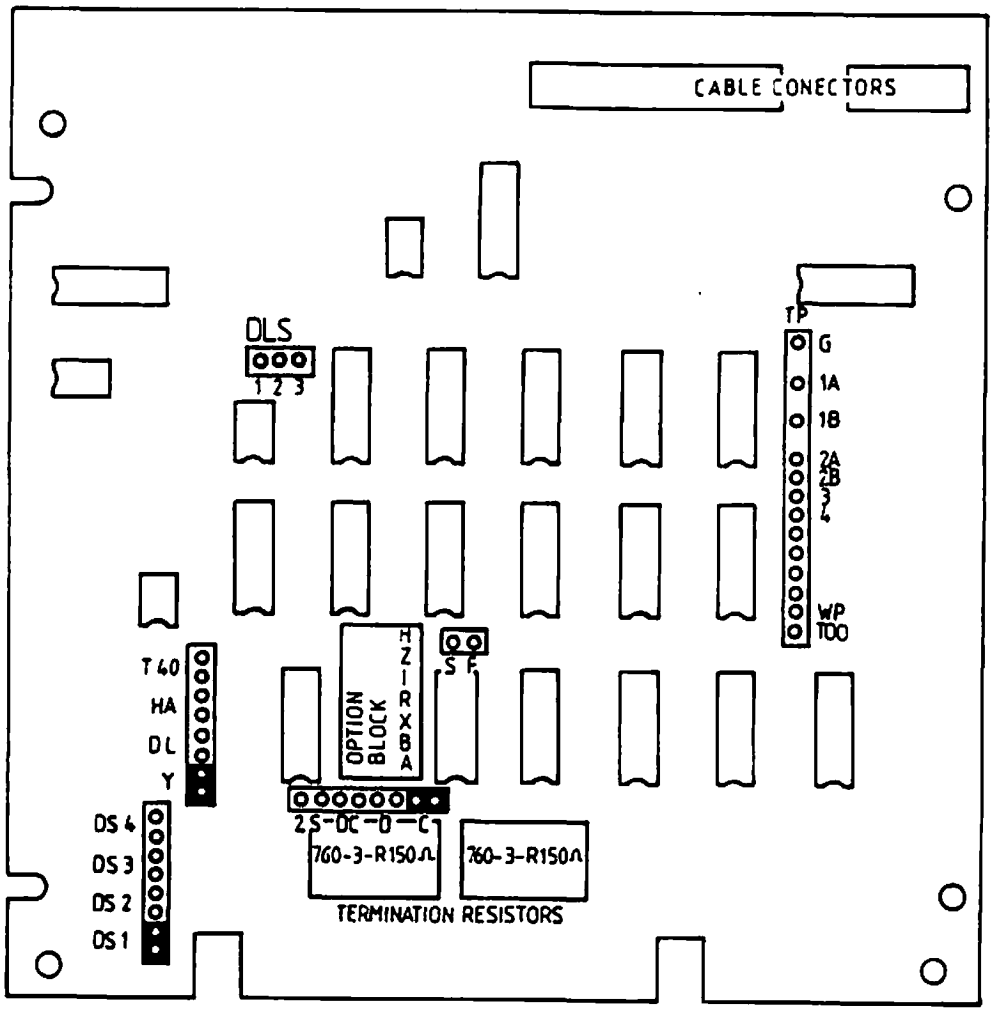


Figure 2

Disk Drive Alignment

Disk drives may require checking to account for any maladjustments which may occur during shipment. This requires the Fairlight Diagnostic Diskette containing the command DSKTST.CM.

The disk drive under test should be able to load test programs, however if the condition of the disk drive under test is suspect then another known good disk drive should be used to load the test program and used to run the tests on the faulty disk drive.

Radial Alignment

Because disk drives utilize double density disk format, radial alignment is critical and is best performed by Fairlight or Mitsubishi. To ascertain whether drive alignment is correct, run the DSKTST command from the Fairlight diagnostic disk.

Disc drive maintenance

Under normal circumstances preventive maintenance is not required on the M2896. If severely dirty environments are encountered, an occasional cleaning of the drive may be performed to assure continued reliable performance.

Only basic corrective maintenance is documented here. If it is determined that a disk drive requires more extensive repairs than are described in this section, return the unit to Fairlight Instruments for service. This document should provide sufficient information to determine whether return of the unit is necessary.

Preventative maintenance - Visual Check

Visual inspection is the first step in any maintenance operation. Always look for corrosion, dirt, wear, binds, and loose connections. Noticing these items may save downtime later.

Cleaning

Cleanliness cannot be overemphasized in maintenance of the M2896.

Caution

The head/carriage assembly is a factory-adjusted and tested assembly. Do not try to adjust or repair this internal component. Do not, for any reason, clean the read/write heads. To do so would cause severe damage to the head surfaces or head spring supports.

Floppy disk system diagnosis

The Floppy Disk System comprises of the QFC-9 Floppy Disk Controller as well as the disk drives themselves. The first step in servicing the system with an apparently faulty disk system is to establish in what subassembly the fault actually lies.

The general procedure to follow in disk system fault tracing is:

(1) Check all disk system cables, especially the 50 way flat cable for open circuits or shorts and ensure all connections are secure.
(2) Use the system test program CHECK to determine if the fault is in the drive itself (or the diskette) or the disk controller/DMA data transfer system.
(3) If the disk drive is faulty, use DSKTST to further analyse the fault.
(4) Otherwise, refer to the CMI Mainframe manual to trace the fault in the QFC-9 controller.

Test Program CHECK

Allows checking of;

- Cyclic Redundancy Check (CRC) errors
- Data transfer between memory and disk
- RAM bit corruption errors

Command Syntax

CHECK <UNIT>,<HEXNUM>;<OPTIONS>

where <UNIT> = <COLON><NUMBER>

<HEX DIGIT> = number 1 to 9 and/or letter A to F

e.g., CHECK<return> performs CRC on DRIVE 0.
CHECK :1<return> performs CRC on DRIVE 1.
CHECK :1;V<return> performs CRC on DRIVE 1 with V option.

1) Disk Integrity Check

Options: none required

This is the default CHECK routine. Entire disk in specified drive is read to check for CRC errors.

2) Read Data D.M.A. Verify

Option: V

Reads entire disk in specified drive twice, into separate blocks of memory and verifies data against itself.

3) Write Data D.M.A. Verify

Options: W,D (May be used together)

The W option creates a file, writes distinctive data to each sector of the file and reads each sector of the file back, twice, into different areas of memory for verification. All unfree disk space will be allocated to the file.

The D option is a destructive (to the disk contents) test which writes a unique "ADD -29" pattern to each sector in an interleaved fashion, reads it back, and verifies the data.

Interleaving of blocks ensures track boundaries are continually being crossed. A delay can be introduced using the "T" option (see below) to isolate head-load timing problems.

4) Other Options

Option	Use with	
R	W	use random number pattern instead of "29" pattern
P=XX	W	use pattern XX where XX = <hex number> write the pattern to disk, read back and verify
E=XX	all	print error if total recoverable where XX = <hex number>. Default value is 0.
T=XX	all	delay XX*10 ms. after a read/write where XX = <hex number>
C	all	test continously alternating between 'add-29' and a random number pattern
L	all	all error messages printed on printer

5) Error messages

a) Disk Read/Write Errors
These are of the form;

**PROM I/O ERROR -- STATUS = <status byte> AT h DRIVE i - PSN j

where h is not significant
i = drive number
j = physical sector number at which the error occurred

and the status byte can be interpreted as follows:

31 data C.R.C. error
32 disk is write protected
33 disk is not ready for some reason
34 deleted data address mark read
35 abnormal command termination
36 invalid sector address
37 seek error (track not found)
38 data mark read error
39 address mark read error

b) Verify Errors

When a verify error is encountered the offending disk sector is re-read into the QDOS sector buffer and matched against system RAM to determine where the error came from. The program then reports the corresponding address in RAM, the data expected, the erroneous data, the physical sector number of the disk where the error occured, and the byte offset within the sector.

6) Termination

Test is terminated by -
ESC key (sets system error status word)
More then 20 errors logged
User supplied iteration counter expired (default 1)

System error status word will be set if any error condition has been reported.

Test Program DSKTST

DSKTST comprises of five main test routines and a number of utility commands. The main routines are as follows -

#1 Write/read test (destructive)
#2 Read C.R.C. test (non-destructive)
#3 Worst case seek test (non-destructive)
#4 Worst case data pattern R/W (destructive)
#5 Sector/drive uniqueness (destructive)

CAUTION

Destructive tests will overwrite the diskette in the drive under test with a testing pattern.

Tests can be run separately or in destructive/non-destruct groups by typing as follows:

DN, (0 or 1 or B) [,X]<CR> (Do all non-destruct tests)

DD, (0 or 1 or B) [,X]<CR> (Do all destructive tests)

ST#<tests>,(0 or 1 or B)[,X]<CR>
where <tests> = up to 10 test numbers separated by '-'

The extended test option X accumulates error counts over a number of passes.

ESC key will abort test in progress.

Typing OS<return> will return the user to QDOS and reboot the system.

Examples: DN,0<return> does all non-destructive tests on drive 0 only.

ST#1-3-5,B,X does tests 1,3 and 5 on both drives with error count accumulation.

If stop on error option is selected (in answer to a prompt) the user may choose -

C continue
L loop
R reset stop on error

if an error stop occurs.

Error reporting
Error printouts take the following form :

<drive no.> <error type> <track no> /<physical sector no> <*>

Presence of '*' indictes a "hard" disk error,

e.g. 1 E3 1F /0325 *

means :- drive no 1
error type 3 (E3)
track no 1F
p.s.n 0325
error was not recoverable on retry (*)

If after three retries the error persists, it will be logged as a hard error (indicated by *).

Error types are as follows (per QDOS ROM codes):

E1 data CRC error
E2 disk is write protected
E3 disk is not ready for some reason
E4 deleted data address mark read
E5 abnormal command termination
E6 invalid sector address
E7 seek error (track not found)
E8 data mark read error
E9 address mark read error

Additional error types are :-

E@ data read back is not the same as data written

Additional error types from the drive uniqueness test are :-

EA body of data buffer is not zero after test data
EB unique data for this drive/sector is incorrect.

Error Graphs

Errors may be summarised by use of the 'PG' command. This summary plots the track no. as the vertical ordinate and the number of errors as the horizontal ordinate.

A horizontal line may contain up to 11 error types (codes) with each character representing (n*horizontal scale) errors.

The error graph is divided into two blocks. The left hand block relates to drive 0 errors, the right hand block to drive 1.

The graph is printed starting at the first track with errors logged and finishes with the last track with errors logged.

To stop the display rolling off the screen, <control W> can be used to stop printing. Subsequent carriage returns will print a little at a time, an escape will terminate the 'PG', and any other character will resume continuous printing.

In the case of double sided systems, each disk 'cylinder' is considered as two tracks, so even track numbers correspond to side 0 of the disk and odd track numbers correspond to side 1.

Utility Commands
Commands for utility programmes are as follows

HD,d,hhhh	Head load timing test on drive d at speed hhhh (100 mS = D8F0)
IX,d	Index sensor alignment test on drive d. t1=tk 1. t2=tk 76.
AT,d,s	Read data amplitude test on drive d. s is optional side select (0 or 1). t1=tk 0. t2=tk 76.
RA,d,s	Radial alignment test on drive d. s is optional side select (0 or 1) t1=0-38. t2=77-38. t3=39-38. t4=37-38.
AZ,d,s	Head azimuth test on drive d. s is optional side select. t1=0-76. t2=75-76.
T0,d	Track zero sensor alignment test on drive d. t1=1-2 lp. t2=0-1 lp. t3=0-2 lp.
SK,d,s	Head skew test on drive d. s is optional side select (0 or 1). t1=1-76 lp.
RS,d,hhhh	Read sector hhhh from drive d to buffer
WS,d,hhhh	Write buffer to sector hhhh on drive d
DB	Display buffer in hex and ascii
FB,hhhh	Fill buffer with repeating pattern hhhh

The running test may be aborted by escape key.
The next test of the sequence is entered by depressing space key.
Tests followed by letters "lp" move head between tracks shown.
Some tests require the appropriate alignment diskette and ask that it be inserted. Other tests require a scratch diskette and ask that it be inserted.
Typing OS<return> will return the user to the operating system (reboot).

Floppy disc removal

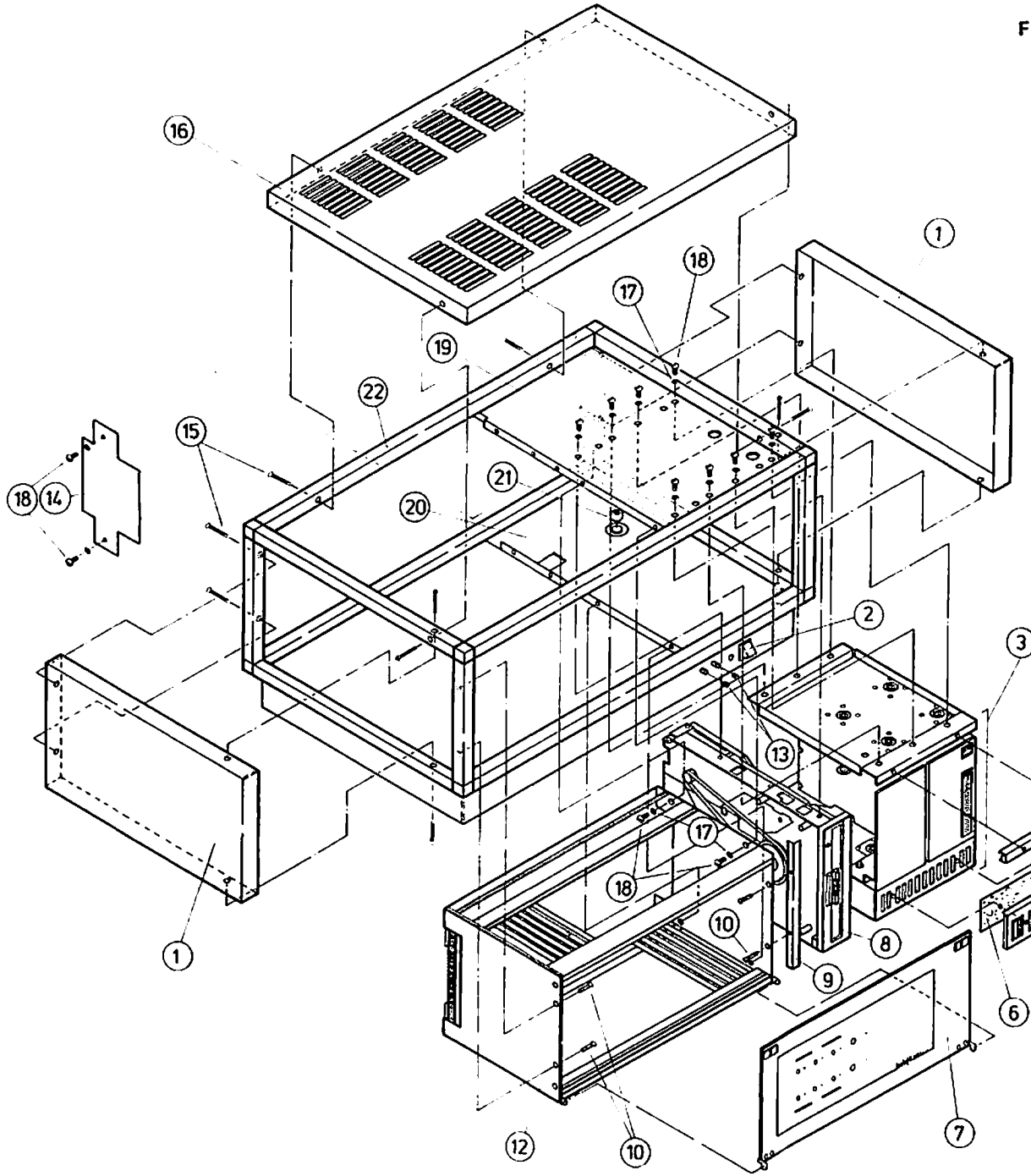
1. Remove top, bottom and blank plate on the rear of the CMI.
2. Place the machine upside down and remove the CMI-310 Card.
3. Remove the two(2) holding screws underneath the CMI-310 securing the floppy drive.
4. Place the machine on its feet.
5. Disconnect the 50 way cable and the four(4) pin power connector from the floppy drive.
6. Remove the two(2) screws holding the drive to the top plate.
7. Remove the floppy drive through the front of the frame.

Floppy Drive Re-assembly

1. Replace the drive through the front of the frame.
2. Install the screws which secure the drive to the top plate.
3. Re-connect the four(4) pin power cable and the 50 way signal cable.
4. Place the machine up-side down.
5. Re-install the CMI-310 card and all the cables.
6. Power up the machine and check for correct operation.
8. Refit the panels.

Note: The machine is placed upside down instead of on its side so that screws do not accidently fall into the switchmode power supply unit.

FLOPPY DISK REMOVAL AND RE-ASSEMBLY
DM 3003



REF. No.	PART No.	DESCRIPTION
1	300/377	END COVER, MAINFRAME
2	D8023	SWITCH, MAINS ENABLE
3	HARD DISC ASSEMBLY-SEE DRAWING #DM3142	
4	300/395	PANEL, HARD DISC COVER, UPPER
5	300/394	PANEL, HARD DISC COVER, LOWER
6	P0020	FILTER, DUST
7	DIGITAL CARD CAGE FRONT PANEL-SEE DRAWING #DMC044	
8	G5018	FLOPPY DISC DRIVE
9	300/432	BLANK PLATE, FRONT PANEL
10	H1072	SCREW, PZ CSK, M4 X 10
12	300/368	END CHEEK,DIGITAL CARD CAGE
13	G5426	KNOB
14	300/409	BRACE, CARD CAGES
15	H1070	SCREW, PZ CSK BLACK, M4 X 25
16	300/407	PANEL, MAINFRAME TOP COVER
17	H0035	WASHER, STAR,M4
18	H1074	SCREW, PZ PHD,M4 X 6
19	300/388	MOUNTING, DISC DRIVE PLATE,UPPER
20	300/389	MOUNTING, DISC DRIVE PLATE,LOWER
21	H2014	NUT, NYLOC, M5
22	300/422	FRAME, MAINFRAME TUBING SET

Introduction

The following information has been included to reduce the risk of damage to the hard disk due to mishandling.

Power Failure

Failure of either of the DC voltages (+12, or +5VDC) will cause an emergency retract. (ie withdrawal of heads to the landing zone.)

Practical Tips

Many problems occur during the installation and commissioning of equipment before experience with a product has been acquired. Common points regularly causing difficulties include:

1. Drive unit number Select Link not fitted in the appropriate position.

2. Terminator not fitted to the last drive in a daisy chain configuration.

3. Terminators not removed from all except the last drive in a daisy chain configuration.

4. Connectors not fitted correctly to cables or not pushed fully into drive connections. Make sure you fit polarising keys to your cable connectors.

5. If the drive does not run-up check that power to the drive is on.

6. Do not operate the drive after large changes of temperature. For example, if a drive has been brought from very cold conditions and installed in a warm area allow an hour for thermal stabilisation before powering up.

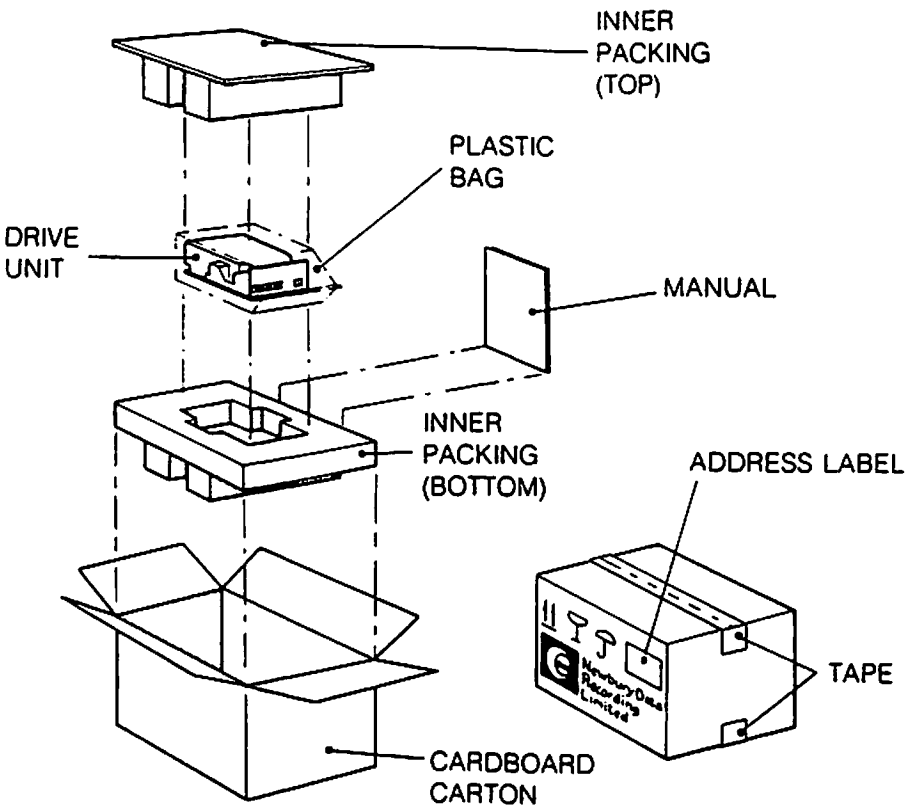
7. Avoid rough handling of the drive.

8. Avoid moving the drive when the spindle is running down as this could result in the heads contacting and damaging the moving disc surface. The discs take about half a minute to come to rest after the power has been switched off.

9. Always allow some spare capacity when allocating files to the drive. On any disk surface a few small media flaws occur either inherant from new or which have developed during use. Each of these cases can result in data not being reliably recorded or retrieved so a method of avoiding these areas is necessary. Normally the host system will find defective areas through the use of an error detection scheme and will allocate alternative locations. Note that this requires spare sectors to be available.

Packing and Unpacking hard disc

If a hard disk has to be replaced the following instructions should be followed. The drive is shipped in a sealed container as shown in the diagram below. This is designed to protect the drive from humidity, vibration and shock.



Upon receipt of the unit from the carrier, inspect the container for damage, then open the container and unpack the drive. Save all packing materials in case reshipment becomes necessary. Labels on the drive and on the packaging contain serial and part numbers. This information must be quoted in any communications about the drive.

NOTE

With no power applied to the drive the heads are automatically positioned over the non-data , dedicated landing zone on each disc surface The automatic shipping lock solenoid is also engaged.

Operating environment

The drive is designed to operate in a standard office environment. High relative humidity conditions should be avoided to prevent the possibility of condensation. Also avoid low relative humidity conditions to prevent particle accumulation by static attraction. The drive is not intended for use in a harsh environment with high dust and dirt concentrations.

The system is booted from the special CMI initialisation floppy disk. When the system is loaded and the Shell prompt appears, type:

CMI INIT<CR>

A query appears:

```
Initialize SCSI drive /SC00 (/K0, /K1, /C0) (y/n)?
```

To continue, type:

Y<CR>

A heading will appear:

```
************Adaptec hard disk format program***************

Enter drive type:
```

At this point, if you have not already done so, examine the hard disk drive and note the drive model number (e.g. xt1140, NDR1140, xt1085, V185), the serial number and the error map. These should all be printed on the drive's outer casing. If you make a mistake in this step, a list of valid drive types appears. Match your drive type to one of the entries in this list and type the entry, followed by a <CR>. Next you will see:

```
Enter serial No:
```

Type the serial number you have noted. The next prompt is:

```
Clear the defect buffer: (y/n)
```

Type:

Y<CR>

The next prompt is:

```
Enter defect list - HEAD CYLINDER BYTE ,<RETURN> <RETURN> TO
END
```

Type in the error map numbers in the order indicated. When the list is completed, type a second carriage return to exit the defect list editor:

<CR>

When completed a prompt reading:

```
Edit defect number:
```

will appear on the screen. You may now correct any errors due to typing inaccuracies, by typing first the number of the entry, then typing the entry. Strike <CR> to exit this editor. The screen will then read:

```
Is the the defect list now correct and complete (y/n)?
```

Type:

Y<CR>

```
A hard format will completely erase all disk data - proceed
(y/n)?
```

Type:

Y<CR>

Note: Now you have typed in your error map there is no need to repeat the defect listing procedure when formatting the same disk in future. The defect map is saved in the root directory of the CMIGEN floppy disk under a name starting with `map_` followed by the drive type and serial number of the hard disk as entered above, e.g.

```
map_xt1140_1120
```

Formatting with an existing error map on the floppy disk.

This is essentially the same procedure as above. The `CMI_INIT` program detects the existing map corresponding to the drive type and serial number you enter and allows you to edit the existing map rather than typing a new one in. After entering the serial type, the query appears:

```
Use existing defect map file: (y/n)?
```

Type:

Y<CR>

There is a query:

```
Add more defects:(y/n)?
```

If you have extra defects to add to your file add them at this point. Then the procedure is as before:

```
A hard format will completely erase all disk data -
proceed(y/n)?
```

Type:

Y<CR>

The hard disk will take approximately 15 minutes to format.
No more user prompts are required.

Hard disk removal

1. Remove top, right side, bottom and the blank plate on the rear of the CMI.
2. Place machine upside down and remove CMI-310 Card.
3. Locate the five (5) holding screws (underneath CMI-310) of the disk mounting plate and remove.
4. Place machine on its feet.
5. Disconnect the 50 way SCSI connector and four (4) pin power connector from Hard Disk and Adaptec controller, and the two (2) pin connector from the Hard Disk fan.
6. Remove faceplate and grill of the Hard Disk (at the front of the CMI).
7. Remove the six (6) screws holding the Hard Disk assembly to the top plate.
8. Remove Hard Disk assembly through the side of the frame.

Hard disk re-assembly

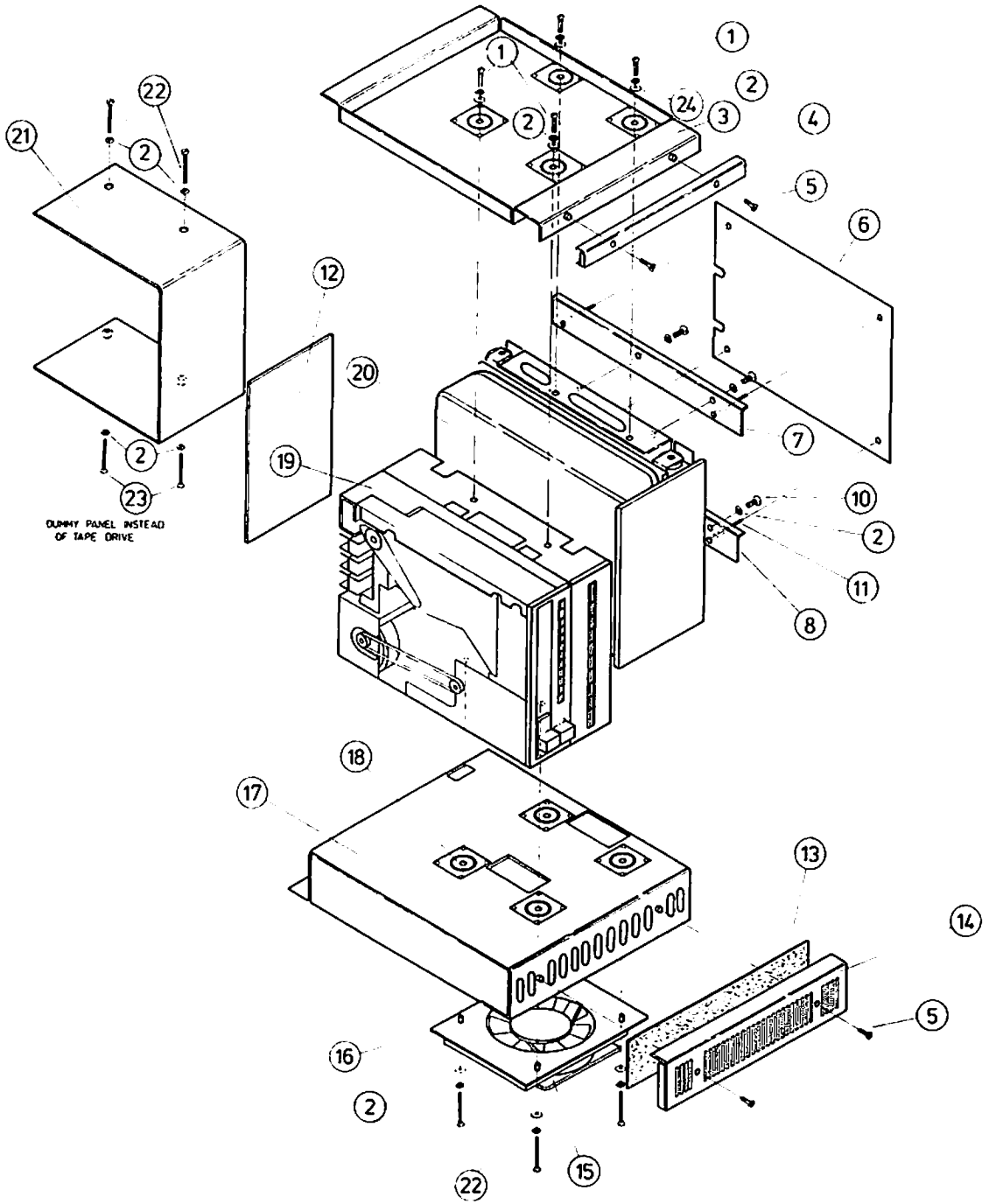
1. Replace hard disk tape assembly through the side of the CMI.
2. Re-connect the 4-pin power cables to the hard disk and note that two (2) power cables go to the tape assembly.

ENSURE THE CABLE TO THE FAN IS CONNECTED

3. Install the screws which secure the top of the hard disk tape assembly.
4. Place machine upside down.
5. Re-install the five (5) screws on the bottom of the hard disc assembbly.
6. Re-install the CMI-310 card and all cables.
7. Power up machine and check for proper operation.
8. Refit panels.

These steps are illustrated on the hard disk exploded diagram on the next page

HARD DISK REMOVAL AND RE-ASSEMBLY
DM3142



REF. No.	PART No.	DESCRIPTION
1	H1080	SCREW, PHD, 6/32" X 1.25"
2	H0035	WASHER, STAR, M4
3	300/391	MOUNTING PLATE, HARD DISC UPPER
4	300/395	COVER PANEL, HARD DISC UPPER
6	G8080	CARD, CONTROLLER, ADAPTEC
7	300/397	SUPPORT, CONTROLLER CARD UPPER
8	300/398	SUPPORT, CONTROLLER CARD LOWER
10	H1052	SCREW, PHD, 6/32" X 1/4"
11	G5138	STANDOFF PILLAR, CARD
12	G0416	PANEL, PLASTIC DUMMY
13	P0020	FILTER, DUST
14	300/394	COVER PANEL, HARD DISC LOWER
15	G5434	FAN, 80 M.M.
16	300/393	PLATE, HARD DISC FAN
17	300/390	MOUNTING PLATE, HARD DISC LOWER
18	G8083	SHOCK MOUNT, RUBBER
19	G8092	DRIVE, STRM TAPE
20	G5020	DRIVE, HARD DISC 140MB
21	300/396	PANEL, HARD DISC DUMMY
22	H1077	SCREW, PHD, 6/32" X 1"
23	H1079	SCREW, PHD, 6/32" X 1.25"
26	H0007	WASHER, FLAT, 4BA

Introduction

The Streaming Tape unit consists of an Archive Scorpion drive with an Emulex controller mounted in a single frame adjacent to the Hard Disc drive. The Tape unit is physically the same size as the Hard Disc and mounting and cabling notes referred to in the Disc section are applicable.

Handling

The Tape unit is more robust than the Disk, however it must always be handled gently. The head assembly of the tape drive is particularly sensitive and any excessive flexure may result in loss of alignment causing data errors.

Cables

Note that the Streaming Tape unit requires two power cables to operate correctly. The 50 way SCSI cable must be plugged into the Emulex controller board which mounts above the tape drive.

Controller

The Emulex controller is normally an integral part of the Tape unit and requires no attention, however in some cases it may be necessary to separate the controller and the drive. The following points should be observed carefully.

1. Remove/attach the controller board by adjusting the 4 Allen screws holding it in place.
2. Note the ribbon cable between the controller and the drive.
3. Ensure the terminating resistors are NOT in the controller card.
4. SW1 should only have pos1 and pos5 set to be ON.
5. Jumper E-F should be shorted.
6. Faulty controller boards should be returned for replacement.

Maintainance

Apart from normal head maintainance (ie. cleaning) there is no maintainance procedure. A drive which has been determined faulty must be replaced.

Adaptec Controller

The adaptec ACB-4000A controller is part of the Hard Disk assembly. The controller requires one power cable and connects to the 50 way SCSI ribbon cable. The controller connects to the disc via a 34 way cable and a 20 way cable. Secure connection of all these cables is necessary for reliable disc operation. The only practical method of determining controller malfunction is by board swapping. In the unlikely event of failure, the faulty board must be replaced.

Alphanumeric Keyboard

7

Contents

CMI-314 Power Supply Adapter

Graphics Pad Alignment

Introduction

The PREH alphanumeric keyboard has been modified for use with the Fairlight Series III CMI, from keyboard Rev. 1.56 onwards:

-The key layout has been changed slightly to incorporate the new ADD(sharp), SUB(flat), SET(natural) and CLEAR keys.

-SHIFT and CTRL can now be used with the graphics pad.

-New graphics and key control codes overcome the problem experienced by some users of characters being detected by the CMI when the graphics pen is used.

The modified keyboard internal software (in ROM) requires that the user have a CMI system of version 2.03m or later. Systems from v2.03m onward expect to use the new keyboard by default. IF YOU DO NOT HAVE A MODIFIED KEYBOARD(ie revision nr 1.56 or later) the system may be converted to use the older keyboard software by typing "/k0/oldkbd". The system may be converted back to using the newer keyboard software by typing "/k0/newkbd".

Keyboard diagnostic mode

A diagnostic mode has been added-keys or graphics pad frames now cause ASCII to be sent, enabling the key/graphics pad operation to be easily checked. Entering the diagnostic mode causes the string "Build .1>/null" to be sent, hence it is best used with OS9 running.

To use diagnostic mode

Exit to OS9 if currently in the CMI system. Toggle the keyboard into diagnostic mode by pressing key combination:

CTRL/SHIFT/REP/DEL (ie all at the same time!)

The keyboard should send the string "Build .1>/null" and print a message with its revision number.

If nothing happens, the keyboard probably hasnt been updated to version 1.56 or later.

If the system hasn't received the build command properly it will try and act on the diagnostic messages (ie the disk will be accessed every time a complete diagnostic message is received), so if this should happen toggle the keybard out of and back into diagnostic mode by pressing the

CTRL/SHIFT/REP/DEL key combination twice(slowly).

Pressing a key will cause the keyboard to send the name of the depressed key, enabling operation of all key switches to be checked.

Touching the pen on the graphics pad will cause graphics pad data to appear on the screen. Check that the data responds to the pen button, the SHIFT and CTRL keys, and pen up/down, and that the 'X' and 'Y' coordinates may be varied from 0 to 3ff (hexadecimal).

Toggle the keyboard out of diagnostic mode by pressing

Introduction

The PREH power supply adaptor is used to allow the new keyboard to operate from unregulated voltages available from the toroidal transformer in the series III. The PREH keyboard is an alpha-numeric type keyboard with a graphics tablet and pen for manipulating a cursor on the screen. This description of circuit operation applies only to the power supply adaptor card. For information or servicing of the keyboard electronics FAIRLIGHT or PREH, the manufacturer, should be contacted.

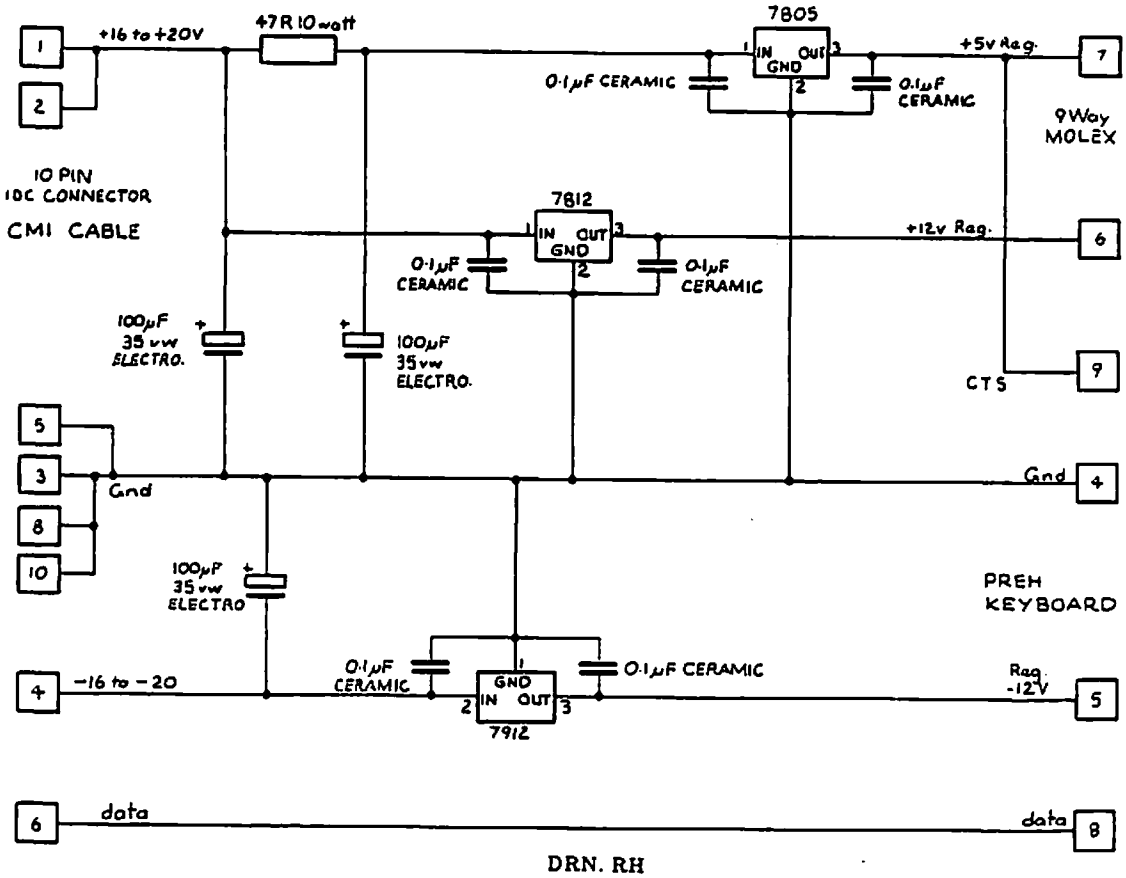
Input/Output

The power supply receives + and - 16volts and returns regulated +5, +12 and -12volts. The circuit board is carried within the PREH keyboard bolted to the aluminium key chassis which acts as a heatsink. The on board regulators of the keyboard are not used due to inadequate heatsinking of the +5volt IC regulator.

Circuit Description

(Refer schematic CMI314-0)

Incoming voltage in the range +16 to +22 is smoothed by a 100uF electrolytic and fed to a 7812 12 volt regulator. The same supply feeds through a 47R 10watt resistor to dissipate some heat, filtered by another 100uF and regulated by a 7805. The -12 volt supply is smoothed in a similar manner however a 7912 regulator is used.
Note: CMI314 carries the data and supplies the keyboard with the necessary CTS signal for correct operation



Graphics pad alignment

The alignment may be carried out either in the CMI system by viewing the cursor on the screen, (for keyboard ROMs KBV1.56 onwards only) in OS9 by using the keyboard in its diagnostic mode.

Alignment in the CMI

1. Lightly mark the top right and bottom left corners of the exposed graphics pad (eg with soft pencil) before removing the top, so that the allowable pen movement range may be readily seen. Remove the four screws from along the top rear of the keyboard casing, and lift off the top cover.

2. Boot the CMI system so that the graphics cursor may be seen on the screen.

3. There are FOUR adjustable potentiometers along the top of the main printed circuit board. Touch the graphics pen in the bottom left corner of the pad on the mark made in Step 1. Note: Touching the pad slightly inside the mark (i.e 1-2 millimetres) will ensure that adequate graphics cursor movement is obtained.

4. Adjust the LEFTMOST potentiometer so that the cursor is just on the bottom of the CMI screen.

5. Adjust the RIGHTMOST potentiometer so that the cursor is just on the LEFT edge of the CMI screen.

6. Touch the pen in the top right corner of the pad.

7. Adjust the potentiometer which is second from the LEFT so that the cursor is just on the TOP of the CMI screen. Do not move the already set potentiometers!

8. Adjust the potentiometer second from the RIGHT so that the cursor is just on the RIGHT hand edge of the CMI screen.

9. Replace the top cover temporarily and check that the pad can move the cursor over the whole screen. If not, repeat steps 3 to 8.

Alignment in OS9

1. Repeat step 1 above.

2. Exit to OS9 if currently in the CMI system.Toggle the keyboard into diagnostic mode by pressing key combination CTRL/SHIFT/REP/DEL (ie all at the same time!). The keyboard should send the string "Build .1>/null" and print a message with its revision number.
   If nothing happens, the keyboard probably hasnt been updated to version 1.56 or later.
   If the system hasn't received the build command properly it will try and act on the diagnostic messages (ie the disk will be accessed every time a complete diagnostic message is received), so if this should happen toggle the keybard out of and back into diagnostic mode by pressing the CTRL/SHIFT/REP/DEL key combination twice (slowly).
   Note that touching the pen on the graphics pad will cause graphics pad data to appear on the screen. Check that the data responds to the pen button, the SHIFT and CTRL keys, and pen up/down, and that the 'X' and 'Y' numbers change as the pen is moved.

3. There are FOUR adjustable potentiometers along the top of the main printed circuit board. Touch the graphics pen in the bottom left corner of the pad on the mark made in Step 1. Note: Touching the pad slightly inside the mark (i.e 1-2 millimetres) will ensure that adequate graphics cursor movement is obtained.

4. Adjust the LEFTMOST potentiometer so that the 'Y' data just becomes 000.

5. Adjust the RIGHTMOST potentiometer so that the 'X' data just becomes 000.

6. Touch the pen in the top right corner of the pad.

7. Adjust the potentiometer which is second from the LEFT so that the 'Y' data just becomes 3ff (this is the Y cordinate in hexadecimal). Do not move the already set potentiometers!

8. Adjust the potentiometer second from the RIGHT so that the 'X' data just becomes 3ff.

9. Replace the top cover temporarily and check that the pad can change both 'X' and 'Y' data from 000 to 3ff. If not, repeat steps 3 to 7.

10. Toggle the keyboard out of diagnostic mode by pressing

    CTRL/SHIFT/REP/DEL.

11. Replace the four screws in the bottom casing.

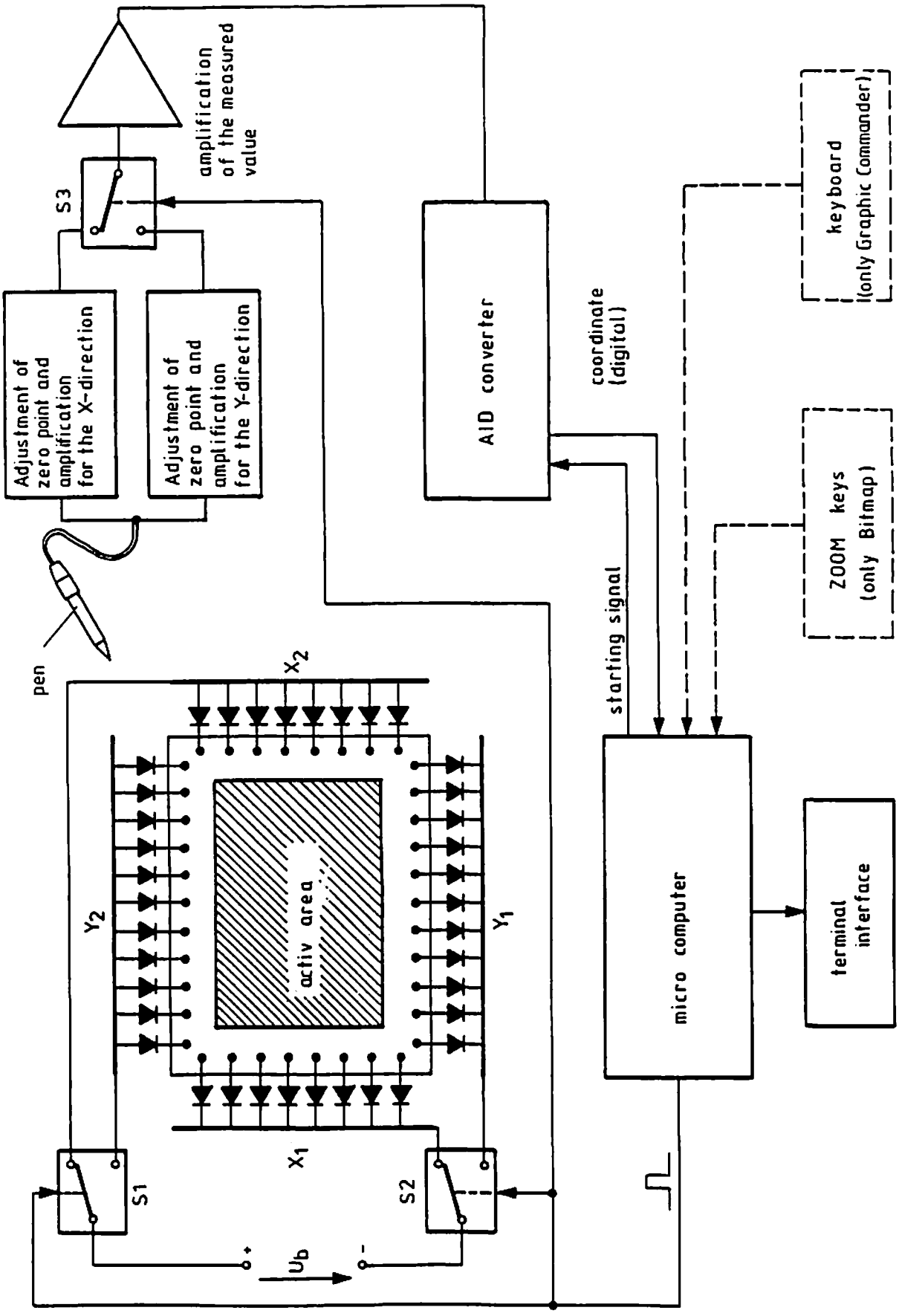
Graphics data frame format

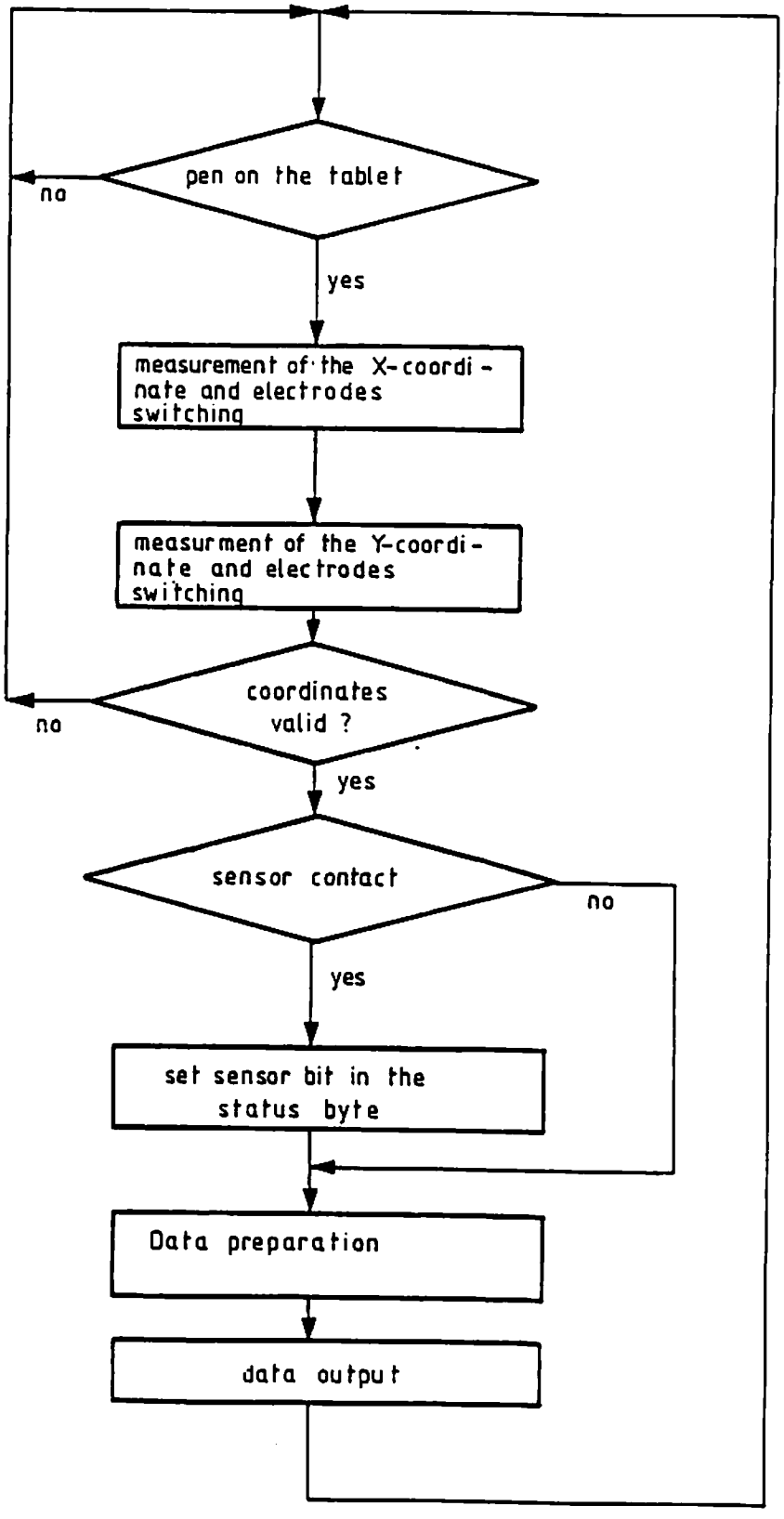
For each X/Y coordinate a frame of six bytes is sent:

byte 1: 10000000	control byte
byte 2: 111ptsc0	status byte
	s=SHIFT key down
	c=CTRL key down
	t=TOUCH sensor on
	p=PEN on pad
byte 3: 111xxxxx (low)	
byte 4: 111xxxxx (high)	X coord (10 bits)
byte 5: 111yyyyy (low)	
byte 6: 111yyyyy (high)	Y coord (10 bits)

Data Rate: 9600 Bit's

Data format
1 startbit, 8 data bits and 1 stopbit. If the data transmission stops, the TXD-signal will be logic "high". (By using the RS 232C logic "high" is between -3V and -25V


amplification of the measured value
S3
Adjustment of zero point and amplification for the X-direction
Adjustment of zero point and amplification for the Y-direction
A/D converter
coordinate (digital)
keyboard (only Graphic Commander)
ZOOM keys (only Bitmap)
starting signal
pen
X2
Y2
Y1
X1
activ area
S1
S2
Ub
+
-
micro computer
terminal interface


pen on the tablet
no
yes
measurement of the X-coordi-nate and electrodes switching
measurment of the Y-coordi-nate and electrodes switching
coordinates valid ?
no
yes
sensor contact
no
yes
set sensor bit in the status byte
Data preparation
data output

Keyboard code table

Shift lock(alphalock): toggled on and off by ALPHALOCK key. When on, Normal and SHIFT codes are swapped for the 26 letter keys.

Automatic key repeat : initial delay of approx. 1 sec, then .. repeats per second.If REP key is pressed, delay is removed and ..repeats per second

Break function: activated by pressing CTRL/ESC

Notes : **up arrow, down arrow, left arrow, right arrow,** are standard cursor control arrow symbols.
Sharp, natural, flat, are musical accidental symbols.
CTRL + SHIFT gives same code as **CTRL.**

Pos.	Name	repeat	Normal	SHIFT	CTRL
A01	REPEAT				
A02	[{	*	5B	7B	1B
A03	] }	*	5D	7D	1D
A04-07	unused				
A08	space	*	20	20	20
A09-12	unused				
A13	ADD (sharp)	*	0E	CE	C6
A14	SUB (flat)	*	0F	CF	C7
A15	SET (natural)	*	19	D9	D1

Pos.	Name	repeat	Normal	SHIFT	CTRL
B01	SHIFT				
B02	\ |	*	5C	7C	1C
B03	Z	*	7A	5A	1A
B04	X	*	7B	5B	18
B05	C	*	63	43	03
B06	V	*	76	56	16
B07	B	*	62	42	02
B08	N	*	6E	4E	0E
B09	M	*	6D	4D	0D
B10	, <	*	2C	3C	BC
B11	. >	*	2E	3E	BE
B12	/ ?	*	2F	3F	BF
B13	SHIFT				
B14	unused				
B15	RUB OUT	*	7F	7F	7F

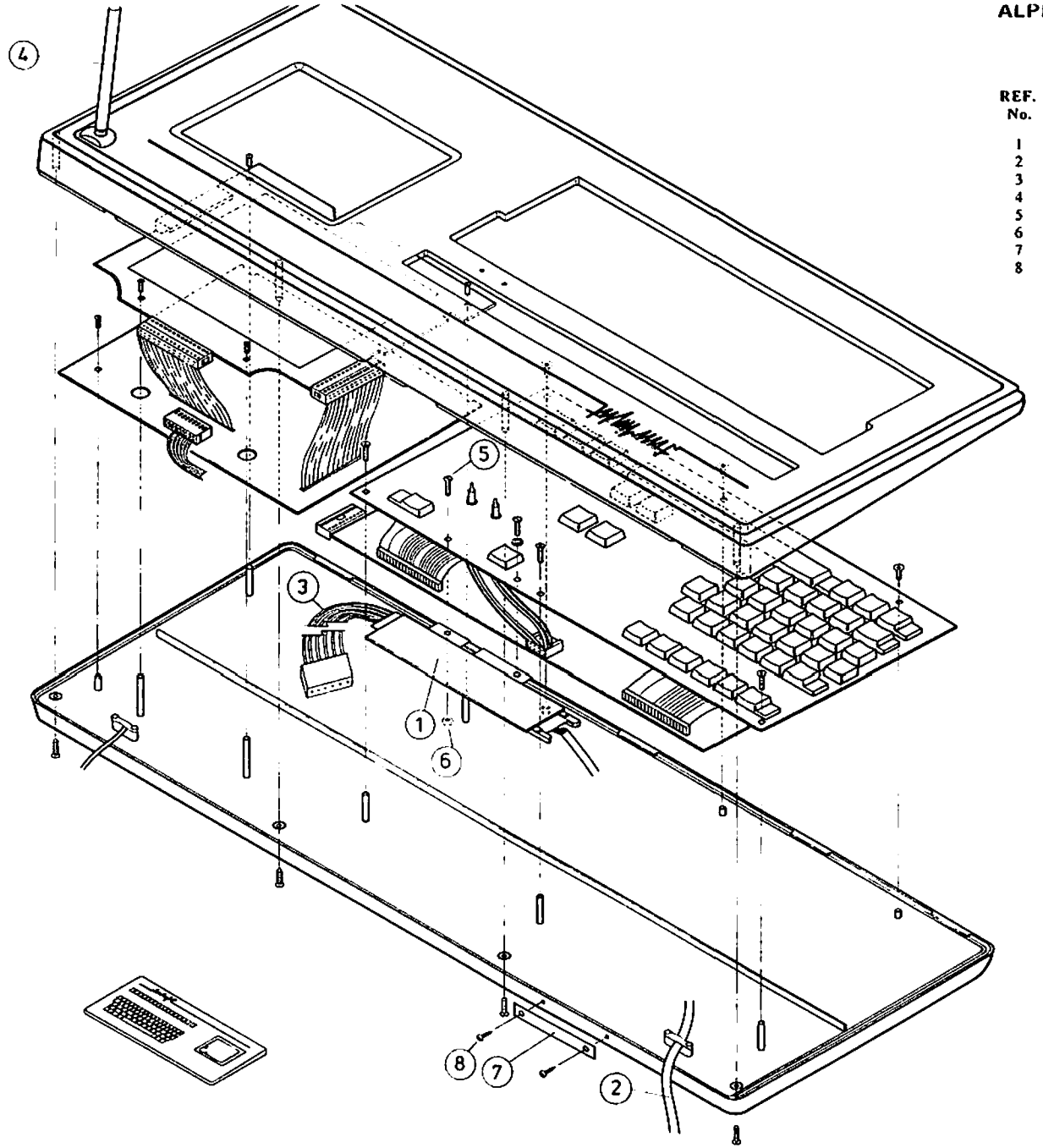
Pos.	Name	repeat	Normal	SHIFT	CTRL
C01	CTRL				
C02	ALPHALOCK				
C03	A	*	61	41	01
C04	S	*	73	53	13
C05	D	*	64	44	04
C06	F	*	66	46	06
C07	G	*	67	47	07
C08	H	*	68	48	08
C09	J	*	6A	4A	0A
C10	K	*	6B	4B	0B
C11	L	*	6C	4C	0C
C12	; +	*	3B	2B	BB
C13	: *	*	3A	2A	BA
C14	^AA-	*	5E	7E	1E
C15	RETURN	*	0D	0D	0D

Pos.	Name	repeat	Normal	SHIFT	CTRL
D01	ESC	*	1B	1B	BREAK
D02	TAB	*	09	C9	C1
D03	Q	*	71	51	11
D04	W	*	77	57	17
D05	E	*	65	45	05
D06	R	*	72	52	12
D07	T	*	74	54	14
D08	Y	*	79	59	19
D09	U	*	75	55	15
D10	I	*	69	49	09
D11	O	*	6F	4F	0F
D12	P	*	70	50	10
D13	@ `	*	40	60	00
D14	(up arrow)	*	1C	DC	D4
D15	(down arrow)	*	1D	DD	D5
D16	CLEAR	*	0C	CC	C4

Pos.	Name	repeat	Normal	SHIFT	CTRL
E01		*	5F	5F	1F
E02	1 !	*	31	21	B1
E03	2 "	*	32	22	B2
E04	3 #	*	33	23	B3
E05	4 $	*	34	24	B4
E06	5 %	*	35	25	B5
E07	6 &	*	36	26	B6
E08	7 '	*	37	27	B7
E09	8 (	*	38	28	B8
E10	9)	*	39	29	B9
E11	0	*	30	30	B0
E12	- =	*	2D	3D	BD
E13	(left arrow)	*	1F	DF	D7
E14	HOME	*	18	D8	D0
E15	(right arrow)	*	1E	DE	D6

Pos.	Name	repeat	Normal	SHIFT	CTRL
G01	F1	*	81	91	A1
G02	F2	*	82	92	A2
G03	F3	*	83	93	A3
G04	F4	*	84	94	A4
G05	F5	*	85	95	A5
G06	F6	*	86	96	A6
G07	F7	*	87	97	A7
G08	F8	*	88	98	A8
G09	F9	*	89	99	A9
G10	F10	*	8A	9A	AA
G11	F11	*	8B	9B	AB
G12	F12	*	8C	9C	AC
G13	F13	*	8D	9D	AD
G14	F14	*	8E	9E	AE
G15	F15	*	8F	9F	AF

ALPHA NUMERIC GRAPHICS KEYBOARD
DM3005



REF. No.	PART No.	DESCRIPTION
1	MCMI314	CARD, P.S.U. ADAPTOR
2	M3195	CABLE ASSEMBLY, 7 CORE
3	MC606	CABLE ASSEMBLY, 10 CORE
4	G8090	PEN, GRAPHICS
5	H1061	SCREW, CHD, M3 X 8
6	H2022	NUT, M3
7	300/067	PLATE, SERIAL #
8	H1050	SCREW, PZ PHD S/T, 1/2"

Music Keyboard 8

Contents

Trouble shooting

The CMI has provision for one Master keyboard and an optional Slave keyboard which operates in parallel with the Master. The CMI mainframe has only one master keyboard input port, to which is connected the Master keyboard. The Slave keyboard, Alpha-numeric keyboard, and other attachments such as pedal controls, all connect to the Master keyboard. The latter contains an intelligent communications interface which monitors all attached devices and routes information from them through the single channel to the CMI. Other devices play into the MIDI sockets on the rear panel.

In addition to the piano type music keyboard, the Master keyboard provides two switch controls, as well as three rotary pot and two rotary wheel analogue controls. The switches (one momentary on, the other on/off) are fitted with lamp indicators whose purpose may be defined by the user by means of the CMI system software. One of the wheels has spring return, while the other wheel is unsprung. A 12 character LED alpha-numeric display and 16 switch keypad constitutes a simple user interface to the mainframe so that during a live performance, operations such as loading voices may be performed directly from the Master keyboard.

The Slave keyboard serves only as an extra music keyboard and contains none of the extra facilities of the Master keyboard.

Related Documents: The following drawings are either referred to directly in this manual or will be of use in servicing the CMI music keyboards -

Exploded diagrams	DMC004 Master Keyboard
	DMC004B Master Keyboard with cover
	DMC015 Keyboard switches subassembly
	DMC005 Slave Keyboard
Drawing	DMC004C Bottom panel screw positions
Schematic Diagrams	MC004-01 Master Keyboard wiring
	CMI10-01 Master controller to CMI10-04
	CMI11-01 Switch module
	CMI12-01 Display/keypad
	CMI14 Slave interface module

Operating Principles

Control over all keyboard functions is centralised upon the CMI-10 Keyboard Controller which is located within the Master Keyboard. Keyboard scanning, of both master and slave keyboards, is accomplished by analogue multiplexing of the voltages on all key switches. The key switch mechanism consists of two brass buss bars running the full length of the keyboard which are supplied with +5 and -5 volts, and a delicate spring contact on each key which is allowed to move between the two buss bars as the key is pressed. By measuring the time it takes the spring contact voltage to change from -5V to +5V, the velocity with which a key is pressed may be calculated.

The analogue multiplexing is performed by the CMI-11 switch modules, each of which has provision for 24 or 25 spring contacts. Each module provides one analogue output which is the state of the contact currently addressed by the select lines from the controller, and each keyboard contains three modules. Six analogue comparators (three for the master and three for the slave) on the master controller receive these analogue signals and determine the state of the currently addressed key.

The user keypad and off/on switches are scanned in the same way although the multiplexed states are read directly as a digital signal.

The wipers of the five rotary controls on the master keyboard and three plug-in pedal pots are similarly multiplexed and fed to a single analogue to digital converter on the master keyboard controller. A change detected in any analogue level read by the converter is reported to the CMI provided that change is greater than 4 digital levels (which translates to a change of 2 in the MIDI output frame). The tolerance of the pitchbend controller under MIDI is set by a 6-pole DIL switch at power-up, but is only used when the pitchbend wheel changes direction.

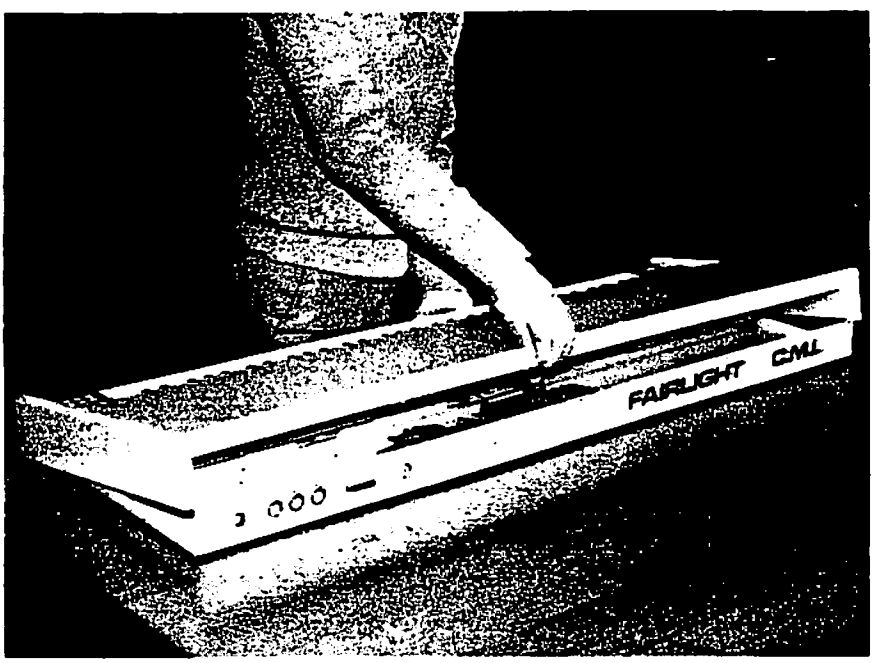
All information reflecting the state of the master and slave keyboards, and attached pedal controls are sent to the CMI via a serial communications channel in MIDI format. Characters received from the alpha-numeric keyboard are sent to the CMI through a serial communications channel at 9600 baud. User information received from the CMI through the same link is displayed on the LED display. The display modules accept ASCII characters directly from the keyboard controller.

Master Keyboard Disassembly

Removal of Wooden cover.
(Refer to drawings DMC-004C and DMC-004B)

1) Switch off CMI power and remove all cable connections to the keyboard.
2) Place keyboard upside-down on a soft surface.
3) Remove the six screws marked "A" and the five screws marked "B" on the drawing DMC-004C from the bottom panel of the keyboard. Do not remove any screws other than these from the bottom panel at this stage.
4) Return the keyboard right way up and remove the five back panel screws attaching the wooden cover to the panel, marked "C" on DMC-004C.
5) Lift the cover from the rear about 5 cm. then slide the cover forward while continuing to raise the rear as illustrated below.

With the cover off, the CMI-10 Keyboard Controller circuit module may now be observed, along with the wiring from the rear panel connections to the module. To remove the module, follow steps 6-9.



Removal of CMI-10 Keyboard Controller
(Refer to drawing DMC-004)

6) Remove all cable connections to the CMI-10 module.
7) Unscrew three nuts and bolts attaching the CMI-10's heatsink to the back panel.
8) Six plastic standoffs secure the module to the base of the keyboard. With a small screwdriver, press the catch of each standoff while gently prising the module up.
9) Lift the module off the standoffs.

Access to Keyboard Switch Mechanism
(Refer to drawings DMC-004C and DMC-015)

10) Slide the keyboard forward so the five screws marked "D" in drawing DMC004C may be accessed from underneath. Remove these screws to release the retaining strip which secures the keyboard assembly to the bottom panel.
11) The entire key assembly may now be swung up on its own hinges by lifting from underneath the keys. Support the assembly from behind on a piece of soft foam to avoid scratching the keys.

At this point the three CMI-11 switch modules may be viewed with the spring switch contacts gently stretched across the brass -5V buss bar and engaged in the plastic "keyhole grips" extending from underneath each key. Each grip has two keyholes: the spring contact should always be engaged with the lower one (closest to the underside of the key).

Removal of CMI-11 switch modules
(Refer to drawing DMC-015)

The following steps should be followed for each module to be removed:

12) Remove the 10-way cable plug from its socket.

> **CAUTION:** This cable should never be plugged or unplugged with the keyboard power applied or damage will result to the switch module circuitry.

13) Using tweezers or fine pliers, gently grip each spring switch contact and stretch it just enough to release it from its keyhole catch. Tuck it down underneath the lower brass buss supply bar (-5V).
14) Use a 6BA nut driver to remove the 9 nuts and star washers securing the switch module to the underside of the key assembly.
15) Unscrew the 3 screws which pass through the buss bar support blocks to the underside of the key assembly.
16) Lift the module off its supports.

Removal of Control Panel and Display/Keypad
(Refer to drawing DMC-004)

17) Slide the keyboard forward again as in step 10, and remove the four screws numbered 31 and 32 on the left in drawing DMC004 for the control panel, and/or the corresponding screws on the right for the display/keypad.
18) Lower the keyboard and remove the 20-way flat cable from the display/keypad or release from its cable clips the 20-way ribbon cable leading from the CMI-10 module to the control panel. This cable is attached to the control panel.
19) Lift the desired assembly out.

Master Keyboard Reassembly

Reassembly of the Master keyboard is essentially a matter of reversing the procedures of Section 2.1. Care should be exercised while replacing the CMI-11 switch modules not to damage the delicate spring switch contacts. Tighten the nine nuts and three buss bar support screws evenly to ensure the module is not warped or distorted in any way and that the buss bars are not bent.

Slave Keyboard Disassembly And Reassembly

To remove the wooden cover and the CMI-11 switch modules from a slave keyboard, follow the same procedures as specified for the master keyboard on pages 8.4 and 8.5 respectively.

Removal of CMI-14 Slave Interface
(Refer to drawing DMC-005)

1) With CMI power off, remove the flat cable connecting the master and slave keyboards, if not already done.

> **CAUTION:** Always turn off CMI power to the master keyboard before connecting or disconnecting the external cable between the master and slave. Omission to do this will cause damage to the switch modules in the slave keyboard.

2) The CMI-14 module is item 7 on DMC-005. Release the 25-way flat cable leading to the CMI-11 switch modules (item 8).
The same caution applies to this cable as to the external cable.
3) Remove the three screws marked 13 which secure the module to the back panel of the slave keyboard.
Reassembly of the slave keyboard is the reverse of the disassembly procedures.

Failure Of Master Keyboard To Power-Up

Successful power-on sequence of the Keyboard Controller is indicated by the control panel switch lights flashing on for approximately 1 second, off for another second, then on again. A "-SERIES III-" message is then written to the keypad display. If this does not occur and the CMI does not respond when the keyboard is played or the keypad operated, try to run the built-in diagnostic program (refer to pg. 8.18). If there is still no response, follow the procedure below.

1) Check that all power supplies (+10V, +20V and -20V) are present on the Music Keyboard cable from the CMI. Refer to drawing MC004-01. If not, check CMI fuses, the Cannon connector on the back panel of the keyboard, and the cable itself.
2) Remove the cover of the keyboard according to page 8.4.
3) Ensure all cables are firmly connected and that the correct power supplies are present on the six pin Utilux connector to the CMI-10 Keyboard Controller. If not, look for the faulty connection between the back panel sockets and the CMI-10, referring to drawing MC004-01.
4) Check that both DIL switches on the CMI-10 are set correctly. SW3 (4-way) should have switches 2 and 3 only closed, SW4 (6-way) should have switches 1 and 2 only off. Refer to pages 8.9 and 8.11.
5) Verify the voltages on each power supply regulator output on the CMI-10. Refer to page 8.14.
6) Check that the power-on restart circuitry holds the processor in reset for approximately 0.4 secs. Refer to section page 8.9
7) Check that the processor crystal is operating and that the processor ø2 output signal is present.
8) Check that the processor is not receiving spurious interrupts due to a faulty SW3. Refer page 8.9.
9) Establish whether the controller is running its program by examining the VMA, data and address lines, and checking peripherals which are accessed in the processor idle loop (refer to page 8.18). If it is, then the controller is powering up but a major I/O problem is preventing all normal indications of this. Otherwise, a fault in the processor itself, the address decoding system, the ROMs or RAM is causing the controller to crash. In both cases, carefully check each of the functions described in page 8.9 to isolate the fault.

Individual Key Failure (Master and Slave)

The failure of a single key to operate will usually be caused by a mechanical problem in the spring switch contact mechanism. Remove the cover of the keyboard according to page 8.4 and hinge the key assembly up as described on page 8.5.

Common causes of failure are damaged, loose or dirty spring contacts, or inadequate contact between the spring and the brass buss bars.

Failure Of Groups Of Keys (Master and Slave)

If all the 24 or 25 keys scanned by a particular switch module fail to operate then the fault lies either in that module (check the voltages on both buss bars) or in the path from it to the analogue key data multiplexer in the Keyboard Controller (incuding the cable). The source of such a fault may be isolated by swapping around the flat cable connectors to the switch modules.

Failure of certain keys belonging to each module is most likely to be caused by incorrect scanning addresses arriving at the switch module: either a cable fault or an I/O problem on the keyboard controller. In this case it is unlikely that the keypad or display will work either.

If no such module-related pattern to the faulty keys exists, then the problem is mechanical. Check that all spring contacts bend across the -5V buss bar by approximately 20 degrees from the horizontal when the keys are released and across the +5V bar by the same angle (in the opposite direction) when the keys are depressed. A tension spring in the back of each key returns it to the original position when it is released.

Slave Keyboard Malfunctions

Failure of groups of keys or individual keys on the slave keyboard can be diagnosed following the same guidelines as for the master keyboard. However two additional possible sources of faults exist: the cable from the master keyboard to the slave, and the CMI-14 slave interface. Since the slave scan address lines are the same as the master scan address lines, faults in the slave keyboard which corrupt those lines can cause the master to malfunction. Page 8.27 describes the use of the 4-pole DIL switch on the CMI-14 to disable individual switch module outputs when isolating slave keyboard faults. Ensure that all switches are open to enable the full keyboard velocity sensing prior to reassembling the slave.

CAUTION: Always turn off CMI power to the master keyboard before connecting or disconnecting the external cable between the master and slave. Omission to do this will cause damage to the switch modules in the slave keyboard.

The function of the CMI-10 Master Keyboard Controller card is to execute all keyboard facilities of the CMI and communicate the status of those facilities through a single cable (two serial channels) to the central processor. The facilities are -

Master keyboard scanning (with CMI-11 multiplexer).
Slave keyboard scanning (with CMI-14 slave interface and CMI-11 multiplexer).
Data link to CMI for the alpha-numeric keyboard.
Master keyboard keypad.
Keypad display of information from CMI.
Three rotary pots.
Two rotary wheel pots.
Two on/off switches.
Three pedal controls with switches.

This section describes the operation of the CMI-10 board.

MPU,DECODING,RAM AND RESTART.

(Refer to drawing CMI10-01)

Microprocessor Unit

The central driver of the Keyboard Controller is the 6802 microprocessor unit (MPU) at location E567 which is activated by a 4MHz crystal. At power-up the MPU reset line is held low for approximately 0.4 seconds at which time it is released to begin execution. It is important that this restart time is less than the CMI's Central Processor restart interval to ensure that no characters sent to the Keyboard Controller are lost. The MPU may also reset manually by depressing SW1 (nearer the heatsink). This switch is debounced through the pair of open-collector NAND gates D12.

While the restart line is held low, the MPU places FFFE (hex) on the address buss and its first operation is to fetch the restart vector from locations FFFE/F. Execution is then transferred to the initialization routines in ROM. Successful completion of this power up phase is indicated by the keyboard switch lamps switching on for about one second, off for another second, then on again. A "-SERIES III-" message is then written to the keypad display.

A 4-pole dual-in-line (DIL) switch, SW3, is used to select the source of Non-Maskable Interrupts to the MPU. This may be either from the manual switch SW2 or a clocked timing signal. The DIL switch functions as follows:

Switch	Effect if closed
1	Select BRCK signal from Baud rate gen. as timing reference.
2	Select ø2 from MPU as timing reference.
3	Select SW2 as NMI
4	Select timing reference as NMI

Clearly, switches 1 and 2 are mutually exclusive and must not be closed simulataneously, as are switches 3 and 4. Before feeding to switch 4, the high frequency reference selected by switches 1 or 2 is divided by 512, 1024, 2048 or 4096 by the binary counter C5. This division ratio is determined by the p.c.b. link next to C5 (normally 2048). The divided reference (signal SCND) is used as a control line signal to the PIAs, in addition to optioning as an NMI source.

With "MIDIKBD" and "KEYDIAG" ROMs, switches 2 and 3 only should be closed. This selects SW2 as NMI source, and this will activate the diagnostic routines if present. The NMI vector is at FFFC/D. Switch 1 of the DIL switch is nearest the edge of the PCB with the heatsink.

The 6802 MPU contains 128 bytes of internal RAM. This is permanently enabled by tying the Ram Enable signal (pin 36) high.

Address Decoding

Selection of all ROMS, external RAM and peripheral devices is performed by four LS139 1-of-4 decoders in ICs E12 and E34. Addresses are decoded when both the ø2 and VMA (Valid Memory Address) signals from the MPU are high.

The address map of the Keyboard Controller is as follows:

Address (Hex)	Function
0 - 7F	Internal RAM. 23 bytes only used, for software variable storage.
80 - 83	Active key input/AD conv. input PIA (K34)
90 - 93	Key address output PIA (F34)
A0 - A1	Alpha-numeric keyboard comms. ACIA (C67)
B0 - B1	CMI communications ACIA (D67)
C0	Software readable switch
4000 - 43FF	External RAM #1 (L67, N67)
5000 - 53FF	External RAM #2 (K67, M67, not normally installed)
9000 - 97FF	ROM #1 (J67, not normally installed)
A000 - A7FF	ROM #2 (HI67, not normally installed)
B000 - B7FF	ROM #3 (G67, "KEYDIAG")
F800 - FFFF	ROM #4 (F67, "MIDIKBD")

Software Readable Switch

The six-pole dual-in-line (DIL) switch SW4 selects a number of software functions. Bit 5 selects MIDI or Series I/IIX Fairlight mode. If bit 5 is off, the keyboard will behave like a Series I/IIX Fairlight keyboard and will communicate with the CMI only via RS-232 at 9600 baud. However if bit 5 is on, key depressions and control changes will be sent to the CMI in MIDI format at 31.25k baud. Keypad depressions are always sent via RS-232 regardless of bit 5. When connected to the Series III CMI, bit 5 should be on. When bit 6 is off, the two rotary wheels are ignored. At power-up, bits 1-4 are read, and this 4-bit number determines the sensitivity of the pitchbend control (wheel #1) under MIDI. When the pitchbend wheel changes direction, no more pitchbend data will be sent to the CMI until the ADC detects a change in digital levels of more than this sensitivity value. Normally sensitivity is set to 3 digital levels so switches 1 and 2 only should be off.

The switch is read through buffer N8 whose inputs are pulled high, unless grounded by a closed switch. Thus a binary '1' corresponds to an open switch.

External RAM

Provision is made on the CMI-10 p.c.b. for 2K of static RAM but normally only 1K is installed: 2114s L67 and N67. Each chip contains 1K x 4 bits storage. The upper nybble is stored in L67, and the lower nybble in N67.

ROMS and Peripherals

(Refer to Drawing CMI10-02)

ROMS

Provision is made on the CMI-10 printed circuit board for four 2516 ROMs. Normally only two of these are installed: "MIDIKBD" at F67, and "KEYDIAG" at G67. The first ROM contains the MIDI keyboard driver routines and is normally the only ROM in use, the second provides diagnostic routines.

Serial Communications ACIAs

RS-232 data from the Alpha-numeric keyboard is received, and MIDI sent, through the 6850 Asynchronous Communications Interface Adaptor (ACIA) at C67. MIDI data is driven as a current loop by LS03 D1,2 through 220R resistors R29 and R30. Series I/IIX keyboards without MIDI use the output side of C6,7 for RS-232 output to the Alpha-numeric keyboard instead of MIDI. RS-232 communication to and from the CMI utilises the 6850 ACIA at D6,7.

The baud rate for RS-232 channels is derived from the baud rate generator at B12 driven by a 1.8432MHz crystal and a PCB link at C12 normally selects 9600 baud operation (from pin 1 of B12). The baud rate generator also provides the BRCK signal, normally linked to 1200 baud at B45. MIDI communication with the CMI occurs at 31.25k baud through C67. However if bit 5 of SW4 is off, the keyboard will behave like an early keyboard and communicate to the CMI at 9600 baud through D67 only. MIDI should be selected when using a Series III CMI. The 31.25k baud signal is derived from ø2 divided by the counter at C5. Note that the timing signals are 16 times the baud rate.

Both ACIAs are normally linked via LK1 and LK2 to the common interrupt request (IRQ) buss signal, and generate interrupts whenever characters are sent or received.

Peripheral Interface Adapters (PIAs)

Two PIAs are used, each containing two 8-bit parallel I/O ports and four control outputs/IRQ input lines. The PIAs are configured during initialization and used as follows:

PIA F34	
I/O port A PA0 - PA1	Peripheral address outputs. Buffered through G23 to address to provide: CMI-11 switch module addresses CMI-12 keypad mutiplexer addresses LED display module data Data inputs to flip-flops (G4) which switch control button lamps. Analogue control input multiplexer addresses.
CA1	Scan Not Done (SCND) timing flag input
CA2	Strobe output to update lamp flip-flops
I/O port B PB0 - PB1 PB2 - PB7	LED display digit select lines LED display all-segments-on (CU) and module select (CS) signals.
CB1	Input flag from keypad multiplexer. Does not generate IRQs.
CB2	Strobe output to update a LED display ($\overline{DWS}$)
PIA K34	
I/O port A PA0 - PA5	Inputs from music key threshold comparators
PA6	Input from control switch multiplexer enabled by BKA7
PA7	Input from keypad multiplexer, also enabled by BKA7
CA1	Inverted timing reference input. Does not generate IRQs
CA2	Threshold select output
I/O port B PB0 - PB7	Data inputs from A/D converter (ADC)
CB1	$\overline{DR}$ (Data Ready) flag from ADC
CB2	B/$\overline{C}$ (Begin Conversion) strobe to ADC

Power Supplies

The Keyboard Controller receives +20V, -20V and +10V from the CMI through a 6-pin Utilux connector. Six on-board regulators (see drwg. CMI10-04) are used to generate three independant +5V supplies, in addition to +12V, -12V and -5V supplies. These power the Controller itself plus the keypad display, analogue controls and switches.

The supply designated "+5V" powers all circuitry on drawings CMI-10-01 and CMI-10-02 except the ROMs, which are powered separately from "+RV". The analogue multiplexers, A/D converter and RS-232 drivers on CMI-10-03 and CMI-10-04 receive power from "+XV" and where necessary, the -5V supply.

"+XV" and "-5V" leave the Controller board via SO1 to power the CMI-11 keyboard switch multiplexers, and the keypad display.
"+XV", "-5V", "+12", "-12" and "+20" leave via SO5 to power lights, switches and pot controls on the CMI-319 Analogue Control Module.

"+XV", "-5V", and "+20V" leave via SO2 if that socket is used (see the end of page 8.15)

Threshold Detection

MD1-3 and SD1-3 are the multiplexed signals representing the position of music keys addressed by the three master keyboard CMI-11 modules and the slave keyboard interface CMI-14, respectively. These signals are compared by the six MLM311s to a known threshold to determine when a key begins to be pressed, and when it is fully depressed.

The THLD signal from PIA K34 sets up one of two thresholds through the 741SC level shifter. If THLD is low, a -2.7V threshold is applied to the comparators. With THLD high, the threshold is +2.3V.

Initially, THLD is low. An unpressed key rests against the -5V buss bar so the corresponding comparator output will be high. When the key is first depressed and the spring contact leaves the -5V buss bar, the output of the module when that key is selected is pulled to just below zero volts by a 10k resistor to ground on the switch module and a 100k resistor to -5V on each comparator input. This causes the comparator to change state to a low. The change is read from the PIA whereupon THLD is switched high to select the +2.3V threshold, setting the comparator high again. It will return low when the key reaches the +5V buss bar at its full depression. The time taken between the two falling edges of the comparator output is noted by the MPU, and this mechanism forms the basis of the velocity sensitive keyboard.

The key continues to be compared to the +2.3V threshold until its release is detected.

Control Signal Multiplexors and A/D Convertor

User control signals enter the Keyboard Controller from several possible sources: two control panel switches, three control panel rotary pots, two control panel wheel pots, three pedal switches and three pedal pots. The switch controls are analogue multiplexed by H3 and read directly as KD6 when gated by a high level on BKA7.

The analogue controls (rotary, wheel and pedal pots) are multiplexed by I3, buffered by I4 (741SC), and fed to the AD570 A/D converter at J4. The low frequency signals used do not require a sample and hold. The converter is strobed to begin a conversion by the B/$\overline{C}$ signal from the CB2 output of PIA K34 and flags the end of conversion to CB1 of the same PIA.

The sensitivity of the pitchbend control may be set by DIL switch SW4. Refer to page 8.11 for further details.

RS-232 Interface

ICs A5 and A6 are the RS-232 drivers for the two ACIAs described on page 8.12.

Lamp driver

The control panel lamps are supplied with 20V and switched on when the MC75452 driver at J2 pulls the appropriate line to ground. The driver is activated by signals LP1 and LP2 latched from PIA F34.

Connections

The Keyboard Controller has five external connections as detailed below. Note that SO2 and SO5 are wired differently between Revs 6, and 5 and that SO5 does not exist on earlier revisions. The different cabling configurations are explained after the connection lists.

SO1 - 50-Way flat cable connector.

Pins 1-5	Master switch module 1 scan address
6	N/C
7	-5V to Master switch module 1
8	"+XV" 5V to module 1
9	Ground to module 1
10	MD1 module 1 multiplexed output
11-20	Master switch module 2 connections as for 1
21-30	Master switch module 3 connections as for 1
31-37	Scan address to keypad and data lines to LED display
38	All segments on, display module 0 (CU)
39	Module select, module 0 (CS)
40-41	CU and CS lines, display module 1
42-43	CU and CS lines, display module 2
44-45	LED display digit select
46	Digit write strobe
47	Keypad multiplexed output
48	$\overline{BKA3}$, selects keypad multiplexer 2
49	Ground to display/keypad
50	"+XV" +5V to display/keypad

SO2 - 10-way rainbow cable connector to pre-pitchwheel analogue controls.

Pin 1	Pulled low to light lamp 2
2	Pulled low to light lamp 1
3	State of switch 2
4	State of switch 1
5	Output of pot 3
6	+20V unregulated
7	-5V
8	+5V
9	Output of pot 2
10	Output of pot 1

SO3 - 26-Way rainbow cable connector

Pin 1	Pedal 1 pot wiper
2	Pedal 1 switch
3	Pedal 2 pot wiper
4	Pedal 2 switch
5	Pedal 3 pot wiper
6	Pedal 3 switch
7-11	Slave keyboard scan address
12	Slave keyboard ground
13-15	Slave switch module outputs
16	RTS flag to alpha-numeric keyboard
17	CTS flag to A/N keyboard
18	A/N keyboard ground
19	Data to A/N keyboard (pre-MIDI only)
20	Data from A/N keyboard
21	Ground
22	MIDI+ (pre-MIDI: Flag from CMI)
23	MIDI- (pre-MIDI: Flag to CMI)
24	Ground
25	Data from CMI
26	Data to CMI

SO4 - 6-Way Utilux Connector

Pin 1	+10V return
2	+10V
3	+20V
4	-20V
5	Ground
6	+/-20V return

SO5 - Rev 4 and below: not installed
Rev 5: 10-way flat cable connector
Rev 6: 20-Way flat cable connector

Pin 1	Ground
3	+12V
4	Output of wheel 2
6	Output of wheel 1 (pitchbend)
7	-12V
9	-5V
10	Unused

Pins 11-20	Only on Rev 6+
11	Pulled low to light lamp 2
12	Pulled low to light lamp 1
13	State of switch 2
14	State of switch 1
15	Output of pot 3
16	+20V unregulated
17	-5V
18	+5V
19	Output of pot 2
20	Output of pot 1

Cable configurations for analogue controls.

The above lists show that the 10-way connector at SO2 is wired in parallel with the upper 10 pins of the 20-way connector at SO5. This is to allow a single 20-way cable connection from SO5 to CMI-319 in Series III keyboards but still allow Rev 6 CMI-10s to be used as replacements in Series I/IIX keyboards, with a 10-way cable from SO2 to the analogue controls.

SO2 is not used if a Rev 6+ CMI-10 is installed in a Series III keyboard.

If a Rev. 5 CMI-10 is installed in a Series III keyboard, 10-way cables go to both SO2 and SO5, from the 20-way connector on the CMI-319 module.

Rev 4 and earlier CMI-10s cannot be used in Series III keyboards with pitch/modulation wheels.

Standard Diagnostic Routines

Built into the keyboard ROM "KEYDIAG", are a set of diagnostic routines designed to discover and isolate hardware faults that are not so serious that they prevent the 6802 processor from operating. These routines are capable of surviving faults which stop the normal driver software.

Although invisible during normal operation, they can usually be activated and the faults confirmed without opening the keyboard or requiring the use of a CMI except to provide keyboard power. The only tool needed is a jumper plug for the 9-pin D-socket labelled "TO CMI" on the back plate of the keyboard. This plug should have pin 6 connected to pin 9. If a fault is detected, you will have to open the keyboard case and locate the fault yourself, however the diagnostic routines can give you a good idea of where the failure is located.

By following this step-by-step guide, you can thoroughly test the music keyboard.

Starting the Diagnostics.

Disconnect the power to the keyboard. Depress "*" and "D" on the keypad and keep them pressed. Now connect the power to the keyboard. Both lamps should come on, go off, and come on again. The LED display should display "KDIAG3MIDIK9" or a similar message. This tells you the names of the two ROMs installed, in this case "KEYDIAG3" and "MIDIKBD9". Obviously, the ROMs may have been updated since this document was written. You should check to make sure that the latest ROMs are installed (if "DIAGNOSTICS:" appeared on the screen, then you have a pair of ROMs that are capable of crashing the CMI by sending spurious MIDI frames: definitely replace these). If all is well, go to RAM test pg. 8.20. Otherwise:

a) The lamps did not come on and the display did not display a message:

1) Check that all power supplies (+10V, +20V and -20V) are present on the Music Keyboard cable from the CMI. Refer to drawing MC004-01. If not, check CMI fuses, the Cannon connector on the back panel of the keyboard, and the cable itself.
2) Verify the voltages on each power supply regulator output on the CMI-10. Refer to page 8.14.
3) Check that the power-on reset lasts 0.4 seconds. Refer to page 8.9.
4) Check that the processor crystal is operating and that the processor $\phi 2$ clock signal is present.
5) Establish whether the program is running by examining the pins of the 6802 with a logic probe or CRO.
6) If the program is running, then the fault probably lies in the PIA's or cables connecting the circuit boards within the keyboard.
7) PIA test, step 2 pg. 8.20 details a method for activating the diagnostics which does not depend upon the PIA's.

b) One or both of the lamps did not come on but a message was displayed:

1) Check the lamps
2) Check the cable to the analogue controller board.
3) Check the driver at J2. Refer to drawing MC004-03.

c) Both lamps came on, turned off and came on again, but no message or a garbled message was displayed:

1) Check the cable to the keypad and display.
2) The PIA's may not be working properly (but the PIA at F3,4 is partially working since the lamps work).
3) The display may be faulty.
4) A program is running, but go to step 4.4.2 anyway.

d) The message "-SERIES III-" appears:

1) Check that "KEYDIAG" is present in socket G6,7.
2) The ROM "KEYDIAG" may be faulty.
3) The addressing to G6,7 may be faulty.
4) You may have released either "*" or "D", or depressed another key when power was applied.
5) The keypad or it's cable may be faulty.
6) The PIA at K3,4 may be faulty.
7) Go to step Alternative Startup Procedure below anyway.

e) Both lamps came on, turned off, but only the left hand lamp came on again. Nothing was displayed:

1) You have just witnessed a lamp blowing!
2) The internal RAM on the 6802 is faulty. You must replace the processor at E5,6,7

Alternative Startup Procedure.

Open the case of the keyboard *(refer to page 8.4).* Press SW2 at C8 to produce an NMI. This will always activate diagnostics if "KEYDIAG" is present. Failure of any sign of program operation to occur is due to faulty PIA's, timing problems or a faulty processor. Other problems are the same as for (1), except that the keypad and PIA at K3,4 are not used. If the program is running and the ACIA at D6,7 is working, then a short transmitted message should be detectable on the jumper plug attached to the "TO CMI" port about 18 seconds after the NMI was pressed.

RAM Test.

The display will change to "TESTING 2114". The 2114 ram chips at L6,7 & N6,7 are now being tested, and this will take about 11 seconds. "OK" should then flash onto the screen. If instead:

a) "R1E=xxxxxxxx" appears, where xxxxxxxx is a hexadecimal number, then the 2114 ram chips have failed that many times. This number is the total number of failures since the diagnostics were started. Since there are $400 ram locations, and each location is tested $100 times, the maximum number of errors detectable in each execution of the ram test routine is $40000.

PIA Test.

The display will change to "TESTING PIA". In about half a second, "OK" should appear. If instead:

a) The left hand lamp turns off or "PIA1ER" or "PIA2ER" appears - a problem was detected in one of the PIA's. PIA #1 is at F3,4 (look for overloaded outputs) and PIA #2 is at K3,4 (it has to be very dead to fail this test)

1) PIA #1 is used for device selection and display. PIA #2 is used for keyboard & keypad addressing, and A/D conversions. If you have come this far and received messages on the LED display, PIA #1 is working, so suspect wiring or contacts instead.
2) It is possible for malfunctioning PIA's to pass this test.

ACIA Test.

The display should change to "TESTING ACIA". In about half a second "OK" should appear. If instead:

a) "AC2RER" appears, then a garbled message was received by the ACIA at D6,7. The program will continue after a few seconds.
b) "AC2TER" appears, then no message or an incorrect message was received at D6,7.
c) nothing appears, then the program has probably hung due to the IRQ line to the 6802 being pulled permanently low.

The ACIA at C6,7 is not tested by this program as it is used to generate MIDI, which operates at a different baud rate from all other communication channels. This should be tested by playing some notes using the normal keyboard driver.

Check the jumper plug, the 9-pin D sockets on the keyboard back panel, and the wires to these sockets before suspecting the ACIA device.

Main Event Loop.

The screen should now show "GO" and the program will be in an endless loop. If any change is detected in the state of any key, switch, pot or pedal then this change will be displayed on the screen as follows:

"MK zzz DN"	Key zzz is fully depressed
"MK zzz UP"	Key zzz has been released
"MK zzz MD"	Key zzz is half depressed
"SW yy DN"	Switch #yy on the analogue controller board is fully depressed
"SW yy UP"	Switch #yy on the analogue controller board has been released
"PS yy DN"	Pedal switch #yy has been pressed
"PS yy UP"	Pedal switch #yy has been released
"DSW xx"	The value of the DIP switch is $xx
"KP w DN"	"w" has been pressed on the keypad
"KP w UP"	"w" has been released on the keypad
"POT yy xx"	The value of rotary pot #yy is now $xx
"PDL yy xx"	The value of pedal #yy is now $xx
"PB xx"	The value of the pitch bend is now $xx
"W2 xx"	The value of wheel 2 is now $xx

where:

w	is a letter available on the keypad,ie 0,1,2,3,4,5,6,7,8,9,A,B,C,D,* or #
xx	is a two digit hexadecimal number
yy	is a number, ie 02, representing the number of a particular device if more than one may be connected at once.
zzz	is the name of a musical note. The first digit represents the number of the octave. The last two digits are the name of the note, ie C# or G. The lowest note on the keyboard is 1F and the highest is 7F. All notes from C to the next highest B have the same octave number.

If the ADC fails to read a value, the message "ADC E=xxxx" will be displayed briefly, every time the program goes through this program loop. xxxx is a hexadecimal value representing the total number of ADC failures detected so far. ADC errors are usually due to problems with the -20V power supply from the CMI, or the -5V and -20V regulators on CMI-10. Fuse #7 in the base of the CMI is attached to the -20V keyboard supply.

Analogue Controller Testing

At this stage you should check that the pots, pedals, wheel and pitchbend controls work properly. They should go from $00 to $FF with $80 being the central value. This display has a window of 2, so you should strive for values accurate to +/-1. Check also that turning any of the controls beyond $FF or $00 does not result in phantom turning of some other control. This problem is a result of overloading the analogue multiplexers, and can be partially cured by adjusting the trim pots on the underside of the analogue controller box at the left hand end of the keyboard.

If the keyboard case is open, and you suspect that the DIL software switch may not be working, then toggle one of the switches in this DIL switch. The default value of the DIL switches is $03, which corresponds to switches 1 and 2 only being on. Remember that the program reports only what the switches have been changed to, not what they were originally. Alternatively, press "*" then "2". The display will report the current value of the DIL switch, and then continue. This command is discussed in more detail below.

Commands

One key that has special significance is "*". This is the command key. Pressing it results in the message "COMND:" being displayed. Pressing another key will allow you to select from the following alternatives:

"A" One time test of the 2114 rams

"B" One time test of the PIA's. Not all problems with the PIA's can be detected.

"C" One time test of the CMI ACIA.

"2" The current setting of the 6-pole DIL switch at M8 will be displayed on the screen as a hexadecimal number. The default setting is $03. The value of switches 1 to 6 correspond to $01, $02, $04, $08, $10, and $20 respectively when they are in the off position.

"3" Perform tests A, B, C and calculate the ROM checksum continuously. If the message "ROM = xx" is displayed, then there is a fault in the ROM "KEYDIAG" which is probably due to an intermittent addressing problem. Such a problem would also cause the ROM checksum to be unstable, and the software will crash with similar regularity. Test 3 is a good soak test for the keyboard. While each test is in progress, a message "TESTING (device)" will be displayed. Successful completion of each test is indicated by "OK" flashing onto the screen. Failure will result in an error message, and a jump into the error handler. If more than 15 seconds goes by without the display changing, then the program has hung.

"1" Continue after a pause of a few seconds when an error is detected. After this command has been executed, the error handler will continue the diagnostics after a pause of a few seconds. This command cannot be undone. It should not be executed before performing an unattended soak test as any error messages will be overwritten.

"#" A check of all keys will be performed by asking you to press every key on the keyboard from lowest to highest. "SCALE" will be displayed on the screen to indicate that the command has been recognised. The display will change to "PRESS 1F". When you have pressed and released this key (the lowest key on the keyboard) it will ask you for the next higher key. After all keys have been pressed, the display will return to "GO". If any key is missed, the display will continue to display that key until it is pressed. In order to avoid multiple key depressions, it is suggested that this test be performed using one finger.

"0" Jump into MIDI driver code. The keyboard will behave just like a regular keyboard and can be used to output MIDI to the CMI. If you wish to return to diagnostics, you should go to step (1).

"D" Return to the diagnostic loop that reports changes in the state of keys etc. This command is used to undo the effect of pressing "*".

Stopping a Continuous Test

To get out of a continuous test, simply turn the power off and on, or press NMI (SW2) or press RESET (SW1). NMI will always get you into diagnostics again, while "*" & "D" need to be pressed if the keyboard is RESET or powered up.

The Error Handler

The error handler is a subroutine designed to make sure that you have received each error message. It waits for all keys on the keypad to be released. Next, it waits for the keys "1", "4", "7" and "*" to be pressed simultaneously. After a wait of a couple of seconds, the program returns to where it left off. If you wish the program to continue after a pause of a couple of seconds without waiting for these key depressions then issue the command "*" & "1" before executing any continuous test.

Soak Tests

A thorough test of the keyboard should also include the following:

a) Turn the keyboard on and leave it for 15 minutes or more. If the message "GO" is still on the screen then there are probably no problems with dirty contacts. If a message such as "MK 6G# UP" is displayed, then this key may not have reliable contacts.

b) Execute the command "*" & "3" and leave it for several hours (days?). This should detect intermittent problems with RAM, ROM, PIA's and CMI ACIA.

c) Finally, connect it to a CMI. RESET the keyboard by turning the power off and on while none of the keypad keys are depressed. The keyboard should now be producing MIDI code whenever keys are hit. Verify this by playing a few notes through the CMI. This is extremely important as the MIDI ACIA cannot be tested by the diagnostic routines because it operates at a different speed from the two serial inputs.

Three Keyboard Switch Modules are installed in each master and slave keyboard used with a CMI. Each module provides a single signal out which represents the state (pressed, released, or in flight) of one of the 24 or 25 keys addressed by the multiplexer inputs. This section describes the operation of the CMI-11.

Keyboard Switch Module Operation
(Refer to drawing CMI-11-01)

Five key address bits are provided provided by the Keyboard Controller CMI-10 as inputs to the CMI-11. The lower three of these are bussed across three 4051 analogue multiplexers (ICs 2-4) so that each 4051 selects one of eight spring key contacts as its analogue input. Normally, a key rests against a -5V buss bar, but when fully depressed, it contacts a +5V buss bar. In between, it contacts neither.

The outputs of ICs 2-4 are fed to another multiplexer, IC1, whose select inputs are the upper two bits of the key address. Thus the output of IC1 may be any of the 24 key contacts accessed by ICs 2-4. It may alternatively be the 25th key contact which is fed directly to IC1 as a fourth analogue input.

Each CMI keyboard has a total of 49 keys so the 25th key is only used on the extreme right hand switch module. Provision is made on the switch module p.c.b. for a 10k resistor (R1) pulling to ground. This is to ensure that if the 25th key is not installed, it appears to the multiplexer as a key which is never pressed. However, the resistor must be removed if the 25th key is installed or the velocity sensing mechanism will not work on that key.

The output of IC1 is fed directly to the Keyboard Controller in a master keyboard or to the Slave Interface in a slave keyboard. Its unused inputs are grounded.

External Connections

SO1 - 10-Way flat cable

Pins 1-5	Key scan address inputs
6	N/C
7	-5V supply
8	+5V supply
9	Ground
10	Multiplexed analogue output

The Display and Keypad Module provides a simple user interface with the CMI from the master music keyboard. A 16-switch keypad is scanned by the Keyboard Controller for commands to be sent to the CMI and a 12 digit LED display receives simple messages from the CMI to the user. This section describes the operation of the CMI-12.

Display and Keypad Operation
(Refer to drawing CMI-12-01)

LED Display

The DL-1416 LED display modules, containing four digits each, accept 7 bit ASCII codes from the data lines to display the desired character. The key scan addresses are used as data inputs. Data is latched into the modules whose chip select line ($\overline{CS}$) is low on the falling edge of $\overline{DWS}$. The DA lines select which digit within the selected module(s) is written to. The $\overline{CU}$ line is a test enable line which causes every segment in each digit to light up.

Keypad

The keypad is simply an array of 16 momentary switches which connect to the common (+5V) line when pressed. Two 4051 1-of-8 analogue multiplexers scan the keypad. Their select and inhibit inputs are taken from the key scan address lines. Only enabling one multiplexer at a time allows the outputs to be wired together on the same KPAD signal.

External Connections

SO1 - 20-Way Ribbon cable connector

Pins 1-2	Digit select
3,5,7	Display module select
4,6,8	Display module test (all segments on)
10	Digit write strobe
9,18, 11-16	Key scan address and data to display modules
17	Keypad multiplexed output
19	Ground
20	"+XV" +5V supply

The Slave Keyboard Interface provides regulated power supplies to the CMI-11 switch modules in a slave keyboard and buffers the analogue outputs of the switch modules before feeding them to the master keyboard controller. This section describes the operation of the CMI-14.

Operation
(Refer to drawing CMI-14)

Scanning and Buffering

The five slave key scan address lines from the master keyboard controller are fed straight through to the CMI-11 switch modules. The output from each module is buffered by a 741SC in a non-inverting configuration and fed to the master controller. A 4-pole dual-in-line (DIL) switch allows the input of each buffer to be pulled to nearly -5V for testing purposes. In the event of a switch module being unplugged, closing the switch corresponding to that module simulates all keys released. Two or more floating buffer inputs result in the keyboard controller going into overflow due to sensing too many keys pressed. All switches should normally be open, otherwise the velocity sensing system will not work.

Power Supplies

The CMI-14 is supplied with +20V and -20V from the CMI via the master keyboard. A 4V7 zener is used on each supply side to provide +12V and -12V to the 741 buffers, and 7805 and 7905 regulators send +5V and -5V respectively to the switch multiplexers.

External Connections

SO1 - 30-Way flat cable connector

Pins 1-5	Slave switch module 1 scan address
6	N/C
7	-5V to Slave switch module 1
8	"+XV" 5V to module 1
9	Ground to module 1
10	MD1 module 1 multiplexed output
11-20	Slave switch module 2 connections as for 1
21-30	Slave switch module 3 connections as for 1

SO2 - 25-Way D series external connector

Pins 1-2	Ground
3-7	Slave keyboard scan addresses
8	Ground
9-11	Slave multiplexer outputs
12-21	N/C
22-23	-20V supply
24-25	+20V supply

The analogue controls permit the musician to convey information to the CMI about tonal quality in a continuously variable manner (actually a large number of digitized values). This controller board consists of two switches, three rotary pots, two rotary wheels and two lamps. This section describes the operation of the CMI-319.

Operation
(Refer to drawing CMI-319)

Rotary Pots
There are three rotary pots on this controller board, and they are labelled control 1, 2 and 3. The pots are connected between the +5V and -5V supplies. By turning the pot, the potential of the wiper is made to vary between +5V and -5V. No adjustment is possible or necessary.

Switches
There are two switches, labelled switch 1 and 2. When a switch is depressed, it connects its line (SW1 or SW2) to +5V. The line is pulled to ground when the switch is open.

Lamps
There is a lamp inside both of the switches. The lamp is independent of the switch position. Each lamp is connected between +20V and the lamps signals, LAMP1 or LAMP2. These signals are controlled by a MC75452 at J2, which is in turn controlled by LP1 and LP2. Refer to circuit diagrams CMI-10-03 and CMI-10-04.

Wheel controls

Wheel 1 is a pitchbend controller. It is used to determine pitchbend frames for MIDI. This control allows the musician to alter the pitch of a note or chord by up to +/- 128 semitones. It is fitted with a spring return, which always returns the control to the central position when not being held. The central position corresponds to no pitch alteration.

Wheel 2 is a modulation wheel. It is not fitted with a spring return, so will remain in the same position when released. Apart from this difference, it is mechanically and electrically the same as the pitchbend control, however the software demands much smaller tolerances on the pitchbend.

Each wheel is connected to the shaft of a rotary pot. This pot is connected between +12V and -12V. Due to mechanical restrictions, the wiper is only able to traverse a smaller range of voltages. An LF353 dual op-amp is used to adjust the output voltage to the range +5V to -5V with 0V at the central position. This is achieved by using two 500K 10-turn trim-pots for each op-amp. The first trim-pot is connected between +5V and -5V, and the wiper is connected to the + input of the op-amp to provide offset adustment. When this voltage is the same as that of the - input, the op-amp output is 0V. The other trim-pot is used as a variable resistor between the output of the op-amp and the - input to provide gain adjustment. This allows control of the range of the output voltage. Two back-to-back diodes are connected, to the output of the op-amp to provide a 0.6V dead-zone around the central position. Finally, two 1N914 diodes are connected between the output signals MOD and PITCH and +5V to prevent circuit damage if the trim-pots are incorrectly adjusted.

Adjusting the wheels

Incorrect adjustment of the pitchbend controller will result in the following problems:

a) The CMI will be out of tune. This is due to the pitchbend assuming a non-zero voltage when not being held.

b) The pitchbend will be unable to bend the sound up or down by one whole octave. This is due to insufficient range on the pitchbend.

c) Turning the pitchbend appear to the CMI as being accompanied by the turning of the other controls. This is due to the range of the pitchbend output voltage, PITCH, exceeding the +5V and -5V supplies, causing the 4051 multiplexer at I3 to become overloaded.

Similar problems will occur with wheel 2, except that this wheel is often unused.

To adjust the wheels, follow these steps:

1) Remove the case of the keyboard. Refer to page 8.4.

2) Unscrew the control panel. Refer to page 8.6.

3) Activate diagnostics, and proceed to the main event loop. Refer to 8.18

4) Set the range of each control to a small value. The pitchbend range trim-pot is the second from the top, wheel 2 at the top. Turn it anti-clockwise to reduce the range.

5) Set the central value to $80. The pitchbend centre adjust trim-pot is the 4th from the top, wheel 2 3rd from the top. Turn it clockwise to reduce the values.

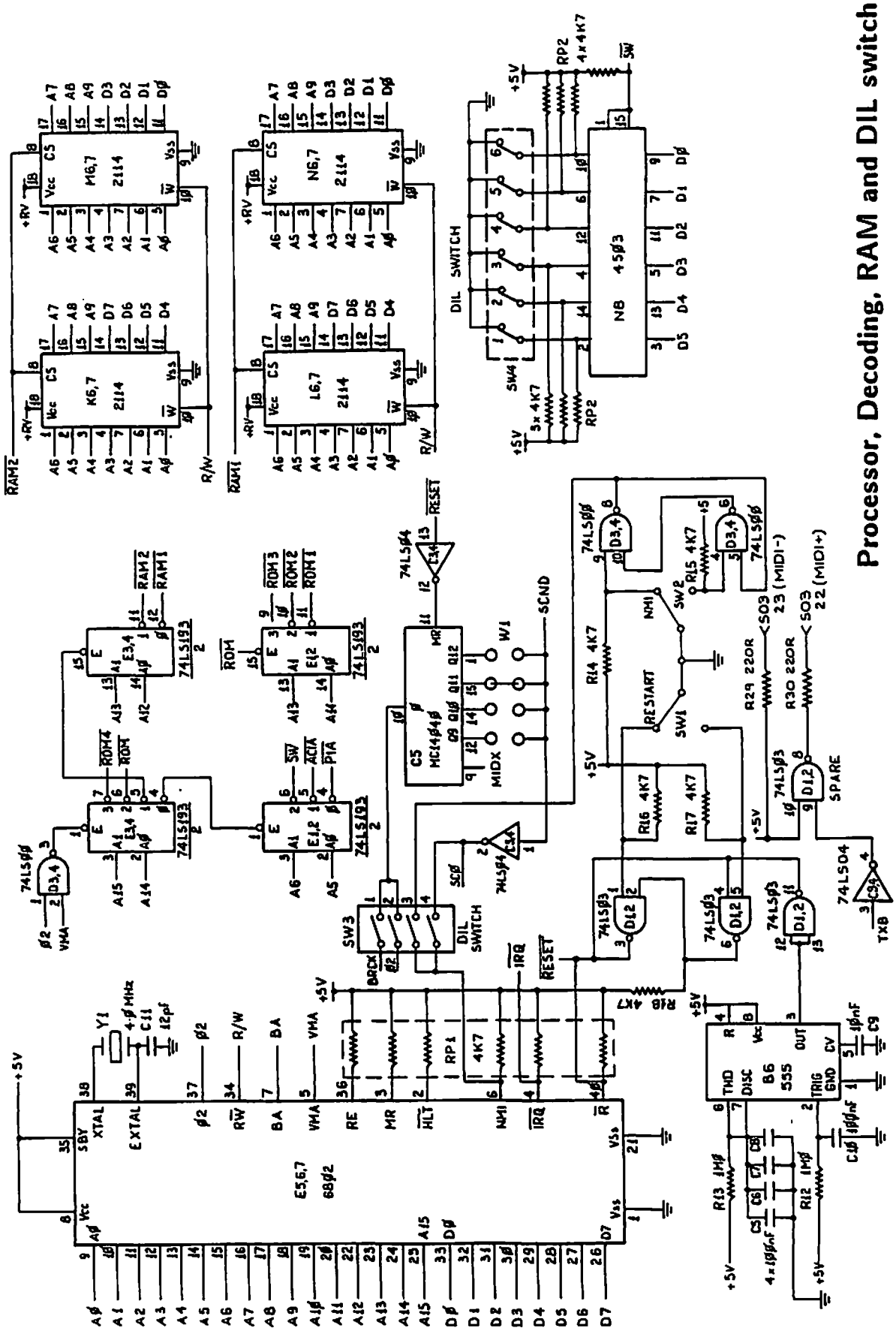
6) Turn the range-adjust trim-pot clockwise to increase the range. Adjust the centre-adjust trim-pot to make the range symmetrical about $80. Repeat step (6) until the wheel just reaches the endpoints of $0 and $FF. Overshoot will result in phantom turning of other controls due to the 4051 at I3 becoming overloaded. Finally, make sure that the centre is still $80.

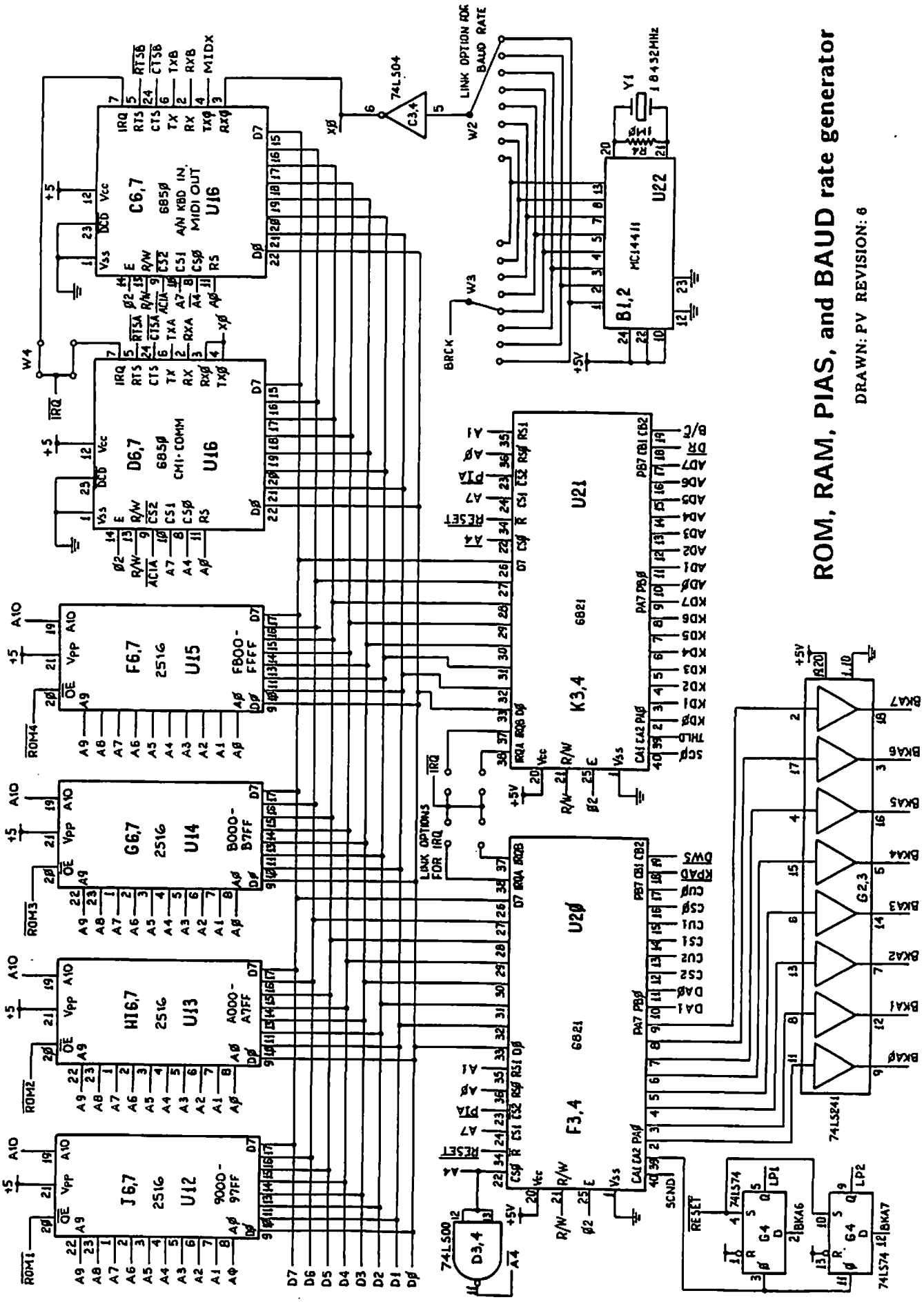
7) Alternatively, you may use a digital voltage meter. Pin 2 of the 4051 at I3 is the output of the pitchbend. Pin 4 is the output of wheel 2. They should go from -5.00V to +5.00V with the voltage when the wheel is in the central position being 0.00V. In practice, all three parameters can rarely be satisfied, however you should try to achieve maximum possible accuracy.

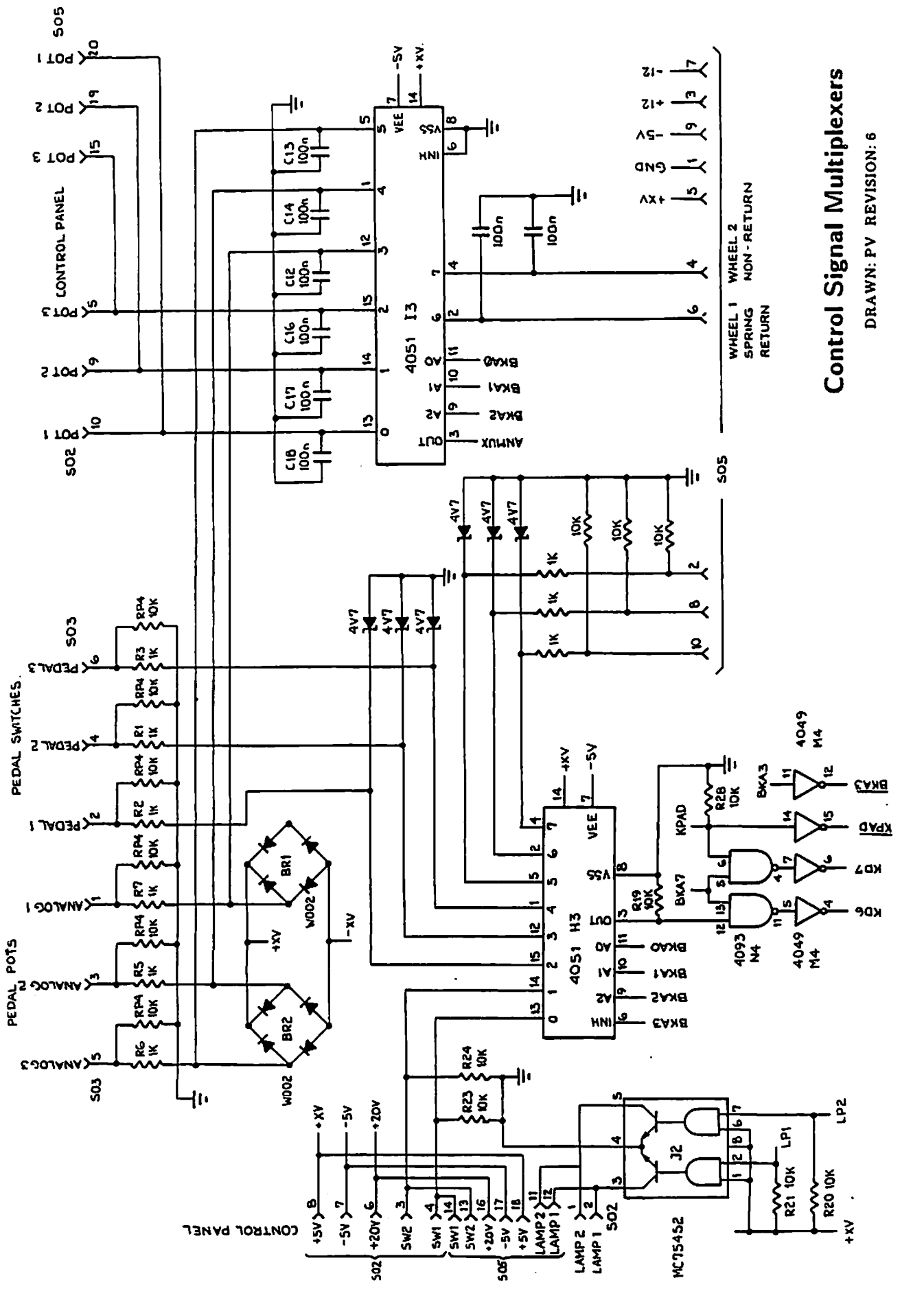
External Connections

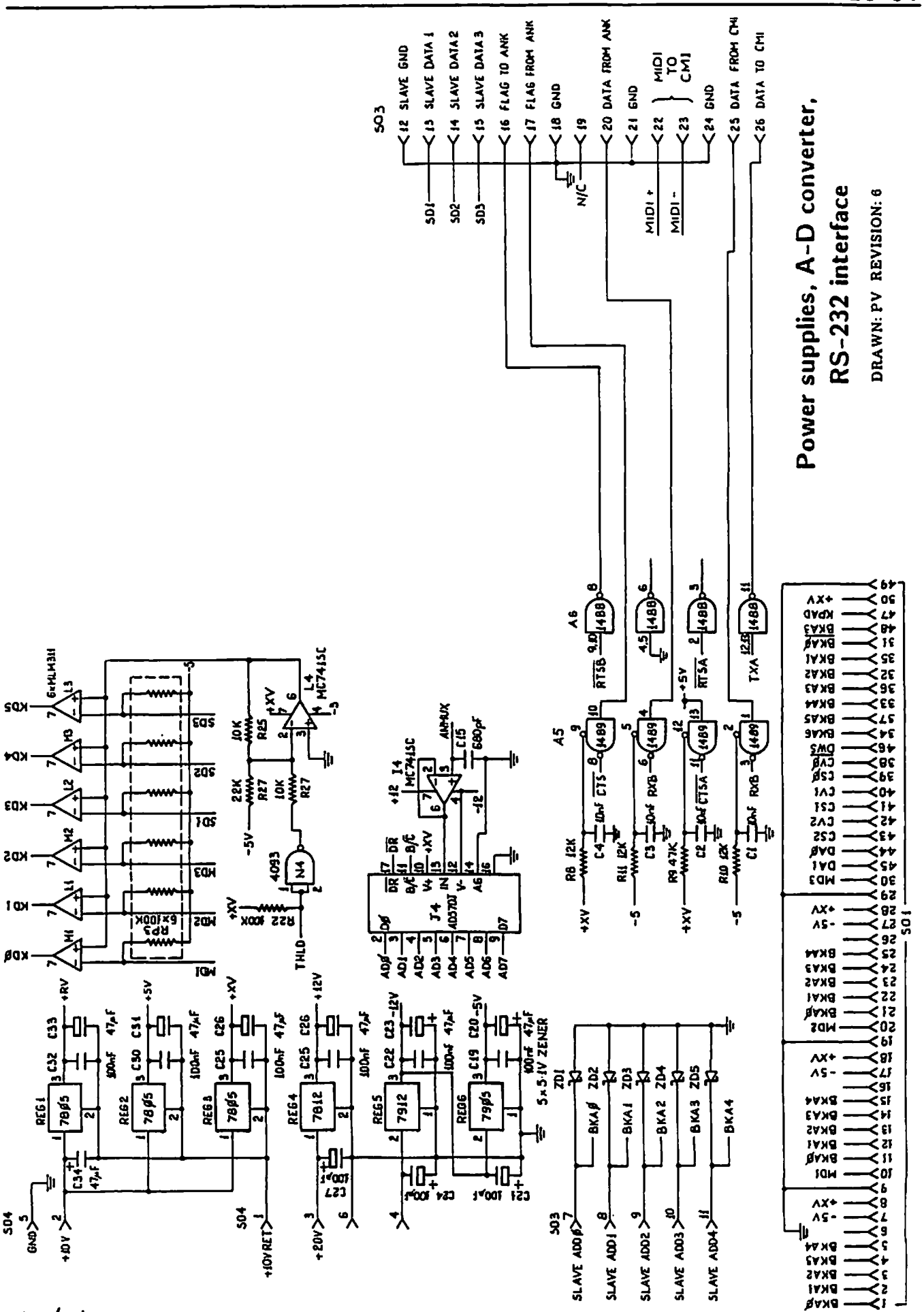
SO1 - 20-Way flat cable connector

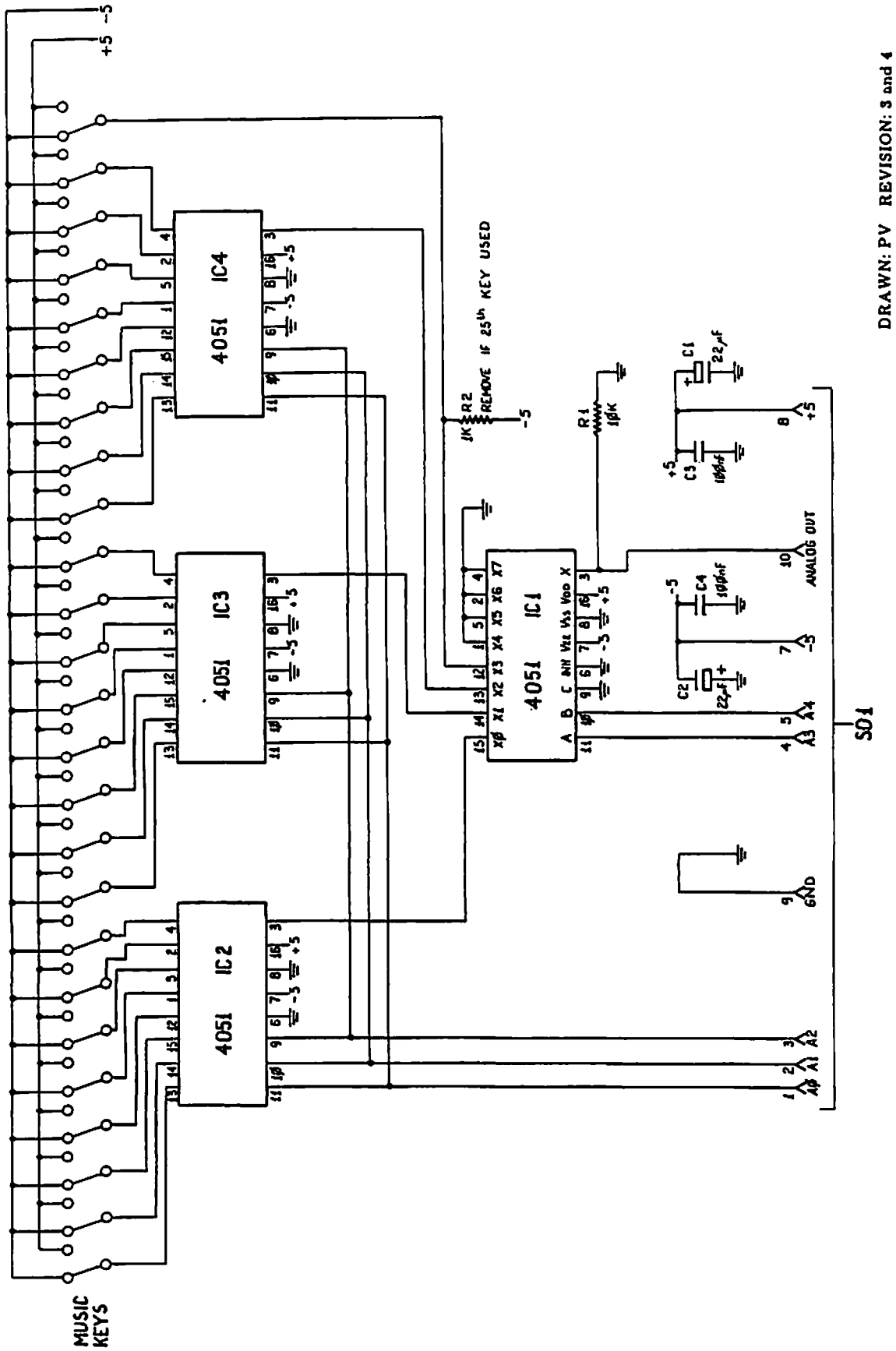
Pin 1	Ground
3	+12V
4	Output of wheel 2
6	Output of pitchbend
7	-12V
9	-5V
11	Pulled low to light lamp 2
12	Pulled low to light lamp 1
13	State of switch 2
14	State of switch 1
15	Output of pot 3
16	+20V unregulated
17	-5V
18	+5V
19	Output of pot 2
20	Output of pot 1

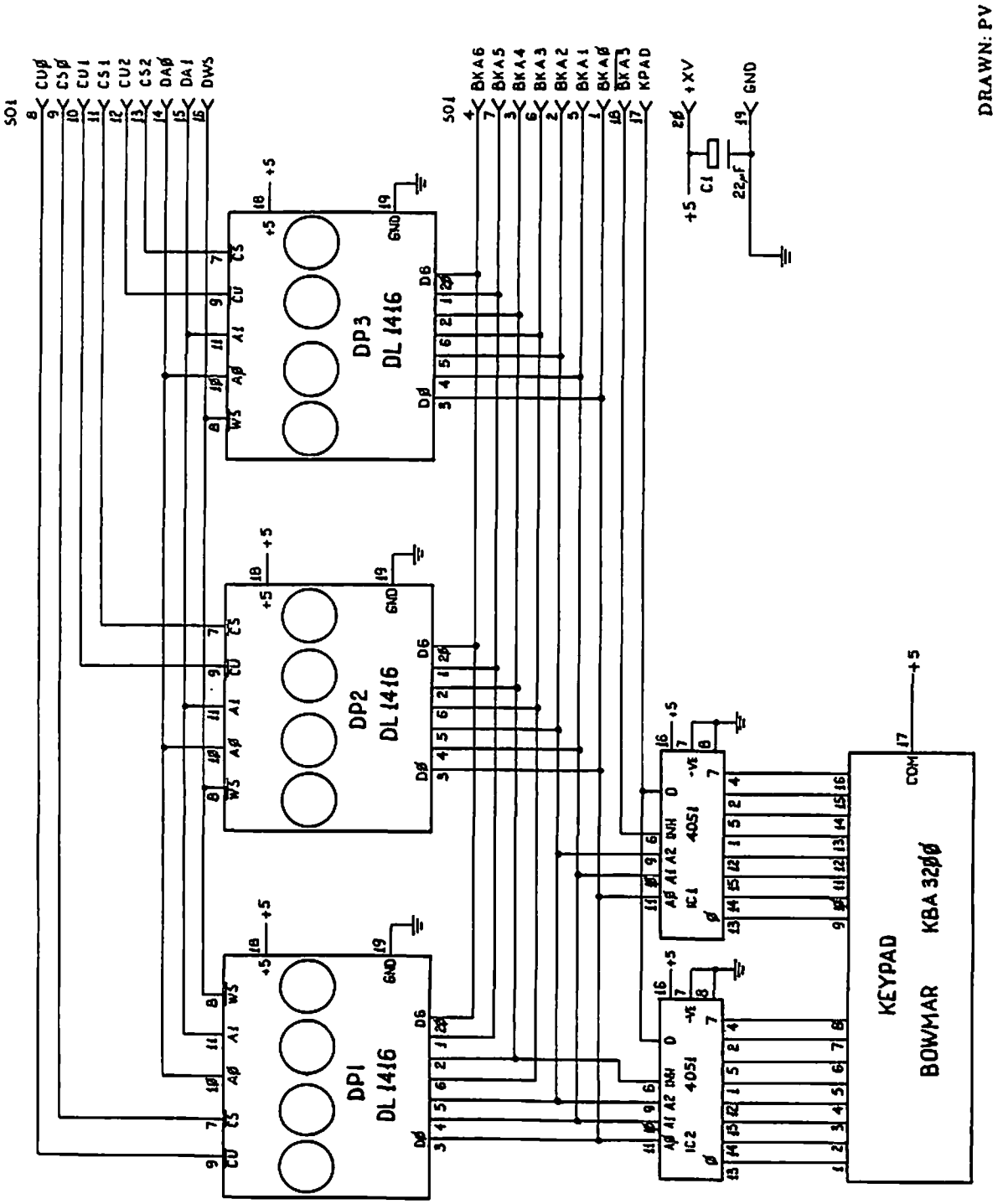

Processor, Decoding, RAM and DIL switch
DRAWN: PV REVISION: 4

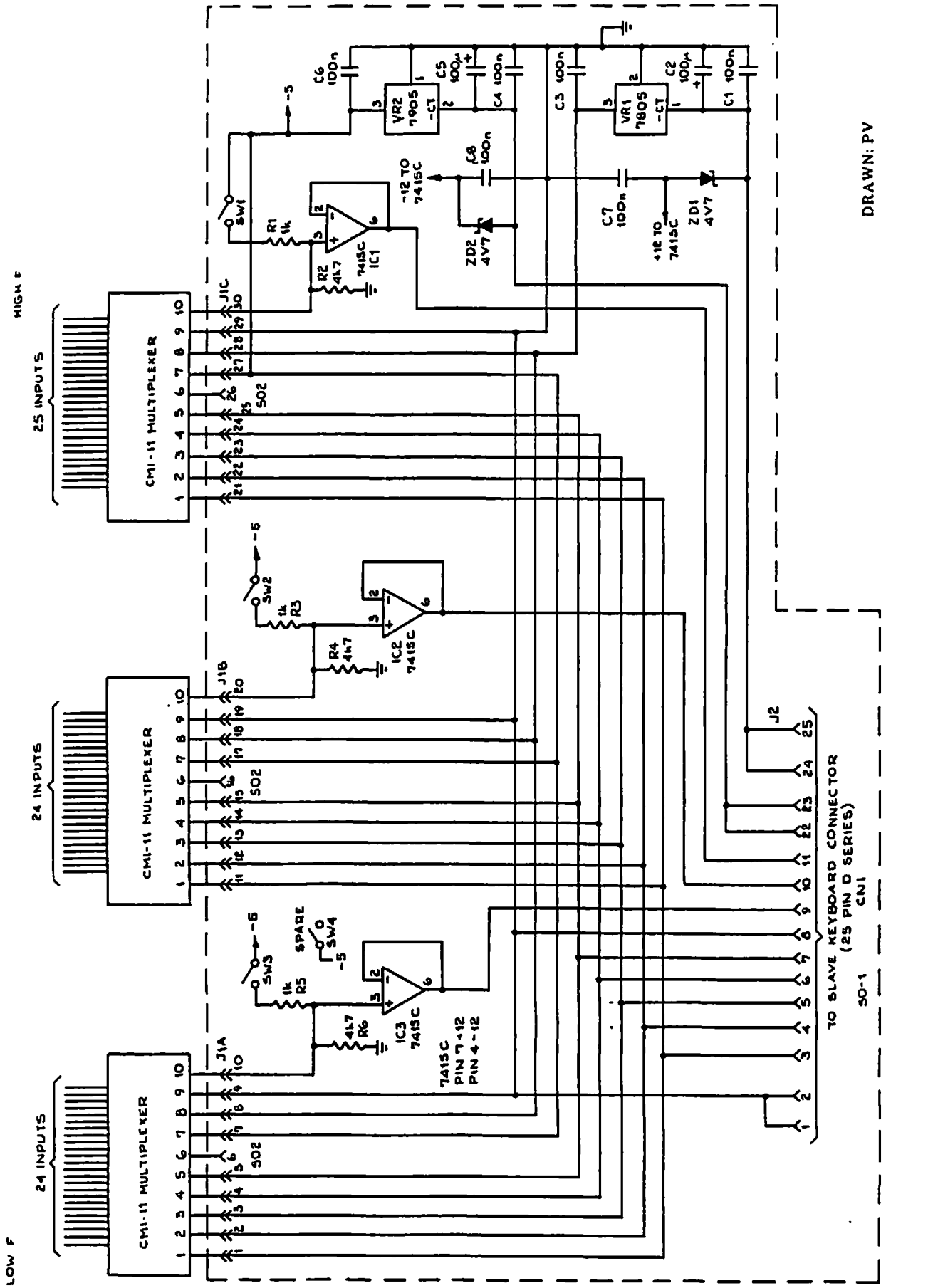

ROM, RAM, PIAS, and BAUD rate generator
DRAWN: PV REVISION: 6
U16
C6,7
D6,7
U15
F6,7
U14
G6,7
U13
H6,7
U12
J6,7
U21
K3,4
U20
F3,4
U22
B1,2
Y1
1.8432MHz
LINK OPTION FOR BAUD RATE
LINK OPTIONS FOR IRQ
741S04
741S00
741S241
741S74
MC14411
6821
2516
6850
A/N KBD IN, MIDI OUT
CMI-COMM
MIDI
RESET
BRCK
IRQ
ROM1
ROM2
ROM3
ROM4


Control Signal Multiplexers
DRAWN: PV REVISION: 6
WHEEL 1 SPRING RETURN
WHEEL 2 NON-RETURN
CONTROL PANEL
PEDAL SWITCHES
PEDAL POTS
4051 I3
4051 H3
MC75452
4093 N4
4049 M4
SO5
SO2
SO3
W002
BR1
BR2
LP1
LP2
J2

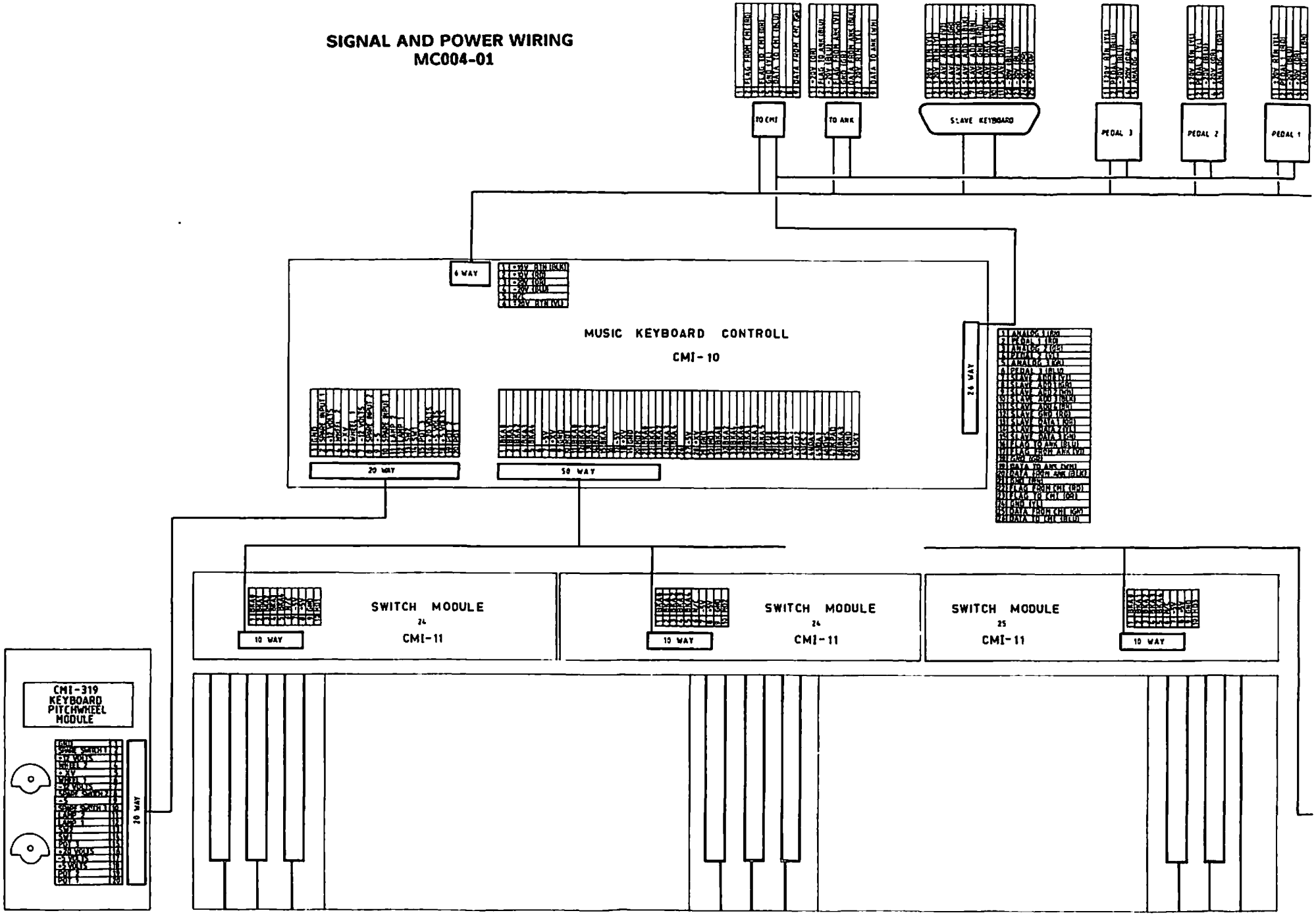

Power supplies, A-D converter,
RS-232 interface
DRAWN: PV REVISION: B
SO3
SLAVE GND
SLAVE DATA 1
SLAVE DATA 2
SLAVE DATA 3
FLAG TO ANK
FLAG FROM ANK
GND
DATA FROM ANK
GND
MIDI TO CMI
GND
DATA FROM CMI
DATA TO CMI
MIDI +
MIDI -
N/C
SD1
SD2
SD3
A6
A5
RTSB
CTS
RXB
RTSA
CTSA
TXA
R8 12K
R11 12K
R9 47K
R10 12K
C4
C3
C2
C1
J4
AD7701
MC7415C
AMUX
C15
680pF
R25 10K
R27 22K
R27 10K
R22 100K
4093
THLD
RP5 6x100K
REG1 78Ø5
REG2 78Ø5
REG3 78Ø5
REG4 7812
REG5 7912
REG6 7905
5x5.1V ZENER
ZD1
SLAVE ADDØ
SLAVE ADD1
SLAVE ADD2
SLAVE ADD3
SLAVE ADD4
SO1
SO4
+10V
+10V RET
+20V
GND


MUSIC KEYS
4051 IC2
4051 IC3
4051 IC4
4051 IC1
+5
-5
1K
R2
REMOVE IF 25th KEY USED
R1
10K
C1
22µf
C2
22µf
C3
100nf
C4
100nf
ANALOG OUT
GND
SO1
DRAWN: PV REVISION: 3 and 4

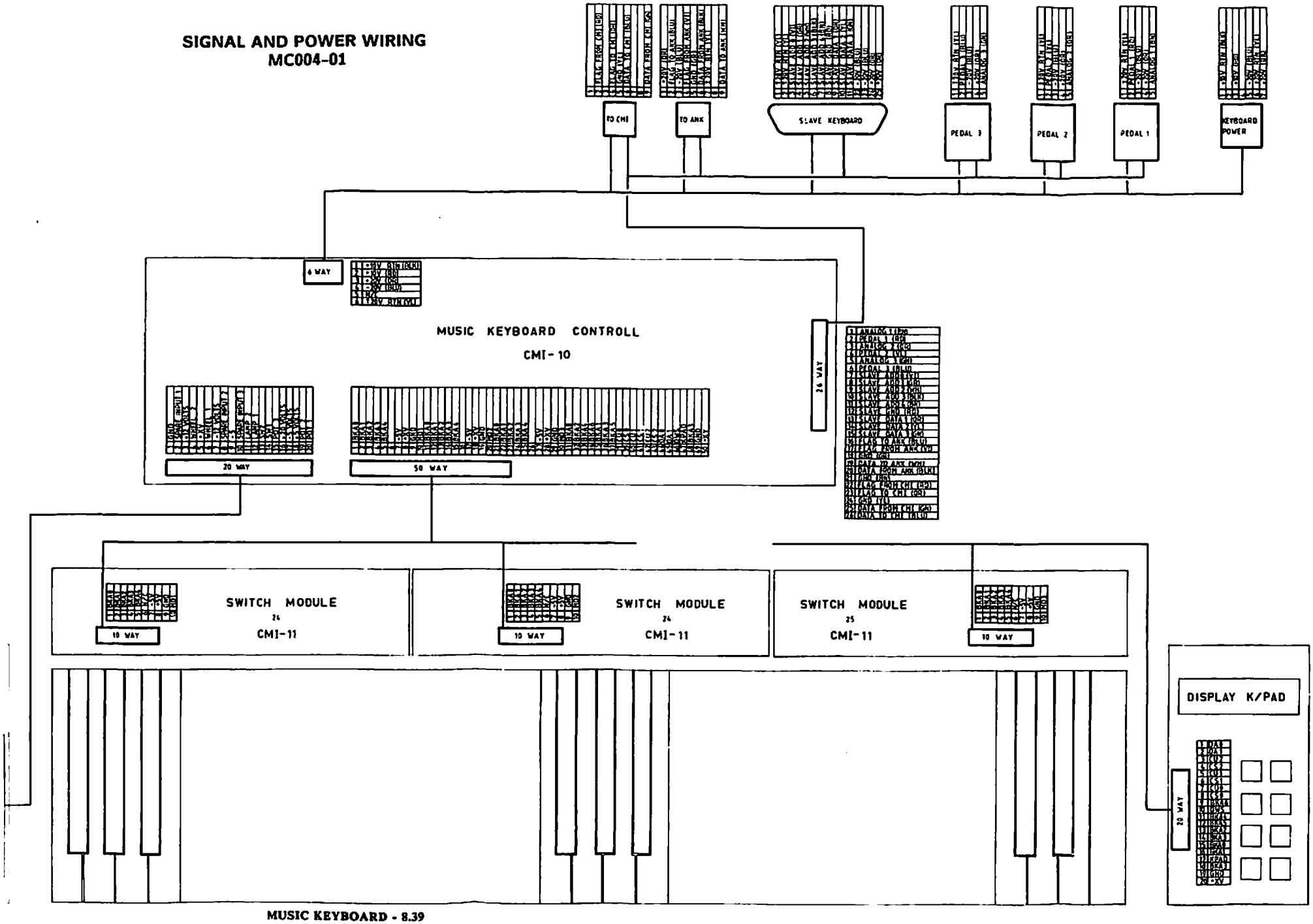

DRAWN: PV
DP3
DL 1416
DP2
DL 1416
DP1
DL 1416
IC1
4051
IC2
4051
KEYPAD
BOWMAR
KBA 32Ø
+5
+XV
GND
C1
22µF
SO1
CUØ
CSØ
CU1
CS1
CU2
CS2
DAØ
DA1
DWS
BKA6
BKA5
BKA4
BKA3
BKA2
BKA1
BKAØ
BKA3
KPAD


HIGH F
LOW F
25 INPUTS
24 INPUTS
24 INPUTS
CMI-11 MULTIPLEXER
CMI-11 MULTIPLEXER
CMI-11 MULTIPLEXER
J1C
J1B
J1A
SO2
SO2
SO2
-5
SW1
SW2
SW3
SW4
SPARE
R1 1k
R2 4k7
R3 1k
R4 4k7
R5 1k
R6 4k7
IC1 7415C
IC2 7415C
IC3 7415C
7415C PIN 7-12 PIN 4-12
C6 100n
C5 100µ
C4 100n
C3 100n
C2 100µ
C1 100n
C8 100n
C7 100n
VR2 7905 -CT
VR1 7805 -CT
-12 TO 7415C
+12 TO 7415C
ZD2 4V7
ZD1 4V7
J2
TO SLAVE KEYBOARD CONNECTOR (25 PIN D SERIES) CN1
50-1
DRAWN: PV

SIGNAL AND POWER WIRING
MC004-01

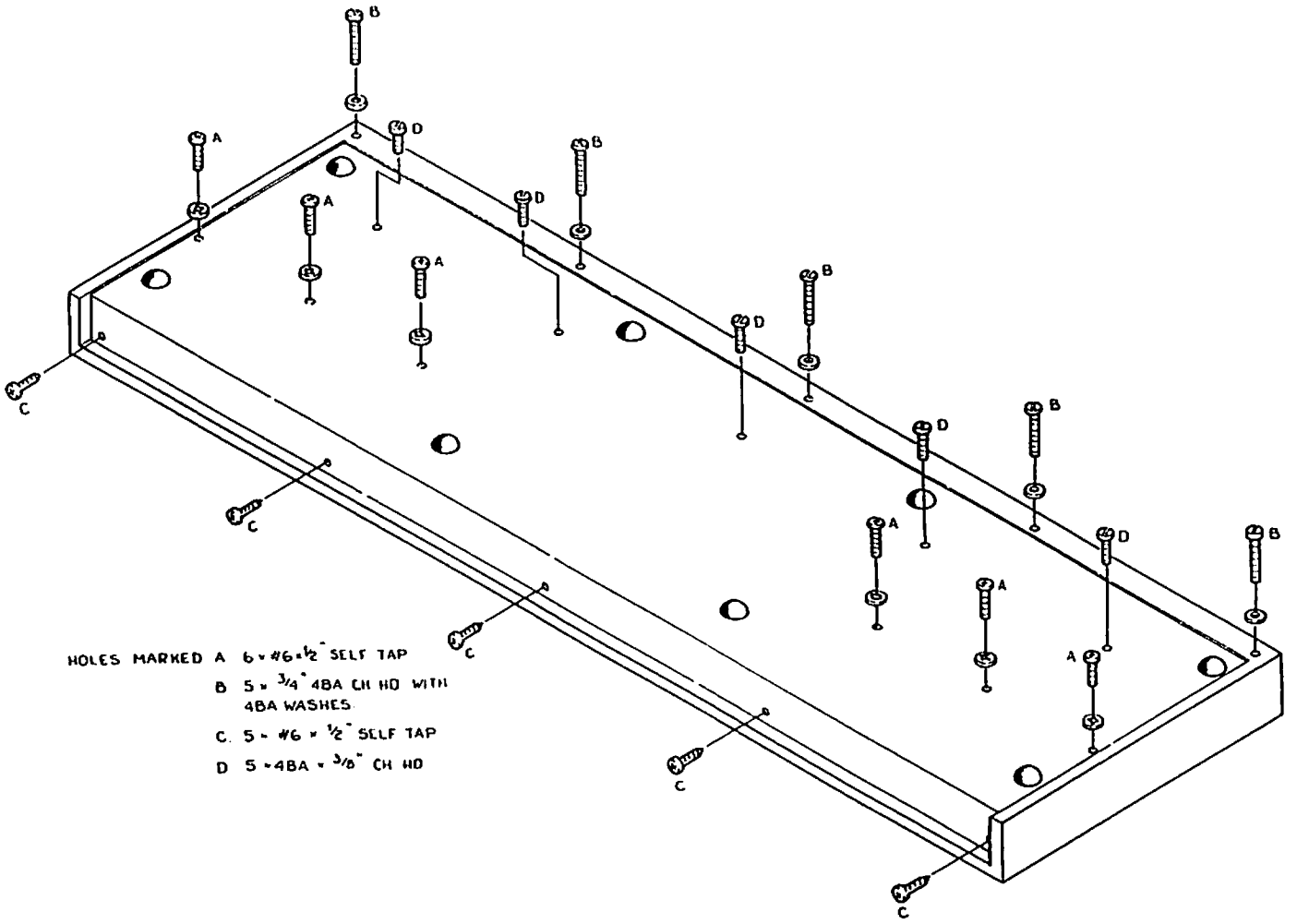


SIGNAL AND POWER WIRING
MC004-01



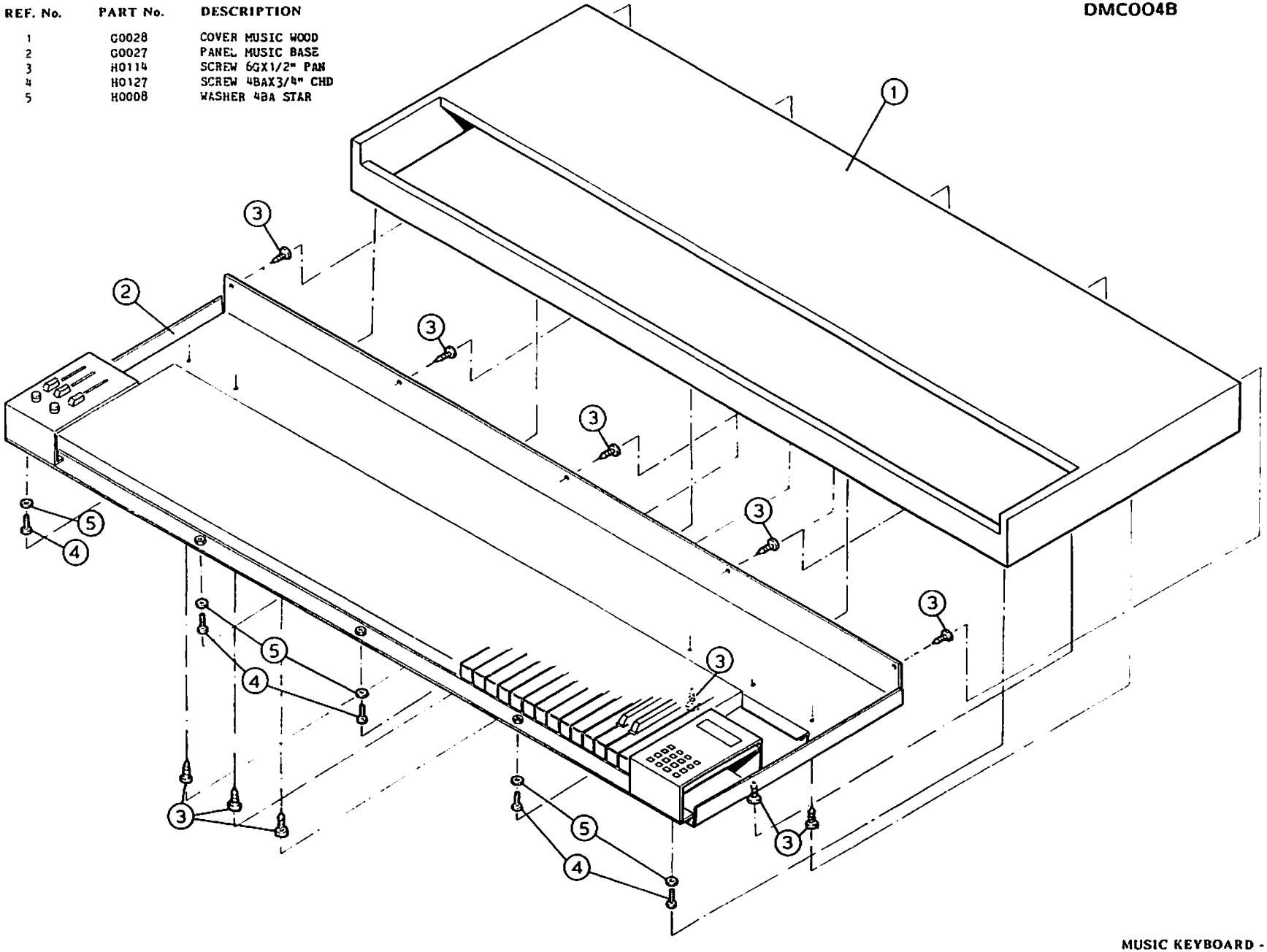
MUSIC KEYBOARD - 8.39

FIXING SCREWS ON MASTER KEYBOARD TOP COVER
DMC004C

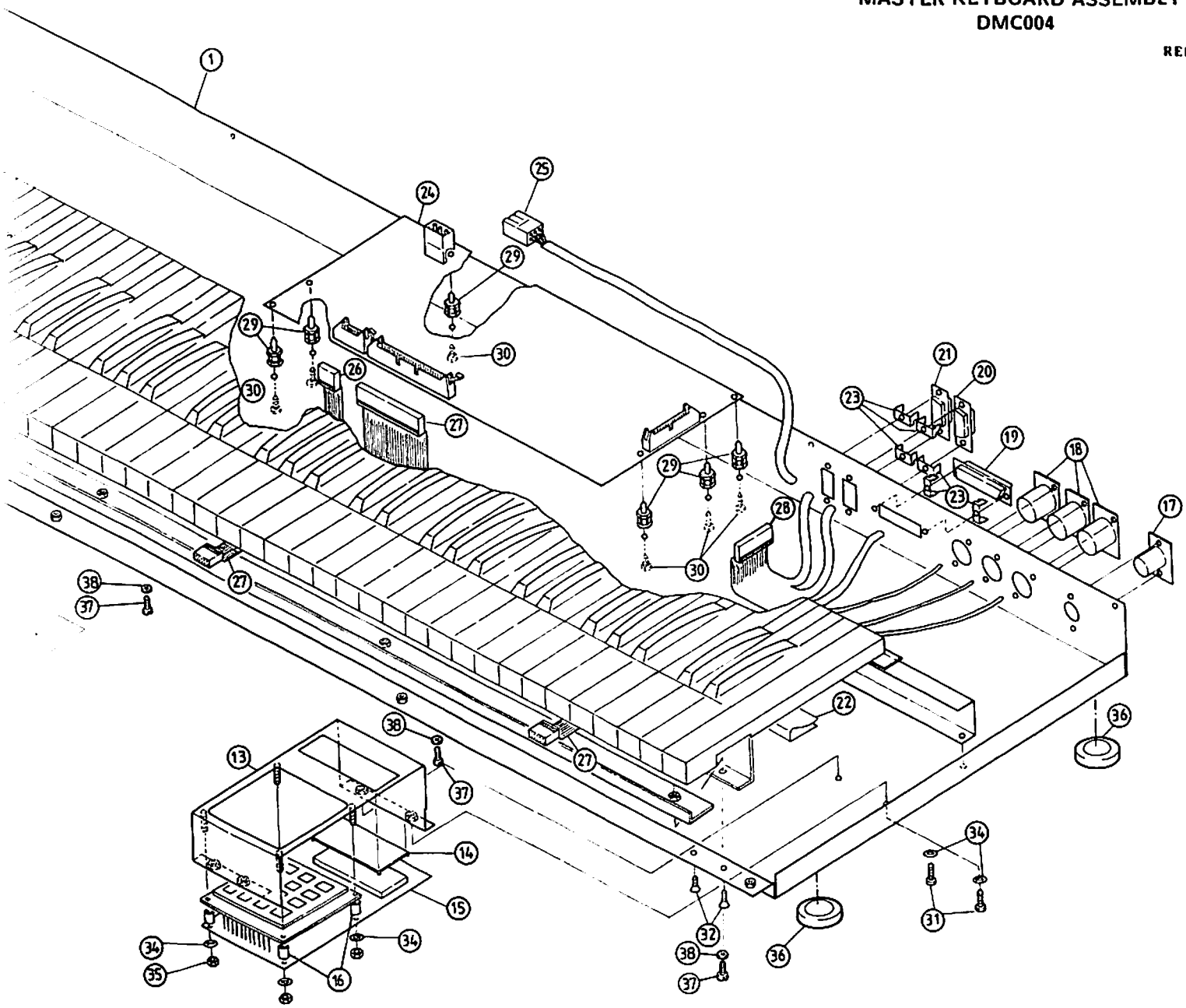


MASTER KEYBOARD COMPLETE
DMC004B

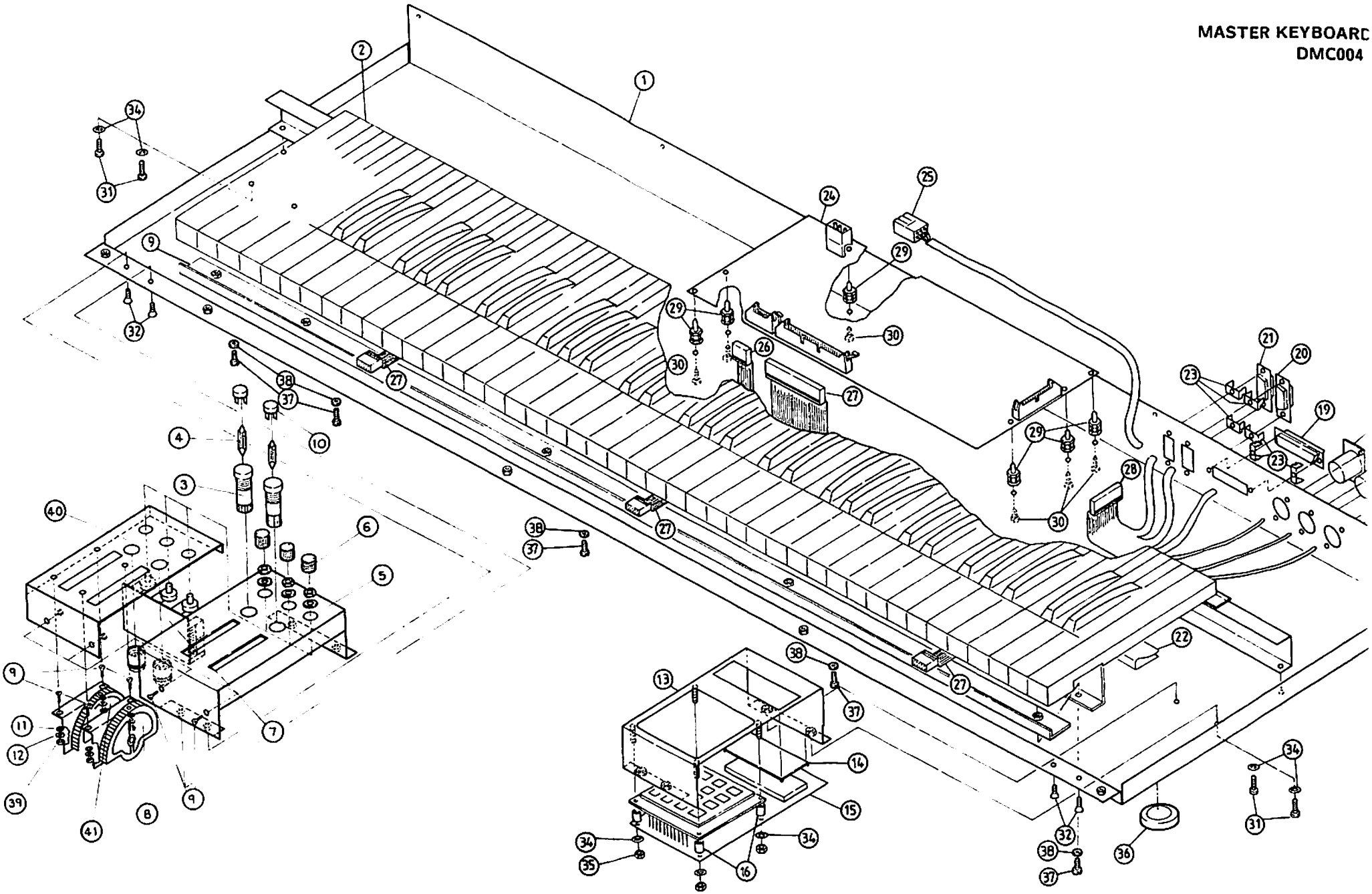
REF. No.	PART No.	DESCRIPTION
1	G0028	COVER MUSIC WOOD
2	G0027	PANEL MUSIC BASE
3	H0114	SCREW 6GX1/2" PAN
4	H0127	SCREW 4BAX3/4" CHD
5	H0008	WASHER 4BA STAR



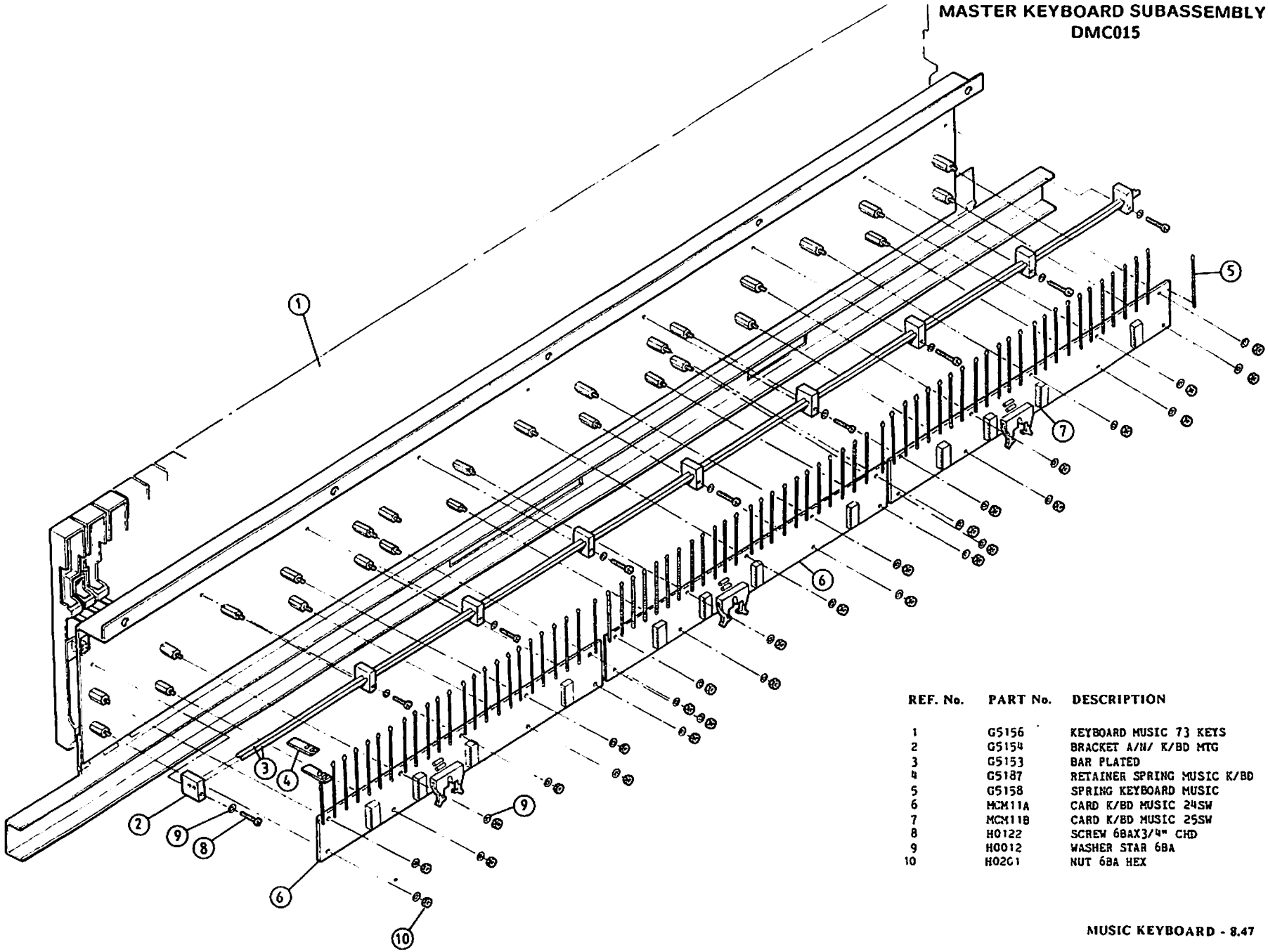
MASTER KEYBOARD ASSEMBLY
DMC004



REF. No.	PART No.	DESCRIPTION
1	300/027	PANEL, BASE
2	MC015	KEYBOARD ASSLY
3	G5405	SWITCH, ILLUM.
4	G5407	GLOBE, 24V
5	300/418	COVER, END CHEEK
6	G5426	KNOB
7	MCMI319	P.C.B. ASSLY
8	G8089	WHEEL ASSLY, VARIABLE CONTROL
9	H1056	SCREW, CSK BLACK, M3 X 10
10	G5406	BEZEL, WHITE
11	H0029	WASHER, STAR, M3
12	H0037	WASHER, FLAT, M3
13	300/025	END CHEEK, R/H
14	G5165	BEZEL, RED
15	MCMI12	CARD, L.E.D. DISPLAY
16	G5142	SPACER, 6BA X 1/4"
17	D6738	SOCKET, CANNON 7P
18	D6710	SOCKET, CANNON 5S
19	D6729	SOCKET, D MINI 25S
20	D6727	SOCKET, D MINI 9S
21	D6728	SOCKET, D MINI 9P
22	G5122	CLIP, CABLE
23	D6731	LUG, D MINI
24	MCMI10	CARD, CMI 10
25	G5219	CONNECTOR, LOOM
26	G6007	CABLE, PITCH WHEEL
27	MC063	CABLE, KEYBOARD ASSLY
28	MC071	CABLE, REAR PANEL
29	G5107	STANDOFF
30	H0125	SCREW, CHD, 6g X 1/4"
31	H0124	SCREW, CHD, 6BA X 1/4"
32	H0117	SCREW, CSK, 6BA X 1/4"
33	H0130	SCREW, CSK, 6BA X 3/16"
34	H0012	WASHER, STAR, 6BA
35	H0201	NUT, 6BA
36	G5183	FOOT, RUBBER
37	H0112	SCREW, CHD, 4BA X 3/8"
38	H0008	WASHER, STAR, 4BA
39	H2020	NUT, M3
40	300/419	CHASSIS, END CHEEK
41	G8091	WHEEL ASSLY, SPRING RETURN

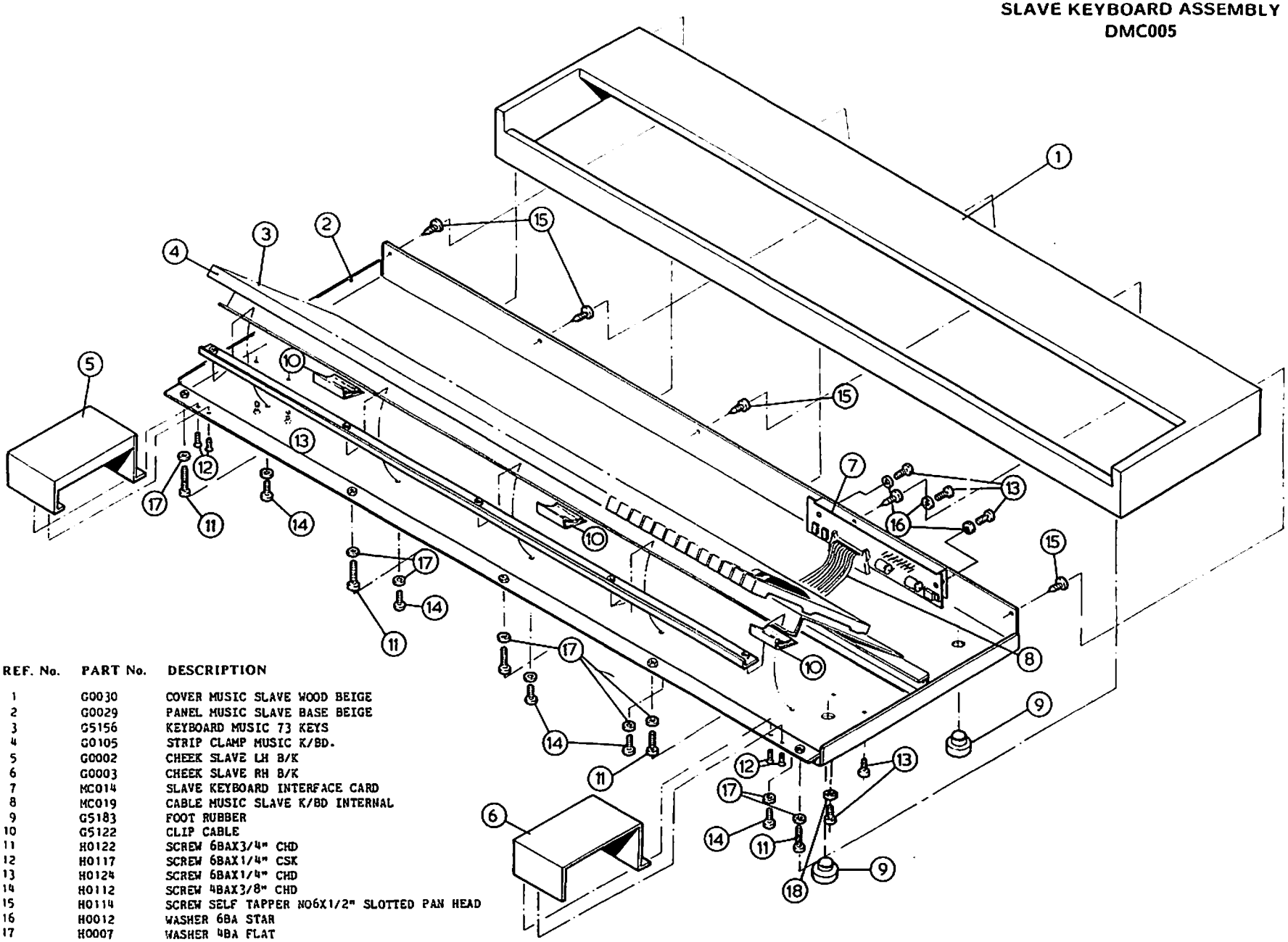


MASTER KEYBOARD SUBASSEMBLY
DMC015



REF. No.	PART No.	DESCRIPTION
1	G5156	KEYBOARD MUSIC 73 KEYS
2	G5154	BRACKET A/N/ K/BD MTG
3	G5153	BAR PLATED
4	G5187	RETAINER SPRING MUSIC K/BD
5	G5158	SPRING KEYBOARD MUSIC
6	MCM11A	CARD K/BD MUSIC 24SW
7	MCM11B	CARD K/BD MUSIC 25SW
8	H0122	SCREW 6BAX3/4" CHD
9	H0012	WASHER STAR 6BA
10	H02G1	NUT 6BA HEX

SLAVE KEYBOARD ASSEMBLY
DMC005



REF. No.	PART No.	DESCRIPTION
1	G0030	COVER MUSIC SLAVE WOOD BEIGE
2	G0029	PANEL MUSIC SLAVE BASE BEIGE
3	G5156	KEYBOARD MUSIC 73 KEYS
4	G0105	STRIP CLAMP MUSIC K/BD.
5	G0002	CHEEK SLAVE LH B/K
6	G0003	CHEEK SLAVE RH B/K
7	MC014	SLAVE KEYBOARD INTERFACE CARD
8	MC019	CABLE MUSIC SLAVE K/BD INTERNAL
9	G5183	FOOT RUBBER
10	G5122	CLIP CABLE
11	H0122	SCREW 6BAX3/4" CHD
12	H0117	SCREW 6BAX1/4" CSK
13	H0124	SCREW 6BAX1/4" CHD
14	H0112	SCREW 4BAX3/8" CHD
15	H0114	SCREW SELF TAPPER NO6X1/2" SLOTTED PAN HEAD
16	H0012	WASHER 6BA STAR
17	H0007	WASHER 4BA FLAT

Graphics Monitor 9

Contents

Introduction

The VDU supplied with each series III CMI is likely to be either a Multitech or Thomas unit. Specifications and adjustment procedures for both units are as follows :

Specifications

Picture tube	12" picture measured diagonally 90- degree deflection angle
Phosphor	Green (Ki38) or Amber (K134)
Scanning Frequency	HorizontaL 15625Hz - 15750Hz Vertical 50 or 60Hz
Signal input	Belling - Lee 5 pin
Signal output connector	RCA
Input signal	1.0 - 2.0 Vp-p composite video
Video band width	18 MHz typical
Resolution	1000 lines
Power consumption	28W max. AC
Dimensions	Approx. 381 x 279 x 315mm (WxHxD)
Weight	Approx. 6.2g (13.6 lb)

Operation

Connect cable from computer input jack.

Put the mode switch in higher position (75 ohm mode) if only one monitor is used, and in lower position (Hi Z mode) for driving several monitors.

Switch power on to the VDU by pressing the POWER switch or by turning the brightness control clockwise.

Adjust the BRIGHTNESS controls to obtain optimum display quality.

Recommended Procedures

Do not install the monitor in a location near heat sources such as radiators or airducts, or in a place subject to direct sunlight, or excessive dust or mechanical vibration or shock.

Allow adequate air circulation to prevent internal heat build-up. Do not place the monitor set on soft surfaces (rugs, blankets, etc.) or near materials (curtains, draperies) that may block the ventilation holes.

Save the original shipping carton and packing material, they will come in handy if you ever have to ship your monitor.

For maximum protection, repack your monitor as it was originally, packed at the factory.

To keep the monitor looking brand new, periodically clean it with a soft cloth. Stubborn stains may be removed with a cloth lightly dampened with a mild detergent solution. Never use strong solvents such as thinner or benzine or abrasive cleansers, since these will damage the cabinet. As a safety precaution, always unplug the monitor before cleaning it.

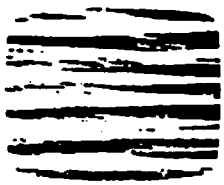
Picture Adjustment

The following adjustments are made using the controls on the front or rear of the VDU.

If the picture is too bright or too dark, adjust the BRIGHTNESS CONTROL.

If the picture contrast is too strong or too weak, adjust the CONTRAST CONTROL.

If the picture is not stable in horizontal, adjust the HORIZONTAL HOLD CONTROL.

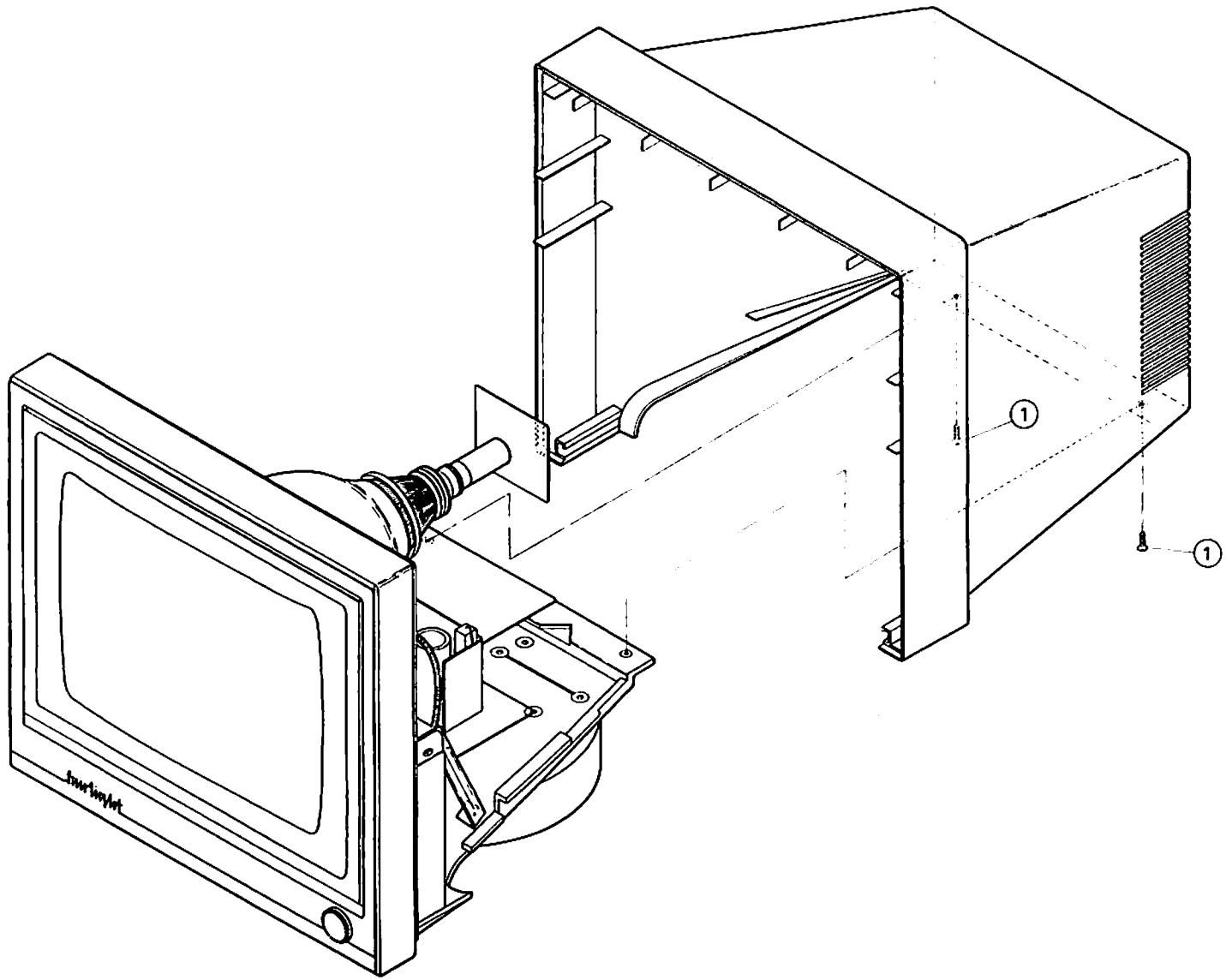


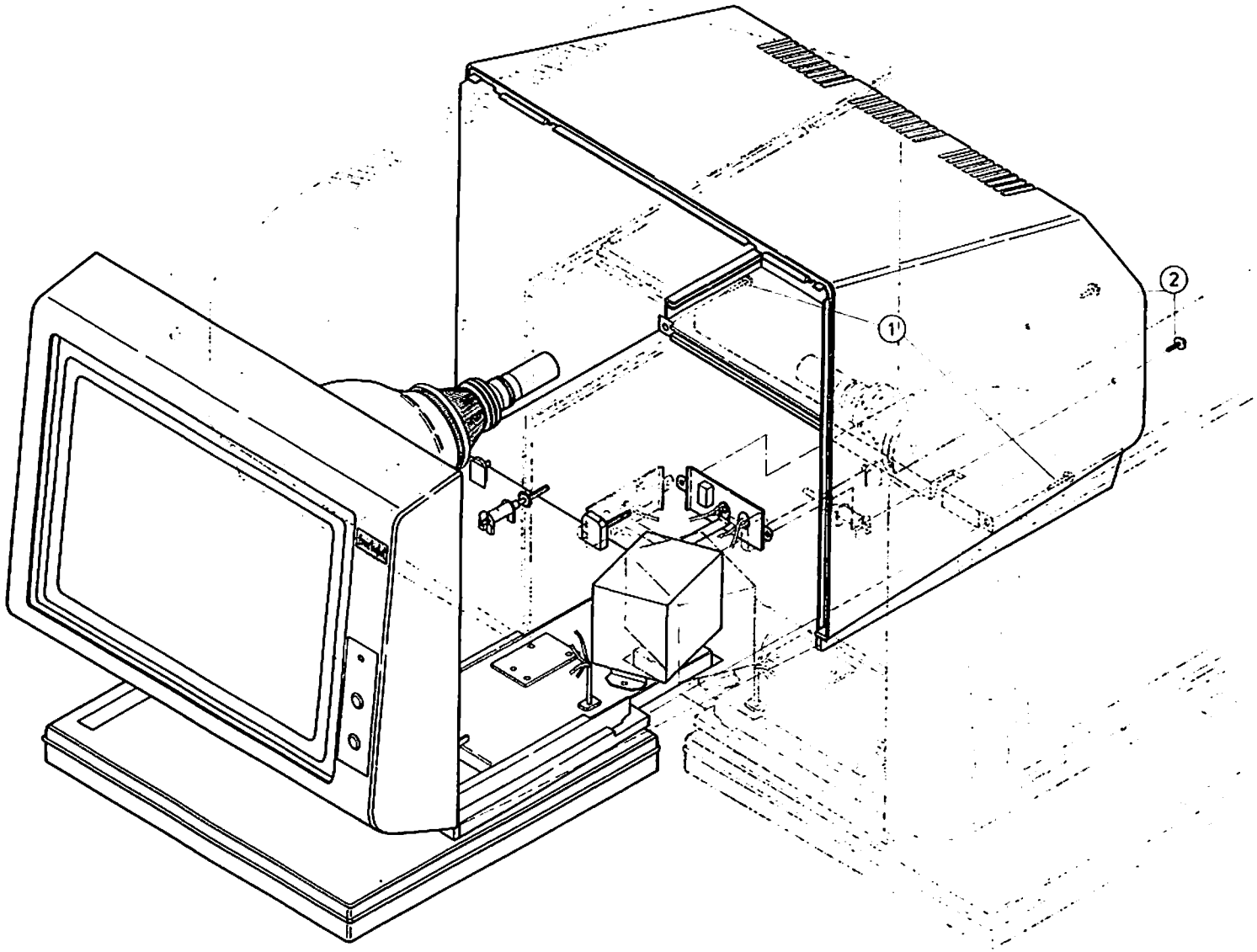
If the picture moves up or down or rolls, adjust the VERTICAL HOLD CONTROL.

If the height of the picture is not suitable, adjust V-SIZE CONTROL.

If the picture is not on the centre of the screen, adjust H-PHASE CONTROL.

If a satisfactory display cannot be achieved by means of the above adjustments, return the unit to Fairlight for repair or replacement.


1
1


1
2