

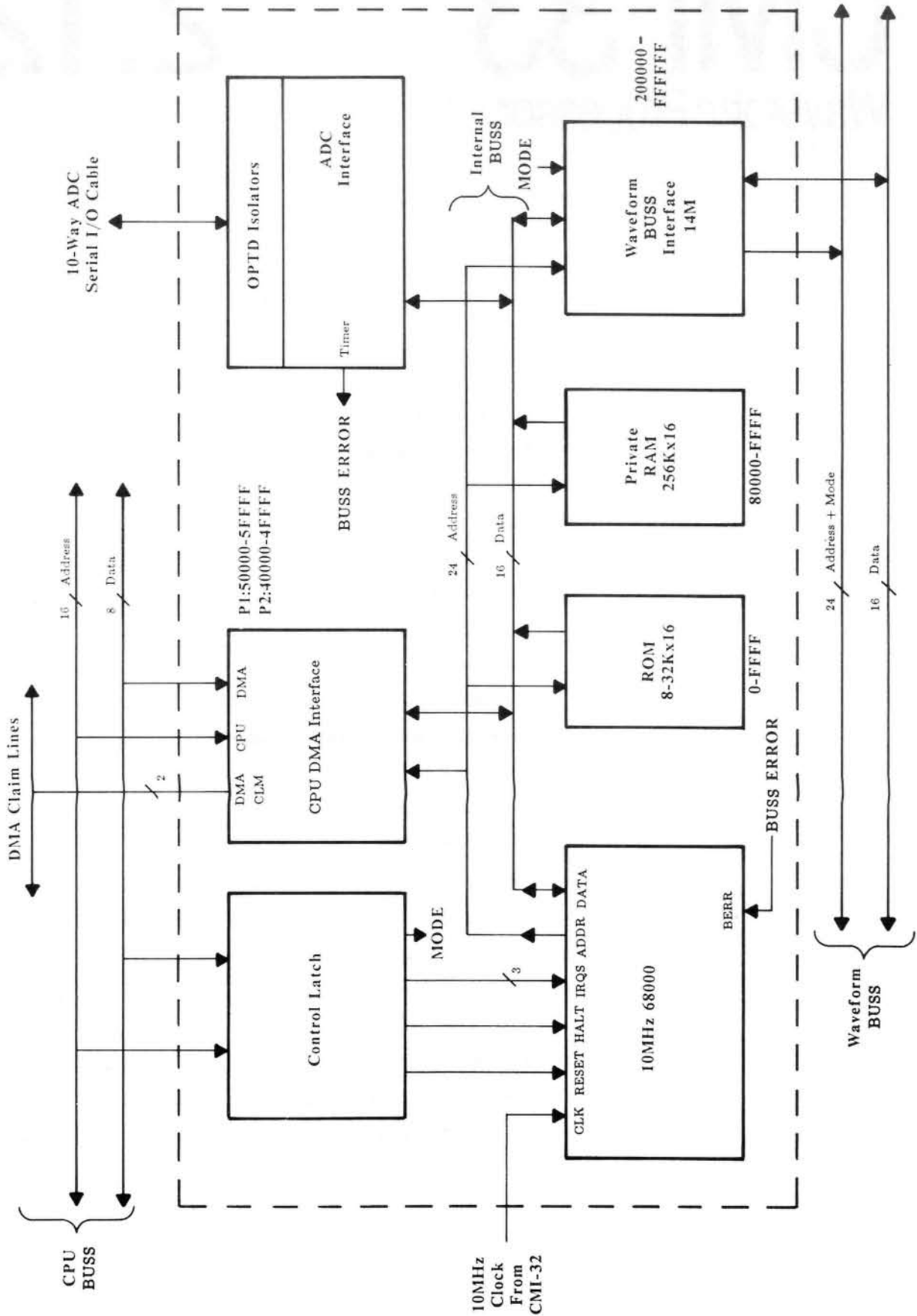
# CMI-33

Waveform Processor

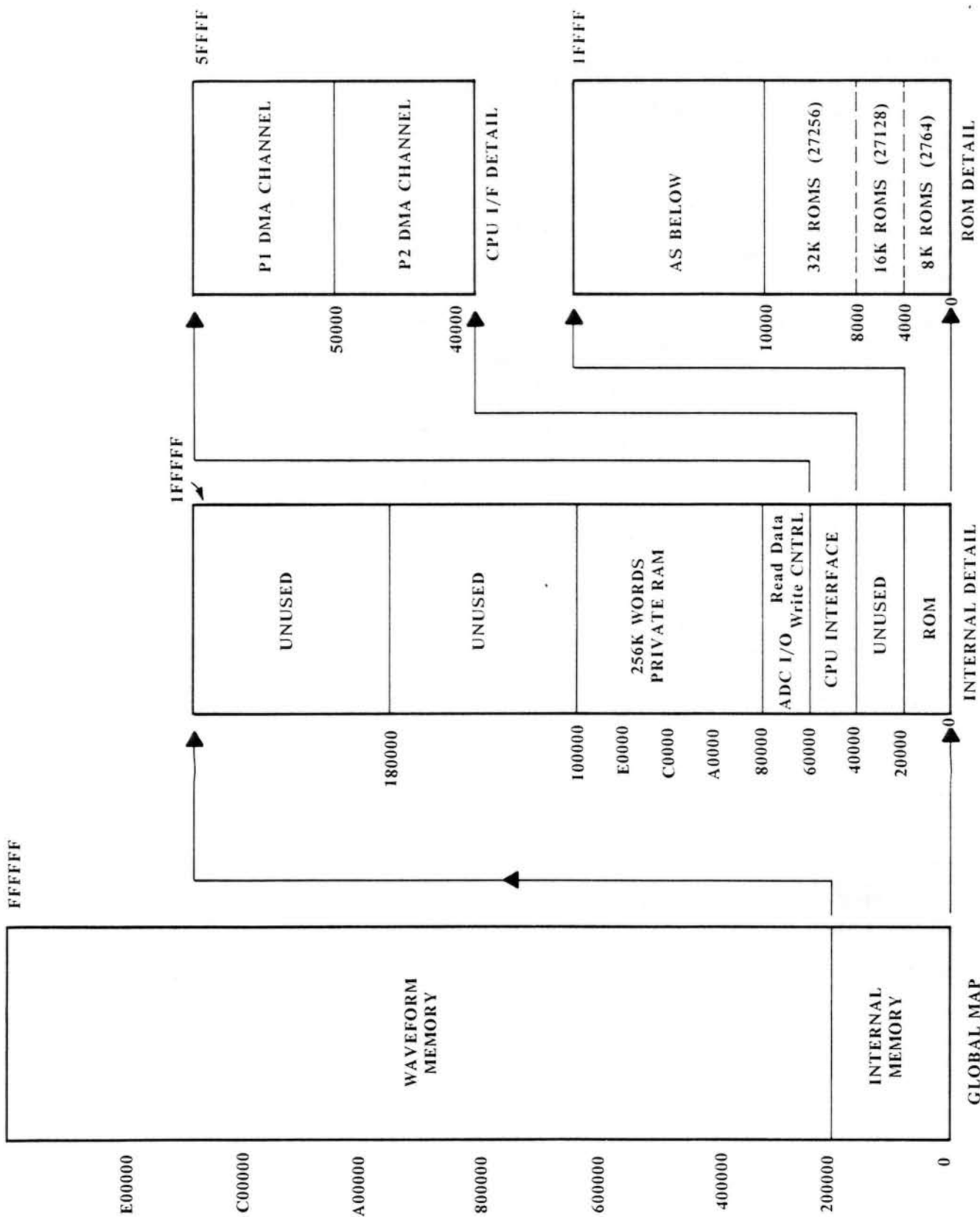
# 2.12

Block diagram.....	2.12.2
Hardware memory map.....	2.12.4
Introduction.....	2.12.3
ROMs.....	2.12.4
Private RAM.....	2.12.4
CPU DMA interface.....	2.12.5
Waveform buss interface.....	2.12.5
ADC serial interface.....	2.12.6
Miscellaneous items.....	2.12.6
Processor, ROMs, and decoding.....	2.12.6
Private RAM.....	2.12.7
Interrupt and reset control.....	2.12.9
68000/6809 DMA buss interface.....	2.12.10
Timing summary.....	2.12.13
Debugging notes.....	2.12.13
Waveform buss arbitration and interface.....	2.12.13
Timing summary.....	2.12.15
ADC I/O Interface.....	2.12.15
Timing summary.....	2.12.16
Schematic diagrams.....	2.12.17
Timing diagrams.....	2.12.23

Block diagram



### Hardware Memory Map



*Fairlight*

---

## CMI-33 Waveform Processor

---

### 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 asynchronous 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.

---

## CMI-33 Waveform Processor

---

### 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 (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  $\overline{AS}$  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  $\overline{DAS}$  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  $\overline{EPROM}$  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 C14 to assert  $\overline{DTACK1}$ . When the processor terminates the cycle,  $\overline{AS}$  is negated so  $\overline{BAS}$  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  $\overline{EPROM}$  through the decoding circuitry plus the two counted clock cycles, plus the delay between  $\overline{DTACK}$  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  $\overline{\text{RAS}}$ . 50nS later, which is sufficient row address hold time, the second flip-flop is clocked by rising edge of  $\overline{\text{PCLK}}$  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  $\overline{\text{DTACK2}}$  while the other side of the flip-flop goes through the AND gate G9 and ALS00 gates at E9 to assert  $\overline{\text{UCAS}}$  or  $\overline{\text{LCAS}}$  or both. Having two separate  $\overline{\text{CAS}}$  signals allows the RAM to be accessed a byte at a time or a word at a time. The 100nS between the recognition of  $\overline{\text{DTACK}}$  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  $\overline{\text{CAS}}$ . 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  $\overline{\text{AS}}$  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  $\overline{\text{DAS}}$  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  $\overline{\text{RAS}}$  low for a specified period.  $\overline{\text{CAS}}$  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").

Bus Grant (/BG): output from the 68000 indicating that  $\overline{\text{BR}}$  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").

Bus Grant Acknowledge (/BGACK): Input to the 68000 to indicate control of the buss has been taken by the alternate buss controller (refresh). Once  $\overline{\text{BG}}$  has been asserted and the current cycle completed (indicated by  $\overline{\text{AS}}$  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  $\overline{\text{BR}}$  has been asserted it asserts Bus Grant (/BG). The buss becomes available when  $\overline{\text{BG}}$  is asserted and the address strobe (/AS) is negated. Shortly after it is negated  $\overline{\text{AS}}$  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 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  $\overline{BGACK}$  (F9-6) then clears all stages of the LS175 ready 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. 1 $\mu$ S.

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 running.

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 IPL0-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 multiplexer. If A16 is low, P2 $\phi$ 2 is enabled through to become P $\phi$ 2, 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 P02 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 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 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  $\overline{ACK1}$  or  $\overline{ACK2}$ ; the selected  $\overline{ACK}$  signal becomes  $\overline{PACK}$ . When a request has been generated (F11  $\overline{Q}$  hi) and this level is enabled ( $\overline{ENL}$  lo), the rising edge of  $\overline{PACK}$  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  $\overline{PACK}$  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.  $\overline{VMA}$  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  $\overline{HIBYTE}$  and  $\overline{LOBYTE}$ . 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  $\overline{HIBYTE}$  and  $\overline{LOBYTE}$  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 DTB (Data To Buss) at the rising edge of BRA when a DMA cycle is in progress. This is performed by flip flop G14. 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 CMI being asserted will drive the  $\bar{G}$  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 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 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 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 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 DTACK4 is not generated because of the propagation delay through to the next flip flop. Since 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 ATB and DTB are negated respectively. If reading, the first byte read in is latched and held by C6 or C7. Since 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 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 CMI, then that the daisy chain signals are present. Then look for an 800nS pulse on DCYCLE, indicating that DMA cycles are actually occurring. Continue through to the ATB, DTB and 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 possibility 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 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:

---

## CMI-33 Waveform Processor

---

/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  $\overline{WATB}$  (Waveform Address to Buss) is low (300nS). The data buss is enabled for 300nS from the rising of  $\overline{DSCLK}$ , 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.  $\overline{WDTB}$  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.  $\overline{WDTB}$  indicates that the last cycle was a WP access, so the PAL ignores WMEM.

$\overline{DTACK3}$  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  $\overline{W8BIT}$  mode signal from the control latch. There is no need to do this except as a means of testing the WRAM cards. The  $\overline{8BIT}$  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.

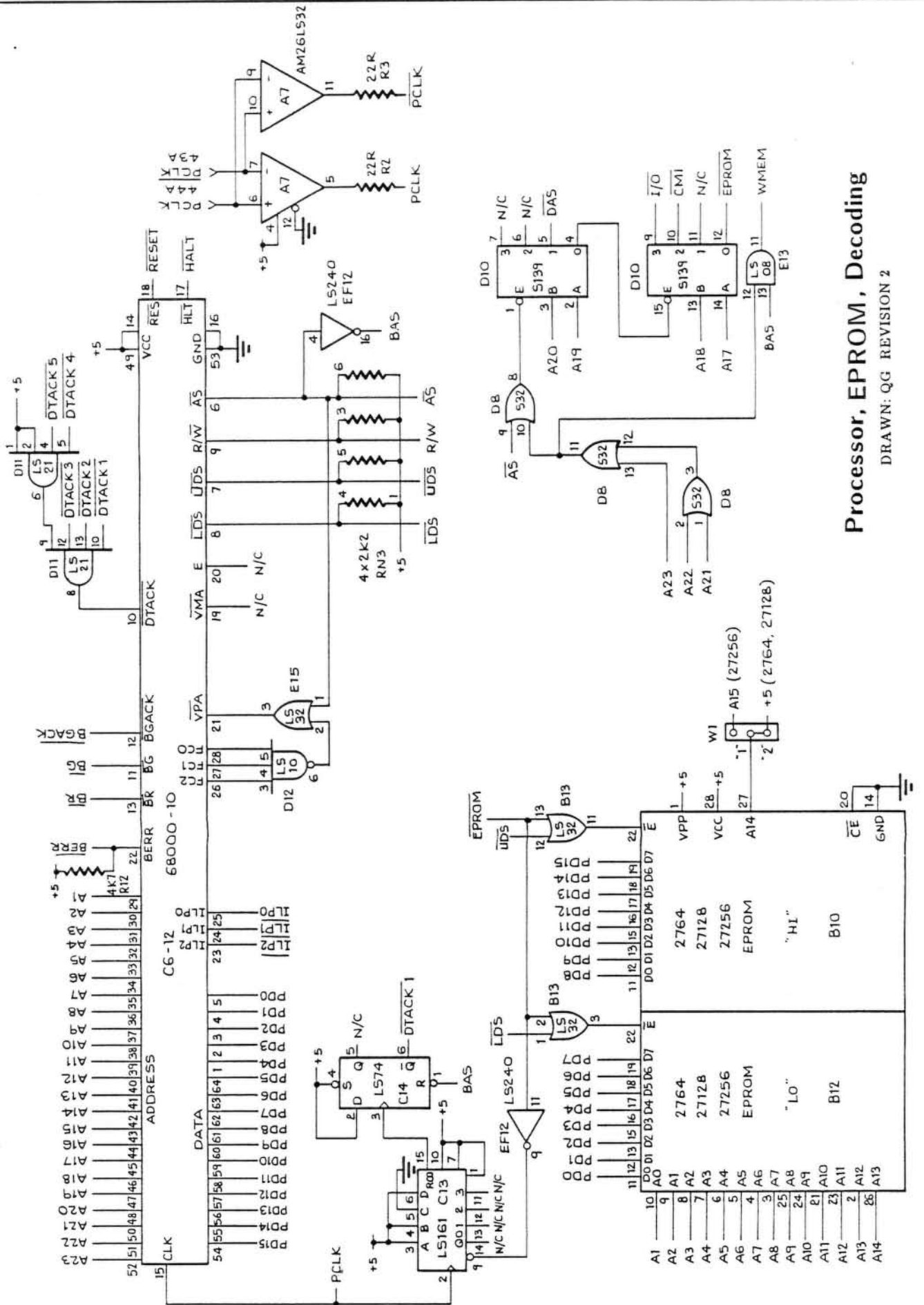
Control data is written from the 68000 to parallel inputs of LS597s D7 and E7, latched on the falling edge of either or both data strobes. The shift register is loaded with the contents of the latch when  $\overline{EOC}$  goes low at the end of the conversion. Termination of the write cycle is the same 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.*

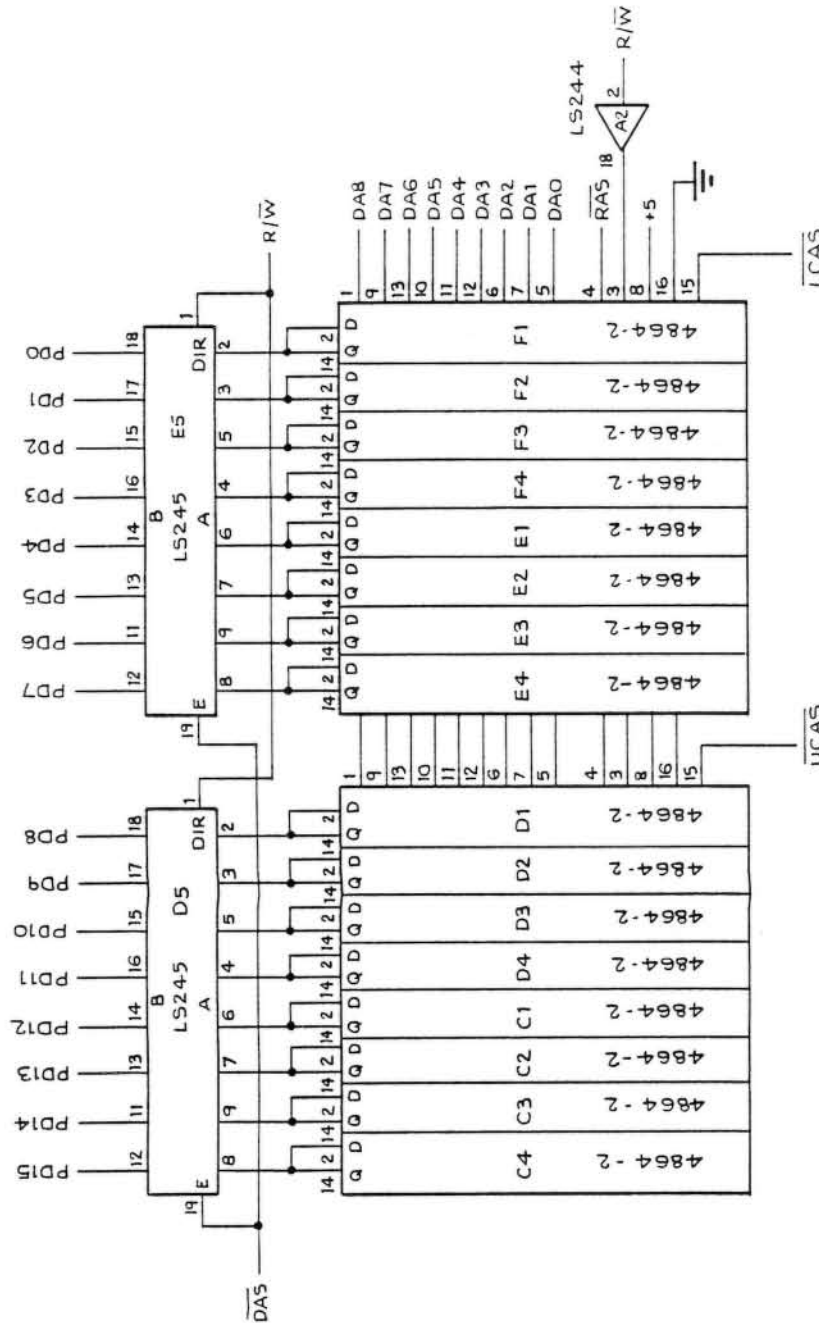
Note that ADC accesses "hang" the processor, with the buss active for up to 22.6uS (assuming a 44kHz mono sample rate) waiting for the next  $\overline{EOC}$  edge. Since refresh requests arrive every 16uS, this means that the next cycle after the completion of an ADC access will always be a refresh 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

Starlight





DRAMS 150 nS ACCESS MAX. 256K x 18BIT NORMALLY INSTALLED

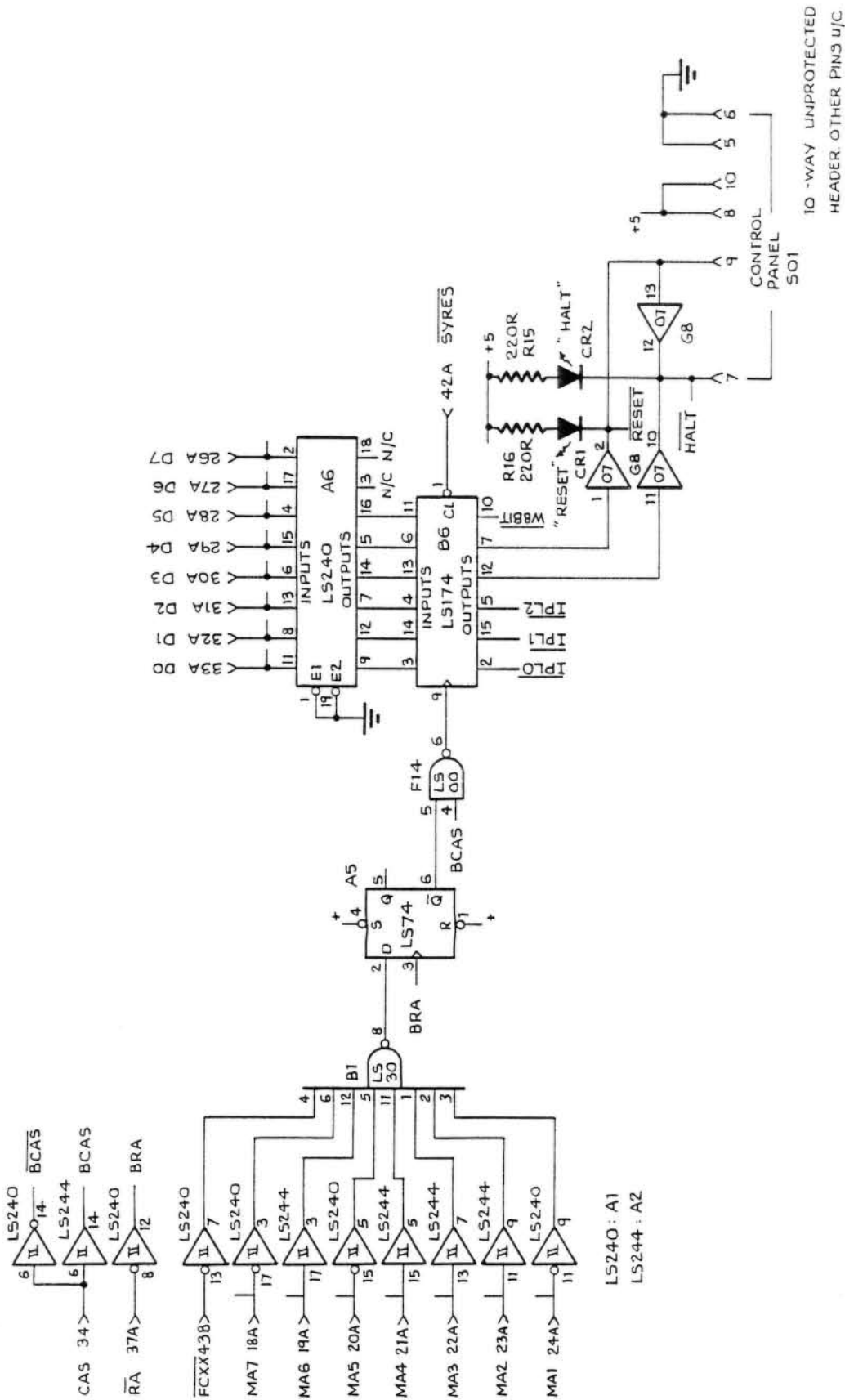
- 64K EQUIVALENTS :
- 6655 - 15 (MOT)
  - 4164 - 15 (FAIRCHILD)
  - 4564 - 15 (MOSTEK)
  - 4164 - 15 (MITSUB)
  - 8264 - 15 (FUJITSU)

256K COMPATIBLE TYPE :

- 6256 - 15 (MOT)
- TMS 4256 (TEXAS)

**Local DRAM 64k/256k Words**

DRAWN: QG REVISION 2

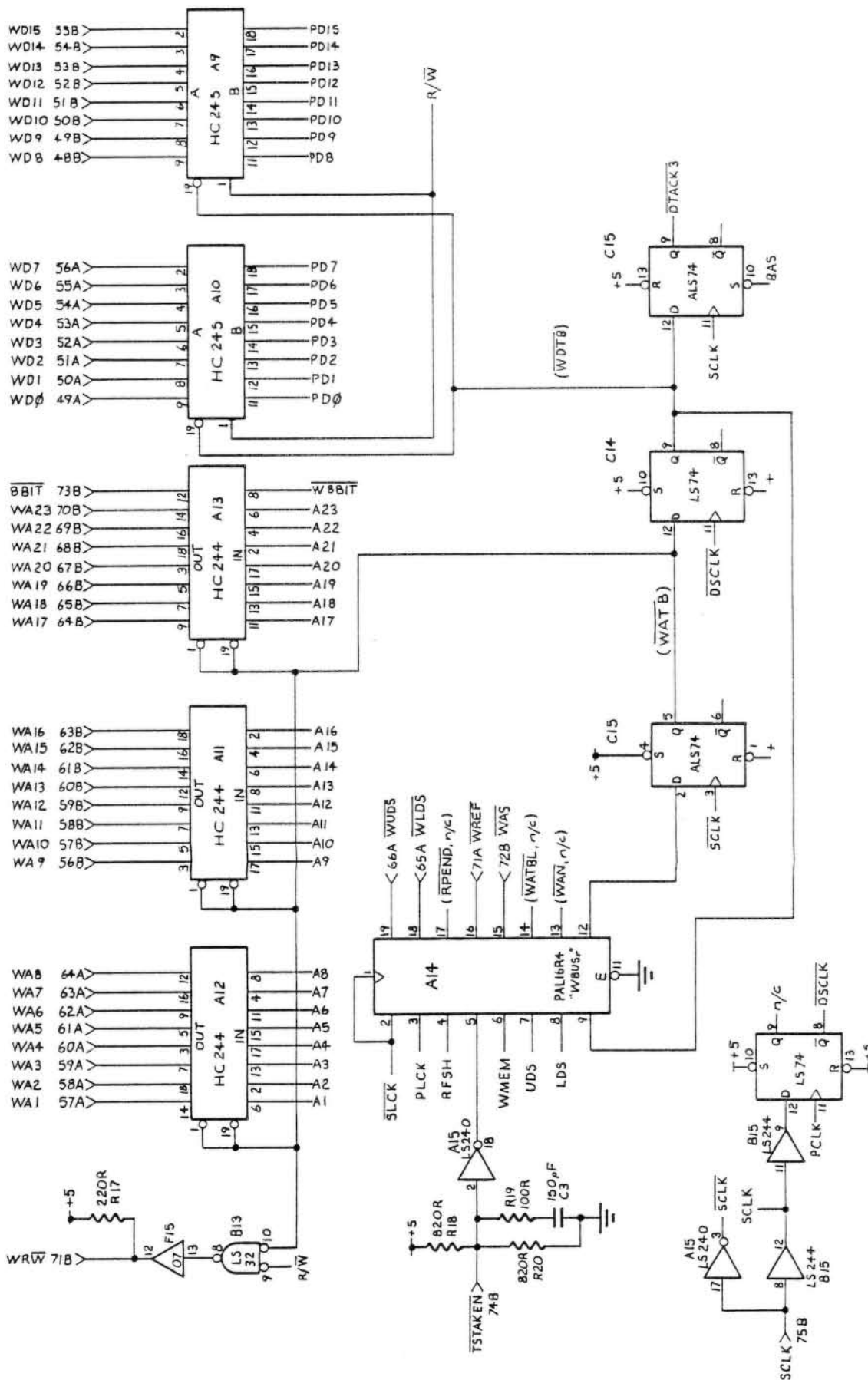


### Interrupt and Reset Control

DRAWN: QG REVISION 2

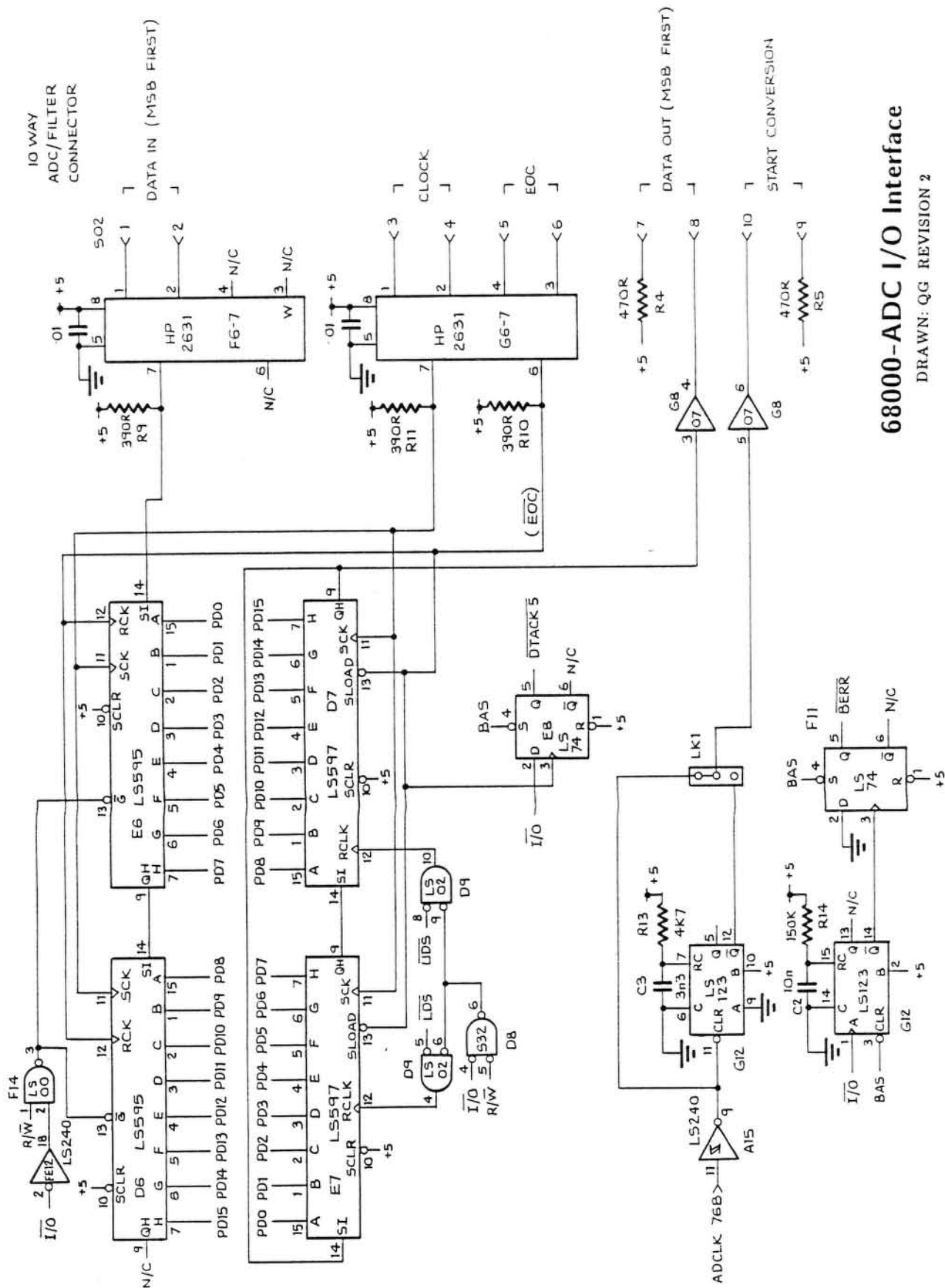


# CMI 33-05 Waveform Processor



**Waveform Bus Interface,  
Waveform Refresh Arbitration**  
DRAWN: QG REVISION 2

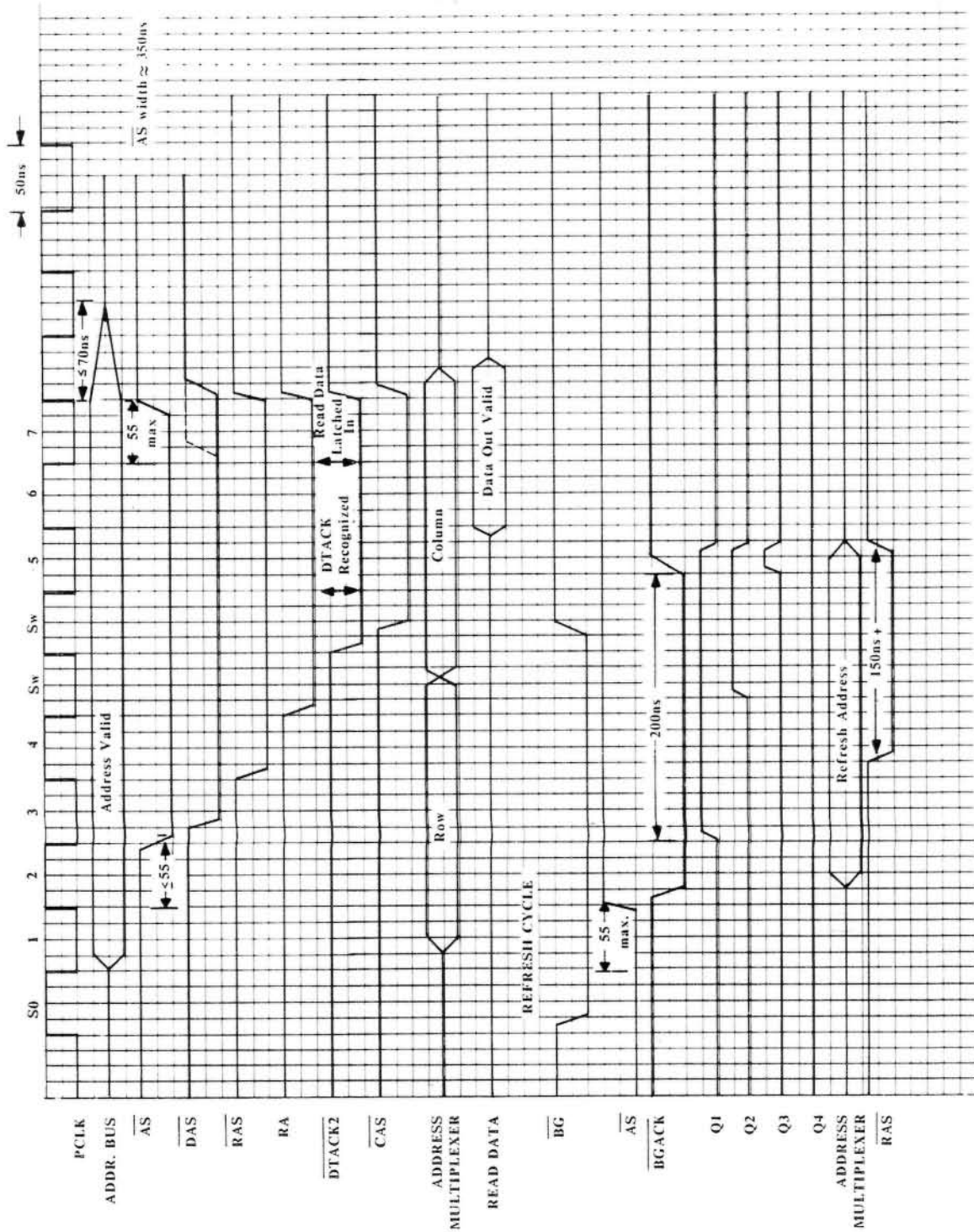
NB. PAL16R4 PROGRAMMABLE LOGIC ARRAY  
PROGRAMMED AS "WBUSr" WHERE r = rev no.



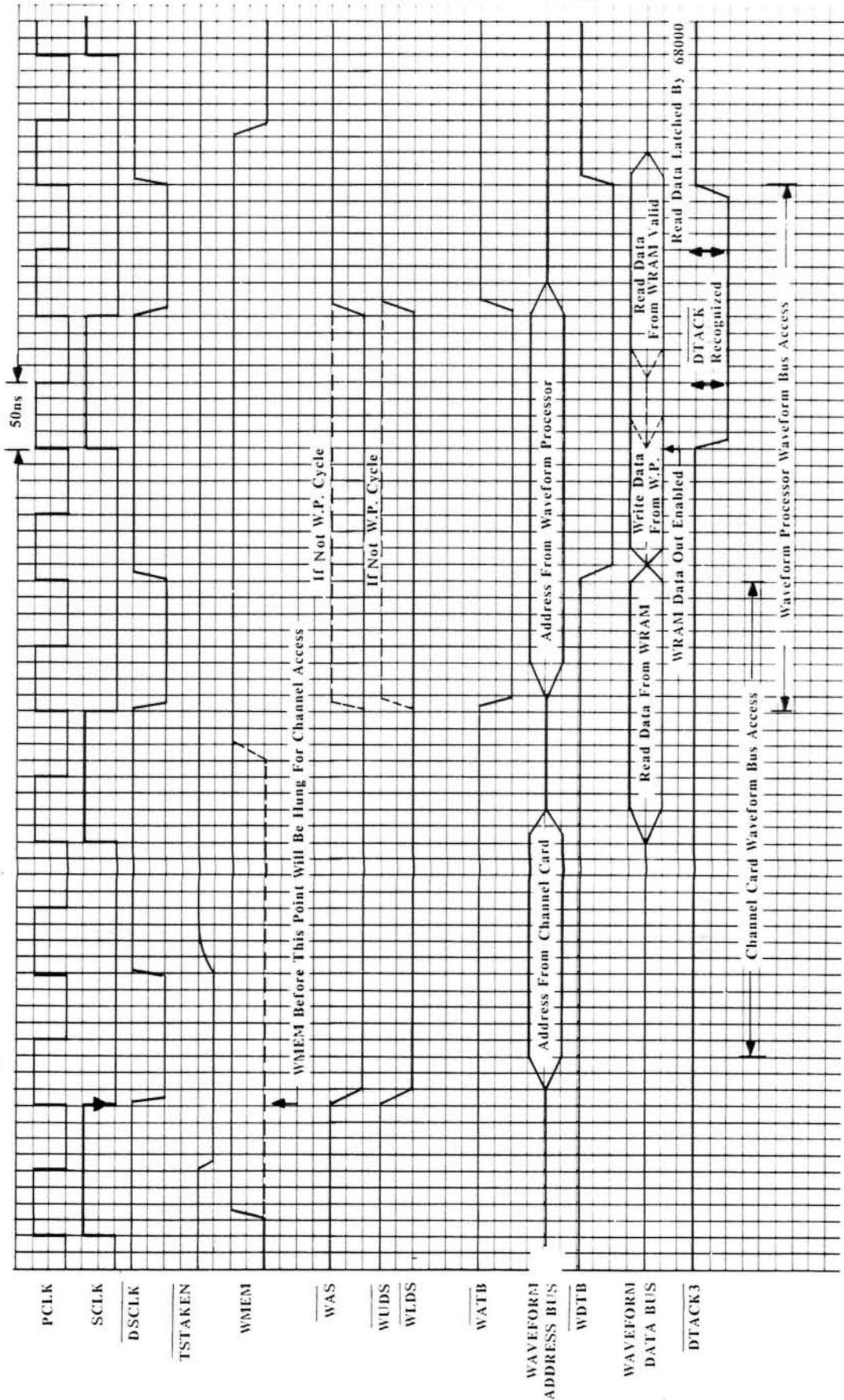
68000-ADC I/O Interface  
DRAWN: QG REVISION 2

*fairlight*

### 68K Timing Dynamic RAM Access



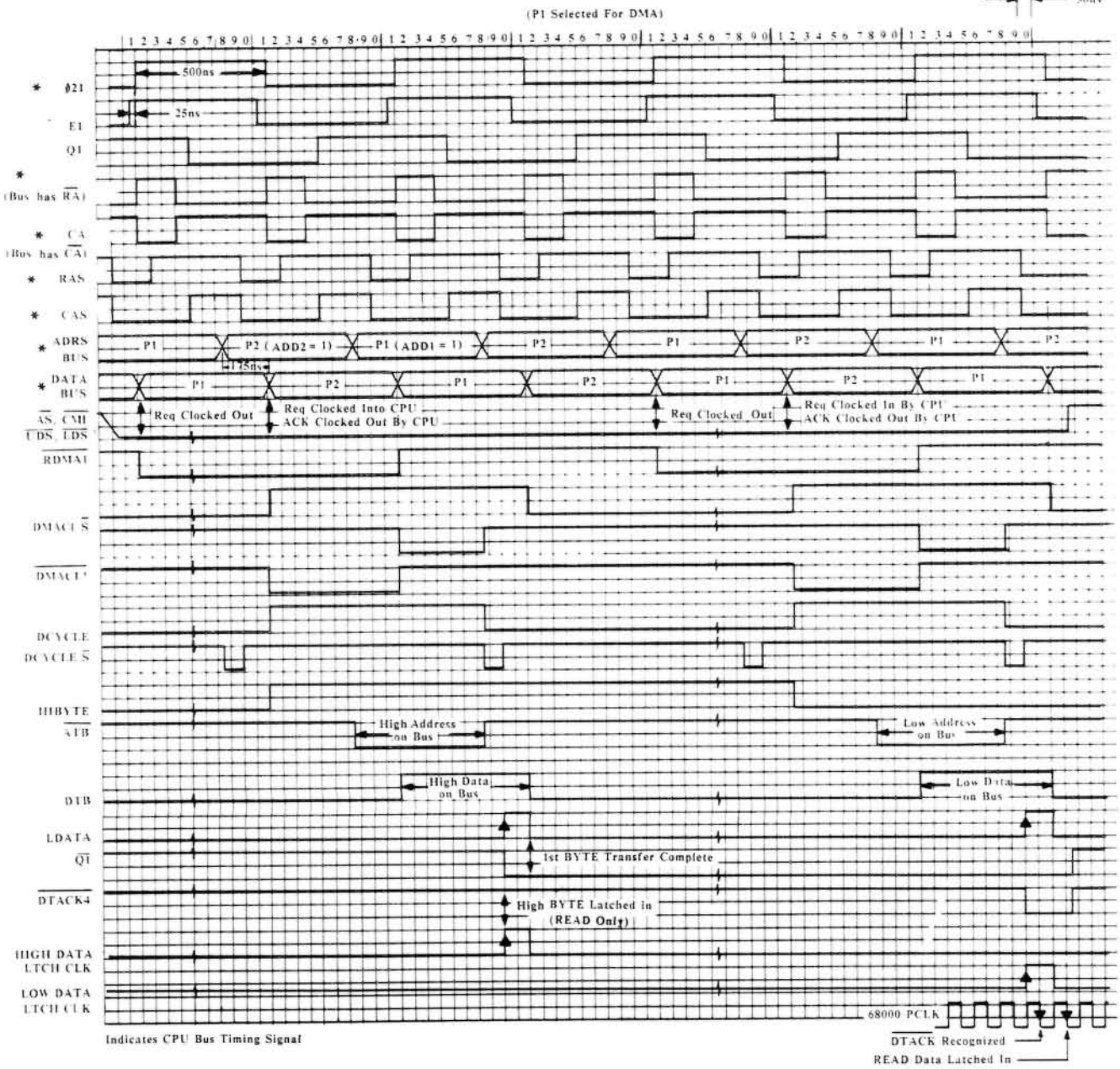
Waveform Bus Interface Timing



*Starlight*

# CMI-33 Waveform Processor

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



# CMI-39

Waveform Ram

# 2.13

Waveform memory map.....	2.13.2
Introduction.....	2.13.3
Memory configuration.....	2.13.3
8/16-bit modes.....	2.13.3
Timing generation.....	2.13.4
Address decoding.....	2.13.5
Address/refresh multiplexor.....	2.13.6
Data buss Interface.....	2.13.6
RAM array.....	2.13.7
Timing tolerances.....	2.13.7
Timing diagrams.....	2.13.9
Schematic diagrams.....	2.13.10

### **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 output 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.

## Waveform RAM CMI-39

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.

3 Upper address lines.

### 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 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 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 disabled 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  $\overline{\text{RAS}}$  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  $\overline{\text{RAS}}$ . In 16-bit mode ( $\overline{\text{B8BIT}}$  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  $\overline{\text{RAS}}$  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  $\overline{\text{CAS}}$ , 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  $\overline{\text{CAS}}$  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  $\overline{\text{BWREF}}$ . 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  $\overline{\text{CAS}}$ . 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  $\overline{\text{CAS}}$  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 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.

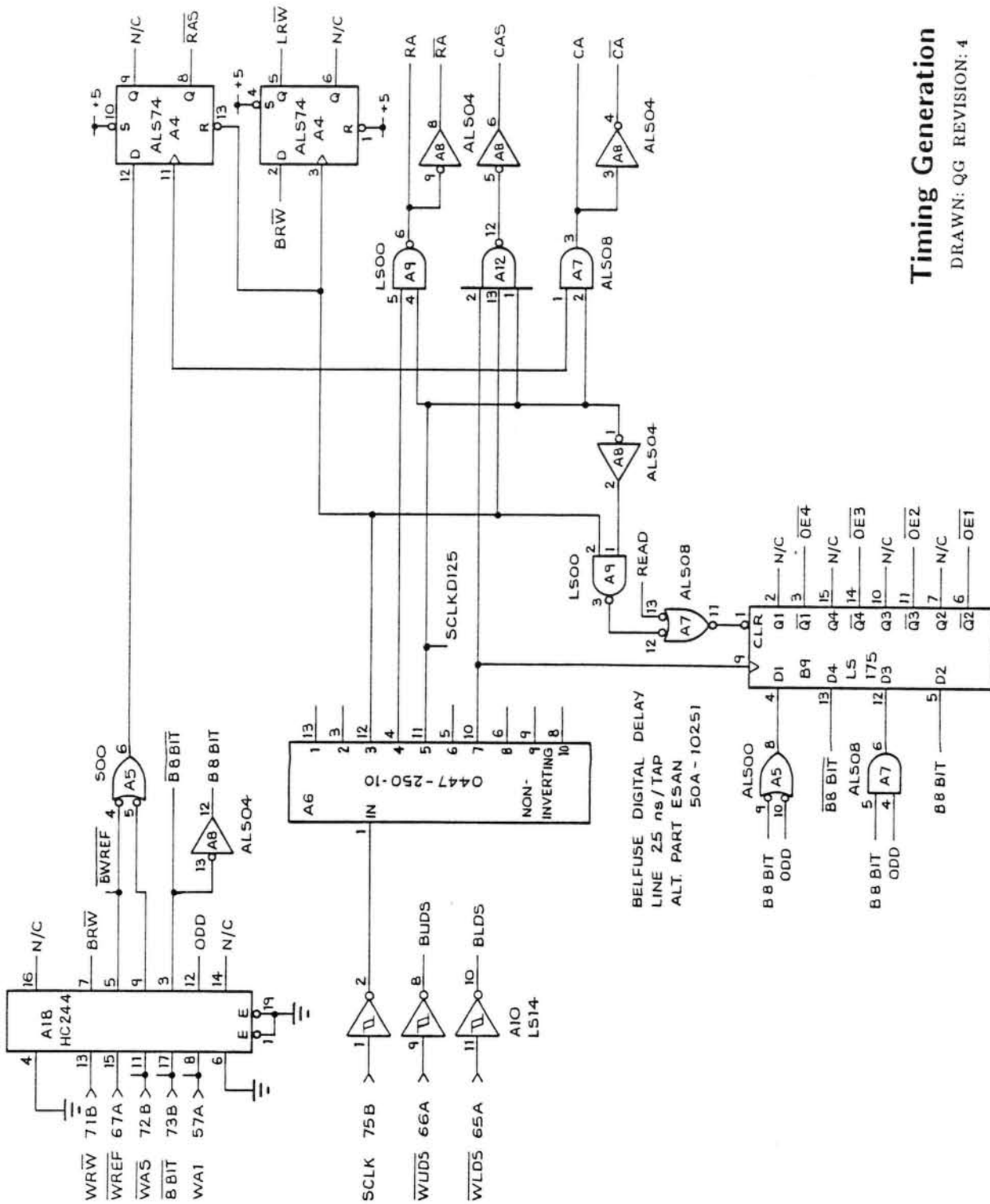
Symbol	Parameter	Spec.	Measured
$t_{RC}$	Cycle time	300	300
$t_{RP}$	RAS precharge	120	120
$t_{RAS}$	RAS width	150	150
$t_{CAS}$	CAS width	75	80
$t_{RCD}$	RAS to CAS lead time	75	75
$t_{RSH}$	RAS hold	75	80
$t_{CSH}$	CAS hold	150	150+
$t_{ASR}$	Row address set up	0	40
$t_{RAH}$	Row address hold	20	20
$t_{ASC}$	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:-

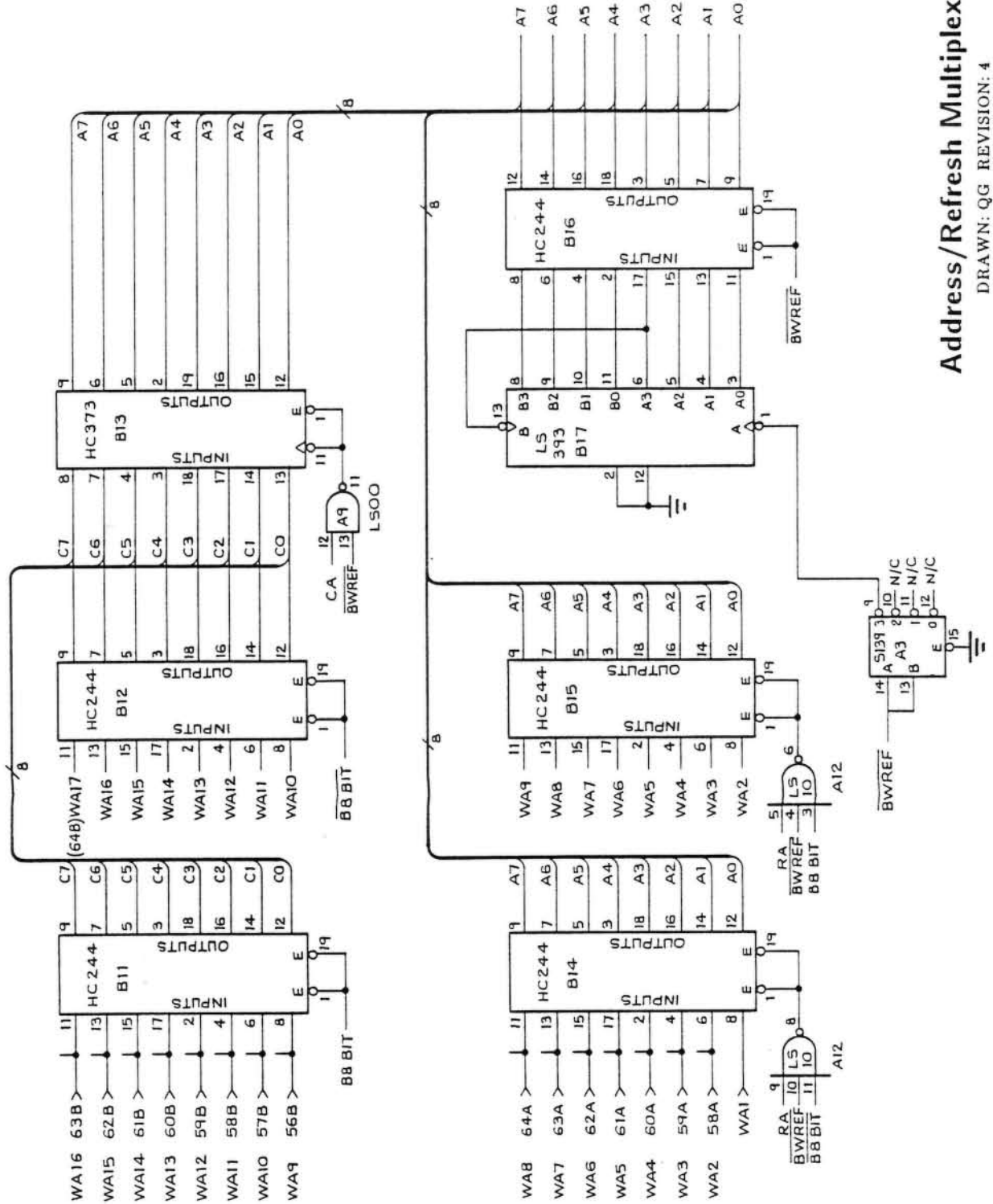
Buffered $\overline{SCLK}$ rising edge to $\overline{RAS}$ falling edge	120
Buffered $\overline{WAS}$ falling edge to $\overline{RAS}$ falling edge	100





**Timing Generation**  
DRAWN: QG REVISION: 4

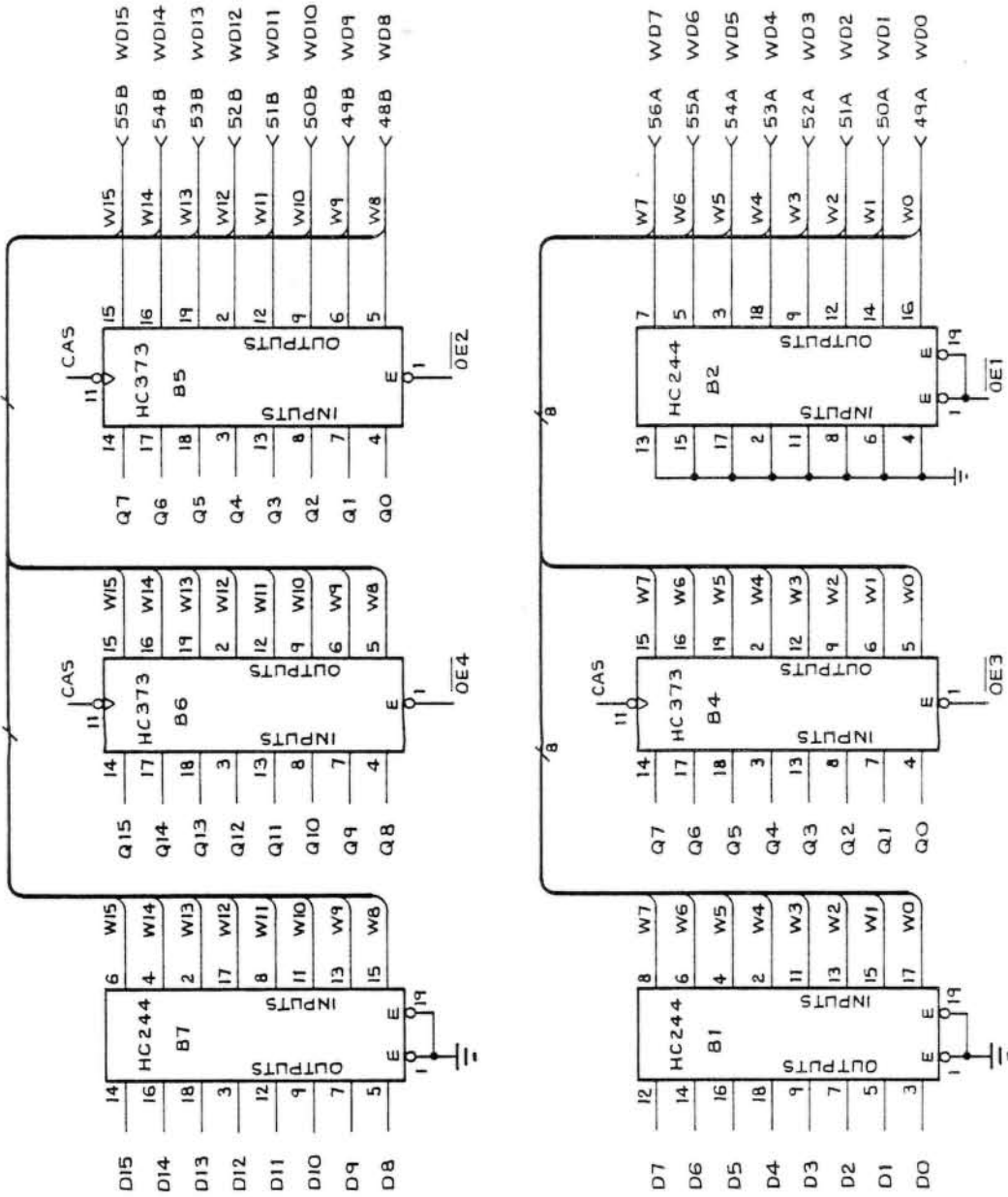




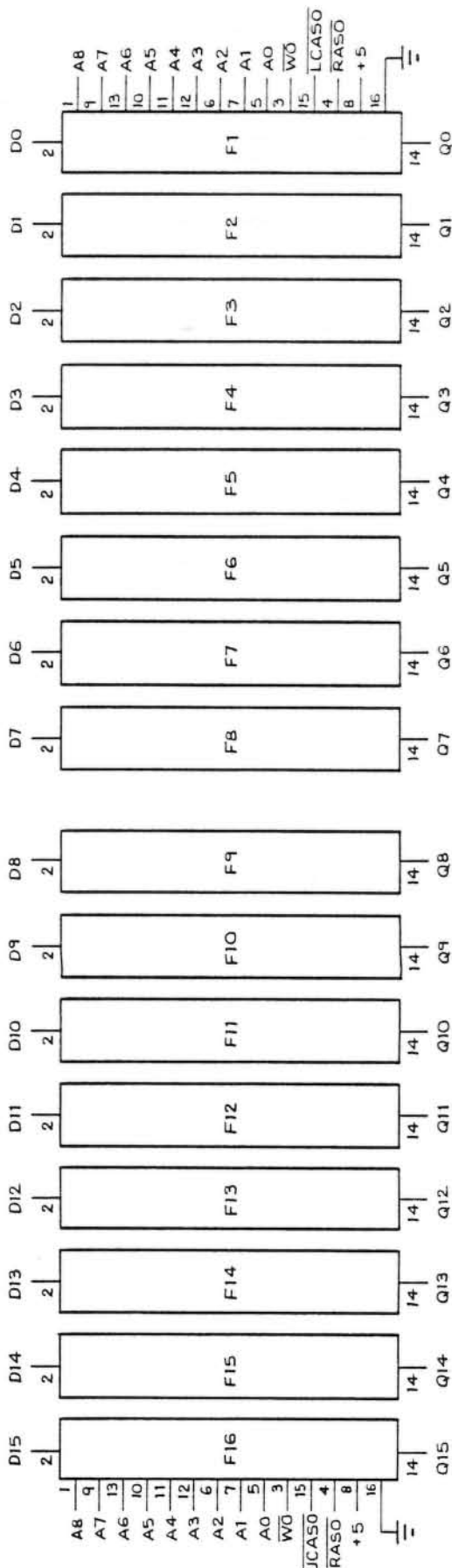
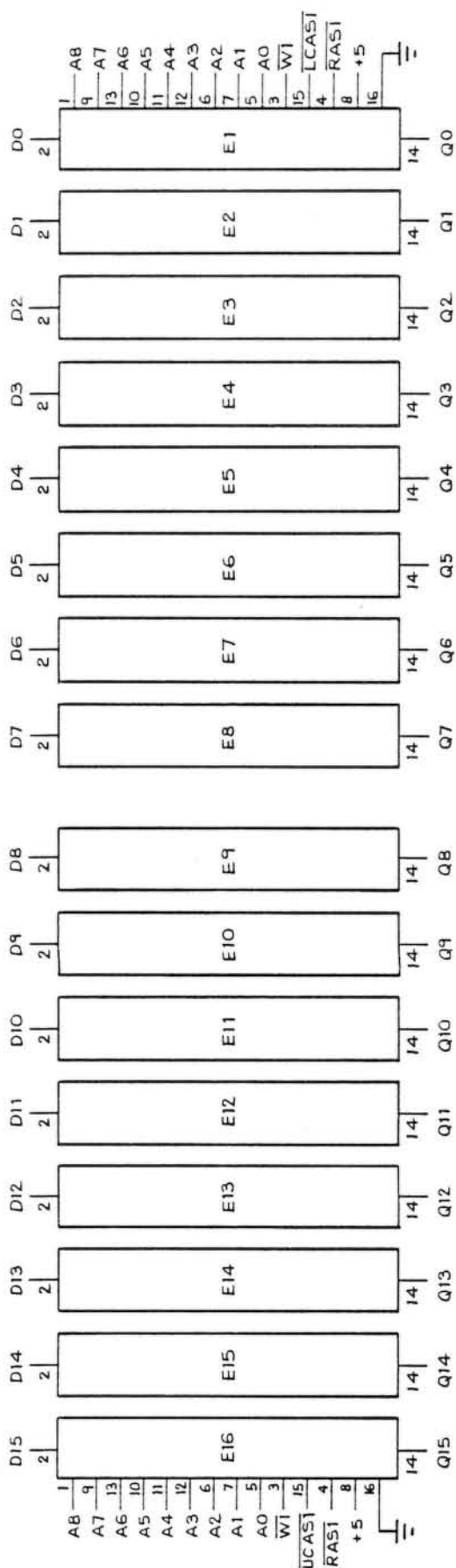
Address/Refresh Multiplexor

DRAWN: QG REVISION: 4

*twilight*



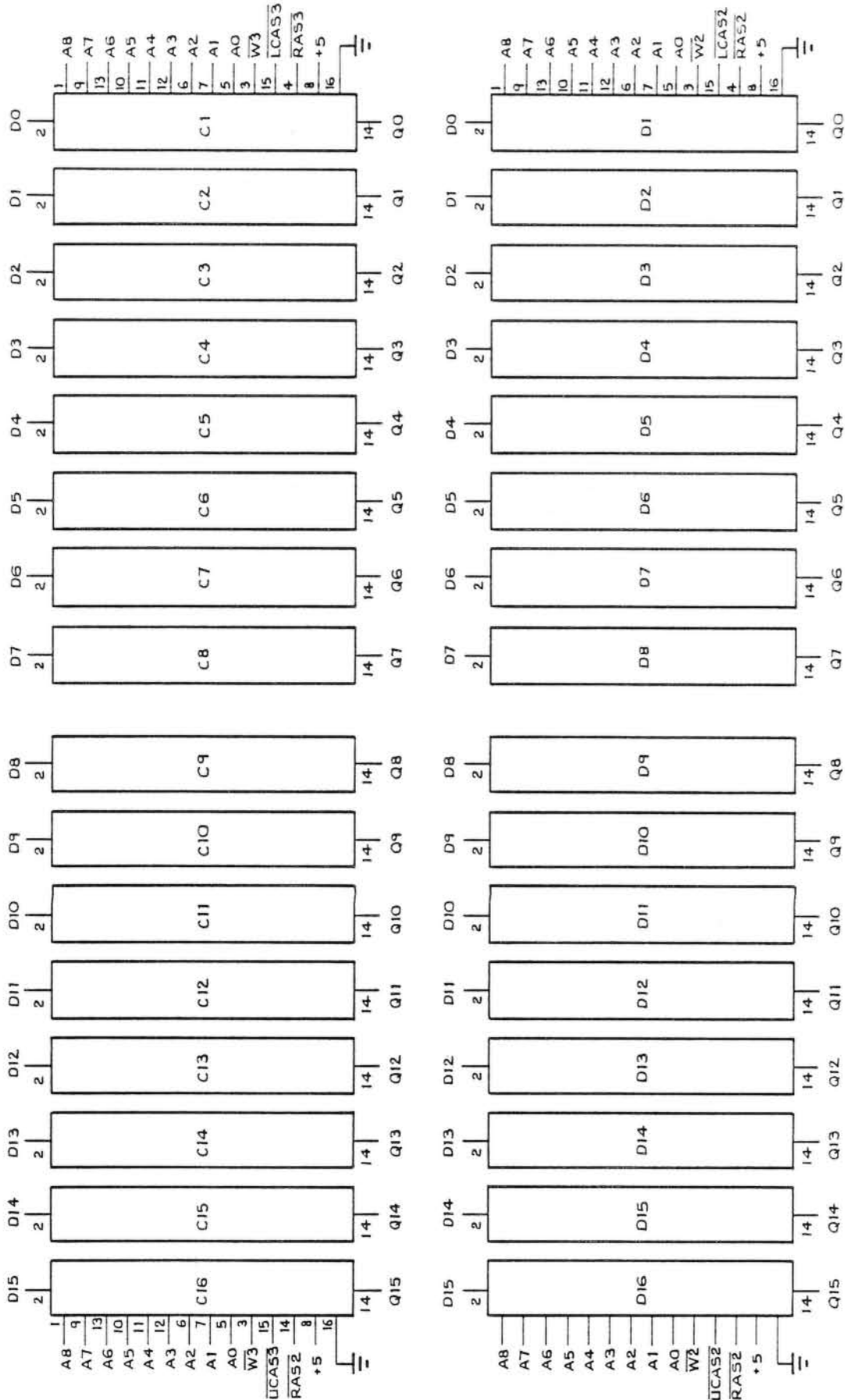
**Data Bus Interface**  
DRAWN: QG REVISION: 4



RAMS MUST BE 150 nS MAX.  
ACCESS TIME OR FASTER

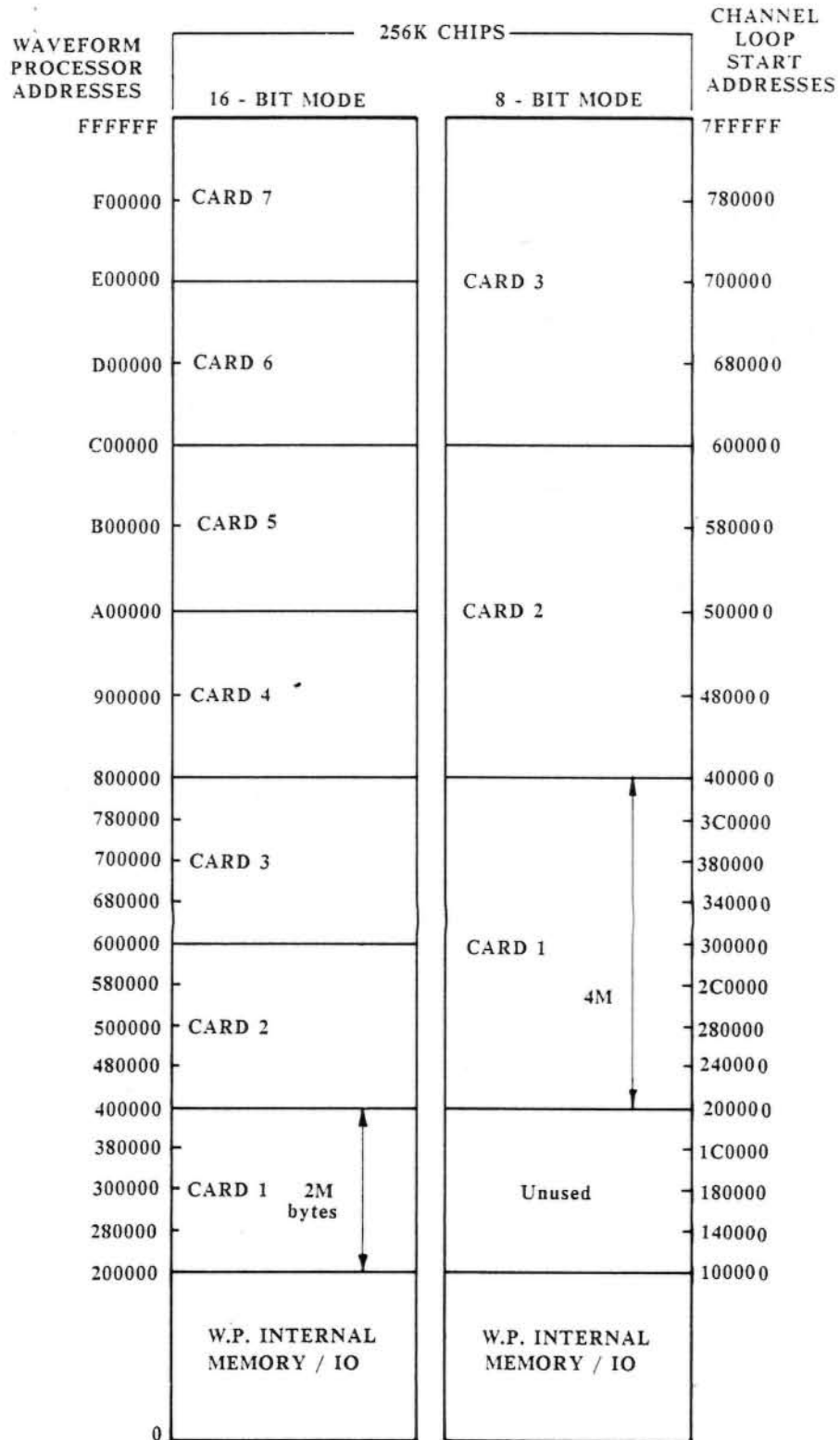
COMPATIBLE 256K RAMS :-  
50256 -15 (HITACHI)  
MN41256 (MATSUSHITA)  
6256 -15 (MOT.)  
41256 -15 (NEC)

**Dynamic RAMS**  
DRAWN: QG REVISION: 4



**Dynamic RAMS**  
DRAWN: QG REVISION: 4

CMI-39 Waveform Memory Map



### Waveform Ram Timing

