

CMI-28

MIDI PROCESSOR





FIELD CHANGE NOTICE

DATE 16/ 08/ 93
NUMBER 135

ORIGINATOR Chris Alfred

PRODUCT: CMI / MFX

ASSEMBLY No. CMI 28

DESCRIPTION MIDI PROCESSOR

This FCN applies to REV No: 5.0

New REV No is: 5.1

REASON FOR CHANGE:

MODIFICATION TO FIX ERRORS IN ORIGINAL REV 5 PCB

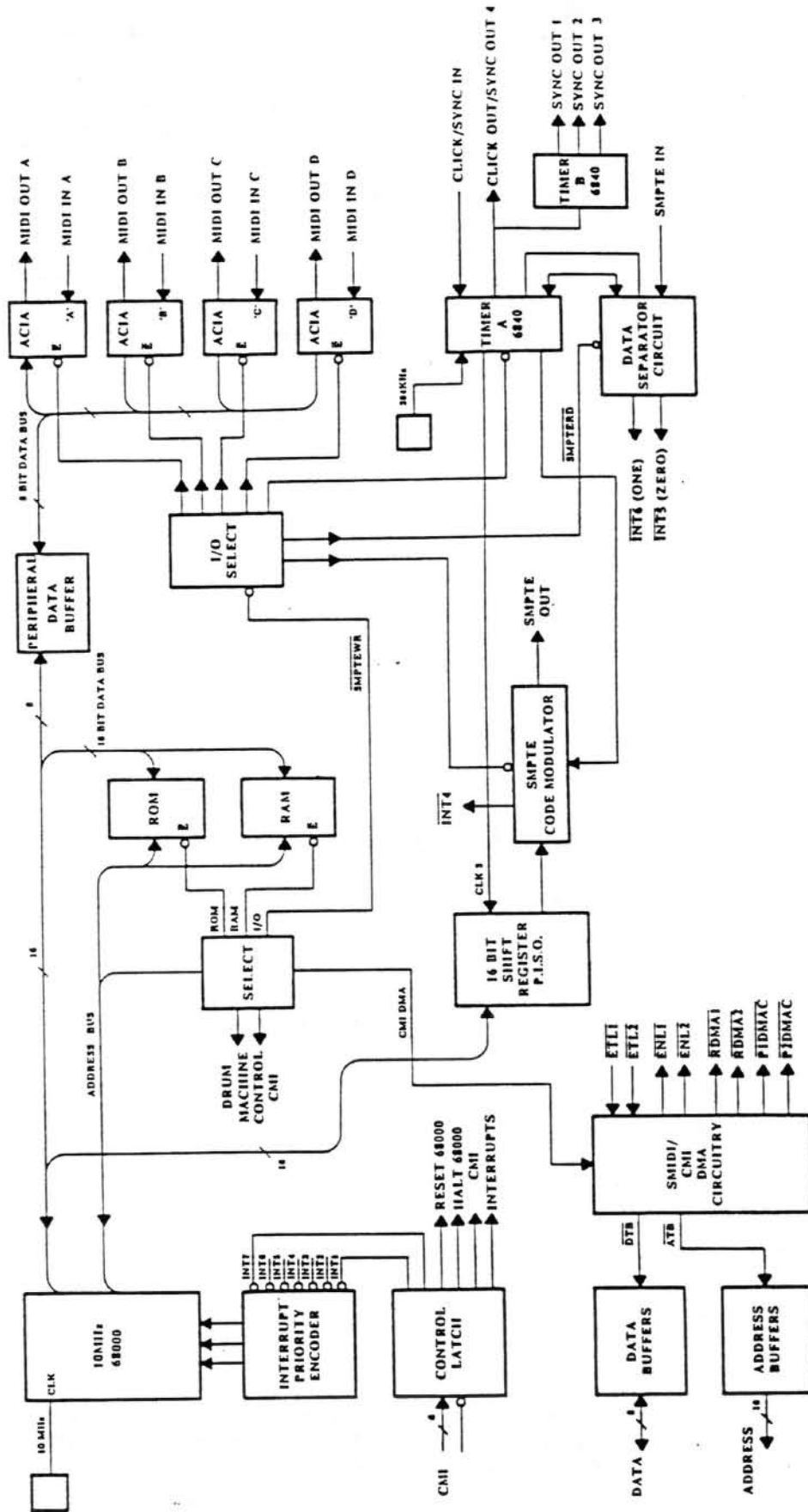
DETAILS OF CHANGE:

- (1) U62 IS 74LS244 (NOT 74LS240 AS ON SILK SCREEN)
- (2) CUT TRACK ON SOLDER SIDE BETWEEN CN1/8A AND NEARBY PLATE THRU (CN1 IS EDGE CONNECTOR)
- (3) CUT TRACK ON SOLDER SIDE BETWEEN CN1/9A AND NEARBY PLATE THRU
- (4) CONNECT PLATE THRU NEAR CN1/8A TO CN1/9A
- (5) CONNECT PLATE THRU NEAR CN1/9A TO CN1/8A
- (6) CUT SOLDER SIDETRACK TO PLATE THRU BETWEEN U30 PINS 4 AND 17 (72LS240)
- (7) CUT SOLDER SIDE TRACK TO PLATE THRU BETWEEN U30 PINS 6 AND 15
- (8) CONNECT U34/1 (74LS00) TO U30/17 (74LS240)
- (9) CONNECT U34/2 (74LS00) TO U30/15 (74LS240)
- (10) MARK PCB AS REV 5.1

ORIGINATOR:	DATE:	TEST:	DATE:
		PROD: <i>lgavin</i>	DATE: <i>19/07/93</i>
SERVICE:	DATE:	KIT LIST CHANGE:	YES NO

IMI-28 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 nnnnnh.

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/output) 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).

CMI-28 General Interface Card

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* (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 Programmable Timers

The SMIDI card has four ACIA's (68B50) for MIDI input and outputs. They can be accessed 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 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 D10b. 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 G1b 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.

CMI-28 General Interface Card

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 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 DTACK line. So that when RAM is selected 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 B2b 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 HALT and RESET must be asserted. The minimum reset period required by the 68K is 10 clock cycles i.e. 1 μ S.

The state of the HALT and RESET signals are indicated by the LEDs near the front of the SMIDI card. Both off means that the processor is running.

System Reset (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 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 IPL0-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 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 temporarily 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.

CMI-28 General Interface Card

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 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 C12a 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, P2 ϕ 2 is enabled through to become P ϕ 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 doc cycles.

Thus at the beginning of the data cycle of whichever processor is selected, the P ϕ 2 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. $\overline{ETL1}$, $\overline{ENL1}$ and $\overline{RDMA1}$ are the chain signals for P1, $\overline{ETL2}$, $\overline{ENL2}$, $\overline{RDMA2}$ are for P2. Which set is 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 buss. After the CMI signal has been latched, nothing happens until this signal is low, whereupon the \overline{RDMA} *Request DMA* is driven low through the selected LS10 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 68K requires a transfer \overline{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 $\overline{\text{ACK}}$ signal becomes PACK. When a request has been generated (C12a Q hi) and this level is enabled (ENL low), the rising edge of PACK clocks a low into flip flop B11a to generate $\overline{\text{DCYCLE}}$. 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 $\overline{\text{DMAC DMA Claim}}$ 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 $\overline{\text{DMAC}}$ signal is asserted during the data cycle preceding the actual transfer.

The address phase of the DMA cycle is indicated when $\overline{\text{ATB Address To Bus}}$ 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 $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$ 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 H12a at the beginning of $\overline{\text{DCYCLE}}$ 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 $\overline{\text{UDS}}$ and $\overline{\text{LDS}}$ 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 H12a. 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 $\overline{\text{DTB Data To Bus}}$ at the rising edge of BRA when a DMA cycle is in progress. This is performed by flip flop C12b. 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 $\overline{\text{CMI}}$ being asserted will drive the $\overline{\text{G}}$ input of the appropriate LS646 for the duration of the DMA cycle (LS02 and LS32 gates B8 and A8).

CMI-28 General Interface Card

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 \overline{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 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 \overline{UDS} or \overline{LDS} will be low. This will cause the LS10 A10 to output a high, and $\overline{DTACK2}$ will be generated as soon as \overline{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 \overline{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 CMI will still be asserted and \overline{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 \overline{LDATA} edge comes along the high on the LS10 output is clocked into the second C10a 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 \overline{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 CMI, then that the daisy chain signals are present. Then look for an 800nS pulse on \overline{DCYCLE} (pin 5 B11a), indicating that DMA cycles are actually occurring. 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 possibility 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 $\overline{DTACK3}$ 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{CS} signal. The first flip-flop F2a is clocked on the first falling edge of E with the IO select and the data strobe high (ie either \overline{LDS} or \overline{UDS} low). The \overline{Q} output of F2a is applied to the NAND gate (H1), asserting \overline{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{Q} output of F2b is clocked low asserting $\overline{DTACK3}$ and latching data in the transceiver (G4) The asserted $\overline{DTACK3}$ signal deselects the peripheral by causing \overline{CS} to go high. Flip-flop F2a is cleared by IO going low when the access terminates. Clearing flip-flop F2a 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, 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 conjunction 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 SYNC SW line is low (from the Control Latch (B4)) the signal coming

CMI-28 General Interface Card

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 SYNC SW 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, ($\overline{\text{INT3}}$).
The Timers can interrupt the 68K on interrupt level 2, (INT2).

SMPTE Generating Circuitry

(refer schematic CMI-28-7)

An oscillator (3.84MHz) is divided by 10 (G2,H1,G1a) 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 (C11b) 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 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 C11a is toggled at the rate determined by CLK2, but if a *one* is shifted out from C8 the flip-flop C11a is toggled at the CLK1 rate. Thus, the word stored on the shift registers is output in SMPTE form.

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

$\overline{\text{INT6}}$ and $\overline{\text{INT5}}$ are cleared by the $\overline{\text{SMPTERD}}$ 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)

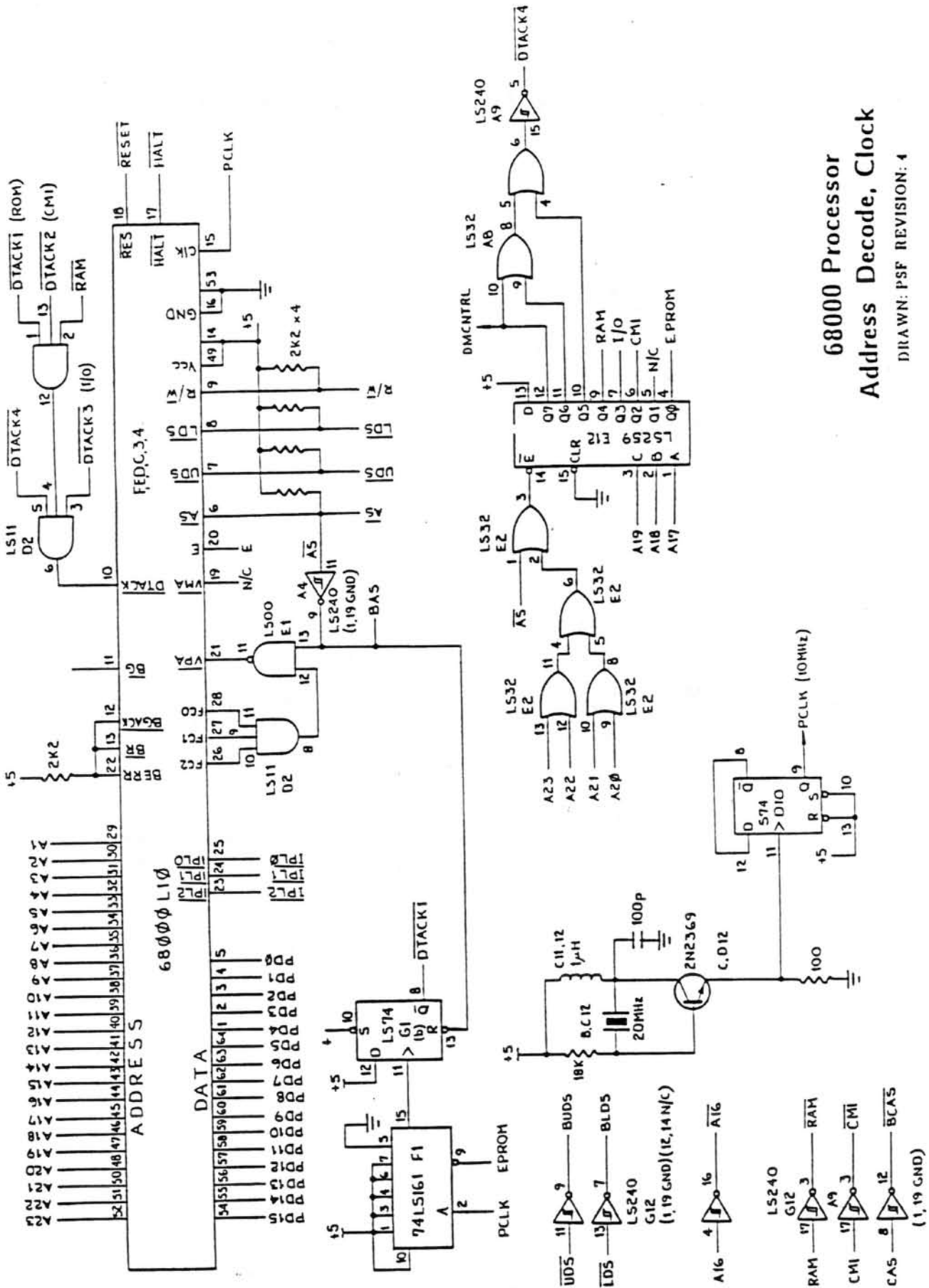
Pin 1	MIDI out A.	
Pin 2	+5 volts.	
Pin 3	MIDI in A.	
Pin 4	SYNC out 1.	
Pin 5	MIDI out B.	
Pin 6	SYNC out 2.	
Pin 7	MIDI in B.	
Pin 8	SYNC out 3.	
Pin 9	MIDI out C.	
Pin 10	Digital Ground.	
Pin 11	MIDI in C.	
Pin 12	Digital Ground.	
Pin 13	MIDI out D.	
Pin 14	RESET/START.	
Pin 15	MIDI in D.	
Pin 16	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.

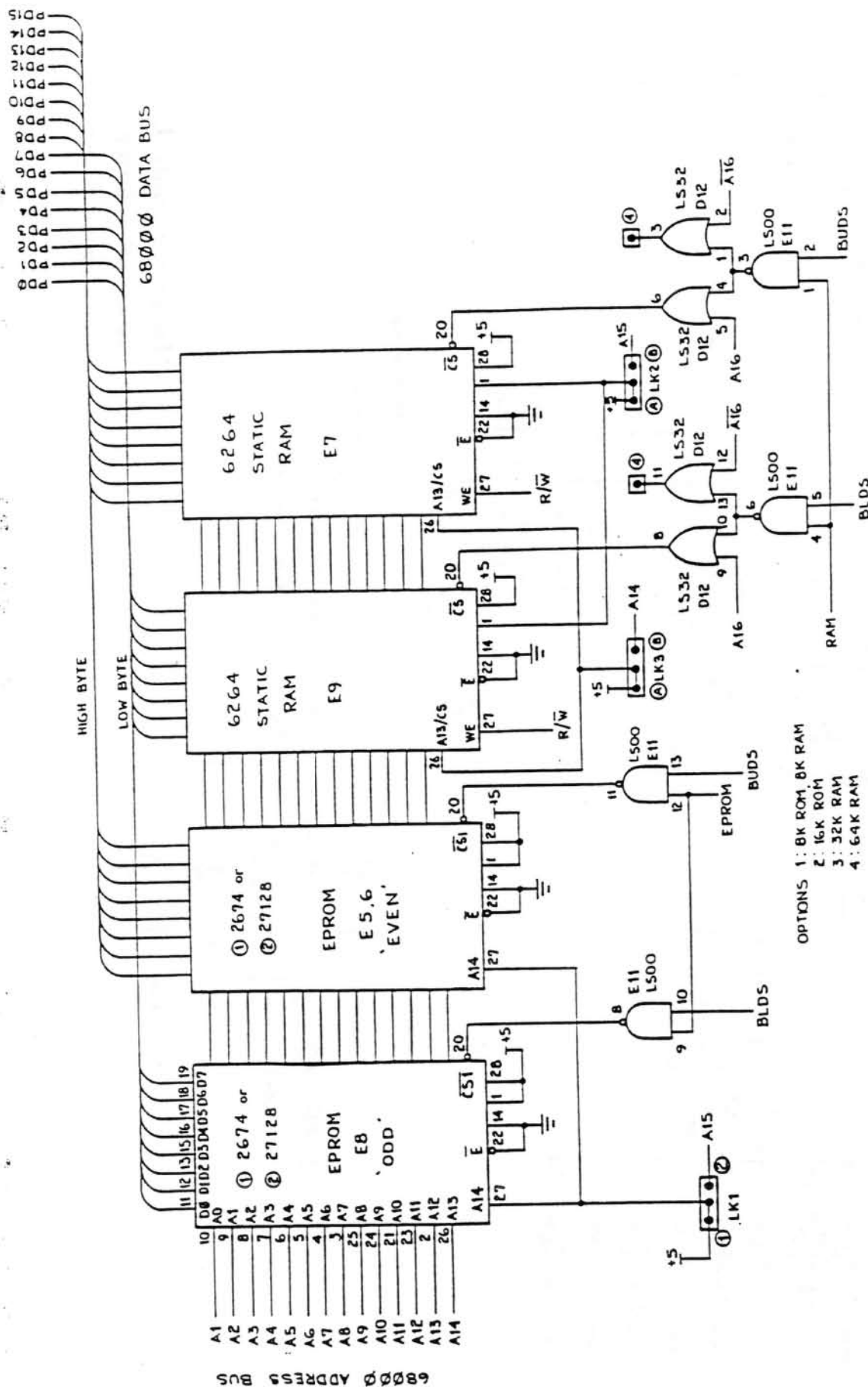
* - these connections (from the CMI28 board only) are for debugging purposes only. If two push-button switches are connected between pins 23 & 24 and pins 25 & 26, they can be used to manually halt and reset the 68K processor, respectively.

CMI 28-1 General Interface Card



**68000 Processor
Address Decode, Clock**

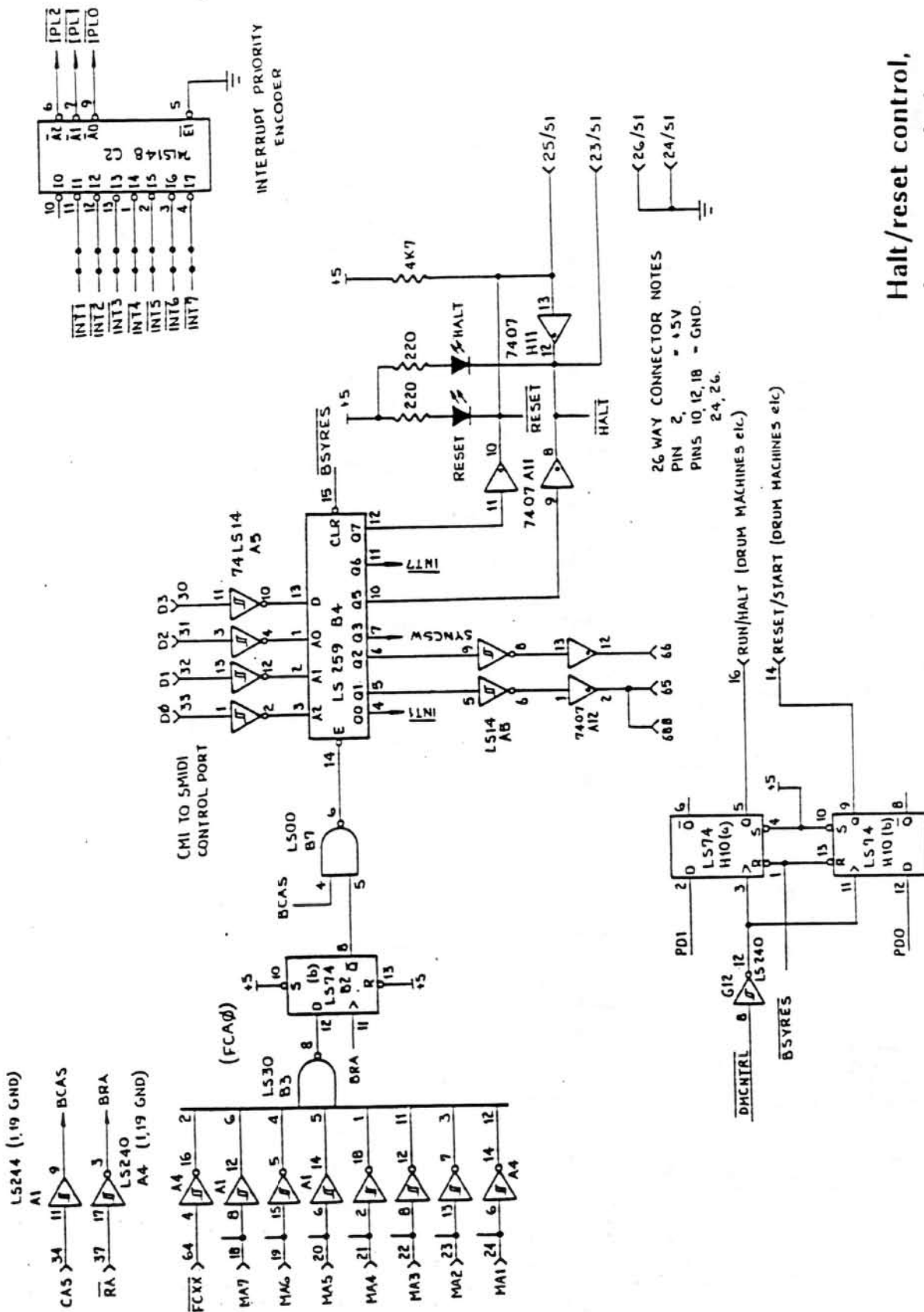
DRAWN: PSF REVISION: 4



Memory

DRAWN: PSF REVISION: 4

fairlight

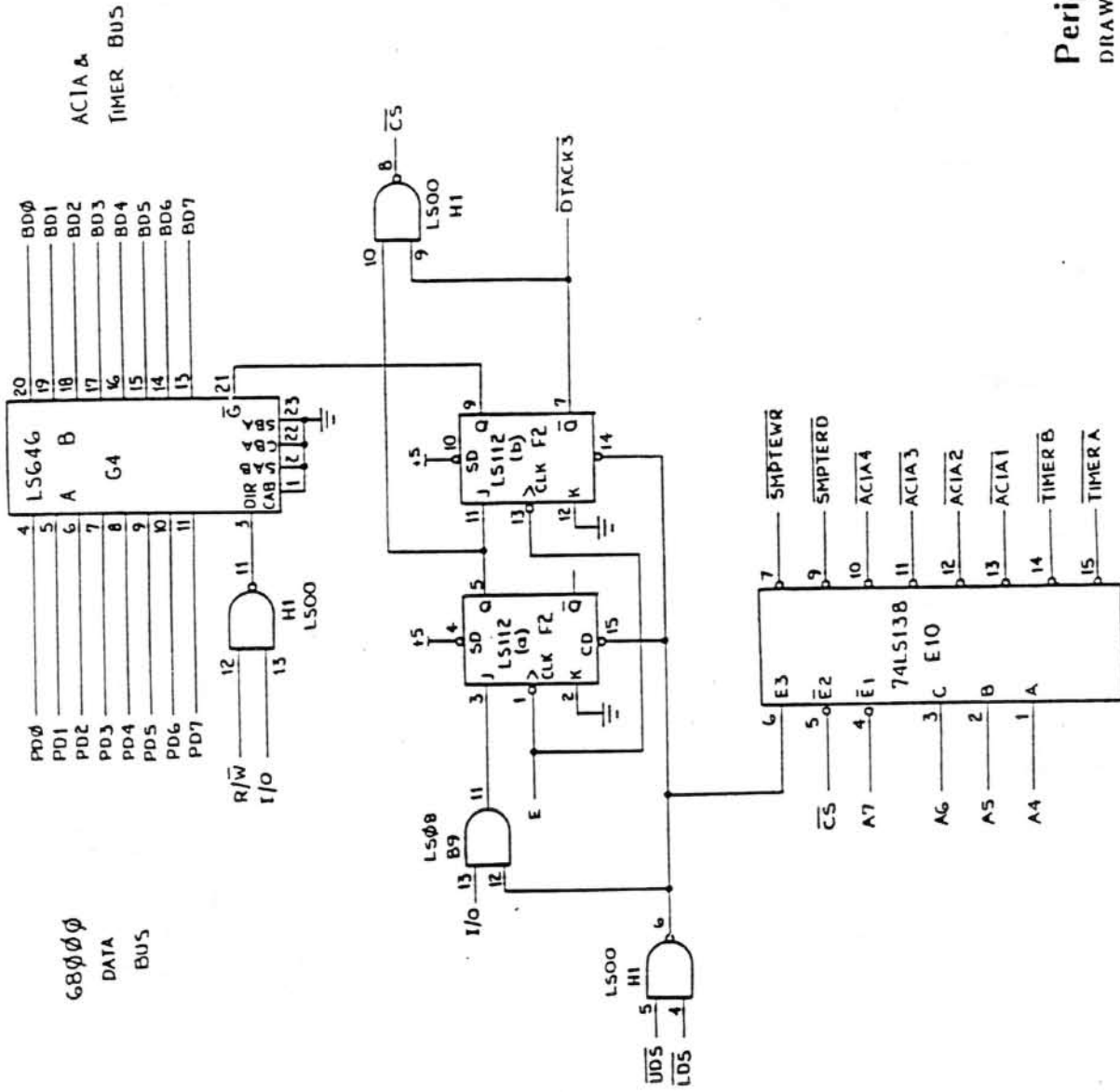


Halt/reset control,
Interrupt priority

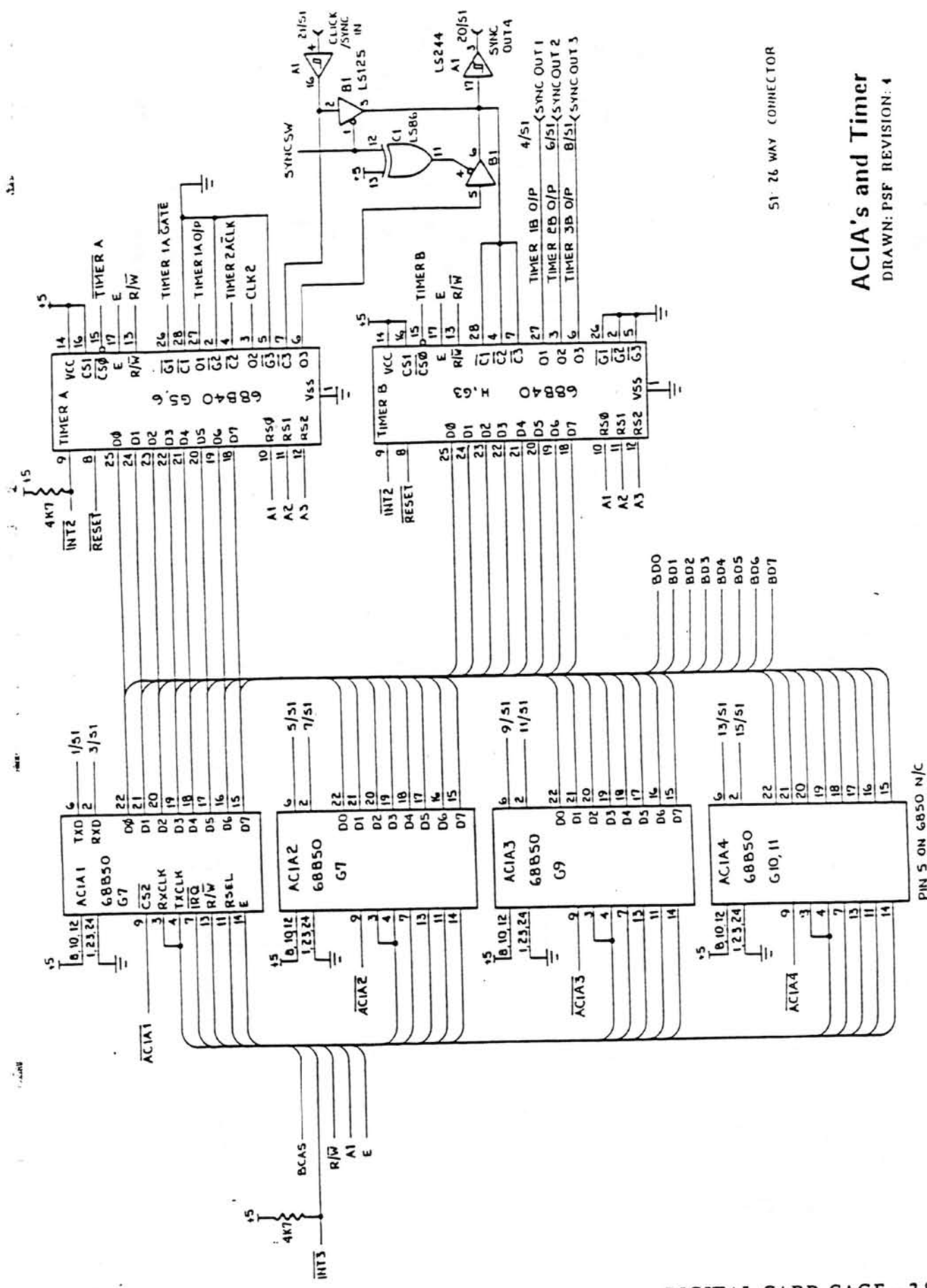
DRAWN: PSF REVISION: 4

26 WAY CONNECTOR NOTES
PIN 2, - +5V
PINS 10, 12, 18 - GND.
24, 26.

CMI 28-5 General Interface Card



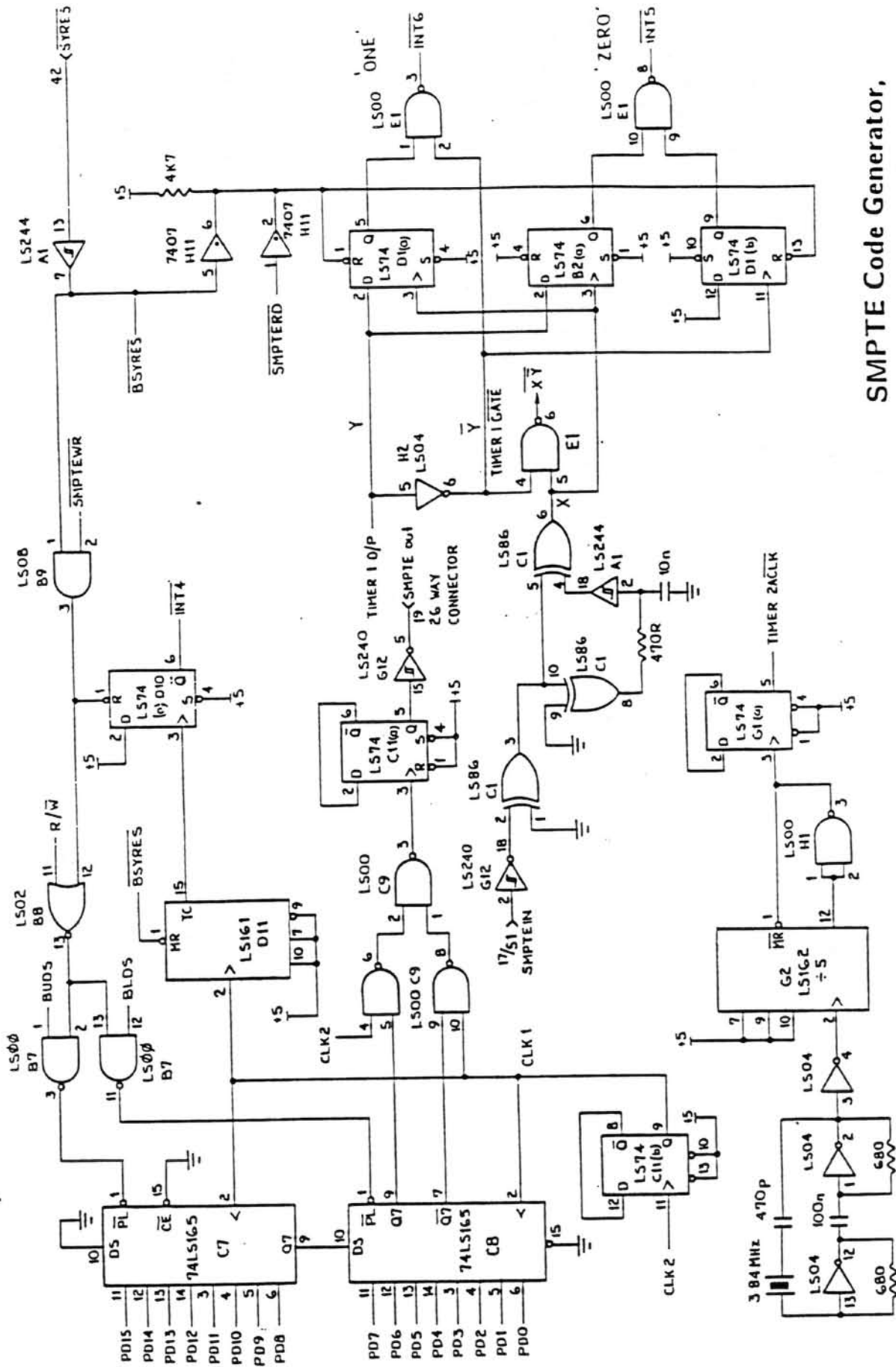
Peripheral Select
DRAWN: PSF REVISION: 4



51-26 WAY CONNECTOR

ACIA's and Timer
DRAWN: PSF REVISION: 4

PIN 5 ON 6850 N/C

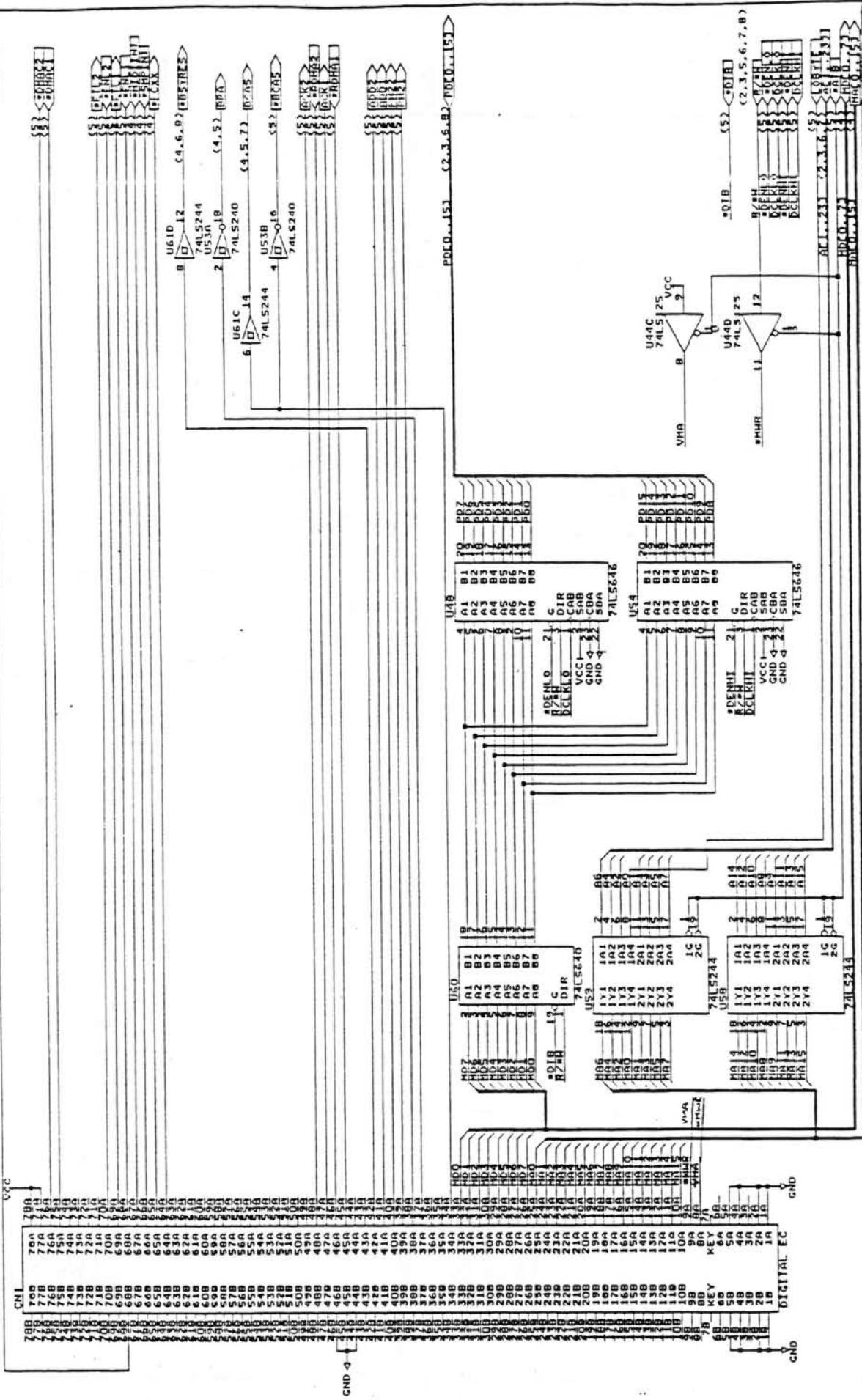


**SMPTE Code Generator,
Data Separator and Reader**

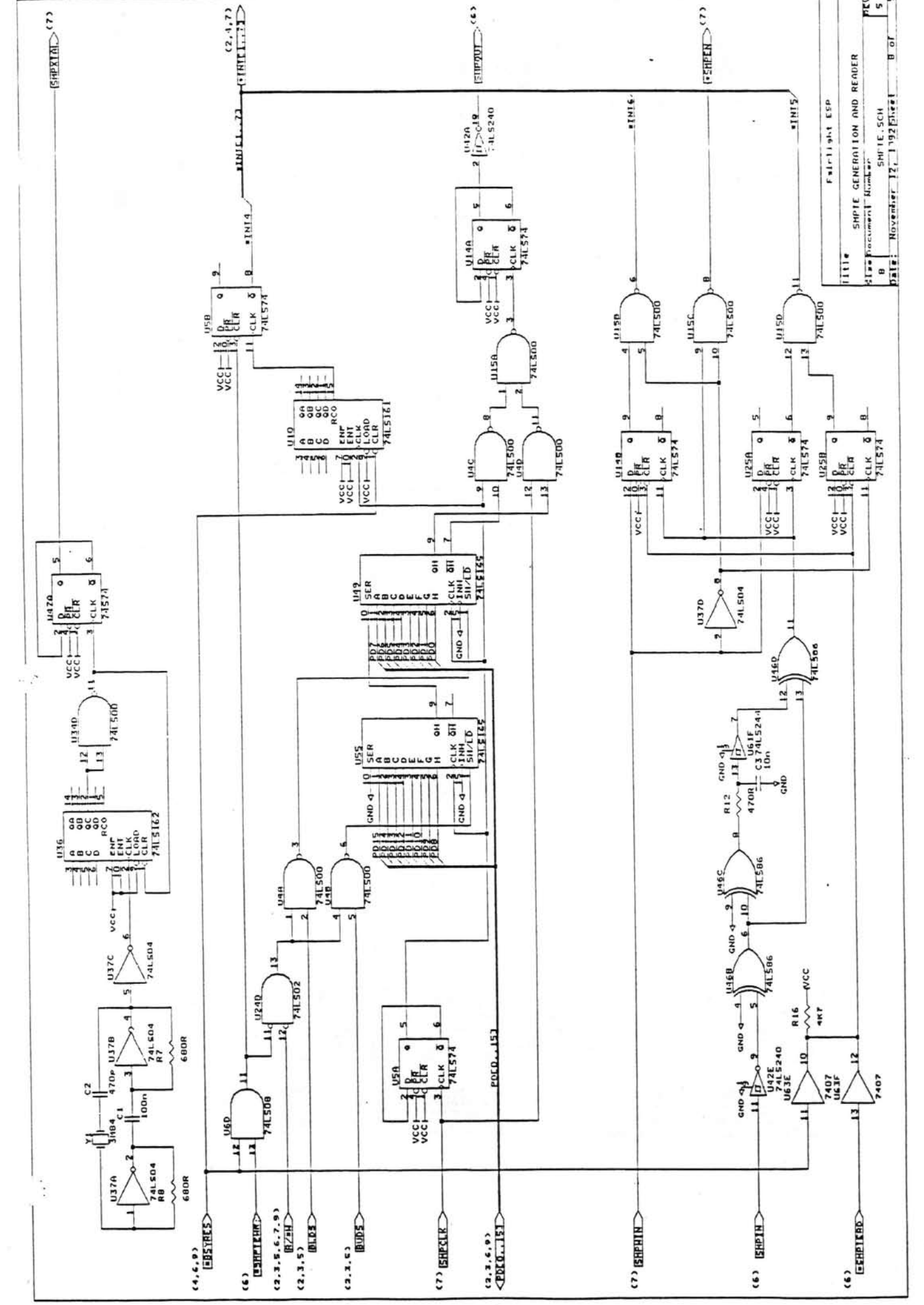
DRAWN: PSF REVISION: 4



(4) PRINT



Title: Fairlight ESP
 Edge Connector: B
 Size/Document Number: B
 Date: November 12, 1992
 Sheet: 9 of 9



SHIFTIN (7)

INITIAL (2,4,7)

OUTPUT (6)

SHIFTIN (7)

Title: Shift Register and Reader
 Assignment Number: SHPTE.SCH
 Date: November 12, 1992
 Page: 5 of 5

(4,6,9)

(6)

(2,3,5,6,7,9)

(2,3,5)

(2,3,5)

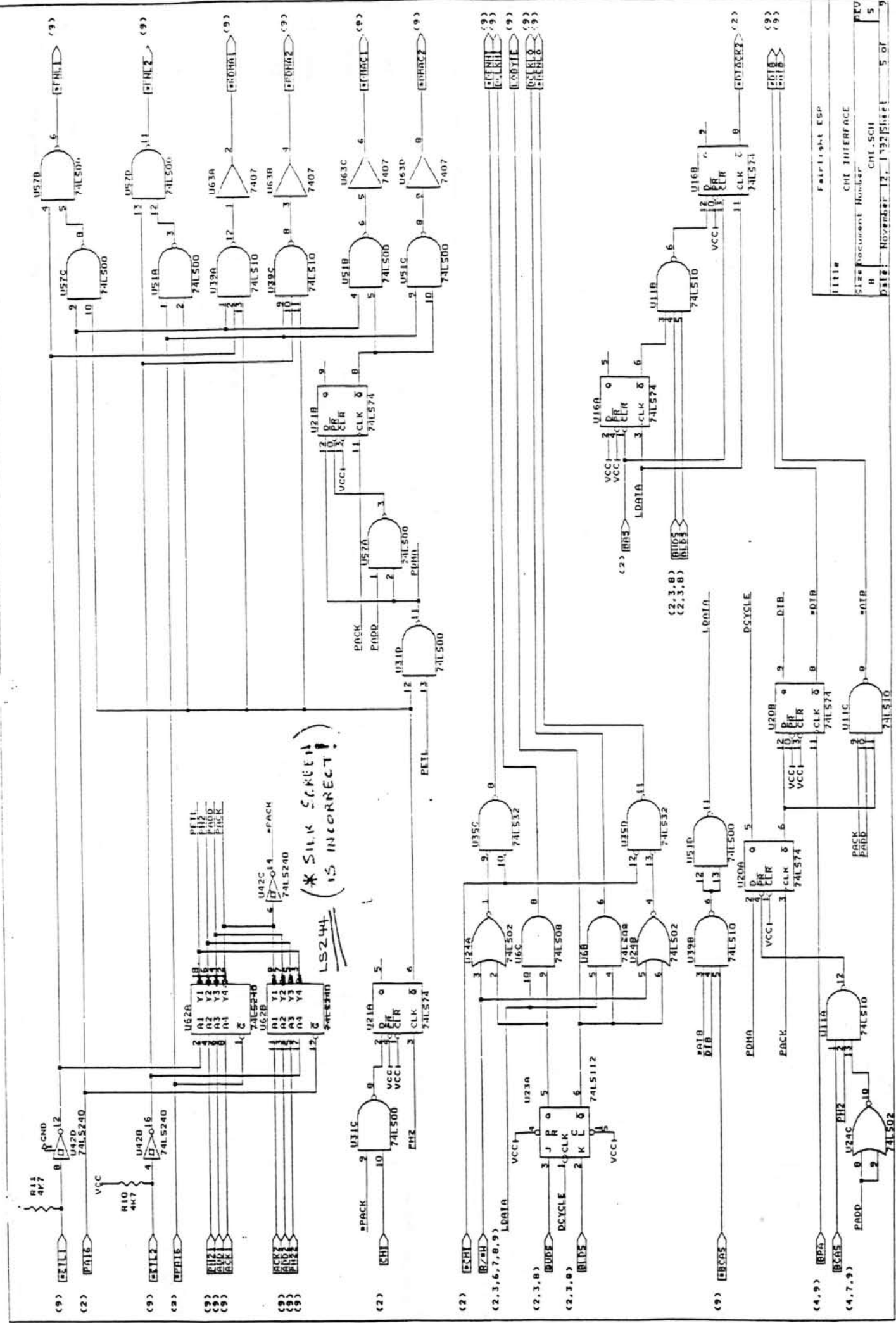
(7)

(2,3,6,9)

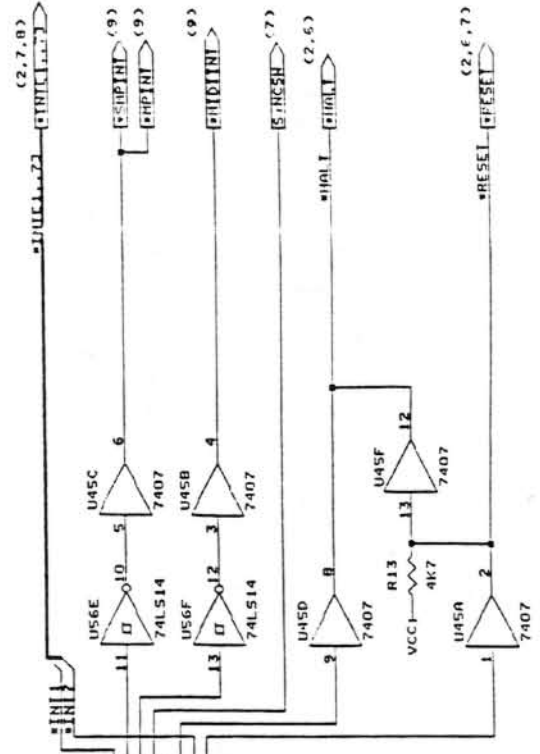
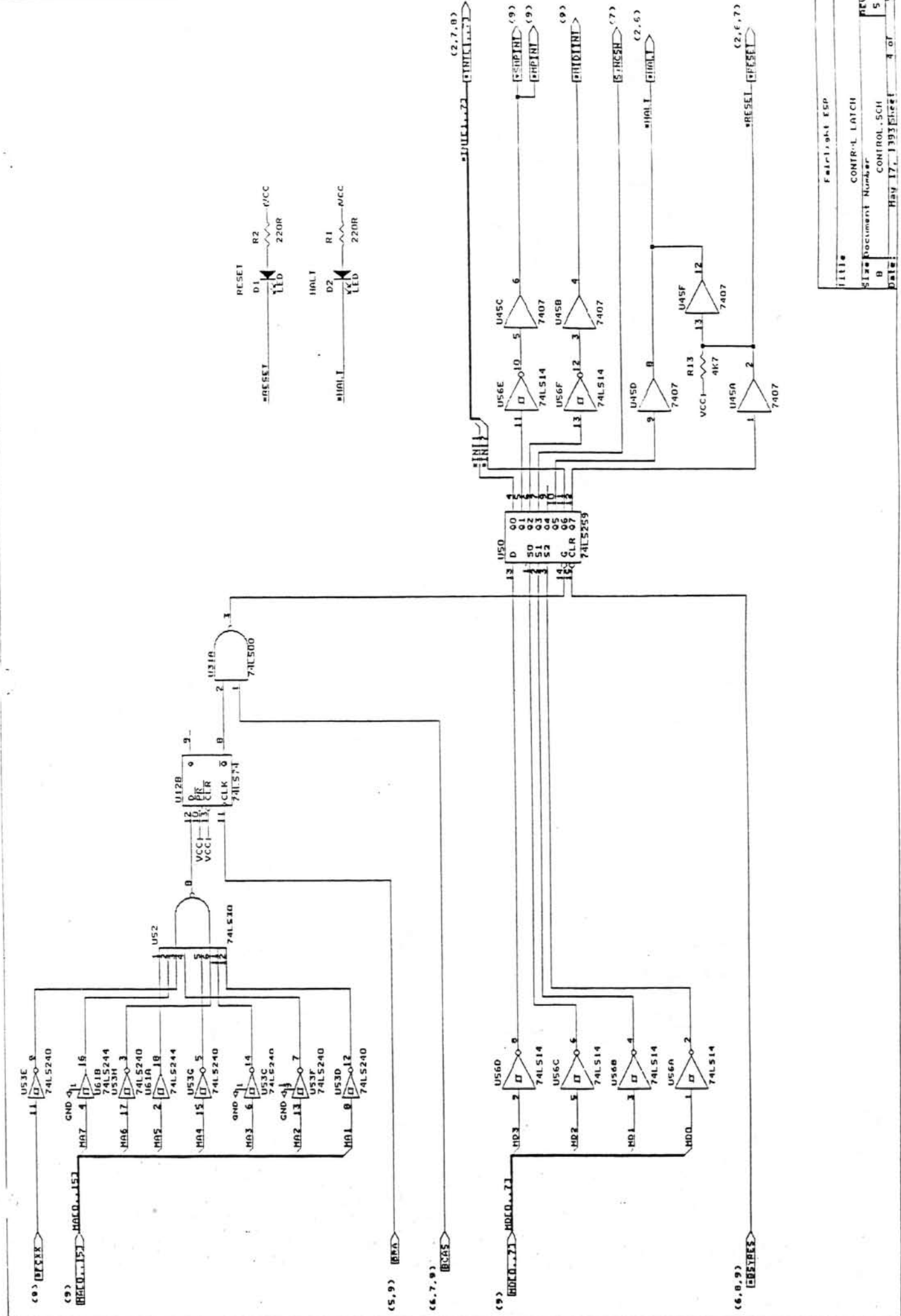
(7)

(6)

(6)

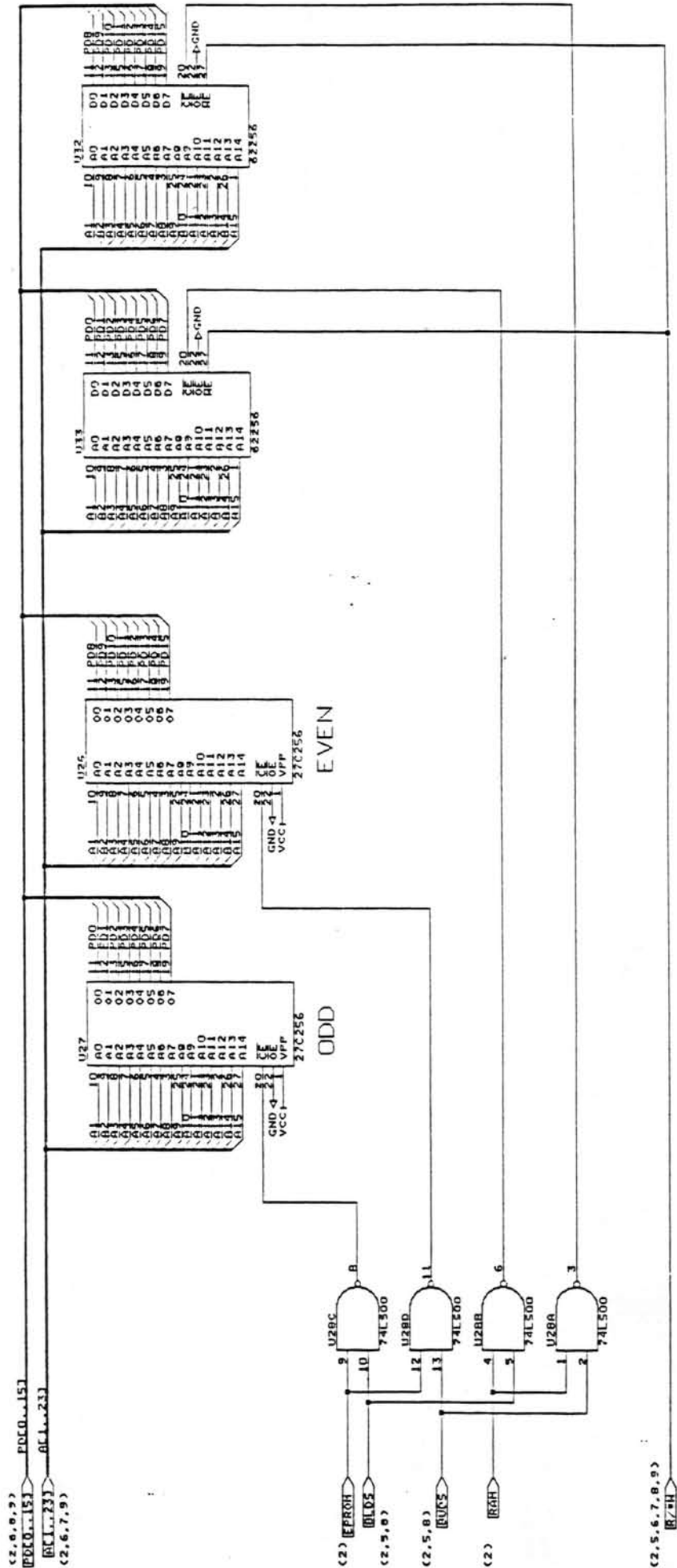


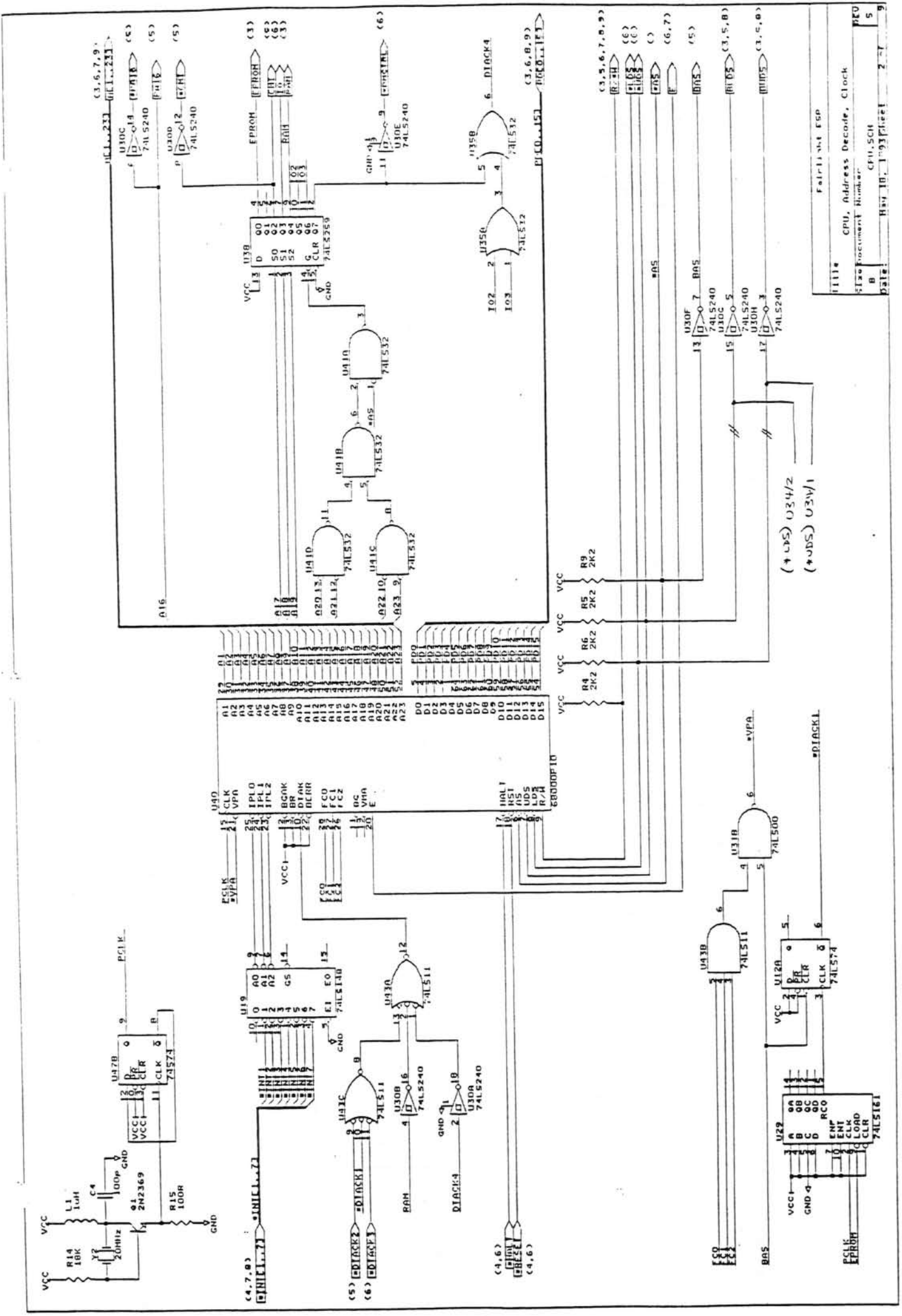
*** SILK SCREEN (IS INCORRECT!)**
LS244



Title	Fail-safe ESP
Size	CONTROL LATCH
Sheet	B
Document Number	CONTROL.SCH
Date	May 17, 1993
Page	5 of 9

(6,0) [XXXX] (5,9) [RKA] (6,7,8) [BC55] (9) [HD0..7] [HD0..7] (6,8,9) [RSTRES]



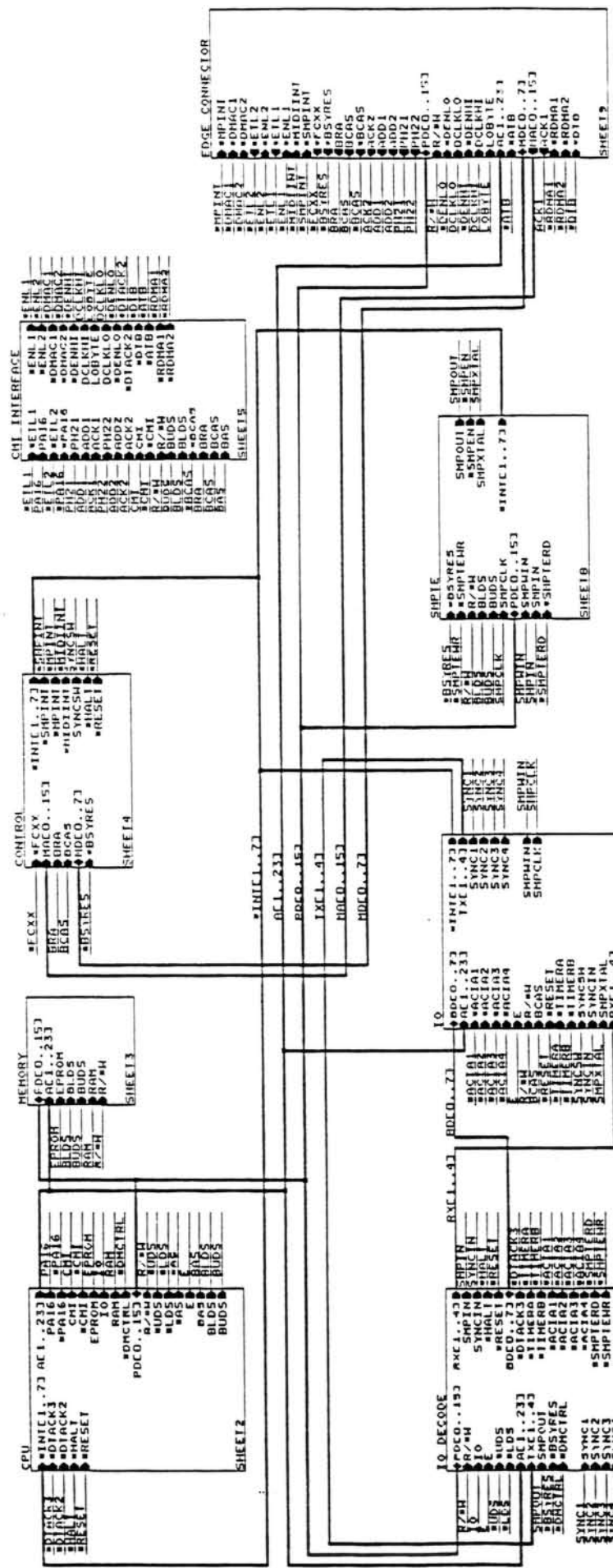


Rev	5
Doc No	CP115CH
Date	May 18, 1993
Sheet	2 of 9

Fairchild ESD
 CPU, Address Decode, Clock
 Part Number CP115CH
 Rev B
 Date May 18, 1993

U100	74LS240
U101	74LS240
U102	74LS240
U103	74LS240
U104	74LS240
U105	74LS240
U106	74LS240
U107	74LS240
U108	74LS240
U109	74LS240
U110	74LS240
U111	74LS240
U112	74LS240
U113	74LS240
U114	74LS240
U115	74LS240
U116	74LS240
U117	74LS240
U118	74LS240
U119	74LS240
U120	74LS240
U121	74LS240
U122	74LS240
U123	74LS240
U124	74LS240
U125	74LS240
U126	74LS240
U127	74LS240
U128	74LS240
U129	74LS240
U130	74LS240
U131	74LS240
U132	74LS240
U133	74LS240
U134	74LS240
U135	74LS240
U136	74LS240
U137	74LS240
U138	74LS240
U139	74LS240
U140	74LS240
U141	74LS240
U142	74LS240
U143	74LS240
U144	74LS240
U145	74LS240
U146	74LS240
U147	74LS240
U148	74LS240
U149	74LS240
U150	74LS240
U151	74LS240
U152	74LS240
U153	74LS240
U154	74LS240
U155	74LS240
U156	74LS240
U157	74LS240
U158	74LS240
U159	74LS240
U160	74LS240
U161	74LS240
U162	74LS240
U163	74LS240
U164	74LS240
U165	74LS240
U166	74LS240
U167	74LS240
U168	74LS240
U169	74LS240
U170	74LS240
U171	74LS240
U172	74LS240
U173	74LS240
U174	74LS240
U175	74LS240
U176	74LS240
U177	74LS240
U178	74LS240
U179	74LS240
U180	74LS240
U181	74LS240
U182	74LS240
U183	74LS240
U184	74LS240
U185	74LS240
U186	74LS240
U187	74LS240
U188	74LS240
U189	74LS240
U190	74LS240
U191	74LS240
U192	74LS240
U193	74LS240
U194	74LS240
U195	74LS240
U196	74LS240
U197	74LS240
U198	74LS240
U199	74LS240
U200	74LS240

P



* Rev S.1 18-5-93

