CPU 286
TECHNICAL MANUAL
451A

CPU 286
TECHNICAL MANUAL

**HIGH–PERFORMANCE 16–BIT 80286
80287 NUMERIC PROCESSOR EXTENSION
EPROM/SRAM/EEPROM SOCKETS
FOR UP TO 64K**

CPU 286
TECHNICAL MANUAL
Copyright 1986 Viasyn Corporation
Hayward, CA 94545

Document No. 8261-0018
Filename: CPU286F.MAN
ASY- 451A

Contents

Tables

Specifications

Timing	Meets all IEEE 696/S100 specifications.
Clock Rate	8MHz. Can support up to a 12.5MHz 80286.
Address Bus	24 Bits; addresses 16 megabytes.
Memory Manager	A23-A20 programmable.
Data Bus Width	16-bit memory or I/O; also supports 8-bit devices.
Wait State Generator	Flexible; can insert 0 to 3 wait states into I/O or memory cycles independently.
Master Status	Implemented as a permanent master.
On-board Memory	Can support up to 64K of EPROM, SRAM, or EEPROM.
Math Co-processor	80287. Clock speed is independent of 80286.
Power Consumption	2500mA maximum at 8VDC.

About the CPU 286

The CPU 286 from CompuPro is one of the most advanced 16-bit processors available for the IEEE 696/S-100 Bus. Based on the high performance 80286 16-bit microprocessor, the CPU 286 includes sockets for optional on-board EPROM, SRAM, and/or EEPROM memory and for Intel's 80287 High Speed Numeric Processor Extension.

Features of the CPU 286 include:

- The 80286 16-bit processor with an integrated memory management unit, virtual memory support, and an instruction set optimized for multi-user operation.

- A hardware byte serializer that allows mixing of any 8-bit and 16-bit memory and I/O devices that conform to the IEEE 696/S-100 protocol for 8-bit and 16-bit transfers; the CPU 286 dynamically adjusts itself to the proper bus width for 8-bit or 16-bit operation.

- The 80287 Numeric Processor Extension with an independent clock generator, adding fast number crunching capability while appearing to the software just like an 8087.

- Compatiblity with all TMA devices conforming to the IEEE 696/S-100 specification.

In the protected virtual address mode, the 80286 has integrated memory management and four-level memory protection for operating systems employing virtual memory. The address space in the protected virtual address mode extends to 16 Megabytes of physical addressing (24 bits), and a full Gigabyte (30 bits) of virtual addressing per task.

In the real address mode, the 80286 runs all software written for the 8086/8088. In addition, the on-board memory manager gives 24 address lines for access to up to 16 megabytes of memory.

Designed to work at clock speeds up to 12.5 MHz, the CPU 286 may always be used with the fastest processor available.

When you couple high speed operation with the power of the 80286/80287 pair, the CPU 286 is truly a processor board for advanced computing systems of the eighties. Thank you for choosing a CompuPro product.

1

Installing the CPU 286

Step 1. Unpack the CPU 286 Board.

Along with the board, you will find two card extractors in the plastic bag.

CARD EXTRACTOR
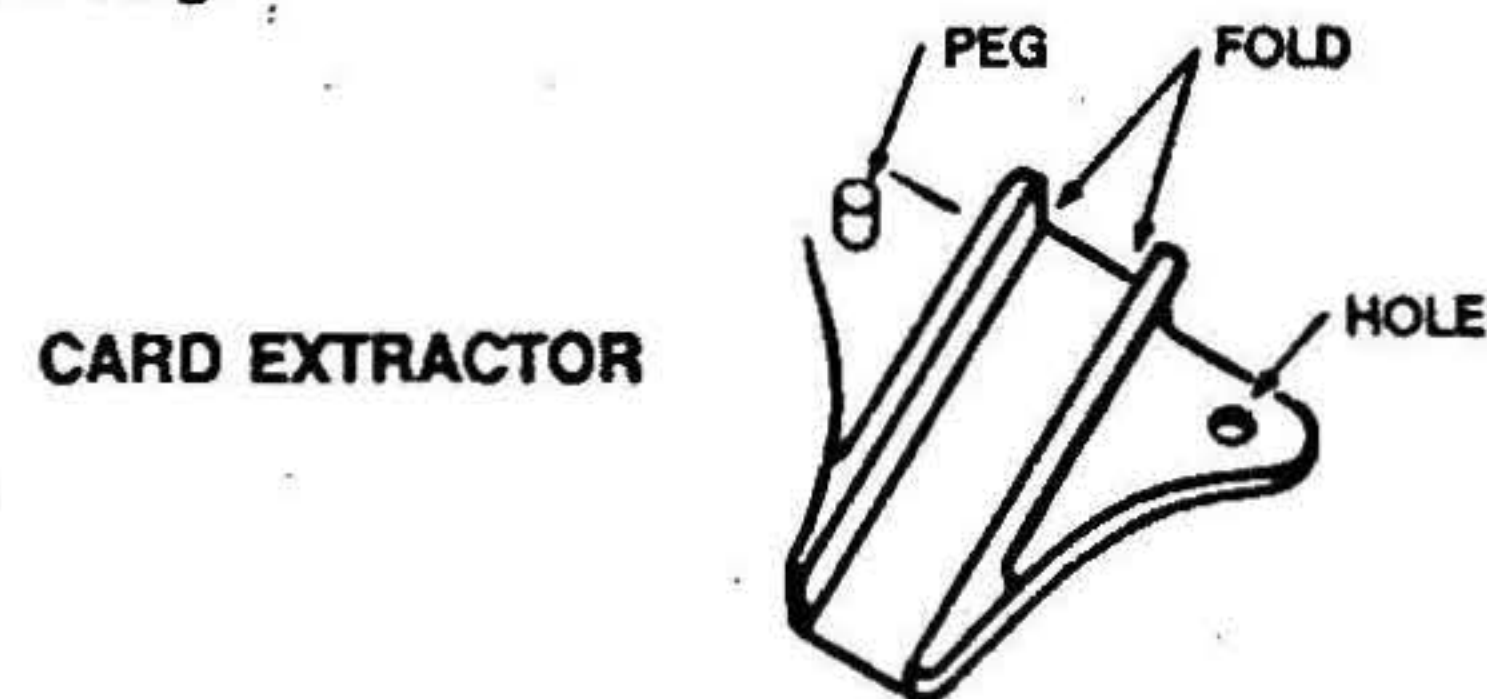
Step 2. Check Jumper Settings.

For the standard jumper settings to use with Concurrent DOS 8-16^{tm}, refer to the *Concurrent DOS 8-16 Installation and Customization Guide*.

See Step 3 if you need to change jumper settings. Otherwise proceed to Step 4.

Step 3. How to Install Jumper Shunt Connectors

notch

A jumper shunt is a small plastic part used to connect two pins on the jumper connector. Jumper shunts should be installed notch side up.

IF: The board is not correctly jumpered.

THEN: Use a pair of needle nose pliers to gently remove, and carefully replace the jumper shunt in its proper location.

jumper shunt
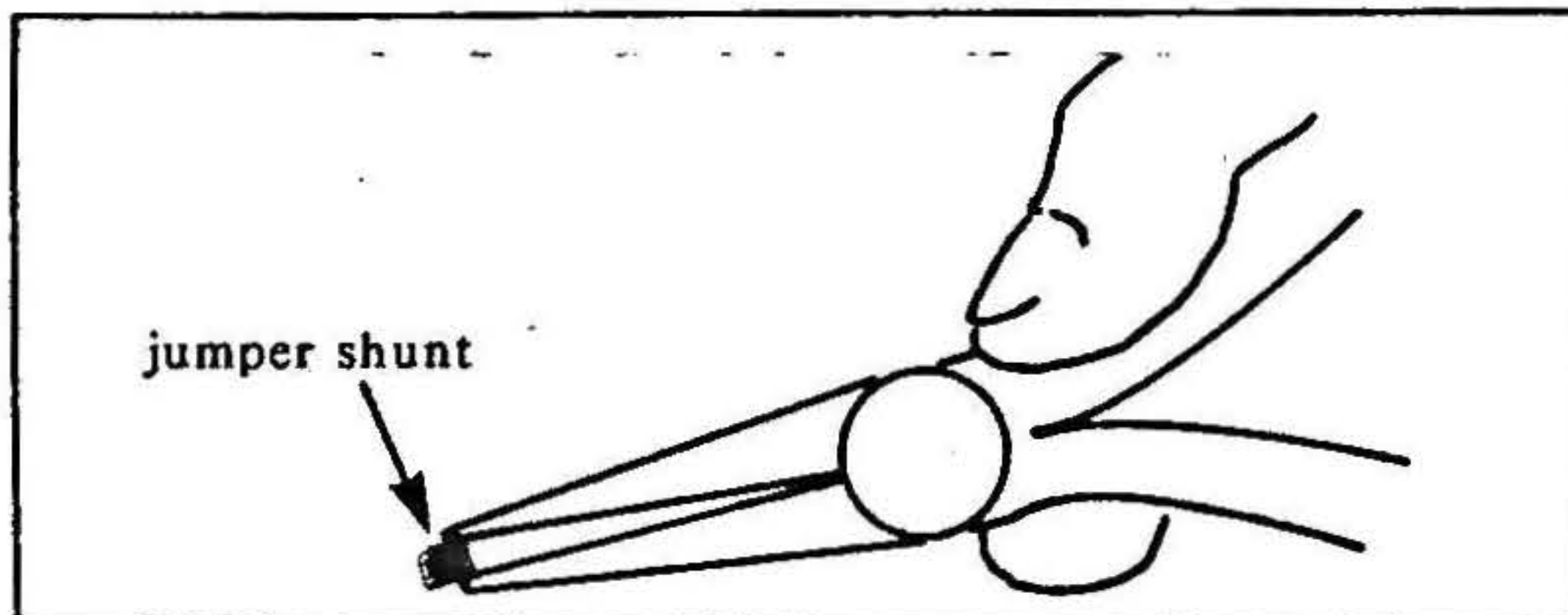
Step 4. Install the Card Extractors.

1. Hold the board so the component side is toward you. (See diagram below.)

2. Insert the peg on the card extractor into the hole in the <u>right</u> corner of the board. Fold the extractor over the board's edge until the extractor's hole snaps over the peg.

   NOTE: Make sure the long edge of the extractor is along the top edge of the board.

3. Repeat for the <u>left</u> extractor.
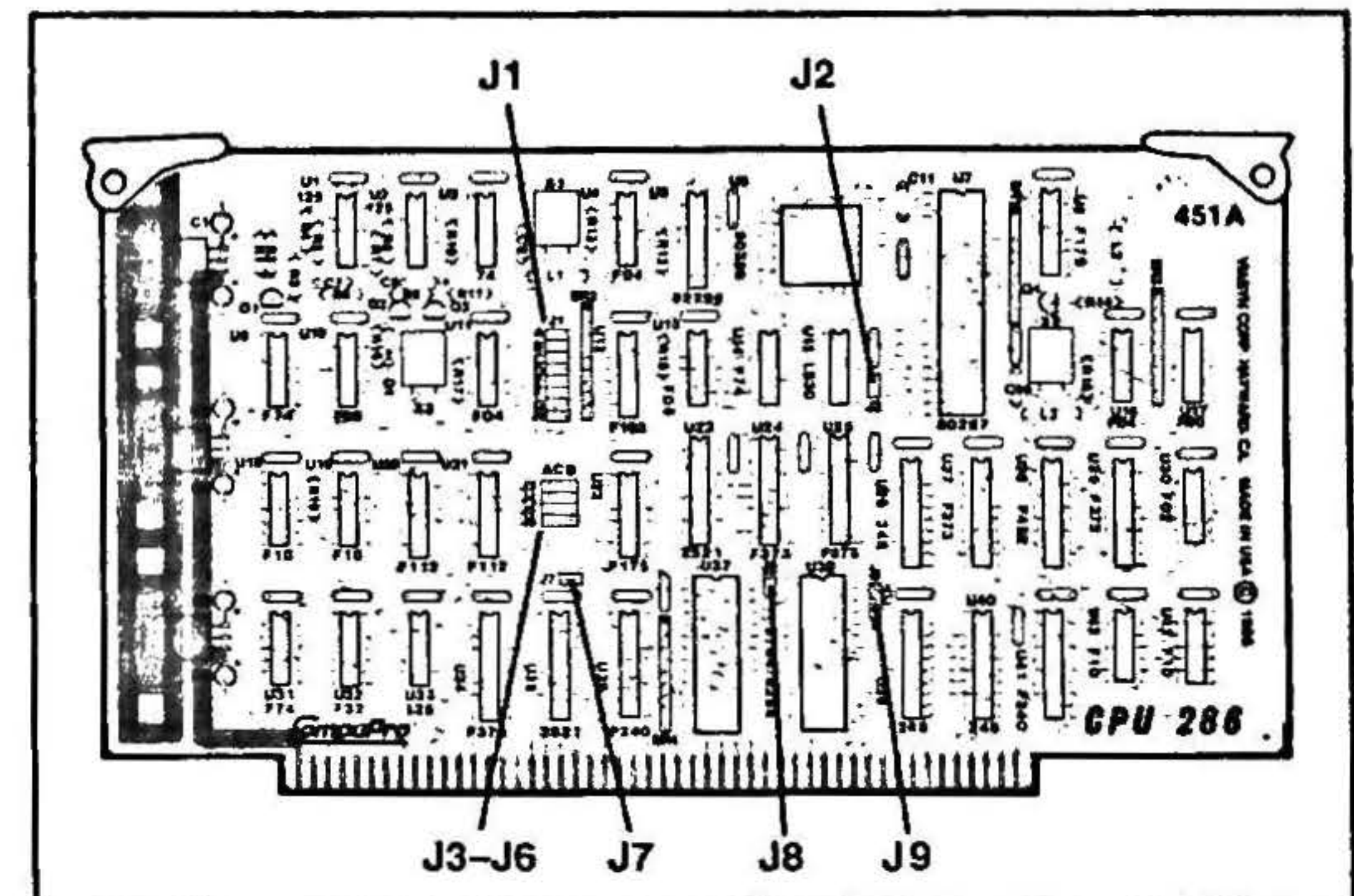
Step 5. Insert the CPU 286 into the S-100 Bus.

The power to the system must be off. Place the board into a slot towards the middle of the enclosure. The edge connector is offset, so the CPU 286 fits only one way. Push down GENTLY until the board is firmly installed.

Jumper Summary

The following summary of jumpers explains the function and logic of each option. For help in locating a switch or jumper, refer to the drawing on the previous page.

Table 1: Jumper Summary

J	Position	Function
1	A	1 Memory wait state
	B	1 I/O wait state
	C	1 Wait state for all cycles
	D	2 Memory wait states
	E	2 I/O wait states
	F	2 Wait states for all cycles
	G	4 EPROM/SRAM/EEPROM wait states
	H	8 EPROM/SRAM/EEPROM wait states
2		80287 NPX/memory manager disable
3	A-C	A23 from memory manager
	B-C	A23 from 80286
4	A-C	A22 from memory manager
	B-C	A22 from 80286
5	A-C	A21 from memory manager
	B-C	A21 from 80286
6	A-C	A20 from memory manager
	B-C	A20 from 80286
7		EPROM/SRAM/EEPROM socket master enable
8		Select EEPROM
9	A-C	EPROM/SRAM/EEPROM size select
	B-C	See Table 2

The following table lists all the possible devices that can be installed in U37 and U38, and gives the settings for J7, J8, and J9 for each.

Table 2: EPROM/SRAM/EEPROM Jumper Selection

	EPROM 2764	EPROM 27128	EPROM 27256	SRAM 6264	EEPROM 2817A
J7	Inserted	Inserted	Inserted	Inserted	Inserted
J8	Inserted	Inserted	Inserted	Don't care	Removed
J9	Removed	Removed	B-C	A-C	A-C

Jumper Settings

J1 Wait State Select

Below is a physical diagram of J1. The inputs from the left reflect the current cycle type, and the outputs to the right request automatic hardware insertion of the given number of wait states.

```
                    J1
      A (Memory)-----o  o--┐
      B (I/O)--------o  o--┤
      C (All)--------o  o--┘----- 1 Wait request
      D (Memory)-----o  o--┐
      E (I/O)--------o  o--┤
      F (All)--------o  o--┘----- 2 Wait requests
      G (ROM)--------o  o-------- 4 Wait requests
      H (ROM)--------o  o-------- 8 Wait requests
```
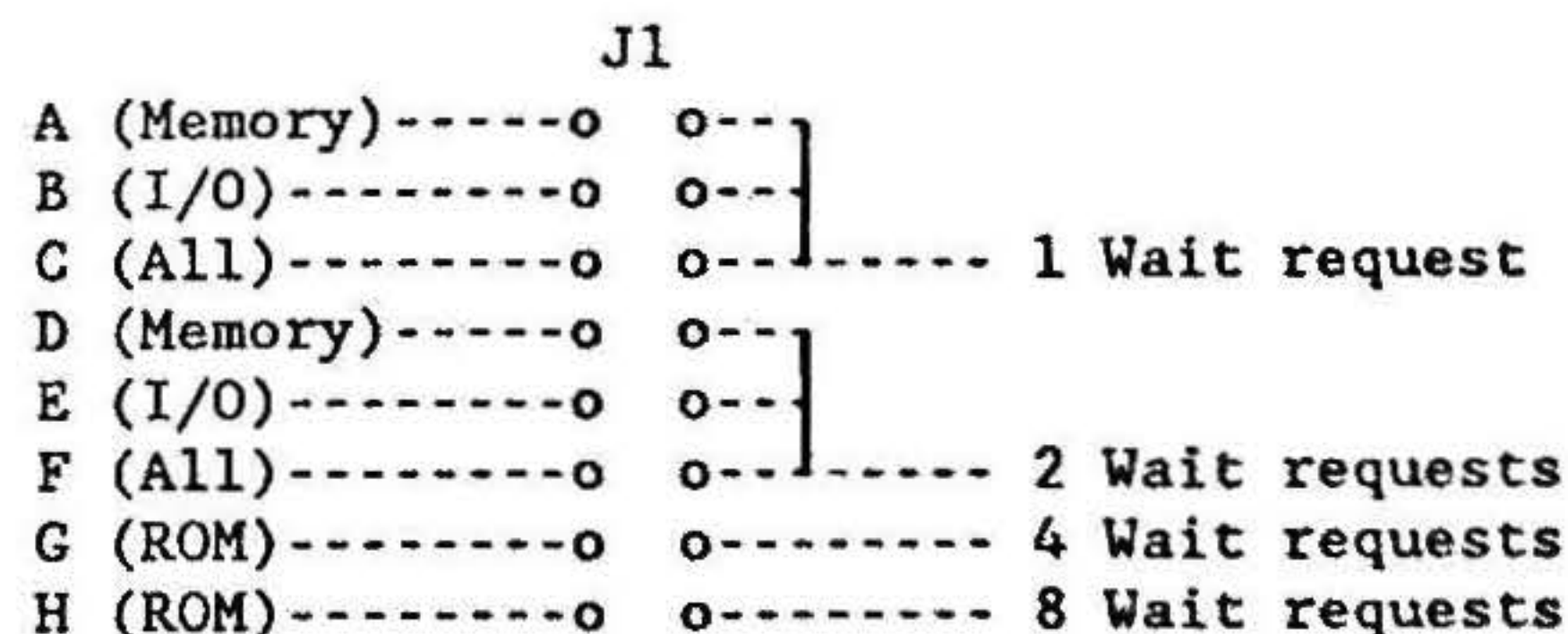
To select the number of wait states the given space (I/O, Memory, RAM/ROM Sockets) has, jumpers must be installed in the proper location(s).

Use the following rules for jumper selection:

- Placing no jumpers on any positions of J1 results in no wait states for any accesses.

- No more than one jumper may select a given number of wait states; i.e., a jumper may be placed on either position A, B, or C, but not more than one of these positions may be jumpered.

- The wait states are additive; i.e., if I/O is jumpered for both one and two wait states, a total of three will be inserted for I/O cycles.

- The RAM/ROM sockets are part of memory space. Therefore, jumpers should only be inserted into positions G and H if the on-board RAM/ROM need more wait states than the off-board memory.

The following are several examples of possible wait state configurations:

	Example 1	Example 2	Example 3	Example 4
	J1	J1	J1	J1
A	o o	o o	o o	o o
B	o—o	o o	o—o	o o
C	o o	o—o	o o	o—o
D	o o	o o	o o	o o
E	o o	o—o	o o	o o
F	o o	o o	o—o	o o
G	o o	o o	o o	o—o
H	o o	o o	o o	o o

	Example 1	Example 2	Example 3	Example 4
MEMORY:	0	1	2	1
I/O:	1	3	3	1
ROM:	0	1	2	5

NOTE: A no wait state cycle is an illegal S-100 bus cycle. Zero memory wait states should only be used if all CompuPro components are used in a system and the installation manual suggests zero wait states. At least one wait state should always be inserted into I/O cycles.

J2 80287 NPX and Memory Manager Disable

This jumper is normally not used.

The CPU 286 decodes sixteen I/O addresses (00F8h through 00FFh and 01F8h through 01FFh) of the 64K I/O port map to its internal devices: the 80287 and the memory manager. Whenever these addresses are accessed (and J2 is removed) the data bus buffers from the S-100 bus as well as the byte serializer are disabled, allowing data to be read from or written to the 80287 or the memory manager. Any peripherals residing on the S-100 bus at these addresses cannot be correctly accessed. If it is imperative to have peripherals at these addresses and neither the NPX nor the memory manager is going to be used, jumper J2 can be installed, allowing the ports back onto the S-100 bus. For more information, refer to the Port Map in the Programming Considerations section of this manual.

J3 through J6 Memory Manager Select

These four jumpers choose whether the high four bits (A20 through A23) come from the 80286 or from the memory manager. When position B-C is shunted, the address lines come directly from the 80286; when position A-C is shunted, they come from the memory manager. J3 through J6 correspond to A23 through A20 respectively.

As shipped from the factory, jumpers J3 through J6 are normally connected by a small trace on the solder side of the board across B-C. The default is then to bring A20 through A23 from the 80286. To connect the memory manager bits, cut the small traces on the other side of the beard from J3 through J6, install pins, and place shunts across A-C.

The Programming Considerations section provides more information about the programming and effects of the memory manager.

The on-board memory manager allows the CPU 286 to set the high nibble (A20 through A23) of its physical address.

The 80286 has two address modes: real mode and protected virtual address mode.

● In real address mode, the 80286 is limited to 1 Mbyte (20- bit) addressing. The CPU 286 memory manager can extend the 20 bit limit up to 24 bits. Software written to take advantage of the memory manager requires J3 through J6 to be set A-C. Software that does not use the memory manager generally does not care about how J3 through J6 are set.

● When in protected mode, the 80286 can produce 24 bits of address (memory up to 16 Mbytes), and the memory manager is not used. In this case J3 through J6 should all be set to B-C (as shipped).

The jumpers can be mixed so that some of the bits come from the memory manager and some from the 80286. This is not normally desirable.

J7 EPROM/SRAM/EEPROM Master Socket Enable

With this jumper removed, the EPROM/SRAM/EEPROM sockets do not respond, the 64K window in page 0FFh appears as memory on the S-100 bus, and the wait state jumpers for the sockets have no effect.

To enable the sockets to respond, pins and a shunt should be installed. When the jumper is installed, the 64K window at page 0FFh (memory addresses 0FF0000h through 0FFFFFFh) will be located in the sockets. If no devices are placed in the universal sockets U37 and U38, this jumper should be left disconnected (as shipped).

J8 Select EEPROM

This jumper is shipped with a normally closed connection that allows for 2764, 27128, 27256 EPROMs, and 6264 SRAMs in sockets U37 and U38.

Since pin 1 of some EPROMs is required to be tied to power (Vcc), and pin 1 of 2817A type EEPROMs is an open drain output, a socket meant to support both requires the ability to connect and disconnect pin 1 from Vcc.

If 2817A or compatible EEPROMs are to be inserted in the sockets, this jumper must be cut. Note that because of this, EPROMs and EEPROMs can't be mixed (unless pin 1 of the EEPROM is removed from the socket).

CAUTION: **Permanent damage** may occur to EEPROMs if J8 is left connected.

J9 Select Memory Chip Type

This jumper selects the proper pinout for the various EPROM/SRAM/EEPROMs that can be installed in the on-board sockets.

Jumper J9 as follows:

● A-C for 6264 type SRAMs and 2817A type EEPROMs
● B-C for 27256 type EPROMs
● No shunt for 2764 or 27128 type EPROMs

Installing an 80287

Numeric Processor Extension

The socket for the 80287 has been fully tested by CompuPro at the factory and is ready to accept a 5 MHz 80287. To install the 80287, plug the part into the 40 pin socket labeled U7 (immediately to the right of the 80286). No board modifications are necessary. Once installed, the 80286/80287 pair become object code compatible with the 8086/8087.

The CPU 286 board supports 5, 8, and 10 Mhz 80287 chips. The table below specifies the values of capacitor C10 and inductor L3 required for the timing circuits for the three 80287 speeds.

Table 3: 80287 Timing Circuit Components

80285	X2	C10	C3
5 Mhz	15 Mhz	120 pF	1.0 uH
8 Mhz	24 Mhz	100 pF	0.47 uH
15 Mhz	30 mHz	100 pF	0.33 uH

If you are not familiar with hardware or have never inserted a large IC into a socket, the time to learn is not with a several hundred dollar part; it is too easy to break a pin and ruin the IC. Return your board to CompuPro for a factory upgrade, and we will run a complete confidence test on the board and numeric processor.

Programming Considerations

80286 Compatibility with 8086/8088

The 80286 is upward software compatible with the 8086 and 8088. This feature makes the 80286 very attractive to users who have an investment in 8086/8088 code and who need a higher performance processor. While we were able to bring up 99% of our existing 8086/8088 software with the 80286, problems did emerge in timing, port mapping, and 24-bit addressing. The following discussion outlines the problems and their solutions. At the end of this section is a discussion of programming for the memory manager.

Timing Compatibility

Because the 80286 is substantially faster than the 8086, and due to the optimization of the pre-fetch queue, the 80286 often executes bus cycles in a different order than the 8086. To see where this is a problem, consider initializing a part such as Intel's 8259A interrupt controller.

The 8086 executes the three-byte initialization sequence as follows:

1. Fetch first operand
2. Output first operand
3. Fetch second operand
4. Output second operand
5. Fetch third operand
6. Output third operand

The 80286 executes this sequence:

1. Fetch first operand
2. Fetch second operand
3. Output first operand
4. Fetch third operand
5. Output second operand
6. Output third operand

Although the actual order in which the 8259A receives the bytes is the same for both (as it must be), the minimum time between outputs is reduced, since no bus cycle is run between the second and third outputs. The 8259A requires a minimum time between command bytes, and with the faster 80286 this time is not met.

One solution to this problem is to force the 80286 to run a bus cycle between every output. The above example used immediate operands as initialization bytes, allowing the 80286 to pre-fetch and use them quickly. By putting the initialization bytes into a small table and fetching them individually for every output, the 80286 is forced to run at least one bus cycle (the operand fetch) between every output. This easily satisfies the 8259A minimum.

Port Mapping

80287 Numeric Processor Extension is I/O mapped into ports 00F8h through 00FFh of the 80286. From a hardware standpoint, this is a logical thing to do, as it allows the NPX and CPU to run at different speeds. The problem is that any software that accesses peripherals at these ports not only messes up the 80287, but it cannot access the peripherals correctly. There is a similar potential conflict in port mapping for the memory manager, which is mapped to ports 01F8h through 01FFh.

The best solution is to relocate the conflicting S-100 boards in the system to different ports. This is the only solution if the NPX is to be used. If it is impossible to relocate these ports and if the 80287 and the memory manager will never be used, jumper J2 can be installed. In this case, be very careful not to execute any NPX "ESCAPE" instructions, as the bus peripherals could get corrupted when the 80286 accesses these ports.

Table 4: CPU 286 Port Map (J2 removed)

I/O Address	Device
0000h-00F7h	Mapped to S-100 bus
00F8h-00FFh	Mapped to 80287
0100h-01F7h	Mapped to S-100 bus
01F8h-01FFh	Mapped to memory manager
0200h-FFFFh	Mapped to S-100 bus.

24-Bit Addressing

The 80286 initializes and starts execution at location 0FFFFF0h. Notice that this is a 24-bit address, not a 20-bit address as produced by the 8086/8088. This is not a problem in complete CompuPro systems, as the boot EPROM on the DISK1A appears in every 64K page while PHANTOM* is asserted during boot up.

While in real address mode, since the 80286 does not know what to do with the most significant four bits (A20 through A23), it leaves them high on boot until a long jump (one that loads the code segment) is executed, at which time it sets the four bits low. This means that if the boot code is in the on-board EPROM, that code cannot execute any long jumps, or it will jump right out of the EPROM's address space. Once out of the EPROM, the program can never get back without going into the protected address mode.

Programming for the Memory Manager

The memory manager on this board allows the CPU 286 to set the high nibble (A20 through A23) of its physical address.

In real address mode, the memory manager can be used to extend the limit of 20 bits (memory up to 1 Mbyte) that the 80286 can produce up to 24 bits (approaching 16 Mbytes).

The memory manager is reset to all 0. Thus A20 through A23 go low on a reset, and stay low until something else is written to the memory manager. To write to the memory manager, place the value into the low nibble of 80286 register AL and write a byte to any even port in the range 01F8h to 01FEh. Reading from any port in this range is not mapped to the S-100 bus, and does not affect the memory manager. Writing a byte or word to an odd address in this range puts random data into the memory manager.

A general model for using the memory manager to extend the accessible memory space of a 80286 in real address mode is to divide the memory into global memory which is always visible to the 80286 and into banked memory which is banked or switched on and off depending on the memory manager ouput. Usually the banked memory space would be much larger than the global memory space.

The global memory addressing does not involve lines A20 through A23 and thus global memory is always visible to the 80286. The bits that are written to the memory manager on lines A20 to A23 choose which of the banked memory pages are visible to the 80286. The addressing must be set up to make sure that this banked memory does not collide with global memory.

Following is an example of how to use the memory manager to address both global and banked memory. This example is only an outline; the details depend upon the memory boards and operating system being used.

This example uses memory in 256 Kbyte blocks. The desired result is to have 256 Kbytes of global memory at the bottom (00h) of memory space, and allow the upper 768 Kbytes of memory to be used for banking into or out of up to 16 banks. Using a 256 Kbyte global memory space in this way, over 12 Mbytes of memory is accessible to the 80286.

Note: It is not necessary to fill a bank with memory before starting the next.

| Global Memory | A23 through A20 are disabled on the memory board. A19 through A18 are set to low. Thus, the global memory board appears at: |

         000000h-03FFFFh
and   100000h-13FFFFh
and   200000h-23FFFFn
...
and   F00000h-FFFFFFh.

| Banked Memory | Each bank requires three 256K memory boards to make 768K. Address each board within the bank with A19 and A18. On the first board, set A19 low and A18 high. On the second board set A19 high and A18 low. On the third board, set A19 high and A18 high. In no case should any board be set with both A19 and A18 low, as that would conflict with the global board. |

| Bank 0 | On all three boards, set A23 through A20 to respond to low. Set A19 and A18 as described in the description of Banked Memory. Thus, the memory appears at: |

board 1 040000h-07FFFFh
board 2 080000h-0BFFFFh
board 3 0C0000h-0FFFFFh

| Bank 1 | On all three boards, set A23 through A21 to respond to low, and A20 to respond to high. A19 and A18 are set the same as in bank 0. Thus, the memory appears at: |

board 1 0140000h-017FFFFh
board 2 0180000h-01BFFFFh
board 3 01C0000h-01FFFFFh

This pattern could continue up through Bank 0Fh. The table below shows how to set A23 through A20 for all the possible banks. For all banks, A19 through A18 are set to select the three boards within the bank.

Table 5: Memory Bank Addressing

BANK	A23	A22	A21	A20
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
...				
E	1	1	1	0
F	1	1	1	1

Another possible configuration would be 256 Kbytes of global memory and 256 Kbytes of banked memory in each bank. Each user could then have a separate 256 Kbyte bank, and would not be able to access other user programs.

Theory of Operation

In the following discussion, it is helpful to refer to the schematic diagrams at the end of this manual.

The CPU 286 is based on the 80286 CPU and the Intel 80287 NPX.

80286 Clock Circuitry

The clock for the CPU is generated by U5, the 82284 clock generator IC. It uses an external oscillator consisting of two inverters, crystal X1, capacitor C9, inductor L1, and resistor R12. The crystal can be a fundamental or overtone, parallel or series, and is twice the desired processor frequency. For example, to run the CPU at 10 MHz, a 20 MHz crystal should be used. If the crystal value is changed, the value of L1 must also be changed.

Unlike the 8086/8087 pair, the 80287 can run at any speed with respect to the 80286. The CLK input for the 80287 is generated by three parts of U16. The circuit consisting of transistor Q4, inductor L2, and resistor R14 provide the large swing, fast rise and fall time clock to the 80287. If a different crystal is used for X2, capacitor C10 and inductor L3 must be changed. Crystal X2 can be either a fundamental or an overtone, parallel or series, and is three times the speed of the 80287. A 15 MHz crystal is used for a 5 MHz 80287 on the CPU 286 board.

The clock output of the 82284 is double the actual processor clock speed. It is divided by two inside the 80286 to get the processor clock. The CLK and CLK* signal are used by the 80287 (into CLK286) to give it a sampling edge for its processor signals (HLDA, S1*, S0*, etc.) and by the S-100 Bus Controller unit to synchronize it with the 80286. Since the 80286 does not indicate what phase its internal clock is with respect to the CLK input until the first bus cycle, the circuit consisting of U13 (74F08), U14 (74F74), and U21 (74F112) is needed to ensure that the proper phase of Φ and IPHCLK* is used during the first bus cycle. This circuit forces Φ to oscillate during power-on-clear, then turns off Φ (to the high state) from the end of IPOC* until the first bus cycle when S1* goes low. This circuit then ensures that the phase of IPHCLK* and Φ matches the phase of the 80286's internal clock.

Numeric Coprocessor

The 80286 (U6) and the 80287 (U7) communicate via eight I/O mapped ports, the PEREQ and PEACK* (peripheral request and acknowledge), BUSY* and ERROR*, and several processor status signals (S1*, S0*, HLDA, C/I*, READY*, and CLK). When the 80286 sees an ESC instruction dealing with the 80287, it checks the status of the BUSY*; if the 80287 is not currently busy, the 80286 instructs the NPX to execute a command. When the 80287 requires data from memory, it can assert PEREQ at which time the 80286 performs the transfer.

Memory Manager

The memory manager consists of NAND gate U19 (74F10), quad register U22 (74F175), and the four three-pin jumpers J3 through J6. When any port from 01F8H through 01FFH is written to, the data that appears on data bus lines D0 through D3 is loaded into the memory manager. The actual loading takes place on the trailing edge of the PWR strobe. When the board is reset, the memory manager is loaded with a 0, causing A20 through A23 to be low. This is different from the 80286, which asserts A20 through A23 high on reset. To load the memory manager, place the desired page into the low nibble of register AL, and run an OUT to port 01F8h.

80287 and Memory Manager I/O Mapping

Intel has defined the 80287 I/O ports to reside at 00F8H to 00FFH. Comparator U23 (25LS2521) and NAND gate U15 (74LS30) decode these I/O ports in addition to the memory manager I/O ports which are at 01F8H to 01FFH. IFPU* is then latched in U25 (74F373) to generate a stable LFPU* throughout the bus cycle. Quad register U8 (74F175) then re-clocks LFPU* as well as the other inputs to the 80287 to ensure that they have adequate setup and hold times around the DBIN* and PWR* strobes. With this circuitry, the I/O ports that the 80287 and memory manager occupy are excluded on the S-100 bus. To get these to map back to the S-100 bus, J2 can be installed, always forcing the output of U15 to be false and never selecting either the 80287 or the memory manager.

Timing

The two 74F112 JK and one 74F74 D Flip Flops (U20, U21, and U9), along with their respective input logic, generate the S-100 strobes from the 80286 output signals.

U9 generates one cycle of SYNC at the beginning of every cycle (S0* or S1* low) and at the beginning of the second half of a byte-serialized fetch (STROBE is not asserted and TWO-CYCLE is). STROBE and either DBIN or PWR, depending on the state of LS1* (the direction status), are generated by either U20 or U21 (both 74F112) respectively immediately following SYNC assertion. DBIN or PWR are then held as long as STROBE is asserted and there is no READY* from the 82284 or STBINH* from the byte serializer indicating the end of a bus cycle.

The HLDA strobe is produced directly by the 80286; the STVAL* strobe follows pSYNC immediately, followed by a minimum of 5 nanoseconds.

The strobes and φ are buffered by U36 (74F240) on to the S-100 bus. The buffer (except for φ) is disabled when S-100 signal CBSD* is asserted.

When either SYNC, a strobe, or TWO-CYCLE is asserted, and is released, ALE is lowered latching addresses and status. Then, when STROBE* ends and pDBIN ends, a new address and status are allowed to the S-100 bus. This guarantees a minimum hold time for addresses and status on the bus after pDBIN and assures a generous hold time after pWR*.

8-Bit and 16-Bit Byte Serializer

U31 and U9 (both 74F74) and two gates in U18 (74F10) include all the logic necessary to handle both onecycle and twocycle fetches. A onecycle fetch is when the processor requests either 8 or 16 bits from memory and the memory is able to handle the transfer in one S-100 bus cycle. A twocycle fetch is when the processor requests 16 bits from the memory but the memory is only able to transfer 8 bits, forcing the internal finite state machine (U31) to complete two S100 bus cycles to fetch two bytes before allowing the 80286 to complete its cycle. A twocycle fetch is also called a byte serial fetch.

The decision to execute either a onecycle transfer or a twocycle transfer is controlled by the S-100 signals sXTRQ* and SIXTN*.

- sXTRQ* is generated by the OR of the signals LBHE* and LA0. If BHE* and A0 are latched low, SXTRQ* will be generated. If a peripheral board responds by asserting SIXTN* low, indicating that a 16-bit transfer can occur, the output of U18 (pin 8) goes high, causing the ONECY signal to be true. This signal tells the CPU to complete the 16-bit transfer at full speed.

- If the SIXTN* signal stays high, U18's output (pin 8) is low, causing the ONECY signal to be false. This starts a process whereby the CPU 286 puts the 80286 into a wait state and reads or writes two bytes serially.

The twocycle is performed by the state machine (consisting of U31, U9, and two gates in U18) which generates the signals STBINH*, TWO-CYCLE, and FLIP. These signals are used to sequence the logic on the board to run the two bus cycles.

- STBINH* is produced by U18 during the last cycle that strobes are to be asserted during the first of the two serial bus cycles. STBINH* is fed to U19 pin 2 to generate KSTB (kill strobe), ending the strobes.

- TWO-CYCLE is asserted by U9 when the first SYNC* falls and the external memory has not yet produced SIXTN*.

- FLIP signals the start of the second half of a twocycle, and is produced by U9 when strobes fall and TWO-CYCLE is asserted.

Finally, the signal CLA0 (corrected LA0 to the S-100 bus) is fed directly from LA0 to the S-100 bus like the rest of the address lines during a onecycle, but is asserted high by FLIP during the second half of a twocycle to enable access of the high byte.

Data Bus

The data bus is buffered, multiplexed and latched (depending on what is required) by U26, U39, and U40 (all 74LS245), and U27 (74F373). The control of these buffers and latches is performed by a 12L6A PAL 452 (U23). LS1 determines the direction of data (transmit or receive) and goes to the direction inputs of the the main data bus buffers (U26 and U40). The data buffers are enabled during SYNC or STROBE for writes and during STROBE for reads. The rest of the inputs to the PAL control which of the various buffers are enabled. The LA0, LBHE* and LS1* signals control the basic 16-bit cycles, while the FLIP* and TWO-CYCLE* signals control the buffers during a byte serial transfer. The INTA* signal routes the byte data correctly during interrupt acknowledge cycles. Note that 8 bits are latched by U16 on the falling edge of DBIN (through U10) during the first half of a twocycle fetch, and are asserted onto the D0-D7 bus during the second half.

Status Bus

The S-100 status lines are generated by gates U17 (74F00), U30 (74F02), U42, and U43 (both 74F08). The status lines from the CPU (S0*, S1*, C/I*, M/IO*, BHE*) are latched by U29 (74F373). The outputs of U29 go to the inputs of the gates. The logic then decodes the proper status and feeds that to U41 (74F240) to drive the S-100 status lines. The status output buffer is disabled when S-100 signal SDSB* is asserted.

Wait Circuitry

The internal wait state generator is controlled by the 4-bit counter U12 (74F163).

When STROBE is not asserted, the counter is loaded with the data presented at its inputs. This data reflects a particular type of bus cycle through jumper J1, and clocks in the number of wait states determined by this status.

When STROBE is asserted, the counter clocks out states with IPHCLK* until CO goes high (count = 15) at which time, if pRDY is not driven low, EVWAIT* out of U13 goes high. EVWAIT* controls wait states during byte serialized accesses when ONECY is low and during full speed accesses when ONECY is high.

By selecting the proper status to feed to the counter, different wait states can be selected for the different spaces (memory, I/O, ROM). If no status is presented, then no wait states are inserted since CO goes high immediately.

NOTE: A no wait state cycle is an illegal S-100 bus cycle. Zero memory wait states should only be used if all CompuPro components are used in a system and the installation manual suggests zero wait states. At least one wait state should always be inserted into I/O cycles.

Memory Addressing

The on-board EPROM/SRAM/EEPROM sockets are decoded by U35 (25LS2521). This puts the memory in sockets U37 and U38 into the highest memory page, 0FFh.

J7 enables the sockets by allowing IROMSEL* to go low when the proper address is asserted.

Depending on how J3 through J6 are set, the high nibble of address can either come from the 80286 or the memory manager.

J9 controls whether PWR*, A15, or just a pull-up goes to pin 27 of the sockets. For SRAMs or EEPROMs, choose PWR*; for 27256s, choose A15; for 2764s or 27128s, choose the pull-up (no jumper inserted).

J8 must be removed to use 2817A EEPROMs since this is the BUSY/INTR* output from the devices.

PAL 452 decodes ROMHI* and ROMLO* from LA0 and LBHE* when ROMSEL* is asserted. Note that A14 must be high to access 6264 SRAMs, putting them only at the top of the 64K page of memory.

S-100 CLOCK

Two sections of U11, buffer U2, crystal X3 (4 MHz), and U3 provide a 2 MHz clock for the S-100 CLOCK signal on pin 49. A separate oscillator ensures that CLOCK will always be 2 MHz, independent of the CPU frequency.

Power-on-Clear and RESET*

The power fail circuit causes a POC* to be issued upon the rising edge of PWRFAIL*. This ensures that the system will recover just as if the power had come on for the first time, and prevents problems that might occur if the power dips for a short period, causing PWRFAIL* to be asserted although the power doesn't actually go away. A sophisticated low Vcc detection circuit is included (Q1, Q2, Q3, D1 + resistors) to ensure that the instant power starts dropping out, pRESET*, POC* and SLVCLR* are driven low. Capacitor C8, resistor R11 and this circuit ensure that the RESET* to the board is not released until power has been stable for a long time. Pins 1 through 6 of U1 (75LS125) and associated components buffer and give hysteresis to RESET* to ensure a clean, glitch free reset to the board.

Schematic — CPU 286 (page 2 of 5)

Schematic — CPU 286 (page 3 of 5)
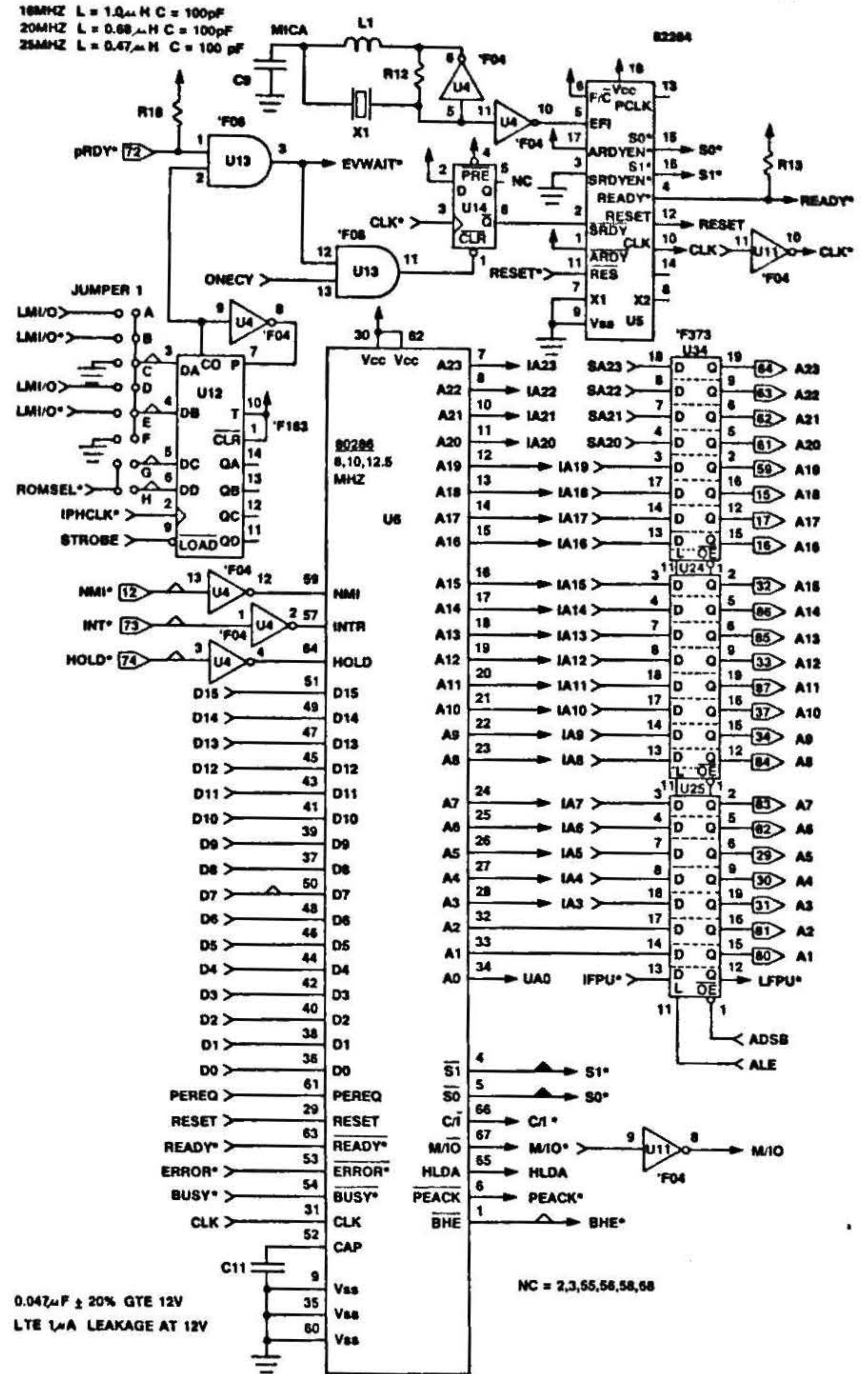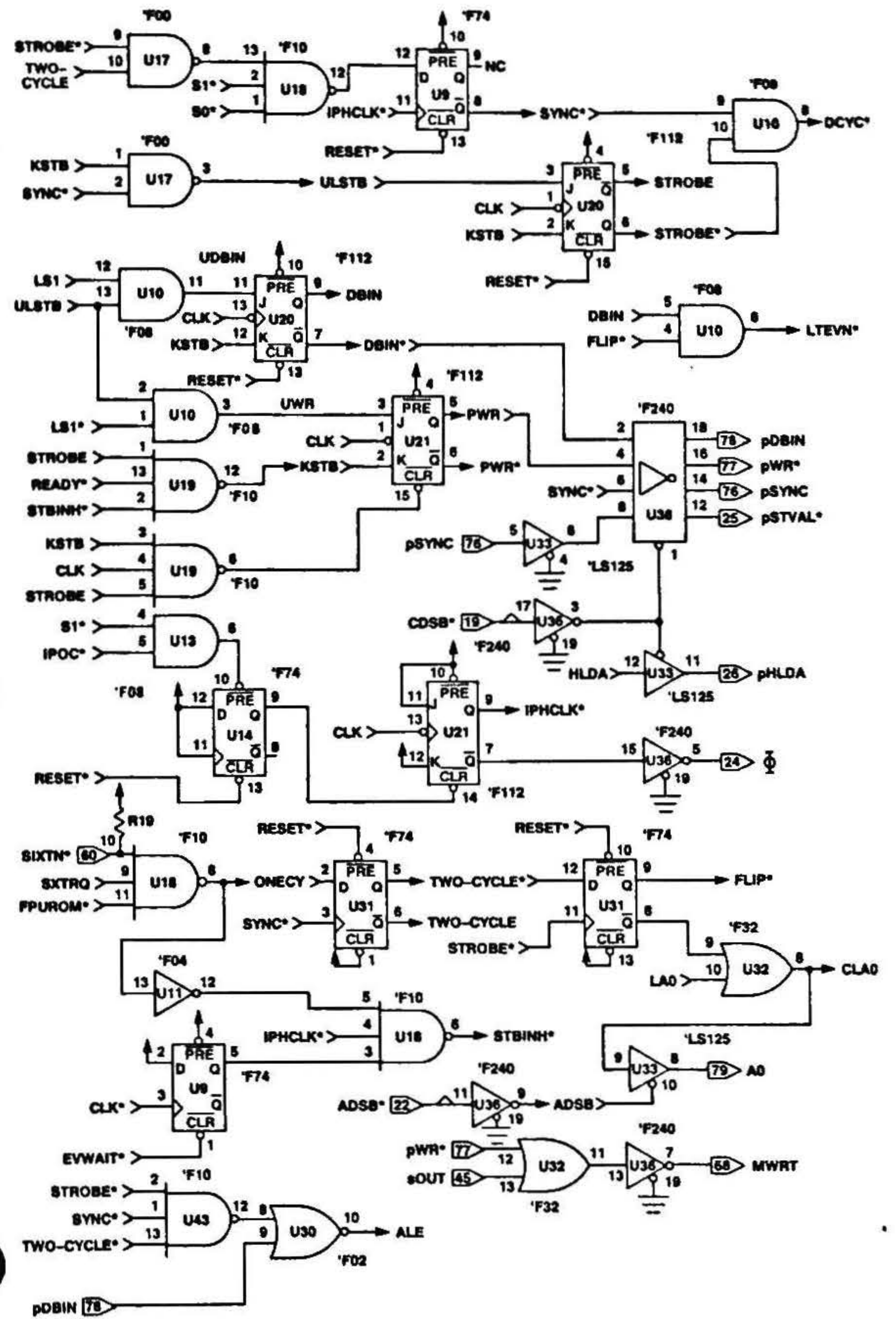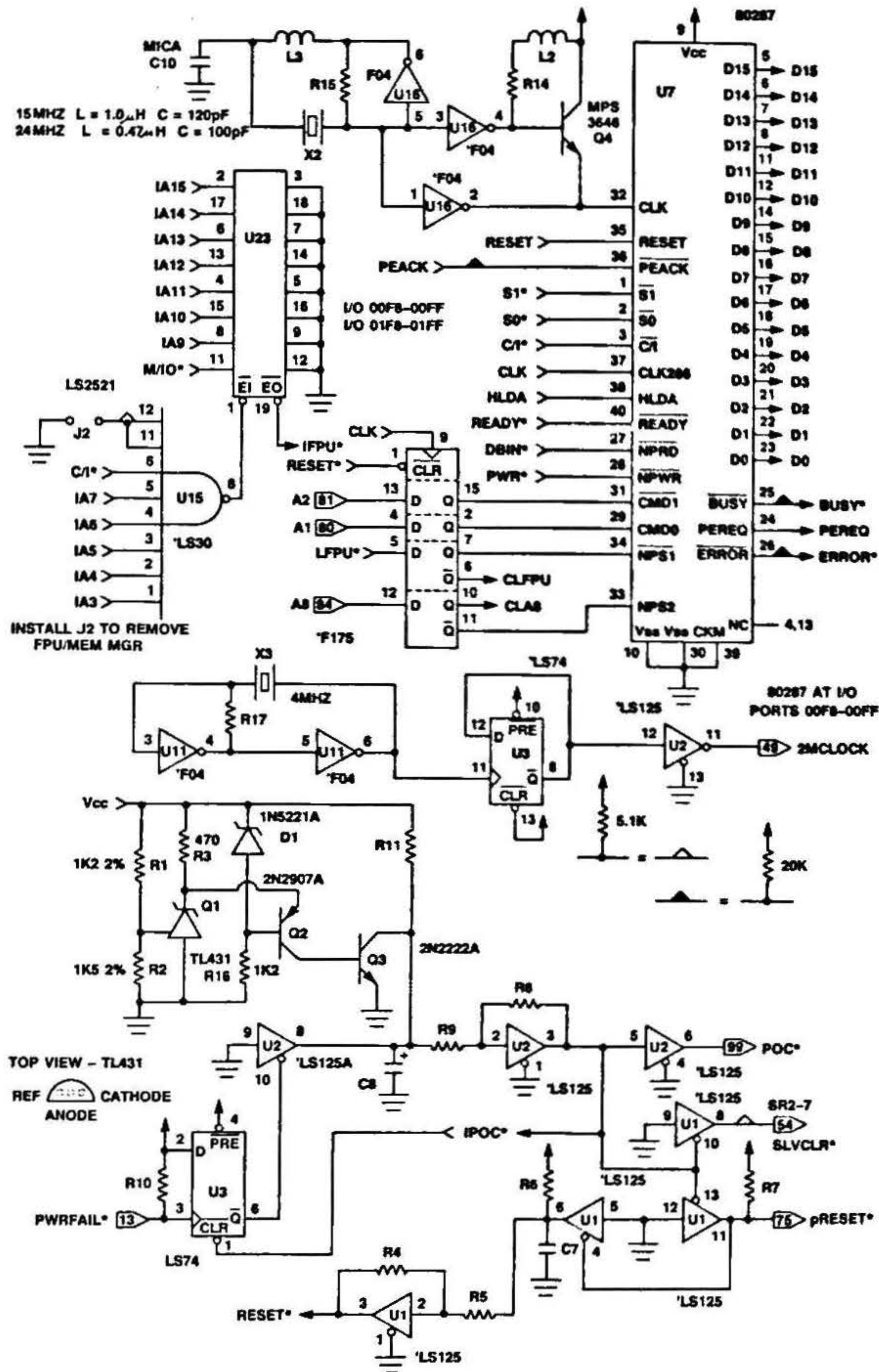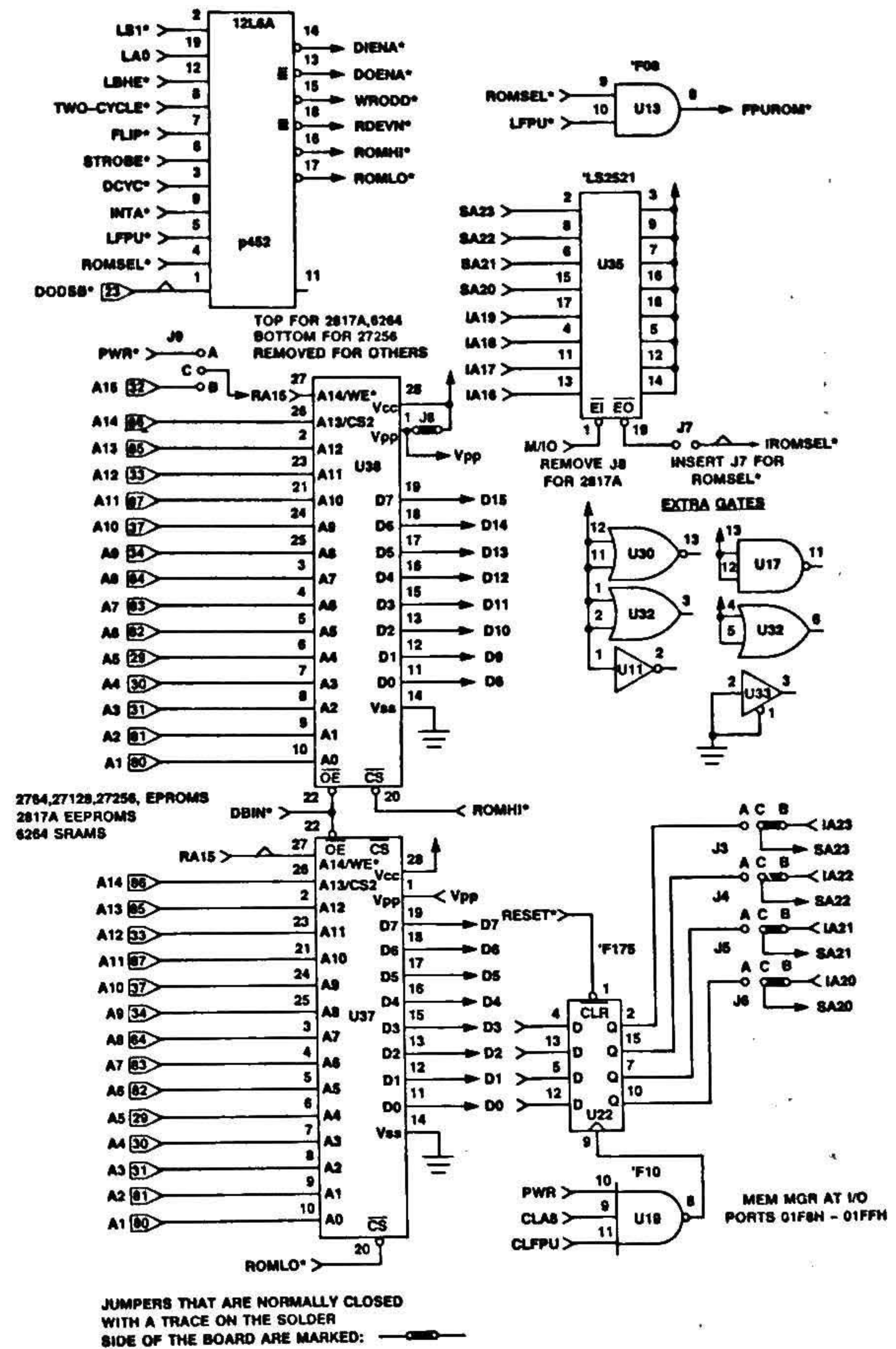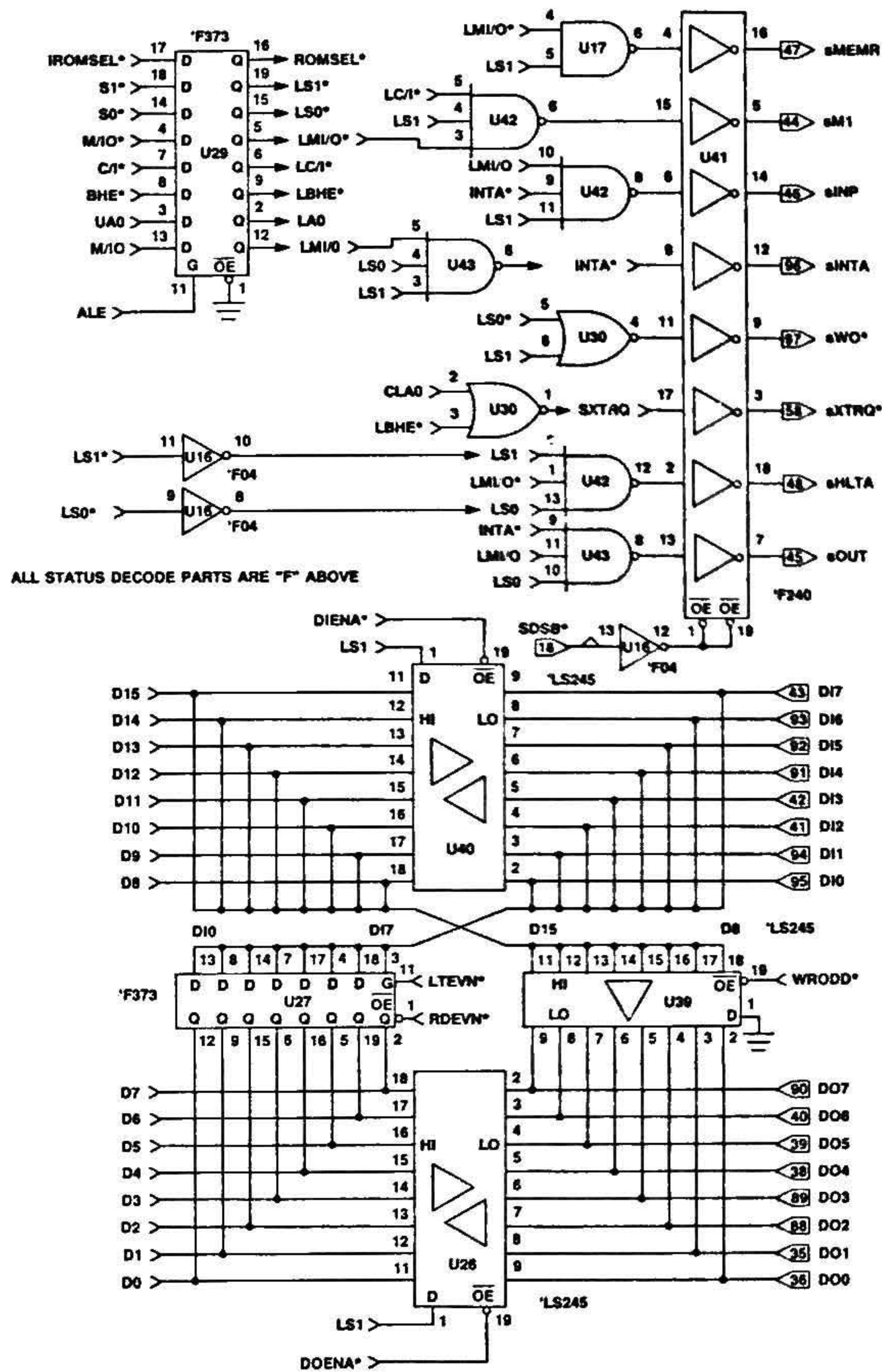
Parts List

Integrated Circuits

74F00	1	U17
74F02	1	U30
74F04	3	U4,U11,U16
74F08	2	U10,U13
74F10	4	U18,U19,U42,U43
74LS30	1	U15
74F32	1	U32
74LS74	1	U3
74F74	3	U9,U14,U31
74F112	2	U20,U21
74LS125	3	U1,U2,U33
74F163	1	U12
74F175	2	U8,U22
74F240	2	U36,U41
74LS245	3	U26,U39,U40
74F373	5	U24-25,U27,U29,U34
25LS2521	2	U23,U35
12L6A 452	1	U28, pgmd to 452.
82284 8MHz	1	U5
80286 8MHz	1	U6
7805	3	U44,U45,U46

Transistors

MPS3646	1	Q4
2N2222	1	Q3
2N2907A	1	Q2
TIL431	1	Q1

Diode

1N5221A	1	D1

Resistors

120 ohm	2	R5,R9
560 ohm	5	R7,R10,R14,R18,R19
470 ohm	1	R3
910 ohm	1	R13
1K ohm	3	R6,R12,R15
1K2 ohm	2	R1,R16
1K5 ohm	1	R2
2K2 ohm	1	R17
2K7 ohm	2	R4,R8
4K7 ohm	1	R11
20K ohm SIP	1	SR1
5.1K ohm SIP	3	SR2,SR3,SR4

Inductors

Inductor 22uH	1	L2
Inductor 1uH	1	L3,L1

Crystals

Crystal 16MHz	1	X1
Crystal 15MHz	1	X2
Crystal 4MHz	1	X3

Capacitors

100pF mica	1	C9
120pF mica	1	C10
0.05uF cer	1	C11
0.1uF mono cer	1	C7
10uF 8V	1	C8
Ceramic Bypass	41	all unmarked
Radial tantalum +5V	6	C1,C2,C3,C4,C5,C6

Component Layout

Notes on Using the CPU 286
With Certain Other CompuPro Products

Revision D-2 or earlier of Concurrent DOS 8-16 version 4.1

If you will be using this CPU 286 board with Concurrent DOS 8-16 4.1D-2 or earlier, you may need to install a patch to RTM.CON.

Problem:

A bug has been identified in RTM.CON that causes system lock-up when files are opened in "unlocked mode" (FCB attribute bit F5=1, F6=0). This lock-up is caused by a specific location in memory being uninitialized, which allows the CPU to execute an illegal instruction. This problem is often seen running NEWWORD.

Solution

The solution to this problem is to patch the RTM.CON module and if desired, the CCPM.SYS file being used on the system. If possible, we recommend that you patch RTM.CON and use this corrected module to GENCCPM a new system file, rather than perform the more difficult patch on your CCPM.SYS file. The procedures for these patches are listed below:

NOTE: This patch is not a Digital Research patch, however they have been informed of the procedure.

Patch Procedure

Make a back-up copy of RTM.CON and your CCPM.SYS files before using SID86.CMD to make the changes on the following pages. User entries are underlined.

Left column:
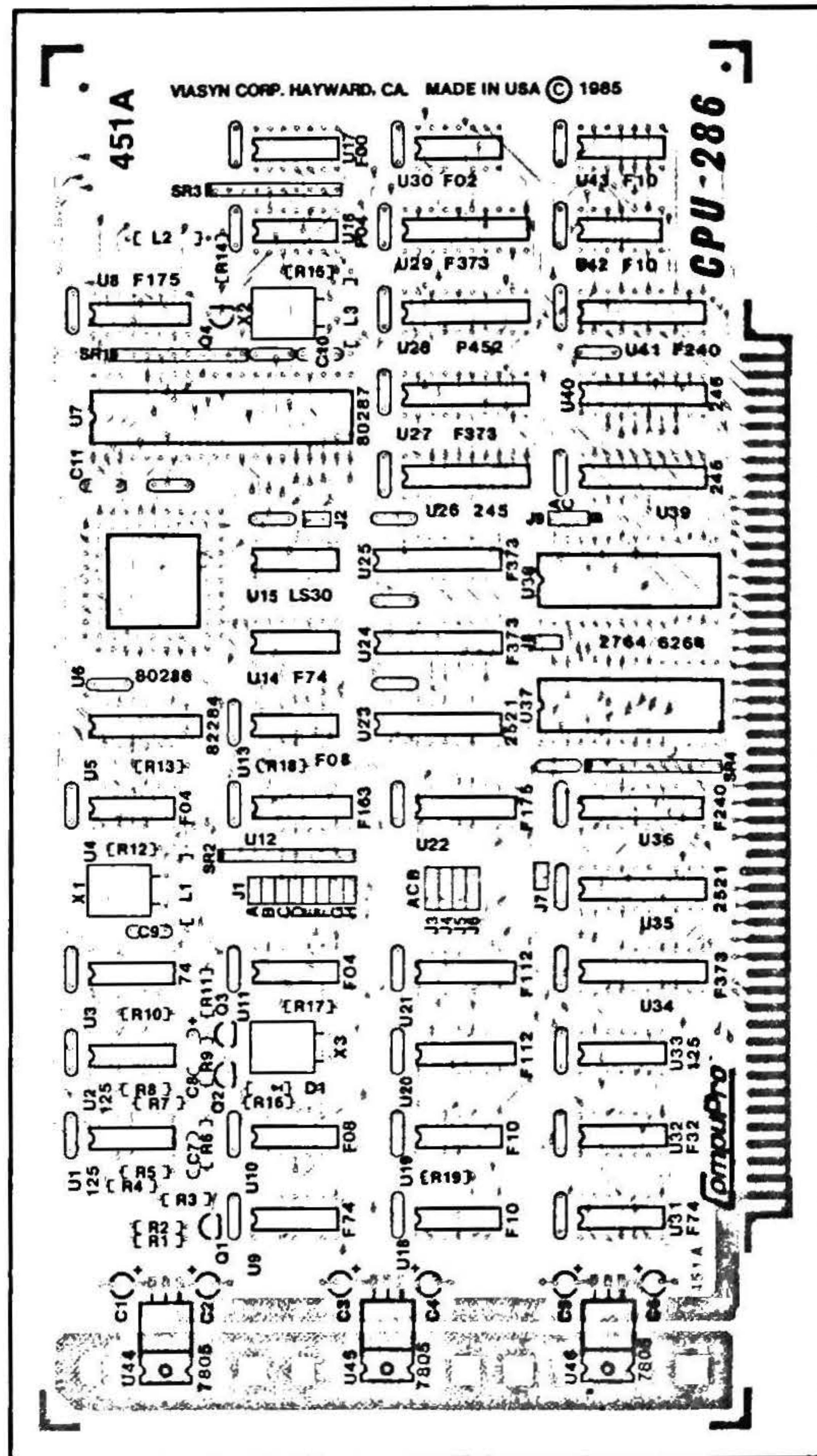
```
0A>SID86
SID86 1.0
#rRTM.CON
   START      END
XXXX:0000 XXXX:1DFF
#L41A,41E
XXXX:041A MOV  [0052],DX
XXXX:041E MOV  DS,[001E]
#a41A
XXXX:041A CALL 1110
XXXX:041D NOP
XXXX:041E .
#L1110,1119
XXXX:1110 NOP
XXXX:1111 NOP
XXXX:1112 NOP
XXXX:1113 NOP
XXXX:1114 NOP
XXXX:1115 NOP
XXXX:1116 NOP
XXXX:1117 NOP
XXXX:1118 NOP
XXXX:1119 NOP
#a1110
XXXX:1110 MOV [52],DX
XXXX:1114 MOV [54],DX
XXXX:1118 RET
XXXX:1119 .
#L41A,41E
XXXX:041A CALL 1110
XXXX:041D NOP
XXXX:041E MOV  DS,[001E]
#L1110,1118
XXXX:1110 MOV [0052],DX
XXXX:1114 MOV [0054],DX
XXXX:1118 RET
#wRTM.CON
#^C
0A>
```

Save this new corrected copy of RTM.CON so that the next time you GENCCPM a system, you will use the corrected module. At this time, you can regenerate your CCPM.SYS file as described in your documentation, or patch the CCPM.SYS file as shown on the following page.

Right column:

```
0A>SID86
SID86 1.0
#rCCPM.SYS
   START      END
XXXX:0000 YYYY:ZZZZ
#sw01
XXXX:0001 08A6 <CR>
XXXX:0003 0060 .
#hWXYZ,8A6          (WHERE WXYZ=XXXX+8)
+ WWWW   - ????   * ????????  / ?????? (????)
#dwWWWW:0,F
WWWW:0000 0003 0060 0000 0060 0003 VVVV 0000 01C7 ..'...
#hVVVV,0060
+ ????   - UUUU   * ????????  / ?????? (????)
#hWXYZ,UUUU
+ MMMM   - ????   * ????????  / ?????? (????)
#LMMMM:39A,39E
MMMM:039A MOV  [0052],DX
MMMM:039E MOV  DS,[001E]
#aMMMM:39A
MMMM:039A CALL 1090
MMMM:039D NOP
MMMM:039E .
#LMMMM:1090,1099
MMMM:1090 NOP
MMMM:1091 NOP
MMMM:1092 NOP
MMMM:1093 NOP
MMMM:1094 NOP
MMMM:1095 NOP
MMMM:1096 NOP
MMMM:1097 NOP
MMMM:1098 NOP
MMMM:1099 NOP
#aMMMM:1090
MMMM:1090 MOV [52],DX
MMMM:1094 MOV [54],DX
MMMM:1098 RET
MMMM:1099 .
#LMMMM:39A,39E
MMMM:039A CALL 1090
MMMM:039D NOP
MMMM:039E MOV  DS,[001E]
#LMMMM:1090,1098
MMMM:1090 MOV [0052],DX
MMMM:1094 MOV [0054],DX
MMMM:1098 RET
#wCCPM.SYS
#^C
0A>
```

This completes the patch procedure.

System Support 1 without ECO 179:

ECO (Engineering Change Order) 179 eliminates the possibility of the System Support 1 board's 8259A interrupt controllers missing an interrupt acknowledge from the CPU, this crashing the system. ECO 179 speeds up the interrupt acknowledge path on the System Support 1 board. ECO 179 is documented in CompuPro Product Assurance Technical Bulletin #35, dated March 6, 1986.

ECO 179 may be applied only to revision 162G of the System Support board. Earlier revisions of the System Support 1 board must be returned to the factory for an upgrade. The "ECO 179" label found on the component side of the board assures that this change has been made at the factory.

System Support 1 boards without ECO 179 already installed at the factory, may be upgraded in one of two ways. The board may be easily ECOed in the field, or it may be returned to the factory for modification. For factory modification, contact the RMA desk for return authorization prior to returning the board.

Field Modification

The field modification of this board is very simple, and requires only 2 wires to be soldered to the board. If you do not feel comfortable making this modification after reading the instructions below, return the board to CompuPro for factory modification.

If you plan to install this ECO, follow the instructions listed below.

1) Locate U44 (74LS74) and remove it from its socket. Carefully bend out pin 9 and replace the IC back into its socket. (Make sure it does not touch resistor R23.)

2) Locate U18 (74LS32) and remove it from its socket. Carefully bend out pin 11 and replace the IC back into its socket.

3) On the "solder side" of the board, solder a short jumper wire (26 to 30 gauge) between pins 9 and 10 of U44.

4) On the same side of the board, solder a short jumper wire between pins 1 and 8 of U46.

5) Attach a small white sticker to the board that says "ECO 179".

Be careful not to short the jumper wires to any other pins or traces, and use good soldering practices.