II. THE 8085 MICROPROCESSOR BOARD (MPU-B)

II. THE 8085 MICROPROCESOR BOARD (MPU-B)

A. THEORY OF OPERATION

1. INTRODUCTION

The MPU-B is a central processing board for the S-100 bus based on the 8085 microprocessor. In addition to the processor, it contains:

- 1K program PROM/ROM
- 1/4 K RAM
- Power on jump
- Serial I/O
- Parallel I/O
- Timers
- 5-level interrupts

The 1K byte PROM/ROM socket is provided with jumpers to configure it for 1K or 2K byte PROMs or 1K, 2K, or 4K byte ROMs. A 1K ROM is included with basic system support firmware. Under software control, the ROM can be located at 0000, or at D800. Power on clear initializes the ROM at 0000 to serve as a power on jump. The processor can write to the RAM at 0000 even when executing code in the ROM at 0000. A custom programmed PROM in this location can initialize the system, load programs in the entire 65K memory space (or more with the IMM installed), and then disable the PROM while jumping to the program start location. The PROM can be re-enabled at any time to use code stored in it, either at location 0000 or at location D800. The PROM enable is delayed to permit a jump or call to a routine even from a program executing directly underneath the PROM location. A system reset automatically re-enables the PROM at 0000.

There are 256 bytes of RAM provided on the board for minimum stand-alone systems, and for variable & data storage at a fixed location for the loaders and monitor in ROM or custom programs in PROM. It can be software enabled/disabled, so that loader and monitor variables do not take up user memory space. It is located at D000, and is enabled/disabled at the same time as the PROM at D800. It is not enabled by the PROM being at 0000, however the PROM may be enabled at both locations simultaneously if the RAM is desired with the PROM at 0000.

The ROM included provides power on system initialization, automatic baud rate select if the serial interface is used, and complete monitor facilities. The monitor includes loaders for floppy disk, casette tape, and paper tape, as well as extensive machine language entry and debugging features. The monitor will automatically run from either the serial interface or a keyboard and VIO. The loaders and other routines can be called from user programs.

The serial I/O provides complete serial interfacing capability. True RS-232 levels are driven for data and handshaking lines both in computer end and terminal end configurations. No jumpering is needed. A 20 mil current loop interface is included for driving ASR-33 type devices. Baud rates are software selectible at any standard or non-standard rate up to 9600 baud in asynchronous mode, or 56K baud in synchronous mode. The input data ready, output

buffer empty, and sync detect status flags can be switched to interrupt the processor, and/or used in a polled status mode. For software compatibility, serial port addressing is the same as an MIO at 00 or 10, or both ports of an SIO at 00, or the first port of an SIO at 10.

The parallel I/O provides a complete 8-bit parallel port, with handshaking and latching available in both in and out directions. The input data ready and output buffer empty status flags can be switched to interrupt the processor, and/or used in a polled status mode. For compatibility, the connector has the same pin definitions used on the MIO or PIO, and the addressing permits a keyboard to look like the serial input of an MIO at 00, or the second port of an SIO at 10. While the parallel port is used by the monitor for keyboard input, the monitor can also use the serial port. The parallel port is otherwise free to be used for any other parallel interfaced device such as the AP-44 mini-printer.

Three 16 bit timers are provided. All three are completely software programmable and readable. They are memory mapped at D100-D103, and are software enabled/disabled along with the RAM and PROM in D000. One timer is normally dedicated to provide the baud rate clock for the serial interface, but can be used differently when serial I/O is not needed. All three timers have clock inputs of 2 MHz, with one timer's clock switch selectible to be either 2 MHz or the output of another timer. This permits a programmable interrupt or periodic interrupt period from microseconds to more than 30 minutes. Both real time clocks and time of day clocks are easily implemented. The outputs of the two timers not used for serial clock can be switch selected to interrupt the processor at count termination, or periodically, depending on the software programmed mode of operation. The timer, once programmed and started, does not have to remain enabled (with the RAM & PROM) to operate. Interrupts will occur regardless of enabled status, or it can be momentarily enabled to read the current counts. The baud rate clock continues to run with the timer disabled from memory space. The two free timers can be used to provide software controlled baud rate clocks for serial interfaces on the MIO or SIO.

The MPU-B can be used either with or without a CP-A front panel. The power on jump and monitor eliminate the need for a front panel for any program operations. A CP-A may be used for hardware development or maintenance.

All address decoding on the MPU-B is done in a fusible link ROM, so that all the specific addresses mentioned may be changed for special applications, including memory mapping the serial or parallel I/O or I/O mapping the timers or any portion of the RAM. Special monitors can be installed in the program ROM socket to support OEM applications and special applications packages. Switching the RAM, PROM, and timers out of the address space facilitates implementing larger, more sophisticated systems by removing all address conflicts.

or

By adding power supplies and a terminal, the MPU-B forms a complete computer system with everything necessary to use it as an evaluation system or develop small machine language games or applications.

2. MPU-B CIRCUIT DESCRIPTION

The IMSAI MPU-B processor board interfaces an 8085 microprocessor chip to the S-100 bus. In addition to the processor and S-100 bus interface, the MPU-B has parallel and serial I/O, PROM/ROM, RAM, timer, and necessary address decoding. Power is regulated on the board. The processor runs at 3.0 MHz, and runs 8080 machine code along with a couple added instructions. Thus all existing 8080 code is 100% software compatible with the 8085 and MPU-B, paying due attention to the specifics of I/O device attachment. The I/O ports included on the MPU-B have been addressed such that IMSAI software will run without any changes, at the same time allowing switch selection of either the serial port (for a terminal) or the parallel port (for a keyboard & VIO) as the system I/O device. Both serial and parallel I/O are always available for systems which use both.

Even though the processor runs 1 1/2 times as fast as a standard 8080, the S-100 bus signals have looser timing requirements due to the 8085's more efficient use of the bus. The clock lines phase 1 and 2 run at the higher 3 MHz rate, however the CLK line is run at 2 MHz for use by existing S-100 boards that depend on a 2 MHz timing source. Most other S-100 bus lines retain their old definitions. Although the timing of some of the signals is slightly different from the 8080, the chip was designed to replace 8080's in existing systems. Any properly designed S-100 boards should experience no problems running with the MPU-B.

Most of the circuitry interfaces to a local bidirectional bus, which is connected to and defined by the 8085 pins. The circuitry in each of the blocks will be described.

3. 8085 PROCESSOR

The 8085 processor section of the board contains only a little circuitry beyond the 8085 chip itself. A 6 MHz crystal is connectec to the on-chip clock generator. All other system timing is generated from the 3 MHz clock out pin on the 8085. A power-on reset time constant of 20 milliseconds is generated by resistor PN2 and capacitor PN3 connected to the schmitt trigger reset input. The delay allows ample time for power supply voltages to stabilize before the 8085 is permitted to start running. Diode PN4 insures that if power is lost even for a short time, the 8085 will come back up running properly. When the +5 volt line falls below the schmitt's lower threshold, capacitor PN3 will be discharged quickly through diode PN4. When power is re-applied, the system begins to run only after PN3 charges to above the schmitt's upper threshold. Any manual system reset also discharges PN3, and the 8085 resumes running only after PN3 charges back to the upper threshold through PN2.

To insure that nothing connected to the bus is falsly activated during a hold or halt operation (during which the 8085 bus is three-stated), the three strobe lines (/WR, /RD, & /INTR) are held inactive by pullup resistors. The IO/M line is also pulled up to indicate an I/O status during hold or halt states, to prevent any of the on-board memory mapped circuitry being enabled, since in one mode of operation they would respond to a write strobe generated external to the MPU-B.

The 8085 is powered only by +5 volts.

The four RST lines into the 8085 have been arranged to permit use of interrupts for basic system I/O without the use of a priority interrupt card or any jumpers. One line is reserved for interrupts from a mass storage device (TRAP, @RST4.5) such as a floppy disk. The other three can be connected to the on-board I/O by activating DIP switches. The interrupts have

been segregated. RST7.5 is used for the extra timer sections, RST6.5 is used for output data buffer empty on both the serial and parallel interfaces, and RST5.5 is used for input data ready on both parallel and serial interfaces, and for sync. character detect on the serial interface. Open collector logic was used throughout the interrupt section, and jumper pads provided to connect these lines to the Vlx lines on the S-100 bus. This permits connecting other interrupt devices to the same lines without a priority interrupt board. If a priority interrupt board is added, it can drive the 8 standard RST locations, and the MPU-B I/O can continue to use the RST lines unique to the 8085.

4. S-100 BUS INTERFACE

Data Out:

The local address/data bus (AD0-AD7) from the 8085 always appears on the data out bus unless an external board pulls data out disable low (/DODSBL, pin 23). It is driven by 8 sections of 8T97 three-state bus drivers.

Data In:

The three-state bus drivers connecting the data in bus to the local address/data bus are always held disabled unless one or the other of the two input strobes are active (/RD or /INTA). This condition is part of the definition of /IDIDSBL. These input data drivers are also held disabled by either an external board pulling data in disable low (/DIDSBL, pin 21), or by other conditions causing an active signal on internal data in disable (/IDIDSBL). /IDIDSBL is active whenever any of the on-board memory or I/O is enabled, or whenever an input strobe is not present.

Address 0-7:

The local address/data bus is latched into PN5 at the trailing edge of ALE (address latch enable). At this time address bits 0 through 7 are on the A/D bus. The three-state driver outputs of the 8212 (PN5) drive the lower half of the address bus directly.

Address 8-15:

The upper half of the address bus (A8 through A15) is driven by 8T97's directly from the internal A8-A15 address bus. Both the 8T97 drivers and the 8212 drivers on the address bus can be disabled by an external board pulling address disable low (/ADDRDSBL, pin 22).

Data and Address Disables:

/DODSBL and /ADDRDSBL are normally driven only during a DMA transfer, when the MPU-B must be removed from the bus so that the DMA system may access the memories. Note that the address and data drivers on the MPU-B are not bidirectional; thus DMA operations to/from the on-board memory and I/O are not enabled.

Address timing:

Both the high and low half of the address bus appear early in T1, and since the 8212 is a gated latch, there is ample setup time to latch the full address at the trailing edge of ALE (PSYNC, pin 76) on an external card

5. STATUS

Status on the 8085 is encoded differently than on the 8080. In order to generate the lines needed to run the S-100 bus, the 8085 status is used as address inputs to a ROM. The pattern in the ROM translates the 8085-type status at the inputs to 8080-type status at the outputs. The 74S288 ROM used (PN6) has three-state outputs with sufficient drive to directly run the S-100 bus. An external board can disable the status drivers by pulling status disable low (/STSDSBL, pin 18), which will remove chip select from PN6.

Status timing:

The status appears at the same time as the address in T1, rather than having to be latched at a later time, as in the 8080. It remains valid until the corresponding time in the proceeding T1. There is ample setup time to permit latching the status with the trailing edge of ALE (PSYNC, pin 76) on an external card.

Status differences from the 8080:

Most of the status looks identical to that of the 8080 on the MPU-A. Two status lines, however, are not among the states presented by the 8085.

One of these is interrupt acknowledge (SINTA, pin 96). The 8085 does not uniquely identify an interrupt acknowledge cycle until the time that the /INTA input strobe appears. Until that time, the status is the same as an instruction fetch status. As a result, the MPU-B interrupt acknowledge status appears on the S-100 bus as an instruction fetch from the beginning of the cycle to the start of the input strobe. At that time it changes to an interrupt acknowledge status, enabling the RST or CALL op-code to be presented by the interrupting board (PIC-8, IMM, etc.). This reduces the access time available to the interrupting board to approximately 280 nsec, which is sufficient since no memory chip access time is involved.

The other status line is not decoded at all by the 8085. The stack operation status line (SSTACK, pin 98) is held not true (low) permanently (the signal can be opened should it be desired to drive it with another board). Since this line has not been used, its absence will not be a problem.

6. CONTROL STROBES

There are two input strobes on the 8085, which are OR'ed together and inverted to form a single positive input strobe (PDBIN) for the S-100 bus.

The 8085 signals /WR, ALE, and HLDA are used directly to drive S-100 signals (respectively) /PWR, PSYNC, and PHLDA. For systems without a front panel, the S-100 signal MWRITE is generated from /PWR and SOUT. The driver for MWRITE can be disabled by moving the dipswitch FP-/FP to the FP position (this permits the front panel to drive this line instead of the MPU-B). Provisions are also made for permanently disabling the MWRITE line for systems in which it is not used.

Wait signal generation:

The S-100 signal PWAIT is not generated by the 8085. This signal has been used on many S-100 bus boards to generate a single wait state for memories between 500 nsec and 1 usec, so the MPU-B includes a circuit to synthesize the required timing for PWAIT. PN7 is a latch which puts the inverse of the RDY line onto PWAIT; when the 8085 is receiving a not ready signal, an active (high) PWAIT signal is put on the S-100 bus. In order to match the timing of the 8080 WAIT, PWAIT should not become true (high) until the end of T2 (beginning of TWAIT). Since RDY will normally be driven low during T1, another signal is needed to prevent PWAIT from going high too soon. PN8 is used to delay ALE at the beginning of each cycle to provide a signal to delay the start of PWAIT.

During T4, T5, and the first part of the next T1, PWAIT can return high if the memory board still returns PRDY = PWAIT. This is of no consequence since the 8085 does not test RDY during this part of a cycle.

Control signal disable:

The five control lines will be disabled if an external board pulls control disable low (/CCDSBL, pin 19). This is normally done only during a DMA operation. If the DMA board waits for PHLDA before disabling the control bus, the 8085 will have just completed returning the strobes (PDBIN & PWR) to the inactive state. If the DMA board waits longer, or during a halt state, the MPU-B will continue to hold the strobes inactive. It is up to the DMA card to insure that the strobes cannot reach an active state during the transition of bus control.

7. CLOCKS

Clock out (COUT) from the 8085 provides a 3 MHz clock which is put on the S-100 phase 2 line (pin 24). This signal is inverted and used to provide phase 1 (pin 25). PN9, PN10 and PN11 form a divide by 3/2 circuit to produce a 2 MHz signal from the 3 MHz clock. The output waveform is a 2 MHz, 33% duty cycle (high) pulse train which is put on the CLK line (pin 49) for use by S-100 boards which need a 2 MHz timing source. Some S-100 boards may have originally used phase 1 or 2 for a timing source and these will need a "blue wire" change. Do not confuse the use of phase 1 or 2 as a strobe with their use as a timing source.

8. CLEAR

Positive going reset out (RESOUT) from the 8085 is inverted and used to drive power on clear (/POC, pin 99). The output of the driver is also used to drive external clear on the S-100 bus through a germanium diode. The diode permits a front panel, if present, to lower the external clear line (/EXTCLR, pin 54) without affecting the /POC line. With no front panel, there is no differentiation between the two clear lines.

The clock and the clear lines are driven all the time.

9. CONTROL INPUTS

Reset:

The /PRESET signal (pin 75) connects directly to the power on reset network described in the 8085 processor section. PN2 and PN3 provide enough time constant that a mechanical switch may be connected directly between /PRESET and ground, as long as the contact bounce is

shorter than approximately 10 msec.

Hold:

/PHOLD is held inactive by a 1K pullup, and inverted to drive HOLD on the 8085. HOLD is sampled on the trailing edge of phase 2, but does not have to be externally synchronized. The hold line is used primarily by DMA interfaces to cause the 8085 to cease executing at the end of the current instruction, and release the bus so that a DMA interface can gain control of the bus.

Ready:

RDY is high when both S-100 signals PRDY (pin 72) and XRDY2 (pin 12) are high. The S-100 signals are pulled inactive wth 1K resistors so that they need not be driven. If they are not driven, a constant ready is presented to the 8085 and it proceeds executing at full speed.

PRDY is used by memories, I/O cards, or interrupt controllers which need to delay the 8085's sampling of input data or cause it to hold output data longer. XRDY2 is used by the front panel (if present) to hold the 8085 from continueing or to allow it to single step only. Either line pulled low will stop the 8085 between the next T2 and T3.

10. INTERRUPTS

The S-100 interrupts enabled line (INTE, pin 28) has no equivalent in the 8085. It is pulled true with a 1K resistor, for any interrupt boards which look at it. There should be no effect on the operation of those boards.

The interrupt request line is pulled inactive with 1K, and drives INTR on the 8085 through an inverter. There are 4 interrupt lines on the 8085 which do not exist on the 8080. Jumper pads are provided so that these may be jumpered to any of the 8 S-100 priority interrupt request lines. This eliminates the need of a priority interrupt board for up to 5 levels of interrupt; the four 8085 restart lines and the standard interrupt request. (If the interrupt request is used without an interrupt board, no one will be driving the bus at interrupt acknowledge time, and the processor will read an FF opcode, which is an RST7.

The four 8085 restart lines are pulled inactive with 1K resistors, so they need not be jumpered. Of the four, only the TRAP line (non-maskable RST 4.5) comes jumpered (to V17, pin 11). The other three can be connected to the on-board I/O via DIP switches.

11. S-100 BUS INTERFACE TABLE

Following is a summary of the standard S-100 bus signals including:

S-100 pin number
High true (H) or low true (L)
Driven/received by the MPU-B (*) jumperable (X)
S-100 signal name

1	– *	+8 VOLTS POWER		51	– *	+8 VOLTS POWER	
2	– *	+16 VOLTS POWER		52	– *	-16 VOLTS POWER	
3	H	XRDY		53	L	/SSW DSB	
4	L	VI0		54	L *	/EXTCLR	
5	L	VI1		55			
6	L	VI2		56			
7	L	VI3		57			
8	L X	VI4		58			
9	L X	VI5		59			
10	L X	VI6		60			
11	L *	VI7		61			
12	H *	XRDY2		62			
13	H	AA15		63			
14	H	AA14		64			
15	H	A18		65			
16	H	A16		66			
17	H	A17		67			
18	L *	/STSDSBL		68	H *	MWRITE	
19	L *	/CCDSBL		69	?	PROTECT STATUS	
20	?	MEM.UNPROTECT		70	?	MEM.PROTECT	
21	L *	/DIDSBL		71	H *	RUN	
22	L *	/ADDR DSBL		72	H *	PRDY	
23	L *	DODSBL		73	L *	/PINT	
24	H *	02 (PHASE 2)		74	L *	/PHOLD	
25	H *	01 (PHASE 1)		75	L *	/PRESET	
26	H *	PHLDA		76	H *	PSYNC	
27	H *	PWAIT		77	L *	/PWR	
28	H *	PINTE		78	H *	PDBIN	
29	H *	A5		79	H *	A0	
30	H *	A4		80	H *	A1	
31	H *	A3		81	H *	A2	
32	H *	A15		82	H *	A6	
33	H *	A12		83	H *	A7	
34	H *	A9		84	H *	A8	
35	H *	DO1		85	H *	A13	
36	H *	DO0		86	H *	A14	
37	H *	A10		87	H *	A11	
38	H *	DO4		88	H *	DO2	
39	H *	DO5		89	H *	DO3	
40	H *	DO6		90	H *	DO7	
41	H *	DI2		91	H *	DI4	
42	H *	DI3		92	H *	DI5	
43	H *	DI7		93	H *	D6	
44	H *	SM1		94	H *	DI1	
45	H *	SOUT		95	H *	DI0	
46	H *	SINP		96	H *	SINTA	
47	H *	SMEMR		97	L *	/SWO	
48	H *	SHLTA		98	H *	SSTACK	
49	H *	CLK (2MHZ)		99	L *	/POC	
50	– *	GROUND		100	– *	GROUND	

12. ADDRESS DECODE

The address decoder on the MPU-B uses a 512x4 bit ROM (PN12), a dual 2 to 4 decoder (PN13), and several small gates. It produces chip enable signals for all the on-board memory and I/O circuits (PROM, RAM, timer, serial I/O, parallel I/O, and control port).

Address decode ROM:

The 9 inputs to the 512x4 rom (PN12) are the 8 high order address bits and the IO/M status line. The address decode portion of the MPU-B does not decode the low order address bits. Thus on board memory circuits either have low order address decoding at the circuit (PROM, RAM) or appear at multiple locations due to incomplete address decoding (timer). The state of address bits 8 through 15 and the I/O operation / memory operation status line (IO/M) is decoded by the ROM pattern. Since greater than 4 circuit enables are needed, the output of the ROM is encoded. The enables are divided into two groups: PROM, RAM, timer, and parallel I/O, serial I/O, control port. Output bit 4 low indicates that one of the enables in group 1 is active (PROM, RAM, or timer). Output bit 3 low indicates that one of the enables in group 2 is active (parallel I/O, serial I/O, or control port). Both bits 3 and 4 high indicate that the address and IO/M presented to the ROM are not in the address space of any of the on-board circuits. The ROM is encoded so that output bits 3 and 4 are never both low. Each of these bits is used as an enable to one half of the dual 2 to 4 decoder.

There is one time when an on-board enable may be indicated by the address and IO/M incorrectly. That is during an interrupt acknowledge. The 8085 appears as though it were going to execute the next instruction fetch at interrupt acknowledge time, except that a /INTA input strobe appears instead of a /RD strobe. In order to avoid a bus conflict and permit the interrupt board to send the interrupt op-code to the 8085, all memory and I/O circuits must recognize this condition and stay off the bus. To allow for this situation, INTA (the inverse of the /INTA strobe) is connected to the ROM chip enable input. Normally this is low, enabling the ROM; but during an interrupt acknowledge cycle, it goes high at data input time. ROM output bits 3 and 4 are pulled high by resistors when the ROM is disabled, and neither half of PN13 is enabled.

Output bits 1 and 2 are encoded to indicate which circuit enable is present, according to the following table:

Group 1:

21	
11	PROM enabled if POJM is true
10	PROM normal location
01	RAM
00	timer

Group 2:

21	
11	Parallel port
10	Serial port
01	System port
00	Control port

In the standard ROM, group 2 is all I/O ports and group 1 is all memory mapped. The board is not restricted to this scheme, but if a different ROM is made for a custom application, some restrictions need to be considered.

Any circuit mapped as I/O cannot use the /MWRT write strobe because it is not active during I/O writes. A jumper is provided so that the two devices using this strobe (timer and RAM) can be switched to /IWR, the internal bus write strobe. If this is done, the circuits will run either memory mapped or I/O mapped, but the front panel (if present) cannot write into the memory mapped circuit since it uses /MWRT. The processor has no restrictions on writing.

The Serial I/O, Parallel I/O, and control port run from the internal strobes and can be memory or I/O mapped or both. They cannot be jumpered to respond to memory mapped writes from CP-A's. The program PROM/ROM is a read only device, and can be memory or I/O mapped. Writes to these addresses affect whatever system memory or I/O has been installed off the MPU-B. The RAM and timer can be memory mapped or I/O mapped, but if either is I/O mapped, the jumper provided must be changed to /IWR instead of /MWRT. This will prevent front panel writes to either as noted above.

When a device is memory mapped on the MPU-B, it takes up a minimum of 256 bytes of memory space. On board I/O device decoding can be complete since the complete I/O address appears on A8 – A15. Note that some of the I/O circuits use address bits from A0 – A7 for further decoding of port address. In these cases the corresponding address bits in A8 – A15 are made don't care bits in the address decode ROM.

Dual 2 to 4 decoder:

The decoder translates ROM output bits 1 and 2 into individual enable lines according to the table above. Section one decodes group 2 (I/O and control) and is enabled only by ROM output bit 3. This group of crcuits is a permanent feature of the address space of the 8085, it cannot be disabled.

Group 1 (memory and timer) is decoded by section 2. ROM output bit 4 is one of the enables for this section; the other enable input is driven by the signal IMEM. One of the four group 1 enable outputs from PN13 is not used. Instead, the output is decoded separately and ANDed with a different enable. Instead of being enabled with IMEM, it is enabled with POJM (Power On Jump Mode), and is ORed with the normal program ROM/PROM enable to provide a separate control of enabling te program ROM/PROM at this location. Typically the POJM PROM address will be 0000H to provide for system start-up. The IMEM PROM address could be anywhere, though D800H is standard. Since POJM and IMEM are program controllable, group 1 circuits can be removed from or returned to the 8085's memory space.

Group 1:

Whenever IMEM is true, and the address is correct, the timer, RAM, or regular PROM is enabled. The timer and RAM are enabled for both read and write operations. The PROM enable is ANDed with read status in PN16, so that the PROM is enabled only during reads. During writes to the PROM's addresses, the PROM is disabled and the write occurs in whatever memory the system has installed at that address. If POJM is true, the PROM is enabled at its second location (typically 0000-07FF). Again, only PROM reads are decoded.

Group 2:

The serial, parallel, or control port is enabled directly whenever the appropriate address is decoded. The system port output is provided with two DIP switch positions to allow it to enable either the serial or the parallel port. Only one of these switch positions should be activated at a time to avoid bus conflicts of reads from these devices. The switches allow either I/O method to be run from the same "system" port addresses so that software need not be changed when the basic I/O device is changed.

Standard ROM addressing:

The standard addressing for the MPU-B circuits, as effected by the address decode ROM are:

Mapped as memory:

PROM @ POJM	0000H – 07FFH
PROM normal	D800H – DFFFH
RAM	D000H – D0FFH
timer	D100H – D103H

Mapped as I/O ports:

serial	4H,5H and 12H,13H
parallel	(14H,15H) EQUIVALENT
system port	(02H,03H) EQUIVALENT
control port	F3H

13. CONTROL PORT

The control port is used to enable or disable the group 1 circuits (PROM, RAM, and timer), and the second PROM location. The second PROM location is a separate enable at a second address of the same (single) PROM chip, so that it can appear at two places (standard is 0000 at power on jump time and D800 for normal operation).

Whether the second PROM location is enabled is controled by data bit 0 in the control port. The main PROM location, the RAM, and the timer are enabled by data bit 1 in the control port. In both cases, a data bit of "0" enables the circuits, a "1" disables them. These two data bits are each latched into their own 74LS195 shift register when the control port is enabled, by putting the shift registers into load mode. The data bit is latched only into bit A, at the trailing edge of the /WR strobe. After being latched, the data bit is shifted by successive /RD or /WR strobes, until it appears at the output (QD,/QD). This takes three cycles after the control write cycle. After the output takes on its new state, it remains static because the J and /K inputs are tied to load A with QA on each shift. Internal power on clear (/IPOC) is tied to both shift registers' clear inputs, and POJM and IMEM are both taken from /QD, so that at power on time (or after any reset) all the circuits controlled by these signals are enabled.

14. RAM

There are 256 bytes of RAM provided on the board, in two 8111's. The lower 8 bits of address are connected directly to the chip, along with the local bidirectional data bus. The interfacing involves no extra chips since the 8111's were designed to go directly onto a bidirectional bus. Chip select is driven by /REN from group 1 of the address decoder. The output disable pin on the memories is driven whenever either the 8085 read/write status indicates a write cycle, or the front panel (if present) drives data in disable (/DIDSBL) in preparation for a write from the front panel. The write strobe is driven by /MWRT, which is derived from MWRITE whether it is driven by the 8085 or the front panel.

If the address decode ROM addresses any of the RAM as I/O (the standard ROM does not) then the RAM write strobe should be jumpered to /IWR with the jumper provided since MWRT is not active during I/O cycles. This also prevents a front panel from writing directly into the RAM. Note: the same /MWRT - /IWR jumper affects both the RAM and the timer circuits.

15. PROM

Read only program memory on the MPU-B is provided in a single ROM/PROM socket. Three jumpers are provided so that the 24 pin socket can be configured to run a 1K or a 2K prom, or a 1K, 2K, or 4K ROM. The standard chip provided is a 2K ROM with a system monitor and bootstraps.

NOTE: Care must be exercised when installing a ROM/PROM other than the one provided, since it is possible to damage some types of ROM/PROM if the jumpers provided are in the wrong position. Refer to the user's guide.

All the ROM/PROM chips were designed to attach directly to a bidirectional bus, so there is no external logic onvolved. The chip is enabled with /PEN from the address decode logic when the high-order address is appropriate.

16. TIMER

An 8253 chip is used as the timer circuit. No periferal gates are needed since it attaches directly to the bidirectional bus. Address lines A0 and A1 are decoded by the 8253 to select among its 4 control bytes. Since only A8 - A15 are decoded for the timer enable (/TEN), A2 - A7 are "don't care" bits in the address. As a result, the timer's 4 locations appear in multiple locations throughout the block of 256 decoded by the address decode logic. The low four locations only are called out throughout the specifications. The read strobe (/RDL) is a local read strobe generated by anding /DIDSBL with the 8085's /RD. This permits the front panel (if present) to remove the read strobe during a front panel write into the (memory mapped) timer. The write strobe (/MWRT) is derived from MWRITE, whether it is driven from the front panel or the 8085.

If the address decode ROM is changed to include accessing the timer as I/O ports, then the write strobe needs to be jumpered to /IWR at the location provided, since /MWRT does not occur during I/O writes. This will prevent the front panel from writing into the timer, even if the timer is also memory mapped at the same time. Note: The same /MWRT - /IWR jumper affects both the RAM and the timer.

The 8253 has three timers in it. Timer 0 is dedicated to providing a clock signal for the serial interface. The timer 0 out line is connected to the serial interface. Timer 0 gate is pulled high, and timer 0 clock is driven by the 2 MHz CLK2. This timer would normally be programmed to the auto-repeat square wave mode, although the system can use it for other timing functions at times when the serial interface is not needed.

Timers 1 and 2 are undedicated and available for system use. Timer 1 is clocked by the 2 MHz CLK2 line, while timer 2 can be switch selected to either run at 2 MHz, or cascaded from the output of timer 1. In the cascaded mode, the total count time of the two timers is over 30 minutes. The outputs of both timer 1 and 2 can be switch selected to cause an interrupt at terminal count (or periodically). These lines were run to RST7.5 (which is edge sensitive) to permit use of modes with short terminal output pulses. The gates on both timers 1 and 2 are pulled high.

The gate and output lines of all three timers are brought to solder-pads to permit easy modification of the board for custom applications where it is desired to drive these lines with external signals.The timers' function is not affected by whether its address decoding is enabled (IMEM). They will continue to count and produce interrupts as programmed until they are re-enabled and the mode changed. The interrupts are driven through open-collector gates so that other devices may also be connected to the same interrupt line if desired.

17. PARALLEL I/O

Data:

Two 8212's are used for the parallel I/O data path. One latches output data and drives the parallel output pins on J4 continuously. The other will buffer the input data and will also latch it if the strobe line is driven by the input device. Both the input and output data paths can use the handshaking available in the 8212, and the input data ready signal or the output buffer empty signal can both switched onto the interruppt pins for interrupt operation of the parallel port. The same two sgnals are available at the parallel I/O status port. The handshaking only runs when the respective external device drives the strobe lines. If the strobe lines are not driven, the input data ready and the output buffer empty will always indicate false. When handshaking is used, the falling edge of the strobe causes the respective status bit to go true, and latches the data on the input port. The processor's response (read or write data) causes the respective status bit to return false. The 8212 chips are selected by /KEN and A0=0, as well as the appropriate read or write strobe.

Status:

The parallel I/O status port is implemented with four sections of a 74LS367 to drive the internal data bus, and some gates to enable the 3-state buffers with /KEN=0 and A0=1 and /RDL=0, which is during the read strobe to the port address decoded at the address decoding. The output port output buffer empty signal is connected to data bit 0. The input port input data ready isconnected to data bit 1. These are the same bits as the respective status signals from the 8251 serial interface chip, to simplify software drivers which run both devices. Also driven during parallel status reads are two upper data bits. Bit 7 is driven with IMEM, and bit 6 is driven with POJM. These two bits are high when the respective on-board functions are enabled. This permits interrupt routines to determine the state of the board so they can return correctly.

Power:

Ground, +5 volts regulated, +18 volts unregulated, and -18 volts unregulated are provided on the parallel I/O connector. The +5 volt regulator was designed to be able to supply up to several hundred milliampers of power to the parallel connector, however it should be realized that any power drawn results in higher dissipation at the regulator. If too much current is drawn or to little ventilation provided, the +5 volt regulator may shut itself down.

Connector:

The parallel I/O connector is a 26-pin dual-row header, suitable for; connection to flat cable connectors. The pin definitions result in the same signal/wire connections as used in the MIO. Pin 26 is unused to permit the use of 25 wire cable and D type 25 pin connectors at the rear of the cabinet. The schematic shows the 26 pin connector pin-numbers in the arrow boxes, and the 25 pin connector pin-numbers in the circles. These pin numbers assume the standard pin numbering for D type connectors, standard flat cable pin numbering for the 26 pin connector (so that the flat cable wires counted from the edge correspond to the pin numbers), and a cable wired to connect pin 1's.

18. SERIAL I/O

The serial I/O port uses an 8251 serial interface chip.This USART will support asynchronous or synchronous serial communications to 9600 or 56000 baud respectively. It is interfaced to standard EIA RS-232 levels and pins for providing either the computer or terminal end of a serial line. It also will drive an ASR-33 Teletype type 20 mil. current loop interface. The output buffer empty and the input data ready or sync. detect status bits can be switched to drive the interrupt lines.

The 8251 is connected directly to the on-board bidirectional bus. A0 drives the command/data input, the other 7 address bits are decoded at the address decoder to form /SEN. The clock input is driven with 2MHz, and power on clear and on-board read and write strobes drive the appropriate inputs.

The EIA RS-232 signals from the USART (including RTS, CTS, DTR, DSR, R and T Data, R and T synchronous clocks) are connected to the serial I/O connectors through RS-232 receivers and drivers. DIP switches permit selecting the on-board receive clock for asynchronous mode, or an off-board clock for synchronous data mode. DTR and CTS inputs are tied true with resistors to eliminate the need for jumpers when they are not driven. Carrier detect is tied true.

The current loop interface will switch and sense a 20 mil. loop compatible with the ASR-33 Teletype. The loop current source is provided, and any other current source should be removed. The current loop connections must not be referenced to ground. Provision for filtering the input data is made to permit reliable operation of the typically noisy ASR-33.

Serial connectors:

There are two serial I/O connectors on the board. They are both 26 pin dual row connectors suitable for flat cable connection to the EIA standard D type 25 pin connectors. Assuming a cable wired so that pin 1's on the two ends are connected, the 26 pin connector pin-numbers are indicated on the schematic in the arrow boxes, and the EIA standard RS-232 pin numbers

are indicated in the circles. The two connectors are arranged so that one is configured as a computer end of an RS-232 line, and the other is configured as a terminal end. Depending on the use of the system, the cable from the RS-232 line should be plugged into one connector or the other, with no jumpers being necessary. The board is not designed to use both connectors at once.

The current loop connections are made to the computer end connector only. If for some reason it is desired to operate in current loop mode as a terminal, a custom cable can be made to effect the proper connections.

Since the S-100 bus does not include a chassis ground, the chassis ground line from the RS-232 connectors is terminated in a .040" pin socket. If it is desired to connect this line at the board, a wire with a .040" pin can be attached to the chassis, and plugged into this socket.

19. POWER REGULATORS

Plus and minus 12 volts for the RS-232 drivers, pullups, and the PROM/ROM are regulated by TO-220 style three terminal regulators. They regulate only a small amount of current, so they are heat-sunk on the foil on the board. Minus 5 volts for the PROMs is regulated by another TO-220 regulator in series with the -12 volt regulator, to more evenly distribute the heat dissipation.

The plus 5 volt regulator is a TO-3 style three terminal regulator, mounted on a heatsink. It is required to supply the 5 volts for this board, plus any additional current taken by an external device connected to the parallel I/O port. The MPU-B 5 volt current drawn is typical 1.3 amps, maximum 1.6 amps.

Tantalum capacitors are provided at the inputs of each regulator for stability, and .1 uF ceramic bypass capacitors are provided throughout the board among the logic chips.

B. USER GUIDE

In this section we describe how to set up and use your MPU-B processor board. The level of detail is intended to supply the assembly language system programmer and OEM user with all the information needed to use the board. There are several option jumpers which must be changed if you are going to operate the MPU-B differently than set-up at the factory. You will probably not want to change these jumper options. There are a number of switch-options for functions which are often set differently in separate systems. These should be checked before operation to insure that the desired mode of operation is enabled. If your MPU-B has arrived as part of a factory-assembled system, it will be set-up to run. In that case you may wish to delay referring to the option set-up portion of this chapter until you wish to change the factory settings.

For normal operations, an operating system program, a high-level language, or an applications program is loaded into the computer from mass storage. The procedure for initializing the system and loading the desired program is described in the Sections IV and V of the PCS-80/30 User Manual. The option selection and machine-language operation of the MPU-B board only is described here. This section is primarily of use to those programming in Assembly or machine language, and during initial installation.

1. MPU-B OPTION JUMPERS

Some of the options provided on the MPU-B will be used only in special situations, such as when a system is being customized for a special configuration or special application. These options have been provided as jumper-pads on the printed-circuit board, and they have a small trace on the board connecting these pads in the standard factory configuration. Unless you have a special situation, you will not need to set or change these jumpers.

Three of these jumpers allow configuring the MPU-B program ROM socket to accept different size ROM's. Assembled boards are jumpered at the factory for the ROM installed. Other options are described in the Program ROM/PROM section. Note: INCORRECT JUMPERS CAN DAMAGE THE ROM, refer to the appendix on PROM installation.

There are jumpers to remove bus drive from the signals SSTACK, INTE, and MWRITE. These jumpers appear on the schematic but are not described since they are primarily for factory use. Should a knowledgeable user wish to change one of these and need information, he should contact the factory.

A .040" pin socket is provided for connecting RS-232 line AA (chassis gnd.). Solder pads for wires or pin sockets are provided for 8085 signals SID and SOD, as well as timer signals GATE and OUT for all three timers.

Jumper pads permit connecting any of the eight vectored interrupt lines on the bus to any of the four restart inputs or the interrupt request line on the 8085. The standard configuration is described in the USING INTERRUPTS section (II-B,8).

2. MPU-B OPTION SWITCHES

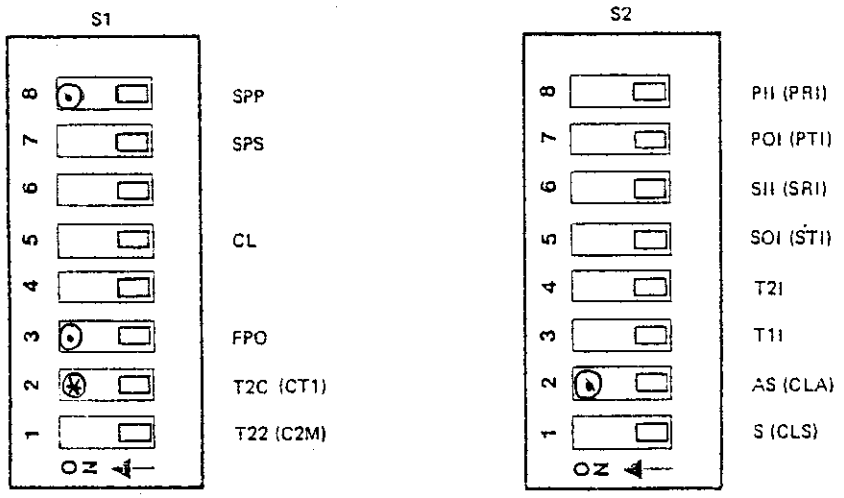
All the options which might normally be changed are provided with switch positions to enable the board to be quickly and easily configured as desired. A brief description of the switches is provided here for reference.

⊙ SHOWS CONFIGURATION AS DELIVERED.
SWITCHES ON

⊛ CHANGE - RLV

FIGURE II-1

SWITCH LOCATIONS
TOP OF BOARD



OPTION SWITCHES

SYSTEM PORT SERIAL ENABLE (SPS IN/OUT)
SYSTEM PORT PARALLEL ENABLE (SPP IN/OUT)

Only one of this pair of switches should be enabled at a time. They cause either the serial or parallel I/O port (or neither) to respond to the system port addresses (02 data, 03 status). (They will continue to respond to their standard port addresses.) If both switches are disabled, then ports 00 through 03 are unused, permitting an external I/O board (such as an MIO) to be installed using the bottom 4 ports.

ASYNCHRONOUS CLOCK ENABLE (CLA IN/OUT)
SYNCHRONOUS CLOCK ENABLE (CLS IN/OUT)

Only one of this pair of switches should be enabled at a time. One of these must be down to enable use of serial receive. They connect either the internal baud rate clock (for asynchronous mode) or an externally generated clock signal (for synchronous mode) to the USART's receive clock input.

CURRENT LOOP ENABLE (CL IN/OUT)

If current loop operation is desired, the CL switch should be enabled. If synchronous mode is to be used, then the CL switch should be disabled.

SERIAL TRANSMIT BUFFER EMPTY INTERRUPT (STI IN/OUT)
SERIAL RECEIVE CHAR. READY INTERRUPT (SRI IN/OUT)

Neither, either, or both of these may be enabled as desired. The first will cause an interrupt to occur (RST6.5) whenever the serial I/O transmit is enabled and the transmit buffer is empty. A jump to the interrupt service routine must be located at 0034H. The second will cause an interrupt to occur (RST5.5) whenever the serial I/O receive is enabled and a character is waiting to be input to the processor. A jump to the interrupt service routine must be located at 002CH.

PARALLEL TRANSMIT BUFFER EMPTY INTERRUPT (PTI IN/OUT)
PARALLEL RECEIVE CHAR. READY INTERRUPT (PRI IN/OUT)

Neither, either, or both of these may be enabled as desired. The first will cause an interrupt to occur (RST6.5) whenever the parallel I/O output buffer empty status is true (set by handshake strobe). A jump to the interrupt service routine must be located at 0034H. The second will cause an interrupt to occur (RST5.5) whenever the parallel I/O input data ready status is true (set by the same strobe which latches data). A jump to the interrupt service routine must be located at 002CH.

TIMER 1 INTERRUPT ENABLE (T1I IN/OUT)
TIMER 2 INTERRUPT ENABLE (T2I IN/OUT)

Neither, either, or both of these may be enabled as desired. Each switch will cause an interrupt to occur (RST7.5) when its respective timer output goes low (terminal count or 1/2 terminal count depending on programmed mode). A jump to the interrupt service routine must be located at 003CH. Timer 0 is the baud rate generator, and does not connect to the interrupts. Note that RST7.5 is an edge triggered interrupt, so that timer modes which produce a terminal pulse may be used without danger of missing the interrupt.

TIMER 2 CLOCK 2 MHz ENABLE (C2M IN/OUT)
TIMER 2 CLOCK T1 OUT ENABLE (CT1 IN/OUT)

Only one of these two switches should be enabled at a time. One or the other must be down to provide a clock input to timer 2. When the first one is enabled, timer 2 counts at a 2 MHz rate (2 million counts/second). With the second switch enabled, timer 2 increments one count at each high to low transition of the output of timer 1. Cascading the timers this way and enabling only timer 2 interrupt permits extended periods to be timed without software intervention.

FRONT PANEL INSTALLATION (Panel IN - FP1)
(Panel OUT - FP0)

This switch must be placed in the enabled position in any system with a CP-A installed, and must be placed in the disabled position in any system without a CP-A. Even when temporarily inserting a front panel for maintenance, this switch should be moved to IN. Note that the sense of IN and OUT is reversed from the other switches.

3. ADDRESS DECODING AND CONTROL

There are six circuits on the MPU-B which appear in the memory or I/O space of the processor. They are:

1. Firmware ROM
2. 256 byte RAM
3. Three TIMERS
4. 8 bit parallel I/O port
5. Serial I/O port
6. MPU-B control port

Some of the six circuits can appear in multiple (redundant) locations in memory or I/O space. Some can be made to appear/disappear from memory space using the control port.

For descriptive purposes, the functions are divided into three groups according to how the sections are enabled/disabled by the control port. The allocation of memory - I/O space to these six functions is as follows:

GROUP	BYTES	FUNCTION	ADDRESS
0	2K	ROM for system initialize	0000-07FF
1	2K	ROM for monitor functions	D800-DFFF
1	256	RAM for monitor variables	D000-D0FF
1	4	TIMERS & serial clk.	D100-D103
1		RESERVED do not use	D104-D7FF
2	2	PARALLEL I/O	14-15
2	2	SERIAL I/O	12-13
2	2	SERIAL I/O	04-05
2	2	SYSTEM PORT for SER. or PAR.	02-03
2	1	CONTROL PORT for address mode	F3

The four digit numbers refer to hexadecimal memory addresses, and the two digit numbers refer to hexadecimal port addresses. The ROM in groups 0 and 1 is actually the same ROM which can appear at two different locations. The serial and parallel I/O can also appear at multiple locations.

CONTROL PORT AND SOFTWARE MEMORY CONTROL

The MPU-B control port is a single I/O mapped port address (F3). It is a write-only port and only two of the eight data bits are active. The two bits control whether group 0 or 1 circuits appear in the memory space or not. When they are 0, the circuits appear in the memory space; when they are 1, the circuits do not appear in the memory space and whatever memory is installed in the system at that location then appears.

Control port:

Group 0 control: bit 6

BIT 6 = 0 2K ROM appears at 0000-07FF
BIT 6 = 1 System memory (as installed) appears at 0000-07FF

Group 1 control: bit 7

BIT 7 = 0 2K ROM appears at D800-DFFF
256 byte RAM appears at D000-D0FF
TIMERS appear at D100-D103

BIT 7 = 1 System memory (as installed) appears at D000-DFFF

CONTROL PORT DEFAULT AND SWITCHING RESTRICTIONS:

At power on time and at any reset, control port bits 6 and 7 are both set to 0 so that the firmware ROM, RAM, and timers are all enabled. The two control bits operate independantly, and there are no restrictions on acceptable combinations. Either group may be enabled or disabled at any time or sequence desired. The control bits are changed with an out F3.

If a read is attempted from the control port, the resulting data will be FF. Due to the logic implementation, this will also result in both groups 0 and 1 being disabled. That is, a read from the control port is equivalent to a write with data bits 6 and 7 equal to 1's.

CONTROL PORT EFFECT:

When the 2K ROM is enabled, all memory reads from those locations read from the ROM. Thus the program in the ROM can be executed. All memory writes from those locations, however, will succeed in writing data to the system memory (if installed).

Both the RAM and timers are read/write devices. When they are enabled, all memory reads and writes from those locations are from/to the RAM and timers. During the time they are enabled, no access is possible to system memory at those locations.

The combination of devices and addresses labeled group 1 above cannot be enabled/disabled separately. They can only be switched in and out as a group. As a result, when group 1 is enabled, system memory from D800-DFFF is accessible for writes only, and system memory from D000-D7FF is not accessible at all.

When group 0 (firmware ROM) is enabled, system memory at 0000-07FF is not accessible for reads, since reads now come from the firmware ROM. Writes, however, are not restricted from system memory at all by group 0 being enabled. If group 1 is disabled to remove its write restrictions, group 0 can be enabled and still permit writing to the entire 65K memory space. Thus a loader or other system initialization routine executed from the firmware ROM can load/initialize the entire memory as needed.

POWER-ON JUMP

When the MPU-B is run without a front panel, it will come up running starting at location 0000 at power-on time or reset time. Since the firmware ROM defaults to being enabled at the same time, the MPU-B will begin to execute the code stored in the ROM at 0000. The code contained in the ROM can range from a disable group 0 and jump to monitor sequence, to a complete automatic system initialize and load from external mass storage, ending with a disable firmware ROM and jump to system start.

The factory installed ROM functions vary for assembled systems. The ROM with kits or assembled boards sold separately is the same as that supplied with the PCS-80 system model 30 unit. The factory installed ROM can be 1K bytes or larger, depending on the system.

CONTROL PORT ENABLE / DISABLE TIMING:

When either group 0 or 1 is either enabled or disabled by program control, there is a 3 CYCLE DELAY before the commanded change in status takes place. This is to allow for a 3-byte instruction fetch, such as a jump or call, before the ROM appears/disappears. If a 3-byte instruction is not desired, the programmer must take precautions to allow for the status change being delayed 3 cycles. The delay is for 3 true 8085 cycles, and is not affected by any DMA cycles which may occur during this time. Interrupt cycles, however, will be counted, so if the 3-byte instruction is in 0000-07FF or D800-DFFF, then the programmer needs to insure that no interrupt will occur. Otherwise the instruction fetched will not be what the programmer

expected.

Example of a POWER-ON JUMP to location E000

ROM ADDR.	CONTENT		Notes:
0000	MVI	A,0C0H	Get byte to turn off group 0 and 1
0002	OUT	F3	Output byte to control port
0004	JMP	E000	Jump to location E000 (Group 0 and 1 will both become disabled after reading all the jump and before the fetch following the jump.)

CONTROL PORT STATUS

The current status of the group 0 and 1 enable can be read by the processor. It appears in the status byte from the parallel port, bits 6 and 7. A 0 indicates enabled, and a 1 indicates disabled. The capability to read this status permits a system to use interrupt routines which can sense the system status, modify it as necessary to service the interrupt, and return it to its original state before returning. The function of the status bits is the same as the function of the control bits. A status read from the parallel pot address followed by an output of the result to the control port will result in no change.

MEMORY STATUS SENSE (in parallel I/O status byte)

Bit 6 = 0 ROM at 0000 enabled

Bit 6 = 1 System memory 0000-07FF enabled

Bit 7 = 0 ROM at D800 enabled
RAM at D000 enabled
Timers at D100 enabled

Bit 7 = 1 System memory D000-DFFF enabled

SERIAL and PARALLEL I/O PORT ADDRESSING

The parallel port is always accessible at I/O port 14 (data) and I/O port 15 (status). If the option switch SPP (System port parallel) is enabled, then the parallel port also appears at I/O port 2 (data) and 3 (status). When it is enabled at both locations, accessing it by either set of addresses or a mixture of the two produces identical results. For example, it would make no difference if data was written to port 02 or port 14, whether the status was read from port 03 or port 15.

The serial port is always accessible at I/O port 12 (data) and I/O port 13 (status) as well as always available at I/O port 04 (data) and 05 (status). If the option switch SPS (System pi serial port also appears at I/O port 02 (data) and 03 (status). As in the parallel port, there is no differentiation between the multiple ports at which it may be accessed.

The port addresses 12,13,14,15 are the same configuration of data and status as would be presented by an SIO 2-2 board addresses at 10H. The port addresses 02,03,04,05 are the same configuration as an SIO 2-2 board addressed at 00.

The typical consol I/O addresses for IMSAI software are 02 and 03. The MPU-B permits switching either the serial port (for a terminal) or the parallel port (for a keyboard to be used with a video display) to respond to these locations. A serial device is expected at ports 04 and 05 by some IMSAI software. New firmware intended for the MPU-B alone would use the permanent addressing at 12,13,14 and 15.

4. PROGRAM ROM / PROM

The firmware ROM supplied with the MPU-B will vary for different systems. For details of functions included in the ROM, the appropriate USER MANUAL should be consulted. For boards purchased separately, the ROM included is that used in the PCS-80 model 30. It provides system initialization and loaders with a limited monitor.

For a system modified for a special application, the firmware ROM may be replaced with a PROM or a ROM containing firmware to support that application. This would be done if the system was not going to be used as a general purpose machine, but rather for only the special application. Since complete system initialization, load, and error recovery can be put in the PROM, special application systems may be easily made which involve no operator actions other than the power switch to begin operation.

Jumper options permit the use of 1K (2708 type) or 2K (2716 type) Eraseable PROM's, and also 1K, 2K, or 4K ROM's for systems produced in higher quantities.

CAUTION: Factory jumpers have been installed for the ROM supplied. Other PROM's or ROM's may be DESTROYED if installed without changing the jumpers. Refer to Appendix 2.

When a 1K PROM/ROM is installed without changing the address decode ROM, it occupies 2K of memory space. It will appear in both the upper and lower halfs of the 2K space reserved for the 2K ROM. If a 4K ROM is installed, the address decode ROM must be changed in order to permit access to the upper half of the 4K ROM. Contact the factory for details.

ADDRESSING:

The ROM/PROM can appear in two locations. It will appear at 0000 when group 0 is enabled (see CONTROL PORT section above). It will also appear at D800 when group 1 is enabled.

This is not two different ROM's, but rather it is the same ROM appearing in two different locations. It can appear at both simultaneously if both groups 0 and 1 are enabled. In that case a read from 0000 will get the same data as a read from D800. A read from 01E4 will get the same data as a read from D9E4, etc.

The programs run from each area (0000-07FF and D800-DFFF) must share the space in the ROM. The program running in 0000-07FF will need to be at the bottom of the ROM to start properly at location 0000 at power-on time. A program running in the D800-DFFF area would then be ORG'ed at D800 plus the length of the program at 0000.

For example, a system initialization and boot from floppy disk routine may take 140H bytes. It would be ORG'ed at 2000 and placed in the bottom 140 bytes of the ROM. A set of standard utilities can then be put in the ROM to run at the upper location. It would be ORG'ed at D940H and could extend up to DFFF. It would be placed in the ROM following the initialization and boot routine.

The ROM may be used at either location at any time, but the two locations have different facilities. Running from only 0000 permits a routine to have WRITE access to the entire 65K memory space of the 8085 for complete system initialization. Loader variable storage is limited to the registers in the processor or reserved system memory, however. When running from D800-DFFF, WRITE operations are restricted from D000-D7FF, but 256 bytes of RAM are available (D000-D0FF) which are not part of the system memory and are switched out when the ROM is switched out. For a routine needing complete access as well as larger variable storage, it is possible to run the routine at 0000-07FF, enable the RAM / upper ROM location before variable access, and disable the RAM / upper ROM location before accessing system memory at D000-DFFF.

5. RAM MEMORY

There are 256 bytes of read/write memory on the MPU-B, which can be enabled or disabled as part of group 1 circuits. When it is enabled, it is at D000-D0FF. The upper part of the RAM area is used for variable storage by the monitor program in the ROM, and the rest of the RAM area is free for user storage or programs. If the monitor is not to be used, nor any routines called in the ROM, then the entire 256 bytes may be safely used.

If the RAM has been disabled (through the control port), then it is not available for use until re-enabled. While it is disabled, whatever memory is loaded into the system at that location responds to the processor; when it is enabled, the memory in the system at the RAM address is disabled and the RAM on the MPU-B responds to the processor. Any time the program enables or disables the RAM, there is a 3 CYCLE DELAY before the status change takes place, to allow for a jump or call if desired. The RAM is enabled/disabled along with the other functions located in DXXX. Its enable is not separable.

Information is not lost in either the 256 bytes of RAM or in the system memory it blocks as the program enables or disables it. At power up time or on reset, the RAM is enabled.

6. SERIAL I/O PORT

The MPU-B has a serial I/O port implemented with an 8251 USART. Both synchronous and asynchronous operation are possible. It is configured to allow both current loop and RS-232 operation with no jumper changes. The baud-rate clock is obtained from a TIMER section under software control. For details on the clock generation, see the TIMERS section (II-B,7).

CURRENT LOOP

The current loop interface provides 20-mil current loop including the current source. It is suitable for ASR-33 teletypes and devices with similar interfaces.

The MPU-B current source is referenced to ground. The interface to which it is connected must be isolated from ground to avoid possible damage.

The normal internal configuration for an ASR-33 teletype is with no current source and isolated from ground. However, it is possible that a unit would be jumpered to provide a current source. Before connecting a unit to the MPU-B, it should be checked to insure that the interface lines are isolated from ground. This can be done easily by measuring the resistance between the interface lines and the chassis in the ASR-33. A standard VOM is adequate, and the resistance measurement should indicate an open circuit.

The MPU-B current loop interface operates full duplex. The ASR-33 jumpers should be configured for full duplex if they are not already configured that way.

DIRECTIONS FOR JUMPERING

There is a terminal strip located at the right rear of the Teletype (ASR33). The terminal strip is behind a panel of square white plastic connectors and also connects to the TTY power cord. The terminals are numbered from 1 to 9. The connections required between the MIO and these terminals are shown in Table II-1. In addition to making these connections, it may be necessary to perform the following operations on your Teletype.

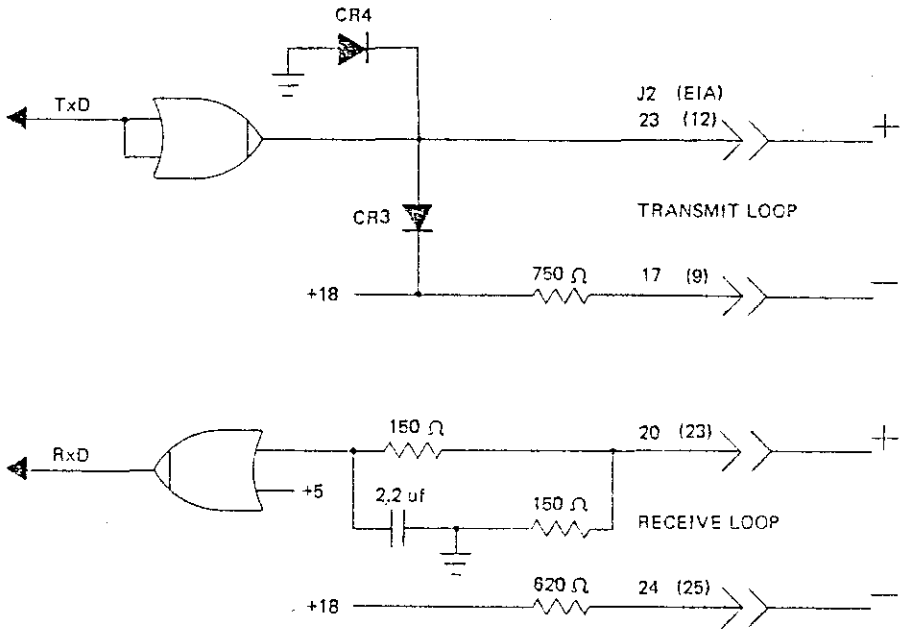
1. Full Duplex Operation - Move YEL/BRN wire from Terminal 3 to Terminal 5 and move WHT/BLU wire from Terminal 4 to Terminal 5.
2. Change receiver current level from 60 ma to 20 ma; move VIO wire from Terminal 8 to Terminal 9.
3. Change current source resistor to 1450 ohms. Locate the current cource resistor in front of the power supply and move the BLU wire to the tap labeled 1450.

TABLE II-1 CONNECTIONS FOR ASR-33

Signal Name	26 Pin Edge Connector	25 Pin EIA Connector	Terminal Strip
Current Loop Out +	20	23	7
Current Loop Out -	24	25	6
Current Loop In +	8	17	3
Current Loop In -	22	24	4

FIGURE II 2

CIRCUIT FOR CONNECTIONS TO ASR-33



The serial interface lines are brought to the rear of the cabinet through a flat cable to a standard 25 pin D type connector. The MPU-B end of the cable should be plugged into either J2 or J3. The current loop pins on J2 are configured the same as those on the SIO board, while the current loop pins on J3 are configured the same as those on the MIO board. On J2, the current loop pins are numbers 9 (+) and 12 (-) for the output loop, and numbers 25 (+) and 23 (-) for the input loop. An external cable should connect these pins to the ASR-33, attaching the wires according to the diagram above. On J3, the corresponding pins are 23 (+) and 25 (-) for output and 17 (+) and 24 (-) for input. Unless a cable wired for the MIO current loop pin configuration is already available, it is suggested that the SIO configuration on J2 be used.

OPTION SWITCHES

To run current loop, option switch CL must be in the enabled (IN) position. Current loop mode may only be used with asynchronous mode, for which switch CLA must be enabled, and switch CLS must be disabled.

RS-232

Devices with RS-232 interfaces can be plugged directly into the 25 pin D type connector at the rear of the cabinet. The MPU-B end of the cable can be plugged into either J2 or J3 for different configurations. It should be plugged into J2 (Computer or modem end configuration) to run terminal type devices, or into J3 (Terminal end configuration) to run with a modem.

Handshaking and clock lines are included in the RS-232 interface as well as send & receive data. They are terminated so that no jumpers are needed whether or not the interface to be driven implements these lines. The lines implemented on the MPU-B are:

RS-232 INTERFACE LINES

PIN	LINE	NAME	FUNCTION
1	AA	CG	Chassis ground
7	AB	SG	Signal ground
2	BA	TxD	Transmit data
3	BB	RxD	Receive data
4	CA	RTS	Request to send
5	CB	CTS	Clear to send
6	CC	DSR	Data set ready
20	CD	DTR	Data terminal ready
8	CF	CD	Carrier detect
15	DB	TxC	Transmit clock
17	DD	RxC	Receive clock

The RS-232 signal names and functions refer to the operation from the terminal end of the line. For example, TxD is data transmitted from the terminal device to the computer device. Thus pin 2 is an output on a terminal device, and an input on a computer device.

All the signals named above are fully implemented except for carrier detect, which is driven true in computer configuration but cannot be sensed for use in a terminal configuration, and chassis ground, which can be jumpered to ground or connected to chassis ground with an external wire and a .040 pin. Only transmit and receive data are required to operate the RS-232 interface in asynchronous mode; and only transmit and receive clock need to be added to operate in the synchronous mode. The other lines can be left unterminated or connected to the peripheral device as desired.

OPTION SWITCHES for RS-232

No jumpers are needed to configure the interface for any situation. However, there are three option switches which must be in the correct position.

Option switches CLA and CLS configure the clock lines to run the interface either asynchronous or synchronous. If it is desired to run the interface in the asynchronous mode, then switch CLA should be enabled (IN) and switch CLS should be disabled (OUT). This connects the clock in the receive section to the internal baud rate generator. If operation in the synchronous mode is desired, then switch CLS should be enabled (IN), and switch CLA should be disabled (OUT). This connects the receive clock to the external (RxC) clock line. The transmit clock is always connected to the internal baud rate generator.

Option switch CL only applies if operation in the synchronous mode is desired. For synchronous mode, make sure switch CL is disabled (OUT).

PROGRAMMING THE SERIAL INTERFACE

GENERAL

The serial interface is capable of supporting virtually any serial data technique currently in use with RS-232 or ASR-33 type current loop, including IBM bi-sync. Before data can be sent and received through the interface, the mode and baud rate must be initialized through software. Refer to the TIMER section for details on initializing the baud rate clock. Programming the mode and data operations are described in this section.

SETTING THE MODE

The mode byte output to the 8251 defines the general operational caracteristics of the serial interface. It MUST follow a reset, either a power on reset, system reset (front panel button), or a software reset. The first byte output to the control port address of the 8251 (same as the status read port address) will be interpreted as a mode byte.

If a change in the operational mode is desired under software control, the 8251 may be reset by the following sequence:

```
SUB     A,A     Set accumulator to 00
OUT     13      Output 00 to 8251 control port
OUT     13      Second time
OUT     13      Third time
MVI     A,40    Put 40 into accumulator
OUT     13      Output to 8251 to reset
```

Port 05 could be used, as could (with option switch SPS enabled) port 03. Outputting three zeros is done to insure that even if (worse case) the 8251 is expecting the first of two sync bytes, the programming sequence is taken far enough that the chip will for sure interpret the 40 as a command byte. A command byte of 40 will reset the 8251, and must be followed by a mode byte.

The mode byte for asynchronous operation is defined as follows:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
S2	S1	EP	PEN	L2	L1	B2	B1

BAUD RATE FACTOR, B2 & B1:

B2	B1	Function
0	0	Synchronous Mode
0	1	1 X clock rate
1	0	1/16 X clock rate
1	1	1/64 X clock rate

CHARACTER LENGTH, L2 & L1:

L2	L1	Character length (data bits)
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

PARITY ENABLE, PEN:

PEN	Function
0	Parity disabled
1	Parity enabled

PARITY SENSE, EP:

EP	Function
0	Odd parity
1	Even parity

STOP BITS, S2 & S1:

S2	S1	Function
0	0	Invalid code, do not use
0	1	1 stop bit
1	0	1.5 stop bits
1	1	2 stop bits

An asynchronous mode byte which will enable running with a model 33 TELETYPE and most CRT terminals is AE. The normal baud rate factor used is x16, and this is assumed in the TIMER section on setting the baud rate.

For synchronous mode the mode byte is defined as follows:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SCS	ESD	EP	PEN	L2	L1	0	0

Bits 0 through 5 are the same as defined under asynchronous.

EXTERNAL SYNC DETECT, ESD

ESD	Function
0	Sync is detected by the 8251 internally
1	An externally detected sync is expected

SINGLE CHARACTER SYNC, SCS

SCS	Function
0	Double sync character mode
1	Single sync character mode

Note that the sync detect pin is pulled true, so that if external sync detect is programmed, the 8251 is told immediately that sync has been detected, regardless of the data. The normal setting for this option will be for internal sync detect.

Either one or two sync characters, as programmed by the mode byte, must be sent to the command port following the mode byte. If the two sync characters are different, the first one given to the 8251 should be the first one expected on the data line.

COMMAND BYTE

Once the mode byte and sync bytes, as required, have been output to the command port, the 8251 will interpret all further bytes to the command port as command bytes until the chip is reset.

The command byte is the same for synchronous and asynchronous, and is defined as follows:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
EH	IR	RTS	ER	SBRK	RxE	DTR	TxEN

TRANSMIT ENABLE, TxEN

TxEN	Function
0	Transmit section disabled
1	Transmit section enabled

DATA TERMINAL READY, DTR

DTR	Function
0	RS-232 line DTR set FALSE
1	RS-232 line DTR set TRUE

RECEIVE ENABLE, RxE

RxE	Function
0	Receive section disabled
1	Receive section enabled

SEND BREAK CHARACTER, SBRK

SBRK	Function
0	Normal send operation
1	Transmit data held continuously true

ERROR RESET, ER

ER	Function
0	No effect
1	Reset error flags PE, OE, and FE

REQUEST TO SEND, RTS

RTS	Function
0	RS-232 line RTS set FALSE
1	RS-232 line RTS set TRUE

INTERNAL RESET, IR

IR	Function
0	No effect
1	Resets 8251, Next command byte must be mode

ENTER HUNT MODE, EH

EH	Function
0	No effect
1	Enables search for sync character(s) in data stream

In asynchronous mode, the normal command byte desired is 37. Once this command is given, no further writes to the command port are normally needed. The serial port will send and receive data through the data port, signalling the processor that it is ready for the next byte through a status read from the command port (or through an interrupt if the option switches are enabled).

In synchronous mode, the normal command byte desired is B7. The serial port will begin scanning the incomming data for a match to the sync character(s). Once a match is found, the status or interrupt will indicate as each character is received. Between each message in synchronous mode, it is often desired to re-synchronize. This can be done by sending another command byte (B7) to the command port to cause the 8251 to re-enter the sync hunt mode.

STATUS BYTE

The status byte can be read by the program at any time, by an INPUT from the command port address. It contains the transmit and receive ready bits, error flags, and other status information needed for the program to interface to the serial port.

The status byte is defined as follows:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DSR	SYNDET	FE	OE	PE	TxE	RxRDY	TxRDY

TRANSMITTER READY, TxRDY

TxRDY	Function
0	Transmitter buffer full, waiting for transmission
1	Transmitter buffer ready to accept next character

RECEIVER READY, RxRDY

RxRDY	Function
0	No further characters received yet
1	Next received character ready to be input to program

TRANSMITTER EMPTY, TxE

TxE	Function
0	Character currently being sent by USART
1	Last character transmission has been completed

PARITY ERROR, PE

PE	Function
0	No parity error has occurred since the last reset command (bit ER in the command byte)
1	A parity error has occurred since the last reset (PE does not inhibit further operation of the USART)

OVERRUN ERROR, OE

OE	Function
0	No character has been lost since last reset command
1	The program has failed to read a character before the next one was received. Further operation is not inhibited; however, the overrun character is lost.

FRAMING ERROR, FE

FE	Function
0	No framing error has occurred since the last reset
1	A valid stop bit was not detected on at least one character since the last reset. FE does not inhibit further operation.

SYNC DETECT, SYNDET

SYNDET	Function
0	No sync byte(s) has been detected since the last status read (or reset operation).
1	A valid sync byte(s) has been detected. This status bit is reset by a status read.

DATA SET READY, DSR

DSR	Function
0	RS-232 line DATA SET READY is false
1	RS-232 line DATA SET READY is true

EXAMPLE INTERFACE ROUTINES

The following instruction sequence will set up the USART for asynchronous X-16 clock, 8-bit characters without generating or checking parity. This mode will run an ASR-33 or any standard CRT terminal. The sequence should be executed after a reset (either hardware of software as shown above) before any other I/O to the serial port is executed. It leaves both the transmitter and receiver sections running, and for normal operation no further mode or command operations need take place.

```
MVI     A,AE    Get mode byte
OUT     13      Output to serial port command address
MVI     A,37    Get command byte
OUT     13      Output to serial port command address
```

The next routine can be called each time another input character is desired. NOTE: Only one character is stored, so if the program asks for characters slowly enough that the operator gets ahead more than one character, then characters will be lost. If too much processing is needed for fast enough operation, then a whole line can be accepted from the operator at once. The

program can then process the input line, prompting the operator when it is ready for the next line. Alternatively, interrupt operation could be implemented, with the interrupt service routine putting the characters in a FIFO buffer, and the main program picking up new characters from the buffer. This routine is not suitable for interrupt operation, but rather the case where the program is to wait for the next character. The routine will return with the character in the accumulator.

```
CHRIN   IN      13      Read serial input status
        ANI     02H     Mask all but the input byte ready status bit
        JZ      CHRIN   Return and wait for character
        IN      12      Read character
        ANI     7FH     Mask off upper bit if present (for ASCII only)
        RET             Return with character in accumulator
```

The third routine is used for sending characters. It is called once for each character sent, with the character in the accumulator. If multiple characters are being sent, as in a message, another part of the program must pick up the characters one at a time and call this routine.

```
CHROUT  MOV.    B,A     Temporarily store the character in B register
CHOUT1  IN      13      Read serial status
        ANI     01H     mask off all but transmitter ready status bit
        JZ      CHOUT1  Return and wait until ready
        MOV     A,B     Get character back in accumulator
        OUT     12      Send character to serial port
        RET             Return to main program
```

OPERATION HINTS

1. Output of a command to the USART will overwrite any character which is stored in the buffer waiting for transfer to the parallel to serial register. This can be avoided by waiting for TxRDY to be true before sending a command if transmission is taking place (bit 0 of the status byte). It is also possible to disturb the transmission if a command is sent while a SYN character is being generated by the device (in synchronous mode if the software fails to respond to TxRDY). When running in synchronous mode, commands should be transferred only when a positive going edge is detected on the TxRDY status.

2. RxE (Receiver Enable command bit) does not control the receiver, it only masks the RxRDY (Receive Character Ready) status bit. It is possible for the receiver to have one or two characters ready at the time the receiver is enabled. Especially in the synchronous mode, the program should read the USART data twice (it is not necessary to check status) to ensure that the next character ready was received after the receiver was enabled.

3. TxEN (Transmitter Enable) and RTS (RS-232 line) should remain asserted until the last bit of the last character has been shifted out of the USART. A delay of 1 msec after the occurrence of TxE (status byte bit 2) is usually sufficient. Loss of CTS or TxE will immediately clamp the serial output line. Also note, the loss of TxE in the synchronous mode clamps the data at a SPACE instead of normal MARK. This does not occur in asynchronous mode.

4. TxE can momentarily go true while data (including USART generated SYNs) is transferred to the parallel to serial register. If TxE is being used, check it for several consecutive status reads.

5. A BREAK results in a string of characters with framing errors. If reception is to be continued after a BREAK, note that the last character received during a break, including any framing error associated with it, is indeterminate.

7. TIMERS

The MPU-B has three 16-bit timers with programmable mode and count. Program access to the timers is through 4 memory-mapped locations (D100-D103), which are enabled/disabled by control port F3 bit 7. They are part of the memory-mapped functions in block D which are all simultaneously enabled/disabled through the control port. Block D may be left normally disabled, and enabled momentarily to access the timers. If this mode is used, note the 3 CYCLE DELAY in both enable and disable. (Refer to ADDRESS DECODE AND CONTROL section.)

The timers are address as follows:

TIMER READ MODE

ADDRESS	FUNCTION
D100	Read counter No. 0
D101	Read counter No. 1
D102	Read counter No. 2
D103	No function - read FF data

TIMER WRITE MODE

ADDRESS	FUNCTION
D100	Load counter No. 0
D101	Load counter No. 1
D102	Load counter No. 2
D103	Write mode byte

Since the timer access is memory mapped, data can be read or written to the timer with any of the many 8085 memory reference instructions. There are some restrictions, however, in the order of operations. These restrictions are described in this section, and must be followed for correct operation of the timers.

The three timers are completely independant of each other, with the single exception that an option switch will allow timer No. 2 to count the output pulses from timer No. 1. All timers count at a 2 MHz rate, except No. 2 when it is counting No. 1 output pulses. They may be programmed to count either in 16-bit binary, or in 4-digit Binary Coded Decimal (BCD).

Single counts or automatic reload and repeat modes are available. Switch options permit connecting the output of timer No. 1 and/or No. 2 to interrupt the processor to location 003C.

TIMER NO. 0

Timer No. 0 is normally dedicated entirely to providing a programmable baud rate clock for the serial I/O interface, but can be programmed for other functions during any time that the serial interface is not being used. No interrupt is provided for on timer No. 0 output, although it could be jumpered to an interrupt jumper pad.

TIMER CONFIGURATION

TIMER SECTION	COUNT MODE	INTERRUPT LOCATION	COUNT RATE	OPTION SWITCH SETTINGS
Timer 0	BINARY	no int.	2 MHz	none
Timer 0	BCD	no int.	2 MHz	none
Timer 1	BINARY	no int.	2 MHz	T1I OUT
Timer 1	BCD	no int.	2 MHz	T1I OUT
Timer 1	BINARY	003C	2 MHz	T1I IN
Timer 1	BCD	003C	2 MHz	T1I IN
Timer 2	BINARY	no int.	2 MHz	T2I,CTI OUT - C2M IN
Timer 2	BCD	no int.	2 MHz	T2I,CTI OUT - C2M IN
Timer 2	BINARY	003C	2 MHz	CTI OUT - T2I,C2M IN
Timer 2	BCD	003C	2 MHz	CTI OUT - T2I,C2M IN
Timer 2	BINARY	no int.	T-1 OUTPUT	T2I,C2M OUT - CT1 IN
Timer 2	BCD	no int.	T-1 OUTPUT	T2I,C2M OUT - CT1 IN
Timer 2	BINARY	003C	T-1 OUTPUT	C2M OUT - T2I,CT1 IN
Timer 2	BCD	003C	T-1 OUTPUT	C2M OUT - T2I,CTI IN

Except for the lack of interrupt on timer No. 0, and the switch selectable clock input on timer No. 2, The three timers are completely identical in capabilities. Each timer consists of a single 16-bit, pre-settable DOWN counter. The counter can operate in either binary or BCD, and each has a hardware gate and output. The exact function of the gate and output is configured by the mode bytes stored in D103. The timers can be used by the software without any external connections to the gate and output connections. However, they have been provided with solder pads for wires or pin sockets in case it is desired to use the gate or output for any external function. (For example, the outputs could be used to provide a programmable baud rate clock to another I/O board such as the SIO.)

The counters are fully independant and each can have separate Mode configuration and timing operation, binary or BCD, single count or repetitive. Special capabilities in the mode byte handle the loading of partial count values so that software overhead can be minimized for these functions. Commands and logic are included so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

PROGRAMMING THE TIMERS

The complete mode of operation is programmed by the software. A set of control bytes MUST be sent out by the processor to initialize each timer with the desired Mode and quantity information. These control bytes program the Mode, loading sequence, and selection of binary or BCD counting. Since the timers have no power on reset, they should be programmed even if they are not to be used immediately, especially if the interrupts are connected. A counter not to be used can be programmed to count to a short termination and stop.

All of the modes for each timer are programmed by writing bytes to D103. Each timer is individually set-up by writing a Mode byte into this location. The Mode byte format is:

D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

The four control fields are defined as follows:

SC SELECT COUNTER

SC1	SC0	Function
0	0	Select Counter No. 0
0	1	Select Counter No. 1
1	0	Select Counter No. 2
1	1	Not Used

RL READ / LOAD

RL1	RL0	Function
0	0	Counter Latching operation (see read procedure)
0	1	Read/Load Least significant byte ONLY
1	0	Read/Load Most significnt byte ONLY
1	1	Read/Load Least significant byte first, then Most significant byte second.

M MODE

M2	M1	M0	Function
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD

BCD	Function
0	Binary count (16 bits)
1	Binary coded decimal (BCD) counter - 4 decades

MODE DEFINITIONS

Many of the mode definition affect the detailed nature of the signal on the out pin of interest if used for external purposes. The program can only read count.

MODE 0: OUTPUT HIGH ON TERMINAL COUNT

The output will be initially low after the Mode set operation. After the count is loaded into the selected count register, the output will remain low and the counter will count. When terminal count is reached the output will go high and remain high until the selected count register is reloaded with the MODE.

If the counter register is reloaded during counting:

Loading the first byte stops the current count
Loading the second byte starts the nw count.

The gate input must be high to enable counting; if it is pulled low, counting will be inhibited.

MODE 1: PROGRAMMABLE ONE-SHOT

This mode needs an external connection to the gate to operate. After being set-up, the output will go low on the clock following the RISING EDGE of the gate signal.

The output will go high on the terminal count. If a new count value was loaded during the timeout, it will not affect the duration of this output-low time. The new count value will be used following the next RISING EDGE of the gate input. The current count can be read without affecting the duration of the one-shot pulse.

The "one shot" is retriggerable, i.e. the output will remain low for the full count after the latest rising edge of the gate input.

MODE 2: PROGRAMMABLE RATE PULSE GENERATOR

Divide by N counter. The output will be low for one period of the clock input. The period from one pulse to the next equals the number of clock pulses input to the count register. If the count register is reloaded between output pulses, the present period will not be affected, but the next period will reflect the new value.

The gate input, if pulled low, will inhibit counting and force the output high. When the gate input returns high, the counter will start from the initial count. Thus the gate input can be used to synchronize the counter.

When setting this mode, the output will also remain high until after the count register is loaded. Thus the output can also be synchronized through software.

MODE 3: SQUARE WAVE GENERATOR

This is the mode that would normally be used in timer 0 to generate a baud rate clock for the serial interface.

This mode is very similar to MODE 2, except that the output remains high for only one half the count, and goes low for the entire other half. If the count loaded in the register was odd, the output will be high for (N+1)/2 counts, and low for (N-1)/2 counts.

If the counter register is reloaded with a new value during counting, the new value will be reflected after the low to high output transition of the current count.

MODE 4: SOFTWARE TRIGGERED STROBE

After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then return high.

If the count register is reloaded between output pulses the present period will not be affected, but the next period will reflect the new value. If the gate input is pulled low, counting will be inhibited. If the counter register is reloaded while the count is inhibited by the gate input, the counter will begin counting the new number when the gate returns high.

MODE 5: HARDWARE TRIGGERED STROBE

The use of this mode requires an external connection to the gate input. The counter will start counting after the RISING EDGE of the gate input and go low for one clock period when the terminal count is reached. The counter is retriggerable, the output will not go low until the full count after the most recent rising edge of the gate input.

SUGGESTED MODES

Several of the modes require external connections to be made to the gate input, some are primarily useful for external connections to the output. For timer operation from software only (no hardware interaction external to the MPU-B board), the following modes are suggested:

Baud rate generation for serial port:

Timer 0, Mode 3

Note: Using the square-wave mode keeps pulse length times within limits for typical UARTS & USARTS (including the 8251 used on the MPU-B). To obtain the typical X16 baud rate clock, set the count value to:

COUNT = 2,000,000 / (BAUD RATE * 16)

This formula is in base 10. COUNT must be converted to hexadecimal to be loaded into the timer, unless the time is run in BCD mode! Round the calculated value to the next LOWER integer. This will result in the transmitted signal being slightly faster than the received signal, preventing occasional lost characters when echoeing all received characters at the maximum rate (as in reading paper tapes). The small percentage difference from nominal speed is insignificant for any standard rates. For operating the 8251 with synchronous data links, the clock should be X1, and the formula is:

COUNT = 2,000,000 / BAUD RATE

Again, this formula is in base 10. The resulting COUNT should be rounded to the nearest integer.

Some count values for common asynchronous baud rates (X16 clock):

BAUD RATE	DECIMAL COUNT	HEXADECIMAL COUNT	ERROR
9600	13	000D	-.0016
4800	26	001A	-.0016
2400	52	0034	-.0016
1200	104	0068	-.0016
600	208	00D0	-.0016
300	416	01A0	-.0016
150	833	0341	-.0004
134.5	929	03A1	-.0004
110	1136	0470	-.00032
75	1666	0682	-.0004
45	2777	0AD9	-.00028

Some count values for common synchronous baud rates (X1 clock):

BAUD RATE	DECIMAL COUNT	HEXADECIMAL COUNT	ERROR
56000	36	0024	+.0079
38400	52	0034	-.0016
19200	104	0068	-.0016
9600	208	00D0	-.0016
4800	416	01A0	-.0016
2400	833	0341	-.0004
1200	1666	0682	-.0004

Timer for tracking real time (as in a time of day clock, timed delay, timed intervals, etc.)

Timer 1, Mode 2 and Timer 2, Mode 2
(Option switch CT1 IN and C2M OUT, so that timer 2 counts output pulses of timer 1.)

A) Timer 1 and 2 both in binary mode, count values both set at maximum (0000). This forms a 32 bit counter with a count rate of 2 MHz. Maximum time is 2**32 periods of .5 microsecond (2147.48 seconds, 35.79 minutes). Timer 1 is the two low-order bytes, and timer 2 is the two high-order bytes. Only timer 2 need be reinitialized each time if an accuracy of +0/-.0328 seconds is sufficient.

B) Timer 1 in binary mode, count value set at 4E20 hex; Timer 2 in BCD mode, count value set at maximum (0000). This forms a 4 digit BCD counter counting .01 second intervals (timer 2), for a maximum count of 100 seconds. Only timer 2 need be reinitialized each time if an accuracy of +0/-.01 second is sufficient.

NOTE: In both A and B above, the counters are counting DOWN from the initial value. To calculate the actual time elapsed from initialization the binary value read from the two timers (A) or the BCD value read from timer 2 (B) must be subtracted from the initial value (0000). If the count read from timer 2 is equal to the minimum count (also 0000), then the maximum timer capacity has been exceeded, assuming care has been taken that it is not tested before one time increment (.0328 sec (A), or .01 sec (B)) has occurred.

LOADING THE MODE AND COUNT VALUES

The order of programming is quite flexible. Since each timer's Mode control byte has a unique address, loading the Mode control bytes is independant of sequence. Not only can the Mode control bytes be written in any sequence, but they can be separated by other operations, such as loading the count registers.

The loading of the count registers with the actual count value, however, must be done in the sequence programmed in the Mode control byte. There is no separate addressing for low or high count values, so the timers will assume the count value bytes output are in sequence according to the mode programmed. Since the three timers have unique addresses for loading count values, the order of timers programmed does not matter.

The one or two count value bytes do not have to immediately follow the associated Mode control byte. They can be loaded at any time after the Mode control byte has been written as long as the correct number of bytes are loaded in order. The two following examples are valid sequences:

D103	36	Mode byte - counter 0, two bytes, mode 3
D100	XX	Low order count value
D100	YY	High order count value
D103	A4	Mode byte - counter 2, High order only, mode 2
D103	36	Mode byte - counter 0, two bytes, mode 3
D103	79	Mode byte - counter 1, two bytes, mode 4, BCD
D100	XX	Low order count value, counter 0
D100	YY	High order count value, counter 0
D102	SS	High order count value, counter 2
D101	QQ	Low order count value, counter 1
D101	RR	High order count value, counter 1

READ OPERATIONS

In many timer applications it becomes necessary to read the value of a count in progress to make a computational decision. The counters contain logic that will allow the programmer to easily read the contents without disturbing the count in progress.

If the gate is being used to control the timer, and the programmer is assured that the counter will be disabled during the read, no special action is necessary. Reads from the appropriate address will be responded to with the current count value.

The first read contains the least significant byte.

The second read contains the most significant byte.

It is necessary to complete the entire reading procedure. If two bytes are programmed to be read then two bytes MUST be read before any loading command can be sent to the same counter.

READING WHILE COUNTING

In order for the programmer to read the contents of a counter without effecting or disturbing the counting operation while in progress, the timers have logic that can be accessed through a write to the mode register. When it is wished to read the contents of a selected counter "on the fly", a write is made to the Mode register with a code which latches the present count value into a storage register so that a stable count value may be read. The programmer then issues a normal read command to the selected counter to read the contents of the latched register. Note from the previous table that the latch command is affected by a Mode byte with bits 4 & 5 (RL1,RL0) equal to 00. The counter to be latched is selected as before. Note that it is still necessary to complete the entire read sequence as programmed.

INTERRUPTS FROM TIMERS 1 & 2

Option switches are provided to connect the outputs of timers 1 & 2 to interrupt 7.5 (to location 003C). Either timer or both may be connected. When both are connected, the program must check the contents of the two timers to determine which one issued the interrupt (i.e. which one has reached terminal count).

The outputs of these two timers have been inverted, so that an interrupt will occur when the output is LOW. This permits use to be made of modes 2 through 5, but not of mode 0 or 1. (Only for interrupts, all modes operate for program reads of count status.) Mode 0 will request an interrupt at the beginning of the count. Mode 1 will request an interrupt once the count is started by the signal on the gate terminal. The other modes will request an interrupt either at terminal count or 1/2 terminal count, depending on the mode. Repetitive modes will produce periodic interrupts. Since RST7.5 is edge sensitive, an interrupt will not be missed even though the termination count output is a short pulse. Also, there is no need to reset the counter in any way before interrupts can be re-enabled. NOTE: Since the outputs of timers 1 and 2 are simply ORed together to create the interrupt signal, an extended low output signal on one (such as during the second half of the count in mode 3) will mask any output change on the

other. If both timers are used in pulse output modes, no interrupts will be lost. It would still be possible for both to request an interrupt on the same cycle, which case must be resolved by the software.

Examples

baud rate clock for 110, 9600
auto baud rate select

8. USING INTERRUPTS

Provisions have been made so that the four levels of interrupt provided completely internally by the 8085 can be used for the MPU-B I/O circuits, or jumpered to any of the VI0 through V17 lines on the backplane for use by other peripheral boards. Switch options are used for connecting the on-board I/O to the interrupt lines.

INTERRUPTS FROM SERIAL AND PARALLEL I/O

Only three of the interrupt lines are used for the on-board I/O. The serial and parallel input status bits are OR'ed together and provided with a switch to the RST 5.5 line. If either the PRI switch (Parallel Rec. Interrupt) or the SRI switch (Serial Rec. Interrupt) is closed, a jump to an interrupt service routine must be located at 002CH. An interrupt will occur whenever the interrupt mask enables that interrupt and a receive character is ready.

If both switches are closed, it is up to the interrupt service routine to determine from the I/O status bytes whether the character which is ready is from the serial or parallel port (or possibly a character will be ready at both ports). Whenever the status bit (Receive Character Ready) is true, an intertupt will occur as soon as RST 5.5 is enabled. The request for interrupt will be reset by a data read from the active port.

The serial and parallel output ready status bits are similarly OR'ed together and provided with a switch each to the RST 6.5 interrupt line. The jump to the interrupt service routine must be located at 0034H. An interrupt will occur whenever the interrupt mask enables that interrupt and the transmit portion of the I/O interface is ready to accept the next character.

Again the interrupt service routine must resolve the ambiguity if both switches are closed.

The parallel I/O port cannot be used in the interrupt mode unless it is connected to a device which uses the handshaking mode of operation with the port, as it is the handshake strobes that set the respective ready bits.

INTERRUPTS FROM TIMERS 1 AND 2

Option switches are provided to connect the outputs of timers 1 and 2 to interrupt 7.5 (to location 003C). Either timer or both may be connected. When both are connected, the program must check the contents of the two timers to determine which one issued the interrupt (i.e., which one has reached terminal count).

The outputs of these two timers have been inverted, so that an interrupt will occur when the output is LOW. This permits use to be made of modes 2 through 5, but not of mode 0 r 1 (only for interrupts, all modes operate for program reads of count status). Mode 0 will request an interrupt at the beginning of the count. Mode 1 will request an interrupt once the count is started by the signal on the gate terminal. The other modes will request an interrupt either at terminal count or 1/2 terminal count, depending on the mode. Repetitive modes will produce periodic interrupts. Since RST 7.5 is edge sensitive, an interrupt will not be missed even though the termination count output is a short pulse. Also, there is no need to reset the counter in any way before interrupts can be re-enabled. NOTE: Since the outputs of timers 1 and 2 are simply OR'ed together to create the interrupt signal, an extended low output signal on one (such as during the second half of the count is Mode 3) will mask any output change on the other. If both timers are used in pulse output modes, no interrupts will be lost. It would still be possible for both to request an interrupt on the same cycle, which case must be resolved by the software.

OTHER INTERRUPTS

The TRAP interrupt (RST 4.5) is not connected to anything on the MPU-B board. It is provided with a traced-in jumper to V16 to be available to peripheral boards. It is a non-maskable interrupt and is the highest priority.

The interrupt request line is present in the 8085 as it is in the 8080. It is still available for use by an interrupt board such as the PIC-8 and in normal use (using RST-0 through RST-7) provides an additional 8 interrupt levels. It is also provided with a traced-in jumper to the V17 line.

APPENDIX 1

8085 INSTRUCTION SET

The following two pages are reproduced with the permission of Intel Corporation, Santa Clara California.

8085 INSTRUCTION SET SUMMARY

Mnemonic	Description	Instruction Code[1] D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock[2] Cycles
MOVE, LOAD, AND STORE										
MOVr1,r2	Move register to register	0	1	D	D	D	S	S	S	4
MOV M,r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r,M	Move memory to register	0	1	D	D	D	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	12
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	12
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	12
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	12
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	16
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	6
JUMP										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	7/10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	7/10
JZ	Jump on zero	1	1	0	0	1	0	1	0	7/10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	7/10
JP	Jump on positive	1	1	1	1	0	0	1	0	7/10
JM	Jump on minus	1	1	1	1	1	0	1	0	7/10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	7/10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	7/10
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	6
CALL										
CALL	Call unconditional	1	1	0	0	1	1	0	1	18
CC	Call on carry	1	1	0	1	1	1	0	0	9/18
CNC	Call on no carry	1	1	0	1	0	1	0	0	9/18
CZ	Call on zero	1	1	0	0	1	1	0	0	9/18
CNZ	Call on no zero	1	1	0	0	0	1	0	0	9/18
CP	Call on positive	1	1	1	1	0	1	0	0	9/18
CM	Call on minus	1	1	1	1	1	1	0	0	9/18
CPE	Call on parity even	1	1	1	0	1	1	0	0	9/18
CPO	Call on parity odd	1	1	1	0	0	1	0	0	9/18
RETURN										
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	6/12
RNC	Return on no carry	1	1	0	1	0	0	0	0	6/12
RZ	Return on zero	1	1	0	0	1	0	0	0	6/12
RNZ	Return on no zero	1	1	0	0	0	0	0	0	6/12
RP	Return on positive	1	1	1	1	0	0	0	0	6/12
RM	Return on minus	1	1	1	1	1	0	0	0	6/12
RPE	Return on parity even	1	1	1	0	1	0	0	0	6/12
RPO	Return on parity odd	1	1	1	0	0	0	0	0	6/12
RESTART										
RST	Restart	1	1	A	A	A	1	1	1	12
INPUT/OUTPUT										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
INCREMENT AND DECREMENT										
INR r	Increment register	0	0	D	D	D	1	0	0	4
DCR r	Decrement register	0	0	D	D	D	1	0	1	4
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	6
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	6
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	6
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	6
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	6
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	6
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	6
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	6
ADD										
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10
SUBTRACT										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
LOGICAL										
ANA r	And register with A	1	0	1	0	0	S	S	S	4

8085 INSTRUCTION SET SUMMARY (Cont.)

Mnemonic	Description	Instruction Code(1) D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock(2) Cycles
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
ROTATE										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
SPECIALS										
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
CONTROL										
EI	Enable interrupts	1	1	1	1	1	0	1	1	4
DI	Disable interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4
HLT	Halt	0	1	1	1	0	1	1	0	5
NEW 8085 INSTRUCTIONS										
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

NOTES: 1. DDD or SSS: B=000, C=001, D=010, E=011, H=100, L=101, Memory=110, A=111.
2. Two possible cycle times, (6/12) indicate instruction cycles dependent on condition flags.

SUMMARY OF 8085 INSTRUCTIONS

INSTR	STATES	CYCLES	NOTES
AR r	4	1	
AR I	7	2	
AR M	7	2	
CALL	18	5	2
CCN	9/18	2/5	2,4
CMC	4	1	
CMA	4	1	
DAA	4	1	
DAD	10	3	
DCR r	4	1	3
DCR M	10	3	
DCX	6	1	2
DI	4	1	
EI	4	1	
HLT	5	1	5
IN	10	3	
INR r	4	1	3
INR M	10	3	
INX	6	1	2
JMP	10	3	
JCN	7/10	2/3	4
LDS	13	4	
LDAX	7	2	
LHLD	16	5	
LXU	10	3	
MVI M	10	3	
MVI r	7	2	
MOV M,r	7	2	
MOV r,M	7	2	
MOV r,r	4	1	3
NOP	4	1	

INSTR	STATES	CYCLES	NOTES
OUT	10	3	
PCHL	6	1	2
POP	10	3	
PUSH	12	3	2
ROT	4	1	
RET	10	3	
RCN	6/12	1/3	2
RIM*	4	1	
RST	12	3	2
SHLD	16	5	
SIM*	4	1	
STA	13	4	
STAX	7	2	
STC	4	1	
XCHG	4	1	
XTHL	16	5	
SPHL	6	1	2

*New Instruction

NOTES:
1. Two possible state-cycle times indicates dependence on condition flag.
2. Increase in states over 8080 due to necessity of a T_6 during M_1.
3. Decreased from 5 to 4 states during M_1.
4. Instruction branches over last address fetch if condition false.
5. 5 cycles to get a HLT state, 1 cycle necessary to get out a HLT.

APPENDIX 2

CHANGING OPTION JUMPERS

Some of the options provided on the MPU-B will be used only in special situations, such as when a system is being customized for a special configuration or special application. These options have been provided as jumper-pads on the printed-circuit board, and they have a small trace on the board connecting these pads in the standard factory configuration.

When it is desired to change any of these options, the original connecting trace must be broken. This is easily accomplished using a SHARP modelling knife to cut the trace in TWO locations 1/16 to 1/8 inch apart, and pealing the short section out using the point of the knife. The cuts are made more easily if they are slanted so that the bases (at the board surface) are closer than the tops (at the surface of the trace). About 45 degrees is optimum. Be sure to use a SHARP blade.

Once the original factory option traces have been opened, the desired connection can be made by soldering a short wire jumper between the pads provided for the alternate option. Should it later be desired to return the option to its original state, unsolder the jumper wire and re-solder it in the pads between which the previously cut trace runs. As in all soldering work on printed circuit boards, care should be taken to use only the heat necessary to avoid damaging the pads on the board. When removing a jumper, cut it first so that you can remove one end at a time without interference from a second still-connected end. Properly unsoldering anything from a printed circuit board can be difficult. If you do not have the proper tools or skill, protect your board and warrantee by seeking professional aid.

APPENDIX 3

PROM INSTALLATION

The following diagrams show the jumper configuration for two types of PROMs, the 2708 and the 2716. MAKE SURE THE BOARD IS PROPERLY JUMPERED BEFORE POWER IS APPLIED TO AVOID DAMAGE!!!! For ROMs or PROMs other than these, jumper the board for the chip to which the desired PROM is equivalent. If you are not sure of the equivalency, get the assistance of qualified technical help or call IMSAI Customer Service.

FIGURE II-3

JUMPERS FOR 2708

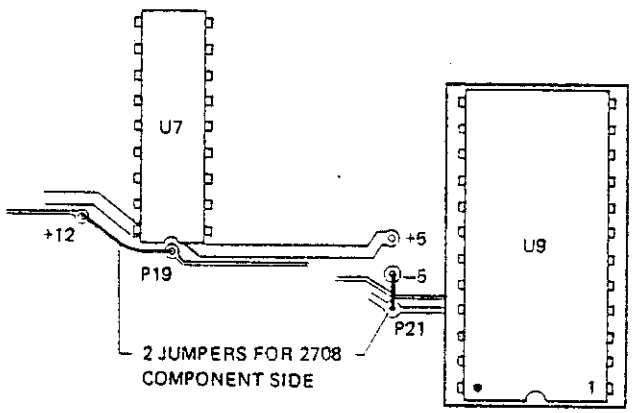
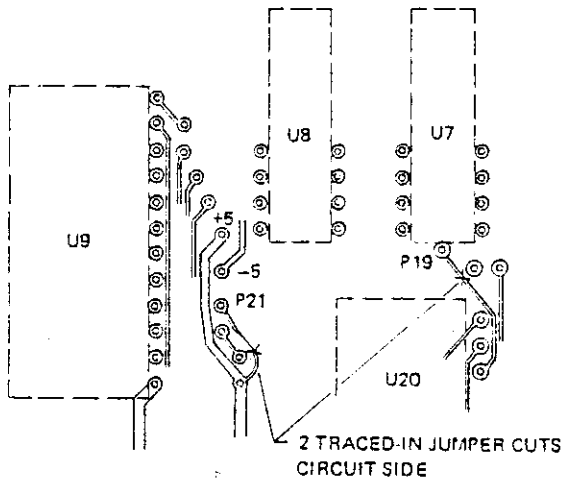


FIGURE II-4

JUMPERS FOR 2708



The bare board is supplied with traced-in jumpers to connect the U9 socket to accept 2716 PROMS.

APPENDIX 4

CUSTOM ADDRESS DECODING

For large quantity OEM uses of the MPU-B, it is possible that an alternate on-board I/O and memory addressing scheme would be desired. IMSAI has the capability of locating the on-board memory-mapped circuits anywhere within 65K in 256 byte blocks. The on-board I/O can be assigned to any ports. Should any change in the addressing scheme described in this Reference Guide be desired, contact IMSAI Customer Service