XPU-80

A Z-80 Processor
With Memory Management
for the S-100 Bus

Edition 1

TABLE OF CONTENTS

COPYRIGHT NOTICE

1.0 INTRODUCTION

The Ithaca Intersystems XPU-80 board is based on the ZILOG Z-80B (TM) processor and features built-in memory management.

The XPU-80 is a processor for 8-bit S-100 bus systems, and is oriented toward medium to large system support. The integration of the memory management enhances support of multi-user multi-tasking environments. The XPU-80 provides basic system features such as full S-100 bus support, optional wait states, optional MWRT generation, and provision for high speed (6 MHz) operation.

This manual provides the information required to prepare, install and operate the board in a system. The manual includes an introductory section, a functional overview of the board's operation, setup instructions for the board, a programmer reference section, and the board parts list.

The functional overview section of this manual contains a complete discussion of each functional section of the XPU-80 board.

1.1 PHYSICAL DESCRIPTION

The XPU-80 board is a standard 5 by 10 inch (12.7 by 25.4 cm), S-100 bus, plug-in circuit board.

1.2 FEATURES

The XPU-80 board contains the following features:

* 6 MHz operation (Optional 4 MHz operation with no wait states)
* Parity error checking
* User/Supervisor Mode
* Hardware/Firmware can detect and correct errors
* A signal LED that can signal messages under software control
* Processor slow-down software selectable wait states
* A page type MMU (Memory Management Unit)
* Board I/O space can be doubled with hardware from 256 to 512 ports
* Latches that store error information
* Compatibility with Front Panel

1.3 SYSTEM INTEGRITY

The XPU-80 board contains a variety of features that enhance system integrity.

1.3.1 Memory Management

The memory management unit (MMU) provides two functions essential to advanced system control: memory address translation and attribute checking.

The XPU-80 hardware can provide absolute protection between the users and between the supervisor and user spaces.

In practice, the operating system determines the degree of protection provided.

In a typical application, the supervisor mode can be used such that only trusted operating code is run in the supervisor space (with potentially unreliable user code excluded from the space), users can be prevented from executing I/O, and controlled re-entry to the supervisor mode from user space can be ensured. The Supervisor/User Circuit (S/U) provides for controlled transfer between supervisor and user spaces.

1.3.2 Real Time Clock

The XPU-80 Real Time Clock (RTC) provides timing for several functions. A 1 Hz clock is output to the time out circuit, and optionally to the interrupt controller Int 5 input. The time-out circuit is used if the system is hung up by a user that has disabled interrupts.

A 50 Hz clock is output to the interrupt controller Int 3 input to provide a real time reference. The 50 Hz tick marks provide a timing reference for time keeping, task swapping, switching between users, and alternating between foreground and background tasks.

1.3.3 Hardware Kernel

The Z80 processor (in Supervisor Mode) and EPROM constitute a hardware kernel that initializes the system and can perform self-diagnostics and error recovery. The hardware kernel is likely to be functional even if the system is down, providing a degraded level of operation, or in the event of major system failure, an indication of the source of the failure.

The kernel can respond to system errors with a combination of the following responses.

* Running system diagnostics stored in EPROM
* Remapping bad memory pages
* Slowing the system down by introducing one or two wait states into every processor cycle
* Signalling an error with the board LED
* Signalling an error with the terminal

1.3.4 Error Detection

The XPU-80 hardware provides several levels of error detection during normal system operation. A generalized error signal is used as an interrupt source.

Parity Circuit

The XPU-80 supports the Intersystems parity scheme. The XPU-80 parity circuit transmits parity on all writes and checks it on reads if PAREN* is asserted by a board on the bus. (Parity is not checked during EPROM, Processor Control, or MMU operations.)

Bus Error

Bus line 98, ERROR*, is monitored. The status signal indicates an error condition during the current bus cycle. When active, the signal becomes an interrupt source for the interrupt controller.

Power Failure

PWRFAIL* (line 13) can be monitored. The signal indicates a power failure on the bus.

MMU Error

MMU Error is generated by an attribute violation.

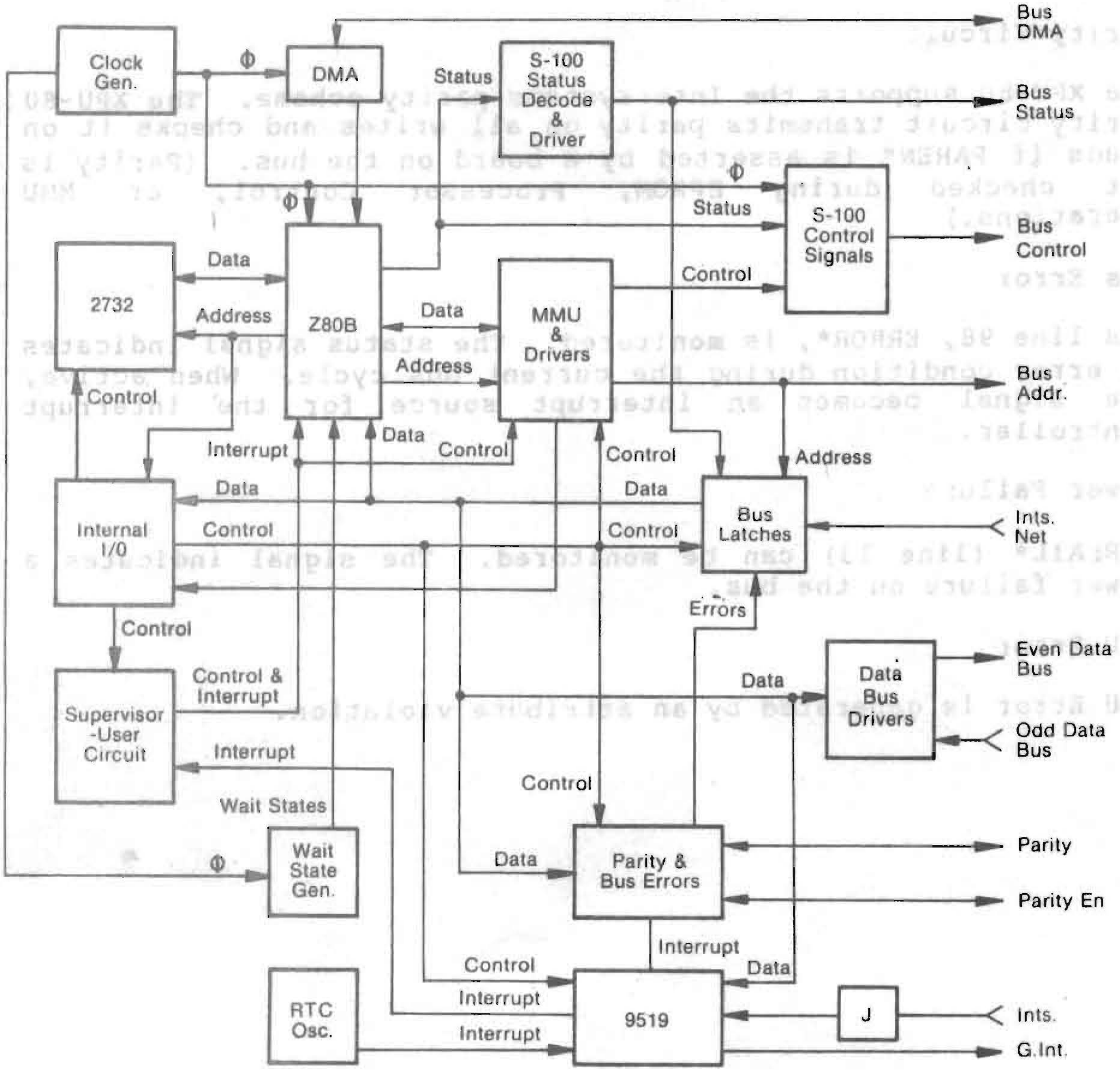


Figure 1 XPU-80 Block Diagram

2.0 FUNCTIONAL OVERVIEW

This section of the manual provides an overview of the operation of the XPU-80 circuit card. The XPU-80 is designed to operate as the permanent bus master in a S-100 bus microcomputer. As permanent bus master, the board is responsible for the initiation of all bus cycles, and for the generation of all signals necessary for the conduction of an unambiguous bus cycle. Bus control can be transfered from the permanent bus master, in this case the XPU-80, to a temporary bus master, a DMA device, as defined by IEEE 696 protocol. An example of a temporary bus master is the FDC II (floppy disk controller) board.

The XPU-80 operates in a system with one or more bus slaves. A bus slave is a circuit board that monitors all bus cycles, and if addressed, accepts or sends the message on the data lines. The bus slave examines and generates only those bus signals necessary to communicate with the bus master. An example of a bus slave is the 256KDR memory board.

The major functional areas of the XPU-80 board are described individually. Each functional description includes an explanation of what the functional area is, what it does, and how it works.

In several circuit descriptions distinction between S-100 bus and processor board signals with a common name is made by adding the designation XPU-80 to the signal name. For example XPU-80 RESET* is the board reset signal.

2.1 BLOCK DIAGRAM

The XPU-80 board is a CPU board. The CPU board consists of a CPU (see 3.2) and the logic elements required to allow tranmission of data between the CPU chip and external devices, to provide the clock generating timing signals required by the CPU, and to manage where data is to be read or where data must be sent.

The block diagram in Figure 1 is a graphic representation of the XPU-80 board. The diagram shows the major functional elements of the board as blocks. The lines connecting the boxes indicate the paths, along which information and/or control flows, between the functional areas. A block diagram is a guide to understanding the interrelationship of the functional areas described in this section of the manual.

2.2 CENTRAL PROCESSOR UNIT

The CPU of the XPU-80 board is a Zilog Z-80B. The Z-80B is a 40-pin MOS/LSI chip, capable of operating at up to 6 MHz. The XPU-80 board is configured at the factory to operate at 6 MHz.

2.3 CLOCK GENERATION

A clock is a repetitive signal used to time or control (synchronize) the events in the computer. The XPU-80 board provides all clock signals required for the CPU and S-100 bus as specified by the IEEE 696/S-100 Bus specification.

Z-80 CLK - 6 (or 4) MHz

S-100 CLOCK - 2 MHz

PHI - 6 (OR 4) MHz

The S-100 bus CLOCK (pin 49) is a 2 MHz clock, not required to be synchronous with any other bus signal.

The S-100 bus PHI (pin 24) is the master timing signal for the bus. The 6 (or 4) MHz signal is normal bus PHI. The XPU-80 board is factory configured for 6 MHz operation and can be factory altered for 4 MHz operation. The 6 MHz board can be identified by the 12 MHz crystal located in the lower left hand corner of the board. The 4 MHz board uses a 8 MHz crystal.

Z80 PHI is bus PHI*. This shifts Z-80 states one-half cycle from bus states.

2.4 REAL TIME CLOCK

A real time clock (RTC) indicates the passage of actual time, as opposed to the fictitious time established by a computer program. The RTC on the XPU-80 provides a 50 Hz output and a 1 Hz output.

The 50 Hz clock is input directly to the interrupt controller Int 3 input to provide a real time reference. Tick marks can be used to provide timing reference for task swapping, switching between users, and alternating between foreground and background tasks. The 1 Hz clock is input to the time-out circuit (Z80 NMI* and EPROM ON) and optionally (jumper option) to the interrupt controller Int 5 input.

The time-out circuit optionally times out a system hung up by a user that has disabled Z80 interrupts. In such a case, the Z80 NMI* (non-maskable interrupt) input is driven within between one-half and one second of the unmasked interrupt (9519 Group Int*). The NMI* pulse also enables the XPU-80 EPROM.

The NMI* timeout does not occur if the system Front Panel board is requesting a wait state, DMA occurs, or the board is in Supervisor mode.

2.5 STATUS LOGIC

At the start of each machine cycle the XPU-80 outputs data on the S-100 status bus to indicate processor status and the type of operation about to be performed. The S-100 status bus consists of eight signal lines.

sM1 (44)	Indicates that the current cycle is an op-code fetch.
sMEMR (47)	Identifies bus cycles that transfer data from memory to a bus master.
sHLTA (48)	Acknowledges that a HALT instruction has been executed.
sINP (46)	Identifies the data transfer bus cycle for an input device.
sOUT (45)	Identifies the data transfer bus cycle to an output device.
sWO* (97)	Identifies a bus cycle that transfers data from the bus master to a slave. S-100 sWO* = Z80B RD*. sWO* is asserted low between cycles. Parity enable has the same timing as sWO*.
sINTA	The XPU-80 does not utilize the sINTA line (the line is driven low). All interrupt acknowledges are handled by the XPU-80 interrupt controller.
sSXTRQ	The XPU-80 does not utilize the sSXTRQ (sixteen-bit data transfer request) line. The XPU-80 is an eight-bit master.

2.6 CONTROL SIGNALS

The control bus is a set of control lines, the function of which is carry the synchronization and control information necessary for the computer system.

2.6.1 Control Output Bus

The lines of the control output bus determine the timing and movement of data during any bus cycle. The S-100 control output bus consists of five signal lines. The lines are:

pSYNC (76)	Indicates the start of a new bus cycle. pSYNC is generated either from the leading edge of MREQ* or the delayed edge of IORQ*. pSYNC is not generated for INTA cycles.
pSTVAL (25)	The Processor Status Valid signal, in conjunction with pSYNC, indicates that stable address and status may be sampled from the bus in the current cycle. Three separate pSTVAL are jumper selectable to accommodate different clock rates and setup times. The board is correctly configured for the installation prior to shipment.
pDBIN (78)	A generalized read strobe that gates data from an addressed slave onto the data bus.
pWR* (77)	A generalized write strobe that writes data from the data bus into an addressed slave. pWR* is gated off during Processor Control accesses, MMU and Supervisor Call stack writes, protected I/O writes, and protected memory writes.
pHLDA (26)	Processor Hold Acknowledge. Used in conjunction with HOLD* (74) to cooridinate bus master transfer operations.

2.6.2 Control Input Bus

A control input bus carries the signals that allow bus slaves to synchronize the operation of the bus master with conditions internal to the bus slave, and to request operations of the bus master. The lines are:

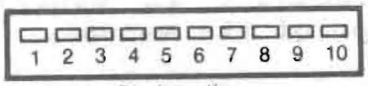
XRDY (3)	One of two ready inputs to the bus master. The ready lines synchronize the bus master to the response speed of the bus slave. Bus cycles are suspended and wait states inserted until both ready lines are asserted. The bus is ready when both inputs are high. The XRDY line is a special ready line used by front panel devices to stop and single step the bus master. RDY is the other ready input.
RDY (72)	The RDY line is the general ready line for bus slaves.
INT* (73)	The INT* line is used by a bus slave to request service from the bus master.
NMI* (12)	The NMI* line is a non-maskable interrupt request line. It is not masked off by the bus master. The signal is asserted as a negative going edge.
SIXTN*	The SIXTN* line is not used in an XPU-80 system.

2.7 FRONT PANEL INTERFACE

The XPU-80 board is compatible for use with the Intersystems Front Panel board and other front panel boards. The XPU-80 is connected to the Intersystems Front Panel via a 10-conductor flat cable. Figure 2 describes the pin assignment for the connector.

In addition to traditional front panel functions, a front panel can be used in systems with the XPU-80 to:

* Read and modify the high speed MMU RAM
* Read and write to the XPU-80 9519 interrupt controller
* Read and write to the Processor Control Registers and Requests
* Read the XPU-80 EPROM



Pin Location

Pin	Description
1	D7
2	D6
3	D5
4	D4
5	D3
6	D2
7	D1
8	D0
9	DDSB* (Data In Disable)
10	Ground

Figure 2 CONN 1

2.8 RESET CIRCUIT

In an S-100 bus computer, system reset functions are controlled by three bus lines: POC*, SLAVE CLR* and RESET*.

POC* (99) is the power-on-clear signal for all bus devices. Bus SLAVE CLR* (54) resets all bus slaves during power-up. RESET* (75) is used to reset all bus masters.

XPU-80 RESET* resets the processor and sets flip flops that enable the EPROM and the Supervisor mode (see sections 2.13 and 2.17 respectively). The Modify Control Register is cleared. After a reset two wait states are requested for all bus cycles, the error circuit is disabled, and the MMU is enabled.

On the XPU-80, an RC network drives POC* and SLAVE CLR* active during power up.

POC* is also used to clear the Service Net counter so that the Service Net line is at a logic ONE state following a power-up. See section 2.18.

The XPU-80 board is in the following state following a reset:

- Supervisor mode
- EPROM enabled in all pages of all spaces
- Software wait request is ON
- Bus and parity error circuits are disabled
- MMU-load mode enabled
- Power-off request is inactive

A typical booting up procedure is described in the EPROM description in this section of the manual.

2.9 WAIT STATE REQUESTS

Wait states are used to synchronize the bus cycles generated by the bus master (the XPU-80) with the response speed of the assorted bus slaves in the system, and certain functions of the XPU-80 board itself. During the bus cycle, a wait state (BSW) is entered if the RDY or XRDY line indicate that the bus slave (or XPU-80 function) is not ready for data transfer.

Ten signals on the XPU-80 can request wait states as described in Figure 3.

XPU-80 Signal	No. of Wait States	Notes
XRDY*	1	Derived from S-100 XRDY
PRDY*	1	Derived from S-100 RDY
9519 Pause*	1	
M1	0, 1, or 2	Jumper Selected (J2)
MEMR	0, 1, or 2	Jumper Selected (J4)
I/O REQ	0, 1, or 2	Jumper Selected (J1)
EPROM ENABLE SET	1	
SUP CALL MUX*	2	During Supervisor Call Sequence
P5 (Modify Control Register)	1	Set by Software, P5=0 at Reset
P4 (Modify Control Register)	2	Set by Software, P4=0 at Reset

Figure 3 XPU-80 wait states

2.10 DMA

The IEEE bus specification defines a special protocol for the transfer of bus control from a permanent bus master (XPU-80) to a temporary bus master (DMA device) for an arbitrary number of bus cycles. The protocol involves a specially timed and overlapped transfer of the various signal groups on the bus such that the DMA device and the CPU are both driving the most critical bus lines in inactive states during the transfer operation.

Two modes of DMA transfer are described by the IEEE standard. The XPU-80 board can be (jumper) configured to operate in either mode. In the first mode, the permanent bus master drives the bus transfer control circuit. That is, the DSB signals (data, address, and control output driver disable signals) are driven by the permanent bus master. In the second mode, the temporary master drives the bus transfer control circuit. The DSB signals are driven by the temporary bus master. When the XPU-80 drives the bus transfer control circuit, the procedure for bus transfer is as follows (see Figure 4):

1) The temporary bus master (DMA device) asserts a hold request. Line 74, HOLD* is active.
2) XPU-80 BUSAK*, active. Line 26, pHLDA, is active.
3) At the rising edge of Phi, XPU-80 ADSB goes active, S-100 ADSB*, DDSB*, and SDSB* are taken low, disabling the address, status and data output drivers of the XPU-80, and enabling the control output drivers of the temporary master.
4) At falling edge of Phi, XPU-80 CDSB goes active, S-100 CDSB* goes active disabling the XPU-80 control output driver and enabling the address, status and data out drivers of the temporary master. Transfer state is terminated.
5) With DMA transfer complete, temporary bus master asserts HOLD* inactive.
7) XPU-80 ADSB clocked inactive, S-100 ADSB*, DDSB*, and SDSB* go inactive, enabling the address, data, and status output drivers of the XPU-80, disabling the control output driver of the temporary master.
8) XPU-80 processor BUSAK* asserted inactive. pHLDA inactive.

The XPU-80 DMA sequencer features a high noise immunity lock out circuit.

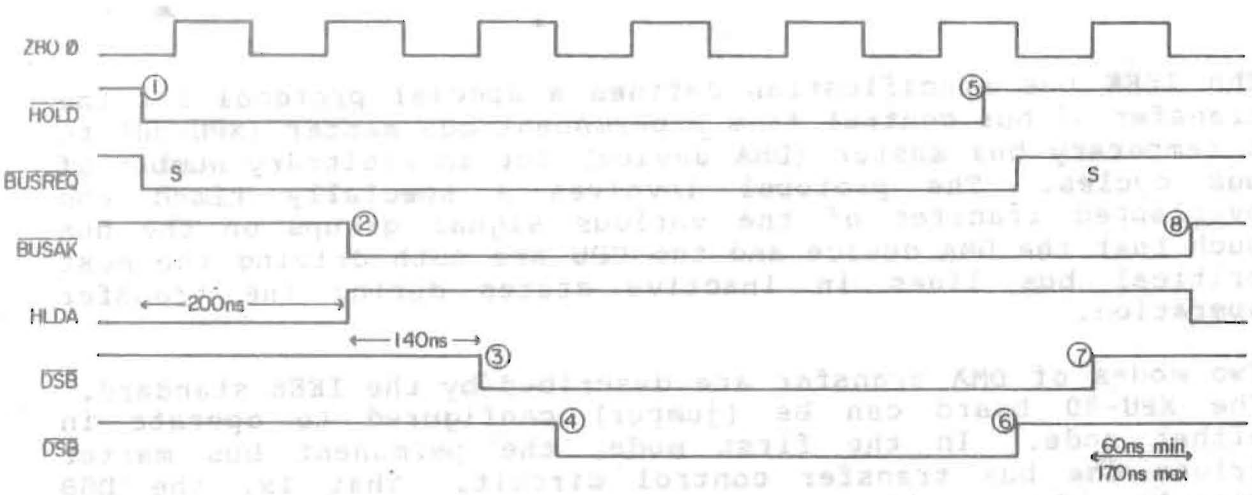


Figure 4 DMA

2.11 PARITY AND BUS ERROR

PARITY ERROR* and BUS ERROR* are input to the Error Latch Circuit (Section 2.12).

2.11.1 Parity

Parity checking is a method of checking the integrity of data when the data is transferred to or from storage. An additional bit, the parity bit, is generated and transferred with the binary information being transferred. The parity bit is the single-digit sum of all the binary digits. The digit value is 0 for an odd number of 1's and 1 for an even number of 1's when the odd parity check is used.

The XPU-80 supports the Ithaca Intersystems parity scheme. In this arrangement, S-100 bus lines 65 and 66 are assigned to PARITY and PAREN* (parity enable) respectively. The PARITY line contains odd parity when the device driving the data bus drives PAREN* active. The reciving device checks the parity and can drive the S-100 ERROR* line active in response to an error.

The XPU-80 parity circuit transmits a parity bit on all writes and checks parity on reads if PAREN* is asserted by the bus slave. If a parity error is detected, XPU-80 signal PARITY ERROR* is active.

The parity circuit is disabled by the Modify Control Register, p3=0. See section 2.15. Parity is not checked during EPROM, Processor Control Register, and MMU operations.

2.11.2 Bus ERROR*

Bus ERROR* (98), is a generalized error line that indicates the current bus operation is producing an error of some sort. Bus ERROR* is input to the Bus Error circuit. When ERROR* is active, the bus error circuit generates BUS ERROR*. The Bus Error circuit can be disabled by the Modify Register Control, P3=0 (ERROR DISABLE*, active).

2.12 ERROR LATCH

It is desirable to save information describing the system status during an error.

In response to MMU ERROR*, PARITY ERROR* and BUS ERROR* (see section 2.11) The error latch circuit performs two functions.

The error latch circuit is the INT 4 source for the interrupt controller.
The error latch circuit drives the latch input of the Processor Control Registers

The stored information is available for retrieval at Processor Control Registers 0 through 3. Refer to section 2.15. Note that stored bus status is lost if an MMU write is performed prior to accessing the stored information.

Figure 5 is a timing diagram with the critical signals. A bus error that occurs very early in the bus cycle causes the storage of bus status from the previous cycle. Data for the current cycle is latched during the period defined by the end of pSYNC to the end of the read or write strobe.

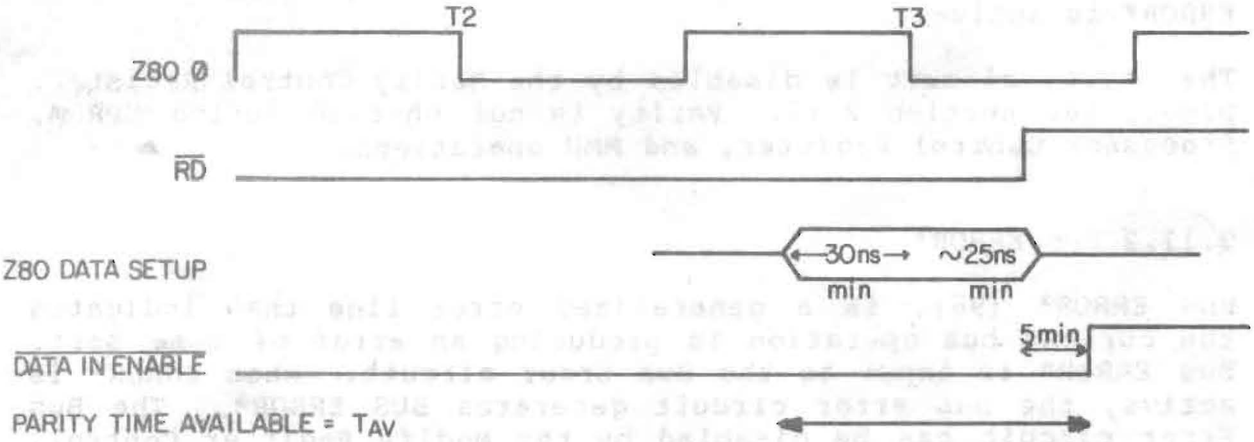


Figure 5 Error

2.13 EPROM

The XPU-80 EPROM can be programmed to provide a variety of functions. The XPU-80 is factory configured to accept a 2732 EPROM. The board can be reconfigured by Intersystems to accept a 2716 EPROM.

The EPROM plays a key role in the process of MMU initialization. The EPROM can contain system diagnostics and code that facilitates the recovery from system errors.

EPROM is enabled in one of two ways.

1) As an attribute function of any page in which d3 of the attribute byte equals ONE.

2) On every page, in every space, when the EPROM enable is set by XPU-80 RESET* (power-up, reset) or NMI*

   Note: Processor Control Request EPROM RESET resets the flip flop.

One wait state is requested when the EPROM flip flop is enabled.

2.13.1 MMU Initialization

The EPROM can be used to initialize the MMU as follows:

1) After reset or power-up, the EPROM is enabled in every page of every space. The Modify Control Register P7 (MMU Enable) is set low, so that the MMU is enabled (as R/W memory) after reset.

2) With the MMU enabled, the EPROM program loads the MMU and enables the Processor Control Registers (d4 high).
3) With the Processor Control Registers enabled, the EPROM code can turn the MMU and itself ON or OFF.
4) Boot the DOS.

2.14 MEMORY MANAGEMENT UNIT

The two major functions of the memory management unit are allocating memory to various tasks, and protecting the memory.

The basic goals of an MMU are:

* To provide a flexable, predictable structure to memory that is independent of the limitations of the CPU logical address structure. Logical address refers to the 16-bit address output by the Z80 processor. Physical address refers to the 24-bit address that appears on the S-100 address bus.

* To protect the system involving:

  a) Protection from inadvertent mistakes such as "runaways".
  b) Protecting data from unauthorized access.
  c) Protecting the operating system from uncontrolled access by users.

* To provide support for multiple independent tasks that can share access to common resources and programs.

The MMU coordinates memory as eight 64 kbyte spaces.

```
Supervisor Space  = Space 111
Stack Write Space = Space 000
User Spaces       = Remaining Spaces
```

The Z80's 64 kbyte space is configured as 16 contiguous 4-kbyte pages. When enabled for loading, the MMU appears as RAM in the highest 16 bytes of every 4 kbyte page. The sixteen bytes consist of eight pairs of attribute byte and relocation byte (described below). The eight pairs correspond to the eight 64 kbyte MMU spaces, selected by the decoding of the MMU RAM multiplexed address bits A3 through A1.

An MMU (logical) address is described in Figure 6.

Address Bit																	
15	14	13	12	11	12	11	10	9	8	7	6	5	4	3	2	1	0
Pg3	Pg2	Pg1	Pg0	1	1	1	1	1	1	1	1	1	1	S2	S1	S0	A*/R

Notes; Pg3 through Pg0 = Page selection
S2 through S0 = Space selection

000 = Stack Write Space
111 = Supervisor Space
All others = User Spaces

A*/R = Attribute byte/Relocation byte selection

0 selects Attribute byte
1 selects Relocation byte

1 = Logic ONE

Figure 6 MMU addressing

2.14.1 Address Translation

Relocation of programs and data areas is accomplished by an address translation mechanism that translates the CPU's logical addresses to physical addresses to be put on the S-100 address bus.

A physical address is formed by concatenating the lower twelve bits (A11 through A0) of the logical address with a relocation byte as shown in Figure 7. The result is a physical address to any of 256 possible 4 kbyte memory spaces in physical memory.

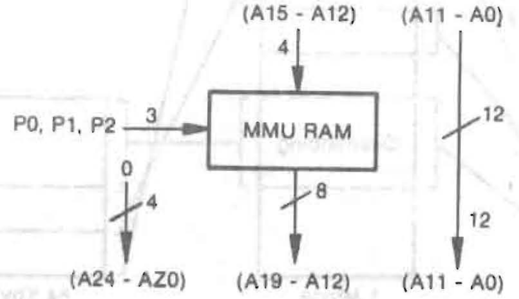


Figure 7 MMU address translation

When a specific page is addressed and the MMU is enabled for loading, a specific relocation byte can be accessed (A0=1). The relocation byte can have any value from 0 to FFH.

The 4 kbyte pages (for each of the spaces) are mapped into physical memory. Pages can be selectively mapped to be overlapping, contiguous, or totally separate. See Figure 8. User/Supervisor spaces share memory wherever relocation bytes are common. Spaces can be prevented from interacting by assigning no common relocation bytes.

When users have common relocation bytes, and therefore share areas of memory, the attribute byte (described below), for each page can be set to protect data against unauthorized use, and to protect supervisor space from access by users.

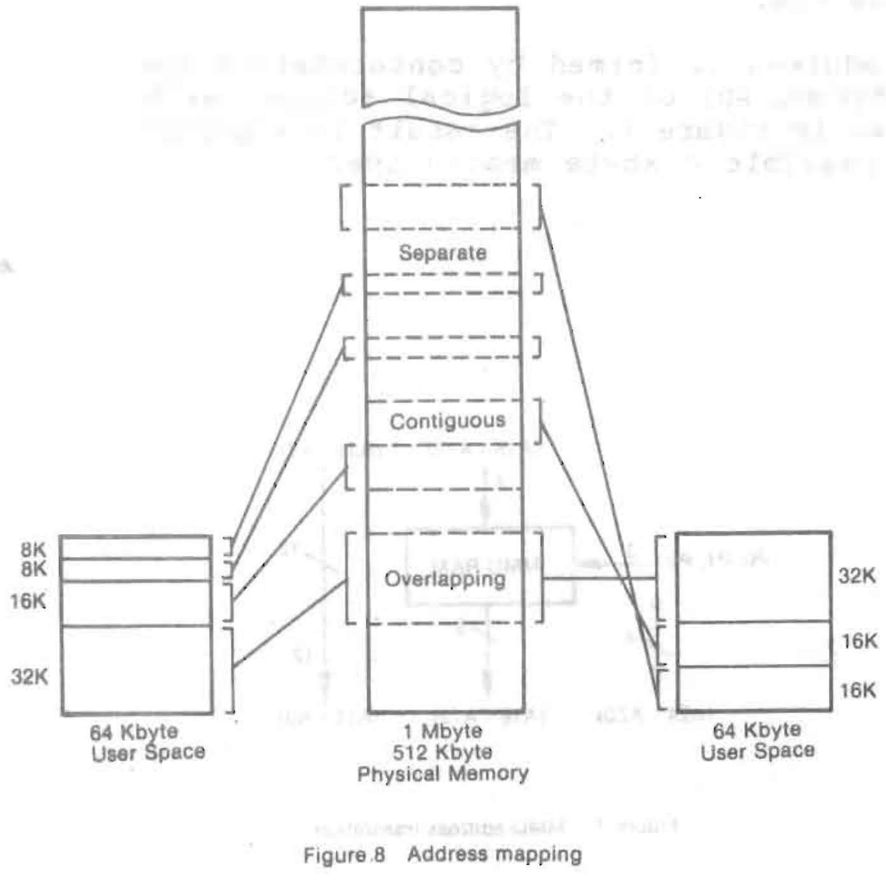


Figure 8 Address mapping

2.14.2 Attribute Byte

There is an attribute byte for each logical page to provide for the restriction of memory access. The attribute byte is checked at each memory access to the page of memory. The attribute byte is described in Figure 9.

Attribute bit d3=1 enables EPROM and d4=1 enables the Processor Control Registers. Refer to section 2.11 for a description of EPROM and section 2.15 for a discussion of the Processor Control Registers.

Attribute bits d0, d2, d5, and d6 are input to a PAL that generates MMU ERROR* if an attribute violation occurs.

MMU ERROR* is output to two circuits. MMU ERROR* is input to the Error Latch circuit (see section 2.12).

2.14.3 MMU Enable

MMU is enabled when the Modify Control Register P7=0 and logical address lines A11 through A4 are high. The space being accessed by the MMU operation is selected by processor address bits A3 through A1. The page being accessed is selected by logical address bits A15 through A12.

Stack write timing - MMU Enable is active throughout the cycle. Uses the same read and write strobes as MMU Read and Write.

Bit	Signal	Description
d0	Read Protect	If ONE, interrupts after a memory read cycle
d1	Not Used	
d2	Dirty Bit	If ONE, interrupts after a memory write
d3	EPROM	If ONE, enables EPROM
d4	System Control	If ONE, enables System Control Registers
d5	I/O Protect	If ONE, disables outputs, and interrupts after I/O cycles
d6	Write Protect	If ONE, disables writes, and interrupts after a memory write
d7	XPU-80 LED	If ONE, turns XPU-80 LED ON

Figure 9 Attribute byte

2.15 PROCESSOR CONTROL REGISTERS AND REQUESTS

The Processor Control Registers are enabled in any user or supervisor page if the d4 attribute bit for that page is set to ONE. When enabled, the lower 16 bytes of the top 32 bytes of the page become Processor Control Registers and Requests. Figure 10 describes the form of a Processor Control Register address.

Address Bits 15	14	13	12	11	12	11	10	9	8	7	6	5	4	3	2	1	0
Pg3	Pg2	Pg1	Pg0	1	1	1	1	1	1	1	1	1	0	X	PC2	PC1	PC0

Notes: Pg3 through Pg0 = Page selection
PC2 through PC0 = Processor Control Register or Request selection
0 = Logic ZERO
1 = Logic ONE
X = Don't Care

Figure 10 Processor Control Register and Request addressing

The Processor Control Registers and Requests structure of the XPU-80 consists of four eight-bit input registers, one eight-bit output Modify Control Register, three decoded Processor Control Requests, and the 9519 interrupt controller as shown in Figure 11.

Address Bits PC2	PC1	PC0	Register	Request
0	0	0	Register 1	Modify Control Register
0	0	1	Register 2	EPROM Reset
0	1	0	Register 3	Service Net Toggle
0	1	1	Register 4	Exit Supervisor
all others			9519	9519

Figure 11 Processor Control Registers and Requests

The Processor Control Registers that save bus status during an error are described in Figure 12.

Bit	Processor Control Register			
	1	2	3	4
0	Service Net	A0	A8	A12
1	sINP	A1	A9	A13
2	sOUT	A2	A10	A14
3	OFF Req*	A3	A11	A15
4	Int Bus Error*	A4	sMEMR	A16
5	Parity Error*	A5	sM1	A17
6	Power Fail*	A6	sWO*	A18
7	Bus NMI*	A7	sHLTA	A19

Figure 12 Processor Control Registers

The Processor Control Requests (Figure 13) are generated by writing any byte to the request address. Following EPROM Reset, reads are from system memory. The Service Net can be jumper (J8) configured as S-100 line 21.

Processor Control Request	Signal	Description
Modify Control Register	P0 } Space No.	Describes selected space.
	P1 } Space No.	000 = Stack Space, 111 Supervisor Space
	P2 } Space No.	all others = user space
	P3 Error Disable	If 0, disables Bus and Parity Error circuit
	P4 Two Wait Req	If 0, requests two wait states
	P5 One Wait Req	If 0, requests one wait state
	P6 Power Off	If 1, drives bus POWER OFF* low
	P7 MMU Enable	If 0, enables MMU
EPROM Reset		After Reset or NMI*, a flip flop is set that turns EPROM ON in all pages of all spaces. EPROM Reset resets the flip flop.
Service Net Toggle		Changes the polarity of the Service Net line.
Supervisor Exit		XPU-80 switches from Supervisor Mode to User Mode at the end of the fourth processor cycle after this request.

Figure 13 Processor Control Requests

2.16 INTERRUPT DEVICE

The XPU-80 board uses a 9519-1 interrupt controller. Features of the interrupt controller include:

* Fixed or rotating priorities
* Common and individual vectoring
* Master mask
* Status register
* Interrupt Service Register (ISR)
* Information Transfers

2.16.1 Interrupt Modes

The XPU-80 processor has three basic modes of interrupt operation that can be changed under software control. The interrupt controller can be programmed to operate in any of the modes. On the XPU-80, the processor interrupt mode and contents of the I register can be changed by the user.

Mode 0

In Mode 0, the interrupt device places an instruction on the processor data bus during the Interrupt Acknowledge (INTA) cycle. The processor executes this instruction instead of the next instruction in memory. In theory, any instruction can be placed on the data bus, but in practice only single byte instructions should be used because the processor only produces an INTA cycle on the first byte of a multiple byte instruction. The single byte call instructions, Restarts, execute a call to one of eight fixed locations in low memory, depending on the coding of the instruction. The interrupt controller can be programmed to supply any of the Restarts in response to any interrupt.

Mode 1

When Mode 1 is selected by the programmer all interrupts in the system respond with a call to location 38H. A common service routine should begin at that location.

Mode 2

Mode 2 is the most powerful interrupt response mode. The processor forms a pointer from the byte received during the INTA and its I register. This pointer would normally be used to read a jump address from memory, but because the system user spaces have the ability to alter the I register, the jump address supplied by memory is ignored; the bus data input buffer is disabled. Instead, two additional INTA pulses are generated, enabling the 9519 to output two more response bytes. The two response bytes are the jump address. The interrupt controller in this situation acts like RAM that overlays system memory.

Memory read or write signals during the period between the first INTA and the first Supervisor M1 cycle, activate the MMU circuit. The Supervisor Mode is forced. Stack writes during the period write into the MMU space 000. Reads during the period generate additional INTA pulses for the interrupt controller.

2.17 SUPERVISOR/USER

Utilization of the XPU-80 takes place in one of two modes: Supervisor mode and User mode.

The Supervisor mode of operation is a non-protected (raw Z-80) mode. Any code sequence is faithfully executed under all conditions, assuming a benevolent Supervisor code sequence. It is possible for a program in Supervisor mode to commit suicide.

The User mode of operation is a protectable mode. The six user spaces that constitute the User mode can be protected against memory write, memory read, I/O write, and I/O read operations.

This section of the manual describes the procedures by which the transition between Supervisor mode and User mode is controlled.

The Supervisor/User circuit controls the transition between supervisor and user modes, and establishes the conditions under which transfers are conducted in the following situations:.

* Leaving the Supervisor (Supervisor to User)
* Returning to Supervisor (User to Supervisor)

3.17.1 Leaving the Supervisor

The XPU-80 is initially in supervisor mode because RESET* enables Supervisor mode.

To leave the supervisor, the following code is executed:

```
STA ;Write to Processor Control Request , Supervisor Exit
EI  ;
Jump;
```

The Processor Control Request is input to a shift register that introduces a delay. The S*/U bit goes high at the end of the fourth processor cycle after the request, allowing time for EI (Enable Interrupts) and the jump. The board, therefore, switches from Supervisor to User mode at the end of the last memory fetch of the jump instruction. The jump is to any address in one of the six user spaces.

2.17.2 Returning to Supervisor Mode

The procedure for returning to Supervisor mode is more complicated than leaving the supervisor. Implementation of a traditional interrupt routine for handling the return to supervisor can generate special problems on the XPU-80.

* The stack write after INTA could overwrite user or supervisor program code.
* On the XPU-80, the user has the ability to alter the processor interrupt mode or the the contents of the I register. A user could therefore return to the supervisor at an incorrect address and crash the system.

To avoid these problems, the XPU-80 has special facilities. To avoid having the stack overwrite either supervisor or user code, a special space (MMU space 000) is reserved for supervisor call stack writes. To overcome problems due to the user changing the processor interrupt mode or the contents of the I register, each processor interrupt mode is handled differently.

Mode 0

The 9519 is programmed to place a restart instruction on the bus during INTA.

Mode 1

The processor jumps to 38H.

Mode 2

A special procedure is implemented. During a Mode 2 interrupt, the processor forms a pointer from the byte it receives during INTA and the contents of the I register. The processor uses the pointer to read a jump address from memory. The XPU-80, given the suspect validity of the contents of the I register, reads the jump address directly from the interrupt controller which performs like a floating RAM that overlays system memory. Two additional INTA pulses are generated by the supervisor call circuit so the interrupt controller outputs two more response bytes.

Memory reads or writes during this period enable the MMU circuit. Bus write strobes are suppressed, stack writes are stored in MMU space 000. Reads during the period generate additional INTA pulses for the interrupt controller.

2.18 SERVICE NET

S-100 line 21 is an XPU-80 jumper selectable (J8) line. When configured as the Service Net line, the line is an open collector serial data transmission line. The line is toggled by writing any byte to Processor Control Request, Service Net Toggle. See section 2.15.

2.19 BUS I/O

Bus I/O is organized so that with hardware modification the following allocation of I/O ports is possible.

	Supervisor	Users
*	Ports 0-255	None (I/O protected)
*	Ports 0-255	Ports 0-255
*	Ports 0-255	Ports 255-511

3.0 BOARD SETUP

The XPU-80 board is configured prior to shipment and should not require modifications.

This section of the manual provides the information required to prepare the XPU-80 board for operation in Intersystems equipment. There are nine jumper areas on the XPU-80 board. Each jumper area is a box with a group of plated-through holes spaced 0.1 inches apart. To configure a jumper area, zero or more connections per box are made by a printed circuit trace on the solder side of the board or by a shunt that slides onto the 0.040 inch square posts that are soldered into the plated-through holes. To change a connection made by a shunt, the shunt is lifted from its position and set across the desired posts at the correct jumper position. To change a connection made by a circuit trace, the trace betweem the plated-through holes is cut, and a shunt is installed across the desired position.

All possible connections within a jumper area are given letter names. Letter names run A B C ... from left to right or top to bottom. Jumper area locations are identified in Figure 15.

J1 J1 selects the desired number of wait states for I/O cycles.

A	0 wait states
B	1 wait state
C	2 wait states

J2 J2 selects the desired number of wait states for M1 cycles.

A	0 wait states
B	1 wait state
C	2 wait states

J3 J3 selects the delay for pSTVAL. a is earliest; c is latest.

A	pSTVAL a
B	pSTVAL b
C	pSTVAL c

J4 J4 selects wait states for MREQ cycles.

A	0 wait states
B	1 wait state
C	2 wait states

J5 J5 configures the Service Net capability.

A-B	Service Net output on S-100 21.
B-C	S-100 21 disables data input receivers.
C-D	Top connector pin 9 drives data input receivers.

J6 J6 selects a source for the 2 MHz clock signal.

A-B	12 MHz crystal
B-C	8 MHz crystal

J7 J7 selects the source for interrupt controller input VI4.

1 Hz clock (solder trace default)
Bus VI5

J8 J8 selects the source for interrupt controller input VI7.

A	VI7 (solder trace default)
B	VI4
C	VI0
C	VI3

J9 J9 configures the board for Front Panel or Front Panelless operation.

A-B	XPU generates MWRITE: Front Panelless mode.
B-C	XPU does not generate MWRITE: Front Panel mode.

3.1 FRONT PANEL OPERATION

Use the following procedures to prepare the XPU-80 to operate with the Front Panel board.

1) Configure the XPU-80 for operation with the front panel by setting jumper J9, B to C.

2) Install the ribbon cable between the Front Panel header and the XPU-80 top connector. The cable is oriented at the XPU-80 connector so that Pin 1 is at the left end of the connector and Pin 10 is at the right end, as viewed from the component side of the board.

Pin Location

Pin	Description
1	D7
2	D6
3	D5
4	D4
5	D3
6	D2
7	D1
8	D0
9	DDSB* (Data In Disable)
10	Ground

Figure 14

5.0 MANUAL APPLICABILITY AND BOARD REVISION

This manual refers to boards identified as Boards are identified in the lower right corner.

6.0 PARTS LIST

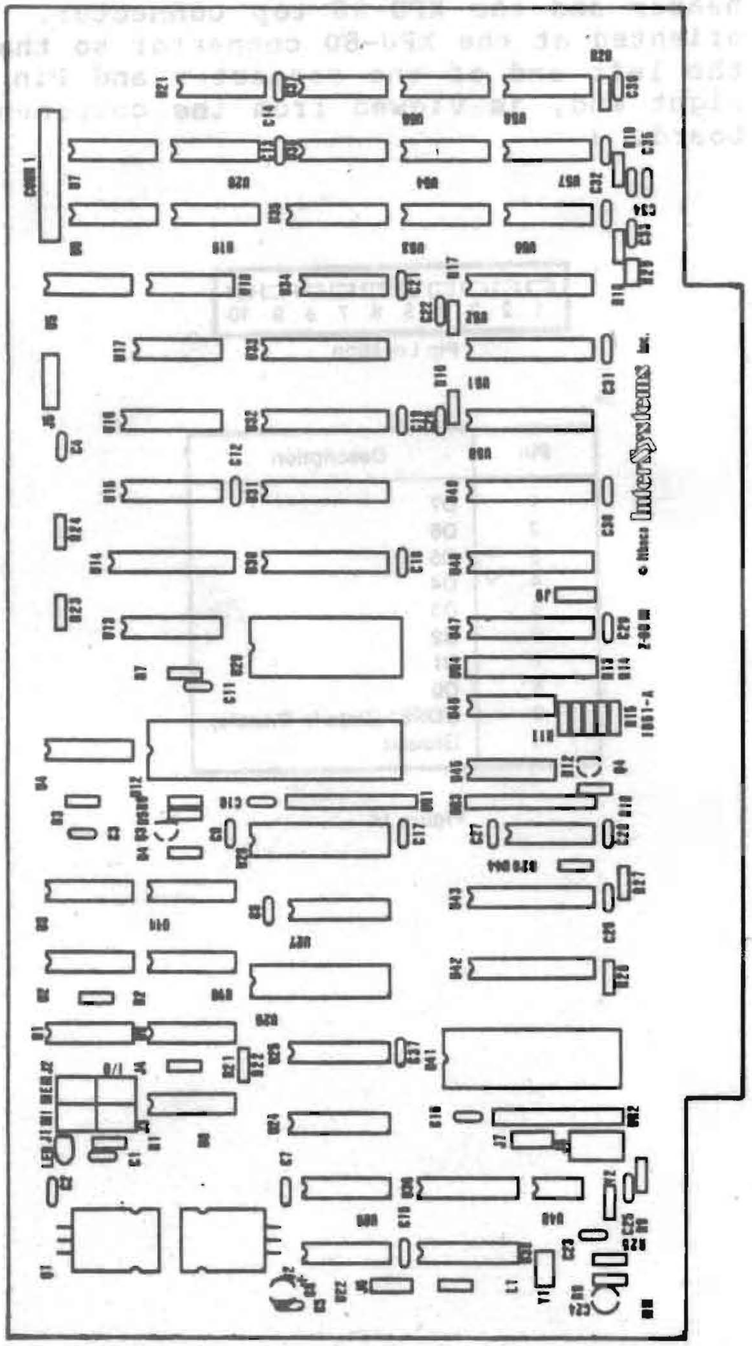


Figure 15

INTEGRATED CIRCUITS

001766	74ALS00		1	U1
001016	74LS00		3	U8,17,23
001015	74S00		1	U58
001018	74LS02		1	U36
001024	74LS04		3	U4,21,45
001023	74S04		2	U11,18
001025	7406		1	U46
001767	74ALS08		1	U54
001028	74LS08		1	U19
001705	74ALS10		1	U5
001030	74LS10		1	U7
001029	74S10		2	U6,20
001033	74LS20		1	U10
001037	74LS30		1	U9
001727	74ALS32		1	U56
001042	74ALS74		3	U2,37,53
001041	74S74		2	U3,57
001768	74LS92		1	U22
001051	74LS125		1	U44
001690	74LS133		2	U13,27
001057	74LS153		2	U24,25
001058	74LS155		1	U16
001059	74LS157		1	U15
001062	74LS164		1	U55
001853	74S175		1	U39
001070	74LS240		1	U47
001072	74LS244		1	U51
001078	74LS273		1	U42
001079	74LS279		1	U35
001080	74LS280		1	U34
001694	74LS321	CLOCK GENERATOR	1	U38
001854	74LS363		4	U43,48,49,50
001083	74LS373		4	U30,31,32,33
001696	74LS533		1	U52
001855	MM5368	NATIONAL	1	U40
001856	AM93422DC	TAA=45ns max	2	U26,28
001857	AM9519-1	AMD	1	U41
001858	Z80B-CPU		1	U12
001859	PAL12L6		1	U14
001860	EPROM 2732-4		1	U29

CAPACITORS

001165	0.1	uf	BYPASS	27	C1-5,7,8,10-19,21, C23,26-32,35
001165	.1	uf	NO # NEXT TO C32	1	
001172	10	uf	10 V DT R	1	C34
001173	10	uf	25 V DT RR	1	C6
001177	5-36	pf	TRIMMER VAR	1	C24
001861	25	pf	50 V CD R +or-10%	1	C25
001862	33	pf	50 V CD R +or-10%	1	C9
001720	56	pf	50 V CD R +or-10%	2	C33,36
001863	100	pf	50 V CD R +or-10%	2	C20,22

RESISTORS & RESISTOR NETWORKS

001285	10 OHM	5% 1/4W CF		4	R11,12,13,14
001286	22 OHM	5% 1/4W CF		1	R6
001287	100 OHM	5% 1/4W CF		5	R16,17,18,19,20
001291	220 OHM	5% 1/4W CF		1	R1
001299	1K	5% 1/4W CF		1	R10
001864	2K	5% 1/4W CF		1	R5
001305	4.7K	5% 1/4W CF		9	R8,15,21,22,25-29
001310	20K	5% 1/4W CF		3	R2,3,7
001865	82K	5% 1/4W CF		1	R4
001866	100K	5% 1/4W CF		1	R9
001867	4.7K OHM	SIP	10P PULLUP	4	UR1-4

001869 COIL
1.) RANGE 5-10uH
2.) NOMINAL VALUE 6uH
3.) RANGE Q-30-40

DIODES, TRANSISTORS & REGULATORS.

001404	5V REGULATOR	2	Q1,2
001868	2N2907	1	Q3
001402	2N3904	1	Q4

L.E.D.

001244		1

CRYSTALS

001879	8 MHz	1	Y1
001202	32.768 KHz	1	Y2