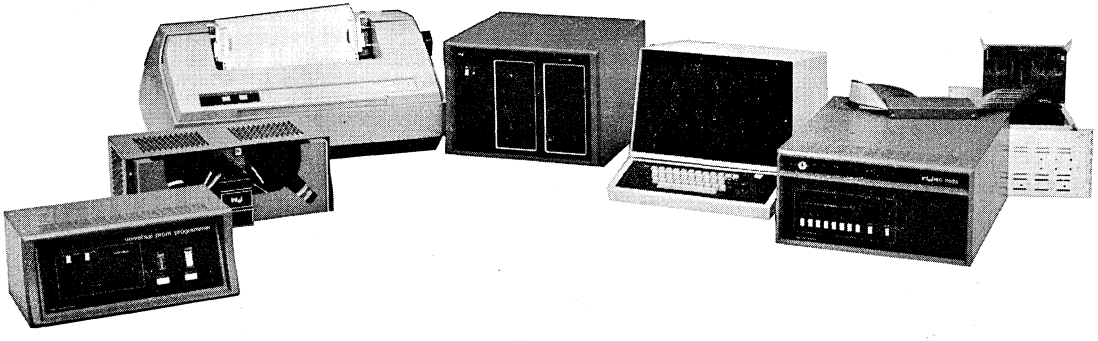
# 10

# DEVELOPMENT SYSTEMS



# MICROCOMPUTER DEVELOPMENT SYSTEMS

This section contains information necessary to select the appropriate Intel design aids required to facilitate microprocessor hardware and software development. Design aids cover the broad range from the Intellec® MDS system with its in-circuit emulator options for both the Intel 8080 (ICE-80) and the Series 3000 Bipolar Microcomputer (ICE-30), to the extensive User's Program Library and a selection of 3 and 4 day Microcomputer Workshops. The purpose of development aids is to shorten the design cycle and thus save time and money in the development and production of microcomputer-based products.

# MDS-800 INTELLEC® MDS MICROCOMPUTER DEVELOPMENT SYSTEM

Modular microcomputer development system for development and implementation of MCS™-80 and Series 3000 Microcomputer Systems

Intel 8080 microprocessor, with 2 $\mu$s cycle time and 78 instructions, controls all Intellec MDS functions.

16K bytes RAM memory expandable to 64K bytes.

2K bytes ROM memory expandable to 14K bytes.

Hardware interfaces and software drivers provided for TTY, CRT, line printer, high speed paper tape reader, high speed paper tape punch, and Universal PROM Programmer.

Universal bus structure with multiprocessor and DMA capabilities.

Eight level nested, maskable, priority interrupt system.

Optional PROM programmer peripheral capable of programming all Intel PROMs.

ICE (In-Circuit Emulator) options extend Intellec MDS diagnostic capabilities into user configured system allowing real time emulation of user processors.

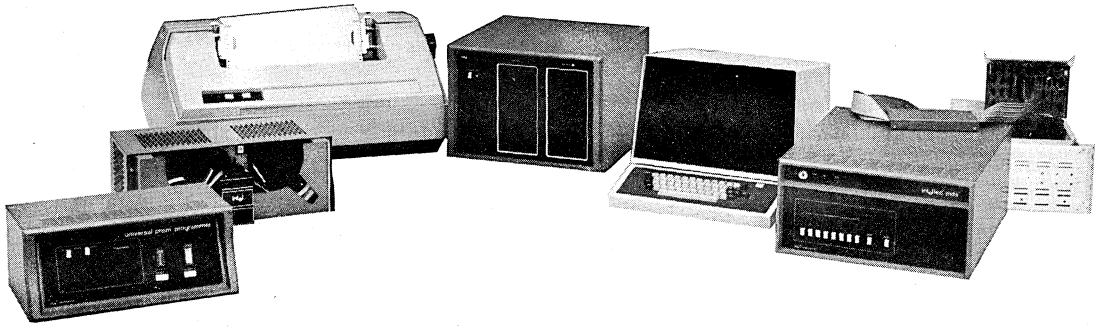
Optional I/O modules expandable in groups of four 8-bit input and output ports to a maximum of 88 ports (all TTL compatible).

ROM resident system monitor includes all necessary functions for program loading, debugging, and execution.

RAM resident macro assembler used to assemble all 8080 machine instructions with full macro and conditional assembly capabilities.

RAM resident text editor with powerful string search, substitution, insertion, and deletion commands.

The Intellec® MDS is a modular microcomputer development system containing all necessary hardware and software to develop and implement Intel MCS™-80 and Series 3000 microcomputer systems. The addition of MDS options and peripherals provides the user with a complete in-circuit microcomputer development system, supporting product design from program development through prototype debug, to production and field test.



## INTELLEC® MDS HARDWARE

The standard Intellec® MDS consists of four microcomputer modules (CPU, 16K RAM Memory, Front Panel Control, and Monitor), an interconnecting printed circuit motherboard, power supplies, fans, a chassis, and a front panel. Modular expansion capability is provided by 14 additional sockets on the motherboard.

The CPU module uses Intel's powerful NMOS 8-bit 8080 microprocessor. The 8080's 2 μs cycle time, 78 instructions, unlimited subroutine nesting, vectored interrupt, and DMA capabilities are fully utilized by the Intellec MDS. Bus control logic resolves bus contention conflicts between the CPU module and other modules capable of acquiring control of the bus. The CPU module interfaces with a sixteen line address bus and a bidirectional eight line data bus. 8080 status signals are decoded and utilized for memory and I/O operations. An eight-level, nested interrupt priority system, complete with an interrupt priority push-down stack, resolves contention for 8080 interrupt servicing.

The RAM memory module contains 16K bytes of Intel 2107 dynamic RAM which operates at full processor speed. All necessary address decoding and refresh logic is contained on the module.

The front panel control module provides system initialization, priority arbitration, and real time clock functions. System initialization routines reside in a 256 byte, PROM resident, bootstrap loader. An eight-level priority arbitration network resolves bus contention requests among potential bus masters. A 1 ms interrupt request generator, which can be disabled under program control, provides real time clock functions. A 10 ms automatic time-out feature is also provided to force an interrupt request if nonexistent memory or I/O is addressed.

The Monitor module contains the Intellec MDS system monitor and all Intellec MDS peripheral interface hardware. The system monitor resides in a 2K byte Intel 8316 ROM. The module contains all necessary control and data transfer circuitry to interface with the following Intellec MDS peripherals:

- Teletype
- CRT
- High Speed Paper Tape Reader
- High Speed Paper Tape Punch
- PROM Programmer
- Line Printer

The Intellec MDS universal bus structure enables several CPU and DMA devices to share the bus by operating at different priority levels. Resolution of bus exchanges is synchronized by a bus clock signal which is derived independently from processor clocks. Read/write transfers may take place at rates up to 5 MHz. The bus structure contains provisions for up to 16-bit address and data transfers and is not limited to any one Intel microcomputer family.

The Intellec MDS front panel is intended to augment the primary user interaction medium, the system console. The simplicity of the front panel coupled with the power of the system monitor provides an efficient user/Intellec interface. The front panel contains eight interrupt request switches with corresponding indicators, CPU RUN and HALT status indicators, a bootstrap loader switch, RESET switch, and a POWER ON switch and indicator.

The basic Intellec MDS capabilities may be significantly enhanced by the addition of the following optional features.

ICE (In-Circuit Emulator) extends Intellec MDS diagnositc capabilities into user configured systems. The Intellec MDS resident ICE processor operates in conjunction with the MDS host CPU and interfaces to the user system via an external cable. The ICE processor replaces the user system processor providing real time emulation capability. MDS resident memory and I/O may be substituted for equivalent user system elements, allowing the hardware designer to sequentially develop his system by integrating MDS and user system hardware. MDS display and debug hardware eliminate the need for specially constructed user system equivalents. Augmenting these capabilities are such powerful ICE debug functions as setting breakpoints, tracing program flow, single stepping, examining and altering CPU registers and memory locations.

The Universal PROM Programmer is an Intellec MDS peripheral capable of programming and verifying the following Intel PROMs: 1702A, 2704, 2708, 3601, 3604, 3624, 8702A, 8704, and 8708. Programming and verification operations are initiated from the Intellec MDS system console and are controlled by programs resident in the Intellec MDS and Universal PROM Programmer.

The addition of a single or dual drive Diskette Operating System significantly reduces program development time. An intelligent controller, constructed around Intel's powerful Series 3000 computing elements, provides diskette interface and control. Intel's software operating system (IDOS) in conjunction with the diskette operating system hardware provides a highly efficient and easy to use method of assembling, editing, and executing programs.

Customized user I/O requirements may be satisfied by adding I/O modules. Each I/O module contains four 8-bit input ports (latched or unlatched), four 8-bit latched output ports (with adjustable strobe pulses), and eight system interrupt lines. All inputs and outputs are TTL compatible. Optional I/O may be expanded to a maximum of 44 input and 44 output ports.

Memory may be expanded by adding RAM or PROM modules in user defined combinations. Up to 64K bytes of RAM may be added in 16K byte increments. PROM (Intel 8702A) may be added in 256 byte increments by adding PROM modules with socket capacity for 6K bytes and populating each module with the desired number of PROMs. Maximum PROM capacity is 12K bytes. RAM/PROM memory overlap is resolved by giving PROM priority.

DMA (direct memory access) modules work in conjunction with the Intellec MDS universal bus to maximize the efficiency of data transfers between MDS memory and selected I/O devices. Each module contains all the necessary control and data transfer logic to implement a complete DMA channel.

A ROM simulator composed of high speed bipolar RAM emulates Series 3000 bipolar microprogram ROM memory. Each ROM simulator module may be used in 512 X 16 or 1024 X 8 configurations.

## INTELLEC® MDS SOFTWARE

Resident software provided with the Intellec® MDS includes the system monitor, 8080 macro assembler and text editor. Used together, these three programs simplify program preparation and speed the debugging task.

The system monitor provides complete control over operation of the Intellec MDS. All necessary functions for program loading and execution are provided. Additional commands provide extensive debug facilities and PROM programming functions. System peripherals may be dynamically assigned either via monitor commands or through calls to the system monitor's I/O subroutines.

Programs may be loaded from the reader device in either BNPF or hexadecimal format. Utility commands which aid in the execution and checkout of programs include:

- initialize memory to a constant
- move a block of memory to another location
- display memory
- modify RAM memory
- examine and modify CPU registers
- set breakpoints
- initiate execution at any given address
- perform hexadecimal arithmetic
- examine and modify the interrupt mask

The Intellec MDS System Monitor contains a powerful and easily expandable input/output system, which is built around four logical device types; console device, reader device, punch device and list device. Associated with each logical device may be any one of four physical devices. The user controls physical device assignment to each logical device through a System command.

Drivers are provided in the system monitor for the Universal PROM Programmer, ASR 33 teletype, high speed paper tape reader, high speed paper tape punch, line printer, and CRT. The user may write his own drivers for other peripheral devices and easily link them to the system monitor.

All system peripherals may be accessed simply by calling I/O subroutines in the system monitor. In addition, the user may dynamically reconfigure his system by monitor commands or by calling system subroutines which can assign a different physical device to each logical device. The user may also determine the current system peripheral configuration, check I/O status and determine the size of available memory.

The monitor is written in 8080 Assembly Language and resides in 2K bytes of ROM memory.

The Intellec MDS Resident Assembler translates symbolic 8080 assembly language instructions into the appropriate machine operation codes. In addition to eliminating the errors of hand translation, the ability to refer to program addresses with symbolic names makes it easy to modify programs by adding or deleting instructions. Full macro capability eliminates the need to rewrite similar sections of code repeatedly and simplifies program documentation.

Conditional assembly permits the assembler to include or delete sections of code which may vary from system to system, such as the code required to handle optional external devices.

The assembler performs its function in three passes. The first pass builds the symbol table. The second pass produces a source listing and provides error diagnostics. The third pass produces the object code. If the punch and list devices are separate (e.g. a high speed punch or printer is available) passes 2 and 3 may be combined into one pass.

Object code produced by the assembler is in hexadecimal format. It may be loaded directly into the Intellec MDS for execution and debugging or may be converted by the system monitor to BNPF format for ROM programming.

The assembler is written in PL/M[T.M.]-80, Intel's high level systems programming language. It occupies 12K bytes of RAM memory including space for over 800 symbols. The symbol table size may be expanded to a maximum of 6500 symbols by adding RAM memory. All I/O in the assembler is done through the system monitor, enabling the assembler to take advantage of the monitor's I/O system. The assembler is shipped in hexadecimal object format on paper tape or diskette and is standard with each Intellec MDS.

The Intellec MDS editor is a comprehensive tool for the entry and correction of assembly language programs for the Intel 8080 microcomputer. Its command set allows manipulation of either entire lines of text or individual characters within a line.

Programs may be entered directly from the console keyboard or from the system reader device. Text is stored internally in the editor's workspace, and may be edited with the following commands:

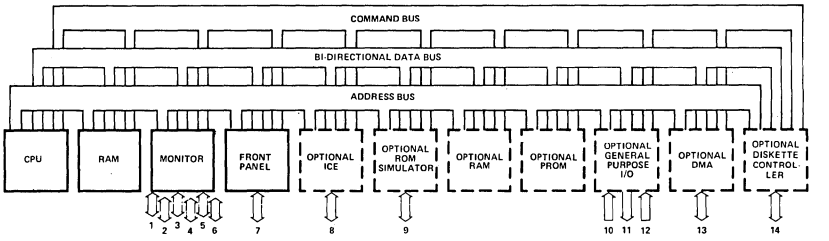
- string insertion or deletion
- string search
- string substitution

To facilitate the use of these editing commands, utility commands are used to change positions in the workspace. These include:

- move pointer by line or by character
- move pointer to start of workspace
- move pointer to end of workspace

The contents of the workspace may be listed to the system console or written to the system list or punch device for future use.

The text editor is written in PL/M[T.M.]-80. It occupies 8K bytes of RAM memory, including over 4500 bytes of workspace. The workspace may be expanded to a maximum of 58K bytes by adding RAM memory. All I/O in the editor is done through the system monitor, enabling the editor to take advantage of the monitor's I/O system. The editor is shipped in hexadecimal object format on paper tape or diskette and is standard with each Intellec MDS.




NOTES:
1. PROM PROGRAMMER DATA/STATUS/COMMANDS
2. HIGH SPEED PUNCH DATA/STATUS/COMMANDS
3. HIGH SPEED READER DATA/STATUS/COMMANDS
4. PRINTER DATA/STATUS/COMMANDS
5. CRT DATA/STATUS/COMMANDS
6. TTY DATA/STATUS/COMMANDS
7. FRONT PANEL STATUS/SWITCH INPUTS
8. USER SYSTEM CPU OR MCU PIN SIGNALS
9. USER SYSTEM ROM PIN SIGNALS
10. EIGHT INTERRUPT LINES
11. FOUR 8-BIT OUTPUT PORTS
12. FOUR 8-BIT INPUT PORTS
13. DMA DEVICE DATA/STATUS/COMMANDS
14. DISKETTE DRIVE DATA/STATUS/COMMANDS

**INTELLEC® MDS BLOCK DIAGRAM**

## HARDWARE SPECIFICATIONS

### WORD SIZE
Host Processor (Intel 8080)
- Data: 8 bits
- Instruction, 8, 16, or 24 bits

### MEMORY SIZE
RAM: 16K bytes expandable to 64K bytes using optional modules.

ROM: 2K bytes expandable to 14K bytes in 256 byte increments using optional PROM modules.

PROM: 256 bytes expandable to 12K bytes using optional modules.

Total: RAM, ROM and PROM may be combined in user defined configurations up to a maximum of 64K bytes.

### MACHINE CYCLE TIME
Host Processor (Intel 8080): 2.0 $\mu$S

### BUS TRANSFER RATE
Maximum bus transfer rate of 5 MHz.

### SYSTEM CLOCKS
Host Processor (Intel 8080) Clock: Crystal controlled at 2 MHz ±0.1%.

Bus Clock: Crystal controlled at 9.8304 MHz ±0.1%.

### I/O INTERFACES
CRT:
- Baud Rates: 110/300/600/1200/2400/4800/9600 (selectable).
- Code Format: 7–12 level code (programmable).
- Parity: Odd/even (programmable).
- Interface: TTL/RS232C (selectable).

TTY:
- Baud Rate: 110
- Code Format:
  - Input: 10 level or greater.
  - Output: 11 level.
  - Parity: Odd.
  - Interface: 20 mA current loop.

High Speed Paper Tape Reader:
- Transfer Rate: 200 cps.
- Control: 2-bit output. 1-bit input.
- Data: 8-bit byte
- Interface: TTL

Punch:
- Transfer Rate: 75 cps
- Control: 2-bit output 1-bit input
- Data: 8-bit byte
- Interface: TTL

Printer:
- Transfer Rate: 165 cps
- Control: 2-bit status input 1-bit output
- Data: ASCII
- Interface: TTL

PROM Programmer:
- Control: 3 strobes for multiplexed output data.
- Data: 8-bit bidirectional
- Interface: TTL

### GENERAL PURPOSE I/O (OPTIONAL)
Input Ports: 8-bit TTL compatible (latched or unlatched); expandable in 4 port increments to 44 input ports.

Output Ports: 8-bit TTL compatible (latched); expandable in 4 port increments to 44.

Interrupts: 8 TTL compatible interrupt lines.

### INTERRUPT
8-level, maskable, nested priority interrupt network initiated from front panel or user selected devices.

### DIRECT MEMORY ACCESS
Standard capability on Intellec bus; implemented for user selected DMA devices through optional DMA module – maximum transfer rate of 2 MHz.

### MEMORY ACCESS TIME
RAM: 450 ns
PROM: 1.3 $\mu$s using Intel 8708A PROM.

### PHYSICAL CHARACTERISTICS
Dimensions: 8.5" X 19" X 17"
21.6 cm X 48.3 cm X 43.2 cm
Weight: 65 lb (29.5 kg)

### ELECTRICAL CHARACTERISTICS

| DC POWER SUPPLY (Volts) | POWER SUPPLY CURRENT (Amps) | BASIC SYSTEM CURRENT REQUIREMENTS (Amps) | |
|---|---|---|---|
| | | Maximum | Typical |
| + 5 ±5% | 35.0 | 9.0 | 6.6 |
| +12 ±5% | 3.0 | 0.7 | 0.4 |
| –10 ±5% | 3.0 | 0.2 | 0.2 |
| –12 ±5% | 0.5 | –– | –– |

### AC POWER REQUIREMENTS
50–60 Hz; 115/230 VAC; 150 Watts

### ENVIRONMENTAL CHARACTERISTICS
Operating Temperature: 0 to 55°C

## SOFTWARE SPECIFICATIONS

### CAPABILITIES

System Monitor:

Devices supported include:

- ASR 33 teletype
- Intel high speed paper tape reader
- Paper tape punch
- CRT
- Printer
- Universal PROM programmer
- 4 logical devices recognized
- 16 physical devices maximum allowed

Macro Assembler:

800 symbols in standard system; automatically expandable with additional RAM memory to 6500 symbols maximum.

Assembles all seventy-eight 8080 machine instructions plus 10 pseudo-operators.

Text Editor:

12K bytes of workspace in standard system; automatically expandable with additional RAM memory to 58K bytes.

### OPERATIONAL ENVIRONMENTAL

System Monitor:

Required hardware:

- Intellec MDS
- 331 bytes RAM memory
- 2K bytes ROM memory
- System console

Macro Assembler:

Required hardware:

- Intellec MDS
- 12K bytes RAM memory
- System console
- Reader device
- Punch device
- List device

Required software:

- System monitor

Text Editor:

Required hardware:

- Intellec MDS
- 8K bytes RAM memory
- System console
- Reader device
- Punch device

Required software:

- System monitor

Tape Format:

Hexadecimal object format.

### MDS OPTIONS

| | |
|---|---|
| MDS-016 | 16K Dynamic RAM |
| MDS-406 | 6K PROM (sockets and logic) |
| MDS-501 | DMA Channel Controller |
| MDS-504 | General Purpose I/O Module |
| MDS-600 | Prototype Module |
| MDS-610 | Extender Module |
| MDS-620 | Rack Mounting Kit |

### MDS EMULATORS/SIMULATOR

| | |
|---|---|
| MDS-ICE-30 | 3001 In-Circuit Emulator |
| MDS-ICE-80 | 8080 In-Circuit Emulator |
| MDS-SIM-100 | Bipolar ROM Simulator |

### MDS PERIPHERALS

| | |
|---|---|
| MDS-UPP | Universal PROM Programmer |
| MDS-PTR | High Speed Paper Tape Reader |
| MDS-DOS | Diskette Operating System |

### MDS INTERFACE CABLES/CONNECTORS

| | |
|---|---|
| MDS-900 | CRT Interface Cable |
| MDS-910 | Line Printer Interface Cable |
| MDS-915 | High Speed Reader Interface Cable |
| MDS-920 | High Speed Punch Interface Cable |
| MDS-930 | Peripheral Extension Cable |
| MDS-940 | DMA Cable |
| MDS-950 | General Purpose I/O Cable |
| MDS-960 | 25-pin Connector Pair |
| MDS-970 | 37-pin Connector Pair |
| MDS-980 | 60-pin Motherboard Auxiliary Connector |
| MDS-985 | 86-pin Motherboard Main Connector |
| MDS-990 | 100-pin Connector Hood |

### EQUIPMENT SUPPLIED

Central Processor Module
RAM Memory Module
Monitor Module (System I/O)
Front Panel Control Module
Chassis with Motherboard
Power Supplies
Finished Cabinet
Front Panel
ROM Resident System Monitor
RAM Resident Macro Assembler
RAM Resident Text Editor
Hardware Reference Manual
Reference Schematics
Operator's Manual
8080 Assembly Language Programming Manual
System Monitor Source Listing
8080 Assembly Language Reference Card
TTY Cable
European AC Adapter
AC Cord

# MDS-DOS DISKETTE OPERATING SYSTEM
## AND MDS-DRV ADDITIONAL DRIVE UNIT

**Floppy diskette operating system providing high speed Input/Output and data storage for the Intellec® MDS.**

**Supports all existing standard Intellec® peripherals.**

**Data on flexible diskette addressed using IBM soft-sectored format which allows 1/4 million byte data capacity per diskette.**

**Up to 200 files per diskette.**

**Dynamic allocation and deallocation of diskette sectors for variable length files.**

**Device independence realized by assignment of unique file names to each peripheral device.**

**Supports optional Intellec MDS ICE-80 (In-circuit Emulator) for Intel® 8080 Microprocessor.**

**Diskette system macro assembler used to assemble all 8080 machine instructions with full macro and conditional assembly capabilities.**

**Diskette system text editor with string search, substitution, insertion, and deletion commands.**

**Listing produced by macro assembler can be directed to diskette allowing interrogation from high speed console device.**

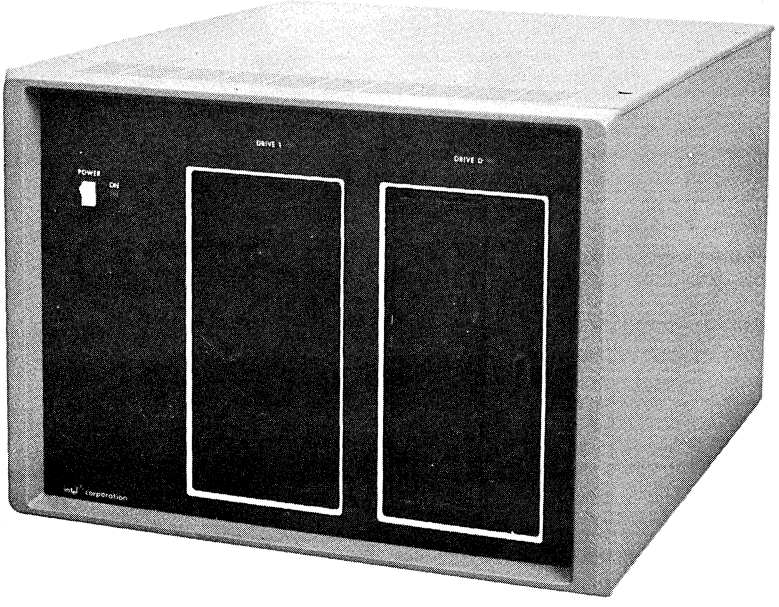
**Diskette operating system software products loaded into Intellec MDS RAM in seconds.**

**Access to all Intellec MDS Monitor facilities.**

**Programs created, edited, assembled, executed and debugged without paper tape handling.**

**Diskette operating system functions callable from user programs.**

The Intellec MDS Diskette Operating System is a general purpose, high speed data handler and file manipulation system for use with the Intellec MDS and its peripherals. The use of a single or dual drive Diskette Operating System significantly reduces program development time. The software system known as ISIS (Intel Systems Implementation Supervisor), provides the ability to edit, assemble, execute and debug programs, and performs all file management tasks for the user.

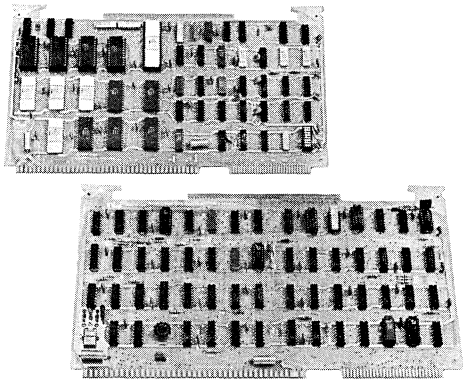



**MDS-DOS Diskette Drive and Optional MDS-DRV**

## HARDWARE

The INTELLEC® MDS diskette system provides direct access bulk storage, intelligent controller, and up to two diskette drives. Each drive provides 1/4 million bytes of storage with a data transfer rate of 250,000 bits/second. The controller is implemented with Intel's powerful Series 3000 Bipolar Microcomputer Set. The controller provides an interface to the INTELLEC MDS bus, as well as supporting the two diskette drives. The MDS diskette system records all data in the IBM-compatible soft sector format.

The MDS diskette controller consists of two boards, the Channel Board and the Interface Board. These two printed circuit boards reside in the INTELLEC MDS chassis and constitute the diskette controller. Each of the systems components is shown in the photograph, and are described in more detail in the following paragraphs.



**DOS Channel and Interface Controller Boards**

## CHANNEL BOARD

The *Channel Board* is the primary control module within the diskette system. The Channel Board receives, decodes, and responds to channel commands from the 8080 Central Processor Unit (CPU) in the INTELLEC MDS system. The Channel Board can access a block of INTELLEC MDS system memory to determine the particular diskette operations to be performed and fetch the parameters required for the successful completion of the specified operation.

The control functions of the Channel Board have been achieved with an 8-bit microprogrammed processor, designed with Intel's Series 3000 Bipolar Microcomputer Set. This 8-bit processor includes four 3002 Central Processing Elements (2-bit slice per CPE), a 3001 Microprogram Control Unit, and 512 X 32 bits of 3604 programmable-read-only-memory (PROM) which stores the microprogram. It is the execution of the microprogram by the microcomputer set which actually effects the control capability of the Channel Board.

## INTERFACE BOARD

The *Interface Board* provides the diskette controller with a means of communication with the diskette drives, as well as with the INTELLEC MDS system bus. Under control of the microprogram being executed on the Channel Board, the Interface Board generates those signals which cause the read/write head on the selected drive to be loaded (i.e., to come in contact with the diskette platter), cause the head to move to the proper track and verify successful operation. The Interface Board accepts the data being read off the diskette, interprets synchronizing bit patterns, checks the validity of the data using a cyclic redundancy check (CRC) polynomial, and then transfers the data to the Channel Board.

During write operations, the Interface Board outputs the data and clock bits to the selected drive at the proper times, and generates the CRC characters which are then appended to the data.

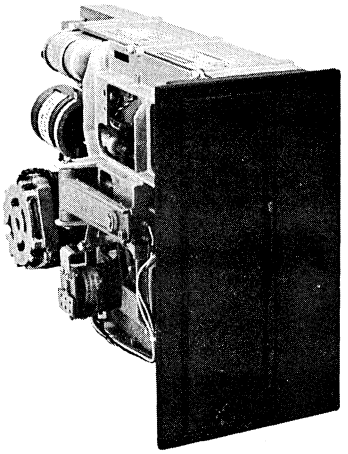
When the diskette controller requires access to INTELLEC MDS system memory, the Interface Board requests and maintains DMA master control of the system bus, and generates the appropriate memory command. The Interface Board also acknowledges I/O commands as required by the INTELLEC MDS bus.

The Diskette System is capable of performing seven different operations: recalibrate, seek, format track, write data, write deleted data, read data, and verify CRC.

## DISKETTE DRIVE MODULES

Each diskette drive consists of read/write and control electronics, drive mechanisms, read/write head, track positioning mechanism, and the removable floppy diskette platter. These components interact to perform the following functions:

- Interpret and generate control signals.
- Move read/write head to selected track.
- Read and write data.



**Additional Drive Unit MDS-DRV**

## SOFTWARE – INTEL SYSTEM IMPLEMENTATION SUPERVISOR (ISIS)

The ISIS programs and subroutines reside on the system diskette and provide a broad range of user-oriented design aid software. Total file management and input editing features greatly reduce software development time. The ISIS 8080 Macro Assembler can be loaded in seconds, from the diskette, and all passes executed without the need for user interaction. Object code and list files may be directed to any output device, or stored as diskette files. A special ISIS utility is provided which converts files from hexadecimal to absolute binary for high-speed retrieval and execution. Powerful system console commands are provided in an easy to use English context. Debugging is initiated by a special prefix to any system command or program call which causes Monitor mode to be entered directly from the program call along with its calling parameters.

A file is a user-defined collection of information of variable length. ISIS also treats each of the standard INTELLEC® MDS supported peripherals as files through preassignment of unique file names to each device. In this manner data can be copied from one device to another (i.e., tape reader to tape punch) using the same command required to copy one diskette data file to another. ISIS provides automatic implementation of random access disk files. Each file is identified by a user-chosen name unique on its diskette. Up to 200 files may be stored on each 1/4 million byte diskette.

## SYSTEM COMMANDS

The user is provided with a wide range of system commands that offer powerful file and program manipulation features:

- The DIR command lists the names, sizes and attributes of files resident on the specified disk directory.
- The RENAME command allows users to change the identifying names of files.
- The COPY command allows users to create new copies of existing files or to transfer files from one device to another.
- The ATTRIB command allows the user to set or reset write-protection and other characteristics of a disk file.
- The DELETE command removes a file from a diskette, thereby freeing space for allocation for other files.
- The HEXBIN command coverts an Intel standard hexadecimal format file into absolute binary format for a reduction in load time and space.
- The FORMAT command formats a diskette on a second disk drive so that it may be used by ISIS.
- The DEBUG command loads the name program and parameters, and gives control to the INTELLEC MDS monitor for execution and/or debugging in the event of an error.
- Programs may be loaded and executed by typing the program name as a command. Users may therefore name their own programs with descriptive verbs and extend their command repertoire.

## ISIS 8080 MACRO ASSEMBLER

The ISIS 8080 Macro Assembler translates symbolic 8080 assembly language instructions into the appropriate machine operation codes. In addition to eliminating the errors of hand translation, the ability to refer to program addresses with symbolic names makes it easy to modify programs by adding or deleting instructions. Full macro capability eliminates the need to rewrite similar sections of code repeatedly and simplifies program documentation. Conditional assembly permits the assembler to include or delete sections of code which may vary from system to system, such as the code required to handle optional external devices.

The ISIS Assembler accepts diskette file input and produces an object file with corresponding symbol table and assembly listing file with any errors. The list file may then be interrogated from the system console or copied to the appropriate list device. The object file may be kept on diskette in its hexadecimal format for loading under ISIS supported software packages such as the optional 8080 In-Circuit Emulator (ICE-80). For loading directly under control of ISIS, the object file may be converted from hexadecimal to absolute binary format using the HEXBIN command.

The ISIS 8080 Macro Assembler is written in PL/M™-80, Intel's high level systems programming language. It occupies 12K bytes of RAM memory allowing space for over 1000 symbols when used with ISIS in a 32K INTELLEC MDS system. The symbol table size may be expanded by adding additional RAM memory.

## ISIS TEXT EDITOR

The ISIS Text Editor is a comprehensive tool for the entry and correction of assembly language programs for the Intel 8080 microcomputer. Its command set allows manipulation of either entire lines of text or individual characters within a line.

Programs may be entered from the console keyboard or may be loaded directly. Text is stored internally in the editor's workspace, and may be edited with the following commands:

- string insertion or deletion
- string search
- string substitution

To facilitate the use of these editing commands, utility commands are used to change positions in the workspace, These include:

- move pointer by line or by character
- move pointer to start of workspace
- move pointer to end of workspace

The contents of the workspace are stored on the diskette and can be immediately accessed by ISIS commands or other programs such as the ISIS 8080 Macro Assembler.

The ISIS Text Editor is written in PL/M™-80. It occupies 8K bytes of RAM memory allowing approximately 12K bytes of workspace in a 32K INTELLEC MDS system.

## SOFTWARE SPECIFICATIONS

### ISIS CAPABILITIES

ISIS commands (User entries at console input device)

File commands:

| | |
|---|---|
| DIR | List diskette directory. |
| COPY | Make a copy of a file. |
| DELETE | Remove a file from diskette. |
| RENAME | Change the name of a diskette file. |
| ATTRIB | Change the attributes of a diskette file. |

Diskette initialization:

| | |
|---|---|
| FORMAT | Initialize a new diskette. |

Program debug and conversion:

| | |
|---|---|
| DEBUG | Execute a program in debug mode. |
| HEXBIN | Convert program from hexadecimal format to absolute binary. |

Program execution:

| | |
|---|---|
| file name | An executable program in a diskette file can be executed by entering the file name as a command. |

ISIS System Calls (System services called by user programs)

Input/output operations:

| | |
|---|---|
| OPEN | Initialize file for input/output operations |
| READ | Transfer data from file to memory |
| WRITE | Transfer data from memory to file |
| SEEK | Position diskette file pointer at any byte in the file |
| RESCAN | Position pointer to beginning of current line |
| CLOSE | Terminate input/output operations on file |

Diskette directory maintenance

| | |
|---|---|
| DELETE | Delete a file from the diskette directory |
| RENAME | Change diskette file name |
| ATTRIB | Change diskette file attributes |

Console Reassignment and error message output

| | |
|---|---|
| CONSOLE | Change console device |
| WHOCON | Determine currently assigned system console |
| ERROR | Output error message on system console |

Program loading and execution

| | |
|---|---|
| LOAD | Load a file of executable code and transfer control to loaded program |
| EXIT | Terminate program and return to ISIS control |

ISIS 8080 Macro Assembler:

1000 symbols in 32K system; automatically expandable with additional RAM memory.

Assembles all seventy-eight 8080 machine instructions plus 10 pseudo-operators.

ISIS Text Editor:

12K bytes of workspace in 32K system; automatcially expandable with additional RAM memory.

### ISIS OPERATIONAL ENVIRONMENTAL

ISIS:

Required hardware:
- Intellec MDS
- 32K bytes RAM memory
- System console
- MDS-DOS Diskette Operating System

Required software:
- System monitor

Macro Assembler:

Required hardware:
- Intellec MDS
- MDS-DOS Diskette Operating System
- 32K bytes RAM memory
- System console

Text Editor:

Required hardware:
- Intellec MDS
- MDS-DOS Diskette Operating System
- 32K bytes RAM memory
- System console

Required software:
- ISIS
- System monitor

Required software:
- ISIS
- System monitor

ICE-80 (Optional)

Required hardware:
- Intellec MDS
- MDS-80 ICE
- 32K bytes RAM memory
- MDS-DOS Diskette Operating System

Required software:
- ISIS
- System monitor

## HARDWARE SPECIFICATIONS

### MEDIA

Flexible Diskette
One Recording Surface
IBM Soft Sector Format
77 Tracks/Diskette
26 Sectors/Track
128 Bytes/Sector

### PHYSICAL CHARACTERISTICS

(Chassis and Drives)

| | |
|---|---|
| Mounting: | Table-Top or Standard 19" Retma Cabinet |
| Height: | 12.08" (30.68 cm) |
| Width: | 16.88" (42.88 cm) |
| Depth: | 19.0" (48.26 cm) |
| Weight: | 1 Drive 51 lb (23 kg) |
| | 2 Drives 64 lb (29 kg) |

### ELECTRICAL CHARACTERISTICS

Chassis
DC Power Supplies

| Voltage | Current |
|---|---|
| 5V | 3A ±5% |
| −5V | 600 mA ±5% |
| 24V | 4A ±5% |

AC Power Requirements
3-wire input with center conductor (earth ground) tied to chassis
Single-phase, 115/230 VAC; 50–60 Hz; 160 watts

INTELLEC® MDS-DOS Controller
DC Power Requirements
Channel Board: 5V @ 3.75A (typ), 5A (max)
Interface Board: 5V @ 1.5A (typ), 2.5A (max)

## DISKETTE DRIVE PERFORMANCE SPECIFICATION

Capacity (Unformatted):
Per Disk . . . . . 3.1 megabits
Per Track . . . . . 41 kilobits
Capacity (Formatted):
Per Disk . . . . . 2.05 M Bits
Per Track . . . . . 26.6 K Bits
Data Transfer Rate . . . . . 250 Kilobits/sec.
Access Time:
Track-to-Track . . . . . .10 ms
Head Settling Time . . . . . .10 ms
Average Random Positioning Time . . . . . .260 ms
Rotational Speed . . . . . .360 rpm
Average Latency . . . . . .83 ms
Recording Mode . . . . . Frequency Modulation

## ENVIRONMENTAL CHARACTERISTICS

### MEDIA

| | |
|---|---|
| Temperature: | |
| Operating | 15.6°C to 51.7°C |
| Non-Operating: | 5°C to 55% |
| Humidity: | |
| Operating: | 8 to 80% (Wet bulb 29.4°C) |
| Non-Operating: | 8 to 90% |

### DRIVES AND CHASSIS

| | |
|---|---|
| Temperature: | |
| Operating: | 10°C to 38°C |
| Non-Operating: | −35°C to 65°C |
| Humidity: | |
| Operating: | 20% to 80% (Wet bulb 26.7°C) |
| Non-Operating: | 5% to 95% |

### MDS-DOS CONTROLLER BOARDS

| | |
|---|---|
| Temperature: | |
| Operating: | 0 to 70°C |
| Non-Operating: | −55°C to 85°C |
| Humidity: | |
| Operating: | Up to 90% relative humidity without condensation. |
| Non-Operating: | All conditions without condensation of water or frost. |

### EQUIPMENT SUPPLIED

Cabinet, Power Supplies, Line Cord, Single Drive
FDC Channel Board
FDC Interface Board
Dual Auxiliary Board Connector
Floppy Disk Controller Cable
Floppy Disk Peripheral Cable
Hardware Reference Manual
Reference Schematics
ISIS System Diskette
ISIS Operators Manual
ISIS/MDS Monitor Bootstrap PROM

### OPTIONAL EQUIPMENT

Rack Mount Kit
MDS-DRV Additional Drive Unit
Blank Diskettes
ISIS System Diskettes

# MDS-UPP UNIVERSAL PROM PROGRAMMER

**Intellec® MDS peripheral capable of programming the following Intel PROMs: 1702A, 2704, 2708, 3601, 3604, 3624, 8702A, 8704, 8708.**

**Personality cards used for specific Intel PROM programming requirements.**

**Zero insertion force sockets for both 16-pin and 24-pin PROMs.**

**Flexible power source for system logic and programming pulse generation.**

**PROM programming verification facility.**

**Stand alone or rack mountable.**

The Universal PROM Programmer is an Intellec MDS peripheral capable of programming and verifying the following Intel PROMs: 1702A, 2704, 2708, 3601, 3604, 3624, 8702A, 8704, and 8708. Programming and verification operations are initiated from the Intellec MDS system console and are controlled by programs resident in the Intellec MDS and Universal PROM Programmer.


## SPECIFICATIONS

### INTERFACE

Data: Two 8-bit unidirectional buses
Commands: 3 Write Commands
2 Read Commands
Initiate Command

### AVERAGE PROGRAMMING TIME

| | |
|---|---|
| 1702A/8702A: | 40 seconds |
| 2708/8708: | 5 minutes |
| 3601: | 2 seconds |
| 3604: | 10 seconds |
| 3624: | 10 seconds |
| 2704/8704: | 2.5 minutes |

### PHYSICAL CHARACTERISTICS

Dimensions: 6" X 7" X 17"
14.7 cm X 17.2 cm X 41.7 cm
Weight: 18 lb (8.2 kg)

### ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0° to 70°C.

### OPTIONS

Personality Cards:
MDS-UPP-361:3601 Personality Card
MDS-UPP-864:8604/3604/3624 Personality Card
MDS-UPP-872:8702A/1702A Personality Card
MDS-UPP-878:8708/8704/2708/2704 Personality Card

PROM Programming Sockets:
MDS-UPP-501: 16-pin/24-pin pair
MDS-UPP-502: 24-pin/24-pin pair

### EQUIPMENT SUPPLIED

Cabinet
Power Supplies
4040 Intelligent Controller Module
Specified Zero Insertion Force Socket Pair
Intellec MDS Interface Cable
Hardware Reference Manual
Reference Schematics

# MDS-PTR HIGH SPEED PAPER TAPE READER

**Intellec® MDS high speed paper tape reader peripheral**

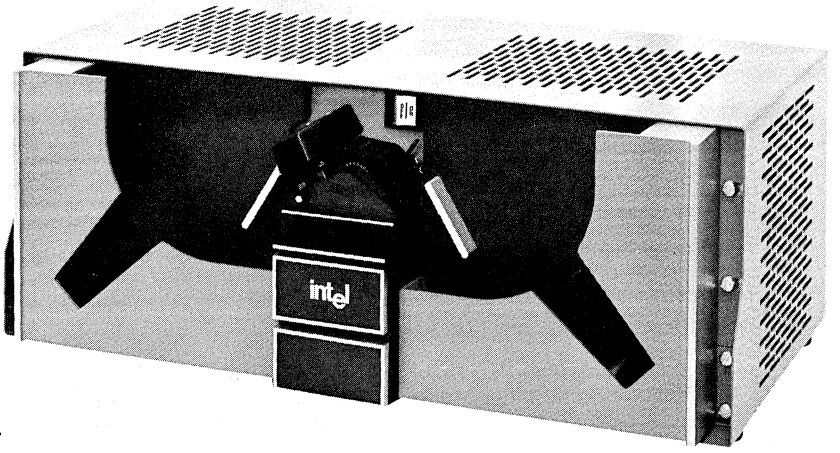
**20 times faster than standard ASR-33 Teletype reader**

**Loads 16K Intellec MDS program memory in less than three minutes.**

**Data transfer at asynchronous rates in excess of 200 characters per second**

**Rack mountable or stand-alone**

The MDS-PTR high speed paper tape reader is an Intellec MDS peripheral that reads paper tape over twenty times faster than the standard ASR-33 Teletype reader. This translates into a significantly faster development cycle due to a marked reduction in the time required for repetitive program loading, assembly, and editing operations.



## SPECIFICATIONS

### TAPE MOVEMENT

Tape Reader Speed:
0 to 200 characters per second asynchronous

Tape Stopping:
Stops "On Character"

### TAPE CHARACTERISTICS

Tape must be prepared to ANSI X 3.18 or EMCA 10 Standards for base materials and perforations.

Reads tape of any material with thickness between 0.0027" and 0.0045" with transmissivity less than or equal to 57% (oiled buff paper tape).

Tape loading: in line

Tape width: 1 inch

### PHYSICAL CHARACTERISTICS

Height: 7.75 in. (19.69 cm)
Width: 19.25 in. (48.90 cm)
Depth: 11.62 in. (29.52 cm)
Weight: 13 lb (5.9 kg)

### ELECTRICAL CHARACTERISTICS

AC Power Requirements:
3-wire input with center conductor (earth ground) tied to chassis. 100, 115, or 127 VAC, single phase at 3.0 amps or 220 or 240 VAC and 1.5 amps; 47 to 63 Hz.

### ENVIRONMENTAL CHARACTERISTICS

Temperature:
Operating: 0 to 55°C (free air)
Non-operating: −55°C to +85°C

Humidity:
Operating: Up to 90% relative humidity without condensation.
Storage: All conditions without condensation of water or frost.

### EQUIPMENT SUPPLIED

Paper Tape Reader
Reader Cable
Fanfold Tape Guide
Fanfold Paper Tape
Hardware Manual
Installation and Operations Guide
Fanfold Guide Installation Instructions

# MDS-ICE-80 8080 IN-CIRCUIT EMULATOR

**Extends powerful Intellec® MDS diagnostic capabilities into user configured system allowing real time (2 MHz) emulation of the user system 8080.**

**User configured system can share Intellec MDS RAM, ROM, and PROM memory.**

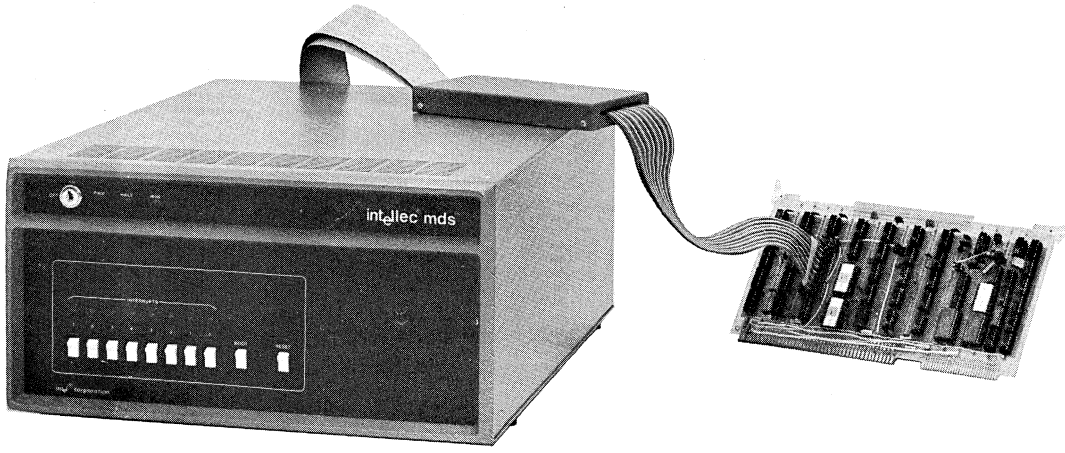
**I/O translation allows user configured systems to share Intellec MDS input/output facilities.**

**Capability to display previously executed instructions with corresponding address, data, and 8080 status information.**

**Capability to examine and alter CPU registers and main memory.**

**Direct Intellec MDS connection to user configured system via an external cable and 40-pin plug.**

ICE-80 is an Intellec MDS resident module that interfaces to any user configured 8080 system and allows the designer to emulate the user 8080 in real time, single step the user system's 8080, substitute Intellec MDS memory and I/O for user system equivalents, and extend powerful debug functions into the user system.




## SPECIFICATIONS

**WORD SIZE**
Instruction: 8, 16 or 24 bits
Data: 8 bits

**CENTRAL PROCESSOR**
8080 CPU, 2 $\mu$S cycle time, 8-bit accumulator, six 8-bit registers, subroutine nesting to any level, multiple level interrupt capability.

**INSTRUCTION SET**
78 instructions including conditional branching, binary arithmetic, logical operations, register-to-register transfers, and I/O.

**CONNECTORS**
Edge Connector: CDC VPB01E43A00A1

**PHYSICAL CHARACTERISTICS**
Width: 12.00 in.
Height: 6.75 in.
Depth: 0.50 in.

**ELECTRICAL CHARACTERISTICS**
DC Power:
$V_{CC}$ = +5 ±5%
$I_{CC}$ = 9.81A max.; 6.90A typ.
$V_{DD}$ = +12 ±5%
$I_{DD}$ = 79 ma max.; 45 ma typ.
$V_{BB}$ = −9V ±5%
$I_{BB}$ = 1 ma max.; 1 $\mu$a typ.

DEVELOPMENT SYSTEMS

# SPECIFICATIONS

**MEMORY ADDRESSING**
Intellec MDS RAM, ROM and PROM may be combined with user system ROM, PROM, and RAM combinations in 4K segments up to a maximum of 65, 536 bytes.

**I/O ADDRESSING**
Intellec MDS I/O ports may be combined with user system I/O ports in 16 port groups, up to a maximum of 256 8-bit input and 256 8-bit output ports.

**USER SYSTEM INTERFACE**
Cable carrying all 8080 address, data, and control signals terminated in a 40-pin plug.

**SYSTEM CLOCK**
Crystal controlled 2 MHz ±0.01%.
Removable by jumper selection when replaced by user clock.

**ENVIRONMENTAL CHARACTERISTICS**
Operating Temperature: 0 to 70°C

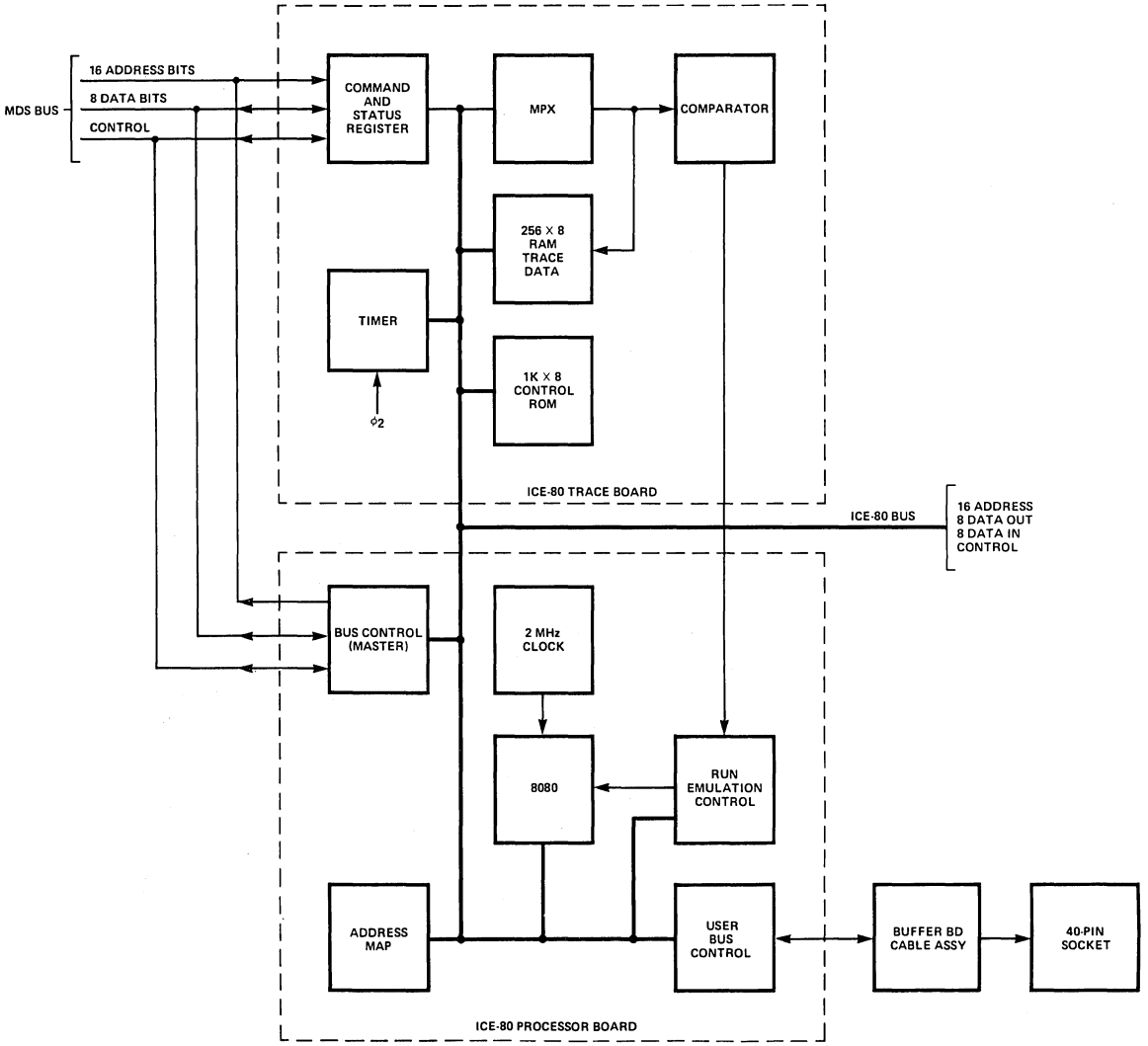
**EQUIPMENT SUPPLIED**
Printed Circuit Modules (2)
Interface Cables and Buffer Board
Reference Manual
Schematic Diagram

ICE-80 allows the user to assign Intellec® MDS resident memory and I/O to the user system. Once assigned, the MDS memory or I/O becomes a part of the user system. The user system may operate with all MDS resident memory and I/O, all user provided memory and I/O, or a combination of both.

ICE-80 debug features include the setting of breakpoints in two hardware comparitors which can trap on any memory read, memory write, I/O read or I/O write operation. Breakpoint extensions, which can be logically ANDED with basic breakpoint parameters, include stack operation, M1 fetch state, or a user defined logic signal. When a breakpoint is encountered in the emulation mode, ICE-80 automatically reverts to the interrogation mode. At this time the memory address, data bus contents, and 8080 status byte from the last 44 machine cycles can be displayed along with the actual number of clock cycles which elapsed since program initiation. In the single-step mode, the user may select single-step or multiple single-step operation. In single-step operation a single instruction is executed, and upon completion, all relevant system status may be displayed. In multiple single-step mode, status information is stored at the end of each machine cycle and the next instruction is executed. When multiple single-step operation is terminated upon a software breakpoint or user command, historical information may be retrieved for display or off line analysis.

The heart of ICE-80 is a microcomputer system utilizing Intel's 8080 microprocessor as its nucleus. This system communicates with the Intellec MDS host processor via I/O commands. Host processor commands and ICE-80 status are interchanged through registers on the ICE module. ICE-80 and the MDS also communicate through a control block resident in Intellec MDS main memory which contains detailed configuration and status information.

The ICE-80 microcomputer system consists of an Intel 8080 CPU, control memory and data storage memory. The system may be driven with either an internal 2 MHz clock or a user supplied clock. The basic ICE-80 system is augmented by several peripheral devices. An 8-bit command register receives Intellec MDS commands and an 8-bit status register provides ICE-80 systems status information to the Intellec MDS. Bus control logic allows the ICE-80 processor to assume control of the Intellec MDS bus as a bus master, when required. A comparitor contains two 24-bit hardware breakpoint registers which provide address and control information associated with breakpoint functions. Finally, buffer/driver circuitry, located in circuit board in the ICE-80 cable, insures that data transmission between the ICE-80 and user system meets the capacitive loading and input current requirements for the 8080.




**ICE-80 BLOCK DIAGRAM**

# MDS-ICE-30 3000 SERIES IN-CIRCUIT EMULATOR

**Extends the Intellec® MDS diagnostic capabilities into user configured systems allowing in-circuit emulation of the user system's 3001 MCU**

**Direct Intellec MDS connection to the user configured system is achieved via an external cable with 3001 compatible 40-pin connector**

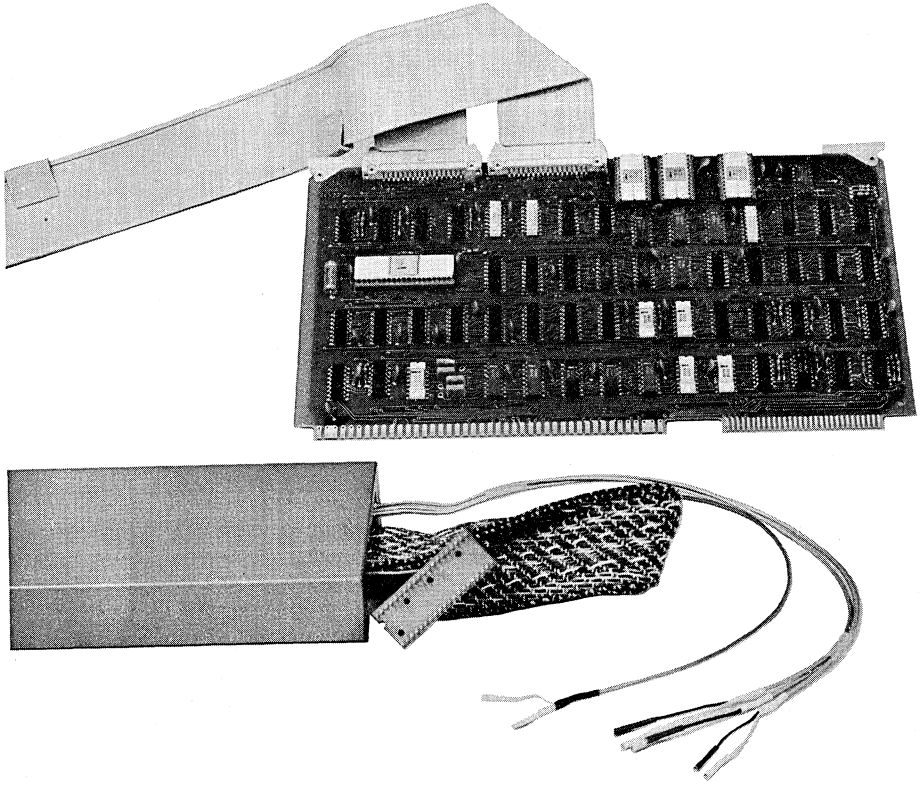
**Provides for the display of all 3001 address, status, and control lines for the current micro-instruction executed**

**Allows for single step microprogram execution**

**Presets the 9-bit 3001 Microprogram Address Register and set two independent breakpoints on micro-instruction addresses generated by 3001**

**Allows two independent breakpoints to be set on the logical combination of any three TTL compatible signals in the user system via three logic probes**

ICE-30 is an Intellec® MDS resident module that provides the user with direct in-circuit emulation of the 3001 Microprogram Control Unit (MCU) and complete control over the execution of user developed microprograms. Through in-circuit emulation, the designer is able to set micro-program address breakpoints, single step micro-program execution and monitor all of the address, status, and control lines of the 3001.



**ICE-30 Module Board with External Cable and 40 Pin Connector**

## A.C. CHARACTERISTICS $T_A = 0°C$ to $55°C$, $V_{CC} = 5.0V \pm 5\%$

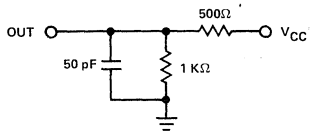
| SYMBOL | PARAMETER | MIN | TYP(1) | MAX | UNIT |
|---|---|---|---|---|---|
| $t_{CY}$ (2) | Cycle Time | 185 | 120 | | ns |
| $t_{WP}$ | Clock Pulse Width | 35 | 20 | | ns |
| $t_{CS}$ | Clock Pulse Separation | 150 | | | |
| | Control and Data Input Set-Up Times: | | | | |
| $t_{SF}$ | LD, $AC_0$–$AC_6$ | 13 | | | ns |
| $t_{SK}$ | $FC_0$, $FC_1$ | 13 | | | ns |
| $t_{SX}$ | $SX_0$–$SX_3$, $PX_4$–$PX_7$ | 13 | | | ns |
| $t_{SI}$ | FI | 13 | | | ns |
| | Control and Data Input Hold Times: | | | | |
| $t_{HF}$ | LD, $AC_0$–$AC_6$ | 15 | | | ns |
| $t_{HK}$ | $FC_0$, $FC_1$ | 15 | | | ns |
| $t_{HX}$ | $SX_0$–$SX_3$, $PX_4$–$PX_7$ | 15 | | | ns |
| $t_{HI}$ | FI | 15 | | | ns |
| $t_{CO}$ | Propagation Delay from Clock Input (CLK) to Outputs ($MA_0$–$MA_8$, FO) | | 90 | 137 | ns |
| $t_{KO}$ | Propagation Delay from Control Inputs $FC_2$ and $FC_3$ to Flag Out (FO) | | 78 | 130 | ns |
| $t_{FO}$ | Propagation Delay from Control Inputs $AC_0$–$AC_6$ to Latch Outputs ($PR_0$–$PR_2$) | | 98 | 150 | ns |
| $t_{EO}$ | Propagation Delay from Enable Inputs EN and ERA to Outputs ($MA_0$–$MA_8$, FO, $PR_0$–$PR_2$) | | | 50 | ns |
| $t_{FI}$ | Propagation Delay from Control Inputs $AC_0$–$AC_6$to Interrupt Strobe Enable Output (ISE) | | 86 | 140 | ns |
| $t_{MH}$ | Propagation Delay from Clock Input (CLK) to Breakpoint Match $\overline{MATCH}$ | | | 158 | ns |

NOTES:
(1) Typical values are for $T_A = 25°C$ and nominal supply voltage.
(2) $t_{CY} = t_{CO} + t_{SF} + t_{WP}$

**TEST CONDITIONS:**

Input rise and fall times of 10 ns between 0.8 volt and 2.4 volts.
Output load of 10 mA and 50 pF.
Speed measurements are taken at the 1.5 volt level.

**TEST LOAD CIRCUIT**




## CAPACITANCE $T_A = 25°C$

| SYMBOL | PARAMETER | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|---|
| $C_{IN}$ | Input Capacitance: | | | 50 | pF |
| $C_{OUT}$ | Output Capacitance | | | 50 | pF |

## D.C. AND OPERATING CHARACTERISTICS $T_A = 0°$ to $55°C$, $V_{CC} = 5.0V \pm 5\%$

### ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias . . . . . 0°C to 55°C

Storage Temperature. . . . . −20°C to +75°C

All Output and Supply Voltages . . . . . −0.5V to +7V

All Input Voltages . . . . . −1.0V to +5.5V

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

## D.C. AND OPERATING CHARACTERISTICS

| SYMBOL | PARAMETER | MIN | TYP(1) | MAX | UNIT | CONDITIONS |
|---|---|---|---|---|---|---|
| $V_C$ | Input Clamp Voltage (All Input Pins) | | −0.8 | −1.5 | V | $I_C = -12mA$ |
| $I_F$ | Input Load Current:<br>CLK Input<br>Logic Probe inputs<br>All other inputs | | | <br>−2.0<br>−3.0<br>−0.4 | <br>mA<br>mA<br>mA | $V_F = 0.45V$ |
| $V_{IL}$ | Input Low Voltage | | | 0.8 | V | $V_{CC} = 5.0V$ |
| $V_{IH}$ | Input High Voltage | 2.0 | | | V | $V_{CC} = 5.0V$ |
| $I_{CC}$ | Power Supply Current | | | 0.0 | mA | |
| $V_{OL}$ | Output Low Voltage<br>$PR_0$–$PR_2$<br>All other outputs | | <br>0.35<br>0.35 | <br>0.45<br>0.45 | <br>V<br>V | <br>$I_{OL} = 16mA$<br>$I_{OL} = 40mA$ |
| $V_{OH}$ | Output High Voltage<br>$MA_0$–$MA_8$, ISE, FO | 2.4 | 3.0 | | V | $I_{OH} = -2mA$ |
| $I_{OS}$ | Output Short Circuit Current<br>$MA_0$–$MA_8$, ISE, FO | −40 | | −120 | mA | $V_{CC} = 5.0V$(2) |
| $I_{O(OFF)}$ | Off-State Output Current<br>$MA_0$–$MA_8$, FO<br>$MA_0$–$MA_8$, FO, $PR_0$–$PR_2$ | | | <br>−100<br>100 | <br>μA<br>μA | <br>$V_0 = 0.45V$<br>$V_0 = 5.25V$ |

NOTES:
(1) Typical values are for $T_A = 25°C$ and nominal supply voltage.
(2) Not more than one output should be shorted at one time.

# SIM-101/SIM-102/SIM-104 ROM SIMULATORS

Extends the powerful Intellec® MDS diagnostic capabilities into user-configured systems, allowing simulation of the user system's bipolar ROM/PROM memory

Direct Intellec MDS connection to the user-configured system via external cables and Intel's ROM/PROM compatible dual-in-line connectors

Simulates Intel's standard bipolar ROMs and PROMs

Modular design allows the user to configure simulation modules to particular memory space requirements

Directly load the ROM Simulator modules from the output of the Intel® Cross Microassembler, CROMIS

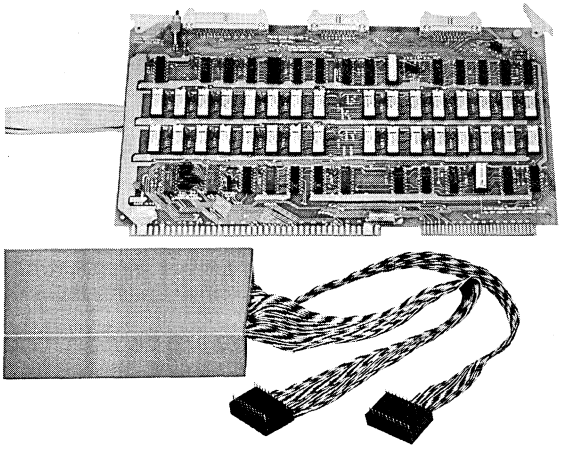
Access the configured memory space from the console keyboard using simulated ROM addresses

Examine an entire word regardless of length; i.e., 8 bits, 10 bits, 32 bits etc.

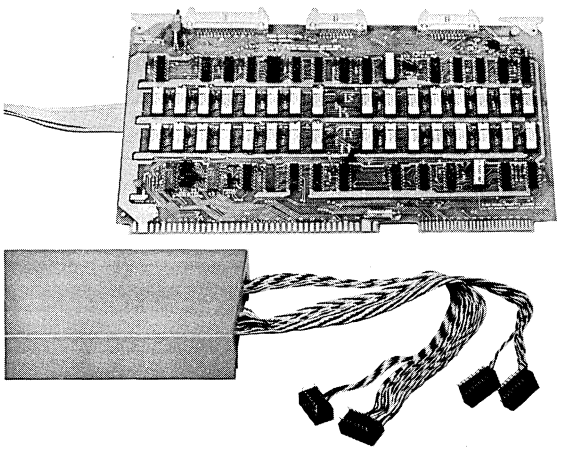
Modify an entire word in a single operation regardless of length

Read access time is 130 ns, maximum

Each ROM-SIM module consists of a high-speed, 130-nanosecond 8K bit RAM board, buffer assembly, external cables, and an interactive software program. The ROM-SIM software is a PL/M™-80 program that operates in the Intellec MDS to provide the user interface for the ROM-SIM hardware. The software loads BNPF or hexadecimal files such as those generated by the Cross Microassembler System, CROMIS. The ROM-SIM software has the capability to compare and verify microcode, load, display and modify simulated control store contents, and output new BNPF or hexadecimal files from the simulated ROM memory for ROM/PROM programming.



SIM-101, 102



SIM-104

# SPECIFICATIONS

## DC CHARACTERISTICS

$T_A$ = 0°C to 55°C, $V_{CC}$ = 5.0V ±5%

| SYMBOL | PARAMETER | LIMITS | | | TEST CONDITION |
|---|---|---|---|---|---|
| | | MIN | MAX | UNIT | |
| $I_I$ | Input Load Current<br>Low Order Addr A0-A8<br>High Order Addr A9-AB<br>Chip Selects | | <br>−1.6<br>−2.1<br>−0.75 | <br>mA | <br>$V_{CC}$ = 5.25V<br>$V_{IN}$ = 0.45V |
| $V_{OL}$ | Output Low Voltage | | 0.45 | V | $V_{CC}$ = 4.75V, $I_{OL}$ = 16 mA |
| $I_{CC}$ | User Power Supply Sensing | | 6 | mA | User $V_{CC}$ = 5.25V |
| $V_{IL}$ | Input Low Voltage | | 0.8 | V | $V_{CC}$ = 5.0V |
| $V_{IH}$ | Input High Voltage | 2.0 | | V | $V_{CC}$ = 5.0V |
| $V_{OH}$ | Output High Voltage | | 2.4 | V | $V_{CC}$ = 4.75V |
| $I_{SC}$ | Output Short Circuit Current at Single Output | −40 | −100 | mA | $V_O$ = 0V, $V_{CC}$ = 5V |
| $I_{CEX}$ | Output Leakage Current | | ±50<br>250 | μA<br>μA | For High Impedance State<br>For Open Collector $V_{CC}$ = 5.25V |

## ABSOLUTE MAXIMUM RATINGS

Temperature Under Bias ..... 0°C to 55°C
Storage Temperature ..... −20°C to 75°C
All Outputs or Supply ..... −0.5V to 7.0V
All Inputs ..... −1.0V to 5.5V

## CAPACITANCE LOAD

| | | |
|---|---|---|
| $C_{IN}$ | Low Order Address, Chip Selects<br>High Order Address (Coaxial) | 45 pF max.<br>50 pF max. |
| $C_{OUT}$ | Data Outputs | 50 pF max. |

# INTELLEC® 8/MOD 8
## MICROCOMPUTER DEVELOPMENT SYSTEM

**Complete Hardware/Software Development System for the design and implementation of 8008 CPU based microcomputer systems**

**Front panel designer's console provides complete system control and monitoring functions**

**8K bytes of random access memory (RAM) expandable to 16K bytes**

**2K bytes of erasable and field programmable read only memory (PROM) expandable to 16K bytes**

**Self contained PROM programming facility with zero insertion force PROM socket**

**Four 8-bit input and four 8-bit output ports**

**Integral asynchronous serial data communications capability at 110, 1200, or 2400 baud**

**Discrete teletype interface (20mA) current loop)**

**Standard system software includes a PROM resident system monitor, RAM resident Macro-Assembler and RAM resident text editor**

**Expansion capability provided for up to 16 standard or custom designed microcomputer modules**

The Intellec® 8/MOD 8 (imm 8-80A) is a complete, self-contained microcomputer development system designed specifically to support the development and implementation of 8008 CPU based microcomputer systems. Its modular design facilitates the development of both large and small MCS-8 system.

The basic Intellec 8/MOD 8 consists of seven standard microcomputer modules (CPU, RAM, PROM, I/O, PROM Programmer, Front Panel Control) and power supplies enclosed in a finished table top cabinet. The heart of the system is the imm 8-82 central processor module built around Intel's 8008 p-channel 8-bit CPU on a single chip.

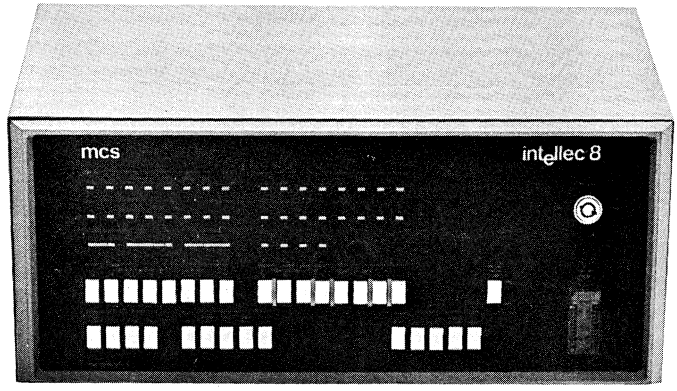
The Intellec Development System directly supports up to 16K of memory, eight input ports, twenty-four output ports, and provides expansion capability for custom designed micrcomputer modules within the system chassis.

The front panel designer's console provides an easy means of monitoring and controlling system operation, manually moving data to and from memory and input/output devices, setting hardware breakpoints, and executing or debugging programs.

The Intellec 8/MOD 8 has 10K bytes of memory in its basic configuration which can be expanded to 16K bytes within the system chassis. Of the basic 10K bytes of memory, 8K bytes are random access read/write memory located on two imm 6-28 RAM memory modules. This memory can be used for both data and program storage. The remaining 2K bytes of memory are located on the imm 6-26 PROM memory module and contain the Intellec 8/MOD 8 system monitor in eight Intel® 1702A erasable and field programmable read only memory chips. Eight additional sockets (2K bytes) are available on the imm 6-26 for expansion.

The PROM and RAM memory modules may be used in any combination to make up the 16K of directly addressable memory housed in the system chassis. Facilities are built into these modules so that combinations of RAM, ROM or PROM may be mixed in 256 byte increments.

The self-contained PROM programming module allows Intel 1602A or 1702A PROMs to be programmed and verified directly from RAM or PROM memory.



# INTELLEC® 8
# HIGH SPEED PAPER TAPE READER

**Directly compatible with all Intellec® 8 Microcomputer Development Systems**

**20 times faster than standard ASR-33 teletype reader**

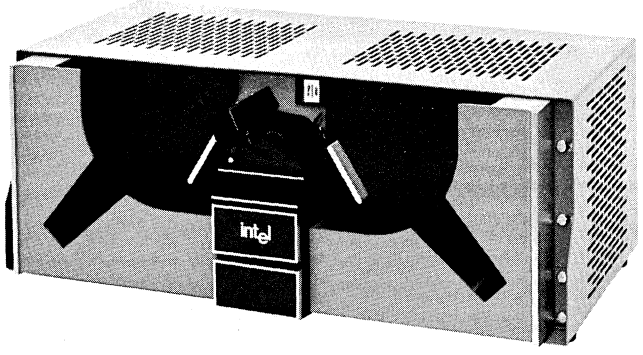
**Loads any 8K Intellec® 8 program memory in less than 90 seconds**

**Data transfer at asynchronous rates in excess of 200 characters per second**

**Rack mountable or stand-alone**

The imm8-90 high speed paper tape reader provides all Intellec 8 Microcomputer Development Systems with a high speed paper tape input that is over twenty times faster than the standard ASR-33 teletype reader. This translates into a significantly faster development cycle due to a marked reduction in the time required for repetitive program loading, assembly, and editing operations.

The Intellec 8 monitor software provides two key capabilities which significantly enhance the systems performance of the imm8-90. A general purpose paper tape reader driver is included in the Intellec 8 Monitor. It enables all systems software to utilize the high speed reader features and is caliable by user written application programs. The monitor also provides dynamic I/O reconfiguration permitting instantaneous reassignment of physical devices to logical devices.



## SPECIFICATIONS

**TAPE MOVEMENT:**

Tape Reading Speed
0 to 200 characters per second asynchronous

Tape Stopping
Stops "On Character"

**TAPE CHARACTERISTICS:**

Tapes must be prepared to ANSI X 3.18 or EMCA 10 Standards for base materials and perforations.

Reads tape of any material with thickness between 0.0027" and 0.0045" with transmissivity less than or equal to 5% (oiled buff paper tape).

Tape loading: in line

Tape width: 1 inch

**ELECTRICAL CHARACTERISTICS:**

AC Power Requirement
3 wire input with center conductor (earth ground) tied to chassis. 100, 115, or 127 VAC, single phase at 3.0 amps or 220 or 240 VAC and 1.5 amps; 47 to 63 Hz.

**EQUIPMENT SUPPLIED**

Paper Tape Reader
Reader Cable
Reader Flat Cable
Fanfold Tape Guide
Fanfold Paper Tape
Hardware Manual
Installation and Operations Guide
Fanfold Guide Installation Instructions

# BAREBONES 80
# MICROCOMPUTER SUBSYSTEM

Complete 8080 CPU based microcomputer subsystem composed of Intel microcomputer modules housed in a card cage and interconnected by a printed circuit motherboard containing module sockets

78 instructions including data transfer; decimal, binary, and double precision arithmetic; logical, branch, stack, and I/O

Vectored interrupt capability

DMA capability

4K 8-bit bytes of RAM expandable to 16K bytes in standard system and 64K bytes in user modified system

Sockets for 4K 8-bit bytes of PROM expandable to 16K bytes in standard system and 64K bytes in user modified system

Four 8-bit input ports expandable to 16 input ports; four 8-bit output ports expandable to 28 output ports. Expansion to 256 input and 256 output ports in user modified system. All ports are TTL compatible

Integral asynchronous serial data communications capability at 110, 1200, or 2400 baud

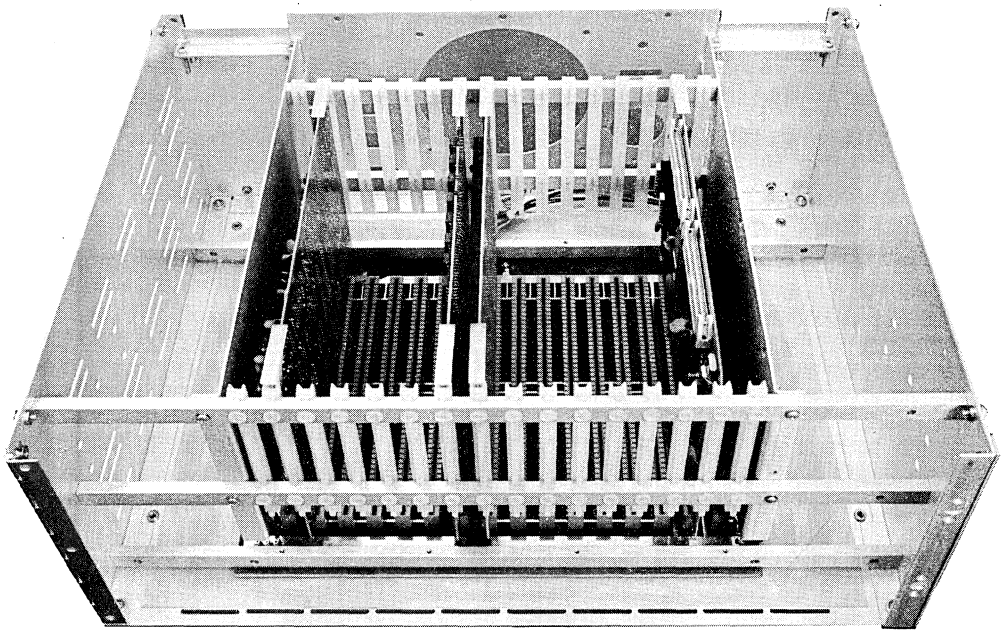
Discrete teletype interface (20 mA current loop)

Expansion capability provided for additional 12 Intel or custom microcomputer modules

Rack mountable.

The Barebones 80 (imm8-85) is a complete microcomputer system intended for OEM applications. The subsystem is composed of Intel microcomputer modules which are housed in a card cage and interconnected by a printed circuit motherboard. The chassis has space allocated for OEM power supplies, fan, and front panel.

Four modules are supplied with the basic system and expansion capability exists for 12 additional Intel-supplied or custom modules. Control signals, data and address lines are present at the 12 expansion connectors.



# BAREBONES 80

## SPECIFICATIONS

**WORD SIZE**
Data: 8 bits
Instruction: 8, 16, or 24 bits

**MEMORY SIZE**
6K bytes expandable to 16K bytes with standard modules, 64K bytes using custom memory modules.

**INSTRUCTION SET**
78, including conditional branching, binary arithmetic, logical, register-to-register, input/output, and memory reference.

**MACHINE CYCLE TIME**
2.5 $\mu$s. (Reduction to 2.0 $\mu$s possible by using faster memory and appropriate bus control signals.)

**SYSTEM CLOCK**
Crystal controlled at 2 MHz ±0.01%.

**I/O CHANNELS**
Maximum Input/Output configuration available with I/O or Output Modules

| | Input Ports | Output Ports |
|---|---|---|
| imm8-61 | 16 | 16 |
| imm8-63 (with one imm8-61) | 4 | 28 |

**INTERRUPT**
User-designed multiple level interrupt capability.

**DIRECT MEMORY ACCESS**
User-designed DMA capability.

**MEMORY ACCESS TIME**
RAM: 1 $\mu$s with standard RAM module. Faster access time available with user-designed memory systems.
PROM: 1.3 $\mu$s with 8702A PROMs. Faster access time available with higher speed PROMs.

**PHYSICAL CHARACTERISTICS**
6¾" X 17" X 12" (suitable for mounting in standard RETMA 7" X 19" panel space).
Weight: 11 lb (4.9 kg).

**ELECTRICAL CHARACTERISTICS**
DC Power Requirement:
$V_{CC}$ = 5V ±5%,
$I_{CC}$ = 6A max., 3.5A typ.
$V_{DD}$ = -9V ±5%,
$I_{DD}$ = 1.2A max., 0.8A typ.
$V_{CC}$ = +12V ±5%,
$I_{GG}$ = 0.06A max., 0.04A typ.

*Requirement based on basic Barebones 80 system.

**ENVIRONMENTAL CHARACTERISTICS**
Operating Temperature: 0°C to 70°C

**OPTIONAL MODULES**
Available for Barebones 80:
- imm8-61 I/O Module
- imm8-63 Output Module
- imm6-28 RAM Memory Module
- imm6-70 Universal Prototype Module
- imm6-72 Module Extender

**EQUIPMENT SUPPLIED**
Central Processor Module
Input/Output Module
PROM Memory Module
RAM Memory Module
Chassis with Mother Board
PROM Resident System Monitor
Complete Hardware and Software Documentation including schematics and assembly drawings
Rack Slides

# INTELLEC® 4/MOD 40
# MICROCOMPUTER DEVELOPMENT SYSTEM

**Complete Hardware/Software Development System for the design and implementation of 4040 and 4004 CPU based microcomputer systems**

**TTY interface, front panel designer's console, and high speed paper tape reader interface, in conjunction with PROM resident system monitor provide complete program loading, punching, monitoring, interrogation, and alteration capabilities**

**Program RAM (4K 8-bit bytes) provides a program development medium which lends itself to rapid and facile program monitoring and alteration**

**Data RAM (320 4-bit bytes expandable to 2560 bytes) provides data storage capacity**

**Program PROM (expandable to 4K 8-bit bytes) in conjunction with the resident PROM programmer provide capability of simulating final ROM resident program**

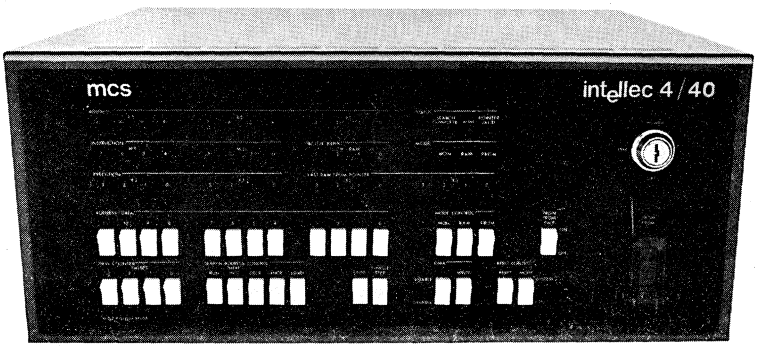
**PROM resident system monitor, RAM resident assembler with edit feature included in standard systems software**

**Includes such standard program development features as program single step, address search (and pass count), next instruction indication, program flow verification**

**I/O expandable to 16 4-bit input ports and 48 4-bit output ports (all TTL compatible) allowing "hands-on" simulation of entire user system (processor and peripheral devices)**

**RESET, STOP, INTERRUPT control signals available to user via back panel**

**Modular design with expansion capability provided for up to eleven optional or user designed modules**




The Intellec® 4/MOD 40 (imm 4-44A) is a complete, self-contained microcomputer development system designed specifically to support the development and implementation of 4040 and 4004 CPU based mirocomputer systems. Its modular design provides the flexibility to adapt to any size user system and the resident software facilitates program development.

The basic Intellec 4/MOD 40 Microcomputer Development System consists of 4 microcomputer modules (CPU, RAM, MEMORY CONTROL, and PROM PROGRAMMER), power supplies, I/O connectors, console, and displays. The heart of the system is the imm 4-43 central processor module built around Intel's high performance 4-bit 4040 CPU. The imm 4-43 is a complete microcomputer system containing the system clock, 1K 8-bit bytes of PROM memory, 320 4-bit bytes of data RAM memory, 3 4-bit input ports and 8 4-bit output ports. The imm 6-28 program RAM memory module contains a 4K x 8 memory array composed of Intel 2102 static random access memory elements. The imm 4-72 control module contains the circuitry required to interface the central processor module to the program RAM module. The imm 6-76 PROM programmer module provides the capability of programming Intel 1702A PROMs in conjunction with the front panel PROM socket and system monitor. All I/O ports are TTL compatible and accessible from the back panel 37-pin connectors. The front panel designer's console provides a means of monitoring and controlling system operation.

The Intellec modular design allows great design system flexibility. Program PROM can be expanded to 4K 8-bit bytes using imm 6-26 or imm 4-22 optional modules. Data RAM can be expanded to 2560 4-bit bytes using imm 4-24 modules. I/O capability can be expanded to 16 4-bit input and 48 4-bit output ports using optional imm 4-60 and 4-24 modules. The universal prototype card (imm 6-70) in conjunction with the eleven optional card sockets (which contain all essential system signals) provide the capability for interfacing custom designed modules.

The user RESET IN/OUT, STOP/STOP ACKNOWLEDGE, and INTERRUPT/INTERRUPT ACKNOWLEDGE control signals are all available at the back panel. Hence, the user can interrupt, halt, and reset the resident CPU via his own interface.

# INTELLEC®4
# HIGH SPEED PAPER TAPE READER

**Directly compatible with all Intellec® 4 Microcomputer Development Systems**

**20 times faster than standard ASR-33 teletype reader**

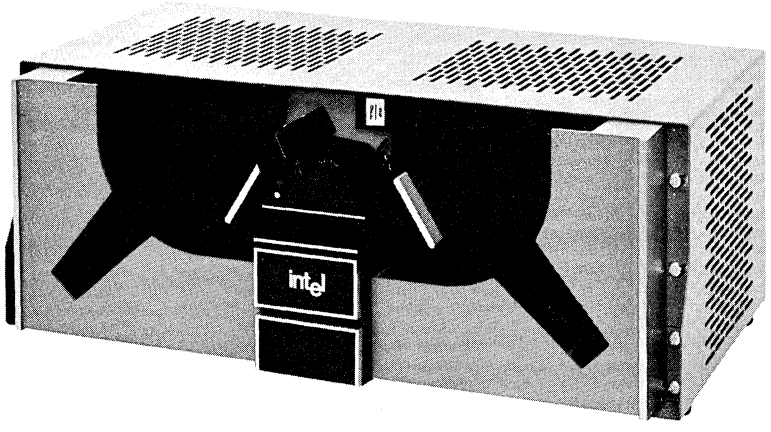
**Data transfer at asynchronous rates in excess of 200 characters per second**

**Rack mountable or stand-alone**

The imm4-90 high speed paper tape reader provides all Intellec® 4 Microcomputer Development Systems with a high speed paper tape input that is twenty times faster than the standard ASR-33 teletype reader. This translates into a significantly faster development cycle due to a marked reduction in the time required for repetitive program loading, assembly, and editing operations.

The Intellec 4 monitor provides the capability of assigning the imm4-90 as an input device and contains the reader driver software. Tapes may be read in BNPF or hexadecimal format.

At least one optional imm4-60 Input/Output Module must be included in the Intellec system to provide the required reader input and output ports.




## SPECIFICATIONS

**TAPE MOVEMENT:**
Tape Reading Speed
0 to 200 characters per second asynchronous
Tape Stopping
Stops "On Character"

**TAPE CHARACTERISTICS:**
Tapes must be prepared to ANSI X 3.18 or EMCA 10 Standards for base materials and perforations.

Reads tape of any material with thickness between 0.0027" and 0.0045" with transmissivity less than or equal to 5% (oiled buff paper tape).

Tape loading: in line

Tape width: 1 inch

**ELECTRICAL CHARACTERISTICS:**
AC Power Requirement
3 wire input with center conductor (earth ground) tied to chassis. 100, 115, or 127 VAC, single phase at 3.0 amps or 220 or 240 VAC and 1.5 amps; 47 to 63 Hz.

**EQUIPMENT SUPPLIED:**
Paper Tape Reader
Reader Cable
Reader Flat Cable
Fanfold Tape Guide
Fanfold Paper Tape
Hardware Manual
Installation and Operations Guide
Fanfold Guide Installation Instructions

*NOTE: Operation of the imm4-90 in conjunction with the Intellec 4/MOD 4 and Intellec 4/MOD 40 requires Version 2.0 software.*

# 8080 SYSTEM DESIGN KIT SDK-80

**Complete single board microcomputer system including CPU, memory and I/O**

**Easy to assemble kit-form**

**High-performance (2 $\mu$s instruction cycle)**

**Interfaces directly with most terminals (75-4800 baud)**

**Large wire-wrap area for custom interfaces**

**Extensive system monitor software in ROM**

**PC board format and power, compatible with Intellec®MDS**

The 8080 System Design Kit (SDK-80) is a complete, single board, microcomputer system in kit form. It contains all necessary components, including resistors, caps, crystal and miscellaneous hardware to complete construction. Included is a pre-programmed ROM that contains the system monitor for general software utilities and system diagnostics.

All that is required for operation are power supplies and a suitable terminal; TTY, CRT, etc., (level conversions and baud rate generation included on board).

The SDK-80 is an inexpensive, high-performance prototype system that has designed-in flexibility for simple interface to the users application.


DEVELOPMENT SYSTEMS

## SPECIFICATIONS

**CENTRAL PROCESSOR**
CPU: 8080A
Instruction Cycle: 1.95 microsecond
Tcy: 488 ns

**MEMORY**
ROM: 2K bytes (expandable to 4K bytes) 8708/8308
RAM: 256 bytes (expandable to 1K bytes) 8111
Addressing: ROM 0000-0FFF
RAM 1000-13FF

**INPUT/OUTPUT**
Parallel: One 8255 for 24 lines (expandable to 48 lines).
Serial: One 8251 USART.
On-board baud rate generator (jumper selectable).
Baud Rates: 75, 110, 300, 600, 1200, 2400, 4800

**INTERFACES**
Bus: All signals TTL compatible.
Parallel I/O: All signals TTL compatible.
Serial I/O: RS232C/EIA
20 mA current loop TTY
TTL (one TTL load)

**INTERRUPTS**
Single level: Generates RST7 vector.
TTL compatible input.

**DMA**
Hold Request: Jumper selectable.

**SOFTWARE**
System Monitor: Pre-programmed 8708 or 8308 ROM
Addresses; 0000-03FF.
Features:

| | |
|---|---|
| Display Memory Contents | (D) |
| Move blocks of memory | (M) |
| Substitute memory locations | (S) |
| Insert hex code | (I) |
| Examine Registers | (X) |
| Program Control | (G) |
| Break Point Capability | |
| Power-up start or system reset start. | |

I/O: Console Device (serial I/O)

**LITERATURE**
Design Library:
8080 Users Manual
8080 Assembly Language Manual
PL/M Programming Manual
MDS Brochure
Reference Card (Programmers)
SDK-80 User's Guide

**CONNECTORS**
I/O: 25 pin female (RS232C)
PCB: MDS format

**PHYSICAL CHARACTERISTICS (MDS MECHANICAL FORMAT)**
Width: 12.0 in.
Height: 6.75 in.
Depth: 0.50 in.
Weight: approx. 12 oz.

**ELECTRICAL CHARACTERISTICS (DC POWER)**

| | |
|---|---|
| $V_{CC}$ 5V ±5% | 1.3 Amps |
| $V_{DD}$ 12V ±5% | 0.35 Amps |
| $V_{BB}$ −10V ±5% | 0.20 Amps |
| or −12V ±5% | |

# MICROCOMPUTER MODULES

## MCS-4/40™

Modules may be ordered individually. All modules are 8" wide, 6.18" high and use standard 100-pin connectors.

### imm4-42 Central Processor Module

- This is a complete microcomputer system with the processor, program storage, data storage, and I/O in a single module.
- The heart of this module is Intel's 4004 single chip four-bit parallel processor — p-channel silicon gate MOS.
- Accumulator and sixteen working registers (4-bit).
- Subroutine nesting up to 3 levels.
- For development work, the CPU interfaces to standard semiconductor memory elements (provided by Intel's standard memory and I/O interface set 4008/4009).
- Sockets for 1K bytes of PROM (Intel 1702A PROM) are provided.
- 320 words (4-bit) of data storage (Intel 4002) are provided.
- Four 4-bit input ports and eight 4-bit output ports (includes TTY interface).
- Bus-oriented expansion of memory and I/O.
- Two phase crystal clock.

### imm4-43 Central Processor Module

- Complete microcomputer system with Intel's high performance 4040 4-bit processor, program storage, data storage, I/O and system clock in a single module.
- 60 instructions including decimal arithmetic, register-to-register transfers, conditional branching, logical operations and I/O.
- Interrupt capability.
- Single step capability.
- 24 index registers.
- Subroutine nesting to 7 levels.
- Direct interface capability to all standard memories (i.e., TTL, NMOS, PMOS, CMOS) through Intel's 4289 Standard Memory Interface chip.
- Sockets for 1K x 8 bytes of program memory (Intel 4702A PROM) expandable to 4K x 8 using optional imm6-26 or imm4-24 modules.
- 320 4-bit bytes of data storage (Intel 4002) expandable to 2560 x 4 using optional imm4-22 or imm4-24 modules.
- Four 4-line input ports and eight 4-line output ports expandable to 16 input and 48 output ports using optional imm4-60, imm4-22 or imm4-24 modules.
- Two phase crystal clock.

### imm4-22 Instruction/Data Storage Module

- This microcomputer module has memory capacity identical to the Central Processor Module and is used for expanding memory and I/O.
- Sockets for 1K bytes of PROM program storage are provided.
- 320 words (4-bit) of data storage are provided.
- Four 4-bit input ports and eight 4-bit output ports.

### imm4-24 Data Storage Module

- This microcomputer module has capacity for sixteen Intel 4002 RAMS — 1280 words (4-bit) of data storage.
- 320 words (4-bit) of data storage are provided.
- A maximum Intellec 4 system may contain up to 2560 words of storage — decoding for this expansion is provided.
- A 4-bit output port is associated with each RAM on this microcomputer module providing sixteen 4-bit output ports on each module.
- All output ports are TTL compatible.

### imm4-60 Input/Output Module

- This module provides input and output port expansion without additional memory.
- Eight 4-bit input ports and eight 4-bit output ports are provided.
- Ports on this module are TTL compatible.

## MCS-8™

### imm8-82 Central Processor Module

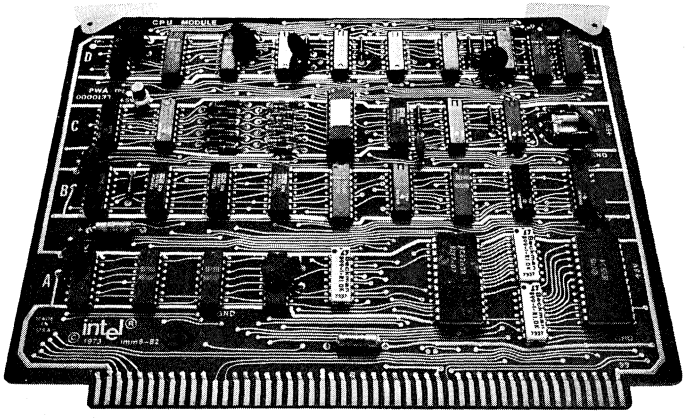
- Intel's 8008 eight-bit parallel single chip CPU — p-channel silicon gate MOS.
- Accumulator and six 8-bit working registers.
- Subroutine nesting up to seven levels.
- Interface to 16K 8-bit bytes of PROM, ROM, or RAM via the PROM Memory Module and RAM Memory Module.
- Interface for expansion to eight 8-bit input ports and twenty-four 8-bit output ports, via the I/O and Output Modules.
- Interrupt capability.
- Two phase crystal clock.
- All module interfaces are TTL compatible.

### imm8-60 Input/Output Module

- Four 8-bit input ports (32 lines).
- Four 8-bit data latching output ports (32 lines).
- One pair of ports for TTY communication.
- All input and output ports are TTL compatible.

### imm8-62 Output Module

- Eight 8-bit data latching output ports (64 lines).
- All output ports are TTL compatible.




**imm8-82 Central Processor Module**

## MCS-80™

### imm8-83 CPU Module

- Complete 8-bit parallel central processor module with system clocks, interface and control for memory, I/O ports, and real time interrupt.
- Utilizes Intel's high performance 8080 single chip n-channel microcomputer.
- 2.5 μsecond instruction execution time.
- 78 basic instructions including the entire 8008 instruction set.
- Direct addressing of up to 64K bytes of any speed ROM, PROM, or RAM memory.
- Unlimited subroutine nesting.
- Seven working registers — six 8-bit general purpose registers and an 8-bit accumulator.
- Separate 16-bit address bus, 8-bit output bus and 3 multiplexed 8-bit input busses for I/O input, memory input and interrupt data.
- Direct addressing of 256 input and 256 output ports.
- Multiple level real time interrupt capability.
- Direct memory access capability.
- All buses TTL compatible.

### imm8-61 I/O Module

- Four 8-bit input and four 8-bit latching output ports.
- Directly compatible with imm8-83 central processor module.
- Integral asynchronous serial data communications capability and teletype interface.
- Jumper selectable transmission rates of 110, 1200 or 2400 baud.
- Crystal controlled clock.
- Capable of high speed serial communications to 9600 baud.
- TTL compatible.

### imm8-63 Output Module

- Eight 8-bit latching output ports.
- Directly compatible with imm8-83 central processor module.
- Decoding provided for the selection of up to 256 individual output ports.
- TTL compatible.

## COMMON SYSTEM MODULES

### imm6-26 PROM Memory Module

- Provides sockets for up to sixteen 1702A electrically programmable and erasable PROMs for a system's fixed program memory (maximum 4K bytes/module).
- For volume requirements, Intel 2048-bit mask programmed MOS ROMs (1302) may be substituted in the same module.

### imm6-28 RAM Memory Module

- A 4K x 8 n-channel MOS memory system using Intel's 1024-bit Static RAM (2102).
- Address latching, data latching, and module select decoding are provided on the card.
- Provides program storage for up to 4K instructions.

### imm6-70 Universal Prototype Module

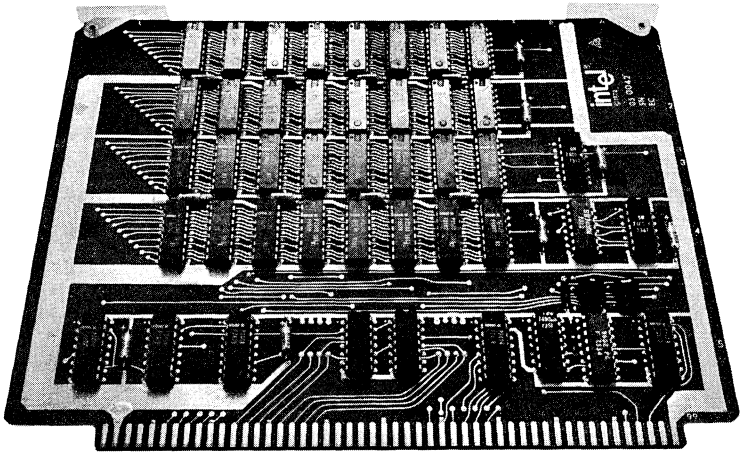
- Accommodates 14, 16, 24, or 40 pin wire wrap sockets (maximum of 52 16-pin sockets).
- Provides breadboard capability for developing custom and specialized interface circuits.

### imm6-72 Module Extender

- Extends Intellec modules out of card chassis for ease in test and system debugging.

### imm6-76 PROM Programmer Module

- Provides all timing and level shifting circuitry for programming Intel's programmable and erasable 1702A PROMs.



IMM 6-28 RAM MEMORY MODULE

# CONVERSION KITS

### imm4-88

The imm4-88 conversion kit provides an upgrade path for Intellec®4/MOD 4 microcomputer development systems. It includes all the hardware and software required to fully support 4040 CPU based microcomputer system development.

The conversion kit contains an imm4-43 CPU module, new memory controller, new front panel, and any software required.

NOTE: Due to necessary wiring changes, these conversions are done at the Intel factory. Contact local Intel salesmen or representatives for instructions.

### imm8-88

The imm8-88 conversion kit provides an upgrade path for Intellec®8/MOD 8 microcomputer development systems. It includes all the hardware and software products required to fully support 8080 CPU based microcomputer system development. With the imm8-88 conversion kit installed in an Intellec 8/MOD 8, complete 8080 CPU hardware and software development capability is provided.

The conversion kit is installed by simply plugging in the three new hardware modules in the appropriate Intellec 8/MOD 8 chassis connectors and installing the new system monitor. The system can be quickly reconfigured to support 8080 CPU chip development by replacing the original boards and system monitor.

# SBC 80/10 SINGLE BOARD COMPUTER

**8080A Central Processing Unit**

**1 K bytes of read/write memory**

**Sockets for 4K bytes of programmable or masked read-only memory**

**48 Programmable I/O lines with sockets for interchangeable line drivers and terminators**

**Programmable Synchronous/Asynchronous communications interface with selectable teletype or RS232C compatibility.**

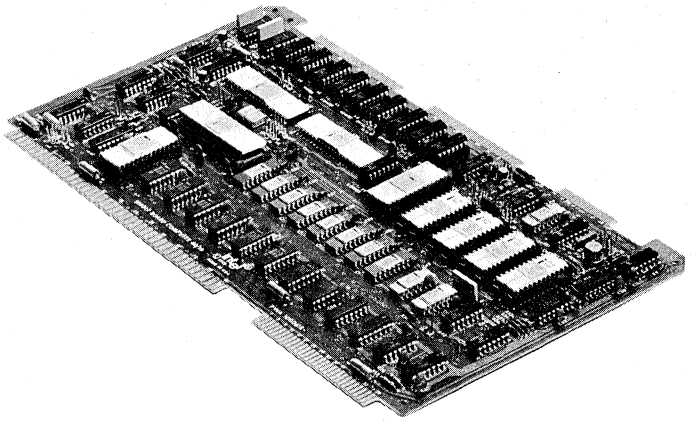
**Six interrupt request lines.**

**Bus drivers for memory and I/O expansion**

**Compatable with optional memory and I/O boards.**

The SBC-80/10 is a member of Intel's complete line of OEM computer systems which take full advantage of Intel's LSI technology to provide economical, self-contained computer based solutions for OEM applications. The SBC-80/10 is a complete computer system on a single 6.75-by-12 inch printed circuit card. The CPU, system clock, read/write memory, non-volatile read-only-memory, I/O ports and drivers, serial communications interface, bus control logic and drivers all reside on the board.

Memory and I/O expansion may be achieved using standard Intel boards. Memory may be expanded to 65,536 bytes by adding user specified combinations of SBC-016 16K byte RAM boards, SBC-406 6K byte PROM boards, and SBC-416 16K PROM boards. Input/output capacity may be expanded to 63 8-bit input ports and 63 8-bit output ports using SBC-508 Input/Output boards. Each I/O board contains four 8-bit input ports and four 8-bit output ports. Memory and I/O may be expanded simultaneously (i.e. 4K bytes of RAM, 4K bytes of PROM, and 48 programmable I/O lines, and a USART) by using the SBC-104 Combination board. Expandable backplanes and card-cages are available to support multi-board systems.





## SPECIFICATIONS

### WORD SIZE

Instruction: 8, 16, or 24 bits
Data: 8 bits

### CYCLE TIME

Basic Instruction Cycle: 1.95 μsec
Note: Basic instruction cycle is defined as the fastest instruction (i.e. four clock cycles)

### MEMORY ADDRESSING

On-Board ROM/PROM: 0-0FFF
On-Board RAM: 3C00-3FFF

### I/O CAPACITY

Parallel: 48 programmable lines

Note: Expansion to 504 input and 504 output lines can be accomplished using optional I/O boards.

Serial: (USART)

| Frequency (KHz) (Jumper Selectable) | Baud Rate (Hz) | | |
|---|---|---|---|
| | Synchronous | Asynchronous (Program Selectable) | |
| | | ÷16 | ÷64 |
| 307.2 | – | 19200 | 4800 |
| 153.6 | – | 9600 | 2400 |
| 76.8 | – | 4800 | 1200 |
| 38.4 | 38400 | 2400 | 600 |
| 19.2 | 19200 | 1200 | 300 |
| 9.6 | 9600 | 600 | 150 |
| 4.8 | 4800 | 300 | 75 |
| 3.49 | 3490 | – | 110 |

# SPECIFICATIONS

## MEMORY CAPACITY

On-Board ROM/PROM: 4K bytes (sockets only)
On-Board RAM: 1K bytes
Off-Board Expansion: Up to 65,536 bytes using user specified combinations of RAM, ROM, and PROM
NOTE: ROM/PROM may be added in 1K byte increments.

## I/O ADDRESSING

On-Board Programmable I/O

| Port | 1 | 2 | 3 | 4 | 5 | 6 | 8255 #1 Control | 8255 #2 Control | USART Data | USART Control |
|---|---|---|---|---|---|---|---|---|---|---|
| Address | E4 | E5 | E6 | E8 | E9 | EA | E7 | EB | EC | ED |

Synchronous:
5-8 bit characters
Internal or external character synchronization
Automatic Sync Insertion
Asynchronous:
5-8 bit characters
Break characters generation
1, 1-1/2, or 2 stop bits
False start bit detectors
Full duplex, double buffered transmitter and receiver
Parity, overrun, and framing error detection

## INTERRUPTS

Single-level with on-board logic that automatically vectors processor to location $38_{16}$ using RESTART 7 instruction. Interrupt requests may originate from user specified I/O (2) the programmable peripheral interface (2), or USART (2)

## INTERFACES

Bus: All signals TTL compatible
Parallel I/O: All signals TTL compatible
Serial I/O: RS232C or 20 mil current loop TTY interface (jumper selectable)
Interrupt Requests: All TTL compatible

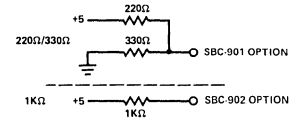
## LINE DRIVERS AND TERMINATORS

I/O Drivers:
The following line drivers and terminators are all compatible with the I/O driver sockets on the SBC-80/10.

| Driver | Characteristic | Sink Current (mA) |
|---|---|---|
| 7438 | I, OC | 48 |
| 7437 | I | 48 |
| 7432 | NI | 16 |
| 7426 | I, OC | 16 |
| 7409 | NI, OC | 16 |
| 7408 | NI | 16 |
| 7403 | I, OC | 16 |
| 7400 | I | 16 |

Note: I = inverting N.I. = non-inverting OC = open collector

I/O Terminators:
Terminators: 220Ω/330Ω divider or 1 kΩ*




*Must be ordered separately.

Bus Drivers:

| Function | Characteristic | Sink Current (mA) |
|---|---|---|
| Data | Tri-State | 25 |
| Address | Tri-State | 25 |
| Commands | Tri-State | 25 |

## SYSTEM CLOCK

2.048 MHz ± 0.1%

## CONNECTORS

Bus:
86 pin double-sided PC edge connector
0.156 inch centers
Mating Connector: Control Data Corp. VPB01E43A00A1
Parallel I/O:
Two 50 pin double-sided PC edge connectors
0.1 inch centers
Mating Connector: 3M 3415-000 or TI H312125
Serial I/O:
26 pin double-sided PC edge connector
0.1 inch centers
Mating Connector: 3M 3462-000 or TI H312113

## PHYSICAL CHARACTERISTICS

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.14 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 14 oz. (0.48 Kg)

## ELECTRICAL CHARACTERISTICS

DC Power:

| | |
|---|---|
| $V_{CC}$ = +5 ± 5% | $I_{CC}$ = 2.9 A max |
| $V_{DD}$ = +12 ± 5% | $I_{DD}$ = 150 mA max |
| $V_{BB}$ = −5V ± 5% | $I_{BB}$ = 2 mA max |
| $V_{SS}$ = −12V ± 5% | $I_{SS}$ = 150 mA max |

Note: Does not include power required for options PROM, I/O drivers, and I/O terminators.

# MICROCOMPUTER SOFTWARE PRODUCTS

The following section contains information on Intel's Cross Software Products and User's Library. These cross products are all written in FORTRAN IV and are designed to run on a large computer system while generating code for or simulating one of Intel's microcomputers. All these products are also available on several computer timesharing services worldwide.

Included among these products are the PL/M™ compilers. The PL/M™ high level programming language was developed by Intel for the 8008 and 8080 microcomputers. Use of this language can significantly reduce programming time and costs.

A partial list of programs in the Intel microcomputer User's Library is also included. New programs are constantly being added to the library in a continuing effort to increase the size of the largest commercially available library of microcomputer programs in the world. You are encouraged to become a member to reap the benefits of the knowledge and experience of hundreds of users. Contributed programs are gratefully accepted.

---


DEVELOPMENT SYSTEMS

# MCS-40 CROSS ASSEMBLER

**Accepts all 4004 and 4040 instructions**

**Conditional assembly capability**

**Full macro facility**

**Hexadecimal or BNPF object code formats**

**Written in ANSI standard FORTRAN IV**

**Comprehensive user documentation**

**Instantly available on worldwide timesharing services**

The MCS-40 Cross Assembler, MAC40, is a powerful program development tool for Intel's® 4-bit microcomputers, the 4004 and the new 4040. MAC40 provides extensive capabilities which reduce the time and effort involved in program development, debug and documentation. The cross assembler allows usage of the high speed I/O and text editing capability of a large computer system to further shorten the programming task.

MAC40 translates 4004/4040 machine assembly language instructions into the appropriate machine operation codes. In addition to eliminating the errors of hand translation, the ability to refer to program addresses with symbolic names makes it easy to modify programs by adding or deleting instructions, or to move the program to another memory location. Full macro capability eliminates the need to rewrite similar sections of code repeatedly and simplifies program documentation. Conditional assembly permits MAC40 to include or delete sections of code which may vary from system to system, such as the code required to handle optional external devices.

Output from MAC40 may be punched to paper tape in hex format for loading into an Intellec® 4 Development System or may be punched in BNPF format to program ROMs.

MAC40 is written in ANSI standard FORTRAN IV and is designed to run on most large scale computing systems with a 32-bit or larger integer format (word size). It is also available for immediate use on several major timesharing services throughout the world.

## SPECIFICATIONS

**CAPABILITIES:**

Accepts all 66 instruction mnemonics plus 10 pseudo-operators.

Allows up to 499 labels in standard configuration; easily expandable.

Allows a total of up to 9 levels of nested conditional assembly and nested macro-calls.

User definable I/O formats.

Batch or interactive mode.

**OPERATIONAL ENVIRONMENT:**

Hardware required

32-bit or larger word size

12–16K words depending on machine

Software required

ANSI standard FORTRAN IV compiler

**SHIPPING MEDIA:**

Magnetic tape

**TAPE FORMAT:**

| | |
|---|---|
| 9 Track | 80 Byte unblocked records |
| EBCDIC | |
| 800 BPI | Unlabeled |

**TAPE CONTENTS:**

MCS-40™ Cross Assembler (FORTRAN IV Source)

MERGE Source File Editing Program (FORTRAN IV Source)

XCNV4 Conversion Program (FORTRAN IV Source)

**DOCUMENTATION PACKAGE INCLUDES:**

4004/4040 Assembly Language Programming Manual

MAC4 External Reference Specification

Pocket Reference Card

# MCS-8™ CROSS ASSEMBLER

**Accepts all 8008 instructions**

**Conditional assembly capability**

**Full macro facility**

**Hexadecimal or BNPF object code formats**

**Written in ANSI standard FORTRAN IV**

**Comprehensive user documentation**

**Instantly available on worldwide timesharing services**

The MCS-8™ Cross Assembler, MAC8, is a powerful program development tool for Intel's® 8008 microcomputer. MAC8 provides extensive capabilities which reduce the time and effort involved in program development, debug and documentation. The cross assembler allows usage of the high speed I/O and text editing capability of a large computer system to further shorten the programming task.

MAC8 translates symbolic 8008 assembly language instructions into the appropriate machine operation codes. In addition to eliminating the errors of hand translation, the ability to refer to program addresses with symbolic names makes it easy to modify programs by adding or deleting instructions, or to move the program to another memory location. Full macro capability eliminates the need to rewrite similar sections of code repeatedly and simplifies program documentation. Conditional assembly permits MAC8 to include or delete sections of code which may vary from system to system, such as the code required to handle optional external devices.

Output from MAC8 may be loaded directly to the 8008 Simulator (INTERP/8) for interactive, symbolic debugging or may be punched to paper tape in hex format for loading into an Intellec® 8/Mod 8 Development System. It may also be punched in BNPF format to program ROMs.

MAC8 is written in ANSI standard FORTRAN IV and is designed to run on most large scale computing systems with a 32-bit or larger integer format (word size). It is also available for immediate use on several major timesharing services throughout the world.

**MCS-8™ ASSEMBLY LANGUAGE PROGRAMMING EXAMPLE:**

```
                                ; UMUL – UNSIGNED INTEGER MULTIPLY
                                ; CALL:  ARGUMENTS IN C & D
                                ; EXIT:  HI ORDER PRODUCT IN B
                                ;        LO ORDER PRODUCT IN C
                                ; REGS:  A,B,E AND FLAGS EXCEPT CARRY ALTERED
001E                            UMUL:
001E  0E00                              MVI B,0
0020  2609                              MVI E,9
0022                            UMUL0:
0022  C2                               MOV A,C         ; ROTATE CARRY INTO
0023  1A                               RAR             ; PRODUCT – MULTIPLIER
0024  D0                               MOV C,A         ;  SHARED REGUSTER
0025  21                               DCR E           ;   FORCING NEXT LSB
0026  2B                               RZ              ;    INTO CARRY
0027  C1                               MOV A,B         ; EXIT IF 8TH ITERATION
0028  402C00                           JNC UMUL1       ; IF CARRY SET
002B  83                               ADD D           ; ADD MULTIPLICAND TO '
002C                            UMUL1:                  ; PRODUCT
002C  1A                               RAR
002D  C8                               MOV B,A         ; ROTATE MOST SIGNIFICANT
002E  442200                           JMP UMUL0       ;  PRODUCT AND REPEAT LOOP
```

## SPECIFICATIONS

**CAPABILITIES:**

Accepts all 48 instruction mnemonics plus 10 pseudo-operators.

Allows up to 499 labels in standard configuration; easily expandable.

Allows total of up to 9 levels of nested conditional assembly and nested macro-calls.

User definable I/O formats.

Batch or interactive mode.

**OPERATIONAL ENVIRONMENT:**

Hardware required

32-bit or larger word size

12—16K words depending on machine

Software required

ANSI standard FORTRAN IV compiler

**SHIPPING MEDIA:**

Magnetic tape

**TAPE FORMAT:**

| | |
|---|---|
| 9 Track | 80 Byte unblocked |
| EBCDIC | records |
| 800 BPI | Unlabeled |

**TAPE CONTENTS:**

MCS-8™ Cross Assembler (FORTRAN IV Source)

MERGE Source File Editing Program (FORTRAN IV Source)

CONV8 Conversion Program (FORTRAN IV Source)

**DOCUMENTATION PACKAGE INCLUDES:**

8008 Assembly Language Programming Manual

MAC8 External Reference Specification

Pocket Reference Card

DEVELOPMENT SYSTEMS

# 4004/4040 SIMULATOR

**Simulates all 4004/4040 machine instructions**

**Accepts output from MAC40, the Intel® 4004/4040 Cross Assembler**

**Contains extensive symbolic debugging capabilities**

**Written in ANSI standard FORTRAN IV**

**Instantly available on worldwide timesharing services**

**COMMAND CAPABILITIES:**

**Set breakpoints**

**Trace program execution**

**Dump and modify memory**

**Examine and modify registers**

**Examine and set I/O ports**

**Simulate the 4040 hardware interrupt**

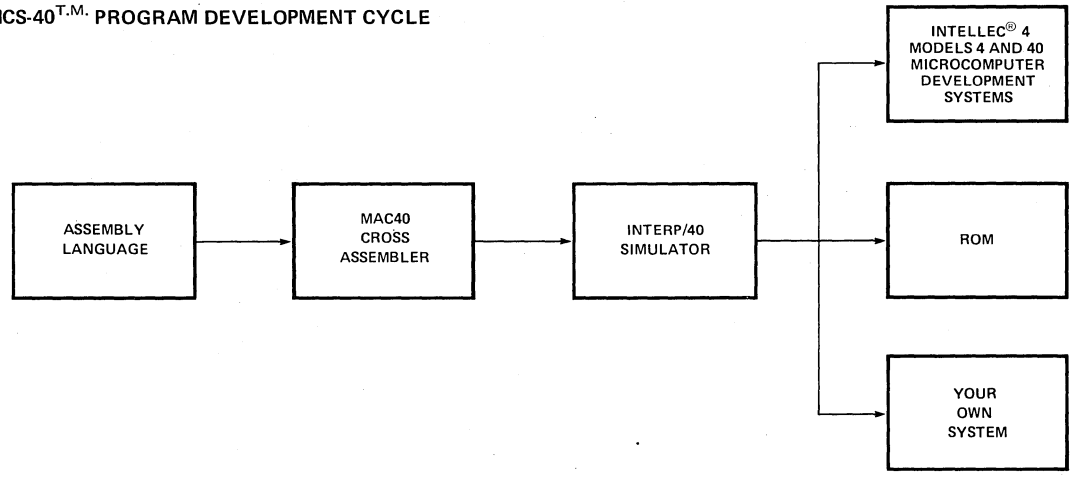
**Measure program execution time**

The 4004/4040 Simulator, INTERP/40, is a complete simulation and debug program for the Intel® 4004 and 4040 microcomputers. Programs can be run, displayed, stopped, and altered allowing step by step refinement without continuous reassembly of the source program. INTERP/40 provides powerful commands to control the execution of 4004 and 4040 programs. Extensive debug features are built in to help reduce the time and cost involved in program checkout.

INTERP/40 also provides symbolic reference to storage locations and operation codes as well as numeric reference in various number bases.

INTERP/40 is written in FORTRAN IV and is designed to run on most large scale computer systems with a 32-bit or larger integer format (word size). It is also available for immediate use on major timesharing services throughout the world.

MCS-40™ PROGRAM DEVELOPMENT CYCLE




## SPECIFICATIONS

**CAPABILITIES:**
Provides total software simulation of the Intel® 4004 and 4040 CPU's.

Can be run in batch or interactive mode.

**OPERATIONAL ENVIRONMENT:**
Hardware required
- 32-bit or larger word size
- 12-15K words of memory, depending on machine

Software required
- FORTRAN IV compiler

**SHIPPING MEDIA:**
Magnetic tape

**TAPE FORMAT:**

| | |
|---|---|
| 9-track | 80-byte unblocked |
| EBCDIC | records |
| 800BPI | Unlabeled |

**TAPE CONTENTS:**
4004/4040 Simulator
(FORTRAN IV Source)
MERGE Source File Editing Program
(FORTRAN IV Source)

**DOCUMENTATION PACKAGE:**
INTERP/40 External Reference Specification

DEVELOPMENT SYSTEMS

# 8008 SIMULATOR

**Simulates all 8008 machine instructions**

**Accepts output from PL/M[T.M.] compiler or MAC8 cross assembler**

**Comprehensive debug features**

**Written in FORTRAN IV**

**Instantly available on worldwide timesharing services**

**Comprehensive user documentation**

The 8008 Simulator, INTERP/8, is a complete simulation and debug program for the Intel® 8008 microcomputer. INTERP/8 provides powerful commands to control the execution of 8008 programs. Extensive debug features are built-in to help reduce the time and cost involved in program checkout.

INTERP/8 simulates execution of all 8008 machine instructions. Programs either compiled on the PL/M[T.M.] compiler or assembled on the MAC8 Cross Assembler may be loaded directly into INTERP/8 for simulation and checkout.

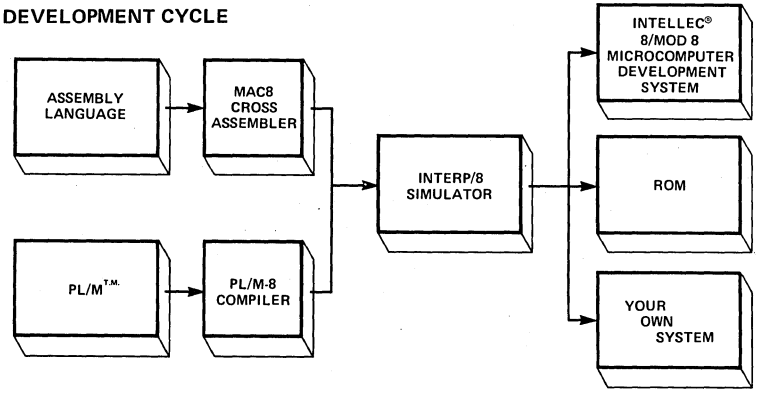
INTERP/8 provides commands to:

- Set Breakpoints
- Trace Program Execution
- Dump and Modify Memory
- Examine and Modify Registers
- Measure Program Timing
- Examine and Set I/O Ports
- Perform Interrupts and Stack Manipulations
- Perform Address Arithmetic

INTERP/8 also provides symbolic debugging capability. Memory locations may be referenced by their symbolic names, either through labels or variable names. This eliminates the need to know the specific absolute address of each variable or label.

INTERP/8 is written in FORTRAN IV and is designed to run on most large-scale computer systems with a 32-bit or larger integer format (word size). It is also available for immediate use on several major timesharing services throughout the world.

**MCS-8[T.M.] PROGRAM DEVELOPMENT CYCLE**




## SPECIFICATIONS

**CAPABILITIES:**
Simulates all 48 machine instructions

Allows full 16K program

Can be run in batch or interactive mode

**OPERATIONAL ENVIRONMENT:**
Hardware required
32-bit or larger word size
15—20K words of memory, depending on machine
Software required
FORTRAN IV compiler

**SHIPPING MEDIA:**
Magnetic tape

**TAPE FORMAT:**

| | |
|---|---|
| 9-track | 80-byte unblocked |
| EBCDIC | records |
| 800 BPI | Unlabeled |

**TAPE CONTENTS:**
8008 Simulator
(FORTRAN IV Source)
MERGE Source File Editing Program
(FORTRAN IV Source)

**DOCUMENTATION PACKAGE:**
INTERP/8 User's Manual
INTERP/8 Installation Guide

# MCS-80™ CROSS ASSEMBLER

**Accepts all 8080 instructions**

**Conditional assembly capability**

**Full macro facility**

**Hexadecimal or BNPF object code formats**

**Written in ANSI standard FORTRAN IV**

**Comprehensive user documentation**

**Instantly available on worldwide timesharing services**

The MCS-80 Cross Assembler, MAC80, is a powerful program development tool for Intel's® 8080 microcomputer. MAC80 provides extensive capabilities which reduce the time and effort involved in program development, debug and documentation. The cross assembler allows usage of the high speed I/O and text editing capability of a large computer system to further shorten the programming task.

MAC80 translates symbolic 8080 assembly language instructions into the appropriate machine operation codes. In addition to eliminating the errors of hand translation, the ability to refer to program addresses with symbolic names makes it easy to modify programs by adding or deleting instructions, or to move the program to another memory location. Full macro capability eliminates the need to rewrite similar sections of code repeatedly and simplifies program documentation. Conditional assembly permits MAC80 to include or delete sections of code which may vary from system to system, such as the code required to handle optional external devices.

Output from MAC80 may be loaded directly to the 8080 Simulator (INTERP/80) for interactive, symbolic debugging or may be punched to paper tape in hex format for loading into an Intellec® MDS Microcomputer Development System. It may also be punched in BNPF format to program ROMs.

MAC80 is written in ANSI standard FORTRAN IV and is designed to run on most large scale computing systems with a 32-bit or larger integer format (word size). It is also available for immediate use on several major timesharing services throughout the world.

---

## MCS-80™ ASSEMBLY LANGUAGE PROGRAMMING EXAMPLE:

```
                        ;       THIS SUBROUTINE PERFORMS DECIMAL ADDITION
                        ;       FOR 16 DECIMAL DIGITS ON THE INTEL 8080
                        ;       MICROCOMPUTER.
                        ;
                        ;       THE ADDRESS OF THE FIRST OPERAND IS EXPECTED TO BE IN THE
                        ;       D AND E REGISTERS AND THE ADDRESS OF THE
                        ;       SECOND OPERAND SHOULD BE IN THE H AND L
                        ;       REGISTERS.  THE RESULT IS STORED OVER THE
                        ;       FIRST OPERAND.  THE ADDITION IS DONE TWO DIGITS
                        ;       AT A TIME.
                        ;
0000                    DECAD:
0000  0E08                      MVI C,8         ; INITIALIZE DIGIT COUNTER (HALF)
0002  AF                        XRA A           ; CLEAR CARRY BIT
0003                    LOOP:
0003  1A                        LDAX D          ; LOAD TWO DIGITS FROM FIRST OPERAND
0004  8E                        ADC M           ; ADD TWO DIGITS FROM SECOND OPERAND WITH CARRY
0005  27                        DAA             ; DECIMAL ADJUST RESULT
0006  12                        STAX D          ; STORE TWO DIGITS OF RESULTS OVER FIRST OPERAND
0007  23                        INX H           ; INCREMENT ADDRESS OF SECOND OPERAND
0008  13                        INX D           ; INCREMENT ADDRESS OF FIRST OPERAND
0009  0D                        DCR C           ; DECREMENT DIGIT COUNT
000A  C20300                    JNZ LOOP        ; CONTINUE IF MORE DIGITS LEFT
000D  C9                        RET
```

# 8080 SIMULATOR

**Simulates all 8080 machine instructions**

**Accepts output from PL/M[T.M.] compiler or MAC80 Cross Assembler**

**Comprehensive debug features**

**Written in FORTRAN IV**

**Instantly available on worldwide timesharing services**

**Comprehensive user documentation**

The 8080 Simulator, INTERP/80[T.M.], is a complete simulation and debug program for the Intel® 8080 microcomputer. INTERP/80 provides powerful commands to control the execution of 8080 programs. Extensive debug features are built in to help reduce the time and cost involved in program checkout.

INTERP/80 simulates execution of all 8080 machine instructions. Programs either compiled on the PL/M[T.M.] compiler or assembled on the MAC80 Cross Assembler may be loaded directly into INTERP/80 for simulation and checkout.

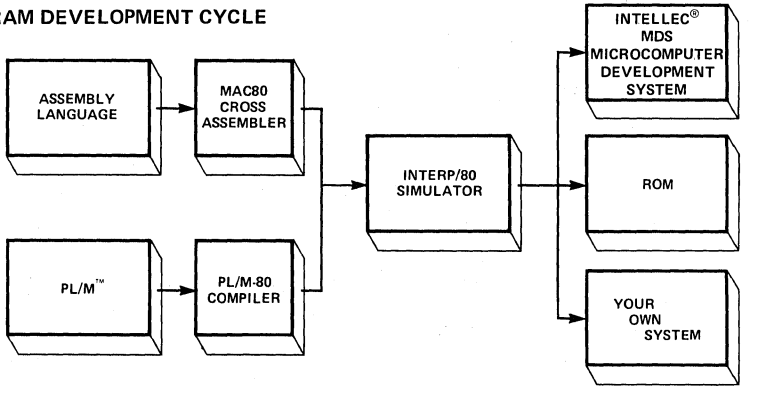
INTERP/80 provides commands to:

- Set Breakpoints
- Trace Program Execution
- Dump and Modify Memory
- Examine and Modify Registers
- Measure Program Timing
- Examine and Set I/O Ports
- Perform Interrupts and Stack Manipulations
- Perform Address Arithmetic

INTERP/80 also provides symbolic debugging capability. Memory locations may be referenced by their symbolic names, either through labels or variable names. This eliminates the need to know the specific absolute address of each variable or label.

INTERP/80 is written in FORTRAN IV and is designed to run on most large-scale computer systems with a 32-bit or larger integer format (word size). It is also available for immediate use on several major timesharing services throughout the world.

**MCS-80[T.M.] PROGRAM DEVELOPMENT CYCLE**




## SPECIFICATIONS

**CAPABILITIES:**
Simulates all 78 machine instructions

Allows 16K program, easily expandable

Can be run in batch or interactive mode

**OPERATIONAL ENVIRONMENT:**
Hardware required
32-bit or larger word size
15–20K words of memory, depending on machine
Software required
FORTRAN IV compiler

**SHIPPING MEDIA:**
Magnetic tape

**TAPE FORMAT:**

| | |
|---|---|
| 9-track | 80-byte unblocked |
| EBCDIC | records |
| 800 BPI | Unlabeled |

**TAPE CONTENTS:**
8080 Simulator
(FORTRAN IV Source)
MERGE Source File Editing Program
(FORTRAN IV Source)

**DOCUMENTATION PACKAGE:**
INTERP/80 User's Manual
INTERP/80 Installation Guide

intel®

# PL/M™ HIGH LEVEL PROGRAMMING LANGUAGE MCS-8™ AND MCS-80™ CROSS COMPILERS

**Reduces program development time and cost**

**Improves product reliability and eases maintenance**

**Available for 8008 and 8080**

**Comprehensive user documentation**

**Hexadecimal or BNPF object code formats**

**Written in ANSI standard FORTRAN IV**

**Instantly available on worldwise timesharing services**

PL/M is a high-level system programming language, specifically designed to ease the programming task for INTEL's 8-bit microcomputers, the 8008 and the 8080. PL/M is a powerful tool, well suited to the requirements of the microcomputer system designer and implementor. The language has been designed to facilitate the use of modern techniques in structured programming. These techniques can lead to rapid system development and checkout, straightforward maintenance and modification, and high product reliability.

The PL/M compilers convert a free-form symbolic PL/M program into an equivalent 8008 or 8080 object program. The compilers themselves take care of all the details of machine or assembly language programming, which permits the programmer to concentrate entirely on effective software design, and the logical requirements of his system.

Output from the PL/M compiler may be loaded directly into the 8008 or 8080 simulator programs for interactive, symbolic debugging or may be punched to paper tape in hex format for loading into an Intellec® Microcomputer Development System. It may also be punched in BNPF format to program ROMs.

The PL/M compilers are written in ANSI standard FORTRAN IV and are designed to run on any large-scale computer system with a minimum 32-bit integer format (word size). They are also available for immediate use on several worldwide timesharing systems.

---

## PL/M PROGRAMMING EXAMPLE:

```
                                    /*  BUBBLE SORT DECLARATION  */
SORT: PROCEDURE (N) ADDRESS;
    /*    N = LENGTH OF A
          COUNT = NR. OF SWITCHES PERFORMED TO-DATE
          SWITCHED = (BOOLEAN) HAVE WE DONE ANY SWITCHING YET ON THIS SCAN?  */
    DECLARE (N, I, SWITCHED) BYTE,
          (TEMP, COUNT) ADDRESS;
    SWITCHED = 1;              /*  SWITCHED = TRUE MEANS NOT DONE YET  */
    COUNT = 0;
    DO WHILE SWITCHED;
          SWITCHED = 0;        /*  BEGIN NEXT SCAN OF A  */
          DO I = 0 TO N-2;
                IF A(I) > A(I+1) THEN
                      DO;                  /*  FOUND A PAIR OUT OF ORDER  */
                      COUNT = COUNT + 1;
                      SWITCHED = 1;        /*  SET SWITCHED = TRUE  */
                      TEMP = A(I);         /*  SWITCH THEM INTO ORDER  */
                      A(I) = A(I+1);
                      A(I+1) = TEMP;
                      END;
          END;
          /*  HAVE NOW COMPLETED A SCAN  */
    END  /*  WHILE  */;
    /*  HAVE NOW COMPLETED A SCAN WITH NO SWITCHING  */
    RETURN COUNT;
END SORT;
```

## SPECIFICATIONS

**OPERATING ENVIRONMENT:**
Required hardware
32-bit or larger word size
20–25K words of memory, depending on machine
Required software
ANSI standard FORTRAN IV compiler

**SHIPPING MEDIA:**
Magnetic tape

**TAPE FORMAT:**
9-track EBCDIC
800 BPI 80-byte unblocked records
Unlabeled

**DOCUMENTATION PACKAGE:**
8008 and 8080 PL/M Programming Manual
8008 (or 8080) PL/M Compiler Operator's Manual

**TAPE CONTENTS:**
PLM Pass 1 (FORTRAN IV Source)
PLM Pass 2 (FORTRAN IV Source)
MERGE Source File Editing Program (FORTRAN IV Source)
Sample Test Program (PL/M Source)

# SERIES 3000 CROSS MICROPROGRAMMING SYSTEM

**Built-in series 3000 fields and mnemonics**
**User definable fields and mnemonics**
**Hierarchical field defaults**
**Free field statement format**
**String macro capability**
**Extended address generation**
**Graphical microprogram memory display**
**Symbolic label reference directory**
**MCU jump address validation**
**RAM/ROM/PROM programming file generation**

The Intel® Series 3000 Cross Microprogramming System, CROMIS, is an advanced software system that supports the generation of microprograms for custom Series 3000 processor and controller micro-architectures. It provides extensive programming facilities that greatly reduce the time and effort required to develop, debug, and document a microprogram.

CROMIS consists of two major software subsystems, XMAS and XMAP, XMAS is a symbolic microassembler which is dynamically user extensible in the size and structure of the target microinstruction format. XMAP is a complementary subsystem which maps the microinstruction bit patterns produced by XMAS into the desired physical microprogram memory locations.

In addition to providing four built-in microinstruction fields and corresponding mnemonic sets for the basic 3001 MCU and 3002 CPE functions, XMAS accepts user definitions for extended microinstruction fields and their associated mnemonics. Graphic debugging aids, string macro capability, definable defaults, and extended address generation further simplify the microprogramming of Series 3000 computing elements.

XMAP accepts the microinstruction file produced by XMAS and generates under user specifications one or more programming files for use with standard memory components. It enables the user to specify the mapping of the field into the physical bit positions of the microprogram memory components.

CROMIS is designed for use on almost any modern computing system with high speed I/O and on-line file facilities. It is available in ANSI (standard) FORTRAN IV source form for user installation or may be immediately accessed on any of several major timesharing services throughout the world. To insure the long term reliability and maintainability of CROMIS, all component programs are written in a highly modular, structured programming style with extensive operational documentation.

## SPECIFICATIONS

### XMAS CAPABILITIES

Translates all 3001 MCU and 3002 CPE mnemonics.

Dynamically allocates storage for labels, values and strings in a user expandable data area.

Accepts microinstruction format definitions of up to 64 total bits.

Provides extended address generation for up to 16K microinstructions.

Includes a four-level user definable field default mechanism.

### XMAP CAPABILITIES

Provides direct or inverted mapping for any bit in any microinstruction field.

Permits explicit 1's or 0's to be specified for unused bit locations.

Generates standard BNPF or hexadecimal programming files.

Accepts memory configuration definitions from 1 X 1 bits to 16K X 16 bits.

### OPERATIONAL ENVIRONMENT

Required hardware:
- 16-bit or larger word size
- 5 rewindable data files (disc or tapes)

Required software:
- ANSI standard FORTRAN IV compiler

### TAPE CONTENTS

TAPE 1
- Part 1 of XMAS FORTRAN IV Source

TAPE 2
- Part 2 of XMAS FORTRAN IV Source
- XMAS Sample Program
- XMAP FORTRAN IV Source
- XMAP Sample Program
- MERGE File Editing Program

### SHIPPING MEDIA

Two 2400' magnetic tapes

### TAPE FORMAT

| | |
|---|---|
| 9-track | 800 bpi |
| 80 byte | unblocked |
| EBCDIC | unlabeled |

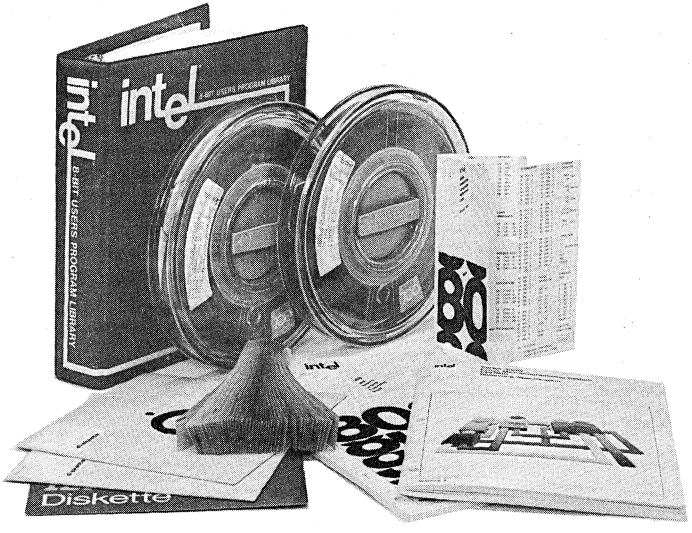
### DOCUMENTATION

Microprogramming the Series 3000
XMAS/XMAP Message Summary
XMAS Installation Guide
(preamble to XMAS FORTRAN source)
XMAP Installation Guide
(preamble to XMAP FORTRAN source)

# MICROCOMPUTER SOFTWARE LIBRARY
# USER'S PROGRAM LIBRARY

The Intel Microcomputer User's Library is a collection of programs, subroutines, procedures and macros written by users of Intel's 4004, 4040, 8008 or 8080 microcomputers. Thanks to customer contributions to the User's Library, Intel is now able to make these programs available to all users of Intel microcomputers. By taking advantage of the availability of these general purpose routines, the microcomputer design engineer and programmer can save many hours of programming and debugging.

A complete, documented listing of each program, procedure or macro in the user's library is sent to each member. This includes information on the program environment, required hardware and software, subroutine calling sequences and memory requirements. As new programs are submitted to the User's Library, they will be sent to all members. Strict documentation standards will be maintained to assure the usability of each library program by every interested member. Several of the available programs are listed in this brochure.




There are two user's libraries, one each containing programs for the Intel 4-bit and 8-bit microcomputer systems, respectively. Membership in each user's library is available to any interested person or organization. A yearly membership fee of $100 is charged, for which the member receives a manual of all programs in that library and all updates during the year. For those prospective members who submit a program to the library, a free one-year membership will be awarded. A submittal form for program donation is included in this catalog. More forms may be obtained from any Intel representative.

DEVELOPMENT SYSTEMS

**Ordering Information**

To become a member, send a program, purchase order, or check to:

User's Library
Microcomputer Systems
Intel Corporation
3065 Bowers Avenue
Santa Clara, California 95051

# MICROCOMPUTER SOFTWARE LIBRARY
# PARTIAL PROGRAM INDEX-4-BIT USERS LIBRARY

| TITLE | FUNCTION |
|---|---|
| **Cross Assembler on PDP-8** | Performs symbolic assembly for 4004 assembly language programs. The assembler runs on a DEC PDP-8 minicomputer. |
| **Cross Assembler for NOVA Computer** | Assemble MCS-4 programs and program PROMs. |
| **BNPF Tape Generator for PDP-8** | Produces a BNPF object tape from the output of the PDP-8 assembler modified to assemble MCS-4 programs. |
| **MCS-4 Simulator for PDP-8** | Simulates operation of MCS-4 system; allows breakpoints, dumps, modification of RAM or ROM, I/O under user control, provides cycle counter for timing. Accepts output of PAL8 assembler in binary form. |
| **MCS-40 Cross Assembler** | Program to perform assembly of 4040 programs on an Intellec 8/Mod 80. |
| **Intel MCS-40 Cross Assembler and Text Editor** | Edit and assemble MCS-40 source language using a Computer Automation Alpha-16/Alpha-LSI producing program listings, error diagnostics, source and object tapes. |

*All programs listed below will execute on the 4004 or 4040 CPU.*

| | |
|---|---|
| **A Chebyshev Approximation Package** | The package contains approximation routines for sine, cosine, arctangent, natural logrithm ($\log_e$), and exponential functions ($e^x$). It also contains routines for performing addition, complement, multiplication, and division on 64-bit binary numbers. |
| **Parity Checker/Generator** | Routine to check or generate parity for 8-bit byte. Utilizes modulo-2 counting technique. |
| **Parity Generator, ASCII Character** | Routine to add even parity bits to 7-bit ASCII character. Utilized modulo-2 counting technique. |
| **Code Conversion: ASCII to EBCDIC** | Table and routine to perform 7-bit ASCII to 8-bit EBCDIC code conversion. Full table with 128 entries provided. |
| **Delay Subroutines** | To conditionally generate a selectable time delay of:<br>1. 1 through 256 ms in one ms increments<br>2. 1 through 256 quarter seconds in one quartersecond increments<br>3. 1 through 15 minutes in one minute increments |
| **Bit Manipulation Routine** | AND, OR, XOR, COMPARE, set unselected bits, clear selected bits, test ones. |
| **Universal Logic Subroutines** | Forms logical AND, OR, XOR, $\overline{\text{XOR}}$ functions functions between the contents of Registers 0 and 1. |
| **8-bit Parity Check Annex** | Compute parity of 8-bit word without affecting any registers or carry. |
| **Binary to BCD Converter** | Converts 16-bit binary value into BCD. |
| **Data Compare** | Compares two 8-digit numbers and returns a pointer to the greater value in the carry. |
| **Paper Tape Edit** | Add, correct, or delete lines in generating a new paper tape without manually controlling tape reader. |
| **IOMEC SERIES THREE (S-3) Cartritape to Intel MCS-4** | Routines which allow full control of the IOMEC S-3 by the MCS-4. |

| TITLE | FUNCTION |
|---|---|
| I/O Test | To exercise all I/O lines to allow for troubleshooting of system design, wiring errors, and chip malfunctions. |
| Bowmar TP 3100 Printer Routine | This program is to run a Bowmar model 3100 thermal printer. |
| 8-Digit Register Dispaly | Program to display 8 digits of data. |
| Intellec 4/Mod 40 — Silent 700 Interface | Program to interface Intellec 4/Mod 40 to TI Silent 700 terminal. |
| PROM Dump Utility Program | This is a program to dump the contents of a PROM in the front panel socket onto the teletype printer. The first address and first word is always printed out in the form "00-00" as address-contents. All subsequent address contents-listings are printed out only if the contents of the respective location are different from the contents of the previous location. |
| PRO FORMA | This program assists in the compiling of source code tapes by eliminating errors and typing mistakes. In the keyboard mode it will only transmit characters to the paper tape punch that are valid in the context of the system assembly language, and automatically formats individual lines and pages to suit. |
| Peripheral Interface Routine for a Thermal Strip Printer | This subroutine controls the printing of data (numbers and selected characters) on thermal print paper using a 4x5 dot matrix thermal printhead. The software provides character generation and controls the timing of the print cycle. |
| MCS-4/40 Dissembler | To convert MCS-4 or MCS-40 machine code programs back into mnemonic or assembly code to assist in the modification of programs. |
| Right Justified HEX Data Shifter | Shifts HEX digit (four binary bits) into RAM right justified. |
| Floating Point Arithmetic Subroutine Package | Performs decimal arithmetic on 16-digit floating point operands (hexadecimal arithmetic is possible with minor changes). Numbers may range from $10^{-130}$ to $10^{125}$. Functions include Addition, Subtraction, Multiplication, Division and a normalization routine. |
| HEXBCD | Convert 2 digit HEX value to decimal range. |
| Fast Binary Multiply: Selectable Bit Precision and Constant Execute Time | The user loads the input variables to CPU registers and specifies one of five multiply precisions (12x12, 12x8, 8x8, 8x4, 4x4) via a code character in register E. |
| Fast Decimal Multiplication Routine | This subroutine computes the product (16 digit maximum) of two fixed point decimal numbers (each 8 digits maximum). |
| Automatic Digital Integration | Program will detect and integrate peaks from an amino acide analyzer and type out the area under the peak on the teletype. Program will detect saddle peaks and do simple baseline correction. |
| Multiply/Divide 8 Decimal Numbers | The multiply/divide subroutine calculates the product/quotient by repeated shifted additions or subtractions and by incrementing/decrementing the multiplier/quotient. |
| Binary to BCD Converter | Converts an 8-bit binary number to a BCD number. |

# PARTIAL PROGRAM INDEX-8-BIT USERS LIBRARY
# 8-BIT MICROCOMPUTER SOFTWARE LIBRARY

| TITLE | FUNCTION |
|---|---|
| **Save/Restore CPU State on an Interrupt** | Saves the CPU registers and flags to memory at the start of interrupt processing and restores the CPU registers and flags after the interrupt has been processed. |
| **RAM TEST PROGRAM** | Performs write and read of all zeros and ones, checkerboard test and unique address test. The RAM to be tested is successively initialized to a value and then tested. |
| **TTY Binary Load Routine** | Will load memory from a paper tape which is formatted into blocks of 256 or less binary bytes and tests each block against a checksum frame for any read errors. The tape format requires a rubout to indicate start of blocks followed by the starting page address, the starting byte address, the word count up to 256 bytes of data and a checksum in that order. A block of data may overlap pages but may not exceed 256 bytes in length. The last block of data should be followed by two consecutive rubouts to indicate end of data. Program will then branch to page 000 byte 000. |
| **TTY Binary Dump Routine** | Program will punch the contents of memory to paper tape which is formatted into blocks of 256 or less binary bytes with checksum. Tape format begins with a rubout to indicate start of block followed by the starting page address, starting byte address, word count, up to 256 bytes of data and checksum in that order. Start and end memory locations are entered from TTY keyboard. |
| **Memory Dump** | Lists memory in octal: start and stop point user definable. |
| **PROM Programmer for Intellec 8 Microcomputer Development System** | Changes programmer from fixed timing to PROM dependent timing. Program 50% more than minimum required, ensuring permanency. |
| **Data Transfer Routine** | Transfer a block of data from any location in memory to another. |
| **Move Routine** | Moves a string with a specified length to a specified location in RAM. |
| **Morse Code** | Program receives message text typed on an ASR33 teletype and sends morse code equivalent to output port 10 bit 0. It contains a 256 character buffer so that text can be typed in faster than it is sent. |
| **Binary Search Routine** | Uses a binary search method to find a character in a table of characters. |
| **Floating Point Math Package** | Package contains subroutines for Addition, Subtraction, Multiplication, Division, Negate, Absolute Value, and Test of Floating Point Numbers. |
| **Floating Point Format Conversion Package** | Provides subroutines for conversion between floating point format and ASCII or BCD. Functions are:<br>1. Floating point to BCD conversion<br>2. BCD to floating point conversion<br>3. Floating point to fixed point (integer) conversion<br>4. Fixed point to floating point conversion |
| **16-Bit Multiply** | Multiplication of two 16-bit positive numbers giving a 32-bit result. |
| **DECHL** | Macro for decrementing the 16-bit binary contents of the H and L registers. |
| **16-Bit Multiply** | Performs 16-bit X 16-bit multiplication giving a 16-bit result. |
| **SIMPY16** | 16-bit 2's complement signed multiply. |
| **LOG2A** | To calculate the Base 2 log of a number between 1 and 256. |

| TITLE | FUNCTION |
|---|---|
| **Single Precision (8-bit) Multiply/Divide Subroutine** | Five subroutines are provided:<br>1. 8-bit by 8-bit multiply for signed integers giving a 16-bit result<br>2. 8-bit by 8-bit multiply for unsigned integers giving a 16-bit result<br>3. 16-bit by 8-bit divide for signed integers giving an 8-bit result<br>4. 16-bit by 8-bit divide for unsigned integers giving an 8-bit result<br>5. 16-bit negate (2's complement) |
| **DIV16** | Performs a 16-bit X 16-bit division giving a 16-bit quotient and a 16-bit remainder. |
| **BCD To/From Binary Conversion** | Subroutines are provided for:<br>1. BCD to binary conversion<br>2. Binary to BCD conversion |
| **Binary Multiplication** | Multiplies two binary numbers:<br>1. Multiplicand: 24-bits<br>2. Multiplier: 1- to 24-bits |
| **8080 I/O System Status Display** | Display current I/O assignment information when invoked by the INTELLEC 8/MOD 80 MONITOR (Version 1.0). |
| **Binary to BCD Routine** | Converts binary value (1- to 24-bits) to its BCD value (1 to 8 digits). |
| **BCD to Binary Routine** | Converts BCD value (1 to 8 digits) to binary value (maximum 24 bits). |
| **Match Game** | A game to match a player against the processor in a test of logic. |
| **Bit Masking Subroutine** | Subroutines for changing 1 to 16 bits of a command word in RAM. |
| **High Speed List** | Permits use of a high speed printer with the Intellec 8 monitor (Version 1.0). |
| **Read and Interrupt Modification** | Allows printing of headings or operator instructions at the beginning of a Read Operation. |
| **MPY16** | Performs a 16-bit X 16-bit multiply giving a 16-bit result. |
| **Binary to ASCII Digit Converter** | Converts binary numbers to ASCII characters. |
| **Gray to Binary Conversion** | Converts up to 16 bits of cyclic Gray into binary data. |
| **I/O Simulation Macros** | I/O simulation macros for INTELLEC 8/MOD 80 systems allows simulation of input and output instruction based on an assembly time variable. |
| **Tape Duplication** | Duplicates a tape read in from the high speed tape reader by punching a copy on the TTY terminal with a leader added at both ends. |
| **CRC GEN** | Generate a 16 bit cyclic redundancy check (CRC) for a data string of up to $2^{16}$ bytes. The generator polynomial and initial conditions are defined by the user. |
| **16 Bit CRC for polynomial X16+X12+X5+1 (polynomial for SDLC)** | Produces a 16 bit CRC with 8 bit input bytes. Care should be taken with Most/Least bit feeding of the data byte and CRC residue. Does not require a table or contain any loops. Requires 24H memory locations and executes in 72.5 μsec. |
| **CRC16** | This macro calculates a CRC16 check word using the generation polynomial X**16+X**15+X**2+1. It can be used to generate a check word for a record that doesn't include one or to check that a record including a check word is correct. |
| **CRECH** | Computes CRC characters for IBM compatible floppy disk. Also works for Synchronous Data Link Control (SDLC). |

| TITLE | FUNCTION |
|---|---|
| **Legible Paper Tape** | The program punches legible characters on paper tape, useful for tape labeling. |
| **Banner Print and Punch** | Create, on a listing device or tape perforator, a graphic representation of certain ASCII characters. |
| **Large Character Paper Tape Punch Program** | The program will convert an inputted TTY ASCII character to a symbolic representation of that character on the paper tape. The program can also be called to output the ASCII character in the accumulator so the punch feature can be used by other programs such as monitors to print leaders, etc. |
| **Page Listing Program** | Provide facility for listing information in a paginated, numbered format. This is accomplished thru the system software with the console printer. |
| **Source Paper Tape to Magnetic Cassette** | Will copy a source paper tape onto a magnetic cassette. End statement must be followed by a carriage return. Program will ignore leading blanks. |
| **I Command** | This program loads HEX code into sequential RAM locations beginning at the address specified. It is useful for loading HEX machine code directly into RAM for corrections, debugging, execution, or PROM programming. |
| **8080 RAM Memory Test** | Memory test for Intellec 8/Mod 80 system |
| **Memory Diagnostic Program** | Writes test bytes in any range of memory and compares the written bit combination with what is read. Upon detection of a defective memory location, an error message is printed specifying the address, reference and actual values. |
| **Compare Object Code Tape with Memory** | This program extends the Intellec 8 system monitor's "Compare" command to check the data from a HEX format tape against the current information stored in memory. |
| **K, Program Trap** | This program provides tow traps (search/wait) for debugging other programs which use RAM memory. Displays the contents of five registers and the trap address when the trap occurs. |
| **DEBUG** | A two PROM debugging package to be used in minimum 8080 systems to inspect, dump, move and find data in memory. |
| **Punch Test or TTY Reader/Punch Test** | (1) Tests paper tape (using high speed or TTY reader),<br>(2) complete TTY reader/punch test. |
| **Reader Test** | Test high speed paper tape reader or TTY reader. |
| **TALLEY R2050 HSPTR Driver** | Extension to the Intel 8080 monitor to handle a TALLY model R2050 photoelectric tape reader at 200 cps. |
| **TALLY** | Allows Tally 2200 line printer to be used in the assembly stage of programming with Intellec 8/Mod 80. |
| **Model 101 Centronics Printer Handler** | Accepts character output for Model 101 Centronics printer from assembler or other source. Buffers print characters in RAM performing TTY compatible operations with control characters. Causes line to be printed upon receipt of line feed. Counts lines and keeps track of pagination. Inserts title at top of each page. |
| **High Speed Paper Tape Reader with Stepper Motor Control** | This circuit and program allow paper tape to be read at approximately 150 characters per second. The reader is assigned by monitor command "AR=1". The program uses electronic damping, under software control, of a stepper motor to increase stepping speed and precision. |
| **Terminal Editor** | Procedure for controlling an ASR733 Texas Instruments terminal equipped with RDC (remote control device) option. Search a line in a file contained in cassette 1 with or without copying on cassette 2. The procedure is linked to the Intellec monitor. |

| TITLE | FUNCTION |
|---|---|
| **Intellec 8/Mod 80-Silent 700 Interface** | Interface TI Silent 700 to Mod 80. |
| **Interrupt Service Routine** | Handles multiple-level interrupts, saving all registers and flags and outputting the status of the current interrupt to an external status latch. |
| **Interrupt Handler Re-entrant** | On processor receipt of an interrupt instruction (RST 0–7), this program saves the machine state and previous interrupt level on the stack, transmits the new service level to the interrupt control unit (ICU), executes a subroutine corresponding to the level interrupt received, then restores the machine and ICU to their pre-interrupt state before resuming executing the interrupted program. |
| **8008 Disassembler** | This program is used to obtain an assembly language listing of a machine coded program in memory. |
| **8080 Dis-Assembler** | This program inputs a HEX tape and generates a symbolic assembly language program suitable for modifications and/or assembling. |
| **DISASM (8080 Disassembler)** | DISASM is intended as a software development and debugging aid. Operating on resident object code, it produces an assembly language equivalent which is printed on a TTY terminal. In its present form, the program starts at a given memory address and steps sequentially through memory until manually halted. |
| **BINLB – 8080 System Loader** | Loads HEX format paper tape produced by macro assembler on GE Timesharing into 8080 system. Also provides TTY input and output subroutines. BINLD can also produce a binary dump of itself for bootstrap loading. |
| **Boot** | To allow for bootstrap loading of program and for patching of programs or data in memory via the teletype. The program uses less than 200 bytes of memory and may be placed in ROM or entered manually. |
| **Octal PROM Programming** | This program accepts sets of 3 octal numbers. The fourth character (unless it is a rubout) will cause the BYTES to be placed in memory starting at 100H. Any invalid octal character input in the first 3 positions will cause a carriage return and line feed to be output to the teletype and the line to be ignored. Any number of sets may be input (up to the practical limit of 100H). Whenever a "bell" is typed, the address of the last valid byte on Page 1 will be displayed on the register/flag lights and control will pass to the system monitor. To program the PROM, type P100, 1NN, 0 where NN is the HEX number displayed on the lights. |
| **8080 IDLE Analyzer for Approximating CPU Utilization** | Displays amount of time 8080 would have spent in an idle loop. When RUN time is compared with idle time, the percent of CPU utilization can be calculated. Time display is in memory, in ASCII. |
| **Real Time Executive** | Performs processor initialization, period and demand scheduling, routine termination, and waits during idle time. |
| **Read/write Routines for Interchange Tapes** | Subroutines read and write blocks and characters for any common audio cassette recorder. Variable redundancy allows high-speed or highly reliable operation. |
| **Proportional Power Control Image Builder** | This program builds an "ON-OFF" image in RAM to allow proportional power control using zero crossing solid state relays. |
| **Flag Processing Routine** | A routine for contact closure debounding and processing. |
| **Software Stack Routines for 8008** | Subroutines provided for PUSH, POP and EXCHange A with top of stack, and to save processor state on stack in case of an interrupt, and to restore it again. |
| **Symbol Table List Routine** | This program will print the user's symbols in alphabetical order followed by the address the 8008 Macro Assembler has assigned to each symbol. |

| TITLE | FUNCTION |
|---|---|
| **Digital to Analog Conversion for Eight Outputs** | The program processes a list of eight 16 bit values to generate eight pulse width modulated voltages which can be filtered to provide inexpensive digital to analog conversion useful for process control or other low speed requirements. |
| **Binary to HEX Routine** | To read a paper tape in binary (EBCDIC) format from the Intel high speed paper tape reader to the MCS-80 system. |
| **Binary to BCD Subroutine** | Converts unsigned binary number in D, E to 5 BCD digits. |
| **HEX to decimal conversion** | Converts any HEX number between 0 and FFFFH to the decimal equivalent. |
| **"VALUE" ASCII to HEX check and convert routine** | Using routines already available in the Intellec 8/Mod 80 monitor, "VALUE", when called will get an ASCII character from the assigned console device, check it for a valid hex digit and convert it to a four bit memory value which is returned in the ACCumulator. |
| **BCD input and direct conversion to binary routine** | Fast and efficient BCD to binary conversion code. Presented in pseudo subroutine form for implementation in ROM to allow reading of BCD input value, conversion to binary representation and branching based on loading H & L registers to PC. |
| **BINDECBIN** | Converts hex numbers input on TTY to decimal numbers and vice versa. Decimal numbers must be ended with D, hexadecimals with H. Conversion begins with space. If first character input is CR, control is given back to monitor. Largest number handled is two bytes binary. |
| **MATH** | Includes routines for fixed and floating point arithmetic together with a demonstration program that performs algebraic evaluation and allows unlimited parentheses nesting. An expression within parentheses can be evaluated and displayed by "=" and is preserved as a subtotal, etc. |
| **Elementary Function Package** | Calculates floating point:<br>square root<br>logarithm<br>exponential function<br>sine<br>cosine<br>arc tangent<br>hyperbolic sine<br>hyperbolic cosine |
| **8080 Foating Point Package with BCD Conversion Routine** | Performs floating addition, subtraction, multiplication, division, fixing, floating, negation, and conversion from floating point to BCD with exponent. |
| **8080 Least Squares Quadratic Fitting Routine** | Performs summations and matrix manipulation for fitting up to 256 floating point X-Y pairs to a function of the form: $aX^2 + bX + c = Y$ |
| **Floating point procedures** | Dummy PL/M interface procedures. |
| **PL/M floating point interface** | Interfaces PL/M conventions with floating point assembler format. |
| **Floating point decimal & HEX format conversion** | The program converts a number of 27 characters maximum to standard 13 digit decimal format and to floating point accumulator form in HEX format on the teletype. |
| **N-byte Binary Multiplication and leading zero blanking** | The program performs binary multiplication on two numbers and returns a result that may be up to 255 bytes in length. |
| **Subroutine DEMULT** | To multiply M decimal digits by N decimal digits and store the product (7 digits x 3 digits as written, but easy to expand as required.) |

| TITLE | FUNCTION |
|---|---|
| **BCD Multiplication** | Multiplies up to a 6 digit BCD number by a 4 digit BCD number, providing a 10 digit BCD result. All numbers are unsigned. |
| **MUL/DIV multi-precision pack for 8080** | Signed fixed-point binary fraction multiply and divide. Double-precision inputs, double-precision output for divide and 4-byte output for multiply. |
| **Double precision multiply** | To multiply two 16-bit numbers, returning the most significant 16 bits (in address form) thru the appropriate registers to the calling program. |
| **16 Bit Square Root Routine** | Return 16 bit square root (8 bit whole number joined with 8 bit fraction) of a 16 bit argument. The result conforms to standard signed number convention; therefore, its highest order bit will always be zero. The argument must have zeros in its two highest order bits, for its square root to lie in the valid range of the signed result. |
| **Floating Point Square Root** | Math & numerical Manipulation Programs. Operations performed are:<br>test for negative argument (overflow set and return)<br>computation of the square root for positive arguments |
| **SQRTF** | Generates 8 bit root of 16 bit number. |
| **Subroutine SQRT** | This subroutine takes the square root of a number in floating point notation. |
| **Fast Floating Point Square Root Routine** | Calculate square root of a floating point number by Heron's method. Execution time less than 50 ms for any number. |
| **Natural logarithm** | Computes the natural logarithm of a number between 1 and 65535. |
| **Subroutine LOG** | This subroutine takes the log to any integer base of any positive floating point number. |
| **Approximating Routine** | To solve functions such as the log, the antilog, the sine and the tangent function. The program given is set up to solve the antilog (base 2) function. |
| **Sin X, Cos X Subroutines** | Generates sine or cosine accurate to 8 bits of an input angle that is accurate to 8 bits. Uses a Chebyshev Economization of Taylor Series for Cosine X. Sine X is generated by complementing the angle X with respect to $90^{\circ}$ ($x' = 90^{\circ} - X$) and then taking Cosine X. |
| **RMSTF** | To calculate the integration "T" of any continuous function "F(x)" between two limits "a" and "b". |
| **Binary search** | Program searches a table of up to 128 entries. Each entry is composed of a 1 byte argument (Search Key) and an associated result. The result field may be up to 255 bytes for each argument. Result fields must all be the same length. |
| **Random Number Generator Subroutine** | Subroutine to generate a random number between 0 and $177_8$ ($125^{10}$). |
| **8-bit Pseudo Random Number Generator** | The program reads data from page 0, address FF and generates a random number. The new number is written back in the same location. All numbers except zero are generated. Zero is a disallowed state and is corrected in the program. |
| **16-bit Random Number Generator** | The subroutine implements a linear congruential sequence which generates 16-bit random numbers. The random numbers produced range from 0000 to FFFF with a period less than or equivalent to 2 ** 16. An 8-bit random number is available as the upper byte of the 16-bit random number.<br>X(N+1) = (2053*X(N) + 13849)mod 2**16 |

| TITLE | FUNCTION |
|---|---|
| **PL/M Histogram Procedure and Random Number Generator** | Main program generates an 8-bit shift register sequence by XORing the first and last bits and shifting the result into the next random numbers bottom bit. 1000 numbers are generated and then histogramed. Histogram procedure sets up an output histogram array and then prints the histogram on the TTY when commanded after printing. The array is not zeroed so that intermediate results may be displayed without effect on the final histogram. |
| **RANDOM$BITS** | A Non-multiplicative pseudo-random number generator. |
| **Clock Subroutine** | Maintains a current time of day, decimal adjusted in BCD, of hours, minutes, and seconds. Must be invoked once each second, usually by an external interrupt. Time is stored in three bytes of memory, in the 24-hour system or optically in the 12-hour system. |
| **Interrupt Driven Clock Routine** | Updating of clock located in RAM based on 100 ms time intervals. Pulses arriving on interrupt line. Four storage locations reserved for 100 ms counter, secs counter, mins counter and hour counter. One location for interval counter one for preset interval and one for flag indicating interval has elapsed. Updating of clock takes about 70 microsecs. |
| **Calendar Subroutine** | Uses three bytes of RAM to store the current date arranged as two BCD digits per byte. The date is adjusted for months with 28, 29, 30 or 31 days and February is adjusted for leap years 1976, 1980 and 1984. |
| **PASS** | Program PASS transfers addresses of parameters between a calling program and subroutine. |
| **Address** | Subroutines for dynamic memory allocation and addressing. |
| **Data Array Move** | A contiguous array of data may be relocated in memory, regardless of the magnitude and direction of the move. The source and destination array locations may overlap. The maximum array size is $2^{16}$ bytes. |
| **Shellsort** | Sorts arrays in place using Shell's method of diminishing increment. |
| **Text Storage Program** | Allows text to be stored in memory using a letter of the alphabet as a pointer. After the message is stored, it can be retrieved by depressing a single key on the TTY. Up to 32 messages may be stored and retrieved independently. |
| **Time Sharing Communications** | To communicate with medium to large scale computer systems as an external time-share user. |
| **A Generalized Stepper Motor Driver Program** | Operations performed by the program are: using entry variables of number of steps, clockwise or counterclockwise direction and speed of steps — several programs are illustrated for moving a stepper motor in either direction then stopping, moving N steps forward then return N steps, moving motor continuously in either direction until interrupted by a TTY keyboard entry, also programs using an LED-photodetector sensor for absolute motor position. |

# 8-BIT

## CROSS PRODUCTS

Cross Assembler for HP2100
Cross Assembler for PDP-11
Macro Assembler for PDP-11
Absolute Loader for PDP-11 LDA Tape Format
8008 Macro Definition Set for Assembly on PDP-11
8080 Macro Definition Set for Assembly on PDP-11
8080 Cross Compiler on PDP-11
Cross Assembler for PDP-11
Cross Assembler for Nova 1200
Cross Assembler for Nova 1220, IBM 360/40 and CDC3300
Nova Cross Assembler for Intel 8080
Cross Assembler for Nova
Intel 8008 Cross Inverse Assembler for HP 2100
8080 Cross Assembler for HP 2100 DOS
8008 Cross Assembler for HP 3000
8080 Cross Assembler for HP 3000
PL/M 80 Pass 3

## GAMES

NIM
NIM
Blackjack
The Word Game
Gambol
Mastermind
Maze
Game of Life
Numbers
Kalah
An Adaptive Game Program
Match Game

# MICROCOMPUTER USER'S LIBRARY SUBMITTAL FORM

☐ 4004 ☐ 4040 ☐ 8008 ☐ 8080 ☐ 3000 (use additional sheets if necessary)

**Program Title**

**Function**

**Required Hardware**

**Required Software**

**Input Parameters**

**Output Results**

| | |
|---|---|
| **Registers Modified:** | **Assembler/Compiler Used:** |
| **RAM Required:** | **Programmer:** |
| **ROM Required:** | **Company:** |
| **Maximum Subroutine Nesting Level:** | **Address:** |

# INSTRUCTIONS FOR PROGRAM SUBMITTAL TO MCS USER'S LIBRARY

1. Complete Submittal Form as follows: (Please print or type)
   a. Processor (check appropriate box)
   b. Program title: Name or brief description of program function
   c. Function: Detailed description of operations performed by the program
   d. Required hardware:
      For example: TTY on port 0 and 1
      Interrupt circuitry
      I/O Interface
      Machine line and configuration for cross products
   e. Required software:
      For example: TTY routine
      Floating point package
      Support software required for cross products
   f. Input parameters: Description of register values, memory areas or values accepted from input ports
   g. Output results: Values to be expected in registers, memory areas or on output ports
   h. Program details (for resident products only)
      1. Registers modified
      2. RAM required (bytes)
      3. ROM required (bytes)
      4. Maximum subroutine nesting level
   i. Assembler/Compiler Used:
      For example: PL/M
      Intellec 8 Macro Assembler
      IBM 370 Fortran IV
   j. Programmer, company and address

2. A source listing of the program must be included. This should be the output listing of a compile or assembly, Extra information such as symbol table or code dumps is not necessary.

3. A test program which assures the validity of the contributed program must be included. This is for the user's verification after he has transcribed and assembled the program in question.

4. A source paper tape of the contributed program is required. This insures that a clear, original copy of the program is available to photo-copy for publication in a User's Library update publication.

---

Send completed documentation to:

**Intel Corporation**
User's Library
Microcomputer Systems
3065 Bowers Avenue
Santa Clara, California 95051

intel

# MICROCOMPUTER TRAINING
# MICROCOMPUTER WORKSHOPS

Microcomputers are being used in hundreds of applications from simple controllers to complex data processing systems. To enable users to bring microcomputers into their applications, Intel is offering a selection of 3 and 4 day workshops that are designed to provide you with the "tools" for making optimum use of Intel microcomputers in system development.

PREREQUISITES: To attain benefit from course presentation, some background in logic design and a basic knowledge of programming is necessary.

ATTENDANCE: Attendance is limited to (15) enrollees.

TUITION: The fee for each workshop is $350, which includes course materials, computer time, and luncheons.

SCHEDULE: These workshops are scheduled to be held at Intel Corporation Training Centers in Santa Clara, CA and Boston, MA.

REGISTRATION: Contact your Intel Sales Office for details.

# MCS-80/ICE-80 WORKSHOP

COURSE OBJECTIVE: This workshop will prepare the student to design and develop a system using the Intel® 8080 microprocessor through the use of lecture, demonstration, and laboratory "hands-on" experience with the Intellec MDS development system and ICE-80.

**COURSE OUTLINE:**

**DAY 1**

**Introduction**
- a. Microprocessor System Fundamentals Function/Organization/ Programming
- b. Introduction to the 8080
  1. Basic System
     - a) CPU
     - b) Memory
     - c) Input/Output: (Programmed/Interrupt/DMA)
  2. Programming Model
- c. Languages
  1. Machine Code
  2. Assembly Language
  3. PL/M

**Instructions**
- a. Input/Output
- b. Register/Memory Reference
- c. Control/Arithmetic

**The Intellec MDS**
- a. Function/Operation
- b. System Monitor

**The Text Editor**
- a. Structure/Commands
- b. Operation

**Laboratory**
- a. Using the System Monitor
- b. Using the Text Editor

**DAY 2**

**The Macro Assembler**
- a. Syntax/Pseudo-Instructions
- b. Operation

**System Timing**
- a. Instruction
- b. Clock
- c. Address/Data

**Input/Output Programming**

**Subroutines**
- a. Invocation
- b. Stack Memory
- c. Parameter Passing

**Teletype Programming Requirements**

**The Interrupt System**
- a. Definition
- b. RST Instruction
- c. Service Subroutines

**Laboratory**
- a. Using the Assembler
- b. Program Assembly and Execution

**DAY 3**

**Branch Table**
- a. Application
- b. Construction

**Direct Load/Store Instructions**

**System Monitor Debugging Function**

**System Design Process**
- a. Memory and I/O Requirements
- b. Bus Control
- c. Clocks
- d. Hardware/Software Trade-offs

**Additional Development Aids**

**Laboratory**
- a. Program Assembly and Execution
- b. PROM Programming

**DAY 4**

**What is "In Circuit Emulation"**

**ICE-80 Description**
- a. Functional Block Diagram
- b. Theory of Operation

**ICE-80 Laboratory**
- a. Operating the ICE-80
- b. Emulating an 8080 Based System

# PL/M LANGUAGE WORKSHOP

COURSE OBJECTIVE: This workshop will prepare the student for writing and debugging PL/M programs using lecture, demonstration, and laboratory "hands-on" experience in operating PL/M interactively from a high-speed, time-shared computer terminal.

**COURSE OUTLINE:**

## DAY 1

**Introduction**
a. Overview of PL/M
b. Preview of Course

**Definitions**
a. Symbols
b. Identifiers
c. Reserved Words
d. Comments
e. Data Elements
f. Expressions
g. Statements
h. Declarations

**Data Elements**
a. Variables
b. Subscripted Variables
c. Data Type
d. Constants

**Operators, Operations and Priorities**
a. Arithmetic
b. Boolean

**Evaluating Expressions Statements**
a. Redefine
b. Basic
c. Conditional

**Blocks**
a. Concept and Use
b. Scope of Declarations
   1. Global and Local
   2. Nested and Parallel Blocks

**Laboratory**
a. Introduction to Data Terminal and Timesharing
b. Compiling a PL/M Program

**Assignment**
Sample Problem to Flowchart and Program Outside Class

## DAY 2

**Review**
**Procedures**
a. Declaration
b. Invocation
c. Program Construction

**Data References**
**Statement Labels**
**Unconditional Transfers**
**Compile-Time Macro Processing**
**Input/Output Processing**
**Simulating an 8080 System**
**Laboratory**
a. Compile Programs
b. Execute Programs

## DAY 3

**Review**
**Memory Mapping**
**Assembly Language Interface**
**Interrupt Processing**
**Predeclared Variables and Procedures**
a. LENGTH and LAST
b. Condition Code
c. Memory Vector
d. TIME Procedure
e. Type Transfer
f. Decimal Arithmetic
g. Shifts and Rotates

**Laboratory**
a. Compile Programs
b. Execute Programs

# SERIES 3000 WORKSHOP

COURSE OBJECTIVE: This workshop will provide the student with an in-depth understanding of the Series 3000 family through the use of lecture and demonstration. Microprogramming and design examples are presented.

**COURSE OUTLINE:**

**DAY 1**

**Introduction**
a. Introduction to Microprogramming
b. The Series 3000 Component Family
c. Series 3000 System Overview

**CPU Design Example**
a. CPU System Requirements
b. Architecture of a CPU
c. Developing a Macro-instruction Set
d. Interrupt Handling
e. Microprogram Mapping

**DAY 2**

**Design Techniques**
a. Conditional Clocking
b. K-Bus
c. Micro-instruction Field Extension
d. Micro-subroutines
e. Pipelining
f. Timing Analysis

**Controller Design Example**
a. Disc Controller System Requirements
b. Architecture of a Disc Controller
c. Microprogram Implementation

**DAY 3**

**Development Support**
a. Introduction to CROMIS, the Series 3000 Cross Micro-Assembler
b. MDS-800 Microcomputer Development System
c. ICE-30 In-Circuit Emulator
d. ROM Simulator
e. Demonstration

# MCS-4/40 WORKSHOP

COURSE OBJECTIVE: This workshop will prepare the student to design and develop systems using the Intel® 4040 and 4004 through the use of lecture, demonstration, and laboratory "hands-on" experience with the Intellec 4 MOD 40 development system.

**COURSE OUTLINE:**

**DAY 1**

**Introduction**
a. Basic Microcomputer Block Diagram
   1. Function
   2. Uses
b. MCS-40 Block Diagram
   1. CPU – 4004/4040
   2. Memory – 4001, 4002, 4308, 4289
   3. I/O – 4207, 4209, 4211, 4003

**Basic System Timing**
**Description of Major Elements of CPU**
**MCS-40 Instructions**
a. Basic Machine Instructions
b. Accumulator Group Instructions
c. I/O and RAM Instructions
d. 4040 Group Instructions

**MCS-40 Assembler**
a. Syntax
b. I/O Formats
c. Coding Examples

**Laboratory – Intellec 40 Operation**
a. Control Console Use
b. TTY Input
c. High-Speed Reader Operation

**Homework Utilizing Assembler Language to Code Sample Programs**

**DAY 2**

**Review Sample Programs**
**The Interrupt System**
a. Definition
b. Instructions
c. Service Subroutines

**System Monitor Description**
**System Development Aids**
a. Intellec 4 MOD 40
b. Cross Assemblers
c. User's Library

**System Interrelation**
a. Connections
   1. Hardware
   2. Software
b. ROM/RAM Configurations
c. Interface Design
d. MCS-40 Family Components

**Sample System Design**
**Laboratory**
a. Using the Assembler
b. Debug Using System Monitor and Console

**DAY 3**

**Review System Design**
**Hardware/Software Trade-offs**
**Laboratory**
a. Hands-On Programming Time
b. PROM Programming

**Summary and Course Review**

# 11

# TIMEKEEPING CIRCUITS