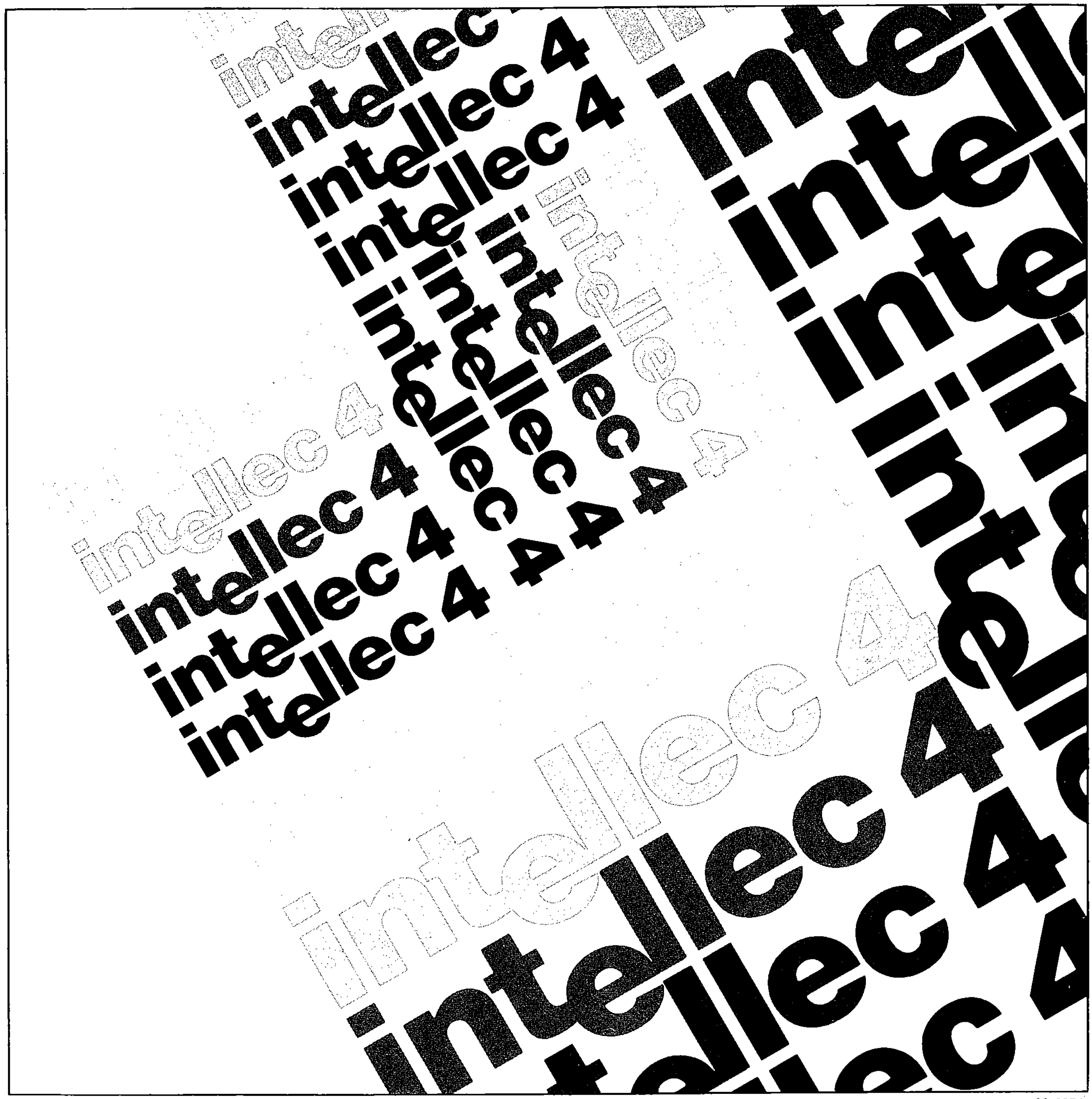
intel®

intellec® 4/ intellec Mod 40 Microcomputer Development System

Reference Manual



98-095A

intel®

intellec® 4/ Mod 40 Microcomputer Development System

Reference Manual

CAUTION
Do not operate the INTELLEC® 4/Mod 40 system with the cover removed. The resulting diversion of cooling air may cause overheating and damage to the internal power supplies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

SCOPE OF THIS MANUAL

The INTELLEC® 4/MOD 40 Reference Manual is part of a three-volume set which describes the INTELLEC® 4/MOD 40 microcomputer development system. The INTELLEC® 4 Operator's Manual tells how to operate the system, and describes the use of its standard software provisions. The 4004/4040 Assembly Language Programming Manual explains how to program the system using the special INTELLEC® 4/MOD 40 assembly language mnemonics.

This manual describes the capabilities of the INTELLEC® 4/MOD 40 system, with special reference to the design concepts that underlie them. It covers the entire system, and it covers the system's individual modular components. It describes the theory of operation of each module both at the functional level and at the circuit level. And it provides guidelines for those who intend to use selected modules in other microcomputer systems.

All the information necessary to install, use and fix the system is included here. But the book is not a maintenance and repair manual. It is principally a guide for those concerned with the system's hardware aspects. Because the INTELLEC® 4 is basically a design aid, its users will include system designers, who use the INTELLEC® system in the modeling and development of their own specialized microcomputer systems. Others who will refer to this book are designers who use individual INTELLEC® modules in other system applications. It is reasonable to assume that users in these categories will not be disadvantaged by the lack of step-by-step procedures.

This manual contains twelve chapters. Chapter 1, which follows this introduction, describes the functional operation of the INTELLEC® 4/MOD 40, at the system level. Chapters 2 through 11 describe the operation and utilization of each of the individual modules. Chapter 12 discusses installation and interfacing.

Appendix A at the rear of this manual contains a summary of the instruction set used to program the INTELLEC® 4/MOD 40 system. Appendix C summarizes activity on the system's main data bus, during the execution of each program instruction.

This manual also contains a number of schematic diagrams, to facilitate the discussion of individual INTELLEC modules. These drawings are provided for convenient reference, but may not reflect the latest engineering changes. The drawing package furnished with the instrument itself is the final authority for such changes.

GENERAL DESCRIPTION OF THE INTELLEC® 4 SYSTEM

Capabilities

The INTELLEC® 4/MOD 40 is a general purpose, stored program, digital computer, which uses components of Intel's MCS-40™ microcomputer set. Its general purpose configuration permits it to simulate dedicated, stored program, digital computers, which use components of the MCS-40 microcomputer set. It is thus a design and development aid.

The heart of the MCS-40 set is the Intel 4040 Central Processing Unit, a monolithic MOS chip that performs all the functions normally associated with the arithmetic and control units of larger machines. This chip functions as a four-bit parallel processor, in conjunction with several specialized memory and I/O components.

Two kinds of memory components are used in MCS-40 microcomputers. We distinguish between the program memory, or that used for the storage of instructions, and read/write memory, that used for the storage of intermediate results. This distinction may seem strange at first to those accustomed to working with large-scale equipment.

Processors that work under the supervision of an operator generally use the same memory bank, both for

the storage of the program and for the storage of results. The processor may write anywhere in memory, with indifference, or even alter its own instructions. Microprocessors, on the other hand, often work in control applications, with no supervision whatever. Since there is no one present to correct obvious blunders, their programs must be foolproof. Program memory must be invariable under all circumstances; that is, firm.

But processors that handle any appreciable volume of data also need memory for the storage of intermediates. The processor's internal registers simply cannot cope with any realistic flow of data, without some help. Thus, a true read/write memory is also a necessity.

It should be obvious that these two requirements are mutually inconsistent. Memory which is firm cannot receive data from the processor, and read/write memory cannot be invariably constant. To meet these differing needs, two kinds of memory are included in the basic MCS-40™ set.

The 4001 256 X 8 Bit Read Only Memory and Four-Bit Input/Output Port is the basic program memory element in systems designed around the MCS-40™ set. It stores eight-bit instructions and is programmed in manufacture, to user specifications. The specifications include not only the data contents of the memory, but also the use of each of the four I/O lines (input or output). Mask programming in manufacture further establishes the chip's addressable identity, as one element in a bank containing as many as sixteen 4001s. Program memory requirements can also be satisfied by 4308 Read Only Memory elements. The 4308 element is functionally identical to four 4001 chips. It provides 1024 X 8 bits of program storage and sixteen I/O lines (four ports of four lines each). Like the 4001, the 4308 is mask programmed in manufacture, to user specifications.

The 4002 320 X 4 Bit Random Access Memory and Four-Bit Output Port is the basic read/write memory element in the MCS-40. It is a true working memory, organized to store four-bit words, and to retrieve them upon command from the processor. Like the 4001, the 4002 is a dual purpose component designed to perform both memory and I/O functions. Its addressable four-line port, however, is used exclusively for output. The chip is mask-programmed, to specify its identity, but this programming is much less comprehensive than that which the 4001 receives. Only two variations exist. The two kinds of RAM elements are designated 4002-1 or 4002-2, to distinguish them from one another. Chapter 2 of this manual explains in detail how the processor identifies and addgresses these memories.

The 4001, 4308 and the 4002 all contain a good deal of control logic, which enables them to perform their respective functions with a minimum of direct processor control. They are thus economical and versatile components, when used for the mass production of OEM microcomputers. The production virtues of the ROM, however, make it something of a liability during the development phases of a project. Program memory, by virtue of its design, cannot be read by its own processor for purposes of output. This makes it difficult to "dump" the contents, for examination by the programmer. Moreover, the re-programming of a segment committed to ROM involves re-ordering of elements from the factory, and that often entails unacceptable delays. To circumvent these difficulties, and to facilitate the development of OEM systems, the INTELLEC®4 was introduced. The next section describes the special features of the INTELLEC®4 system that enable it to serve as a development aid.

In addition to the 4040 processor, which recognizes and responds to 60 program instructions, the system contains 1K X 8 bits of basic program memory and 320 X 4 bits of working storage. It also contains a special 4K X 8 RAM memory, described in the next section, and provision for up to 4K X 8 bits of programmable read-only memory (PROM), a special purpose program memory. Three 4-line input ports and eight 4-line output ports are provided. The standard system is expandable to accommodate up to 2560 X 4 bits of read/write memory, 16 input ports, and 48 output ports. These figures, however, reflect only the limits inherent in the system's basic addressing structure. If the user is willing to provide a modest amount of additional decoding logic, even greater expansion is feasible. The flexible, general purpose design of the INTELLEC® 4 system thus permits bench simulation of almost any conceivable OEM configuration.

Special Features

The INTELLEC®4/MOD 40 contains provisions that simplify and expedite the development process.

The 1K X 8 program memory of the standard system is occupied by the firmware instructions of the INTELLEC® 4/MOD 40 System Monitor, a general-purpose program designed expressly to simplify and speed the process of developing specialized, application-oriented programs. Additional memory must therefore be provided in which to store the instructions of the program actually being developed and tested. For that purpose, the system's 1K X 8 program memory is augmented by a 4K X 8 memory of special design, based on the use of 2102 RAM memory components and known as program RAM. (Program RAM is used solely for program storage and should not be confused functionally with the 4002 data RAM memory described previously.) This memory is arranged to simulate the operation of 4001 program memory, but with an important exception. Special instruction sequences permit the operator to read and alter the contents of program RAM, using the standard system software.

Moreover, the INTELLEC® 4/MOD 40 system incorporates a Control and Display Panel which not only performs the routine functions of power control and RESET, but also monitors the contents of any selected location in program RAM. The operator can alter the contents of that location simply and quickly, using switches on the panel. This mode of operation is known as console memory access (CMA).

Still another feature of the panel is the search and display function. In the search mode, the operator presets the panel controls to a given address in program memory. He also presets a specific number of passes of that location, as desired. The console display will latch and hold the information on the internal main data bus during the entire cycle in which the specified number of passes of the specified address occurs. This information, including the program address, the instruction stored there, and any data in or out of the processor, can be inspected by the operator at leisure.

In addition, the INTELLEC® 4/MOD 40 system contains a built-in teletype interface, enabling direct communication through a Model ASR 33 teletype set. With the minor modifications described in Chapter 12 the teletype set can function as operator's console, paper tape reader, paper tape punch, and listing device. A convenient point of interface is thereby established for the testing and debugging of programs.

System Software

The foregoing provisions furnish very powerful and effective programming, debugging, and development aids. They are supported by a versatile software package, included with the standard system.

The System Monitor is a firmware program which is built right into the system's Central Processor Module. It is selectable by means of a switch on the Control and Display Panel. It contains a teletype I/O routine, and service routines which permit:

a) Loading program RAM from paper tape

b) Dumping the contents of program RAM

c) Modification of individual RAM instructions

d) Writing program RAM onto paper tape

e) Programming of Programmable Read Only Memory (PROM)

Refer to the INTELLEC® 4/MOD 40 Operator's Manual for a full description of the System Monitor and its capabilities.

The software package includes an Assembler, furnished on perforated tape, which permits the rapid and convenient assembly of object level programs. The INTELLEC® 4 Assembler is a three-pass assembler. That is, the user's source program must be read three times in order to produce an object program tape on a listing of the assembly. During the first pass, the assembler collects information from the source program and builds internal tables, but produces no output. During the second pass, a listing of the assembled program including any errors detected is printed. During the third pass, an object program tape in the form of a hexadecimal file readable by the Monitor is produced on the Teletype paper tape punch. Once the first pass has been completed, the second and third passes may be done in any order, or either may be omitted. Use of the Assembler consists of the following steps:

1) The teletype console, in the off-line mode, is used to prepare a paper tape containing the entire assembly language listing of the program.

2) With the teletype (or high-speed paper tape reader) on-line, and with the System Monitor selected, the tape containing the Assembler program is run through the teletype's (or the high-speed) tape reader. This loads the Assembler into program RAM.

3) The program RAM memory is selected, using the console RAM switch.

4) The tape containing the assembly language listing is run through the tape reader (teletype or high-speed) for the first pass.

5) The assembly language listing is run through the reader again. The user can specify pass 2, which will produce a listing of the assembled program, or can specify pass 3, which will produce an object program tape. Once the first pass is completed, the second and third passes may be done in any order, or either may be omitted.

6) If pass 3 has been executed (i.e., if an object tape has been punched), the System Monitor is again selected, using the console switch, and the object tape is run through the reader to be loaded into program RAM.

7) Program RAM is again selected, and execution may begin.

This procedure eliminates entirely the laborious and time-consuming process of translating instructions into binary machine code by hand. Because it is much easier for a programmer to remember and use mnemonic words like HLT (for "halt") than it is to remember and write the equivalent machine code (00000001) a great deal of time and expense are saved in the course of programming the system.

Programmable Read Only Memory

The INTELLEC® 4/MOD 40 system has provision for another kind of memory, intermediate between program RAM and 4001 or 4308 ROM. This is the 4702A Programmable Read Only Memory, known as PROM.

The 4702A is a 256 X 8 bit read only memory, but one considerably different from the 4001 ROM. The most significant difference is that the 4702A may be programmed by the user, with higher-than-usual current pulses from an electronically controlled power supply. Moreover, the PROM is erasable, by deliberate exposure to a threshold ultraviolet level, permitting it to be erased and reprogrammed as often as desired.

The 4702A PROM thus has obvious advantages over the 4001 or 4308 ROM, as a vehicle of program development. This suggests the possibility of a substitution, using the

4702A to simulate the 4001 or 4308. Unfortunately, the PROM has neither the internal control logic nor the I/O provisions contained in the 4001 and 4308 ROM.

A special logic element, the 4289 Standard Memory Interface, permits the simulation of ROM, using PROM. This element is incorporated into the Central Processor Module of the INTELLEC® 4/MOD 40 system.

The standard system has no user-programmed PROMs. Four factory-programmed PROMs, however, are used to contain the System Monitor. These are mounted in sockets on the Central Processor Module. Under certain circumstances, the user may wish to replace them with PROMs of his own choice. That is perfectly feasible.

Optional modules are available, however, to implement an extended PROM memory capacity. Up to 4K X 8 bits may be accommodated, with no modifications required. The modules simply plug in, and the PROM program memory is switch-selected at the control panel.

A PROM Programmer Module is also provided, giving the user complete PROM programming capability. The System Monitor contains a routine which permits the automatic transfer of information from program RAM to non-volatile, firmware storage in PROM. Thus, still another dimension of flexibility is added to the INTELLEC® 4/MOD 40 system.

Developing An OEM System

Given the convenience features built into the INTELLEC® 4/MOD 40 system; it is easy to visualize a straightforward, fast, and inexpensive procedure for developing specialized MCS-40™ microcomputer systems. Steps in development will include:

a) the design and construction of interfaces;

b) the writing of an assembly language program;

c) assembly of the program, using system software;

d) preliminary testing and de-bugging of the system, using system software;

e) commission of the program to semi-permanent storage in PROM, using the system software;

f) final checkout of the system;

g) transfer of the program to paper tape, automatically formatted by the system software for 4001 or 4308 ROMs from Intel.

With the advent of the INTELLEC® 4 system, the most formidable obstacles of OEM microcomputer development have been removed. Once the program is firm, it becomes a relatively routine matter of preparing the printed circuit artwork and ordering components for the production of a tailor-made microcomputer system.

The Physical System

The hardware of the INTELLEC® 4/MOD 40 system is based on a mainframe assembly which consists of a chassis, a mother board and power supplies furnishing +5 Volts, −10 Volts and +80 Volts DC (for PROM programming). The standard system includes the Control and Display Panel, and four plug-in modules:

a) the Central Processor Module;

b) the Control Module;

c) the RAM Memory Module;

d) the PROM Programmer Module.

The Central Processor Module is essentially a self-contained computer on a card, with processor, memory, and I/O provisions included. The RAM Memory Module contains the program RAM. The Control Module interfaces between the two. The PROM Programmer Module enables programming of 4702 PROMs.

The standard INTELLEC® 4/MOD 40 system includes a furnished cabinet enclosure, with cooling fan.

Expansion of the system's basic capability is accomplished through the addition of plug-in modules. The available modules include:

a) the Data Storage Module;

b) the Input/Output Module;

c) the PROM Memory Module;

d) the Instruction/Data Storage Module;

e) the Universal Prototype Module;

f) the Module Extender;

g) drawer slides and extenders for rack mounting.

MODULAR COMPONENTS OF THE INTELLEC® 4/MOD 40 SYSTEM

The capabilities of the individual system modules are summarized below. Refer to the appropriate section in this manual for a detailed description of each module.

With the exception of the mainframe and the Control and Display Panel, all modules are implemented on 6.18 X 8.0 inch printed circuit cards. All are designed to mate with standard 100-pin, double-sided PC edge connectors, like those used in the universal socket of the INTELLEC®'s mother board.

The standard system is built up from a mainframe consisting of chassis, a mother board, and two OEM power supplies. The supplies provide DC power, at levels of +5 VDC, −10 VDC, and +80 VDC. The mainframe assembly accommodates the following system components:

imm4-43 Central Processor Module

The imm4-43 is a complete microcomputer system, with the processor, program memory, and I/O all included on the card.

The 4040 Processor is the workhorse of the module. This microprocessor handles 60 different program instruc-

tions, incorporates 16 working registers, and provides for seven level nesting of subroutines.

The processor is augmented by sockets which accommodate 1K X 8 bits of PROM program memory and by 320 X 4 bits of 4002 RAM storage.

Three input ports and eight output ports are provided on the Central Processor Module. Four of the output ports are those associated with the 4002 RAM elements. The remaining three input and four output ports are implemented in random logic. They represent the I/O ports that would normally be provided by the 4001 ROM program memory chips, in a production version of the system. A teletype interface is included on the module.

A crystal-controlled clock and a 4289 memory interface complete the module's provisions.

Memory Control Module

The Control Module contains the circuitry required to interface the Central Processor Module to the RAM Memory Module and to the Control and Display Panel. The module also contains the RESET, STOP, SINGLE-STEP, and memory selection logic.

imm6-28 RAM Memory Module

The RAM Memory Module is a 4K X 8 memory system, which employs the Intel 2102 1024 X 1 bit Static Random Access Memory element. Address latching, data latching, and module selection decoding are included on the card. The imm6-28 is often referred to as the program RAM.

Control and Display Panel

The Control and Display Panel permits operator control of the INTELLEC® 4/MOD 40 system and displays the machine's internal status on the console indicators. Power control, RESET, STOP, SINGLE-STEP, and memory selection functions are performed by means of switches on the console.

In the search mode, the console permits the selective examination of any program step. The CMA mode provides the operator with the ability to alter the contents of program RAM, using switches on the console. The panel thus provides complete program development and de-bugging facilities.

A socket for the programming of PROMs is installed on the panel.

imm4-22 Instruction/Data Storage Module

This module duplicates the memory and I/O capacity of the Central Processor Module itself. It contains 320 X 4 bits of data storage, sockets for 1K X 8 bits of PROM program storage, four input ports, and eight output ports. It expands the capabilities of the basic system.

NOTE: While this module is compatible with the INTELLEC® system, it is usually not used for INTELLEC® system expansion; other modules, such as the I/O and PROM modules, provide greater expansion capabilities.

imm4-24 Data Storage Module

This module has space for sixteen 4002 RAM memory elements (four banks). Four 4002's are provided with the card, giving the module an initial capacity of 320 X 4 bits of working storage. A maximum INTELLEC®4/MOD 40 system may contain two imm4-24 modules, providing as many as 2560 X 4 bits of read/write storage (32 RAMs). The decoding logic for this expansion is provided with the modules. Each RAM used incorporates a 4-bit output port, with the result that as many as 32 additional output ports are provided. This module expands the I/O capabilities of the basic system.

imm4-60 Input/Output Module

This module provides expansion of the INTELLEC® 4/MOD 40's basic I/O capacity, where no corresponding expansion of the memory is desired. Eight 4-bit input ports, and eight 4-bit output ports are provided.

imm6-26 PROM Memory Module

This module provides sockets for as many as sixteen 4702A PROMs. Address decoding logic is included on the card.

imm6-76 PROM Programmer Module

This module is an electronically controlled, pulsed power supply, with level shifting and data complementing circuitry included. It is equipped to program 4702A PROMs under the supervision of the INTELLEC® 4/MOD 40 System Monitor. Programming is entirely automatic. The procedure is described fully in the INTELLEC® 4/MOD 40 Operator's Manual.

imm6-70 Universal Prototype Module

This is a blank module which accommodates wire-wrap sockets in 14, 16, 24, or 40-pin sizes. A maximum of 52 sixteen-pin sockets may be used. The module provides full breadboarding flexibility, for the development of custom circuits and specialized interfaces. Installation is simplified by the 100-pin edge connector, which mates directly with any vacant connector on the INTELLEC® 4's "universal socket".

imm6-72 Module Extender

This module is a simple PC board extender, providing access to points on the individual modules, for trouble-shooting and for system de-bugging.

imm4-90 HIGH-SPEED PAPER TAPE READER

At the customer's request, an optional paper tape reader will be supplied. The optional unit contains an optical character reader and a high-speed incremental drive. These enable it to transfer data at a minimum asynchronous rate of 200 characters per second, 20 times faster than the teletype's paper tape unit.

Using the ASR 33's tape reader, the System Monitor takes about 200 seconds to load 1K (1024) locations in program RAM. It therefore takes nearly 14 minutes to load completely the 4K locations in program memory. The High-Speed Paper Tape Reader, however, only requires about 40 seconds to load the entire 4K.

The convenience and economy of the high-speed unit will therefore be most apparent in those situations that involve the loading of lengthy programs or the frequent re-assembly of object code. In either case, the high-speed reader can shorten the development cycle considerably.

SPECIFICATIONS OF THE INTELLEC® 4/MOD 40 SYSTEM

The system specifications are summarized in Table 0-1.

DESCRIPTIVE CONVENTIONS USED IN THIS MANUAL

The INTELLEC®4/MOD 40 system uses both MOS and TTL logic components. TTL components operate in the positive zone, from 0 to +5 Volts. MOS elements operate between −10 Volts and +5 Volts.

The voltage levels which define a logic HIGH or a LOW are different for these two families, but the elements still enjoy a limited compatibility. An MOS output is often used to drive a TTL input, with the addition of a suitable pull-up resistor. In general, however, a TTL output must always be conditioned before being used to drive an MOS Input. Several circuits in the INTELLEC® 4/MOD 40 system use discrete level-shifters for this purpose.

TTL drive levels, moreover, conform to relatively standard definitions. This is true of both inputs and outputs. The same is not true of MOS elements. MOS-level signals are so identified in the pin lists.

In many cases, logic definitions shift several times between input and output. Both positive-true and negative-true definitions are used. This is routine for those accustomed to working with digital logic, and should present little difficulty. To minimize confusion, the text refers specifically to HIGH or to LOW whenever possible, without mentioning negative-true or positive-true signals. A HIGH is always more positive than a LOW.

INTELLEC® 4/MOD 40 Specifications

WORD SIZE:	4 Bit Data 8 Bit Instruction
MEMORY:	1K X 8 — Monitor PROM. Expandable to 4K X 8. 4K X 8 — Program RAM. 4K X 8 — Optional PROM. 320 X 4 — Data (4002 RAM). Expandable to 2560 X 4. All switch selectable.
INSTRUCTION SET:	60, including conditionals, binary and decimal arithmetic, and I/O.
SYSTEM CLOCK:	Crystal-controlled, at nominal 5.185 MHz.
MACHINE CYCLE:	10.8 microseconds
MEMORY CYCLE:	900 nanoseconds
I/O CHANNELS:	3 input/8 output. Expandable to 15 input and 48 output. All ports 4-line TTL.
POWER REQUIREMENTS:	110-120 VAC/60Hz @ 200 W, or 230 VAC/50Hz @ 200 W
INTERNAL POWER:	+5 VDC ±5% @ 8 Amps (max) −10 VDC ±5% @ 1.8 Amps (max)
OPERATING TEMPERATURE:	0°–55° Centigrade
DIMENSIONS:	7" H X 17-1/8" W X 12-1/4" D (Suitable for mounting in standard RETMA 19" rack, with optional rack mount kit).
WEIGHT:	30 lb.
STANDARD SOFTWARE:	System Monitor (4 PROMs) Assembler (Paper Tape)

Table 0-1.

In a few cases, however, it is vital to know how signals are defined. One example is the data outputs from the 4040 processor chip. The levels here are referred to "invisible" events within the processor, and it is absolutely essential to know that the output pins are defined to be negative-true. Thus, an internal "1" which reaches an output pin results in a LOW at that pin.

The only time this becomes problematical is in the labeling of signals. Does the term $\overline{D}$, applied to a signal previously defined as negative-true, mean that the signal is negative-true? Or does it mean that the initial definition is reversed, and that the signal is positive-true at this point?

To minimize this kind of confusion, we adopt the following convention, and apply it consistently throughout text and drawings. All signals are named with respect to positive-true logic, regardless of how the events they refer to are defined. Thus a signal line which is grounded by the actuation of a switch will be labeled $\overline{\text{SWITCH}}$, irrespective of the stipulation that the switch's output is negative-true.

CHAPTER 1
INTELLEC® 4/MOD 40 SYSTEM OVERVIEW

FUNCTIONAL COMPONENTS

In its most basic form, the INTELLEC® 4/MOD 40 system consists of five functional modules. They are:

a) the Central Processor Module

b) the Memory Control Module

c) the RAM Memory Module

d) the Control and Display Panel

e) the PROM Programmer Module

All systems contain these five modules.

Some systems contain additional modules, designed to augment the capabilities of the standard INTELLEC® 4/MOD 40 system. These expand the system's program memory, working memory, or I/O capacity, but they do not affect the system functionally. Optional modules use existing data busses, and they use control signals that are common to the basic configuration. The five modules listed above therefore constitute the functional essence of the INTELLEC® 4/MOD 40 system.

The Central Processor Module contains the processor, program memory, working memory, I/O ports, and a teletype interface. With the simple addition of DC power, the Central Processor Module itself becomes an entirely self-contained computer. The functions performed by the RAM Memory Module, the Control Module, and the Control and Display Panel are subordinate to the operations of the Central Processor Module. Operation of that module is described fully in Chapter 2 of this manual.

The RAM Memory Module is simply a specialized extension of the imm4-43's internal program memory.

The Control Module performs a number of incidental minor functions, but it is basically an interface. As an interface, it does three essential jobs. It contains the logic that permits the RAM Memory Module to simulate the processor module's own internal program memory. It contains the logic that permits programmed reading and writing to program RAM. And it contains the logic that permits the operator to enter data in program RAM, using the selector switches on the Control and Display Panel.

The PROM Programmer Module permits rapid, automatic loading of Intel 4602A and 4702A Programmable Read Only Memories, as described in Chapter 11.

The Control and Display Panel also performs multiple functions, but its principal purpose is to display the internal status of the Processor Module on its panel indicators. The panel also contains the switches that enable console memory access (CMA) operations.

DATA FLOW IN THE SYSTEM

Figure 1-1 shows the flow of data among the five basic modules. Four kinds of data transactions occur at the modular level:

a) **instruction mode**—from program RAM to the processor's instruction register.

b) **program read**—from program RAM to the processor's accumulator.

c) **program write**—from the processor's accumulator to program RAM.

d) **CMA write**—from the Control and Display Panel to program RAM.

In the instruction mode, the RAM Memory Module receives a twelve-bit address from the Central Processor Module. This address passes through the address switching section of the Control Module, before being forwarded to the memory module. The memory module returns an eight-bit instruction to the processor module, as shown.

The program read is accomplished in several stages. First the data being read passes from the memory module to storage in the Control Module. The processor then transfers the data from the Control Module to its own internal registers. In the program read mode, the memory module receives a twelve-bit address from the processor. The processor writes the most significant four bits of the address into I/O ports in the address switching logic. The remaining

eight bits of the address are transmitted to the Control Module on the lower eight lines of the processor's address bus. The eight bits which are returned go to a storage latch in the Control Module, rather than going directly to the processor. The latch is designed to simulate a pair of four-bit input ports, like those found on the processor module itself. From here, the data can be read into the processor module four bits at a time, using two successive I/O instructions. Program instructions can also be read into the processor, four bits at a time, using the Read Program Memory (RPM) instruction.

In the program write mode, the processor module transmits a twelve-bit address to the memory module. This address passes through the Control Module's address switching logic, just as in the program read mode. The processor module simultaneously outputs two successive four-bit words to the memory module, on the appropriate lines of the eight-line bus used to transmit write data. Note that two four-bit I/O transactions are required to store each eight-bit instruction in program memory.

In the CMA mode, the memory address and the data to be written in originate at the console. The address switching and data logic in the Control Module disconnects the processor module, and connects the memory module's data inputs instead to the panel's data outputs. Note that the CMA transaction is an eight-bit parallel write-in, in contrast to the dual four-bit program write.

Observe also that the panel receives a four-line input from the processor module, through an intermediate data buffer on the Control Module. This bus is an extension of the processor module's main internal data bus. The panel logic uses this data input in conjunction with the system timing signals, to develop the operator's status display.

Timing and control signals coordinate the flow of data between the several blocks. The timing is established by circuitry on the Central Processor Module, and will be described in Chapter 2.

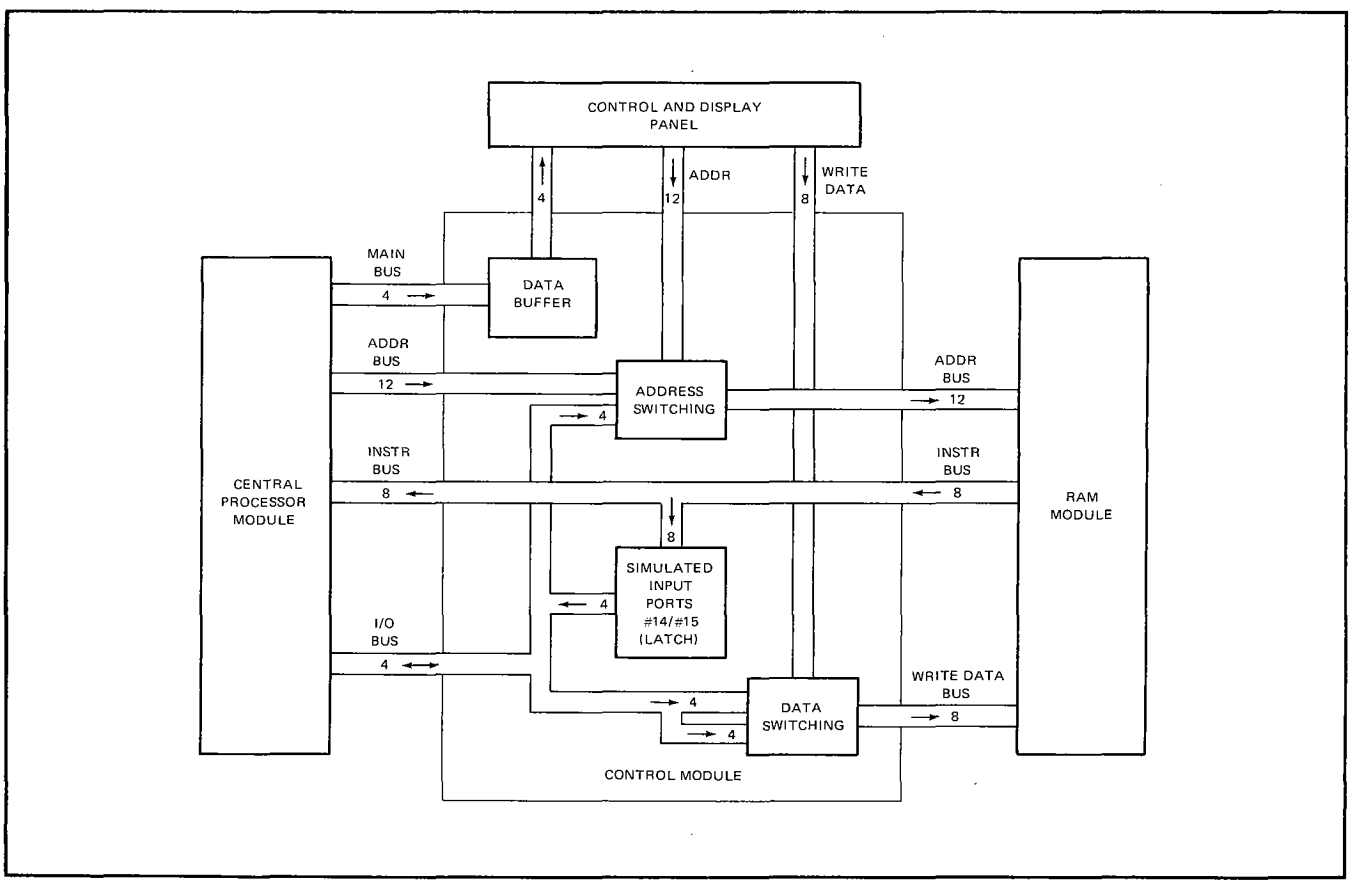


Figure 1-1. INTELLEC® 4 System Block Diagram

CHAPTER 2 THE imm4-43 CENTRAL PROCESSOR MODULE

The imm4-43 Central Processor Module contains the logic necessary to serve as a general purpose microcomputer. All the classic elements of a computer's architecture are assembled on the 6.18 X 8.0 inch card, including:

a) system clock
b) control logic
c) arithmetic logic
d) memory
e) memory control
f) addressable data inputs
g) addressable data outputs

The only external requirement is DC power, at levels of +5 VDC and −10 VDC. All connections to the card are effected through a 100-pin, double-sided PC edge connector (0.125" contact centers).

Control and arithmetic logic functions are performed by an Intel 4040 monolithic CPU. The CPU incorporates accumulator and adder, instruction register and program counter, 24 index registers, and the miscellaneous timing and control circuitry necessary to its operations. The processor recognizes and responds to 60 programmed instructions.

The Central Processor Module's memory includes 1024 eight-bit words of fixed program memory (4702A PROM) and 320 four-bit words of working storage (4002 Random Access Memory). Outputs on the module permit the expansion of both program and working memories, and interface, as well to special read/write program memory elements (2102 Random Access Memory). With the modest addition of a few logic elements, the imm4-43 can address over 4K X 8 locations in PROM, and 2560 locations in 4002 RAM. Intel 2102 RAM storage, up to 4K X 8, can also be accommodated. Additional memory elements are available, both individually and module-mounted, for the purpose of expanding the memory capacity.

Three 4-line input ports and eight 4-line output ports are incorporated into the module. All lines are TTL-compatible. Teletype interface circuitry is also included on the board. KEYBOARD IN, PRINTER OUT, and READER CONTROL lines are provided.

The four 4702A PROM memory elements furnished with the module contain the INTELLEC® 4/MOD 40 System Monitor. This program is designed principally as an aid to the prototype development of OEM microcomputer systems and may not be suitable for some intended applications. PROMs programmed to any specification may replace the standard Monitor PROMs in the four receptacles on the board itself, permitting the user to implement any alternative control program desired.

As a component of the INTELLEC® 4/MOD 40 system, the imm4-43 Central Processor Module is primarily a development model. Its general purpose configuration, however, makes it adaptable to almost any application that calls for the capabilities inherent in a four-bit microprocessor system. In terms of engineering utilization, the Central Processor Module gives the user a set of edge connector pins in which the input-output relationship is determined by the firmware contents of program memory. The hardware investment is minimal, yet such a configuration is capable of flexible and sophisticated responses to its data environment. This permits the imm4-43 to replace many large and cumbersom random logic systems, particularly where limited production proscribes the high cost of developing a more specialized, dedicated system.

For the OEM user, who does not utilize the INTELLEC® chassis, an additional input port can be added with strapping options (see CPU Module schematic, Figure 2-25).

GENERAL FUNCTIONAL DESCRIPTION

The Central Processor Module contains four functional sections:

a) clock
b) processor
c) memory

d) input and output (I/O)

The relationship among these is illustrated in Figure 2-1.

Control signals time and coordinate the functions that each block performs.

The major blocks exchange information via a four-line arterial, known as the main data bus, which links the processor, the memory, and the I/O elements.

Other circuitry on the module provides teletype current interface capability. Level shifting provisions translate external TTL-level control inputs to the MOS levels used on the board.

Timing And Synchronization: The System Cycle

A crystal-controlled clock oscillator establishes system timing. Circuitry on the module divides the 5.185 MHz clock output, to obtain the seventh submultiple. Then the resulting 740.7 KHz signal is separated into two non-overlapping phases and distributed to the other functional elements on the board. The two clock phases are labeled $\overline{01}$ and $\overline{02}$ on the module schematic (see Figure 2-23).

The 4040 processor divides $\overline{02}$ (a 740.7 KHz signal) into one eighth submultiples to obtain a 92.589 KHz $\overline{SYNC}$ signal. The processor then distributes the $\overline{SYNC}$ signal to the memory and to the I/O blocks (see Figure 2-2).

The relationship between the three timing signals is shown in Figure 2-3. The $\overline{SYNC}$ signal establishes a basic 10.8 μsec system cycle (1/92,589). The $\overline{01}$ and the $\overline{02}$ signals divide that basic interval into eight equal sub-intervals, each of which is assigned a specific cyclic function.

The first sub-interval following a $\overline{SYNC}$ pulse is designated A_1. It is followed by the A_2, A_3, M_1, M_2, X_1, X_2, and X_3 intervals, in that order.

The A_1-A_2-A_3 portion is allocated to addressing. During the A_1-A_2-A_3 interval the processor transmits an address to the memory controller, specifying the memory location whose contents is to be read. In the M_1-M_2 interval, the memory returns to the processor the contents of the addressed memory location. The X_1-X_2-X_3 time slot is reserved for processing responses, which may include the input or output of data.

There are two reasons for this timing scheme:

1) The first is that the processor, memory, and I/O all share the main data bus. Conflicts would arise, should two or more blocks attempt to use the bus simultaneously, for incompatible data transfers. By cyclical regulation of all data exchanges, we eliminate the possibility of conflicts. Thus, any data on the bus during A_1-A_2-A_3 is part of an address, and the memory logic recognizes it as such. In like fashion, the processor interprets the bus levels during M_1-M_2 as the contents of the memory location just specified. Because I/O transfers occur only at certain times, output data cannot be mistaken for an address, and vice versa. In this way, the system coordinates the use of the main data bus.

2) The second reason for the timing is less obvious, but equally important. All data exchanges between functional elements of the module take place on the main data bus. The main bus, as stated earlier, is a four-line path. Most practical transactions, however, require more than four bits. It takes twelve bits, for example, to specify one of 4K locations in memory. And a minimum of eight bits are necessary, in order to distinguish the 60 separate instructions that the processor recognizes. Such multiple-byte fields must be dismantled and transmitted serially on the bus, as successive parallel bytes of four bits each. Circuitry at the destination then assembles the bytes, in local registers, to reconstruct the original field. This is referred to as multiplexing. Sectioning of the 10.8 μsec system cycle, as described above, facilitates multiplexing. One four-bit address byte is

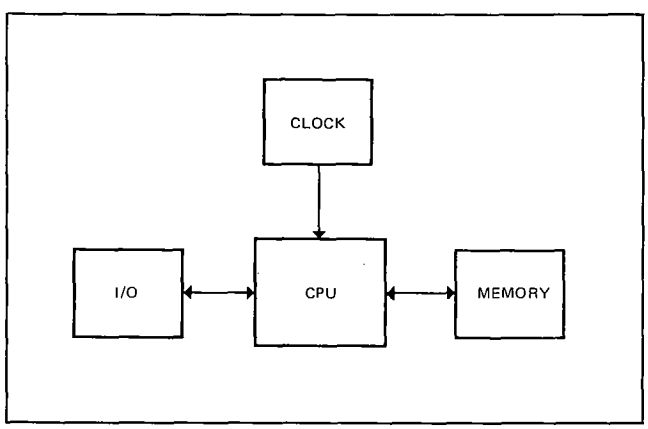


Figure 2-1. Central Processor Module: Functional Block Diagram

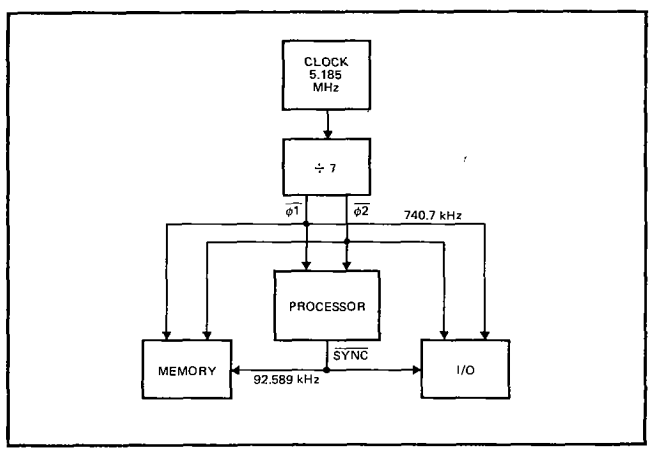


Figure 2-2. System Synchronization

exchanged during each of the subintervals A_1, A_2, and A_3, to form a twelve-bit address field. The instruction word returned during M_1 and M_2 is a two-byte or 8-bit field.

No general rule exists for the X_1-X_2-X_3 portion of the cycle. Exchanges of data may or may not occur, depending upon the instructions given to the processor. The contents of the processor's accumulator are displayed on the bus during X_1. When transfers are indicated, at least one byte will be transmitted, during X_2. A few transactions require transmission of an additional byte, in the X_3 interval. This depends, however, upon the kind of instruction involved.

We shall say more about the sequence of steps in the cycle at the close of this section, when we give some examples of system operation. It is important to be aware of the cycle, since much of the logic of main bus transactions is performed within the monolithic elements on the module, in response to the $\overline{SYNC}$ and the $\overline{01}$ and $\overline{02}$ signals. Neither the mechanism of synchronization, nor the complex switching functions that it governs, are particularly obvious upon first examination of the module schematic.

Memory On The imm4-43

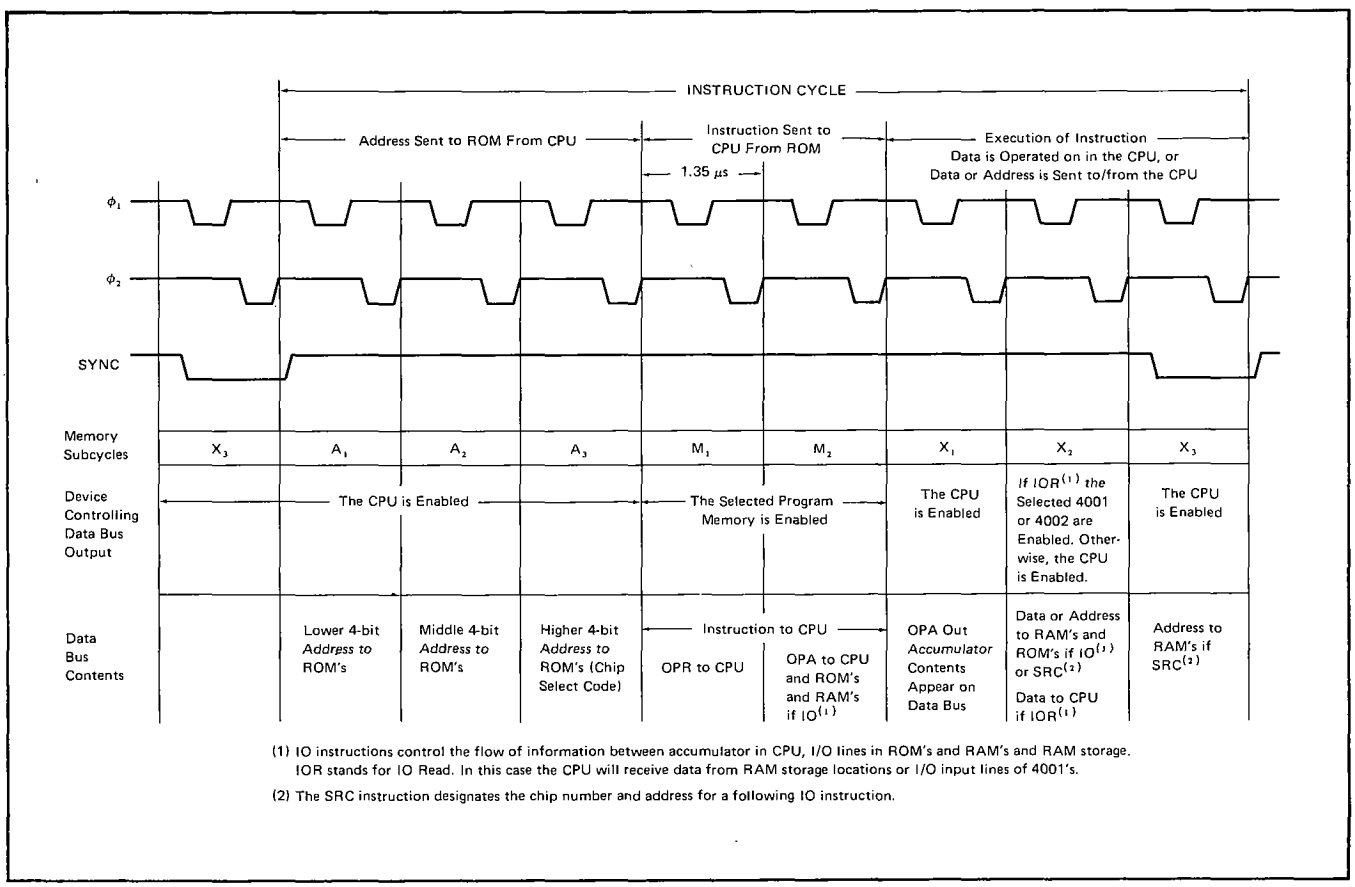
Figure 2-4 shows the organization of the memory elements on the Central Processor Module. It is, however, a conceptual representation only. The diagram helps to explain logically how the processor chip sees and controls the memory I/O. It DOES NOT represent the real electrical configuration, which is described later.

As Figure 2-4 indicates, the module contains two kinds of memory elements:

a) read only memory (ROM), also known as program memory, holds program data. This includes both instructions and program constants. The processor cannot alter the contents of program memory. That is, it cannot store data in a ROM memory location. Thus, although the data in ROM functions as software, it also resembles a hardwired program. For this reason, it is sometimes referred to as firmware.

b) random access memory (RAM) is working storage, or read/write memory. The processor can store data in RAM and later retrieve it, as instructed by the program. RAM memory permits the processor to save intermediate results, in the course of executing a program. It is sometimes called data RAM.

In speaking descriptively, the term Random Access Memory might seem misleading. ROM storage is also randomly accessible. Use of these terms is more conventional than accurate, but as long as they are understood in their intended sense, they serve conveniently to distinguish the two kinds of memory on the board.



(1) IO instructions control the flow of information between accumulator in CPU, I/O lines in ROM's and RAM's and RAM storage. IOR stands for IO Read. In this case the CPU will receive data from RAM storage locations or I/O input lines of 4001's.

(2) The SRC instruction designates the chip number and address for a following IO instruction.

Figure 2-3. System Timing Diagram

Each ROM memory element stores 256 eight-bit words. The four ROMs together give the module a program capacity of 1K X 8 bits.

Each RAM stores 80 four-bit words. The four RAMs together therefore hold 320 words. There are sixty-four directly addressable locations per RAM. The remaining memory words are reached by using special instructions in the processor's repertoire.

Addressing Structure And Memory Control

The processor chip controls the memory elements, by enabling command lines, and by placing address bytes on the main data bus at the appropriate times. Command lines and bus are shown in Figure 2-4.

Because of their different capacities, and because of their different uses, RAM and ROM elements require different addressing schemes.

The processor addresses ROM memory using a twelve-bit field. The eight low-order bits within that field can specify one of 256 locations in any given ROM. They are said to span a "page" of memory. The four high-order bits are available to specify one of sixteen pages. Thus an expanded memory, such as that used within the INTELLEC® 4/MOD 40 system, could contain as many as sixteen ROM elements, all commanded by a single enabling line, and differentiated from one another by a four-bit page number. A single processor command line, $\overline{\text{CM-ROM}}$, is provided for that purpose. The 4040 CPU contains a second ROM command line as well, but the INTELLEC® 4/MOD 40 system does not use this provision.

The processor addresses RAM on the other hand using an eight-bit address. There are 64 directly addressable locations in a RAM element, and six bits are therefore sufficient to span one page of RAM. Two bits remain, to identify the page. Two bits can distinguish among four pages, and a separate command line will therefore be required for every four RAMs used. A system such as the INTELLEC® 4/MOD 40 may have as many as 32 pages of RAM.

To provide for such an expansion, the processor is equipped with four $\overline{\text{CM-RAM}}$ lines, subscripted 0 to 3. The $\overline{\text{CM-RAM}}$ 0 line controls the four RAMs on the Central Processor Module. The remaining three lines can be decoded to service any number of RAM banks within the system's upper limit.

The $\overline{\text{CM-ROM}}$ line, when enabled, connects the ROM inputs to the main data bus. The $\overline{\text{CM-RAM}}$ lines perform a similar function for the RAM elements. By placing a four-bit byte on the bus, and by simultaneously manipulating command lines, the processor directs data to the RAM bank, to the ROM bank, or to both at the same time.

Module Input And Output Ports

As shown in Figure 2-4, each of the module's four input ports, and each of its eight output ports, is associated with one of the memory elements on the module. All are four-line ports. This arrangement has three principal advantages:

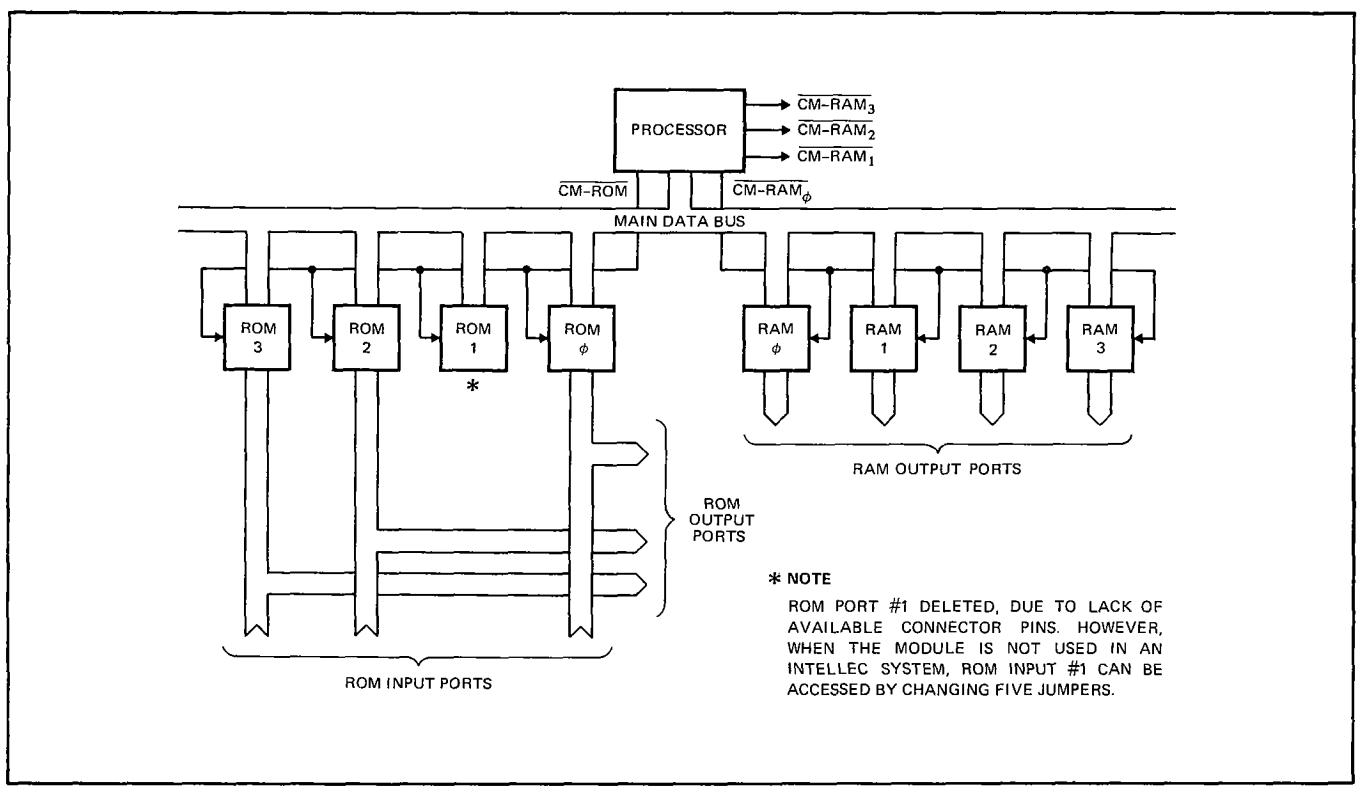


Figure 2-4. Memory and I/O Control

a) It eliminates the need for special address logic for each port, since the port can share the memory's logic.

b) It permits a greater functional density on the memory chips, a consolidation that results in system economies.

c) It obviates the need for an extended addressing system in order to perform I/O operations. An input or output port is specified automatically, every time a memory page is addressed. A subsequent instruction then identifies the operation as an input, output, or memory transaction.

The Processing Cycle

Everything that the processor does is the direct result of an instruction, or the lack of one, as stored in memory. Every program step therefore has two parts. The first is the fetch, in which the processor obtains an instruction word from the program memory. The second is the execution, in which the processor performs the specified operation. These two segments together complete a system cycle.

The processor maintains a counter which tells it where to find the next instruction in ROM memory. This address is placed on the main data bus in three bytes, during A_1-A_2-A_3 of the processor's cycle. The least significant byte is transmitted in A_1, the middle byte in A_2, and the most significant byte in A_3. The ROM elements sense the page and memory location of the instruction, and return two bytes to the processor, on the main bus, during M_1-M_2. A_1-A_2-A_3-M_1-M_2 constitutes the fetch portion of the processing cycle.

The two bytes returned to the processor during M_1 and M_2 are assembled in the processor's instruction register, to make up an eight-bit field. The four bits returned during M_1, are called the OPR. They specify the kind of operation to be performed: procedural, I/O, or arithmetic. The OPA, returned during M_2, is a modifier. It may specify the location of an information byte that the processor will need to execute the instruction, or it may specify a sub-operation of the instruction group indicated by the OPR. The function of the OPA byte depends upon the nature of the instruction.

In most cases, the processor executes the instruction during the remainder of the 10.8 microsecond processor cycle, X_1-X_2-X_3.

There are five instructions which are complicated enough to require more information than can be conveyed in the four bits of the OPA. These instructions use a double-length format, requiring 21.6 microseconds to complete.

The first four bits, the OPR, of a double-length instruction are distinctive, so that the processor immediately recognizes it as part of a double-length instruction. When this happens the processor will interpret the results of its next eight-bit fetch, not as a single instruction, but as the additional information necessary to complete the execution of an instruction. For this reason the successive portions of double-length instructions are always stored in sequentially adjacent memory locations, so that the processor always knows where to look for the second half of the instruction.

An address is sent out on the bus during A_1-A_2-A_3 of the second half of the double-length instruction cycle, just as for a normal instruction fetch. The information returned during M_1-M_2 is used to finish executing the command, during X_1-X_2-X_3 of the second 10.8 microsecond interval. The processor reverts to its normal instruction mode, beginning with the next fetch operation.

TYPICAL MODULE OPERATIONS

A few illustrative examples will help to clarify the interaction between processor, memory, inputs and outputs. We give examples of:

a) a RAM read
b) a RAM write
c) an input sequence
d) an output sequence
e) a ROM read

All these transactions require several instructions to execute. Every transfer of data between the processor and another element consists of three logically distinct steps:

1) Specification of an address
2) Specification of the kind of transfer
3) The actual transfer

RAM Read

A RAM Read sequence will be performed whenever the program requires the four bits of information stored in a particular RAM location. Readout is nondestructive; that is, the contents of the cell being read are not altered or erased in the process.

In the symbolic notation used by the programmer, a typical read sequence might look like this:

```
LDM     XDDD
DCL
LDM     DDDD
XCH     RRR0
LDM     DDDD
XCH     RRR1
SRC     RRR1
RDM
```

The letters in the left-hand column are instruction mnemonics. They represent the kind of operation to be performed. Those in the right column are operands. They represent the additional information that is required for the execution of certain commands.

The first instruction (LDM XDDD) loads the program constant "XDDD" into the processor's accumulator. The programmer, by selecting the value of "XDDD", designates one of eight RAM command lines (specified by the octal

field "DDD"), where each RAM command line controls four RAMs. The binary digit "X" is unspecified, since it is not essential to the identification of the line. In the case where the Central Processor Module stands alone, the programmer would choose XDDD = X000, identifying $\overline{\text{CM-RAM}}_0$ as the command line to be enabled.

The next instruction (DCL) transfers the accumulator's contents to the processor's command control register, thus enabling the designated $\overline{\text{CM-RAM}}$ line. This line remains selected until specifically altered by another DCL, and is used to activate a memory bank consisting of four RAM pages.

Having designated the RAM bank, the programmer must specify the page and the location within the page. Remember that this requires an eight-bit address. The two most significant bits identify the RAM, and the remaining bits specify one of 64 four-bit locations therein. The next four instructions are designed to assemble the eight-bit field required.

The third instruction (LDM DDDD) again loads a constant of the programmer's choice into the accumulator. "DDDD" will be the four most significant bits of the desired address.

The fourth instruction (XCH RRR0) places the contents of the accumulator in one of the processor's index registers (specified by RRR0). The register is chosen by the programmer, but observe that the last digit of the operand is always "0". This means that the most significant address byte will always be stored in one of the **even-numbered** index registers.

Instructions 5 and 6 select and load the four least significant digits of the address.

Note, however, that the last digit of instruction number 6 (XCH RRR1) is a "1". Thus, the least significant address byte will always be stored in an **odd-numbered** index register. Furthermore, this odd-numbered register will always be the next register in sequence following that which contains the most significant byte. If one byte is in register number 0, for example, the next must be placed in register number 1. The reason for this will be apparent in a moment.

The seventh command (SRC RRR1) directs transfers of the two address bytes to the Main Data Bus. To do this, the processor forwards the most significant byte during X_2 of its cycle. The second byte is placed on the bus during X_3. The reason for storing these bytes in numerically adjacent registers is now apparent. By specifying a single odd-numbered register, "RRR1", the instruction implicitly indicates the location of both bytes. An orderly transfer of data is thus assured.

The address transferred to the bus during execution of the SRC command enters the data bus inputs of the RAMs on the enabled $\overline{\text{CM-RAM}}$ line. The RAM addressed by the two most significant bits registers the address. The other three RAMs ignore it. The designated RAM saves the address in internal latches, pending the next I/O instruction.

Steps 1 through 7 have now specified an address. All that remains is to specify the operation desired and to execute the transfer. The next step (RDM) indicates a readout of the previously designated RAM address. That is accomplished as follows.

The first four instruction digits returned to the processor during M_1 of the eighth cycle indicate an I/O operation to follow. When the processor senses this condition, it enables the $\overline{\text{CM-ROM}}$ line and the presently designated $\overline{\text{CM-RAM}}$ line, during M_2. This permits the OPA portion of the instruction on the bus to enter the input of any memory location alerted by the preceding SRC instruction. The OPA specifies the kind of I/O operation required, a RAM readout in this case. The designated RAM responds by placing the contents of the requested cell on the main data bus during the X_2 interval of the RDM instruction. The processor recognizes data on the bus at this time as the contents of the requested cell and responds by placing this byte in the accumulator. The memory readout is now complete.

RAM Write

Writing into a RAM cell is very similar to the process of reading from RAM. The same steps are used to designate an address. Only the last two steps are different. A typical RAM Write sequence looks as follows:

```
LDM     XDDD
DCL
LDM     DDDD
XCH     RRR0
LDM     DDDD
XCH     RRR1
SRC     RRR1
LDM     DDDD
WRM
```

It is immediately apparent that Steps 1 through 7 of the write sequence are identical to the Steps 1 through 7 of memory readout. These steps establish the address, as required.

The eighth step (LDM DDDD) loads the accumulator with the data to be written into the RAM. This step, or one similar, is necessary, since all transfers to RAM are executed via the processor's accumulator. For the sake of this example we assume that the data is a program constant "DDDD" established by the programmer. It might just as easily be an intermediate result loaded into the accumulator from one of the index registers.

The final step (WRM) writes the contents of the accumulator into the specified memory location. The processor executes this instruction as follows.

The OPR byte fetched during M_1 alerts the processor to an impending I/O operation. The processor enables the selected $\overline{\text{CM-RAM}}$ line in time for the designated RAM to receive the OPA byte, at time M_2. The processor places its

accumulator's contents on the bus during X_2, and the RAM, recognizing a write instruction, responds by routing the contents of the bus to the selected memory location.

Input Operation

Input operations transfer data at one of the input ports into the processor's accumulator, via the main data bus. An input sequence typically looks like this:

```
LDM     DDDD
XCH     RRR0
SRC     RRR1
RDR
```

Recall that the module's input ports are associated with the ROM memory elements. To specify an input port, we need only identify a page in ROM memory. Since no more than sixteen ROMs are used, four bits suffice to identify the page.

The first instruction (LDM DDDD) specifies the desired ROM port. The programmer selects the port, by choosing the value of "DDDD". This value is then loaded into one of the even-numbered index registers, by the second instruction (XCH RRR0).

The third instruction (SRC RRR1) puts the contents of register pair "RRR1" on the Main Data Bus, during X_2 and X_3. The value stored in register RRR0 will be sent out at time X_2, specifying the input port. The $\overline{\text{CM-ROM}}$ command line is enabled simultaneously. Thus the processor alerts the selected ROM port to an impending transfer.

Note that the SRC instruction also causes the processor to release the contents of index register "RRR1" to the data bus, at time X_3. This information, however, is superfluous for the purpose of output. The ROM bank simply ignores the nonspecific second byte, when executing an I/O instruction.

The last instruction (RDR) specifies an input operation in which the data at the previously selected ROM input port is transferred to the processor's accumulator. Execution is similar to other I/O group instructions. The processor, sensing the OPR of an I/O group instruction, enables the $\overline{\text{CM-ROM}}$ line during M_2. The subsequent OPA then directs the selected ROM to place the input data on the main data bus during X_2. The processor transfers the X_2 contents of the bus to its accumulator register, completing the input sequence.

Output Operation

An output operation transfers data from the processor's accumulator to one of the system output ports, via the main data bus. The output levels remain latched in the designated output port, until they are displaced by another command which addresses the same port, or until the memory element is reset ($\overline{\text{4002 RESET}}$).

An output may be addressed to either a ROM or a RAM port. The procedure in both cases is similar. The chief difference is that a $\overline{\text{CM-RAM}}$ line must be designated when writing to a RAM port. ROMs, of course, do not require a DCL. The general preface to a RAM output is this:

```
LDM     XDDD
DCL
```

These steps select and designate the $\overline{\text{CM-RAM}}$ line, just as in RAM operations described previously. The subsequent steps, which identify the output port, are applicable to both RAM and ROM output sequences:

```
LDM     DDDD
XCH     RRR0
SRC     RRR1
LDM     DDDD
WRR     (or WMP)
```

The next three steps load the output port address into the accumulator, transfer the load to one of the even-numbered index registers and place the contents of "RRR0" on the main bus during X_2 of the fifth cycle. This process alerts one ROM and one RAM chip to an impending I/O operation. As in the case of an input operation, the unspecified contents of register "RRR1" are placed on the bus at time X_3. Just as in the input operation, these four bits are ignored whenever an I/O operation is executed.

The next step loads data, a program constant in this example, into the accumulator. Conditions are now set for the output transfer.

The final steps command the output. The instruction will be a WRR, calling for a write at the enabled ROM port, or a WMP, specifying the enabled RAM port. In both cases, execution is similar. The processor enables $\overline{\text{CM-ROM}}$ and $\overline{\text{CM-RAM}}$ after receiving an I/O OPR at time M_1. The OPA at M_2 causes the processor to place the accumulator contents on the bus at X_2 time, and the designated memory element latches this data in its output port, completing the output transfer.

ROM Read

Data in program memory is of two kinds: instructions and program data. Data stored in memory for use by the program is known as a program constant. The term constant is used, because this data has the same value in all program runs, regardless of the arguments or variables that may be furnished. All program data committed to ROM is naturally constant, since the contents of a read only memory never vary.

One way of obtaining a data constant from program memory is by using an LDM instruction. The use of this command has already been described in earlier examples. When an instruction such as (LDM DDDD) is performed, the constant "DDDD" is loaded directly into the processor's accumulator. An operation such as this is known as an "immediate instruction".

It is sometimes more efficient to transfer an eight-bit

field directly into one of the processor's eight pairs of index registers than to execute two successive four-bit transfers. For example, we might find it convenient to load the binary constant $D_2D_2D_2D_2D_1D_1D_1D_1$ into register pair RRR0 and RRR1. In programming symbology, such a transfer looks like this:

$D_2D_2D_2D_2$ ⟶ RRR0

$D_1D_1D_1D_1$ ⟶ RRR1

Each ROM memory location is eight bits deep and accommodates an eight binary digit field. But the procedure for fetching the constant $D_2D_2D_2D_2D_1D_1D_1D_1$ differs from the procedure for a RAM memory read. One way of doing this is to use a double length instruction known as an immediate fetch.

This instruction has the form:

FIM	RRR0
$D_2D_2D_2D_2$	$D_1D_1D_1D_1$

where the mnemonic "FIM" indicates the fetch and "RRR0" indicates the even-numbered register of the pair where the constant will be stored.

The ROM location next in sequence, following the address where the "FIM" OPR is stored, contains the constant $D_2D_2D_2D_2D_1D_1D_1D_1$. The processor sends the contents of its instruction counter out on the data bus at time A_1-A_2-A_3, but instead of interpreting the M_1-M_2 contents of the bus as an instruction the processor simply stores this data in the designated index registers. In this way, the ROM data is consigned to a location where it may be addressed by the program (note that the processor cannot address its own instruction register).

There is no counterpart for the RAM read process, when reading out of ROM. RAM is always read in the X_1-X_2-X_3 portion of the system cycle. ROM, on the other hand, is always addressed in A_1-A_2-A_3 and read in M_1-M_2. Readouts from ROM, other than normal instruction fetches, always make use of an immediate fetch (FIM) or an indirect fetch (FIN).

The indirect fetch is similar to the immediate fetch, in that it requires two cycles for its completion (21.6 μsec). The FIN command, however, is a single-length instruction and looks like this:

FIN	RRR0

"RRR0" indicates the even-numbered register of the pair where the results of the fetch will be stored.

To fetch data from ROM, the processor uses the eight bits in register pair "0000" and "0001" as the low-order segment of an address. This naturally assumes that a suitable address has been stored previously in those two registers. The instruction further assumes that the cell to be read is on the same memory page as the FIN instruction itself. The processor therefore sends the high-order four bits in its program counter, plus the eight bits in register pair number 0, as an address during A_1-A_2-A_3 of the next system cycle. The data returned during M_1-M_2 is placed in the designated register pair, completing the fetch. The system returns to a normal instruction cycle, beginning with its next program instruction.

The net result of either fetch, FIM or FIN, is thus the same. The principal difference is that an indirect fetch may specify any cell on the same memory page, while the immediate fetch always accesses the next sequential location in memory. The indirect fetch is thus a somewhat more versatile instruction. The immediate fetch, however, requires no preparation sequence and may therefore be more economical of processing time.

4040 FOUR-BIT CENTRAL PROCESSOR UNIT

The Intel® 4040 Central Processor Unit is the functional core of the imm4-43 Central Processor Module. This monolithic MOS processor performs the logical and control functions generally associated with the central processing units of larger machines. The 4040 recognizes and executes 60 different eight-bit binary instructions.

The 4040 CPU requires input power at levels of −10 VDC and +5 VDC. Its average power consumption is 225 mW.

Control inputs to the 4040 consist of:

$\overline{\phi_1}$ and $\overline{\phi_2}$ $\overline{\text{CLOCK}}$ signals (741 KHz) – MOS levels

$\overline{\text{CPU RESET}}$
$\overline{\text{STOP}}$
$\overline{\text{INTERRUPT}}$
$\overline{\text{TEST}}$
} TTL levels

The $\overline{\text{CLOCK}}$ is a periodic signal input obtained from an external reference oscillator. $\overline{\text{CPU RESET}}$, $\overline{\text{STOP}}$, $\overline{\text{INTERRUPT}}$, and $\overline{\text{TEST}}$ are intermittent inputs that provide for external control of the program which the processor executes. All inputs to the CPU are defined as negative-true.

The 4040 CPU generates the following outputs:

$\overline{\text{SYNC}}$
$\overline{\text{CM-ROM}}_{0\text{-}1}$
$\overline{\text{CM-RAM}}_{0\text{-}3}$
$\overline{\text{STOP ACKNOWLEDGE}}$
$\overline{\text{INTERRUPT ACKNOWLEDGE}}$

$\overline{\text{SYNC}}$ is the eighth submultiple of the $\overline{\text{CLOCK}}$ input frequency. It is derived by countdown circuitry on the chip itself and is used externally to synchronize various kinds of data transfers, including memory reference and input or output transactions. $\overline{\text{CM-ROM}}$ and $\overline{\text{CM-RAM}}$ outputs are under program control and permit the processor to address selected groups of ROM and RAM chips. These command lines are instrumental in the fetching of instructions, in the temporary storage of data, and in the execution of input and output instructions. $\overline{\text{STOP ACKNOWLEDGE}}$ indicates externally when the processor is halted as a result of a

$\overline{\text{STOP}}$ command input. $\overline{\text{INTERRUPT ACKNOWLEDGE}}$ is true whenever the CPU is engaged in the processing of an $\overline{\text{INTERRUPT}}$ request. Control outputs from the 4040 are at MOS levels, negative-true.

Data in and out of the CPU passes through a four-line port, the principal nexus of the main data bus which connects the processor to other elements in the system. The port's lines are negative-true, and its electrical characteristics are typical of the other MOS inputs and outputs.

Functional Description

Figure 3-5 is a block diagram of the 4040 CPU, showing the most important details of its internal construction. The chip contains the following functional sections:

a) timing
b) control logic
c) arithmetic logic
d) program counter and stack
e) instruction register
f) bus multiplexer
g) memory control
h) index registers
i) I/O buffer

The timing section accepts the $\overline{01}$ and $\overline{02}$ clock phases from the external reference oscillator. It contains the divide-by-eight countdown circuitry which develops the SYNC reference from these signals. The output of the timing section coordinates the operations of the control logic.

The control logic section uses the timer output to direct the operations of the remaining blocks. It initiates an instruction fetch, at the beginning of the cycle, by sending an address to the program memory during A_1-A_2-A_3 and by simultaneously enabling the $\overline{\text{CM-ROM}}$ command line. It directs the storage of the memory's reply, in the chip's instruction register. The control logic decodes and interprets the contents of the instruction register, determines the operation to be performed, and develops the signals that cue the arithmetic and memory control sections during execution of the instruction.

The arithmetic logic section is equipped to perform all the required data manipulations, under supervision of the control logic. This block contains the main working register, the accumulator, into which the results of all additions are placed. It also contains a four-bit adder, an auxiliary register for storing addends, a carry flag for extended arithmetic operations, and sensing logic to enable conditional branching.

The program counter and stack is a set of eight twelve-bit registers. The program counter is used to hold the address in memory where the next instruction to be executed is stored. The control logic maintains the program counter automatically, incrementing the counter's contents immediately following each A_3 fetch. Thus the location

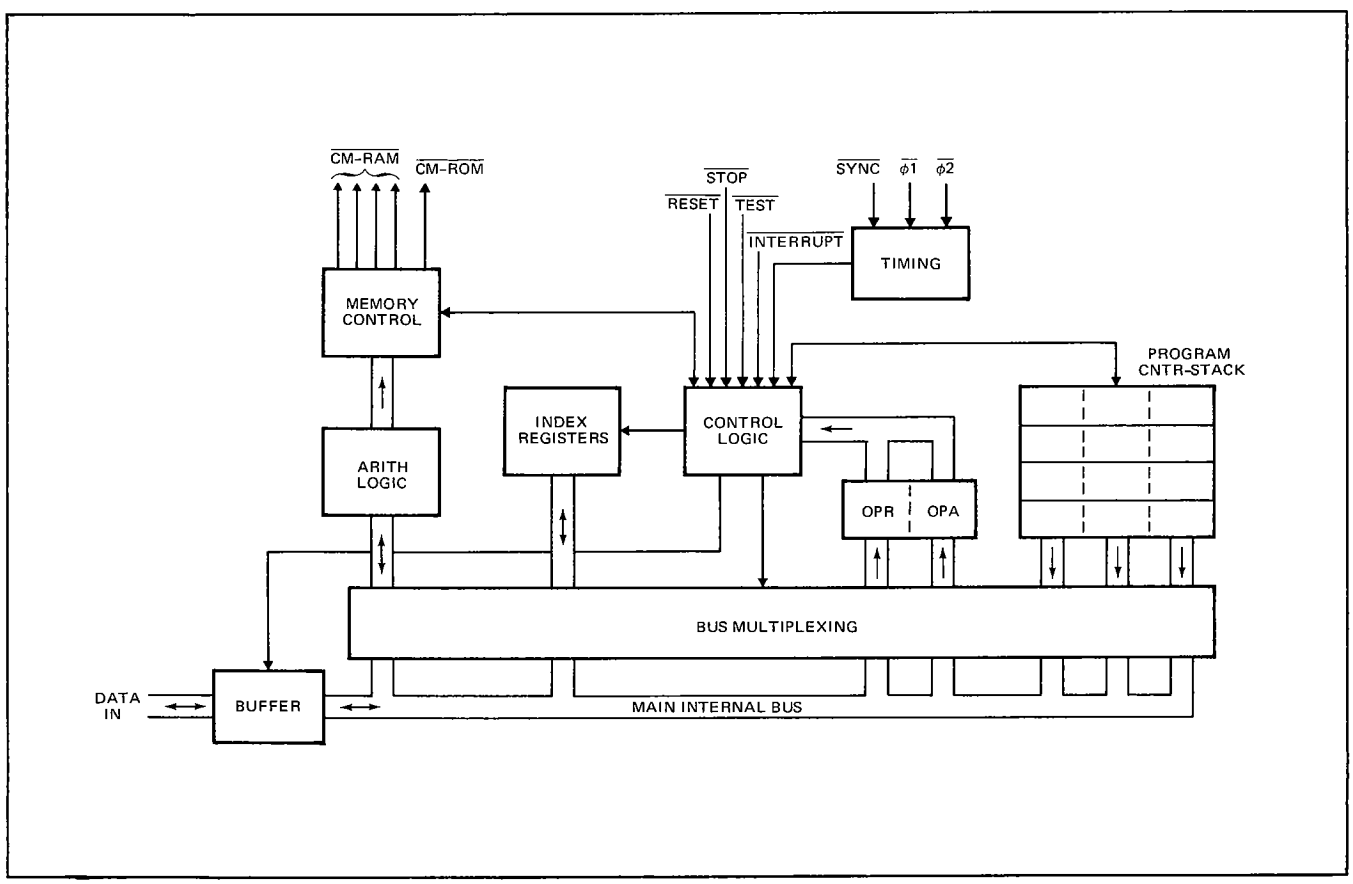


Figure 2-5. 4040 Processor Functional Block

of the next instruction is always available in the program counter. In the event of a program jump, the control logic replaces the current contents of the program counter with a new memory address, and program execution continues from that point. The program counter is always set to zero, following the application of a CPU RESET.

The procedure for maintaining the program counter during subroutine jumps is different. A subroutine sequence has to merge with the main program again, following execution of the subroutine. An ordinary program jump looks like this graphically:

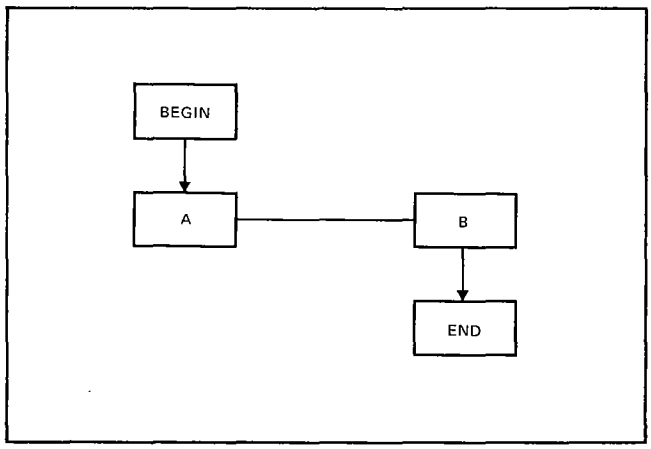


Figure 2-6. Program Jump

A subroutine jump, on the other hand, looks like this:

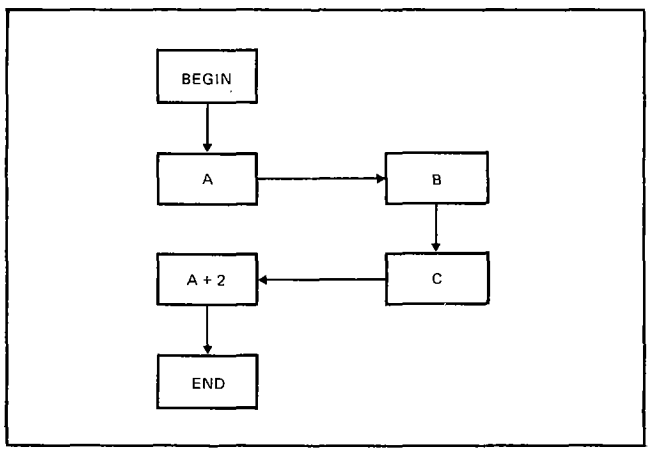


Figure 2-7. Subroutine Jump

In order to execute a subroutine properly, the processor must save the contents of its program counter, before executing a jump. This preliminary assures that the main program can resume its operation, following execution of the subroutine. The stack provides the storage necessary for holding return addresses.

The stack consists of seven registers, each of them similar to the program counter itself. Together, they form a pushdown address stack, for accommodating nested subroutines. A stack pointer is maintained automatically by the processor logic.

By way of example, suppose that a "jump to subroutine" instruction occurs in the course of the main program flow. The processor logic stores the jump address in the "nearest" stack register; that is, the secondary level register, as determined by the internal pointer logic. The secondary register now becomes the program counter. This register contains the effective program address, and this register is incremented after each instruction fetch.

The end of the subroutine sequence is indicated by a "branch back" instruction. When a branch back occurs, the stack pointer moves in the reverse direction. The original register again becomes the program counter, and execution of the main program resumes immediately following the point where the subroutine was called.

The stack pointer and the eight program counter/stack registers allow the program to call seven levels of subroutines. That is, the program can call subroutine 1, subroutine 1 calls subroutine 2, and so on until the sixth subroutine calls the seventh subroutine. Figure 2-8 shows three-level subroutine nesting.

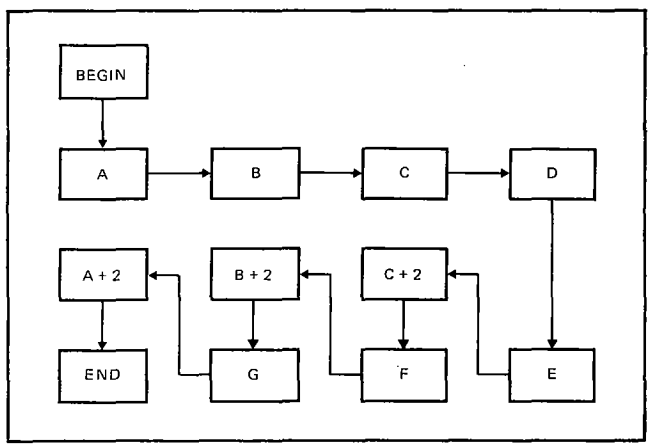


Figure 2-8. Nested Subroutines

Should an eighth subroutine be called, the deepest stack address would be displaced by the address of the eighth subroutine. Thus, the contents of the original program counter would be lost, and return to the main program would be impossible.

The stack also allows for seven-level interrupt processing. When an interrupt occurs, the program counter is not incremented. Instead a Jump to Subroutine (JMS) at location 3 instruction is forced onto the internal data bus. The processor stores the subroutine address (in this case 3) in the next available stack register just as it does during any subroutine call. The stack pointer points to this stack register, making it the current program counter; thus, the instruction at location 3 is executed next. This instruction will normally direct program control to an interrupt service routine. The first instructions of the interrupt service routine should save the status of the processor (i.e., the contents of the processor's registers) in one of the index register banks. Before "branching back", the service

routine must restore the processor status. When the branch back occurs, the stack pointer returns to the previous program counter. Because the contents of this program counter were not incremented when the interrupt occurred, execution of the interrupted program resumes where it was stopped.

The instruction register is an eight-bit register used to store the OPR and OPA bytes returned from the program memory during M_1 and M_2 of the system cycle. The instruction field stored in the register is available to the control logic section. The control logic decodes the instruction, to determine the appropriate sequence of operations, then directs the execution of the command.

The bus multiplexing logic controls use of the main internal data bus, in response to the directions of the control logic. It places the program address on the bus, in three sequential bytes, during A_1-A_2-A_3 of the processing cycle. It then routes the contents of the bus during M_1 and M_2 to the OPR and OPA segments of the instruction register. The multiplex logic controls the switching of the bus as required, during the execution phase of the cycle.

The memory control section selects and buffers the $\overline{\text{CM-ROM}}$ and the $\overline{\text{CM-RAM}}$ lines as necessary in the course of the cycle. It is supervised in this function, by the control logic. Note that the memory control section communicates with the arithmetic logic section, via a bussing arrangement. The $\overline{\text{CM-RAM}}$ command line, as described in Section 3.2, is selected by first depositing a numerical designator in the accumulator register. This designator is then transferred to a register in the memory control, when the DCL command is issued. The $\overline{\text{CM-ROM}}$ command line is under the direct control of the control logic section.

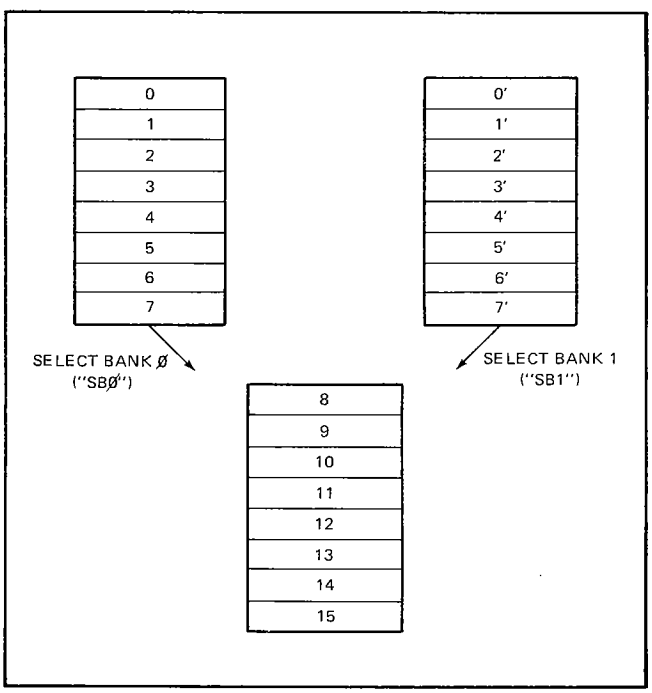


Figure 2–9. Index Register Addressing

The index registers serve as the working internal storage for the processor chip. Twenty-four individually addressable four-bit registers are provided.

Index registers are organized internally as three distinct blocks, each containing eight consecutively addressable registers. One bank is enabled continuously, and registers in that bank may be referenced at any time by any program. Only one of the two banks that remain, however, will be enabled at any given time. Bank selection is a programmed function, and any reference to a particular register in one of these two banks must therefore have been preceded at some point in the program by an instruction explicitly enabling that bank. Figure 2–9 shows the addressing structure within the processor.

The program may address registers individually, or it may address them as members of a logical pair. For the purpose of some instructions, the selected register bank and the resident bank of registers may be visualized as consisting of eight 8-bit registers, as illustrated in Figure 2-10. When an instruction of this kind is employed, the pair will be identified in the instruction operand (OPA byte) by specifying either the even- or the odd-numbered member of the pair.

The buffer section of the 4040, under the supervision of the control logic, acts principally as a switch. It connects or disconnects the main internal data bus from the main external data bus, thus maintaining logical communication between the processor and the components of the system.

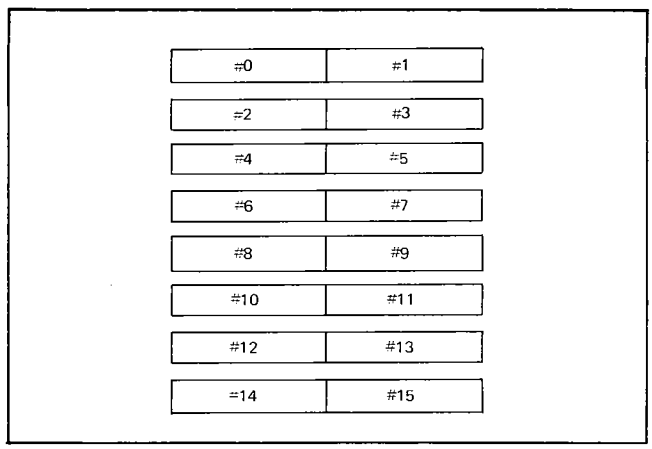


Figure 2-10. Paired Index Registers

4040 Instruction Set

The instruction set for the 4040 CPU consists of 60 instructions which provide for register-to-register and register-to-memory transfers, input and output of four-bit quantities, binary and decimal arithmetic, and boolean logic operations. A full range of program control instructions enable the programming of conditional, unconditional, and indirect jumps as well as the efficient management of subroutine calls and returns. The built-in interrupt facility may be enabled or inhibited as the program logic may require.

Both single word and double word instruction for-

mats are used. Each single word instruction occupies one eight-bit ROM memory location. Five instructions require a sixteen-bit field and are stored in two consecutive memory locations. Appendix A describes the CPU's instruction set in detail.

Single word instructions are generally completed within one processor cycle, or eight clock periods, as determined by the system clock. The first three periods, A_1-A_2-A_3 are used to address program memory. The two four-bit bytes of the instruction are returned to the processor in the next two periods, M_1 and M_2. Execution of the instruction is accomplished in the remaining three clock periods, X_1-X_2-X_3.

An exception is the "indirect fetch" command (FIN), a unit length instruction requiring two full processor cycles for completion. The OPR and OPA of the indirect fetch specify the fetch plus the even-numbered member of a register pair. The contents of this pair is sent out as an address during A_1-A_2 of the succeeding cycle, and is followed at A_3 by the high-order byte in the processor's instruction register.

The contents of the addressed ROM memory location are returned on the bus in M_1 and M_2, the first byte being stored in index register #0 and the second in #1.

The complete instruction set for the 4040 is listed in Appendix A, with a brief explanation of the logical operations effected by each command.

Typical Processor Cycle

As an illustrative example of processor operation, consider the sequence of steps involved in a typical double length instruction, the "immediate fetch" (FIM). The FIM instruction is stored in two successive ROM locations, like this:

	OPR	OPA
LOCATION #1	0 0 1 0	R R R 0
LOCATION #2	$D_2D_2D_2D_2$	$D_1D_1D_1D_1$

$D_2D_2D_2D_2D_1D_1D_1D_1$ represents a binary program constant consisting of two four-bit bytes. RRR is an octal field which specifies one of the eight index register pairs in the processor. The object of the instruction is to transfer $D_2D_2D_2D_2D_1D_1D_1D_1$ to the index register pair stipulated by the OPA RRR0 as follows:

$D_2D_2D_2D_2 \longrightarrow$ R R R 0

$D_1D_1D_1D_1 \longrightarrow$ R R R 1

Suppose that the first half of the instruction is stored in the thirty-second cell (#31) of the second page in program memory (page #1). The contents of the program counter will be (0001) (0001) (1111) at the beginning of the first instruction cycle, indicating the location of the current program instruction.

In response to the signals from the timing section, the control logic sends cues to the multiplexer and to the buffer logic. The multiplexer responds by placing the least significant byte in the instruction counter on the main internal bus at time A_1. The buffer gates this byte through to the processor's output port, and to its ultimate destination in the ROM memory bank.

The control logic now increments the least significant byte (1111 + 1 = CARRY + 0000) and returns it to storage at the top of the program counter stack. Note that incrementation results in a carry, so that the byte returned to the program counter is 0000. The control logic records the occurrence of the carryover, for use in A_2.

In the next period, A_2, the multiplex logic gates the middle address byte onto the main internal bus, where it enters the buffer and passes to the program memory. The control logic, finding that a carryover resulted from the incrementation of the first address byte, increments the second byte as well, before returning it to storage in the top of the program stack. The second byte now contains: 0001 + 1 = 0010 + $\overline{\text{CARRY}}$.

The most significant byte in the program counter is forwarded to the bus during A_3, just as the first two bytes were previously. The control logic, finding that no carryover resulted from the incrementation of the second byte, returns the most significant byte to the program stack unchanged: (0001 + 0 = 0001 + $\overline{\text{CARRY}}$).

Transmission of the address field to program memory is now completed, and the current address of the next memory cell is stored in the program stack.

The memory stores the address transmitted by the CPU and responds by returning the OPR to the processor, during M_1: (0010). This byte is gated through the enabled input buffer, through the multiplex block, to storage in the OPR section of the instruction register. In similar fashion, the OPA sent from the memory during M_2 is gated through the multiplex section to the OPA section of the instruction register.

Instruction decoding depends on the type of instruction. The control logic always examines the OPR byte, immediately upon its return in M_1, to determine if any immediate action is required. A code of (1110), for example, would indicate that the instruction in progress was one of the I/O and RAM group, as shown in Appendix A. Execution of any of these instructions requires that the memory elements on the module should also receive the OPA instruction byte in M_2. The control logic, under these circumstances, would enable the $\overline{\text{CM-ROM}}$ and the $\overline{\text{CM-RAM}}$ lines, in time for the memory elements to be alerted to the operation in progress.

In our present example, the OPR indicates to the control logic that one of two kinds of instructions will be performed, either the "immediate fetch" (FIM) or the "send register control" (SRC). The processor logic will therefore have to discriminate on the basis of the OPA to follow. The control logic examines the last bit of the OPA field, and finding it even (containing a "0"), concludes a fetch.

For the purposes of an "immediate fetch," X_1-X_2-X_3 of the first cycle is idle time. No processor action is required.

The addressing cycle for the second half of the instruction is similar to the first. The three bytes of the program address are multiplexed to the main bus, during A_1-A_2-A_3, accessing the location where the program constant $D_2D_2D_2$-$D_2D_1D_1D_1D_1$ is stored.

The control logic, however, has been alerted by the preceding OPR that the normal instruction cycle is temporarily suspended. As a result, it instructs the multiplexer logic to route the M_1 and M_2 bytes to the index register pair specified by the first cycle OPA. $D_2D_2D_2D_2$ is stored in the even numbered register and $D_1D_1D_1D_1$ in the odd. In the case of the FIM, execution of the instruction is completed at the end of M_2. X_1-X_2-X_3 is again idle time, and normal instruction fetch resumes with the beginning of the next system cycle.

$\overline{\text{CPU RESET}}$

The $\overline{\text{CPU RESET}}$ pin on the processor enables an operator to restart the program at any time. Execution following the $\overline{\text{CPU RESET}}$ always begins at program memory location zero.

The $\overline{\text{CPU RESET}}$ is a negative-true input. Reset is effected by the momentary application of V_{DD} (GND, for the Central Processor Module) to pin 9 of the CPU.

To obtain a complete reset of the CPU chip, the reset level must be maintained for a minimum of eight full instruction cycles, or 64 periods of the external clock reference. Reset is effected by disabling the internal memory refresh circuitry momentarily, with the result that all dynamic memory elements (accumulator, index registers, program counter, instruction register) are filled with logic zero voltage levels. This refresh counter completes its scan of the chip every eight clock cycles, and a shorter $\overline{\text{CPU RESET}}$ would thus fail to clear the entire memory.

$\overline{\text{STOP}}$

$\overline{\text{STOP}}$ is a negative-true control input which permits the system operator, or an external device, to stop the execution of the program in progress.

Since the $\overline{\text{STOP}}$ input is asynchronous, the external command which clamps this line may originate at any time during the execution of a program. The CPU will finish executing the instruction in progress, then respond to the external request by clamping the $\overline{\text{STOP ACKNOWLEDGE}}$ status line. While the $\overline{\text{STOP ACKNOWLEDGE}}$ output is low, the processor remains in a pseudo-halt condition, in which the "no operation" command is executed repeatedly. The program counter is not incremented, and the program will therefore resume at the point of interruption whenever the $\overline{\text{STOP}}$ request is terminated.

Exit from the $\overline{\text{STOP ACKNOWLEDGE}}$ condition occurs automatically when the clamp on the $\overline{\text{STOP}}$ request line is lifted. A delay of up to eight clock periods may be necessary, to complete the NOP in progress.

The $\overline{\text{STOP}}$ input may also be used to re-start a processor which has been placed in the halted state by a programmed "halt" instruction (HLT). This is accomplished by clamping the $\overline{\text{STOP}}$ request line until a $\overline{\text{STOP ACKNOWLEDGE}}$ output is obtained. Once the status line is true (low), the machine may be re-started automatically by simply releasing the $\overline{\text{STOP}}$ request input. As before, several clock periods may elapse before the program is resumed.

$\overline{\text{INTERRUPT}}$

The $\overline{\text{INTERRUPT}}$ request line is a negative-true control which permits an external device to alter the normal program sequence, at a regular interval or intermittently.

The $\overline{\text{INTERRUPT}}$ request is an asynchronous input, which the CPU acknowledges upon completion of the instruction in progress. Recognition is indicated by a low at the processor's $\overline{\text{INTERRUPT ACKNOWLEDGE}}$ output. This output remains true until processing of the interrupt is finished, as indicated by the execution of a "branch-back and send register control" instruction (BBS).

When the CPU acknowledges an $\overline{\text{INTERRUPT}}$ request, it temporarily inhibits the routine incrementation of the program counter and executes an automatic subroutine jump to program memory address 003_H. In this way, the contents of the program counter are saved in the stack, and execution of the interrupted program will be resumed without loss of continuity when a "branch-back" is used to terminate the processing of the interrupt.

ROM location 003_H is a dedicated address where the first step of the interrupt service routine will always be stored. The service routine is a special purpose program designed to accommodate the interrupting device. Among other functions, this program may be required to (a) save and later restore the processor's internal status, (b) determine the identity of the requesting device and transfer control accordingly, and (c) return control eventually to the interrupted program. The exact requirements depend on the intended application. In most cases, the interrupt routine should save the following internal processor status values:

- Content of ACCUMULATOR and CARRY flip-flop
- Content of COMMAND REGISTER
- Content of as many INDEX REGISTERS as required
- The value of the last SCR address sent out prior to the interrupt
- The current ROM bank (CM-ROM_0 or CM-ROM_1)

These status values can be saved in the expanded index register array, as is done in the following example of an interrupt service routine:

EXAMPLE INTERRUPT SERVICE ROUTINE:

ROM Location Bank / Page / Word	Instruction	Comment
0 6 82	SRC 4	(IR 8, 9) sent to ROM & RAM, Load SRC Reg.
0 6 83	INC 9	Interrupt occurs here.
0 6 84	(JMS 0)	Interrupt acknowledged, 6, 84
0 6 84	(3)	saved in stack; instruction at 6, 84 ignored.
0 0 3	SB1	Select IR Bank 1
0 0 4	XCH 7	(ACC)→IR7* – ACC saved
0 0 5	LCR	(CR)→ACC
0 0 6	RAL	(CY)→Acc0, Acc_0→Acc_1 . . . Acc_3→CY
0 0 7	XCH 6	(ACC)→IR6* CY, CR saved
0 0 8		
0 0 9	.	Routine for determining and servicing interrupt
. .	.	is executed here
. .	.	
0 P n	XCH 6	(IR6*)→ACC
n+1	RAR	ACC_0→CY – CY restored
n+2	DCL	ACC_0→CR_0, ACC_1→CR_1, ACC_2→CR_2, CR restored
n+3	XCH 7	(IR7*)→ACC
n+4	SB0	Select IR Bank 0
n+5	BBS	Address 6, 84 loaded into PC; contents of SRC register sent out;
0 6 84	WRM	program restored.
	.	
	.	
	.	

If a suitable start-up routine is stored in location 003_H the $\overline{\text{INTERRUPT}}$ request line may be used to re-start a processor that has been halted by a programmed HLT. Start-up is automatic, once the request has been acknowledged. Note, however, that the $\overline{\text{INTERRUPT ACKNOWLEDGE}}$ output must be reset by execution of a programmed "branch-back," before the CPU will be able to respond to other incoming $\overline{\text{INTERRUPT}}$ requests.

$\overline{\text{TEST}}$

By clamping the processor's $\overline{\text{TEST}}$ input low, an external device sets a condition flag that the processor's internal control logic can test. The flag is maintained only as long as the control signal persists.

This enables the operator to program a conditional jump in the program sequence, contingent upon the level at the $\overline{\text{TEST}}$ input. Refer to Appendix A for the description and use of the "conditional jump" instruction (JCN).

PERIPHERAL LOGIC

The following section describes the peripheral logic elements on the Central Processor Module; that is, the blocks which support the functional requirements of the 4040 CPU. Functions to be discussed include:

- Clock Circuitry
- Memory
- Input/Output Provisions

The Central Processor Module schematic is shown in Figure 3-25, at the end of this chapter.

Clock Reference Oscillator And Timing Generator

The timing logic on the Central Processor Module consists of a crystal-controlled square wave oscillator, a divide-by-seven synchronous counter, decoding circuitry, and a TTL-to-MOS level shifter. This circuitry is used to derive two non-overlapping negative-going clock pulses, each of 386 nanoseconds' duration, at a nominal frequency of 740.7kHz. These timing signals, referred to as the $\overline{01}$ and $\overline{02}$ CLOCK, furnish the timing reference for all logical operations on the module.

The quartz crystal, Y1, is the principal frequency determining element. Cascaded TTL inverters, type 74H04, provide the power amplification and phase shift required to sustain oscillations. The resulting 5.185 MHz output is applied to a type 74H00 inverting gate, used here as a buffer, and directed to the clock pulse input of a Fairchild type 9316 synchronous decade counter. The positive-true out-

puts from the second and third counter stages are applied to another 74H00 NAND-gate, used to indicate coincidence on the sixth count following counter reset. The coincidence output is routed to the $\overline{PE}$ input of the 9316, enabling the seventh clock pulse to reset the counter to zero.

The second stage output of the counter, and opposite phases of the third stage output, are combined in two more 74H00 coincidence gates, to obtain the two non-overlapping clock phases. The TTL output of each of these gates is buffered, inverted, and applied to one input of an MH0026, used to obtain the MOS levels required for driving the processor and its peripheral logic elements.

Module Memory

Memory on the Central Processor Module is of two kinds, RAM memory or working storage, and ROM or program memory. This section provides detailed information on the elements used. The ROM memory is described first. A description of the 4002 RAM and Four-Bit Output Port will follow.

ROM MEMORY AND SIMULATOR LOGIC

Up to this point, our description of the Central Processor Module implies that ROM and RAM memory consists of monolithic elements which incorporate the addressing and I/O logic in the chips themselves, an implication that is only partially accurate. For reasons we shall shortly explain, this black box concept of memory is justifiable, since it greatly simplifies our initial description of how the module works.

Although the Central Processor Module may be used independently of the INTELLEC® 4/MOD 40 system, often to considerable advantage, it was designed primarily as part of that system. The basic purpose of the INTELLEC® 4/MOD 40 system is to simplify the development of OEM microcomputers, compact, dedicated machines that use components of Intel's MCS-40 Microcomputer Set. The INTELLEC® 4/MOD 40 system is designed to model such a system, prior to finalizing of the design and its firmware package.

The MCS-40 Microcomputer Set contains at least one 4040 Central Processor Unit (monolithic CPU). It will also contain one or more of the following components:

4001 256 x 8-Bit ROM and 4-Bit I/O Port
4308 1024 x 8-Bit ROM and 4 I/O Ports (4-Bits/Port)
4002 80 x 4-Bit RAM and 4-Bit Output Port

The 4001 and 4308 ROM and the 4002 RAM correspond functionally to the ROM and the RAM elements that we have so far been describing. Four 4002 RAMs are, in fact, used for working storage on the CPU module. No 4001s or 4308s are used, however, anywhere in the INTELLEC 4/MOD 40.

The 4001 and 4308 elements are production-oriented devices. They are programmed during manufacture, to the user's specifications. The 4001 and 4308 ROMs are extremely useful, compact and economical components. Unfortunately, the same traits that are virtues in production are liabilities in the development phase of a project. It is extremely difficult, expensive, and time-consuming to detect and correct program errors in ROM.

For this reason, ROM memory is simulated in the INTELLEC® 4/MOD 40 system using a memory element known as a PROM. PROMs are Intel 4702A Programmable Read Only Memories. The 4702A is a read only memory with a 256 X 8-bit capacity, similar to the 4001's basic storage capacity. The most important difference is that the 4702A is programmed electrically, by high amplitude pulses, rather than by metal mask, in manufacture. It is thus feasible for users themselves to program these devices, eliminating the longer lead time required when ordering 4001s. Moreover, the contents of the 4702A are readily erasable by exposure to high intensity ultraviolet radiation, permitting the user to erase and re-program them as often as is necessary in the course of developing a firmware program. They are useful for simulating the behavior of a mask programmed ROM.

There are other important differences, however, between the ROM and the PROM:

a) The PROM has no input or output port provisions.

b) The PROM is not page addressable, as is the ROM. It does not recognize itself as one of sixteen elements in a ROM array. It has only a single line, $\overline{CS_0}$, which enables or disables its function.

c) The PROM and ROM are accessed differently. The ROM has a single four-line interface to the main data bus. This main data port serves both as the point of entry for multiple byte addresses and as the exit for the memory's reply. The 4702A, on the other hand, has separate ports for addressing and reply. Moreover, the PROM contains none of the internal multiplexing logic of the ROM. It responds only to eight-bit parallel addressing, by placing an eight-bit reply on its output lines.

Accordingly, the addressing, multiplexing, and I/O logic of the 4001 ROM must be simulated on the Central Processor Module, using other discrete and monolithic components. In this way, the processor sees a "virtual" 4001 ROM memory, as accurate modeling requires.

Figure 2-11 shows the details of program memory and I/O bussing on the Central Processor Module. The module employs a type 4289 memory interface to simulate the multiplexing logic of the 4001 ROM.

The Intel 4289 controls the address multiplexing functions. The 4289 communicates with the main data bus, through a four-line input port.

A $\overline{SYNC}$ signal from the processor, and the $\overline{01}$ and $\overline{02}$ CLOCK reference signals, are also applied to the 4289. The interface uses the timing signals to coordinate the assembly of the three program address bytes, during the A_1-A_2-A_3

phase of the system cycle. The two low-order bytes are applied in parallel to an eight-line address bus connecting the inputs of all four 4702A PROMs on the module.

The most significant byte of the address, the four bits transmitted in A_3, is directed in parallel to the module's chip select bus where it is used to drive the chip select decoder. An OR-gate section (A27-1-2-3) and one section of a dual 2-to-4 demultiplexer (A18), perform the decoding function. The demultiplexer's one-to-four output is enabled by a coincidence of the $\overline{\text{ENABLE MON PROM}}$ signal applied to pin #14 of the demultiplexer and simultaneous lows on the C_2 and C_3 chip select lines, indicating a program page address in the range of #0 to #3. The demultiplexer's selected output line is instrumental in enabling one of the four PROMs on the CPU module.

The selected output line from the 74155 demultiplexer, O_0-O_3, enables one page in program memory. Each line is applied to the chip select input ($\overline{CS_0}$) of one of the four PROMs.

The 4289 interface element disassembles the contents of the eight-bit PROM reply bus. The contents of the specified PROM cell are returned to the module's main data bus, as two sequential four-bit bytes, during M_1 and M_2 of the system cycle.

The 4289 also incorporates bus switching logic which enables it to sense when an I/O instruction is in progress. This is much the same function performed internally by the 4001 ROM when its input or output port is accessed. When an SRC is executed, the 4289 latches the X_2 page address into internal storage elements, forwarding this code to the chip select bus during each succeeding X_2 period. Any subsequent I/O command is detected by the 4289's instruction decoder, causing the 4289 to effect connection between the I/O bus and the main data bus.

Input ports are simulated on the module, using a pair of dual 4-to-1 multiplexers (type 8214). Each multiplexer utilizes the C_0 and C_1 lines from the chip select bus to switch one of four input lines to the I/O bus input gate.

Output ports are simulated using four 3404 hex inverting latches. Four latches on each chip are used to implement one four-line output port. All data inputs on the latches are tied in common to the I/O bus.

Chip selection for the output ports is obtained through the coincidence of the $\overline{\text{OUT}}$ strobe and the four-line chip select output from the 4289.

This function is performed in the second stage of the 74155 demultiplexer, A18. Here the coincidence of an inverted $\overline{\text{OUT}}$ strobe and simultaneous lows on both the C_2 and C_3 lines enables the demultiplexer's remaining decoder section, producing a one-of-four output that is

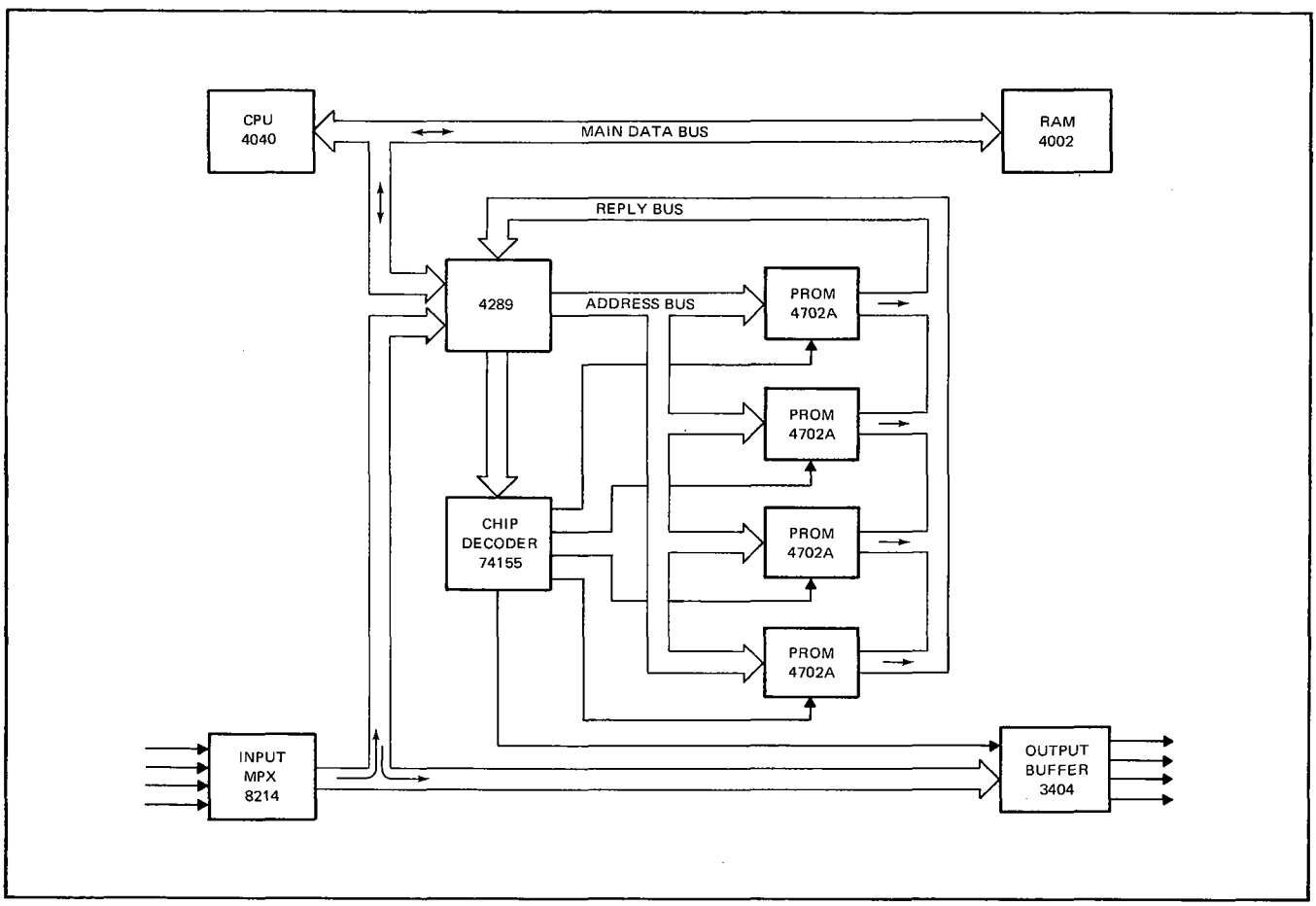


Figure 2-11. Central Processor Module Data Bussing

determined by the combined states existing on the C_0 and C_1 chip select lines. The selected output is then used to strobe the appropriate ROM output latch.

The 4289 programs an input or an output operation by means of two control lines, the $\overline{IN}$ and the $\overline{OUT}$. One of these lines will be driven true during X_2, depending upon the kind of instruction code that the 4289 receives. If the $\overline{IN}$ line is pulsed, then data at the selected input port passes through the I/O bus input gate, through the bus switching logic of the 4289, to the module's main data bus. Should the $\overline{OUT}$ line be pulsed instead, then data on the main bus will pass through the 4289 in the other direction, to be strobed into the output latches.

RAM MEMORY

The 4002 RAM memory element is used for working storage on the Central Processor Module. The module incorporates four of these chips.

The 4002 is a 320-bit dynamic storage element, organized as an 80 X 4-bit array. The 80 bytes are further specified as belonging to one of four registers within the chip. Each register contains 20 bytes.

Sixteen bytes in each register are numerically addressable. The remaining four bytes in the register are known as status characters, and are addressable by means of special commands in the processor chip's I/O and RAM instruction group. Status character read and write instructions are described in Appendix A.

The RAM is also equipped with a four-line output port which includes latch and buffer. The logic required for the interpretation and execution of output instructions is provided within the chip.

Figure 2-12 shows the internal organization of the 4002. The RAM communicates with its associated processor by means of an internal four-line data bus, which is a simple extension of the bus shared externally by the processor. A bus multiplexing section controls access to the bus internally, in response to signals from the main control logic section. The control logic, in turn, receives the $\overline{01}$ and $\overline{02}$ CLOCK signals from the external timing generator, and the $\overline{SYNC}$ reference from the processor itself, and uses these signals to coordinate the logical functions within the chip.

The remaining functional blocks are the actual memory cell arrays, the instruction and address decoding circuitry, and the output port latch.

All RAM operations have three distinct phases, and normally require at least three corresponding program

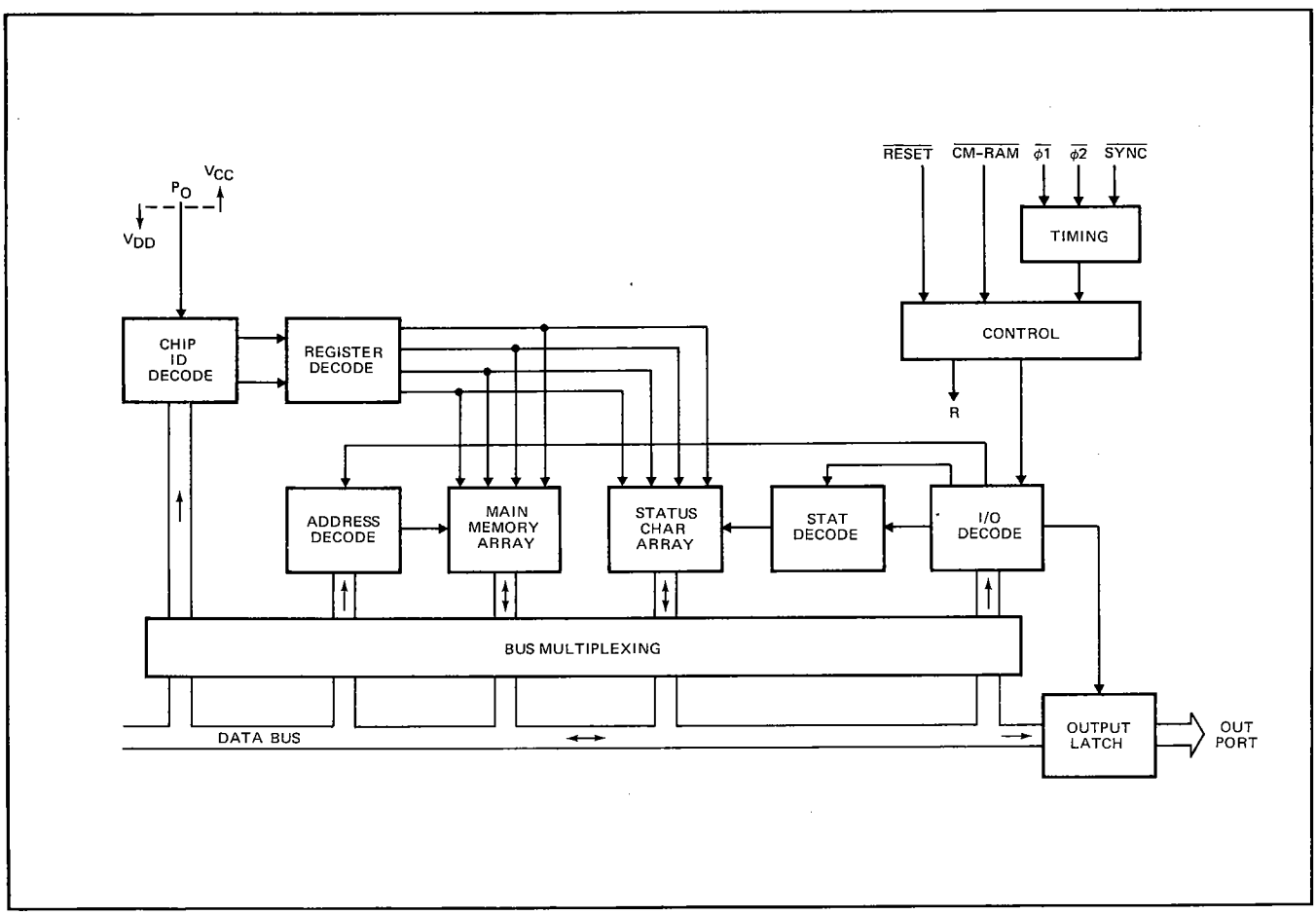


Figure 2-12. 4002 RAM Functional Block Diagram

steps. These phases are:

a) command line enabling
b) addressing
c) execution

A typical read or write sequence is as follows:

First, the $\overline{\text{CM-RAM}}$ line must be enabled, a function normally delegated to the processor with which the RAM is associated. In a typical interface scheme, such as the one used in the imm4-43, the application of an MOS low on the command line selects a bank of four 4002s. This line will enable the RAMs control logic only at certain times in the system cycle. The designated command line is enabled, for example, when an SRC instruction is executed during X_2 and X_3. The line would also be enabled during M_2 of any cycle in which an I/O operation was indicated, permitting the RAM to receive the OPA portion of the instruction code.

Next, the chip must be addressed. An eight-bit address field is used for this purpose. The first two bits of the address identify one of four chips sharing the memory command line. The second two bits specify one of the four internal registers in the chip. And the remaining four bits designate one of sixteen numerically addressable four-bit cells within that register.

The chip is normally addressed through the processor, by execution of an SRC instruction. In a case such as this, the processor places two bytes on the main data bus, at times X_2 and X_3 of the SRC instruction cycle. The processor simultaneously enables the previously designated $\overline{\text{CM-RAM}}$ line, and the control logic of all chips on the enabled line instructs the multiplex sections to gate the first byte to the chip ID decoders. The second byte is gated through the multiplex logic and stored in the address decoder section of the chips on the enabled line.

One of the four chips on the line recognizes and responds to the identification field contained in the first two digits of the X_2 byte. The chip's identity is established in two ways. First, the chip is programmed in manufacture, and thereafter identified as a 4002-1 or as a 4002-2. Next, the user has the option of wiring the P_0 input (pin 10) either to V_{CC} (+5 VDC) or to V_{DD} (−10 VDC) at the time the element is installed. There are thus four possible ID combinations, as summarized in Table 2-1.

4002 Chip Identification

Chip No.	Binary Code	4002 Option	P_0
0	00	4002-1	V_{SS} (+5)
1	01	4002-1	V_{DD} (−10)
2	10	4002-2	V_{SS} (+5)
3	11	4002-2	V_{DD} (−10)

Table 2-1.

The chip specified by the ID field gates the remaining two digits in the first byte to the register decoder logic within the chip. The decoder logic is a 2-to-4 line converter, using these two bits to enable and drive one of four register selection lines. The chosen line enables one register in the main memory array and one in the status character array.

Thus, at the end of an SRC instruction cycle, one RAM chip will contain both a cell address (the 4 bits of the X_3 byte) and an enabled register. Conditions are now set for the execution of a subsequent I/O or RAM instruction.

An instruction in the I/O and RAM group is uniquely identified by the M_1 OPR byte (1110). Any instruction so tagged causes the processor to enable the $\overline{\text{CM-RAM}}$ command line, during M_2. The bank of memory chips thus alerted generates a control signal, enabling the multiplex logic to gate the M_2 instruction byte to its I/O decoder section.

At time X_2, the processor again enables the command line of the selected RAM bank. The control logic enables the I/O decoding section as a result, causing the decoder to initiate one of three kinds of operations:

a) main memory
b) status character
c) output

depending upon the operation code stored in the decoder.

Should a main memory operation be indicated, the decoder signals the address decoding logic of the Main Memory array. The multiplexing logic is enabled, and data is gated into or out of the array, to or from the bus, as required by the instruction.

In the event of a status character transaction, the I/O decoder forwards the two least significant bits of the OPR to the status character decoder. The intersection of lines selected by the register decoder and the status character decoder determines a single cell within the status character array. The multiplexing logic is appropriately gated, and the selected cell is read or written, as required.

If the I/O decoder senses an output operation, it simply pulses the output latch. Data on the internal bus is then latched up on the output lines, to complete the instruction cycle.

The $\overline{\text{RESET}}$ input (pin 9) permits an external signal to clear the RAM array. The $\overline{\text{RESET}}$ line is a negative-true input, at MOS logic levels. A low level must be maintained for 32 full instruction cycles, in order to completely clear the memory, allowing the internal refresh circuitry to scan and store zeros in all memory cells.

Teletype Interface

The Central Processor Module contains a built-in teletype interface which enables the input and output of data via an ASR 33 teletype set. Three interface circuits are provided: the receiver circuit ($\overline{\text{TTY IN}}$), the transmitter circuit ($\overline{\text{TTY PRINTER}}$), and the reader control circuit.

In order to understand how the interface operates, one must know something about the teletype itself. A Model ASR 33 teletype set consists of:

a) a keyboard unit
b) a printer unit
c) a tape punch unit
d) a tape reader unit
e) an electrical service unit

The keyboard unit is essentially a transmitter, and the printer unit is essentially a receiver. The keyboard originates messages, by causing a time-dependent series of current interruptions, in a DC loop connecting the keyboard and one or more printer units. The printer shares a loop, in series with one or more keyboards, printing a message in response to the pattern of interruptions in the loop current. An interruption originating at any keyboard within a loop causes a response at every printer within that loop.

There are two ways of connecting a pair of teletype sets, half duplex (HDX) and full duplex (FDX). In the half duplex configuration, a single loop is used. This loop passes through both the keyboard unit and the printer unit of each teletype set. A message originating at either keyboard therefore causes both printers to respond. In a half duplex configuration, only one terminal can transmit at any given time. Chaos would result if two or more keyboards attempted to transmit messages simultaneously.

The full duplex hookup uses two current loops. The first loop links the keyboard of one terminal with the printer of the second. The other loop connects the second terminal's keyboard with the printer of the first set. In this configuration, the keyboard of each set is independent of its associated printer. Thus the terminal can receive a message on its printer while simultaneously using its keyboard for transmission.

Although it is possible to use the half duplex mode for data processing applications, this is seldom done in practice. The principal advantage in using full duplex is that the receiving terminal can "echo" the transmitter; that is, transmit the received message back to the originating terminal after a short processing delay. The delay involved is generally so brief that the terminal operator is never aware of the fact that he is working in full duplex. His printer unit will apparently respond to the keyboard just as rapidly as it would in the half duplex mode, but any transmission error or hardware malfunction will promptly reveal itself in the form of a garbled printout. This affords a highly desirable check on the operation of the entire system. The teletype interface on the Central Processor Module is of this "echoplex" type.

The tape punch unit is associated with the printer unit of the teletype set. When enabled, it punches a paper tape record of all messages received by the printer.

The tape reader unit is associated with the keyboard unit of the teletype set. The reader accepts a 1-inch paper tape, punched in ASCII code, and produces a corresponding series of current pulses in the loop it shares with the keyboard. A sprocket wheel on the reader advances the tape automatically, at the maximum TTY rate of 10 characters per second.

The electrical service unit is essentially a junction box, interfacing the other units to the external transmit-receive loops.

Figure 2-13 shows the functional interrelationship of the five major units in the teletype set. Observe that the mode of operation is established by shifting jumpers on a terminal block within the electrical service unit. Connecting a jumper between terminals C and D places the printer/punch in series with the keyboard/reader, enabling operation in the half duplex mode. Terminals A and F then become the points of connection for the external section of the current loop.

By deleting jumper C-D, and substituting jumpers B-C and D-E, we obtain full duplex operation. The printer and punch unit become part of the receive loop, established through terminals A and B. The transmit loop, containing the keyboard and reader units, connects to terminals E and F. (Terminals A-F are illustrative examples, only, and do not correspond to actual terminals within the electrical service unit). Page 30 describes installation procedures, including the wiring required for HDX or FDX operation.

Refer to Figure 2-25.

Transistor Q5 and its associated circuitry make up the teletype receiver section of the Central Processor Module. The input loop is established by current flowing from the +5 Volt supply, through an external 220-ohm resistor, the keyboard of the teletype set, and R23 on the Central Processor Module, to the -10 Volt supply. The schematic of the module, Figure 2-25, shows the input loop. Note that the 220-ohm dropping resistor is incorporated into the mother board, in the INTELLEC® 4/MOD 40 system. Other users of this module must make a separate provision.

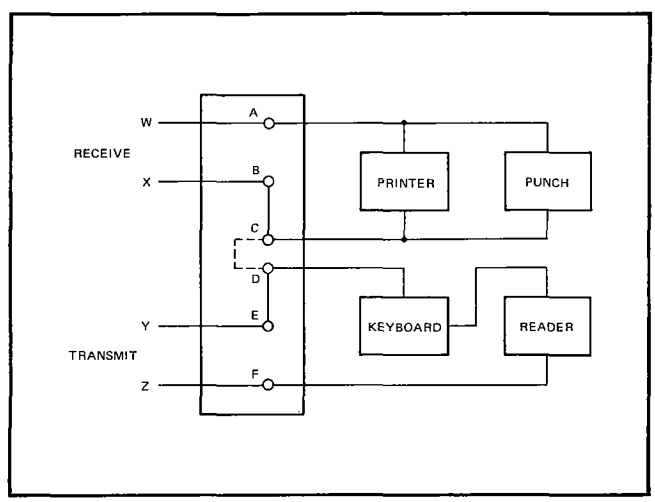


Figure 2-13. TTY: Half Duplex/Full Duplex

Any disruption of the loop current cuts off Q5, and the resulting signal variations at its collector are applied to A25-10, which is also data line 0 of the ROM 0 input port. Thus the processor can detect any signal fluctuations in the input loop, by sampling the level on this data line.

Current in the TTL PRINTER loop flows from the +5 Volt source, through the emitter-collector circuit of Q4 and through R18 of the Central Processor Module, through the printer unit of the Teletype Set, and through an external 270-ohm limiting resistor, returning to the -10 Volt supply (see Figure 2-22). Note that the 270-ohm current limiter is incorporated into the mother board of the INTELLEC® 4/MOD 40 system. When the module is used outside the system, that resistor must be provided separately.

Observe that the open-circuit loop voltage and the resistances within the loop establish a loop current of approximately 32 mA, when transistor Q4 is saturated. Signals applied to the base of Q4 result in signal fluctuations in the output loop. The base of the transistor is tied to data line 0 of RAM output port 0. By writing data into RAM 0's output latches, the processor can thus control the teleprinter link.

The teletype set's printer contains a current option provision, enabling it to function on a nominal loop current of either 60 mA or 20 mA. In the 20 mA mode, any loop current in excess of 13 mA will assure reliable operation of the printer. The user makes the selection of loop current at the time the set is installed, by shifting the position of a wire jumper in the set's electrical service unit. Since the factory ships the Teletype Model 33 wired for 60 mA operation, users of the Central Processor Module must convert the machine to 20 mA operation, before connecting the terminal into their system. Page 23 describes the installation procedure.

The teletype set's tape reader unit is often used as a convenient way to feed large volumes of data into a processing machine. The data may be prepared off-line; that is, with the teletype disconnected from the processor, by using the punch unit to prepare a paper tape. A check of the printer's output enables the operator to correct any errors, before feeding the tape into the reader. Once the tape is prepared, the operator inserts the leader into the tape reader and pushes the lever that starts the unit.

The reader operates automatically, once it is started, at the maximum possible transfer rate. Here, however, a hitch may develop. Segments of data on the tape may be interspersed with periods when processing action is required. If the processor were to execute these functions while the tape continued to run, it is possible that several characters might be lost irretrievably, by the time the processor returned to the I/O routine. To prevent this, the processor must have some means of controlling the tape reader: of stopping the reader intermittently, and of starting it up again. That is the purpose of the reader control section of the module's teletype interface.

The tape reader advances when 115 VAC is applied to its distributor trip magnet. In order to provide the processor with some means of controlling the reader, a relay must be inserted in series with this circuit. The processor then controls the reader, by controlling the power applied to the relay's pull-in coil. Page 23 contains a full description of the modifications required.

As the module schematic shows, the solenoid of the reader control relay is in series with Q3 on the Central Processor Module. The entire circuit is connected between the +5 VDC and the -10 VDC power supplies, so that the signal applied to the base of Q3 controls the current in the solenoid. Q3's base is connected to data line 0 of RAM output port 1, and the processor controls the reader's advance by writing into the latches of the port. Writing a "0" on that line stops the reader, while a "1" enables it.

Observe also, that the reader control loop includes a 270-ohm current limiting resistor which is external to the module. The mother board of the INTELLEC® 4 system already contains such a resistor. Other applications of the imm4-43 will have to make separate provisions.

The physical provisions of the teletype interface are only half the story. They enable electrical communication between the processor and the TTY terminal, but they do not provide for the logic of such transfers. That is properly a part of the programming function. A full discussion of I/O programming is beyond the scope of this manual, but a brief logical description of a typical input sequence is given on page 23. Those faced with the task of programming such a routine should refer to the INTELLEC® 4 Programming Manual, or to the MCS-40™ User's Manual, for detailed directions.

UTILIZATION

This section provides information on utilization of the imm4-43, for those who contemplate using the module outside the INTELLEC® 4/MOD 40 system.

Installation

In installing the Central Processor Module, the user must take account of:

- Environmental Extremes
- Mounting
- Electrical Connections
- Power Requirements
- Signal Requirements

ENVIRONMENT

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conductive sources of heat. Remember that the module itself when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the

convective dissipation of heat from the elements on the card.

Relative humidity is not critical to the module's operation.

MOUNTING

Avoid locating the card near vibrating machinery. Physical damage to the card is less likely than is the possibility that the 4702A PROM memory elements may be loosened in their mounting sockets.

Dimensions of the module are 6.18 X 8.00 inches. Be sure to allow enough additional clearance to ensure cooling.

The card is designed to plug directly into a standard 100-pin, double-sided PC edge connector.

The connector will serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the card be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the card, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the elements on the module.

ELECTRICAL CONNECTION

All connections to the Central Processor Module are effected by means of a standard 100-pin, double-sided PC edge connector. CDC #VPB 01 C50E00A1 is one suitable type. Pin allocations are given.

POWER REQUIREMENTS

The central Processor Module requires DC power, at the following levels:

+5V ± 5% @	1.2	Amperes (max
	0.8	Amperes (typ)
-10V ± 5% @	0.2	Amperes (max)
	0.1	Amperes (typ)

Refer to the pin list for the power connections.

SIGNAL REQUIREMENTS

No controlling signals, other than input data, are required for the operation of the module. The input/output relationship is entirely under the control of the internal program.

It is necessary, however, to strap pin 42 to power ground whenever the card is used outside the INTELLEC 4/MOD 40 system. This enables the program memory. The same pin would normally be enabled by logic within the memory controller board, when the Central Processor Module is a part of the system.

Five kinds of control are provided, for applications requiring them. They are:

a) $\overline{\text{CPU RESET}}$
b) $\overline{\text{4002 RESET}}$
c) $\overline{\text{TEST}}$
d) $\overline{\text{STOP}}$
e) $\overline{\text{INTERRUPT}}$

All are TTL level signal inputs. All are negative-true. Refer to page 30 for the destination pins.

The $\overline{\text{CPU RESET}}$ may be used to clear all working registers in the processor chip, and to reset the processor's program counter to zero. A minimum pulse width of 90 μsec is required.

The $\overline{\text{4002 RESET}}$ clears all data in RAM memory storage, including the memory cells, and the addressing registers. A minimum pulse width of 350 microseconds is required.

The $\overline{\text{TEST}}$ input is used to control the sequence of program execution, by enabling a tab that can be tested by the conditional jump instruction (JCN). Refer to Appendix A for explanation of the JCN.

Use of the $\overline{\text{STOP}}$ and $\overline{\text{INTERRUPT}}$ controls is discussed in detail.

ROM input port 1 is not used as an input port in the INTELLEC® 4/MOD 40 system because the many intra-system signals use all of the available pins on the Central Processor Module. When the module is used outside of the INTELLEC® system, however, ROM input port 1 can be used. To enable the port, delete jumper wire connections 1-2, 5-6, 9-10, 12-13, and 15-16; instead, connect jumpers 1-3, 6-7, 8-10, 11-13 and 14-16, respectively (see Note 4 on the CPU module schematic, Figure 2-25).

Memory Expansion Provisions

The basic imm4-43 module has a program memory capacity of 1K x 8 bits, and a working memory capacity of 320 x 4 bits. The module contains provisions for expanding this capacity.

PROGRAM MEMORY

Program memory may be expanded, to a maximum of 4K x 8, using Intel 1702A PROMs. Each PROM has an inherent capacity of 256 x 8. Eight address and eight data return lines are provided on the Central Processor Module, to facilitate any desired expansion. Refer to the pin list for line allocations.

Chip selection is obtained by decoding the chip select lines (C_0 – C_3) out of the module. A random logic gate network will have to be provided for this purpose. Section entitled Module Memory explains the method of addressing the chips.

A PROM Memory Module (imm6-26) and an Instruction/Data Storage Module (imm4-24) are available from Intel. Both contain additional PROM elements. These modules incorporate all the required chip addressing logic right on the cards. If a modular expansion of the memory is feasible, the addition of these boards will probably prove to be the most economical means of enlarging the imm4-43's

program memory. Refer to Chapter 1 of the INTELLEC® 4/MOD 40 Reference Manual for a description of these modules.

RAM MEMORY

The Central Processor Module's 4002 RAM memory may also be expanded, by adding external elements. The upper limit to data RAM storage is 2560 x 4 bits. Additional elements are connected in parallel to the main data bus, receiving $\overline{SYNC}$ and $\overline{01}$ and $\overline{02}$ timing signals from the outputs provided on the imm4-43. The connector pin allocations of these signals are given.

One of the $\overline{CM\text{-}RAM}$ lines, 1-3, is used to command every bank of four RAMs added to the module's basic capacity. Each bank will consist of two 4002-1 and two 4002-2 elements, with each identical pair having alternating wiring of the P_0 identity pin.

If more than three additional banks of RAMs are added (12 RAM elements), then the three memory command lines, $\overline{CM\text{-}RAM}_1$ – $\overline{CM\text{-}RAM}_3$, will have to be decoded in order to provide the number of control lines required. A maximum of eight command lines may be developed from the $\overline{CM\text{-}RAM}_1$ – $\overline{CM\text{-}RAM}_3$ signals. An Intel 3205 three-to-eight line decoder would be one suitable means of accomplishing the necessary conversion.

The imm4-22 Instruction/Data Storage Module and the imm4-24 Data Storage Module contain additional RAM storage. Both incorporate the necessary command line decoding logic into the modules themselves. This eliminates the necessity of providing that logic separately. As in the case of program memory, a modular expansion is often the most convenient and economical. Refer to Chapter 1 of the INTELLEC® 4/MOD 40 Reference Manual, for a description of these modules.

Input/Output Expansion

The imm4-43 Central Processor Module is equipped with four 4-line input ports and eight 4-line output ports. There are several ways of expanding the I/O capacity of the basic module:

a) implementing ports in random logic
b) the addition of RAM memory elements
c) the addition of memory modules
d) the addition of I/O modules

Both input and otuput ports can be implemented in random logic, using the Central Processor Module's I/O bus as the point of connection. Port selection addressing is obtained by decoding of the C_0 – C_3 chip select lines on the module. The $\overline{IN}$ and $\overline{OUT}$ lines from the board provide the signals necessary for developing the port strobes.

As in the case of memory expansion, however, it is often more practical to implement additional ports by using modules. The imm4-60 Input/Output Module is available for that purpose. This card provides eight input and eight output ports, together with all the requisite addressing logic. The Input/Output Module is described more fully in Chapter 1 of the INTELLEC®4/MOD 40 Reference Manual.

Several of the modules available for expanding the system memory also contain additional input and output provisions.

The imm4-22 Instruction/Data Storage Module incorporates four 4-line input ports and eight 4-line output ports.

The imm4-24 Data Storage Module provides 16 additional output ports. One port is associated with each RAM element on the module.

The Central Processor Module is capable of addressing a maximum of 16 input and 48 output ports, in any combination of the foregoing provisions.

Program RAM

The Central Processor Module is also equipped for interface with the imm6-28 RAM Memory Module. The imm6-28 provides 4K x 8 bits of read/write memory, using Intel® 2102 RAM memory elements. Control program RAM should not be equated with 4002 data RAM storage.

The 2102 RAM element is a 1K x 1 array. Eight such elements are arranged in parallel, to form each 1K x 8 block of program RAM.

The imm6-28 is designed to substitute for semi-permanent PROM storage, in those applications where a larger and more flexible program memory is required. Like ordinary program memory, the imm6-28 can be addressed during the $A_1 - A_2 - A_3$ phase of the system cycle. It will respond by transmitting the contents of the requested location during M_1 and M_2 of the cycle. In this mode, it simulates the operation of 4702A PROM memory.

Special instructions, however, permit read and write operations during the execution phases of the cycle. These instructions are used to analyze and reload the contents of program memory. Operations of this kind are difficult, or impossible, using ROM or PROM elements.

The PM and the $F/\overline{L}$ control lines on the Central Processor Module are part of the interface provisions for the imm6-28. The program RAM makes use of the MEMORY ADDRESS and the MEMORY DATA IN lines as well. The operation of the interface is fully described in that section of the INTELLEC® 4/MOD 40 Reference Manual which pertains to the imm6-28 RAM Memory Module and to the imm4-72 Control Module.

MODIFICATION OF THE TELETYPE SET

Minor modifications to the ASR 33 will be necessary, before the set can be used with the Central Processor Module. These include:

- Conversion from factory-wired operation, at 60 mA, to a 20 mA loop current.
- Conversion from factory-wired operation, in the

half duplex mode, to full duplex.

- Provision for external control of the tape reader drive.

20 MILLIAMPERE LOOP CONVERSION

Proceed as follows, to prepare the ASR 33 for operation in a 20 mA current loop:

(1) Change the current source resistor in the electrical service unit of the teletype, from 750 ohms to 1450 ohms. The current source resistor is a tapped resistance, and conversion is accomplished by shifting the position of a single wire lead. "Kwik-Connect" push-ons are used, and no soldering is required. Figure 2-14 shows the location of the resistor. The physical arrangement of the taps is diagrammed in Figure 2-15.

(2) Change the position of a single wire on the barrier terminal strip at the rear of the electrical service unit. A screwdriver will be necessary. Figure 2-16 shows the location of the terminal strip, and Figure 2-15 shows the electrical change required. The violet wire attached to terminal #8 should be moved to terminal #9.

CONVERSION TO FULL DUPLEX OPERATION

Conversion to full duplex is accomplished simply, by shifting the position of two wires on the barrier terminal strip at the rear of the electrical service unit. A screwdriver will be required. Figure 2-16 shows the location of the terminal strip, and Figure 2-17 shows the electrical change to be accomplished. Broken lines in the diagram indicate the half duplex hookup. Solid lines indicate the wiring required for full duplex operation.

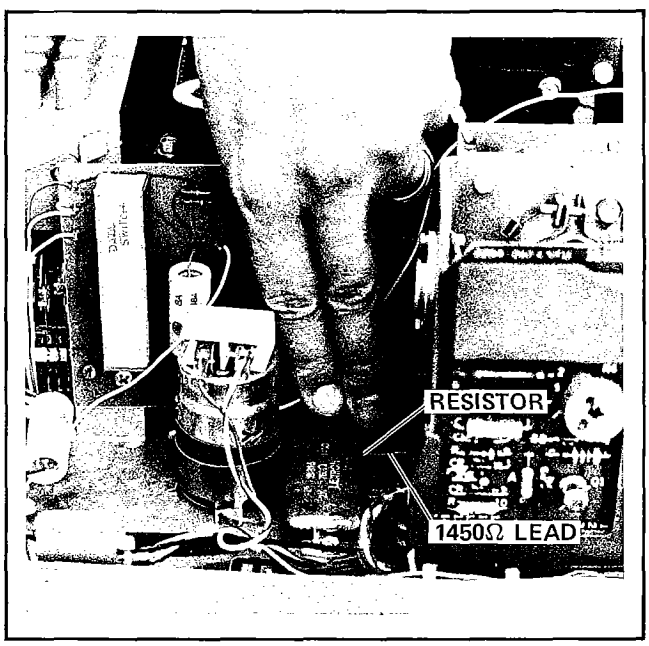


Figure 2-14. Current Source Resistor

(1) Move the blue/white lead from terminal #4 to terminal #5 on the strip.

(2) Move the brown/yellow lead from terminal #3 to terminal #5.

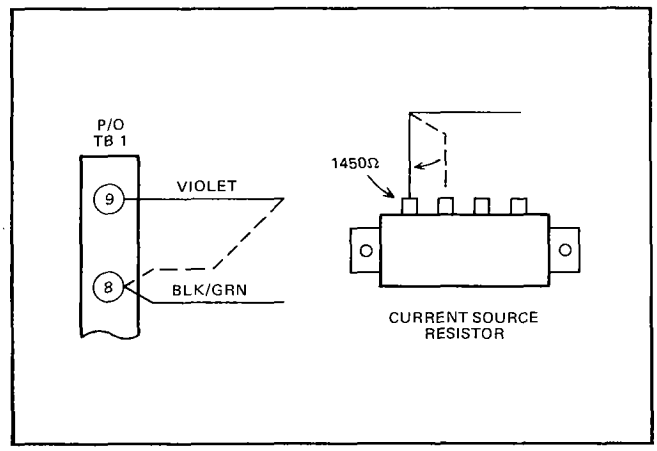


Figure 2-15. TTY: 20 mA Wiring

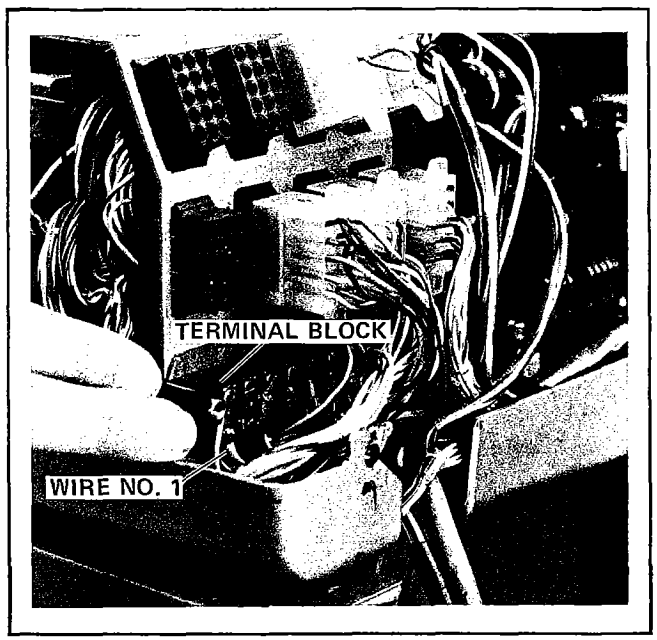


Figure 2-16. Terminal Block Location

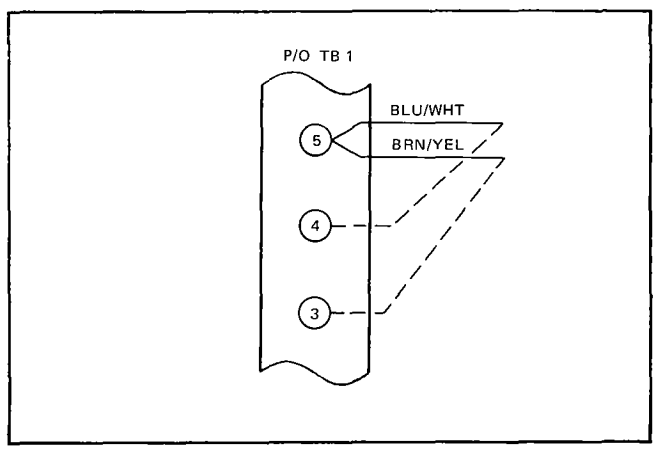


Figure 2-17. TTY: Full Duplex Wiring

READER DRIVE MODIFICATIONS

In order to provide control of the tape reader drive, you must install a relay circuit in the electrical service unit of the teletype set.

Construction of the circuit is simple. In addition to wire and a small piece of vector board, you will need the following parts:

miniature relay (Potter-Brumfield #JR-1005)	1 each
diode IN914	1 each
thyracter (GE 6RS20-5P4B4)	1 each

These parts will be wired as shown in Figure 2-18.

Figure 2-19 shows how to locate and mount the auxiliary circuit in the teletype's electrical service unit, on the vertical metal tab which is just forward of the large filter capacitor. Drill the circuit board in two places. Then drill and tap the tab. Mount the auxiliary board using two machine screws.

The auxiliary circuit board is wired into the electrical service unit in three steps:

(1) Remove the brown wire from the molex plug (see Figure 2-20 for location of the wire). Splice two leads into the disconnected brown wire.

(2) Connect one of these leads (labeled wire "A" in Figure 2-18) to one side of the Potter & Brumfield relay contact pair.

(3) Connect the remaining lead (labeled "LOCAL" in Figure 2-18) to terminal L2 of the LOCAL/LINE selector switch. Connect the other contact of the Potter & Brumfield relay to terminal L1 of the LOCAL/LINE selector switch. Figure 2-21 shows how to identify the switch terminals.

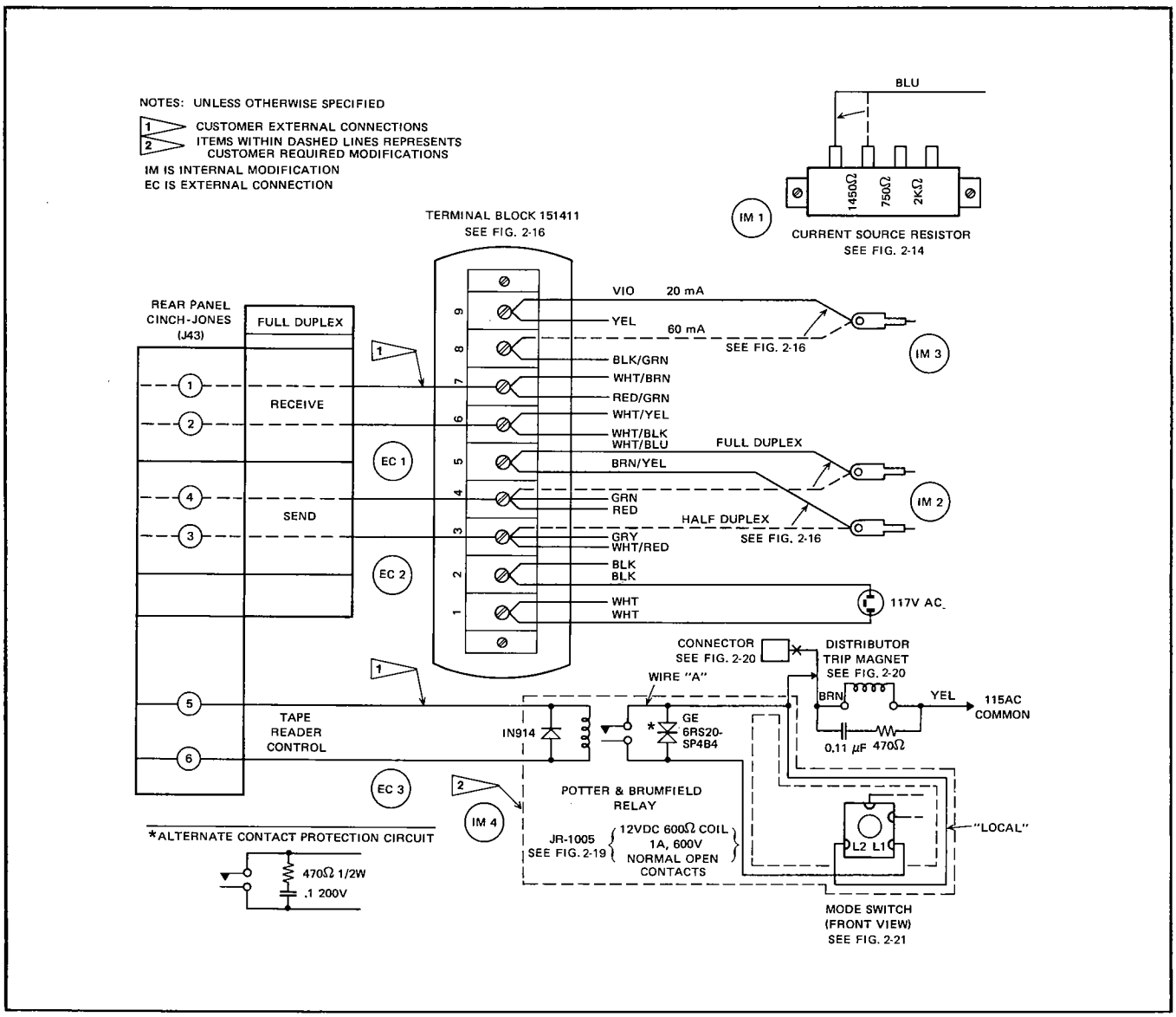


Figure 2-18. TTY Modifications

INTERCONNECTION

Interconnection of the Central Processor Module and the teletype set requires three signal loops (see Figure 2-22):

a) a send loop
b) a receive loop
c) a reader control loop

The send and the receive loops are connected between the module and the a barrier terminal strip at the rear of the electrical service unit.

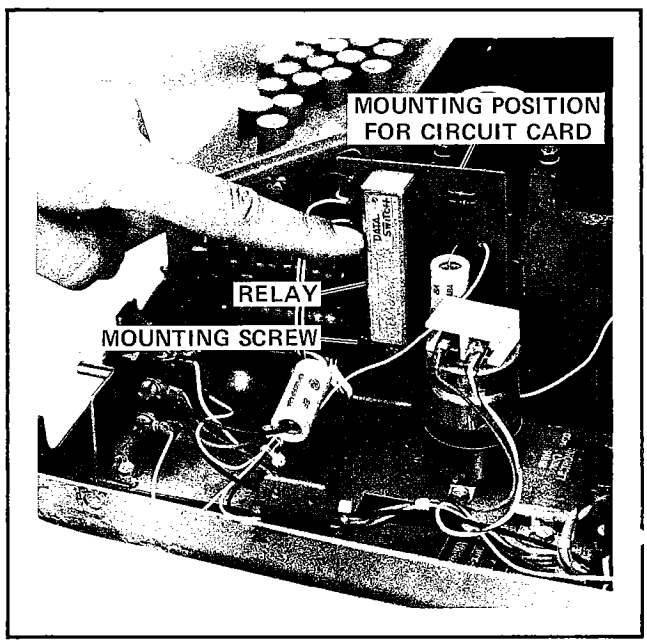


Figure 2-19. Relay Board Location

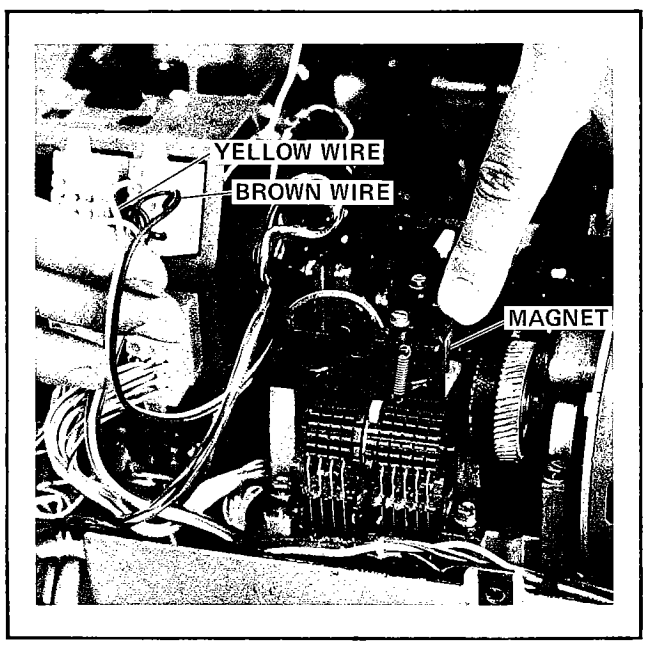


Figure 2-20. ESU Wiring

The reader control loop is connected between the module and the solenoid of the auxiliary relay circuit in the ESU.

All connections are shown in Figure 2-22. No special provisions are required.

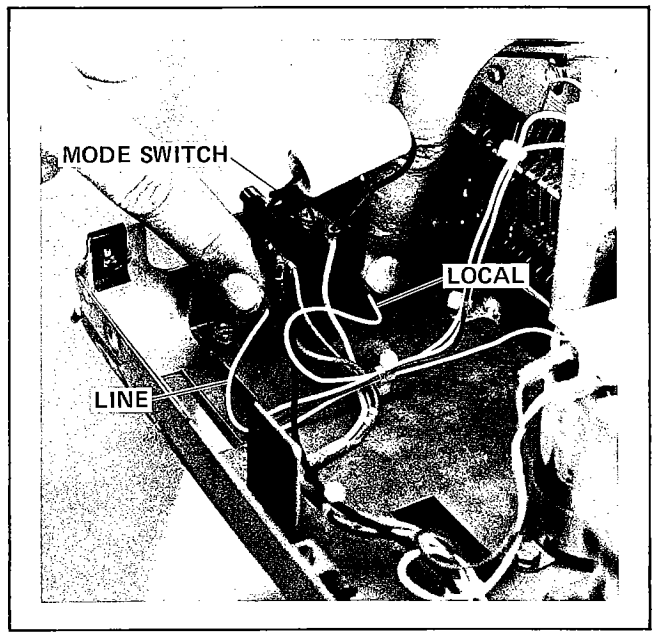


Figure 2-21. Teletype Mode Switch

I/O PROGRAMMING

Interconnection of the CPU module and the teletype is only part of the interface. Connection enables electrical communication between processor and TTY, but it does not provide for the logic of such transfers. A software routine is required which is designed to handle the logical demands of teletype input and output. In essence, such a routine accepts bit-serial data from the teletype and assembles this data in parallel form within the processor.

Typically, the program instructs the processor to idle in a test loop, until such time as it senses the start bit (current zero) on the TTY input line. When the start bit arrives, the processor skips to an input routine, designed to draw in and assemble the serial bits of the ASCII character. With the eight-bit character fully assembled, the processor jumps to a monitor routine which determines whether the message calls for any further processor action. If so, the monitor directs a program jump to the required service routine. If not, the program jumps back to the idling loop, and waits for the next start bit to arrive. Figure 2-23 is a flowchart diagram of a typical programmed input routine.

The INTELLEC® 4/MOD 40 System Monitor contains teletype input and output programs that are suitable for use in the program development phases. Some users, however, will find it necessary to provide their own teletype I/O

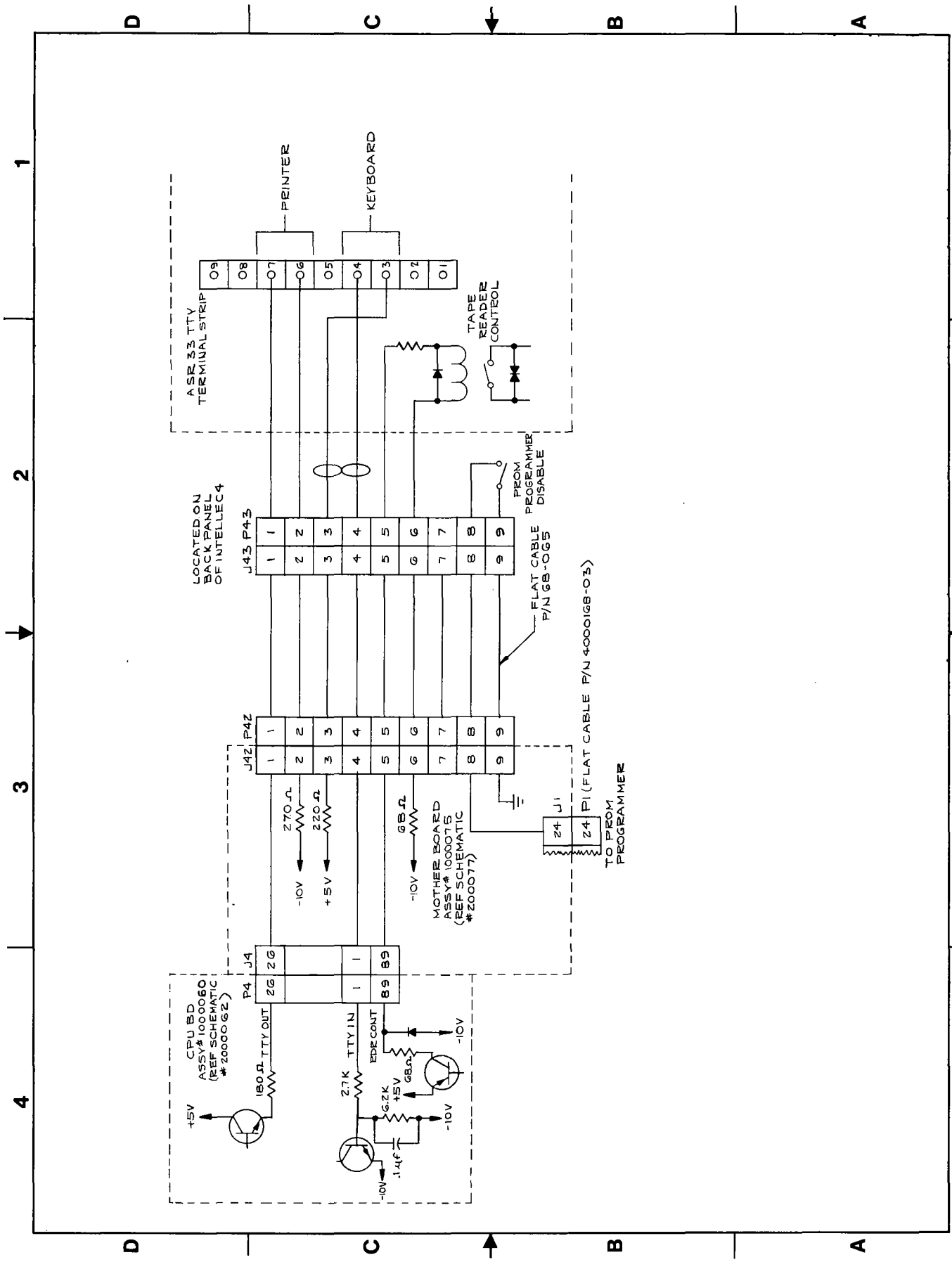


Figure 2-22. TTY Interface

routines for specific applications. Such provisions are a part of the programming function and are technically outside the scope of this manual, but the programmer will not be able to write a suitable routine without having some knowledge of the teletype's requirements. For that reason, we present here some guidelines for writing a suitable program. Detailed instructions on coding will be found in the INTELLEC® 4 Operator's Manual, or in the MCS-40™ User's Manual.

A Typical Teletype Input Routine

The teletype transmits information at a maximum rate of 10 characters per second. Each character occupies a 0.1 second time frame.

Each frame contains 11 sub-intervals, periods in which the loop current may be either on or off, according to the predetermined logical profile of the character being transmitted.

The teletype normally idles in the "marking" condition; that is, with loop current on and indicating a logic "1." The current-off condition is called "spacing," and it indicates a logic "0."

The first interval in all character frames is a space, signalling that the loop is no longer idle, and that transmission of a character is about to begin. This interval is known as the START bit. Its function is to synchronize transmitter and receiver. It has no significance as data.

The next seven intervals, which we shall refer to as b_1 - b_7, are reserved for intelligence transmission. The bits transmitted correspond to the 7 bits of an ASCII character.

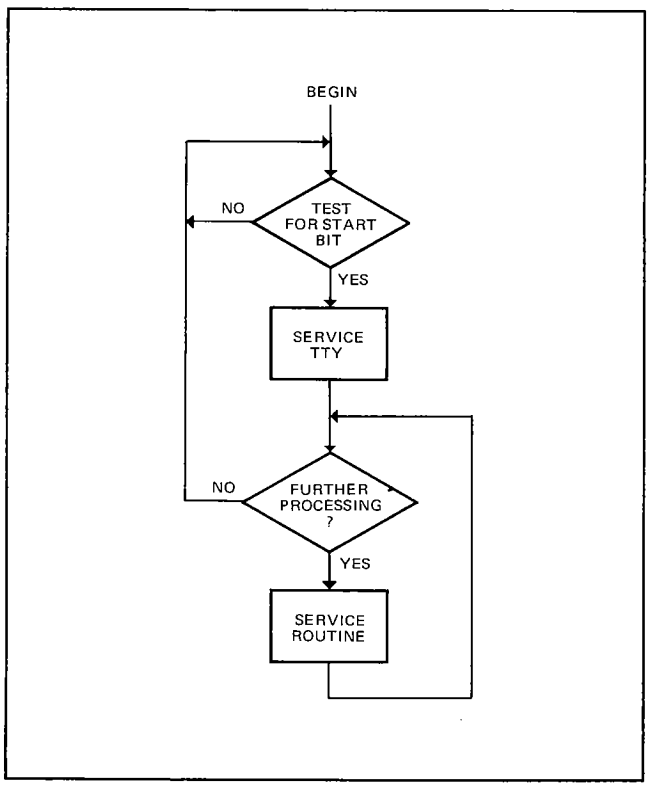


Figure 2-23. Teletype Input Routine

The b_8 interval contains the parity bit. (NOTE: The INTELLEC® 4/MOD 40 System Monitor ignores parity).

Interval b_9, is the STOP bit, and is always a marking interval. Like the START bit, the STOP has a control function rather than any data significance. The interval following the STOP, b_{10}, is likewise a marking bit. This is a recovery interval, prior to transmission of the next character.

The timing relationships are important, and must be observed in the program. Transmission of each bit in the TTY character frame requires 9.09 milliseconds.

The programmer implements software delays by knowing the instruction cycling time of the machine he is working with. In the case of the 4040 CPU, the basic cycle time is 10.8 μsec, with certain instructions requiring a double cycle for completion. Knowing this, and knowing that a delay of 9.09 milliseconds may be required, he can construct a nested iteration which will require approximately 842 cycles to complete. The routine might look like this, symbolically:

```
DELAY   FIM   2 50H
LOOP 1  FIM   0 DEH
LOOP 2  ISZ   1 LOOP1
        ISZ   0 LOOP2
        ISZ   2 LOOP1
        NOP
        NOP
        NOP
        BBL
```

Those unfamiliar with these mnemonics, should refer to Appendix A for explanation. This coded sequence requires 842 cycles for completion, providing a timed delay of approximately 9.09 milliseconds.

The programmer also needs to know that he can cause a mark on the teleprinter line, by writing any odd number into RAM output port 0. Writing any even number causes a space.

In similar fashion, an odd number written into the RAM 1 output port enables the tape reader. An even output turns off the drive.

Knowing this information, and knowing the limitations imposed by the instruction set of the processor chip, we can construct a routine for inputting a teletype character. This routine looks as follows, in logical outline (refer to the INTELLEC® 4 Programming Manual for an explanation of the machine instructions at your disposal).

1) Write a numerical "1" at RAM port 1, this enables the tape reader drive.
2) Initialize the working registers, by placing numerical "0" in index registers 2 and 3 and "8" in index register 4; registers 2 and 3 will be used to hold the input character, while 4 will be used as an overflow counter to determine when all eight bits have arrived.
3) Read ROM input port 0; a "1" indicates the TTY START bit.

4) If a "1" is not present, return to Step 3, otherwise continue; the program will loop between Steps 3 and 4, until such time as the START bit does arrive.

5) Delay for 4.55 milliseconds, to sample the "middle" of the teletype bit.

6) Write a numerical "0" at RAM port 1; this halts the reader, until the program determines what action may be required.

7) "Echo" the START bit by writing a numerical "0" into RAM port 0.

8) Delay for 9.09 milliseconds, to wait for the "middle" of the next bit.

9) Read ROM input port 0, to detect whether a "1" or a "0" is present.

10) Complement the accumulator, to obtain the proper logical polarity, and write the contents of the accumulator into the RAM 0 output port, this echoes the bit read in.

11) Shift right through the carry bit, placing the data bit in the carry position.

12) Transfer the contents of index register 2 to the accumulator, and again shift right through the carry bit; this places the most recent input data bit in the most significant position in the accumulator.

13) Transfer the contents of the accumulator back to index register 2.

14) Transfer the contents of index register 3 to the accumulator, and again shift right through the carry bit; this transfers any significant overflow from the preceding shift into index register 3.

15) Transfer the contents of the accumulator back to index register 3.

16) Increment index register 4; if the contents of register 4 is now zero, continue below; otherwise return to Step 8 and get the next bit.

17) Delay for 9.09 milliseconds, to wait for the middle of the STOP bit.

18) Write a numerical "1" into RAM port 0; this places a terminal "mark" on the teleprinter line.

19) Skip to the executive routine, to determine whether any processing is necessary; the executive will return control of the input routine when it is ready to receive the next character.

Upon completion of Step #18, the input will be halted, and index registers 2 and 3 will contain the eight bits of the TTY character, as shown in Figure 2-24.

A TTY output routine will be similar in its construction to the input program just described. The programmer should now have some idea of how to develop such a program. Detailed help on programming procedures will be found in the INTELLEC® 4 Programming Manual.

For reference, Appendix G of the INTELLEC® 4 Operator's Manual contains the TTY input and output routines used in the INTELLEC® 4/MOD 40 System Monitor, as they appear in assembly language mnemonics. The input routine is a specialized program which ignores the parity bit, but should otherwise prove helpful for illustrative purposes.

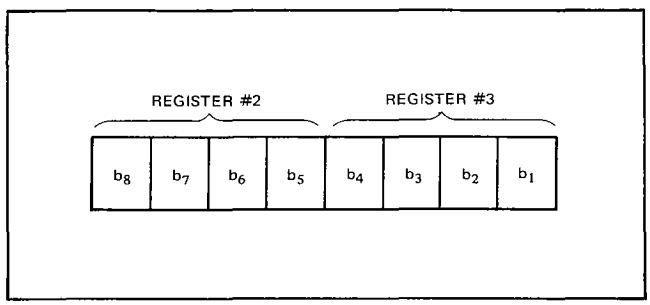


Figure 2-24. Teletype Character Input

Pin List

The following section describes connector pin allocations on the Central Processor Module. The pins and their signal functions are listed in Table 2-2.

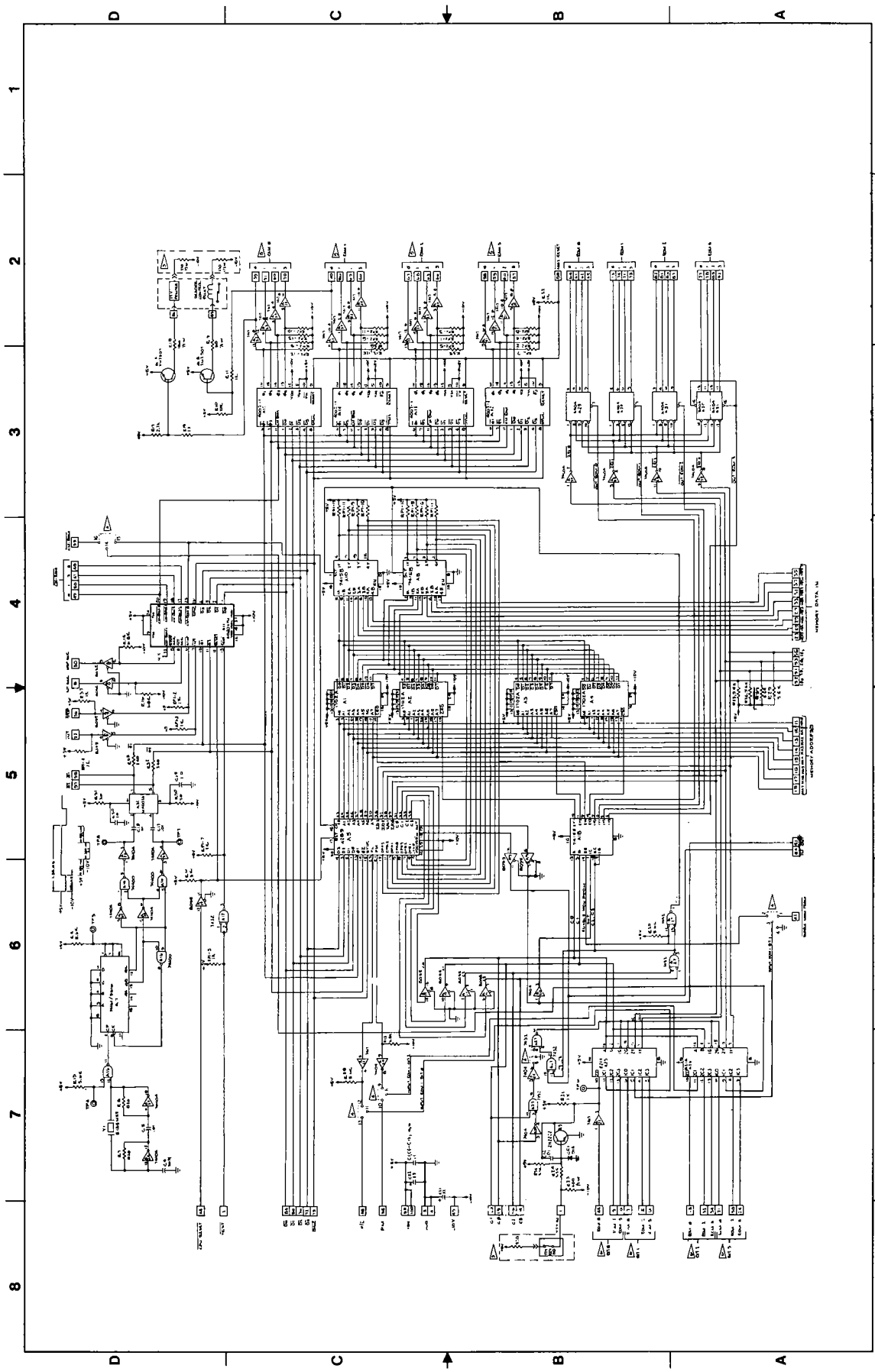


Figure 2-25. Central Processor Module Schematic

P1 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1	TTY IN	Rear Panel J43[2]
2	$\overline{\text{TEST}}$	I/O Connector
3	GROUND	
4	GROUND	
5	ROM IN #2-1	
6	ROM IN #3-1	
7	ROM IN #0-1	Rear Panel Input Connector
8	$\overline{\text{INTERRUPT ACKNOWLEDGE}}$	
9	ROM IN #2-0	
10	ROM IN #3-0	
11	MEMORY ADDRESS 0	
12	MEMORY ADDRESS 1	
13	MEMORY ADDRESS 2	
14	MEMORY ADDRESS 3	PROM/RAM Modules
15	MEMORY ADDRESS 4	
16	MEMORY ADDRESS 5	
17	MEMORY ADDRESS 6	
18	MEMORY ADDRESS 7	
19	CHIP SELECT 0	Memory and I/O Modules
20	CHIP SELECT 1	
21	ROM IN #0-3	Rear Panel Input Connector
22	$\overline{\text{INTERRUPT}}$	
23	$\overline{\text{MEMORY DATA 0}}$	PROM/RAM Modules
24	ROM IN #3-3	Rear Panel Input Connector
25	$\overline{\text{MEMORY DATA 1}}$	PROM/RAM Modules
26	TTY PRINTER	Rear Panel J43[2]
27	$\overline{\text{MEMORY DATA 3}}$	PROM/RAM Modules
28	ROM IN #0-2	Rear Panel Input Connector
29	$\overline{\text{MEMORY DATA 2}}$	PROM/RAM Modules
30	$\overline{\text{STOP ACKNOWLEDGE}}$	Rear Panel Input Connector, Memory Controller
31	$\overline{\text{MEMORY DATA 5}}$	PROM/RAM Modules
32	ROM IN #2-2	Rear Panel Input Connector
33	$\overline{\text{MEMORY DATA 4}}$	PROM/RAM Modules
34	ROM IN #3-2	Rear Panel Input Connector
35	$\overline{\text{MEMORY DATA 7}}$	PROM/RAM Modules
36	$\overline{\text{STOP}}$	Rear Panel Input Connector, Memory Controller
37	$\overline{\text{MEMORY DATA 6}}$	PROM/RAM Modules
38	ROM IN #2-3	Rear Panel Input Connector
39	$\overline{\text{RAM OUT #0-0}}$	Rear Panel Output Connector
40	$\overline{\text{RAM OUT #1-0}}$	Rear Panel Output Connector
41	$\overline{\text{OUT}}$	I/O Modules
42	$\overline{\text{ENABLE MON PROM}}$	Console Switch
43	-10 VDC	Power Supply
44	ROM IN #0-0	Rear Panel Input Connector
45	$\overline{\text{CPU RESET}}$	Control Module, I/O Connector
46	RAM OUT #1-3	Rear Panel Output Connector
47	$\overline{\text{CM-RAM2}}$	
48	$\overline{\text{CM-RAM3}}$	Memory & I/O Modules
49	$\overline{\text{CM-RAM0}}$	
50	$\overline{\text{CM-RAM1}}$	
51	I/O 1	
52	I/O 0	Control & I/O Modules
53	I/O 2	

Table 2-2.

P1 Pin List (Continued)

PIN	SIGNAL FUNCTION	DESTINATION
54	$\overline{IN}$	I/O Modules
55	F/$\overline{L}$	Control Module
56	I/O 3	Control & I/O Modules
57	$\overline{RAM\ OUT\ \#\ 3\text{-}3}$	Rear Panel Output Connector
58	$\overline{RAM\ OUT\ \#\ 3\text{-}0}$	
59	$\overline{RAM\ OUT\ \#\ 3\text{-}1}$	
60	$\overline{RAM\ OUT\ \#\ 3\text{-}2}$	
61	$\overline{ROM\ OUT\ \#\ 0\text{-}1}$	
62	$\overline{ROM\ OUT\ \#\ 0\text{-}2}$	
63	RAM OUT # 2-2	
64	ROM OUT # 0-0	
65	ROM OUT # 0-3	
66	$\overline{RAM\ OUT\ \#\ 2\text{-}3}$	
67	$\overline{RAM\ OUT\ \#\ 2\text{-}0}$	
68	$\overline{RAM\ OUT\ \#\ 2\text{-}1}$	
69	$\overline{RAM\ OUT\ \#\ 1\text{-}2}$	
70	$\overline{RAM\ OUT\ \#\ 0\text{-}3}$	
71	ROM OUT # 1-2	
72	$\overline{DATA\ 3}$	Control and Memory Modules
73	ROM OUT # 1-0	Rear Panel Output Connector
74	ROM OUT # 1-1	
75	ROM OUT # 1-3	
76	$\overline{DATA\ 2}$	Control and Memory Modules
77	ROM OUT # 3-0	Rear Panel Output Connector
78	ROM OUT # 3-1	
79	$\overline{SYNC}$	Control and Memory Modules
80	$\overline{DATA\ 1}$	
81	ROM OUT # 2-1	Rear Panel Output Connector
82	ROM OUT # 2-0	
83	$\overline{DATA\ 0}$	Control and Memory Modules
84	ROM OUT # 2-2	Rear Panel Output Connector
85	$\overline{ROM\ OUT\ \#\ 3\text{-}2}$	
86	$\overline{RAM\ OUT\ \#\ 1\text{-}1}$	
87	ROM OUT # 2-3	
88	$\overline{4002\ RESET}$	Control Module
89	READER CONTROL	Rear Panel J43[2]
90	$\overline{RAM\ OUT\ \#\ 0\text{-}2}$	Rear Panel Output Connector
91	$\overline{RAM\ OUT\ \#\ 0\text{-}1}$	
92	ROM OUT # 3-3	
93	$\overline{CM\text{-}ROM}$	Control and Memory Modules
94	CHIP SELECT 3	Control and Memory Modules
95	W	Control Module[2]
96	CHIP SELECT 2	Control and Memory Modules
97	$\overline{Ø2}$ CLOCK	Control and Memory Modules
98	$\overline{Ø1}$ CLOCK	
99	+5 VDC	Power Supply
100	+5 VDC	

NOTES: [1]All signal inputs and outputs are at TTL levels, except as otherwise noted.
[2]TTY wired for 20 mA operation.
[3]Function available but not used in –/MOD 40 systems. Refer to MCS-40™ User's Handbook.

Table 2-2. (Continued)

CHAPTER 3 THE imm6-28 RAM MEMORY MODULE

The imm6-28 RAM Memory Module provides users of Intel's INTELLEC® 4/MOD 40 System with an expanded data storage facility of 4K X 8 bits. Its peripheral logic is designed to permit convenient interface with the Central Processor Module, which is the functional core of the INTELLEC® 4/MOD 40 system. Although designed primarily for that application, the RAM Memory Module will also serve as a versatile and convenient modular memory unit in any microcomputer system designed around the 4040 Monolithic Central Processor Unit, Intel's® LSI microprocessor.

This chapter provides information on the RAM Memory Module, its operation and its utilization. For details of its logical interface with the Central Processor Module, refer to Chapter 5 of the INTELLEC® 4/MOD 40 Reference Manual. That chapter describes the imm4-72 Control Module.

FUNCTIONAL DESCRIPTION

Figure 3-1 is the functional block diagram of the imm6-28 module. The RAM Memory Module contains the following blocks:

a) main memory array
b) address buffer
c) address logic
d) input and output buffers
e) command logic

The main memory array contains 32 individual 1K x 1 memory elements, arranged in four logical rows of eight chips each. Each row is further divided into two four-element banks. Thus, there are 8K x 4 separately addressable locations on the module.

The organization and the addressing structure of the memory is chosen in consideration of the 4040 microprocessor, a four-bit parallel CPU which recognizes an eight-bit instruction word. With reference to the 4040, the module provides a storage capacity of 4K x 8 instructions, each consisting of two separately addressable four-bit bytes.

The address buffer and the address logic mediate between the processor circuitry and the memory elements themselves. A total of twelve address lines are used to specify the basic 4K x 8 locations.

Ten of the address lines, A_0 - A_9, are forwarded directly to all elements of the array, in parallel. The remaining two lines, A_{10} and A_{11}, are applied to the address logic and decoded, to obtain a four-line row selection signal.

Two additional lines are used to specify BYTE 1 or BYTE 2 of the selected row in the array, during write operations. All eight bits of the selected row are enabled simultaneously during readout.

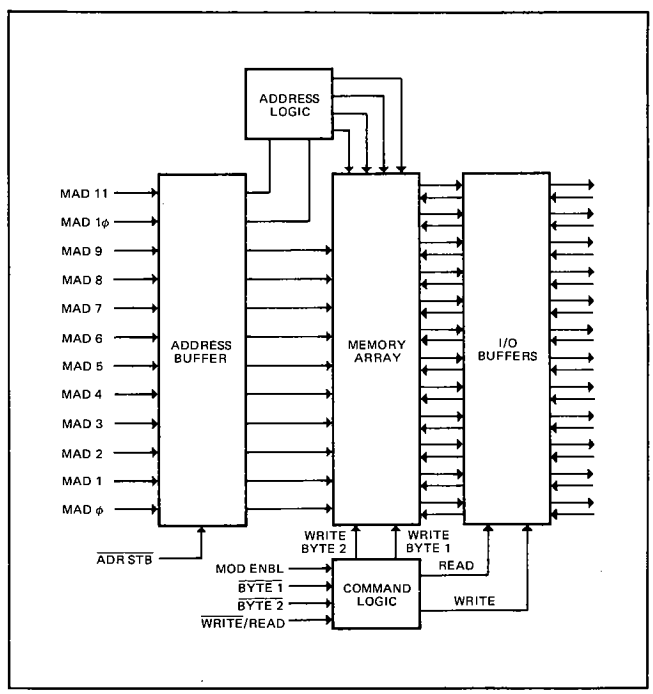


Figure 3-1. RAM Memory Module Functional Block

The DATA OUT lines of the four elements in each column of the array are tied together to one of the eight output buffers. Similarly, the DATA IN lines of all elements in a column are connected to the output of one of the eight input buffers. The input and output buffers handle all data passing in and out of storage.

The command logic is minimal. It enables the module's inputs and outputs, when the module in turn is enabled by an external selection signal. This section also receives the external $\overline{\text{WRITE}}$ command, and uses it to activate the appropriate buffer and memory control lines. Its action is fully described in the section entitled "Theory of Operation of the RAM Memory Module."

2102 RANDOM ACCESS MEMORY ELEMENT

The Intel® 2102 is a static MOS random access memory element, featuring internal address decoding and read/write logic. Capacity of the 2102 is 1024 x 1 bits (abbreviated 1K x 1), and access time is typically on the order of 650 nanoseconds.

The memory chip requires power of +5 VDC. Signal inputs and outputs are at TTL levels, and all except $\overline{\text{CE}}$ are positive-true.

Figure 3-2 is the block diagram of the 2102's internal logic. The address, in positive-true form, is applied to 10 parallel input lines, A_0 through A_9. As shown, the 2102 uses the 10-line address to identify a single location within the 32 x 32 array.

The memory chip is enabled, for purposes either of READ or of WRITE, whenever the $\overline{\text{CE}}$ line is clamped to a logic LOW level. READ occurs by signal default; that is, a continuous READ state exists whenever a WRITE is not in progress. The readout is non-destructive.

The application of a logical LOW to pin 3 (R/$\overline{\text{W}}$) initiates the writing process. Data present at pin #11 (DATA IN) is stored in the addressed location, when the WRITE occurs. Data read out has the same polarity as the data written in. The timing of the READ and WRITE operations is shown in Figure 3-3.

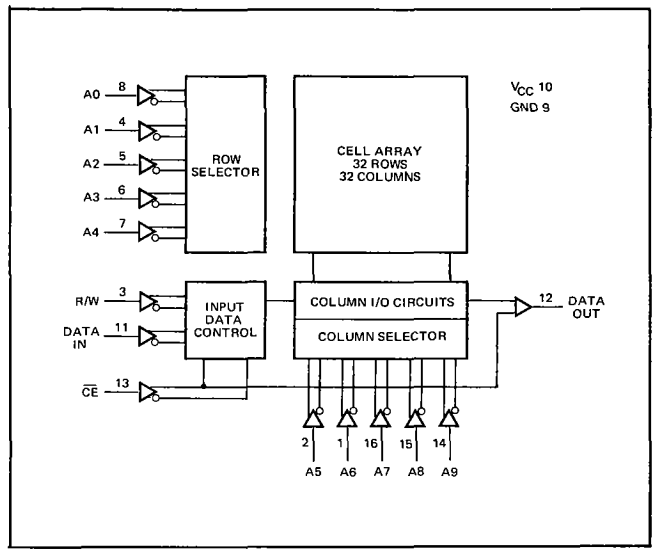


Figure 3-2. 2102 RAM Logical Block Diagram

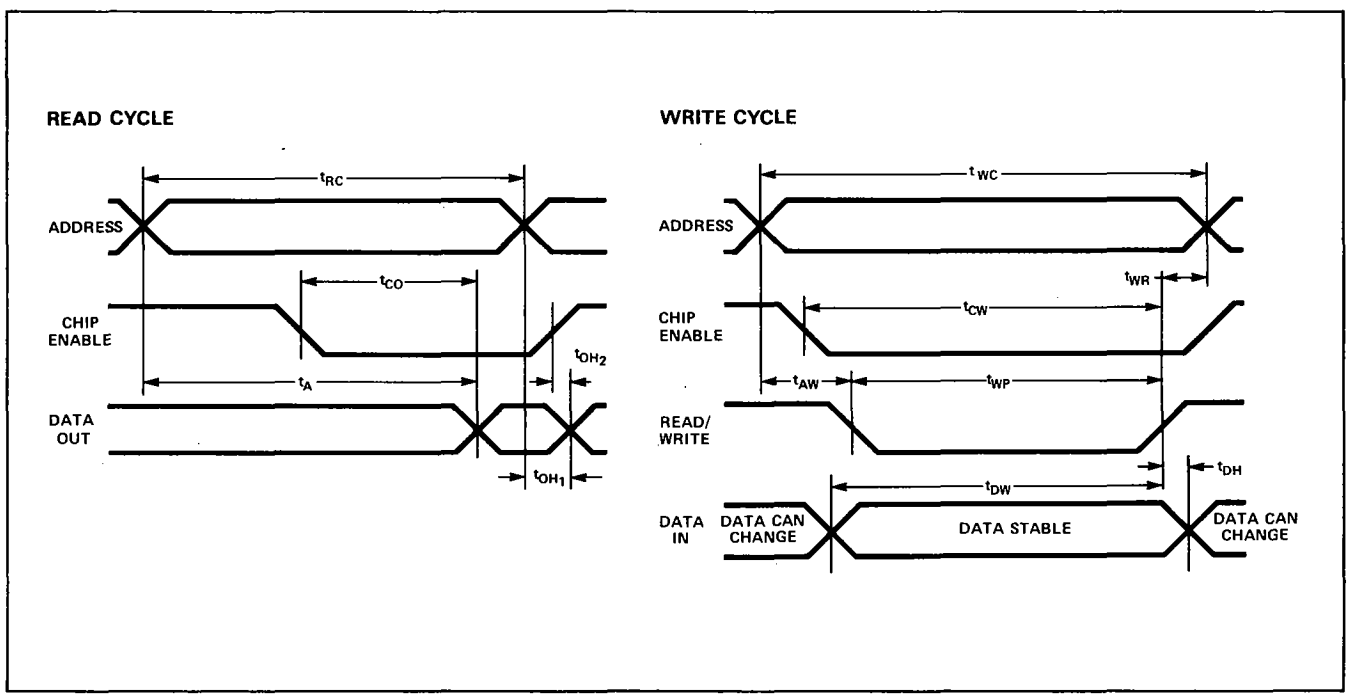


Figure 3-3. 2102 RAM Timing Diagram

THEORY OF OPERATION OF THE RAM MEMORY MODULE

On page 35 we have described the functional organization of the RAM Memory module. This section describes its operation in more detail.

Refer to the module schematic, Figure 3-4.

The address buffer consists of the ten 3404 inverting latch sections on the left side of the diagram, plus the two similar latches in the upper right hand corner. The Intel 3404 is a hex inverting latch, which transfers data to its output pins when a negative-true signal level is applied to the $\overline{\text{STROBE}}$ inputs.

The $\overline{\text{ADR STB}}$ input, at pin #91 of the assembly, passes through a double inverter consisting of two spare 3404 sections whose $\overline{\text{STROBE}}$ inputs are tied permanently LOW. The output of the second inverter stage is tied in common to the $\overline{\text{STROBE}}$ inputs of all the address latching elements. When the module is used in the INTELLEC® 4/MOD 40 system, pin #91 is tied permanently to signal ground in the memory Control Module. A logical LOW is therefore applied continuously to all STROBE inputs, and the address latch is permanently enabled as a result.

The functional core of the module is the memory array itself. As the schematic shows, this array consists of 32 Intel 2102 RAM memory elements, in a 4 x 8 matrix. The A_0 - A_9 address lines of all 32 elements are connected in parallel to the outputs of the 10 address drivers shown at the left of the schematic. The detailed connections are omitted on the diagram, for the sake of clarity, but they are summarized on the generalized 2102 shown to the left of the actual array. These 10 lines intersect a single location within the 32 x 32 internal matrix of each array element.

The two remaining address lines, A_{10} and A_{11} pass through latch inverting elements and are applied to the 2-to-4 line decoding network which consists of a pair of 7404 inverters and a 7400 quad NAND-gate. The decoder derives a four-line output on the basis of the four possible state-combinations on its two input lines. Each output line is tied to the CE pins of eight 2102s on the board, selecting and commanding one of four rows within the array.

One of the eight elements in each memory row has its output pin tied to the output pins of three corresponding elements, one in each of the other rows on the board. In similar fashion, the input pins of corresponding elements are tied together. Input and output columns are connected to the input and output buffers, shown on the schematic below the main array.

The input and output buffers are coincidence banks. The input bank consists of eight 3404 inverting latches, and the output consists of eight 7438 NAND-gates. Both banks invert the signals input, and the logical relationship between memory inputs and memory outputs is thus preserved: a true input will be read as a true output. The input bank, however, is enabled by a LOW logic level on its common STROBE line; the output bank requires a HIGH enabling level for coincidence. The control whips of both banks are under supervision of the command logic.

The command logic accepts a five-line MOD SEL/MOD ENABLE signal, the purpose of which is to identify the module when it is part of a system that contains several such memory modules. This input is used to enable the RAM Memory Module, either for READ or for WRITE operations. The command logic also receives a $\overline{\text{WRITE}}$/READ signal, via pin #95 of the assembly. This signal is used to develop the internal control commands for the module.

The MOD SEL and the MOD ENABLE inputs are applied to an expanded NAND-gate, which enables the read and write logic when HIGH coincidence occurs. The MOD SEL and MOD ENABLE lines are normally tied together in the memory Control Module, when the imm6-28 is used in the INTELLEC®4/MOD 40 System. A single selection signal is thus able to command the RAM Memory Module.

The $\overline{\text{WRITE}}$/READ signal is routed to a pair of inverters, in order to buffer the input, and to obtain opposite signal phases. One phase is applied to one input of a NAND-gate element, and the other is applied to a second element. The alternate side of both gates is enabled simultaneously, by the module enabling signal. Thus, when the module is enabled, one coincidence output will be LOW. If the 1-2-3 section is LOW, the input buffer will be gated on; the contrary condition causes enabling of the output bank.

When a command to $\overline{\text{WRITE}}$ is received, 4A-3 will go LOW, and the negative transition applied to 5B-11 triggers a delay one-shot. After a short propagation delay, the positive pulse output at 5B-9 triggers the 1-2-3-4-5-6-7 section of the dual one-shot, producing a positive-going pulse at 5B-6. The delay in writing allows time for setup of the data in the input latches.

The pulse output at 5B-6 is directed in parallel to corresponding inputs of a pair of 7400 NAND-gates. Here the selection of the byte is made. One or both of these gates may be enabled, depending upon the state of the $\overline{\text{BYTE 1}}$ and $\overline{\text{BYTE 2}}$ inputs. When the module is used in the INTELLEC® 4/MOD 40 System, one or the other line will be LOW, as selected by logic on the memory Control Module. Only one of the two NAND-gates will therefore be open, with a LOW on its output pin.

The output of the gate (or gates) is directed to the inputs of all elements in four of the columns of the main array. The four (or eight) elements at the intersection of the row selection whip and the byte selection line will react, by storing the data present at the input latches in the addressed location.

When the write cycle subsides, and the $\overline{\text{WRITE}}$/READ line again returns to a HIGH level, the module relaxes into the READ state. The whip controlling the output buffers is enabled, as long as the module itself is enabled, and data in the addressed location will be present continuously, at the module's data outputs.

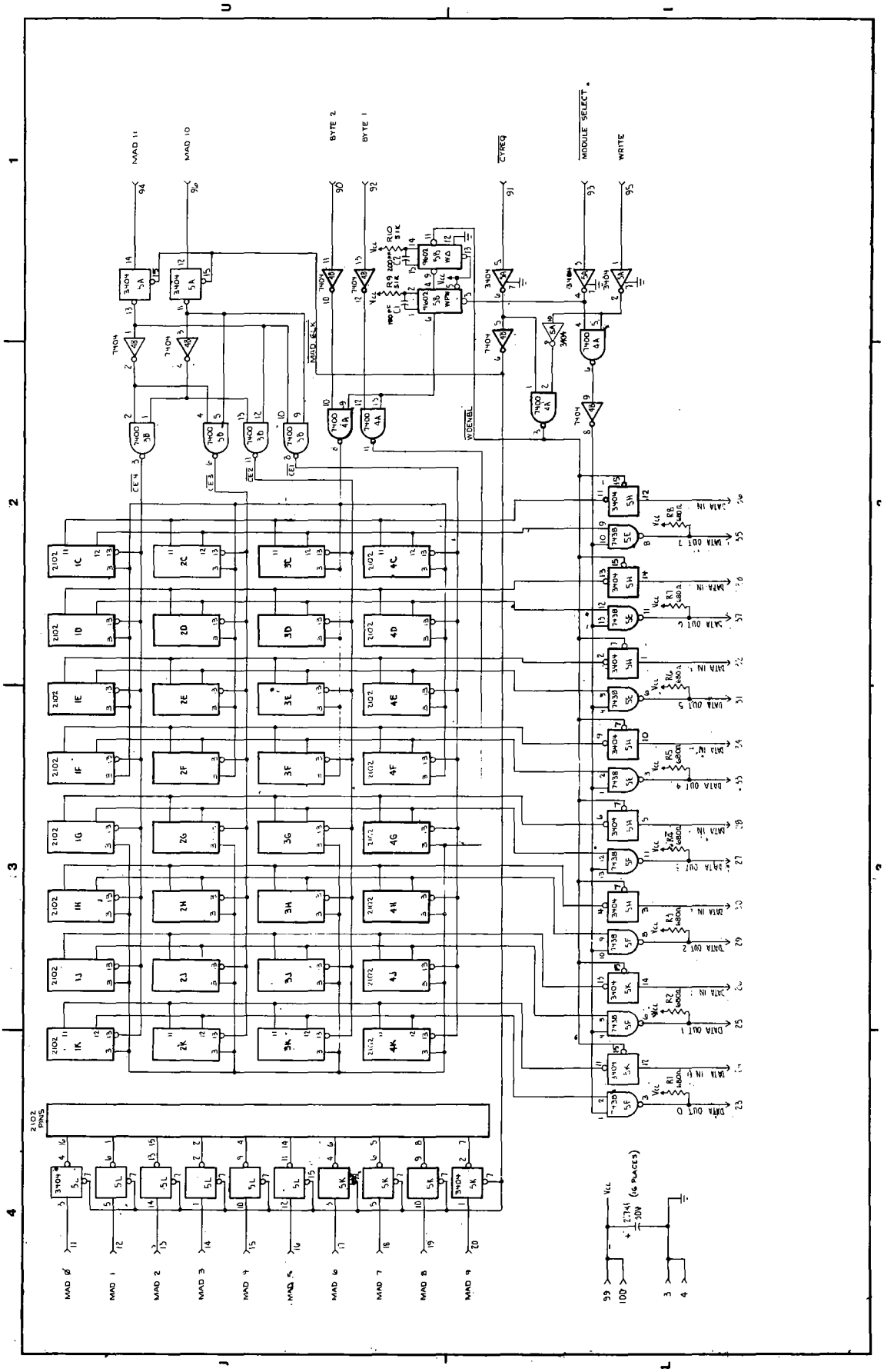


Figure 3-4. RAM Memory Module Schematic

The four signal inverters shown in the lower right hand corner of the module schematic (Figure 3-3) are not used in the INTELLEC® 4/MOD 40 System. They are provided as a means of extending the addressing range of systems where several RAM Memory Modules are in use.

An extended addressing scheme is implemented by connecting external jumpers between the output pins of selected inverters and the MOD SEL input pins, used for enabling of the module. As many as sixteen modules may be accommodated in this manner.

As an example, suppose that our module is designated number 10, in a bank of sixteen modules, numbered 0 to 15. Four addressing digits are allocated for selection of the module, and the state-combination on these lines which corresponds to the card in question is given by the Boolean equation:

$$SELECT = MAD15 \bullet \overline{MAD14} \bullet MAD13 \bullet \overline{MAD12}$$

To identify the card, we connect the following jumpers:

a) pin #63 to pin #64
b) pin #57 to pin #58

We then use the following module pins as positive-true address lines:

Address Line #15	pin #67
#14	pin #66
#13	pin #61
#12	pin #60

The parallel coincidence of the binary address prefix 1010 will, in conjunction with the MOD ENABLE signal, open gate 5X and enable the module.

Other card identities are established in similar fashion.

UTILIZATION

This section provides information on utilization of the imm6-28, for those who contemplate using the module outside the INTELLEC® 4/MOD 40 System.

Installation

In installing the RAM Memory Module, the user must take into account:

a) environmental extremes
b) mounting
c) electrical connections
d) power requirements
e) signal requirements
f) wiring options

ENVIRONMENT

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conductive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity is not critical to the module's operation.

MOUNTING

Avoid locating the card near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections. The result might be abnormally high noise levels, with the problems entailed, or outright failure of the card.

Dimensions of the module are 6.18 x 8.00 inches. Be sure to allow enough additional clearance to ensure adequate cooling.

The card is designed to plug directly into a standard 100-pin, double-sided PC edge connector. The connector will serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the card be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the card, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the elements on the module.

ELECTRICAL CONNECTION

All connections to the Central Processor Module are effected by means of a standard 100-pin, double-sided PC edge connector having .125" centers. CDC #VPB 01C50E0-0A1 is one suitable type.

POWER REQUIREMENTS

The RAM Memory Module requires only one source of DC power:

+5V ± 5% @ 2.5 A (max)
1.25 A (typ)

Refer to the pin list for the power connections.

SIGNAL REQUIREMENTS

All module inputs and outputs are TTL-compatible.

The RAM Memory Module requires a 12-bit parallel address. When the module is used outside the INTELLEC® 4/MOD 40 System, the user will also have to provide a two-line byte selection signal. In some applications, it will be convenient simply to tie both byte selection pins to signal ground, and leave them permanently enabled. If the card is used with a four-bit processor, however, byte selection is necessary. A simple TTL inverter will serve to split a single command line into two alternative signal phases. A flip-flop bistable would be another acceptable solution. The $\overline{BYTE\ 1}$ and $\overline{BYTE\ 2}$ inputs are negative-true.

The address strobe ($\overline{ADR\ STB}$) is also negative-true. This line may be tied directly to signal ground, if address

gating is not a requirement. Otherwise, provide a negative-going pulse in synchronism with the address data.

The $\overline{\text{WRITE}}$/READ input is a two-state command line. A TTL LOW initiates a data write-in, while a HIGH causes a readout. A single processor output line will serve this function.

The data inputs and outputs are also TTL-compatible; eight input and eight output lines are needed. For signal pin For signal pin allocations, refer to Pin List.

In the event that an extended address structure is desired, four additional address lines will be necessary. These will be tied either to the MAD12 - MAD15 inputs, or to the MOD SEL 12 - MOD SEL 15 inputs, in combination as the board identity requires. If such a scheme is unnecessary, the MOD SEL and the MOD ENABLE lines should be tied together to a positive-true enabling signal. If module switching is altogether unnecessary, simply return all these pins to +5 VDC, by jumpering through a 1K pull-up resistance to V_{CC}.

WIRING OPTIONS

Wiring options on the imm6-28 consist simply of the edge connector pin jumpers necessary to establish the card's identity. Such provisions are required only in those cases where more than one module is used. Page 10 describes the procedure for determining the card's identity. As many as sixteen modules may be accommodated in this fashion.

Pin List

The following section describes connector pin allocations on the RAM Memory Module. The pins and their signal functions are listed in Table 3-1.

P1 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1		
2		
3	GROUND	
4	GROUND	
5		
6		
7		
8		
9		
10		
11	ADDRESS 0	From Control Module
12	ADDRESS 1	
13	ADDRESS 2	
14	ADDRESS 3	
15	ADDRESS 4	
16	ADDRESS 5	
17	ADDRESS 6	
18	ADDRESS 7	
19	ADDRESS 8	
20	ADDRESS 9	
21		
22		
23	DATA OUT 0	Control and CPU Modules
24	DATA IN 0	Control Module
25	DATA OUT 1	Control and CPU Modules
26	DATA IN 1	Control Module
27	DATA OUT 3	Control and CPU Modules
28	DATA IN 3	Control Module
29	DATA OUT 2	Control and CPU Modules
30	DATA IN 2	Control Module
31	DATA OUT 5	Control and CPU Modules
32	DATA IN 5	Control Module
33	DATA OUT 4	Control and CPU Modules
34	DATA IN 4	Control Module
35	DATA OUT 7	Control and CPU Modules
36	DATA IN 7	Control Module
37	DATA OUT 6	Control and CPU Modules
38	DATA IN 6	Control Module
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		

Table 3-1.

All signal inputs and outputs are at TTL levels, unless otherwise noted.

P1 Pin List (Continued)

PIN	SIGNAL FUNCTION	DESTINATION
51		
52		
53		
54		
55		
56		
57	$\overline{\text{ADDRESS 12}}$	Control Module
58	MOD SELECT 12	Control Module
59	ADDRESS 13	RAM Module
60	ADDRESS 12	RAM Module
61	MOD SELECT 13	Control Module
62	$\overline{\text{ADDRESS 13}}$	
63	$\overline{\text{ADDRESS 14}}$	
64	MOD SELECT 14	
65	ADDRESS 15	RAM Module
66	ADDRESS 14	RAM Module
67	MOD SELECT 15	Control Module
68	$\overline{\text{ADDRESS 15}}$	
69		
70		
71		
72		
73		
74		
75		
76		
77		
78		
79		
80		
81		
82		
83		
84		
85		
86		
87		
88		
89		Control Module
90	$\overline{\text{BYTE 2}}$	
91	$\overline{\text{ADDRESS STROBE}}$	
92	$\overline{\text{BYTE 1}}$	
93	MOD ENABLE	
94	ADDRESS 11	RAM Module
95	$\overline{\text{WRITE/READ}}$	Control Module
96	ADDRESS 10	RAM Module
97		
98		
99	+5 VCD	Power Supply
100	+5 VDC	

Table 3-1. (Continued)

All signal inputs and outputs are at TTL levels, unless otherwise noted.

CHAPTER 4 THE CONTROL MODULE

The Control Module is a memory controller, designed for use in the INTELLEC® 4/MOD 40 system. The Control Module acts as an interface between the Central Processor Module, and the RAM Memory Module.

The Control Module is also equipped to perform several service functions which are related to the operation of the INTELLEC® 4/MOD 40 system. These include:

a) memory selection—the INTELLEC® 4/MOD 40 system uses three kinds of program memory:

   1) monitor PROM, which contains the INTELLEC® 4/MOD 40 System Monitor

   2) PROM, which contains a user-specific firmware program, or programs

   3) program RAM, which may also be used for program storage

   The module contains circuitry for the selection and enabling of these several memory banks.

b) system reset—the Control Module generates and distributes the system RESET signals, including: $\overline{\text{CPU RESET}}$, $\overline{\text{4002 RESET}}$, and $\overline{\text{PANEL RESET}}$.

c) processor control—the module operates in conjunction with the system Control and Display Panel to generate a $\overline{\text{STOP}}$ command for the Central Processor Module. Included on the Control Module is circuitry which enables the operator to interrupt the $\overline{\text{STOP}}$ output momentarily, so that single-step operations may be performed. This feature is particularly convenient in program debugging.

d) signal buffering—the Control Module contains an inverter bank, used to buffer the signals en route from the Central Processor Module to the system Control and Display Panel; these include the four lines of the main data bus ($\overline{D}_0$ - $\overline{D}_3$) and the memory command lines ($\overline{\text{CM-ROM}}_0$ and $\overline{\text{CM-RAM}}_{0-3}$).

As a memory interface, the Control Module permits the imm6-28 RAM Memory Module to simulate a 4K x 8 block of PROM memory. The PROM memory comprised of Intel® 4702A Programmable Read Only Memory elements, is in itself arranged to simulate yet another kind of Intel® memory, the 4001 Mask Programmed Read Only Memory. This two-level simulation requires a brief explanation.

The 4001 ROM is a production memory element which is programmed during its manufacture to a predetermined set of user specifications. While this element is an extremely economical and efficient memory vehicle, when used in the production of dedicated OEM microcomputers, it is poorly suited to the early phases of system development, when programs are being written and de-bugged. The design of the 4001 makes it awkward to "dump" the memory's contents for examination. And the turnaround time involved in obtaining de-bugged prototype ROMs is liable to be prohibitive.

To overcome the latter difficulty, Intel developed a user programmable memory element, one that could be erased and re-loaded as often as required. This element, the 4702A PROM is programmed by application of higher-than-usual voltage pulses in a carefully controlled timing sequence. The entire programming operation is performed automatically by the imm6-76 PROM Programmer Module. Erasure of the element is even easier, requiring only that the element be exposed to a high intensity ultra-violet source.

With all its flexibility, the 4702A has one disadvantage as a development device. It lacks quite a bit of the internal logic which is incorporated in manufacture into the Intel 4001. When the PROM is used to simulate the 4001, this additional logic must be provided externally. For this purpose, the 4289 memory interface set was developed. This monolithic component performs much of the addressing and I/O logic normally associated with the 4001. A very satisfactory simulation is thus achieved.

Two problems remained, however. ROM can be simulated effectively and economically by PROMs used in conjunction with the 4289, but there are still times when it is

desirable to alter a program step without resorting to the removal, erasure, and re-programming of the PROM. Furthermore, it is still difficult to "dump" the PROM contents during debugging.

Dumping is difficult, because the processor normally uses the program memory for storage of instructions only. Instruction fetching is a **reflexive** operation, involving careful timing and closely coordinated activities on the part of both processor and memory logic. The instruction passes directly from memory to the processor's instruction register, without ever passing through the arithmetic logic within the processor. As a consequence, there is no opportunity for outputting the instruction, so that an operator can examine its contents.

By way of contrast, memory elements such as the 4002's used for scratchpad on the Central Processor Module are accessed **transitively**. In transitive operations, both an operation code and an operand are specified. By using the appropriate program instructions, the operator can specify a location and extract or store information therein. Operations of this kind are not possible in the case of ordinary program memory.

Thus PROM is excellent for the semi-permanent, non-volatile storage of programs undergoing development and testing, but it needs to be augmented by a more flexible read/write memory, in which the program steps may initially be tested and corrected as necessary. A memory capable of operating in both the transitive and reflexive modes is required. The Control Module enables the RAM Memory Module to perform that function.

The Control Module contains circuitry which enables the operator to read the contents of program RAM, using a special sequence of instructions. The operator can also alter the contents of the program RAM, using another special instruction sequence.

In addition, the operator has the option known as **console memory access** (CMA). By using switches on the Control Panel of the INTELLEC® 4/MOD 40 system, he can enter data into program RAM directly, without resort to the system software. While it would be awkward to load an entire program in this manner (the System Monitor does that), it is extremely convenient to correct individual instructions, using the CMA feature. The Control Module provides the switching and logic circuitry necessary to accomplish that evolution.

The combination of the Central Processor Module, the RAM Memory Module, and the Control Module therefore permits three phase program development:

a) initial writing and debugging, in program RAM
b) intermediate storage and final testing in PROM
c) production, in mask programmed ROM

System software enables automatic programming of PROMs, directly from the contents of the control program RAM. It also permits the automatic punching of perforated tapes, in BNPF format, for the ordering of production ROM elements.

The Control Module is thus designed principally for incorporation into the INTELLEC® 4/MOD 40 system. It could conceivably, however, find application in other systems, where a 4040 CPU and a 4289 interface set must access a RAM memory whose structure is similar to that of the imm6-28 RAM Memory Module.

The module requires only DC power, at levels of +5 VDC and -10 VDC. All signal inputs and outputs with the exception of $\overline{4002\ RESET}$, are at TTL levels. The $\overline{4002\ RESET}$ is a MOS level signal.

FUNCTIONAL DESCRIPTION OF THE MODULE

Refer to Figure 4-1, the Control Module Functional Block Diagram.

The Control Module contains the functional blocks shown in Figure 4-1. They include:

a) the timing generator
b) the write enable multiplexer
c) the write generator
d) the console memory access (CMA) enable multiplexer
e) the address multiplexer
f) the write data multiplexer
g) the A/X multiplexer
h) the I/O ports
i) the port decoder
j) the memory selection logic
k) the RESET & STOP generator

The easiest way to understand the Control Module is to "walk" through each of the four major operations that the module must perform. They are:

1) reflexive read (program mode)
2) CMA write
3) transitive write
4) transitive read

After discussing these, we shall briefly describe the miscellaneous control functions on the module, including RESET, STOP, and memory selection.

Reflexive Read (Program Mode)

The 4040 processor chip, located on the Central Processor Module, has an eight part cycle. In the natural order of their occurrence, these phases are labelled: A_1 - A_2 - A_3 - M_1 - M_2 - X_1 - X_2 - X_3. The first three phases of the cycle are devoted to the addressing of program memory. During this interval, the processor transmits three four-bit bytes which specify the address of the required instruction word. The least significant byte occupies A_1, and the most significant byte occupies A_3. The memory must respond by returning two four-bit bytes, one during M_1, and one during M_2. The operation code (OPR) occupies M_1, and the operand (OPA) occupies M_2. The remaining three phases, X_1 through X_3, are reserved for execution of the instruction.

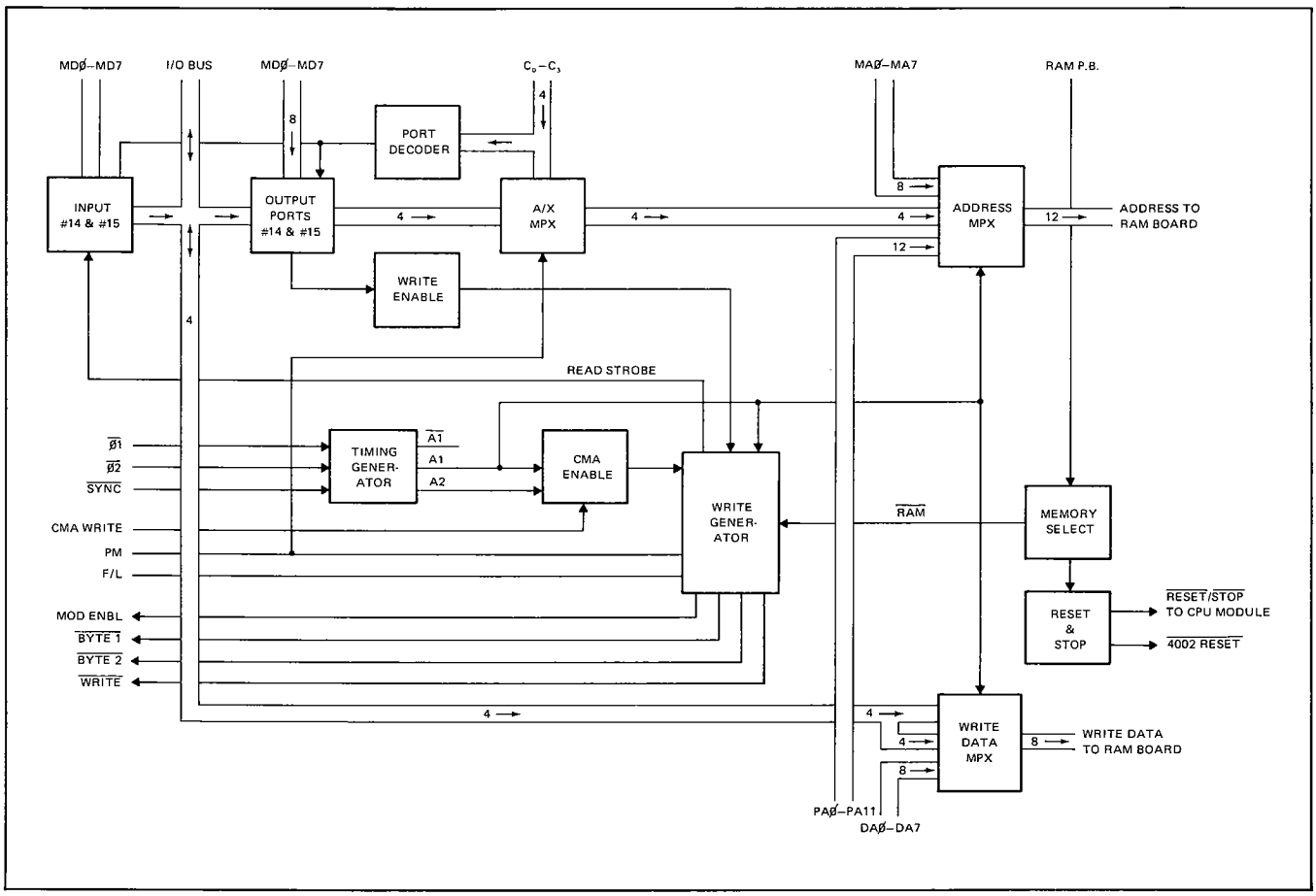


Figure 4-1. Control Module Functional Block

The 4289 Memory Interface chip, also located on the Central Processor Module, accepts each of the three address bytes from the processor and stores them in internal registers. The A_1 - A_3 bytes will thus be available in parallel, during M_1 and M_2 of the cycle.

The eight low-order bits of the address are transmitted to the Control Module on one bus, and the four high-order bits are sent on another. The low-order address (MA_0-MA_7) goes directly to the address multiplexer, as shown on the Functional Block Diagram. The high-order address (C_0-C_3) is applied to an intermediate switch, the A/X multiplexer.

The A/X multiplexer receives two 4-line data inputs, and forwards one of them to the address multiplexer. The multiplexer transmits data selectively, on the basis of the PM timing signal that it receives from the 4289 memory interface. PM occurs whenever a transitive read (RPM) or write (WPM) is executed. We therefore have a twelve-line parallel address assembled at one input to the address multiplexer. The address will remain available during PM.

Observe that the address multiplex receives a second twelve-line data input, from the Control and Display Panel of the INTELLEC® 4/MOD 40 system. The address multiplexer forwards one of these data inputs to the RAM Memory Module, via its twelve-line output bus. The selection depends upon the phase in progress. The $\overline{A}_1$ signal from the timing generator causes the address multiplexer to select the data from the panel. During the rest of the system cycle, the address multiplexer relaxes into its alternate state. The address data from the Central Processor Module is therefore available to the RAM Memory Module, during A_2 through X_3 of any given cycle.

The timing generator receives its inputs from the timing section of the Central Processor Module. Inputs include: $\overline{01}$, $\overline{02}$, and $\overline{SYNC}$. Circuitry within the timing generator uses these signals to derive the synchronization pulses for the remainder of the circuitry on the Control Module. The operations that the Control Module performs require this accurate timing, since they are referenced ultimately to the operations of the processor chip itself.

The instruction address which originated in the processor is presented to the RAM Memory Module during M_1 and M_2 of the system cycle. But the RAM Memory Module must also receive an enabling signal, before it can respond to the address.

When the operator puts the system into the RAM mode he pushes the MODE CONTROL:RAM selector on the INTELLEC® 4/MOD 40 system. In so doing, he activates a control line connected to the memory selection section of the

Control Module. The memory selection circuitry sends a $\overline{RAM}$ signal to the write generator section, and the write generator responds by forwarding a MOD ENABLE signal to the RAM Memory Module.

The enabled RAM Memory Module immediately returns the contents of the addressed memory location. This data goes directly to the Central Processor Module, on an eight-line bus (MD_0 - MD_7). The 4289 Memory Interface on the Central Processor Module dismantles this field, transmitting two four-bit bytes back to the processor chip, during M_1 and M_2. The logical requirements of the processor's fetch cycle are thus fulfilled.

CMA Write (Console Memory Access)

The console memory access feature permits an operator to write directly into program RAM, by using switches on the front panel of the INTELLEC® 4/MOD 40 system. Several steps are required.

The operator sets up a twelve-digit binary address, using the twelve ADDRESS/DATA selector switches on the console panel. He depresses the SEARCH ADDRESS CONTROL: LOAD switch momentarily, to latch this address into the panel logic. The next step is to place the CMA:ENABLE/DISABLE switch in the ENABLE position, and verify that the correct address is displayed on the A_{11} - A_1 indicator lamps.

When he is sure of the correct address, the operator proceeds to set up the data to be written into memory, using ADDRESS/DATA selector switches 0 through 7. To enter this data into memory, he simply depresses the CMA: WRITE switch momentarily. The panel's M_1 and M_2 indicator lamps will register the new contents of the specified memory location.

Finally, the operator returns the CMA:ENABLE/DISABLE switch to the DISABLE position, and the operation is complete.

Refer again to Figure 4-1. The CMA address, from the Control Panel logic is presented to the address multiplexer on the Control Module. The write data from the panel is presented to the write data switch on the Control Module. The timing generator section sends an $\bar{A}_1$ signal to both switches, causing them to gate the panel data through to the RAM Memory Module. Note, however, that this gating is only momentary. It is repeated with each system cycle, but lasts only as long as the $\bar{A}_1$ signal is applied.

The write generator section also receives an $\bar{A}_1$ timing signal from the timing generator. The write generator responds by activating the MOD ENABLE line briefly, specifically enabling the RAM Memory Module during the A_1 interval. This enabling is necessary, because the normal RAM enabling signal from the memory selection logic has been turned off at the panel.

When the operator depresses the CMA:WRITE switch, he applies a CMA WRITE command to the CMA enable section of the Control Module. This signal, in coincidence with the $\bar{A}_1$ signal from the timing generator, turns the CMA enable section on. It remains on during the A_1 interval, until the $\bar{A}_2$ pulse from the timing generator arrives to shut it off. During the time that the CMA enable block is on, it signals the write generator, and the write generator reacts by transmitting a $\overline{WRITE}$ command to the RAM Memory Module. The write generator simultaneously enables the $\overline{BYTE\ 1}$ and the $\overline{BYTE\ 2}$ lines to the module.

The coincidence of the MOD ENABLE signal, the $\overline{WRITE}$ command, and the panel data, all occurring at A_1, causes the RAM Memory Module to record the data. The CMA operation is thus complete.

Transitive Write (WPM)

The WPM instruction, in conjunction with the Control Module, permits the operator to program write operations using program RAM. We should emphasize, however, that the WPM is not the direct equivalent of the WRM used to write data into the 4002 RAM.

The 4002 RAM is a scratchpad memory element associated with the Central Processor Module. It is not to be confused in any sense with the 2102 RAM which is the basic storage element on the RAM Memory Module. The two are entirely different functionally, and different routines are necessary for fetch and store operations.

A three step routine permits the storage of data in the 4002. Typically, that routine would look something like this:

```
LDM     DDDD
SRC     0P
WRM
```

These steps (1) load the constant DDDD into the processor's accumulator, (2) send the contents of the index register pair 0 and 1 to the memory as an address, and (3) order writing of the accumulator's contents into the memory location designated by the SRC. This sequence of steps is so short and simple, that it is more economical to write it into the program whenever required than it would be to provide a separate subroutine, with the necessary jumps and returns.

Suppose, however, that we wish to write an eight-bit data item into program memory. If we assume the following pre-conditions:

a) upper address in index register 0
b) middle address in index register 2
c) lower address in index register 3
d) upper data in index register 4
e) lower data in index register 5

then the routine for writing into program RAM looks like this:

```
FIM     OEH   F0H
SRC     OEH           /DESIGNATE PORT #15
LD      0
WRR                   /UPPER ADDRESS TO PORT #15
SRC     3             /REST OF ADDRESS TO 4289
```

```
LD    4
WPM             /WRITE UPPER BYTE
LD    5
WPM             /WRITE LOWER BYTE
```

The first five steps set up the address, by writing the high-order four bits into output port #15, and by sending the low-order eight bits (in register pair 2 and 3) to the 4289 memory interface via an SRC instruction. The next four-step segment actually writes the two data bytes into the RAM Memory Module.

NOTE

For 4002 RAM elements with date code prior to Nov. 1973:

When using the WPM instruction to load program RAM, it is possible to destroy data previously committed to storage in the 4002 RAM memory. The control logic in the 4002 is incapable of distinguishing between the WPM used to write to program RAM, and the WRM used to write into data RAM. The only way to avoid this consequence is to designate a vacant CM-RAM command line (using a DCL instruction), prior to the loading of program RAM.

A sequence as lengthy as this one, is normally written in the form of a subroutine, and called as needed by the main program. You can easily appreciate the inconvenience and the processing time required to perform such a routine. Because it is a cumbersome process, it is only suitable for the loading of the RAM Memory Module, and for the occasional program changes that are made by using the system software. The INTELLEC® 4/MOD 40 System Monitor, for example, contains such a routine. But the program RAM will not ordinarily be used for the storage of intermediate program results, since the 4002 RAM can serve this purpose far more economically and efficiently.

Refer now to the Functional Block Diagram, Figure 4-1.

When the Central Processor Module executes a WPM instruction, it directs a four-line signal to the Central Module, on the I/O bus as shown in Figure 4-1. The write enable section of the Control Module is equipped to sense the condition, and recognizes it as a command to turn on the write generator section. The write generator thereafter remains enabled, until explicitly turned off again by the write enable.

The twelve-line address presented to the module has a four-line component and an eight-line component.

The eight-line component, containing the low order address digits, goes directly to the address multiplexer on lines MA_0 through MA_7.

The four-line component of the address, the high order four bits, is applied to one input of the A/X multiplexer. The function of the A/X multiplexer was described in Section entitled Reflexive Read. Its action in the case of a transitive operation, however, is opposite to its action during reflexive operations. The PM signal from the 4289 causes the multiplexer to select and forward the four-line output from output port #15 to the address multiplexer section.

We thus have a twelve-line address field applied to the address multiplexer. During PM, this field consists of the four high-order bits contained in the latch of output port #15, plus the eight low-order bits stored in the 4289 when the last SRC was executed. The address multiplexer selects and forwards this data to the RAM Memory Module. It is continuously available to the memory board, during the PM portion of the system cycle.

When the processor encounters a WPM instruction, it responds by placing the contents of its accumulator on the main data bus during X_2. The 4289 Memory interface routes this byte to the Control Module, on the I/O bus, where it is applied to the one side of the write data multiplexer. The action of the write data multiplexer was also described in Section entitled Reflexive Read. This multiplexer selects and forwards the data on the I/O bus, during the A_2 through X_3 phases of the system cycle. The data from the processor's accumulator is thus available to the RAM Memory Module at the same time as the data from the address multiplexer.

The 4289 memory interface on the Central Processor Module contains a circuit known as the F/$\overline{L}$ flip-flop. The output from the F/$\overline{L}$ flip-flop is used to determine whether the first or the last byte will be accessed by any given WPM command. The output of this flip-flop is a single bistable line which is always set to the HIGH condition when power is first applied. The flip-flop is toggled subsequently each time that a WPM is executed, and each succeeding WPM therefore accesses an alternate byte in program RAM. For this reason, WPM instructions **always** occur in pairs. Byte 2 (the most significant 4 bits in the memory location) is written first, and byte 1 follows immediately. This is the only way of avoiding program errors resulting from mistaken assumptions about the state of the F/$\overline{L}$ flip-flop.

The F/$\overline{L}$ signal from the 4289 is applied to the write generator section of the Control Module. The write generator reacts by activating the $\overline{\text{BYTE 2}}$ or the $\overline{\text{BYTE 1}}$ line, as appropriate. The byte selection information will therefore be presented to the RAM Memory Module at the same time as the address data and the write data.

When a WPM occurs, the 4289 responds by sending a PM signal to the write generator section of the Control Module. The PM signal is timed to coincide with the X_2 phase of the cycle. The write generator is enabled by the write enable block, and responds by transmitting a $\overline{\text{WRITE}}$ command to the RAM Memory Module. The write generator simultaneously directs a momentary MOD ENABLE signal to the memory board, and we thus have the following signal coincidence, at X_2:

a) address data from the Central Processor Module
b) write data from the Central Processor Module
c) byte selection information from the Central Processor

d) a MOD ENABLE signal from the Control Module
e) a $\overline{\text{WRITE}}$ command from the Control Module

The RAM Memory Module responds by recording the write data in the specified memory location.

Byte 1 is written in similar fashion. Only the active byte selection line, and the contents of the accumulator, will be different.

Transitive Read (RPM)

The RPM instruction enables the operator to program steps which read the contents of program RAM. The sequence of steps involved is not as simple and direct as is the reading of 4002 RAM, but the RPM is not conceived as a working substitute for the RDM instruction. Rather, it permits the operator to dump the contents of program memory for examination and de-bugging. As with the WPM write sequence, readout is somewhat involved, but it is useful for certain purposes. It allows the software designer considerable flexibility during the early phases of program development. It is much easier to obtain a listing of the contents of control program RAM, than it is to dump a program previously committed to PROM storage.

Assume that index register 0 contains the upper four bits of the program RAM address, and that the middle and lower address bytes are stored in registers 2 and 3, respectively. The programmed segment used to read that memory location would typically look like this:

```
FIM   OEH  F0H
SRC   OEH        /DESIGNATE PORT #15
XCH   0          /UPPER ADDRESS TO PORT #15
WRR              /REST OF ADDRESS TO 4289
SRC   2
RPM              /UPPER BYTE TO ACCUMULATOR
XCH   4          /STORE UPPER BYTE
RPM              /LOWER BYTE TO ACCUMULATOR
XCH   5          /STORE LOWER BYTE
```

The first five steps specify the memory address, in exactly the same way as was done for the write sequence.

The next step, RPM, causes the upper byte of the addressed location to be input to the 4040's accumulator. The XCH 4 instruction transfers the byte from the accumulator to index register 4. A second RPM is subsequently executed, causing the lower byte of the addressed location to be input to the accumulator. The next instruction XCH 5 stores the byte in index register 5.

Refer again to Figure 4-1.

The low-order eight bits of the address are applied to the address A/X multiplexer, just as in the write operation. The four high-order bits are applied to the A/X switch, and gated through the address multiplexer, during PM, just as in the write operation. Both components of the address are gated through the address switch and applied to the RAM Memory Module's inputs during the execution phases of every cycle, just as in the write operation.

When the processor encounters the RPM instruction, it cues the 4289, which sends a PM command to the Control Module's write generator section, just as in the write operation. In the case of a read, however, the write generator receives no $\overline{\text{OUT}}$ signal from the 4289. As a result, no $\overline{\text{WRITE}}$ signal is generated. The write generator does, however, send a MOD ENABLE signal to the RAM Memory Module. And the memory module responds, in the absence of an explicit $\overline{\text{WRITE}}$ command, by depositing the contents of the requested location of its eight output lines. The 4289 accepts the data from the memory module and, in turn, furnishes it to the 4040, based on the state of the F/$\overline{\text{L}}$ flip-flop.

Miscellaneous Control Functions

Miscellaneous functions performed by the Control Module include:

a) program memory selection
b) RESET generator
c) TEST generator

The memory selection section of the Control Module receives its input signals from the MODE CONTROL selector switches on the INTELLEC® 4/MOD 40's front panel. It produces a 3-line output, consisting of a $\overline{\text{RAM}}$ line, a $\overline{\text{PROM}}$ line, and a $\overline{\text{MON}}$ line. Each of the three lines is logically exclusive of the others. The activated line selects the program memory bank to be used by the Central Processor Module.

The RESET generator receives its input from the RESET switch on the front panel of the INTELLEC® 4/MOD 40 system. It also receives an input from each of the MODE CONTROL selectors. Actuation of the manual RESET, or the re-selection of program memory, thus triggers the RESET generator.

There are two kinds of resets, as determined by the position of the RESET CONTROL: MODE selector on the Control and Display Panel. When the selector is in the SYSTEM position, a $\overline{\text{4002 RESET ENABLE}}$ signal is applied to the RESET generator section. Under these circumstances, actuation of the RESET switch causes a complete system reset, including the clearing of the 4002 memory on the Central Processor Module. When the MODE selector is in the CPU position, however, the 4002s are specifically exempted from the general RESET applied to the rest of the system, thus saving all data in transient storage.

The RESET clears the write enable section of the Control Module, as well as the panel logic and the circuitry on the Central Processor Module.

The STOP section of the Control Module receives inputs from the RUN/HALT and SINGLE STEP switches on the system's Control and Display Panel. When the two-position RUN/HALT selector is in HALT, it clamps the corresponding control input of the Central Processor Module's 4040 CPU, inducing a pseudo-halt in which the processor executes the "no operation" (NOP) continuously.

The SINGLE STEP is a momentary switch which may be used to interrupt the HALT command briefly. This permits the operator to execute his program one step at a time and examine the results displayed on the console indicator. Stepping through the program in this manner, the operator can determine the exact effect of each and every instruction.

THEORY OF OPERATION

This section explains the detailed theory of operation of each of the blocks described on page 35. In the intended order of discussion, these blocks are:

a) timing generator
b) address multiplex
c) write data multiplex
d) A/X multiplex
e) port decoder
f) input ports and latch
g) output port and latch
h) write enable
i) CMA enable
j) write generator
k) memory selection
l) RESET generator
m) STOP generator

Timing Generator

Refer to the Control Module schematic, Figure 4-2.

The timing generator consists of a 74161 binary counter (A23), a 3205 3-to-8 line decoder (A15), and the miscellaneous gates and inverters shown immediately to the left of these two elements on the module schematic.

The timing generator receives $\overline{01}$, $\overline{02}$, and $\overline{SYNC}$ signals from the Central Processor Module, and uses these to develop all the internal timing signals for the Control Module. Its outputs include: $\overline{01B}$, $\overline{02B}$, $\overline{EOC\ SYNC}$, $\bar{A}_1$, $\bar{A}_2$, and $\overline{X_1}$.

The $\overline{01}$ signal from the Central Processor Module passes through three inverters on the Control Module, and is applied to the CLOCK input of the 74161. The counter registers successive pulses, until it is reset, producing a four-line output.

The $\overline{SYNC}$ and the $\overline{02}$ inputs each pass through a pair of inverters of co-inputs of a 2-input negative-true AND-gate. The coincidence of the two signals resets the 74161, in the latter third of the X_3 interval. The counter thus has eight discrete states, reflected by the levels on the three output lines 14-15-16. The output at pin 11 will never go high, since the counter's internal state never exceeds binary "7".

The output lines from the counter are applied directly to the three input lines of the 3205 decoder. The "D" output from the counter goes to the ENABLE inputs at A15-4 and A15-5. Since the "D" line will always be low, as just explained, the decoder will be continuously enabled under all normal circumstances.

The 3205 decoder produces an eight-line sequential output, corresponding to the eight phases of the system cycle. Only the $\bar{A}_1$, $\bar{A}_2$, and $\overline{X}_1$ outputs are necessary to to the module's operation.

Note that the signal used to reset the 74161 is routed out of the board on pin J2-31. This negative-going pulse is labeled $\overline{EOC\ SYNC}$ and is directed to the Console and Display Panel logic circuitry. The 01B and the 02B signals are derived by inversion and these two are transmitted to the panel logic.

Address Multiplexer

The address multiplexer consists of the three 8-to-4 line multiplexers A17, A26, and A25, shown in the upper right hand corner of the module schematic. The multiplexers used are type SG8266, in which four of the inputs are inverted with respect to the outputs.

Each of the multiplexers receives a four-line data byte from the Central Processor Module on its inverting inputs, and a corresponding byte from the panel address switches on its remaining four inputs. The four-line output from each is transmitted to the RAM Memory Module.

The only control input to the multiplexers is the $\bar{A}_1$ signal from the timing generator section of the module. The design of the multiplexers is such that they select the 1-6-10-15 inputs when pin 9 is HIGH, and the 2-5-11-14 inputs when it is LOW. The $\bar{A}_1$ signal is LOW during A_1, and HIGH during the rest of the cycle. Thus, the address multiplex forwards the address data from the panel during A_1 and the data from the Central Processor Module at all other times.

Write Data Multiplexer

The write data multiplexer forwards the data from the I/O bus, or from the Console and Display Panel bus, to the RAM Memory Module's data inputs.

The write data multiplexer consists of the two SG 8266 multiplexers, A3 and A4, which are identical to the multiplexers used in the address switch section.

Like the address multiplexer, the write data multiplexer receives the $\bar{A}_1$ output from the timing generator section. This signal, applied to the pin 9 inputs of the multiplexers, causes them to select panel data during A_1, and data from the Central Processor Module during the remainder of the cycle.

A/X Multiplexer

The A/X multiplexer selects either the contents of the port #15 output latches, or the contents of the chip select bus (C_0 - C_3), and forwards this data to the address switch section. Selection depends upon the phase of the system cycle in progress.

The A/X multiplexer consists of the SG 8233 multiplexer (A18) shown on the module schematic. Note that the SG 8233 has no inverting inputs, and is therefore distinguished from the SG 8266 elements used in the address and

write data multiplexer. Its operation, however, is similar. It selects 1-6-10-15 when pin 9 is HIGH; 2-5-11-14, when pin 9 is LOW.

The A/X multiplexer receives PM from the 4289 (PM occurs during execution of a WPM or RPM instruction). The PM output is applied directly to the pin 9 input of the A/X multiplexer, causing the multiplexer to select and forward latch data during the execution phase, and chip select data during the address phase.

Port Decoder

The port decoder receives four-line data from the Central Processor Module, via the chip select bus (C_0 - C_3). Information on this bus specifies the most significant address byte. The Control Module uses the information to selectively enable I/O port #14 or I/O port #15, as called for by the address. This is the function of the port decoder.

The port decoder consists of A27, a 3205 3-to-8 line converter. Observe that the C_3 line of the chip select bus is tied to A27-6. Pin 6 is an enabling input, which must be HIGH in order for the decoder to operate. Since the C_3 line will always be HIGH, when port #14 or port #15 is designated, this precondition is fulfilled. PM disables the decoder (at pins 4 and 5) so that output port #15 is not written into during execution of a WPM instruction.

The remaining three chip select lines are directed to the decoder's data inputs. The circuit responds by clamping output pin 9 LOW whenever port #14 is selected, or pin 7 LOW whenever port #15 is selected.

Input Port and Latch

The input port and latch allow transitive read operations using the WPM instruction instead of RPM. This enable system software (e.g., the Monitor) to be compatible with both INTELLEC® 4/MOD 4 and INTELLEC® 4/MOD 40 systems (the 4004 used in the INTELLEC® 4/MOD 4 does not include RPM in its instruction set).

Read data from the RAM Memory Module is transferred to the processor in two stages. First the data is latched into a pair of four-bit latches on the Control Module. Then two successive ROM READ instructions (RDR) transfer the data to the processor itself. For this purpose, two quad-latches and two simulated ROM ports are provided on the Control Module.

The data latches A10 and A11 are 3404 hex inverting latches. Four of the latches on each chip are tied to a common $\overline{\text{STROBE}}$ input, at pin 7. The four corresponding data inputs to each chip are connected to the MDI_0 through MDI_7 input lines from the RAM Memory Module, as shown in the upper left hand corner of the Control Module schematic.

The $\overline{\text{STROBE}}$ inputs of both quad sections are tied in common to the write generator section. When the write generator receives a PM command from the Central Processor Module, indicating that a WPM is in progress, it places a negative-going pulse on this line. The $\overline{\text{STROBE}}$ causes the latches to record the data from the memory and to hold it in inverted form at the eight output pins. Byte 2 is registered in A11 and byte 1 is registered in A10.

The simulated input ports consist of the SG 8233 multiplexer (A12), the four buffer-gates of A20 shown to the immediate right of the multiplexer, the 11-12-13 section of A5 shown just below the port decoder, and the 4-5-6 section of A21 shown at the extreme left of the schematic.

The 4289 Memory Interface on the Central Processor Module sends an $\overline{\text{IN}}$ signal to the input ports whenever a ROM read is executed. These ports also receive the $\overline{14}$ and the $\overline{15}$ signals from the port decoder section. The addressed port responds to these inputs, by placing the appropriate byte on the I/O bus, when so instructed by the processor.

The A12 multiplexer is identical to that used in the A/X switch section. The four-line output of each of the input latches is applied to one side of the multiplexer. A12-9 is tied to the $\overline{14}$ output from the port decoder section. The multiplexer thus selects the byte 2 output of latch A11, when the chip select lines specify port #14. When port #15 is selected, the pin 9 input to the multiplexer will be HIGH, causing the multiplexer to forward the byte 1 output of latch A10 to its four output lines. The selected byte is directed to the four gated buffers of A20, as shown in Figure 4-2.

The $\overline{\text{IN}}$ signal from the Central Processor Module is applied to A21-4. A21-5 receives its input from the $\overline{14}$ and the $\overline{15}$ lines of the port decoder, through the 11-12-13 section of NAND-gate A5. A21-6 will therefore reveal a coincidence whenever an $\overline{\text{IN}}$ and a $\overline{14}$ or a $\overline{15}$ occur simultaneously. The resulting LOW is applied to the enabling inputs of the four A20 buffer-gates, during the X_2 phase of the cycle. The buffers react by gating the contents of the selected port through to the I/O bus.

Output Port and Latch

A simulated ROM output port, identified as port #15, is also provided on the Control Module. This port is used to latch and hold the high order byte of the object address, for presentation to the A/X switch during transitive read and write operations.

Two of the latches on A10, and two on A11 are superfluous to the operation of the input port. These serve output port functions. Their inputs are connected to the four lines of the I/O bus, through four cascaded inverter pairs. Details of the connection are shown at top center of the Control Module schematic. The four inverted outputs of the latches are connected to one side of the A/X multiplex, A18.

The OUT signal from the Central Processor Module, and the $\overline{15}$ output from the Control Module's timing generator are applied to pins 12 and 13 of A21, a negative-true AND-gate. The output of the gate, which is LOW at coincidence, is used to pulse the STROBE inputs of the two dual latches, A10 and A11. The latches thus register and hold any data that the processor directs to port #15.

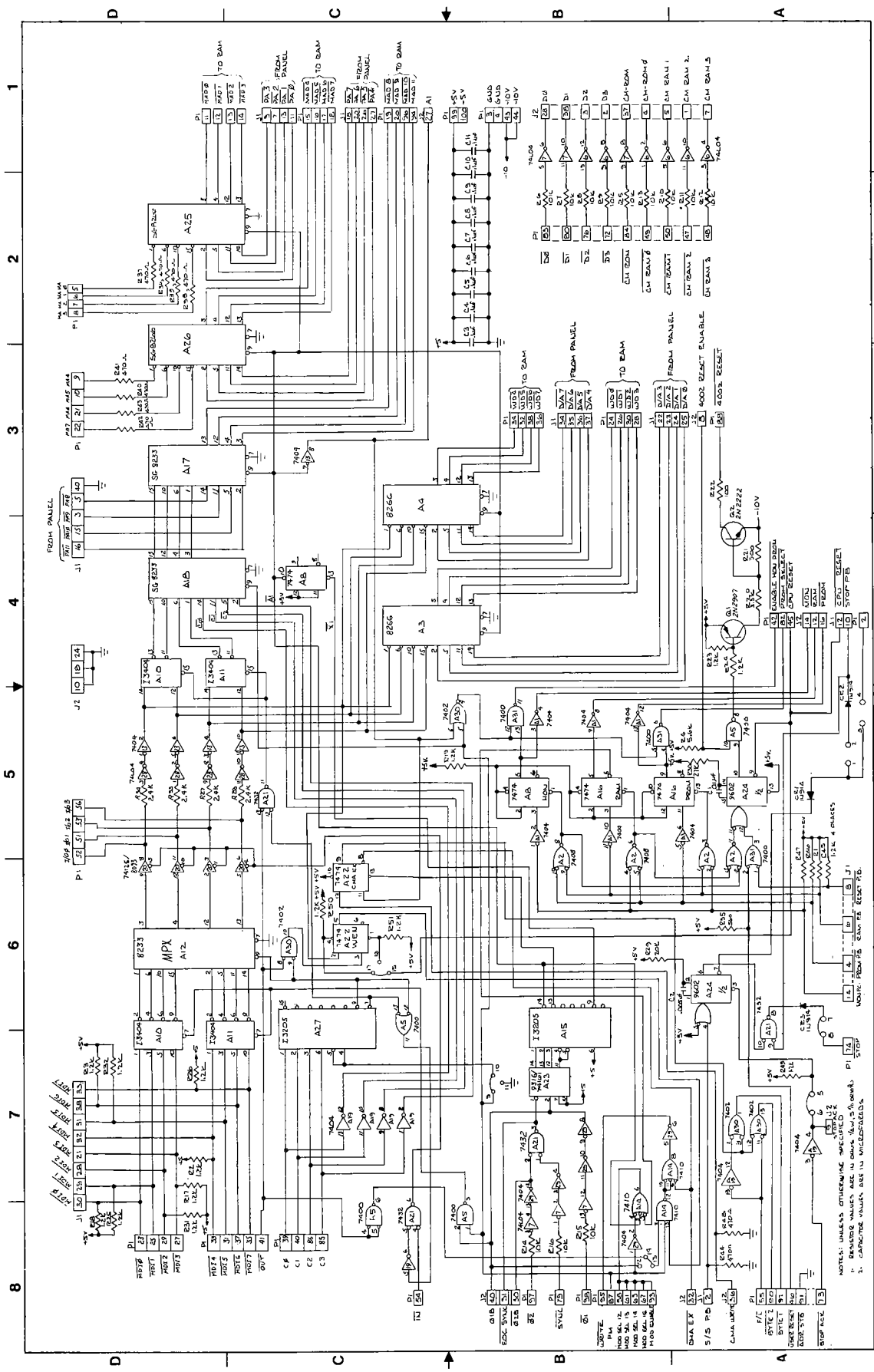


Figure 4-2. Control Module Schematic

Write Enable

This section allows transitive read operations using the WPM instruction, instead of RPM. This enables system software (e.g., the Monitor) to be compatible with both the INTELLEC® 4/MOD 4 and INTELLEC® 4/MOD 40 systems.

The write enable section consists of the 1-2-3-4-5-6 section of the 7474 dual D flip-flop (A22), together with the 8-9-10 section of NOR-gate A30.

Pins 8 and 9 of the A30 receive the $\overline{OUT}$ command from the Central Processor Module, and the $\overline{14}$ from the timing generator section, respectively. A 30 is employed as a negative-true coincidence indicator, and it transmits a HIGH to A22-3 when the two input signals concur.

A22-3 is the CLOCK input of the 7474. Pin 2, the D input, is attached to the I/O_0 line. The coincidence of an $\overline{OUT}$, a $\overline{14}$, and a "XXX1" on the I/O bus therefore sets the write enable flip-flop. Conversely, when the I/O bus contains "XXX0" concurrently with an $\overline{OUT}$ and a $\overline{14}$, the flip-flop will reset. Observe also, that the preset input, pin 4, is tied (through a jumper) to the RESET section. A general reset will therefore set the write enable.

The output of the write enable is a single line to the write generator, HIGH as long as the write enable switch is turned on.

CMA Enable

The CMA enable generator consists of one-half of a 7474 dual D flip-flop, A22, which is labeled CMA EX on the Control Module schematic.

The CMA enable section receives the A_1 output of the timing generator, applied to its CLOCK input, pin 11. It also receives a CMA WRITE command, which originates in the Control Panel logic when the CMA:WRITE switch is actuated.

The coincidence of these two signals SETS the flip-flop, and it remains SET until the $\bar{A}_2$ output of the timing generator arrives at its CLEAR input, pin 13.

The outputs of the section consist of positive- and negative-going pulse phases, coincident with the A_1 portion of the CMA write cycle. The positive-going pulse is applied to the byte selection section of the write generator, causing both the $\overline{BYTE\ 2}$ and the $\overline{BYTE\ 1}$ lines to be enabled for the period of the CMA transaction. The negative-going pulse output goes to A14-9 of the write generator section, where it triggers a $\overline{WRITE}$ command.

Write Generator

The write generator consists of three subsections, which:

a) generate the MOD ENABLE
b) generate $\overline{BYTE\ 2}/\overline{BYTE\ 1}$ selection
c) generate the $\overline{WRITE}$ command, when required

The MOD ENABLE section consists of a negative-true three-input NAND-gate, A14-3-4-5-6, and an inverter, A29-1-2. The gate is used as a negative-true OR. As shown on the schematic, the MOD ENABLE output will be HIGH whenever one of three input conditions occurs: (1) $\bar{A}_1$, (2) $\overline{RAM}$, indicating that the RAM Memory Module is selected at the Control Panel MODE switch, or (3) PM, indicating that a RPM or WPM is in progress. Any one of these conditions calls for a response on the part of the memory module, and the MOD ENABLE line to the module is activated accordingly.

Byte selection is performed by two NOR-gates and an inverter, shown in the lower left corner of the schematic. This network receives the $F/\overline{L}$ signal from the Central Processor Module, and translates the bistable input to a two-line output. Note that the wiring of the gates permits the CMA enable section to select **both** lines during console memory access operation.

Two three-input NAND-gates generate the $\overline{WRITE}$ command. They are the 1-2-12-13 and the 8-9-10-11 sections of A14, shown at the left side of the module schematic. These gates are wired to produce a HIGH at A14-8, whenever one of two conditions occurs:

a) A CMA EX, or
b) the coincidence of PM, $\overline{OUT}$ and a write enable "on" condition

The HIGH at the output is inverted in A29-5-6 and routed to the memory module, as the $\overline{WRITE}$ command.

Memory Selection

The memory selection section of the Control Module consists of three bi-stable latches, and their associated logic. The bi-stable elements are contained on A16, as A8 (1-2-3-4-5-6), as shown in the bottom center of the Control Module schematic, Figure 4-2.

The latch elements are type 7474 dual D flip-flops with their D inputs tied to the +5 volt supply. A positive-going transition applied to the CLOCK input of any of the three flip-flops therefore causes its Q output to go HIGH.

This network receives three input lines, one each from the MON, PROM, and RAM pushbuttons on the INTELLEC® 4/MOD 40 front panel. Each line is applied, through an inverter, to the CLOCK input of one of the flip-flops. The line is also directed through two OR-gates, to the CLEAR input of the two remaining flip-flops. Thus, the momentary ground resulting from actuation of any one of the MODE selector switches sets one of the flip-flops and clears the other two.

The Q output of each of the flip-flops is inverted and used to enable one of the program memory banks. The three lines, $\overline{MON}$, $\overline{PROM}$, and $\overline{RAM}$, are logically exclusive, with the result that only one memory bank will be enabled at any given time.

The $\overline{Q}$ output from the RAM flip-flop goes to A14-3, in the write generator section, where it is used to develop the MOD ENABLE signal for the RAM Memory Module.

Reset Generator

The RESET generator consists of one-half of a monolithic dual one-shot, type 9602, together with several associated gates. These are shown immediately below the memory selection section on the Control Module schematic. The multivibrator is labeled A24-9-10-11-12-13-14-15 on the diagram.

As shown, the RESET one-shot will be triggered whenever the RESET pushbutton on the console panel is depressed. The momentary ground on the $\overline{\text{RESET PB}}$ line is directed to the Control Module via connector J1. Observe that the multivibrator will also be triggered by a LOW applied to P1-46, labeled $\overline{\text{USER RESET}}$. This line is connected through the system's mother board to J3-43 on the rear panel of the unit. It is available for remote control or other special system configurations.

Note that the RESET one-shot receives still a third actuating signal. The manual operation of any of the three MODE selectors on the Control Panel will cause a momentary LOW at one of the two OR-gates A2-4-5-6 or A2-8-9-10, and the resulting LOW at the output of either gate will in turn cause A2-11 to go LOW. This pin is connected to the negative-true trigger input of the RESET one-shot. Thus any change in the selection of program memory will automatically reset the system.

The RESET generator provides for two kinds of RESET. When the SYSTEM RESET mode is selected, a LOW is applied to A5-10, defeating coincidence, and preventing the RESET at A5-9 from being coupled through. In the CPU RESET mode, however, A5-10 is allowed to swing HIGH, and the positive-going RESET pulse at A24-10 is coupled through to the base of Q1. Q1 and Q2 shift the TTL output of A5 to MOS levels, for application to the 4002 RAMs on the Central Processor Module.

Regardless of the position of the RESET MODE selector, the negative-going pulse output at A24-9 is coupled to the TTL circuitry of the panel logic and the Central Processor Module. Duration of the RESET pulse is 500 microseconds (minimum).

Stop Generator

The stop generator portion of the Control Module consists of the 1-2-3-4-5-6-7 section of A24, one-half of a dual one-shot multivibrator, and the dual-input OR-gate section A21-8-9-10. These components are shown in the lower left-hand corner of the Control Module schematic, Figure 4-2.

When the RUN/HALT switch on the system's console is in the HALT position, a LOW applied to A21-8 through contacts of the switch coincides with the quiescent LOW at the Q output of A24 (pin #6) to produce a $\overline{\text{STOP}}$ command at P1-74 of the module. This is forwarded to the Central Processor Module, where it induces a pseudo-halt state in the 4040 CPU. The CPU remains in the halted state, until such time as the clamp on its $\overline{\text{STOP}}$ command line is lifted.

Single-step operation is obtained by triggering the one-shot multivibrator. Actuation of the SINGLE STEP switch on the Control and Display panel produces a momentary HIGH at A24-4. The firing of the multivibrator momentarily interrupts the enabling level at A21-10, inhibiting the gate and lifting the clamp on the STOP command line. When this occurs the 4040 CPU begins the fetch of the next instruction in the program sequence.

As soon as the processor module exits the halt state, its $\overline{\text{STOP ACKNOWLEDGE}}$ output goes HIGH. This transition is coupled to P1-73 of the Control Module, where it is inverted and applied to A24-3, promptly resetting the one-shot and restoring the LOW on the $\overline{\text{STOP}}$ command line. As a result, the CPU re-enters the halted state as soon as it completes execution of the current program instruction.

Use with INTELLEC® 4/MOD 40 SYSTEM

When the Control Module is used in INTELLEC® 4/MOD 40 systems, various jumper connections must be changed. The HALT function is disabled by deleting the wire jumpers between pads 5-6 and 7-8 on the printed circuit board. Wire jumpers 1-2 and 3-4 are installed in their stead, to enable the TEST function. The TEST function is not used in INTELLEC® 4/MOD 40 systems. In addition, wire jumpers between pads 9-10, 13-14 and 15-17 are deleted. Instead wire jumpers 10-11, 12-14 and 15-16 are installed respectively. These jumper connections allow use of the 4289 interface.

UTILIZATION

This section provides information on utilization of the Control Module.

Installation

In installing the Control Module, the user must take account of:

a) environmental extremes
b) mounting
c) electrical connections
d) power requirements
e) signal requirements

ENVIRONMENT

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conductive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from components on the card.

Relative humidity is not critical to the module's operation.

MOUNTING

Avoid locating the card near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections. The result might be abnormally high noise levels, with the problems entailed, or outright failure of the card.

Dimensions of the module are 6.18 x 8.00 inches. Be sure to allow enough additional clearance to ensure adequate cooling.

The card is designed to plug directly into a standard 100-pin, double-sided PC edge connector. The connector will serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the card be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the card, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the elements on the module.

ELECTRICAL CONNECTION

The basic power and control connections to the Control Module are effected by means of a standard 100-pin, double-sided PC edge connector. CDC #VPB 01C50E 00A1 is one suitable type. Pin allocations on this connector are given in Section entitled Pin List.

Connections to the INTELLEC® 4/MOD 40 Control and Display Panel are made via two miniature 50-pin connectors mounted on the board. The appropriate mating connector is 3M #3433-1002. Pin allocations are given in Section entitled Pin List.

POWER REQUIREMENTS

The Control Module requires DC power, at the following levels:

+5 V ± 5%	@	1.5	Amperes (max)
	@	0.7	Amperes (typ)
−10 V ± 5%	@	0.1	Amperes (max)

Refer to the Pin List for the power connections.

SIGNAL REQUIREMENTS

Inputs and outputs on the module, both data and control, are generally at TTL levels. The sole exception is the 4002 RESET, which is an MOS level output.

Signal descriptions, and connector pin allocations, are given in Tables 4-1 through 4-3.

Pin List

The following section describes connector pin allocations on the Control Module. The pins and their signal functions are listed in Tables 4-1 through 4-3;

P1 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1		
2	$\overline{\text{TEST}}$[2]	CPU Module
3	GROUND	
4	GROUND	
5	MEMORY ADDRESS Ø	From CPU Module
6	MEMORY ADDRESS 1	
7	MEMORY ADDRESS 2	
8	MEMORY ADDRESS 3	
9	MEMORY ADDRESS 4	
10	MEMORY ADDRESS 5	
11	$\overline{\text{MEMORY ADDRESS Ø}}$	To RAM Module
12	$\overline{\text{MEMORY ADDRESS 1}}$	
13	$\overline{\text{MEMORY ADDRESS 2}}$	
14	$\overline{\text{MEMORY ADDRESS 3}}$	
15	$\overline{\text{MEMORY ADDRESS 4}}$	
16	$\overline{\text{MEMORY ADDRESS 5}}$	
17	$\overline{\text{MEMORY ADDRESS 6}}$	
18	$\overline{\text{MEMORY ADDRESS 7}}$	
19	$\overline{\text{MEMORY ADDRESS 8}}$	
20	$\overline{\text{MEMORY ADDRESS 9}}$	
21	MEMORY ADDRESS 6	From CPU Module
22	MEMORY ADDRESS 7	
23	$\overline{\text{MDI 0}}$	RAM Module and Console
24	$\overline{\text{WRITE DATA Ø}}$	To RAM Module
25	$\overline{\text{MDI 1}}$	RAM Module and Console
26	$\overline{\text{WRITE DATA 1}}$	RAM Module
27	$\overline{\text{MDI 3}}$	
28	$\overline{\text{WRITE DATA 3}}$	
29	$\overline{\text{MDI 2}}$	
30	$\overline{\text{WRITE DATA 2}}$	
31	$\overline{\text{MDI 5}}$	
32	$\overline{\text{WRITE DATA 5}}$	
33	$\overline{\text{MDI 4}}$	
34	$\overline{\text{WRITE DATA 4}}$	
35	$\overline{\text{MDI 7}}$	
36	$\overline{\text{WRITE DATA 7}}$	
37	$\overline{\text{MDI 6}}$	
38	$\overline{\text{WRITE DATA 6}}$	
39	CHIP SELECT Ø	CPU Module
40	CHIP SELECT 1	
41	$\overline{\text{OUT}}$	CPU Module
42	$\overline{\text{ENABLE MON PROM}}$	CPU Module
43	-10 VDC	Power Supply
44	-10 VDC	
45	$\overline{\text{CPU RESET}}$	CPU Module
46	$\overline{\text{USER RESET}}$	Rear Panel Input Connector
47	$\overline{\text{CM-RAM 2}}$	CPU Module
48	$\overline{\text{CM-RAM 2}}$	
49	$\overline{\text{CM-RAM Ø}}$	
50	$\overline{\text{CM-RAM 1}}$	

Table 4-1.

NOTES:

[1] All signal inputs and outputs are at TTL levels, except as otherwise noted.

[2] Not used in INTELLEC® 4/MOD 40 systems.

P1 Pin List (Continued)

PIN	SIGNAL FUNCTION	DESTINATION
51	I/O 1	
52	I/O Ø	CPU Module
53	I/O 2	
54	$\overline{IN}$	CPU Module
55	F/$\overline{L}$	CPU Module
56	I/O 3	CPU Module
57		
58	MOD SELECT 12	
59		
60		
61	MOD SELECT 13	
62		
63	MOD SELECT 14	RAM Module
64		
65		
66		
67	MOD SELECT 15	
68		
69		
70		
71		
72	$\overline{DATA\ 3}$	CPU Module
73	STOP ACKNOWLEDGE[3]	CPU Module
74	STOP	CPU Module
75		
76		
77		
78		
79	$\overline{SYNC}$	CPU Module
80	$\overline{DATA\ 1}$	
81		
82	$\overline{PROM\ SELECT}$	CPU Module
83	$\overline{DATA\ Ø}$	CPU Module
84	$\overline{CM\text{-}ROM}$	CPU Module
85	CHIP SELECT 3	CPU Module
86	CHIP SELECT 2	
87	PM	CPU Module
88	$\overline{4002\ RESET}$	CPU Module
89		
90	$\overline{BYTE\ 2}$	
91	$\overline{ADDRESS\ STROBE}$	
92	$\overline{BYTE\ 1}$	
93	MOD ENABLE	RAM Module
94	$\overline{MEMORY\ ADDRESS\ 11}$	
95	$\overline{WRITE}$	
96	$\overline{MEMORY\ ADDRESS\ 10}$	
97	$\overline{Ø2}$	CPU Module
98	$\overline{Ø1}$	
99	+5 VDC	Power Supply
100	+5 VDC	

Table 4-1. (Continued)

NOTES:

[1] All signal inputs and outputs are at TTL levels, unless otherwise specified.

[2] Not used in INTELLEC® 4/MOD 40 systems.

[3] INTELLEC® 4/MOD 40 function.

J1 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1		
2	S/S PB (TEST PB[3])	
3	PANEL ADDRESS 9	
4	PROM P.B.	
5	PANEL ADDRESS 8	
6	RAM P.B.	
7	PANEL ADDRESS 2	
8	RESET P.B.	
9	PANEL ADDRESS 3	
10	STOP PB (TEST HOLD[3])	
11	PANEL ADDRESS 0	
12	PANEL RESET	
13	PANEL ADDRESS 1	
14	MONITOR P.B.	
15	PANEL ADDRESS 10	
16	PANEL ADDRESS 11	
17		
18	PANEL ADDRESS 7	
19		
20	PANEL ADDRESS 6	Control and Display Panel
21	MDI 3	
22	DATA/ADDRESS 3	
23	DATA/ADDRESS 2	
24	DATA/ADDRESS 1	
25	DATA/ADDRESS 0	
26	PANEL ADDRESS 5	
27	PANEL ADDRESS 4	
28	MDI 2	
29	MDI 1	
30	MDI 0	
31	MDI 5	
32	MDI 4	
33	MDI 7	
34	DATA/ADDRESS 7	
35	DATA/ADDRESS 6	
36	DATA/ADDRESS 5	
37	DATA/ADDRESS 4	
38	MDI 6	
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		

Table 4-2.

NOTES:

[1] All signal inputs and outputs are at TTL levels, unless otherwise specified.

[2] Not used in INTELLEC® 4/MOD 40 systems.

[3] INTELLEC® 4/MOD 40 function.

J2 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1	CM-RAM 2	Control and Display Panel
2	DATA 3	
3	DATA 2	
4	CM-RAM 0	
5	CM-RAM 1	
6		
7	CM-RAM 3	
8	$\overline{\text{4002 RESET ENABLE}}$	
9	STOP ACK	
10		
11		
12	$\overline{\text{RAM}}$	
13		
14	$\overline{\text{MON}}$	
15		
16	$\overline{\text{PROM}}$	
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36	CMA WRITE	Control and Display Panel
37	CM-ROM	
38	DATA 1	
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		

Table 4-3.

NOTE:
[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

CHAPTER 5 THE CONTROL AND DISPLAY PANEL

The imm4-74 Control and Display Panel monitors and controls the operation of the INTELLEC® 4/MOD 40 system. In addition to the routine functions of power control, reset, and memory selection, the control panel contains a number of special purpose controls and displays, designed to simplify program development and debugging.

In the search mode, the panel display indicates the contents of the system's main data bus during each of the eight phases in a specified program cycle. The instruction address and the number of passes of that location are selected by means of switches on the panel. Other indicators display the most recent I/O pointer (SRC) and the status of the processor's memory command lines (CM-ROM and CM-RAM).

In the RUN mode, the panel display does not latch up at the search address. Rather, the logic generates a synchronizing pulse for external instrumentation, each time the search address comes up.

The CMA mode permits the operator to examine and re-write the contents of any location in program RAM, using switches on the console panel.

Included on the panel are HALT and SINGLE STEP functions that permit the operator to execute his program one instruction at a time. Since the processor's current status can be displayed simultaneously on the console indicators, this mode is particularly useful for program debugging.

For a complete description of the panel's controls and indicators, consult the INTELLEC® 4 Operator's Manual.

FUNCTIONAL DESCRIPTION OF THE CONTROL & DISPLAY PANEL

Figure 5-1 shows the relationship among the major functional sections of the Control and Display Panel.

The control panel has three operational modes:

a) search
b) run
c) console memory access (CMA)

We shall first describe the timer and the display latches, since these blocks are basic to all panel modes. We then discuss each of the modes, in turn.

Timer

The Control Panel receives three secondary system timing signals, from the Control Module: $\overline{01B}$, $\overline{02B}$, and $\overline{\text{EOC SYNC}}$. These are applied to the timer logic on the panel.

From these inputs, the timer derives all the synchronization and timing signals necessary for the operations of the panel logic. The output of the timer consists of seven lines, each corresponding to one of the phases in a system cycle. $\overline{A}_1$, $\overline{A}_2$, $\overline{A}_3$, $\overline{M}_1$, $\overline{M}_2$, $\overline{X}_2$, and X_3 outputs are provided. Note that no X_1 signal is generated, or required. Each of the timer's output lines is activated during a different portion of the cycle, to cue the circuitry that must operate in that phase.

In referring to the block diagram, observe that the timer receives an enabling signal from the timer control section. The timer control starts the timer immediately following panel reset, stopping it again when certain logical conditions are fulfilled. The conditions under which stoppage occurs depend upon the panel's operating mode.

Display Latches

There are five sections to the display latch. They are:

a) address display
b) instruction display
c) execution display
d) pointer display
e) active bank display

Each of these sections receives multiple-line data input and a latching control signal. When the latch control line is activated, the instantaneous data at the input to the latch is stored and held for display. Corresponding output lines from each of the latch sections are applied to indicator lamps on the front panel. The lamps thus display the machine's inter-

nal state, at specific times and under specific logical conditions.

The address display latch receives its data input from a 24-line to 12-line multiplexer, shown on the functional block diagram as the CMA address multiplexer. The multiplexer is bi-stable and selects one of two 12-line data inputs for presentation at its outputs, in response to the controlling level applied to the switch. In the ordinary search mode, the address display latch receives data from the system's main bus. But when a CMA operation is in progress, the latch input is connected to switches on the Control Panel, enabling the operator to determine manually the object address.

The instruction display latch receives its data input from a 16-line to 8-line multiplexer, labeled CMA instruction multiplex in Figure 5-1. The CMA instruction multiplex connects the instruction display latch to the system's main bus when the panel is used in the search mode. In the CMA mode, the instruction display latch inputs are connected instead to the eight-line output bus from the RAM Memory Module.

The execution display latch receives a 4-line data input directly from the system's main data bus. The pointer display latch receives a similar input.

The active bank display latch receives a 4-line input from the Central Processor Module, consisting of CM-RAM command lines 0-3.

Several of the display latches thus share a common data input, in certain modes of operation. But the data they forward to the display depends upon the timing of the strobes that the latches receive. The strobe timing depends, in turn, on logical conditions established by the particular mode of operation.

Search Mode

In the search mode, the console displays the contents of the main data bus during each of the significant phases of a particular system cycle. The operator specifies the instruction address by setting up its binary location number on the front panel ADDRESS switches. The operator also specifies the number of passes that occur prior to latching, using PASS COUNTER selection switches on the front panel. When in the course of a run, there is a coincidence in the search address and the preset number of passes of that address, the display latches store and hold the main data bus contents for presentation to the display indicators.

When the search conditions have been set up on the panel selector switches, the operator presses the LOAD button momentarily. This does two things. First, it loads the data from the panel selector switches into storage circuits in the panel logic. Then it initiates a reset of the panel logic (NOTE: this is not the same as a system RESET). The search begins, and the display locks up on the preselected cycle. Note that the machine executes an instruction **before** testing

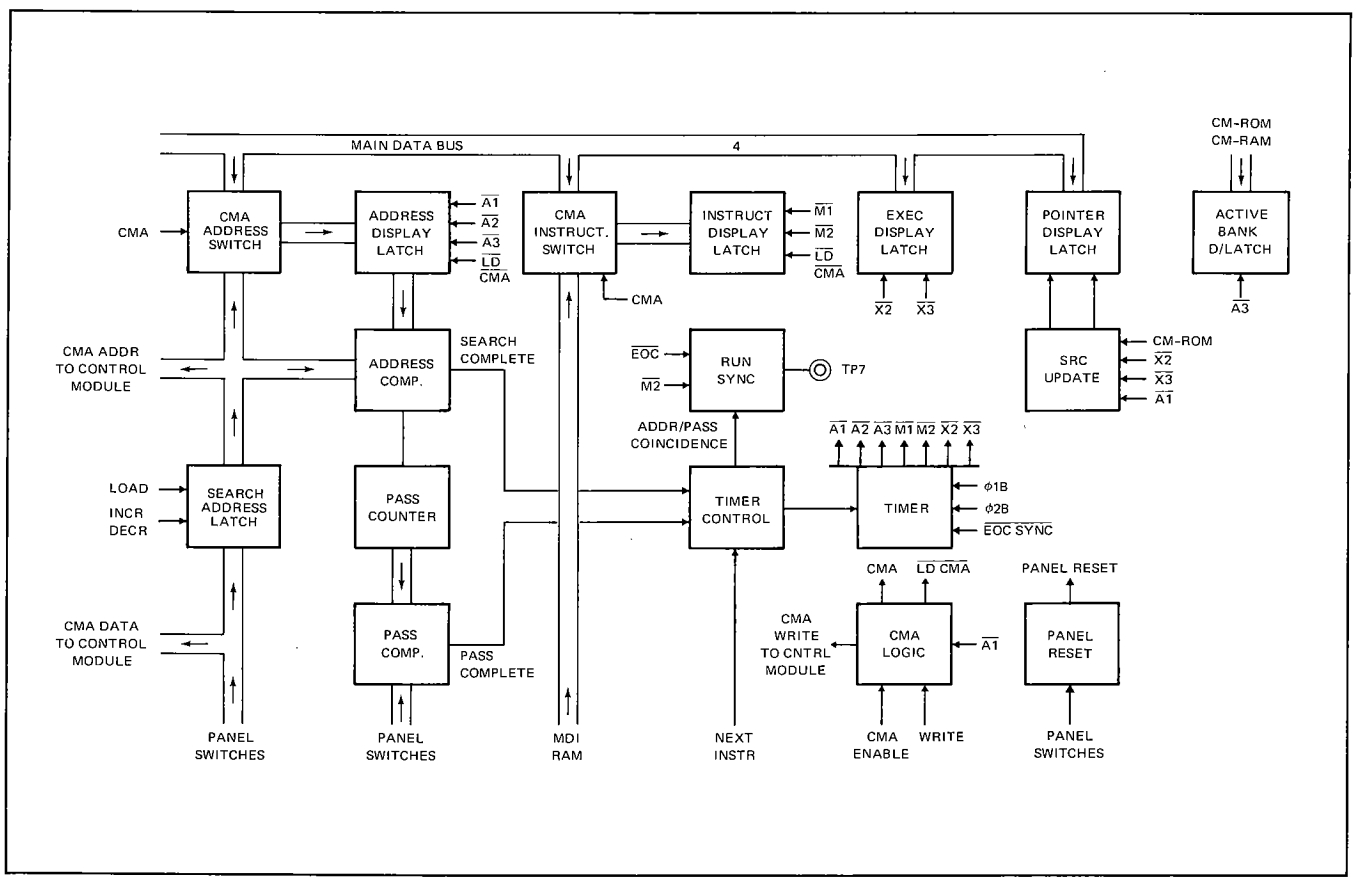


Figure 5-1. Control & Display Panel Functional Block

for pass coincidence. Thus the system actually executes **one more** than the preset number of passes before latching its display.

The pointer display latch holds the contents of the main data bus during X_2 and X_3 of the cycle in which the latest SRC instruction was executed, identifying the object port of a subsequent I/O instruction, or the destination address of a RAM read (RDR) or RAM write (WRM) instruction. (NOTE: RAM here refers to 4002 RAM, not to program RAM).

The address display latch contains three sections. Each section controls four lines of the panel's address display. All three sections receive the same four-line input from the CMA switch section, but the strobing signal to each section is different. In the search mode, the latches react to the A_1, A_2, and A_3 signals from the timing section. They therefore register the address of the most current instruction conducted on the main data bus, and present this to the display indicators on the front panel. They are updated with each new cycle, until the timer is stopped.

The CMA address multiplex controls the data input to the address display latches. In the search mode, the switch forwards a 12-line signal to the latches, consisting of three parallel 4-line branches of the main data bus.

The instruction display latch contains two sections. As with the address latch, each section receives a four-line input from the main data bus, when the panel is operating in the search mode. The strobing signals to the latch, however, are the $\overline{M}_1$ and the $\overline{M}_2$ outputs from the timing section. These latches therefore record the instruction that the program memory returns to the processor. Like the address display latches, they are updated on every cycle, until the control logic halts the timer.

Note that the eight-line data input to the instruction display latch is selected by the CMA instruction multiplex. In the search mode, the multiplexer forwards an eight-line signal to the instruction display latches, consisting of two parallel four-line branches of the main data bus.

The execution display latch contains two sections. Both receive 4-line inputs, directly from the main data bus. The strobing signals to this section are the $\overline{X}_2$ and $\overline{X}_3$ outputs from the timer section. These latches record any I/O or RAM memory transactions during the execution phases of the machine cycle.

The pointer display latch also receives two four-line inputs from the main data bus. This section, however, is updated only during X_2 and X_3 of a cycle in which an SRC instruction occurs. The strobing arrangement for the pointer display latch thus contains some extra logical provisions. The latch infers an SRC transaction, by sensing the coincidence of a CM-ROM signal from the Central Processor Module and an $\overline{X}_2$ signal from the timer logic. Whenever that precondition is fulfilled, the SRC updating logic permits the pointer display latches to receive the $\overline{X}_2$ and the $\overline{X}_3$ timer signals. The result is that the pointer display latches will always indicate the last object of the most recent SRC. One latch holds the X_2 byte, and the other holds X_3. When the pointer display is updated, the SRC update section lights the POINTER VALID indicator, signifying at least one SRC since PANEL RESET.

The active bank display latch receives a 4-line data input from the Central Processor Module, through inverters mounted on the Control Module. The input consists of the 0-3 CM-RAM signals. The strobe input to this stage is the unqualified $\overline{A}_3$ from the timer. Since the processor always pulses the designated command line during A_3, this latch will reveal the concurrent status of the memory control circuitry.

The 12-line output from the address display latch goes to the address comparator section. The reference input to this section is a 12-line signal from the ADDRESS/DATA selector switches on the console panel. The comparator sense any coincidence in the two inputs, and responds by sending control signals to both the timer control section and the pass counter section.

The search address latch holds the data from the panel switches, entered when the LOAD button is depressed. The latch, as shown in Figure 6-1, receives INCREMENT and DECREMENT inputs from switches on the console panel. The operator can use these switches to vary the search location, without resorting to the resetting of the selector switches.

The pass counter is a four-stage binary counter which is reset to zero each time that a PANEL RESET occurs. A PANEL RESET will clear the panel logic at the time the LOAD button is depressed. From that point in time, the pass counter registers successive impulses from the address comparator, and forwards its four-line output to the pass comparator section.

In addition to the input from the pass counter, the pass comparator receives a four-line input from the PASS COUNTER selector switches on the front panel. When the requisite number of instruction passes are completed, and the pass counter input coincides with the setting of the PASS COUNTER selectors, the pass comparator forwards an indication to the timer control section.

The timer control, you recall, also receives a coincidence indication from the address comparator. When these two signals concur, the timer control disables the timer section. The strobe inputs to the display latches are inhibited and the display is frozen at that point. The timer control then lights the SEARCH COMPLETE indicator.

Observe that the timer control section receives an input from the NEXT INSTRUCTION switch on the front panel. When this switch is activated, the timer control pauses for one full cycle after coincidence, before stopping the timer. This provision permits the operator to follow a conditional branch in the program, where the INCREMENT feature would not apply.

Run Mode

In the run mode, the panel indicators do not lock up when the search address comes around. This mode is designed to provide a negative-going $\overline{\text{ADR SYNC}}$ pulse when the preset address is encountered. The output pulse is available at TEST POINT 7 on the Control and Display Panel logic board, and will serve to synchronize an external oscilloscope used for test or for troubleshooting. This output will be particularly valuable in the development of interfaces, where the INTELLEC® 4/MOD 40 system is controlling external processes.

Operation in the run mode requires that the PASS COUNTER selectors be set to "0000." No output is possible unless this precondition is fulfilled. Run operation then occurs when the SEARCH ADDRESS CONTROL: RUN selector is set.

With the PASS COUNTER switches set to zero, the timer control section senses coincidence every time the search address comes up. A signal indicating the coincidence is sent to the run synchronizaiton section, where the concurrence of the M_2 timer signal causes the TP7 output to go HIGH. This cycle will be repeated every time the search address is encountered, producing a series of negative-going sync pulses with a duration of approximately 6.75 microseconds.

CMA Mode

CMA operation is selected by placing the CMA selector in the ENABLE position. The CMA mode makes quite different use of the display indicators.

The operator first sets up the desired address, using the ADDRESS/DATA selectors on the console. He presses the LOAD switch momentarily, to enter the address into the search address latch. The address will be registered on the display indicators.

The operator then sets up the data to be entered, using the lower eight ADDRESS/DATA switches, and depresses the CMA:WRITE button. The data will be recorded in program RAM, and the new contents of that location will be displayed on the INSTRUCTION indicators.

When he presses the LOAD switch, the operator places an address in the search address latch. This 12-line address is presented to the RAM Memory Module via the Control Module's switching logic, presetting conditions for the write-in which is to occur.

When he subsequently sets up the data to be written in, the operator sends an eight-line data field to the RAM Memory Module's data input. Again this passes through switching circuitry on the Control Module.

Observe that when the CMA function is enabled, the CMA logic section sends a CMA signal to both the CMA address multiplex and the CMA instruction multiplex.

Consequently, the address display latches now receive their input from the search address latch, rather than from the main data bus. In similar fashion, the instruction display latch will receive its eight-line data input from the RAM Memory Module.

When the operator depresses the CMA:WRITE switch, the CMA logic generates a $\overline{\text{LOAD CMA}}$ signal. This, in coincidence with the A_1 timing signal, causes the address and the write data to be entered in the address display and instruction display latches, respectively.

At the same time, the CMA logic sends a CMA WRITE command to the Control Module. The Control Module in response generates a signal that writes the panel data into program RAM. This completes the CMA operation.

Reset

A general reset of the system, initiated by the front panel pushbutton, will reset the logic circuitry of the Control and Display Panel. However, other conditions also require a localized reset of the panel's logic.

The panel reset section is equipped to generate a $\overline{\text{PANEL RESET}}$ whenever one of the following conditions occurs:

a) system RESET
b) actuation of the LOAD switch
c) actuation of the INCR switch
d) actuation of the DECR switch
e) selection of CMA ENABLE
f) selection of the RUN mode

The $\overline{\text{PANEL RESET}}$:

1) clears the pass counter
2) restarts the timer
3) clears the POINTER VALID indicator

THEORY OF OPERATION

This section describes the operation of the Control and Display panel, at the circuit level. We shall discuss, in turn, each of the functional blocks presented on page 59. These include:

a) timer
b) display latches
c) SRC update
d) search address latch
e) search comparator
f) pass counter
g) pass comparator
h) timer control
i) RUN sync
j) CMA logic
k) panel reset logic

Timer

The timer receives the 01B, $\overline{\text{02B}}$, and $\overline{\text{EOC SYNC}}$ signals from the Control Module. It produces a seven-line output. Each output line carries a negative-going 386 microsecond pulse, and each pulse output stands in a different phase relationship to the basic SYNC reference. The seven

lines correspond to seven of the eight phases in the basic system cycle: A_1, A_2, A_3, M_1, M_2, X_2, and X_3.

The timer logic consists of a 74161 monolithic binary counter (A19), a 3205 3-line to 8-line decoder (A17), a buffer, and an inverter. All are shown in the upper left hand corner of the Control and Display Panel Schematic, Figure 5-2.

The ϕ1B signal is applied to the CLOCK input of the 74161, causing it to count. The $\overline{\text{EOC SYNC}}$ signal is applied to the counter's clear input, resetting the device immediately following every eighth count. The counter therefore has eight distinct internal states, and these are reflected on its A, B, and C output lines. Note that the D output may go unused, since the counter need never indicate more than eight states (corresponding to binary 000 - 111).

An unusual feature of this configuration is the use of an inverted $\overline{02B}$ signal, to enable the 3205 (pin 6). This ensures that each of the output pulse phases is well separated from its two adjacent phases, and that there is no logical ambiguity due to signal overlap. It also ensures time for stabilization of data on the bus.

The two enable inputs on the 3205 are also linked to specific logical functions. The input at pin 4 comes from the timer control circuit, and is used to disable the timer when the search is complete. The pin 5 input is under control of the CMA logic. This input disables the timer when a CMA transaction is in progress, so that the latch strobes from the timer will not interfere with the $\overline{\text{LOAD CMA}}$ strobe.

And finally, observe the use of the 7417 buffer (A29-10-11) in conjunction with the $\overline{\text{STOP ACK}}$ signal from the Memory Control Moudle. The buffer's output lights the RUN indicator on the panel, providing a positive indication when the system is running.

Display Latches

The display latches consist of three address latches, two instruction latches, two execution latches, two pointer latches, and one active bank latch. These register and hold data for display, as explained in Section entitled Functional Description of the Control and Display Panel.

The display latches are implemented using 3404 hex latches. All 10 of the four-line latches are shown at the right of the schematic diagram. All are clearly labelled as to their data function, and would seem to require no special description. Note, however, that the data outputs on the latches are logically inverted. The panel indicators (not shown) are tied in common to the +5 Volt supply on the far side, and grounded through selected latch outputs to illuminate the display.

Observe that each of the quad-latch sections reacts to a negative-going strobe input, applied to pin 7 or to pin 15. Eight of the latch sections receive strobe signals from the timing section as appropriate to their individual display functions. The pointer display latches, however, have special logical conditions attached to their strobing signals. These are explained in the next section.

SRC Update

The SRC update section is equipped to sense an SRC in progress. It performs two functions in response. First, it gates the X_2 and the X_3 timing signals to the strobe inputs of the pointer latches. Then it lights the POINTER VALID indicator, revealing that at least one SRC has in fact occurred since the panel was last reset.

The SRC update section consists of a 7474 dual D flip-flop, two sections of a quad NAND (A27-4-5-6 and A27-8-9-10), and one section of a hex inverter (A35-3-4).

Observe that pins 4 and 9 of the A27 coincidence gates receive inverted phases of the $\overline{X}_2$ and $\overline{X}_3$ signals from the timer section, and that pins 5 and 10 are tied in common to A28-5. The X_2 and X_3 UPDATE signals to the pointer latches are thus contingent upon a HIGH at the Q output of the SRC flip-flop. This flip-flop is cleared at the beginning of every cycle, by the $\overline{A}_1$ signal applied to pin 1. But the concurrence of a CM-ROM at pin 2 and an X_2 at pin three will set the latch, causing Q to go HIGH and gating the $\overline{X_2}$ $\overline{\text{UPDATE}}$ pulse through to the SRC X_2 latch.

The Q output remains HIGH, until the end of the cycle. As a result, the $\overline{\text{SRC } X_3 \text{ UPDATE}}$ is subsequently gated through to strobe the SRC X_3 latch.

The SRC X_3 UPDATE is also directed to A28-11, the CLOCK input of the "pointer valid" flip-flop. Note that the flip-flop was cleared by the $\overline{\text{PANEL RESET}}$, but that the input to pin 11 now SETs the device. The LOW output at A28-8 illuminates the POINTER VALID indicator on the front panel.

Search Address Latch

The search address latch registers the data set up on the front panel's ADDRESS/DATA switches, when the operator presses the LOAD pushbutton. In the search mode, this address is presented to the address comparator, as the reference against which the address display latch is compared. In the CMA mode, the latch forwards its contents to the RAM Memory Module, to point the location of the impending write-in.

The contents of the latch may be incremented or decremented, by means of the INCR and the DECR buttons on the console. The operation may be repeated indefinitely, stepping the latch counter through entire sections of the program memory. The ADDRESS/DATA switches remain unchanged, but the console indicators always display the current address in the latch, when the SEARCH COMPLETE lamp is lighted.

The entire search address latch consists of three presettable binary up-down counters, type 74193. These are shown below and to the left of center of the Control and Display Panel schematic, as A21, A23, and A25.

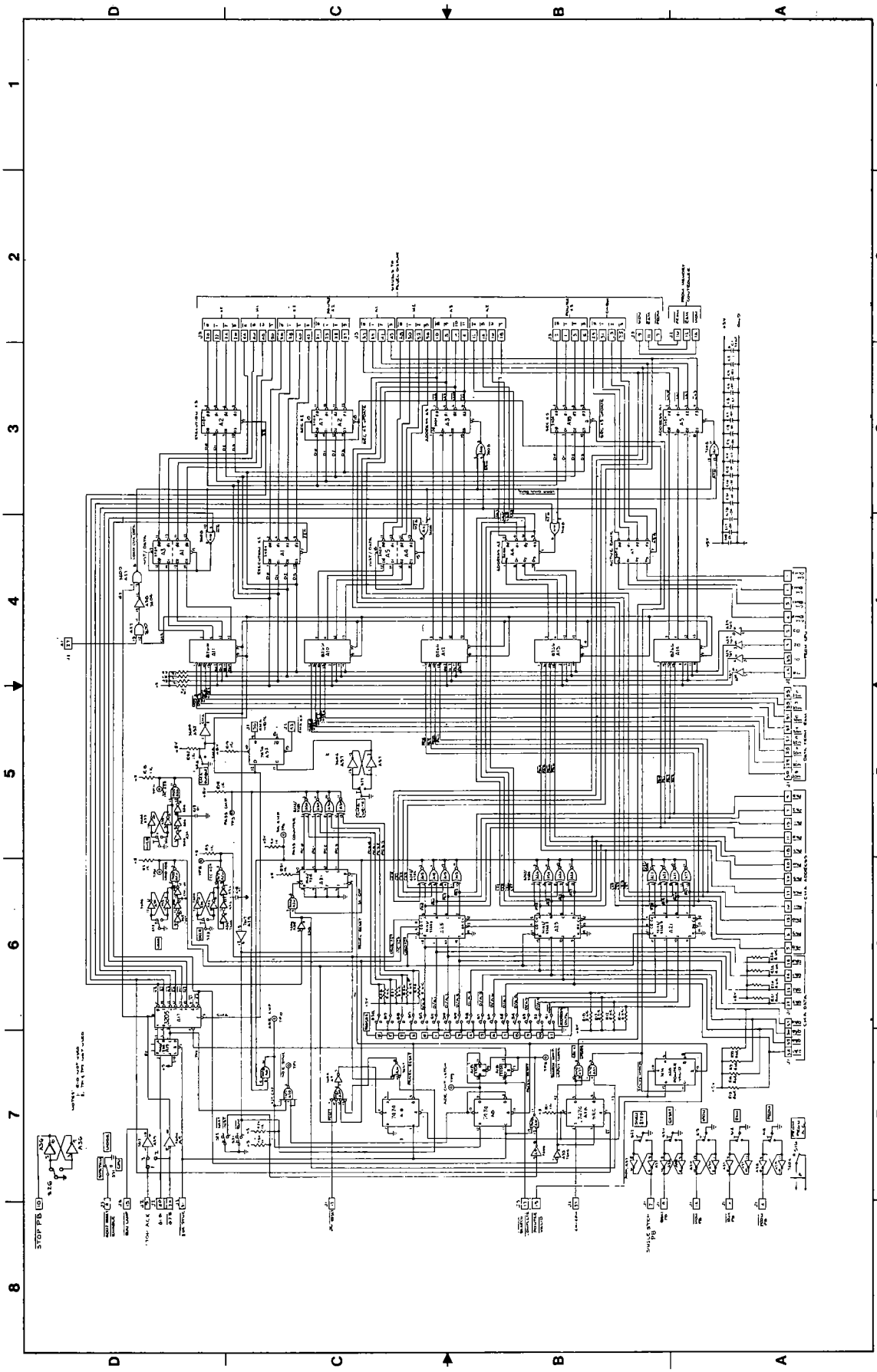


Figure 5-2. Control & Display Panel Schematic

As shown, each counter controls four bits of the twelve-bit address field. The address switches on the panel present a binary address to the data inputs of the counter elements. A negative-going transition from the LOAD push-button is applied in parallel to the $\overline{L}$ inputs (pin 11) of all three counters, causing them to register the switch data.

Note that the LOAD, INCR, and DECR switches, shown above and left of center near the top of the schematic. These controls consist of a pair of cross-connected inverters used as a bi-stable, and a momentary switch. Each feeds a signal conditioning network made up of three inverters, a NAND, and a capacitor which form a stored coincidence de-bouncing circuit.

The negative-going INCR and DECR pulses generated in the signal conditioning elements go to the count-up ($\overline{CU}$) and the count-down ($\overline{CD}$) inputs of the first counter. The borrow ($\overline{BW}$) and the carry ($\overline{CY}$) outputs from this stage are cascaded to the $\overline{CD}$ and $\overline{CU}$ inputs of the next stage, and that stage in turn cascades into the next.

Search Comparator

The search comparator receives the 12-line output of the search address latch, plus the 12-line address currently stored in address display latch. It produces a HIGH output when it senses coincidence.

The search comparator consists simply of four quad XNOR-gates, A15, A22, and A24, with their output lines tied together in a wire-OR configuration. When coincidence occurs, this line swings HIGH. The indication is forwarded both to the pass counter section and to the timer control section.

Pass Counter

The pass counter is cleared at the time of $\overline{\text{PANEL RESET}}$, and is incremented thereafter each time the preset search address comes up.

The pass counter consists of a 74161 binary counter (A31), plus A26-4-5-6 and A35-5-6 shown immediately to its left on the schematic.

The SA CMP output of the search comparator enables the coincidence gate A26 at time A_3 of the cycle in which the search address comes up. The $\overline{X}_3$ output of the timer is inverted in A35, and applied to A26-5. When the two inputs to A26 concur, the output goes LOW. The subsequent positive-going transition at the end of X_3 is applied to the CLOCK input of the counter (pin 2). This increments the pass counter, at the end of the system cycle.

The four-line output of the pass counter goes directly to the inputs of the pass comparator, shown to counter's left on the schematic.

Pass Comparator

The pass comparator consists of a single quad XNOR-gate, A30. The four outputs are tied together in a wire-OR, just as in the search comparator.

Data inputs to the comparator are the four lines from the pass counter, plus the corresponding four lines from the PASS COUNTER switches on the console panel. Coincidence in these two inputs causes the comparator's output line to go HIGH. The output is directed to the timer control section.

Timer Control

The timer control section receives coincidence outputs from both the search comparator and the pass comparator. When it senses a concurrence in both inputs, the timer control shuts down the timer, freezing the display and lighting the SEARCH COMPLETE indicator on the panel.

The timer control section consists of one-half of a 7474 dual D flip-flop (A8-8-9-10-11-12-13), an inverter (A9-8-9), a 7408 AND-gate section (A6-4-5-6), and two sections of a 3404 hex latch (A18-11-12-13-14-15).

The timer control is configured to sense a coincidence in:

a) SA CMP from the search comparator
b) PASS CMP from the pass comparator
c) the positive-going trailing edge of the $\overline{\text{EOC SYNC}}$ signal from the Control Module

The latter signal event occurs at the end of the X_3 interval, at the completion of a system cycle.

The SA CMP and the PASS CMP signals are combined in the A6-4-5-6 coincidence section, to produce a HIGH output which goes to the D input of the timer control flip-flop (A8-12).

During a search, the Q output of the timer control flip-flop is LOW, the result of a prior $\overline{\text{PANEL RESET}}$ applied to its clear input (pin 1). With a HIGH at pin 12, however, the positive-going trailing edge of the EOC SYNC at pin 11 latches the Q output HIGH. This output is presented to a cascaded pair of inverting latches contained on A18.

The common $\overline{\text{STROBE}}$ input of these latch sections is held LOW continuously, as long as the system is not in the NEXT INST mode. The HIGH from the timer control flip-flop therefore passes through the latches. Applied to A17-4 this HIGH disables the timer, inhibits the timing strobes, and freezes the panel display. The LOW at A18-11 lights the SEARCH COMPLETE indicator on the console.

When the NEXT INST mode is selected, the timer control logic delays for one full cycle before disabling the timer. This permits the subsequent instruction cycle to be registered before the display is frozen.

When the console is in the NEXT INST mode, the NEXT INST switch applies a LOW to A9-9, and the signal coupled through at A6-11 disables the LOW normally applied to the $\overline{\text{STROBE}}$ inputs of the 3404 latch sections. These latches can now respond to their data inputs only when a simultaneous LOW is presented to A6-13.

When the coincidence of the SA CMP, the PASS CMP, and the trailing edge of the $\overline{\text{EOC SYNC}}$ occurs, the timer

control flip-flop is set as usual. The flip-flop holds its state, but no strobe will enable the A18 latches until the **leading** edge of another $\overline{\text{EOC SYNC}}$ pulse arrives, at the end of the next system cycle. At this time, the Q output of the timer control flip-flop will be coupled through the latches to disable the timer. By then, however, the display latches will contain the transactions of the subsequent system cycle.

Run Sync

The RUN mode provides a synchronizing pulse at the time the preset search address is encountered. The search address is set and loaded normally, but the PASS COUNTER selection switches must be set to zero.

$\overline{\text{ADR SYNC}}$ is provided by the lone NAND-gate A20-1-2-3-4-5-6. The output of this gate swings LOW when the following coincidence is indicated on its four input lines:

a) no PANEL RESET is in progress (pin 5)
b) the timer control flip-flop is cleared (pin 4)
c) ADR CMP from the timer control is HIGH (pin 1)
d) phase M_1 begins

The output of the gate remains LOW, until the end of A_3, when the trailing edge of the $\overline{\text{EOC SYNC}}$ sets the timer control flip-flop. This produces a negative-going pulse at TP7, approximately 6.75 microseconds in duration.

In the absence of search address coincidence, a subsequent $\overline{\text{EOC SYNC}}$ pulse will reset the timer control flip-flop at the end of the next cycle. The RUN cycle is then free to repeat, on the next search pass.

CMA Logic

The CMA logic permits the entry of data into program RAM, through the use of the ADDRESS/DATA switches on the console. Such transactions require the cooperation of circuitry on the system Control Module.

The CMA address switches and the CMA instruction switches consist of the five 8266 monolithic 8-to-4 line multiplexers, labelled A10, A11, A12, A13, and A14 on the Control and Display Panel schematic. All multiplexers have their pin 9 control inputs clamped HIGH by A35-1-2, when the CMA mode is selected. Thus the input lines from the system's main data bus are inhibited, and the multiplexers feed data from the search address latches and from the RAM Memory Module's data out lines to the A_1 through M_2 sections of the display latch.

The CMA enable section also performs the following related functions, when CMA ENABLE is selected:

a) clamps the pass counter's output lines, by application of a LOW level to A31-9
b) inhibits the timer outputs, by transmitting a HIGH to A17-5
c) places a HIGH enabling level at pin 12 of the CMA write flip-flop (A32)
d) places a HIGH level at A27-12, enabling subsequent generation of a $\overline{\text{LOAD CMA DATA}}$ strobe

With a HIGH at its D input, the CMA write flip-flop responds to actuation of the CMA WRITE switch by latching pin 9 HIGH. This output, the CMA WRITE signal, goes to the Control Module where the writing commands for the RAM Memory Module are generated. The Control Module returns a negative-going $\overline{\text{CMA EX}}$ signal to pin 13 of the CMA write flip-flop, abruptly clearing the latch.

With the CMA function enabled, an A_1 signal from the Control Module is gated through A27-11-12-13. This output concurs with the 02 clock signal, in A27-1-2-3, to produce the $\overline{\text{LOAD CMA DATA}}$ strobe. The strobe is applied, through OR-gates, to the strobing inputs of the A_1 - M_2 display latches. Thus, the instruction address and the contents of that address are displayed on the panel indicators continuously, during the time that CMA operation is enabled.

Panel Reset

The reset generator clears the panel logic, under specific control conditions.

The panel reset section consists of one-half of a 7404 dual D flip-flop (A8-1-2-3-4-5-6), two OR-gates (A6-8-9-10 and A20-8-9-10-12-13), and an inverter (A9-10-11).

The A6 gate operates in tandem with the A20 gate, to provide a LOW output at A9-10 whenever any of the following conditions occur:

a) system RESET	A20-9
b) LOAD	A20-10
c) INCR	A20-12
d) DECR	A6-9
e) CMA ENABLE	A6-10
f) RUN mode selection	A6-10

When A9-10 goes LOW, it sets the reset flip-flop. The flip-flop remains set until the trailing edge of the EOC pulse (pin 1) clears it. The output at A8-6 is therefore a negative-going pulse, of indefinite duration, which clears the pass counter, the timer control flip-flop, and the pointer valid flip-flop preparatory to the next run of the panel logic.

UTILIZATION

This section contains physical and electrical data pertaining to the Console and Display Panel.

Environment

Temperature extremes can cause instability, or result in permanent damage to the circuits on the panel. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade.

Relative humidity is not critical to the panel's operation.

Electrical Connections

The Console and Display Panel contains three connectors. J1 and J2 connect the panel to J1 and J2 respectively, on the system Control Module. This provides the proper interface to the RAM Memory Module and to the

Central Processor Module. J3 connects the panel's logic board to the controls and display indicators on the console.

The mating connectors for J1, J2, and J3 are described in Table 5-1.

Mating Connectors

PANEL JACK	DESCRIPTION	Mfr.	Part No.
J1	40-pin Male	3M	3432-1002
J2	40-pin Male	3M	3432-1002
J3	50-pin Male	3M	3433-1002

Table 5-1.

Pin allocations on these connectors are given on following pages.

Power Requirements

The Console and Display Panel requires DC power of:

+5 VDC @ 3.0 Amperes (max)
@ 1.4 Amperes (typ)

Refer to Pin List for the points of connection.

Signal Requirements

All signals in and out of the panel are at standard TTL levels. Refer to Pin List for the connector pin allocations.

Pin List

The following section describes connector pin allocations on the Control and Display Panel. The pins and their signal functions are listed in Tables 5-2, 5-3, and 5-4.

J1 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1		
2	S/S PB (TEST PB)[2]	
3	$\overline{\text{PANEL ADDRESS 9}}$	
4	$\overline{\text{PROM P. B.}}$	
5	$\overline{\text{PANEL ADDRESS 8}}$	
6	$\overline{\text{RAM P. B.}}$	
7	$\overline{\text{PANEL ADDRESS 2}}$	
8	$\overline{\text{RESET P. B.}}$	
9	$\overline{\text{PANEL ADDRESS 3}}$	
10	$\overline{\text{STOP}}$ PB ($\overline{\text{TEST HOLD}}$[2])	
11	$\overline{\text{PANEL ADDRESS 0}}$	
12	$\overline{\text{PANEL RESET}}$	
13	$\overline{\text{PANEL ADDRESS 1}}$	
14	$\overline{\text{MON P. B.}}$	
15	$\overline{\text{PANEL ADDRESS 10}}$	
16	$\overline{\text{PANEL ADDRESS 11}}$	
17		
18	$\overline{\text{PANEL ADDRESS 7}}$	
19		
20	$\overline{\text{PANEL ADDRESS 6}}$	Control Module
21	$\overline{\text{MDI 3}}$	
22	$\overline{\text{DATA/ADDRESS 3}}$	
23	$\overline{\text{DATA/ADDRESS 2}}$	
24	$\overline{\text{DATA/ADDRESS 1}}$	
25	$\overline{\text{DATA/ADDRESS 0}}$	
26	$\overline{\text{PANEL ADDRESS 5}}$	
27	$\overline{\text{PANEL ADDRESS 4}}$	
28	$\overline{\text{MDI 2}}$	
29	$\overline{\text{MDI 1}}$	
30	$\overline{\text{MDI 0}}$	
31	$\overline{\text{MDI 5}}$	
32	$\overline{\text{MDI 4}}$	
33	$\overline{\text{MDI 7}}$	
34	$\overline{\text{DATA/ADDRESS 7}}$	
35	$\overline{\text{DATA/ADDRESS 6}}$	
36	$\overline{\text{DATA/ADDRESS 5}}$	
37	$\overline{\text{DATA/ADDRESS 4}}$	
38	$\overline{\text{MDI 6}}$	
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		

Table 5-2.

NOTES:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

[2] INTELLEC® 4/MOD 4 function.

J2 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1	CM-RAM 2	Control Module
2	DATA 3	
3	DATA 2	
4	CM-RAM 0	
5	CM-RAM 1	
6		
7	CM-RAM 3	Control Module
8	$\overline{\text{4002 RESET}}$	
9	STOP ACK	
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27	A1	Control Module
28	DATA 0	
29		
30	$\overline{\text{Ø2B}}$	
31	$\overline{\text{EOC SYNC}}$	
32	$\overline{\text{CMA EX}}$	
33		
34		
35		
36	CMA WRITE	
37	CM-ROM	
38	DATA 1	
39		
40	Ø1B	
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		

Table 5-3.

NOTES:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

J3 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1	$\overline{\text{SRC X3-1}}$	
2	$\overline{\text{SRC X3-0}}$	
3	$\overline{\text{SRC X3-2}}$	
4		
5	$\overline{\text{SRC X3-3}}$	
6		
7	$\overline{\text{PROM}}$	
8		
9	$\overline{\text{MON}}$	
10		
11	$\overline{\text{RAM}}$	
12		
13	$\overline{\text{POINTER VALID}}$	
14		
15	$\overline{\text{RUN LAMP}}$	
16		
17	$\overline{\text{SEARCH COMPLETE}}$	
18		
19		
20	$\overline{\text{X3-0}}$	
21	$\overline{\text{SRC X2-0}}$	
22	$\overline{\text{X3-1}}$	
23	$\overline{\text{SRC X2-1}}$	
24	$\overline{\text{X3-2}}$	
25	$\overline{\text{SRC X2-2}}$	
26	$\overline{\text{X3-6}}$	Display Panel
27	$\overline{\text{SRC X2-3}}$	
28	$\overline{\text{M2-0}}$	
29	$\overline{\text{CM-RAM 0}}$	
30	$\overline{\text{M2-1}}$	
31	$\overline{\text{CM-RAM 1}}$	
32	$\overline{\text{M2-2}}$	
33	$\overline{\text{CM-RAM 2}}$	
34	$\overline{\text{M2-3}}$	
35	$\overline{\text{CM-RAM 3}}$	
36	$\overline{\text{X2-0}}$	
37	$\overline{\text{A1-0}}$	
38	$\overline{\text{X2-1}}$	
39	$\overline{\text{A1-1}}$	
40	$\overline{\text{X2-2}}$	
41	$\overline{\text{A1-2}}$	
42	$\overline{\text{X2-3}}$	
43	$\overline{\text{A1-3}}$	
44	$\overline{\text{M1-4}}$	
45		
46	$\overline{\text{M1-5}}$	
47		
48	$\overline{\text{M1-6}}$	
49		
50	$\overline{\text{M1-7}}$	

Table 5-4.

NOTES:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

CHAPTER 6 THE imm4-22 INSTRUCTION/DATA STORAGE MODULE

The imm4-22 Instruction/Data Storage Module expands the control storage, data storage and I/O capacity of the INTELLEC® 4/MOD 40 Systems' Central Processor Module.

The control storage, data storage and I/O capacity of the module parallel those of the Central Processor Module. Each Instruction/Data Storage Module contains four 4002 RAMs, sockets for four 4702A PROMs, four TTL input ports, and four corresponding output ports. The capabilities of the module are summarized in Table 6-1 below.

imm4-22 Capability

Control Memory	1024 x 8 bits
Data Storage	320 x 4 bits
RAM Output Ports	4
Simulated ROM Output Ports	4
Simulated ROM Input Ports	4

Table 6-1.

Logic built into the module permits the use of as many as three imm4-22's in a single INTELLEC® 4/MOD 40 system. Wire jumpers on the module establish the chip addressing, or paging, at the user's discretion.

The Instruction/Data Storage Module uses existing data busses in the system and existing control functions of the Central Processor Module. No special interface is necessary. The card plugs directly into the unviersal socket on the system's mother board. All control and power connections enter via this connector. The input and output ports connect to the rear panel of the INTELLEC® 4/MOD 40 system, through separate connector plugs at the top of the card.

FUNCTIONAL DESCRIPTION OF THE MODULE

The Instruction/Data Storage Module contains six functional sections:

a) chip decoding
b) instruction storage
c) data storage
d) RAM output ports
e) input multiplexer
f) output latches

The functional relationship among these sections is shown in Figure 6-1.

The chip decoding section of the module receives its input from the Central Processor Module's chip select bus (C_0 through C_3). These lines both enable the module and identify a simulated ROM page within the module. The output from the chip decoder thus selects one of the four PROMs on the module, as well as one of the input ports and one of the output ports. Each PROM, with its associated input and output ports, simulates a single 4001 ROM of the kind used in the MCS-40™ microcomputer set.

The instruction storage section receives an eight-line input address from the Central Processor Module (MA_0 - MA_7), as well as the output from the chip decoder section. This twelve-bit field specifies a single eight-bit word within the instruction storage block. The output of the instruction storage section is an eight-line data signal which is returned to the Central Processor Module on the instruction bus ($\overline{MDI}_0$ - $\overline{MDI}_7$).

The output latches receive the $\overline{OUT}$ signal from the Central Processor Module, as well as the four-line output from the chip decoder section. When an output operation is called for, and when the chip-select bus simultaneously addresses an output port located on the module, the data contents of the I/O bus is stored in the designated output latch.

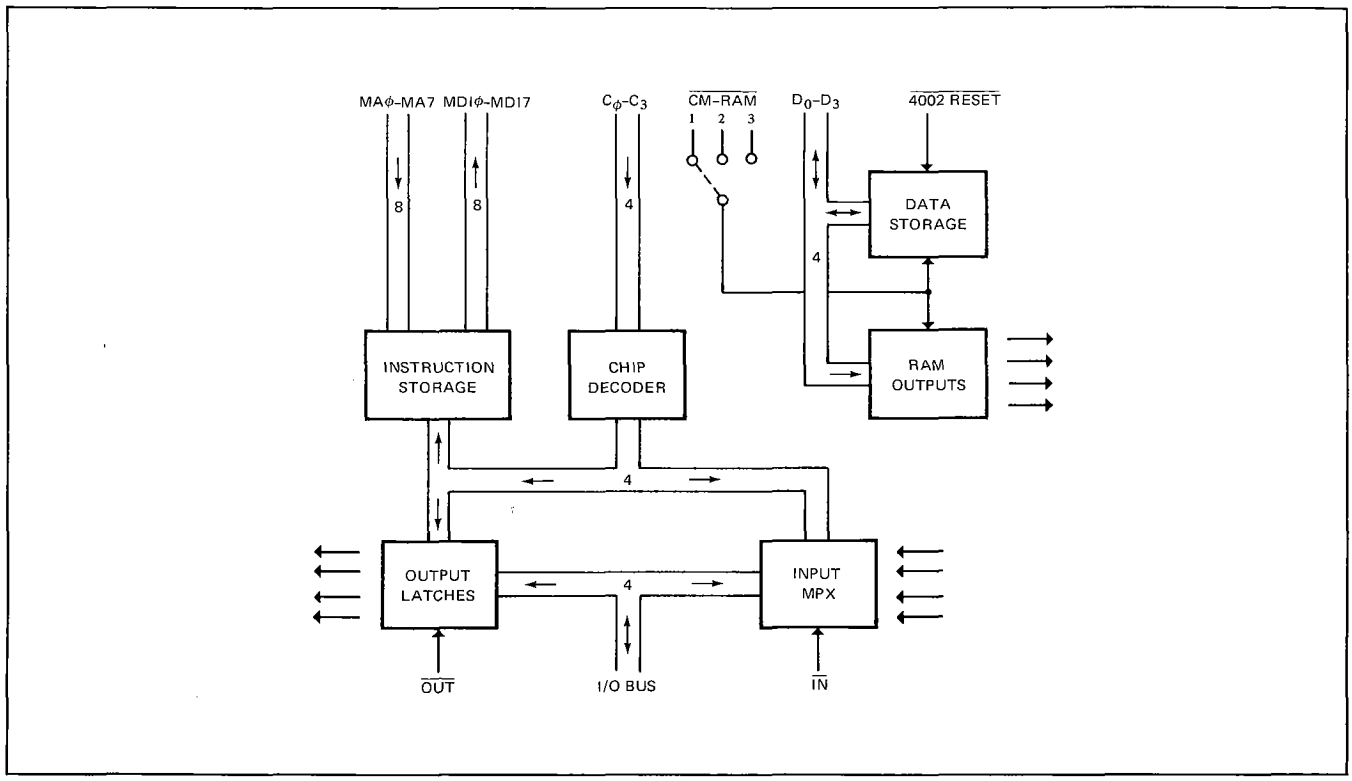


Figure 6-1. imm4-72 Functional Block Diagram

The input multiplexer section receives the $\overline{IN}$ signal from the Central Processor Module, when an input operation is in progress. The multiplexer also receives the four-line output from the chip decoder. When an input operation is called for, and when the chip-select bus simultaneously addresses an input port on the module, the data at the specified port is gated through to the I/O bus, passing on to the Central Processor Module.

The data storage elements, and their associated output ports, are functionally independent of the other sections on the module. They are selected by the $\overline{CM\text{-}RAM}$ signals from the Central Processor Module, receiving and transmitting data via the main data bus ($\overline{D}_0$ - $\overline{D}_3$).

THEORY OF OPERATION

This section describes the imm4-22 Instruction/Data Storage Module at the circuit level. Refer to the module schematic, Figure 6-2, for the following discussion.

Chip Decoding

Each of the 4702A PROMs on the module, and each of the input and output ports, is associated with a **virtual** ROM chip. Two ports and one PROM are therefore uniquely specified by the four-bit contents of the chip-select bus (C_0 - C_3). The Central Processor Module commands this bus unilaterally. The bus is used to display the most significant four bits of the instruction address, during the M_1 and M_2 phases of the processor cycle. During the X_2 and X_3 phases, when I/O and memory reference are executed, the chip-select bus specifies the destination of the memory or I/O transaction.

A 3205 one-of-eight decoder (A10) performs the chip select function for the Instruction/Data Storage Module. Wire jumper options permit selective enabling of the decoder's output lines, so that elements on the module are selected only when the appropriate logical quadrant is addressed (#4-#7, #8-11, or #12-#15).

Each of the four output lines from the 3205 is applied to the enabling input of one of the four PROMs on the module. A LOW at the $\overline{CS}_0$ input (pin 14) selects the chip during addressing operations.

Each of the decoder's output lines also enables one of four coincidence indicators which gate the $\overline{OUT}$ signal from the processor module to the $\overline{STROBE}$ inputs of the output latches. By this expedient, a single output port is selected.

The input ports use a slightly different selection mechanism. The C_0 and C_1 chip select lines operate directly on the input multiplexer. One of the four input ports will therefore be selected, regardless of which quadrant is indicated on the C_2 and C_3 lines. The input multiplexers, however, are connected to the I/O bus only when an $\overline{IN}$ signal from the processor module coincides with an output on one of the chip decoder's four output lines. Thus the chip-select bus also specified one of four input ports in the quadrant occupied by the module.

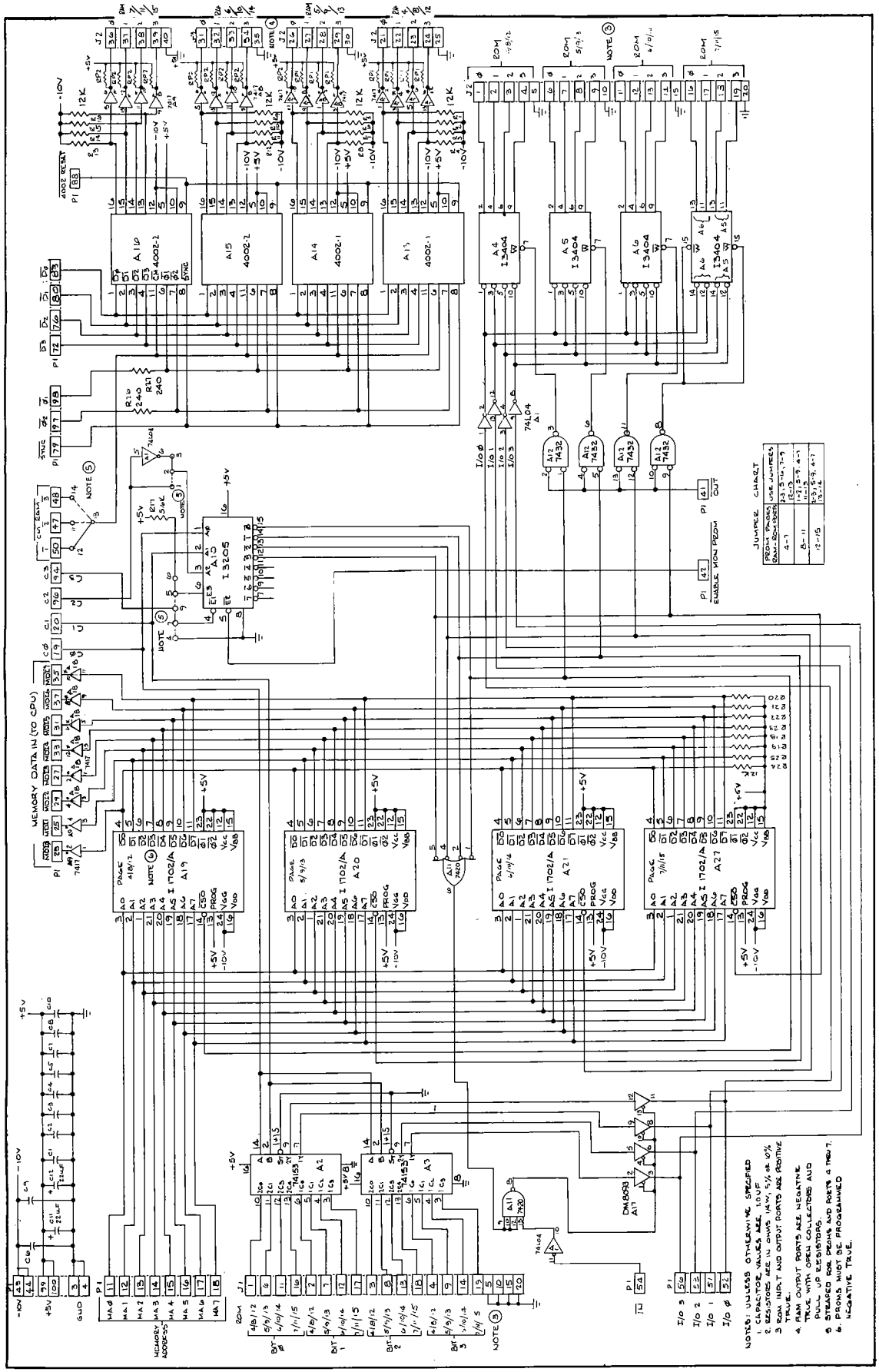


Figure 6-2. imm4-22 Schematic Diagram

Instruction Fetch

During phases A_1 through A_3 of a typical system cycle, the Central Processor Module assembles a twelve-bit address field, consisting of an eight-line memory address (MA_0 - MA_7) plus a four-line chip-select signal (C_0 - C_3). This address is present during M_1 and M_2, when the processor is receiving an instruction from program memory.

The chip select decoder enables a single PROM chip, and the eight address lines, applied to the four chips in parallel, specify a single location within that memory element. The eight bit output is returned to the Central Processor Module on the eight lines of the MDI bus. The negative-true output is present as long as the address and enabling signals are applied, during both the M_1 and M_2 phases of the cycle.

ROM Input

When the Central Processor Module performs an input operation, it transmits a four-line signal on the chip-select bus (C_0 - C_3), during X_2 of the processor cycle. It simultaneously transmits an $\overline{IN}$ command to the input ports.

The two low order chip select lines (C_0 and C_1) control the output states of two 74153 dual 4-to-1 multiplexers which serve as the port inputs. The switched four-line output from the multiplexers is applied to a bank of four gated tri-state buffers, and thereafter to the I/O bus. The buffers receive a LOW enabling signal only when a coincidence exists, in the chip decoder output and the $\overline{IN}$ signal from the processor module. The I/O bus transmits the data at the port inputs to the Central Processor Module, where it is forwarded to the 4040 CPU.

ROM Output

When the Central Processor Module performs a ROM output operation, it transmits a four-bit chip-select byte (C_0 through C_3) during the X_2 phase of the cycle. At the same time, it transmits an $\overline{OUT}$ signal to the output ports under its command. The processor chip places a four-bit object byte on the main data bus at X_2. This four-line signal passes through switch logic to the I/O bus and is directed to the data inputs of the output ports.

The data on the chip select bus enables the decoder section of the imm4-22 to select the $\overline{OUT}$ strobe and route it to a specific ROM port. The data on the I/O bus is presented to the inputs of the latch sections in parallel, and the strobing signal causes the selected latch to register the I/O data. The latches used for the output function are three 3404 hex latch elements, organized to provide sixteen output lines. Each strobe line commands four latches simultaneously, creating four 4-line outputs.

RAM Memory Reference

The 4002 RAM bank on the imm4-22 is controlled by one of the processor $\overline{CM\text{-}RAM}$ lines, 0 through 3. The control line is selected by the placement of a wire jumper on the board, as shown in Figure 6-2.

The 4002 RAMs are addressed using an SRC instruction, which precedes the I/O instruction as explained in Chapter 2. The RAMs on an enabled bank register and hold this address, pending execution of an I/O or memory reference instruction.

Each RAM on the command line is uniquely identified, by the combination of memory version and P_0 wiring, as explained in Chapter 2.

The designated $\overline{CM\text{-}RAM}$ line is enabled during M_2 of the cycle when an instruction in the I/O and reference group is encountered. This enables the selected RAM to receive the M_2 OPA code which signifies the type of operation to be performed during X_2.

If the code indicates a RAM read, the RAM places the contents of the addressed location on the main data bus ($\overline{D}_0$ - $\overline{D}_3$) during X_2. If a RAM write is indicated, the RAM responds by storing the X_2 contents of the main data bus in the location specified by the preceding SRC. If the OPA code calls for an output operation, the RAM latches the bus onto its output lines, for presentation to the external device.

UTILIZATION

This section provides information on utilization of the imm4-22, for those who contemplate using the module outside the INTELLEC® 4/MOD 40 system.

Installation

In installing the Instruction/Data Storage Module, the user must take account of:

a) environmental extremes
b) mounting
c) electrical connections
d) power requirements
e) signal requirements
f) wiring options
g) external logic required for stand-along operation

ENVIRONMENT

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conductive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity is not critical to the module's operation.

MOUNTING

Avoid locating the card near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections. The result might be abnor-

mally high noise levels, with the problems entailed, or outright failure of the card.

Dimensions of the module are 6.18 x 8.00 inches. Be sure to allow enough additional clearance to ensure adequate cooling.

The card is designed to plug directly into a standard 100-pin, double sided PC edge connector. The connector will serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the card be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the card, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the elements on the module.

ELECTRICAL CONNECTIONS

All power and control connections to the Instruction/Data Storage Module are effected by means of a standard 100-pin, double-sided PC edge connector, with .125" centers. CDC #VPB 01C50E00A1 is one suitable type. Pin allocations on this connector are given in Table 6-3, at the end of this section.

All data inputs to the Instruction/Data Storage Module are connected through J1, at the top of the card. The appropriate mating connector is 3M #3432-1002. Pin allocations on J1 are given in Table 6-4, at the end of this section.

All data outputs from the Instruction/Data Storage Module pass through J2, at the top of the card. This connector is physically identical to that used for inputs. Pin allocations on J2 are given in Table 6-5, at the end of this section.

In the INTELLEC® 4/MOD 40 system, both J1 and J2 are connected, through 24 inches of ribbon cable, to their associated input and output connectors on the rear panel of the unit.

POWER REQUIREMENTS

The Instruction/Data Storage Module requires DC power at the following levels:

+5 ± 5% VDC @	1.4	Amperes (max)
	0.8	Amperes (typ)
-10 ± 5% VDC @	0.5	Amperes (max)
	0.25	Amperes (typ)

Refer to the Pin List for the power connections.

SIGNAL REQUIREMENTS

The $\overline{\text{CM-RAM}}$ lines 1-3, $\overline{D}_0$ - $\overline{D}_3$, $\overline{\text{SYNC}}$, $\overline{01}$, $\overline{02}$, and the $\overline{\text{4002 RESET}}$ inputs require MOS drive levels. Refer to Appendix A. All other inputs and outputs are at TTL levels.

WIRING OPTIONS

Up to three imm4-22 modules may be used in an INTELLEC® 4/MOD 40 system. The user has several wire jumper options that permit him to establish the logical quadrant of the simulated I/O ports. Other jumpers establish the RAM bank number of the 4002s on the module.

Keep in mind that the Central Processor Module contains PROM chips 0 through 3, the simulated I/O ports 0 through 3 and RAM bank 0. To prevent any logical overlap of functions on the Central Processor Module, and to preserve an orderly chip identity sequence, the jumpering scheme shown in Table 6-2 is recommended.

Chip ID Wiring Options

Module	PROMs I/O Ports	RAM Bank #	Jumpers
#1	4 thru 7	1	2-3 5-6 7-9 12-13 8-10
#2	8 thru 11	2	1-2 4-7 5-9 11-13 8-10
#3	12 thru 15	3	2-3 4-7 5-9 13-14 8-10

Table 6-2.

ADDITIONAL LOGIC REQUIRED FOR STAND-ALONE OPERATION

When using the Instruction/Data Storage Module as a stand-alone memory, it will be necessary to provide external multiplexing circuitry for the address and data lines to the PROMs. The simplest way to accomplish this is through the use of the Intel 4008/4009 Memory Interface Set. These monolithic LSI elements contain all logic necessary to interface the PROMs and the ROM I/O ports to the four-bit processor. The user who contemplates such an application is referred to Chapter 2 of this manual, where the interaction of processor and interface is discussed in detail.

Pin List

The following section describes connector pin allocations on the Instruction/Data Storage Module. The pins and their signal functions are listed in Tables 6-3, 6-4, and 6-5.

P1 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11	MEMORY ADDRESS 0	CPU Module
12	MEMORY ADDRESS 1	
13	MEMORY ADDRESS 2	
14	MEMORY ADDRESS 3	
15	MEMORY ADDRESS 4	
16	MEMORY ADDRESS 5	
17	MEMORY ADDRESS 6	
18	MEMORY ADDRESS 7	
19	CHIP SELECT C0	
20	CHIP SELECT C1	
21		
22		
23	$\overline{\text{MDI 0}}$	CPU Module
24		
25	$\overline{\text{MDI 1}}$	
26		
27	$\overline{\text{MDI 3}}$	
28		
29	$\overline{\text{MDI 2}}$	
30		
31	$\overline{\text{MDI 5}}$	
32		
33	$\overline{\text{MDI 4}}$	
34		
35	$\overline{\text{MDI 7}}$	
36		
37	$\overline{\text{MDI 6}}$	
38		
39		
40		
41	$\overline{\text{OUT}}$	CPU Module
42	$\overline{\text{ENABLE MON PROM}}$	Control Module
43	−10 VDC	Power Supply
44	−10 VDC	
45		
46		
47	$\overline{\text{CM-RAM 2}}$	CPU Module
48	$\overline{\text{CM-RAM 3}}$	
49		
50	$\overline{\text{CM-RAM 1}}$	

Table 6-3.

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

P1 Pin List (Continued)

PIN	SIGNAL FUNCTION	DESTINATION
51	I/O 1	
52	I/O 0	CPU Module
53	I/O 2	
54	$\overline{IN}$	CPU Module
55		
56	I/O 3	CPU Module
57		
58		
59		
60		
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		
72	$\overline{DATA\ 3}$	CPU Module
73		
74		
75		
76	$\overline{DATA\ 2}$	CPU Module
77		
78		
79	$\overline{SYNC}$	CPU Module
80	$\overline{DATA\ 1}$	CPU Module
81		
82		
83	$\overline{DATA\ 0}$	CPU Module
84		
85		
86		
87		
88	$\overline{4002\ RESET}$	Control Module
89		
90		
91		
92		
93		
94	CHIP SELECT C3	CPU Module
95		
96	CHIP SELECT C2	CPU Module
97		
98		
99		
100		

Table 6-3. (Continued)

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

J1 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1	ROM IN #4/8/12-0	Rear Panel Input Connector
2	ROM IN #4/8/12-1	
3	ROM IN #4/8/12-2	
4	ROM IN #4/8/12-3	
5	GROUND	
6	ROM IN #5/9/13-0	Rear Panel Input Connector
7	ROM IN #5/9/13-1	
8	ROM IN #5/9/13-2	
9	ROM IN #5/9/13-3	
10	GROUND	
11	ROM IN #6/10/14-0	Rear Panel Input Connector
12	ROM IN #6/10/14-1	
13	ROM IN #6/10/14-2	
14	ROM IN #6/10/14-3	
15	GROUND	
16	ROM IN #7/11/15-0	Rear Panel Input Connector
17	ROM IN #7/11/15-1	
18	ROM IN #7/11/15-2	
19	ROM IN #7/11/15-3	
20	GROUND	
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		

Table 6-4.

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

J1 Pin List (Continued)

PIN	SIGNAL FUNCTION	DESTINATION
1	ROM OUT #4/8/12-0	Rear Panel Output Connector
2	ROM OUT #4/8/12-1	
3	ROM OUT #4/8/12-2	
4	ROM OUT #4/8/12-3	
5	GROUND	
6	ROM OUT #5/9/13-0	Rear Panel Output Connector
7	ROM OUT #5/9/13-1	
8	ROM OUT #5/9/13-2	
9	ROM OUT #5/9/13-3	
10	GROUND	
11	ROM OUT #6/10/14-0	Rear Panel Output Connector
12	ROM OUT #6/10/14-1	
13	ROM OUT #6/10/14-2	
14	ROM OUT #6/10/14-3	
15	GROUND	
16	ROM OUT #7/11/15-0	Rear Panel Output Connector
17	ROM OUT #7/11/15-1	
18	ROM OUT #7/11/15-2	
19	ROM OUT #7/11/15-3	
20	GROUND	
21	$\overline{\text{RAM OUT \#4/8/12-0}}$	Rear Panel Output Connector
22	$\overline{\text{RAM OUT \#4/8/12-1}}$	
23	$\overline{\text{RAM OUT \#4/8/12-2}}$	
24	$\overline{\text{RAM OUT \#4/8/12-3}}$	
25	GROUND	
26	$\overline{\text{RAM OUT \#5/9/13-0}}$	Rear Panel Output Connector
27	$\overline{\text{RAM OUT \#5/9/13-1}}$	
28	$\overline{\text{RAM OUT \#5/9/13-2}}$	
29	$\overline{\text{RAM OUT \#5/9/13-3}}$	
30	GROUND	
31	$\overline{\text{RAM OUT \#6/10/14-0}}$	Rear Panel Output Connector
32	$\overline{\text{RAM OUT \#6/10/14-1}}$	
33	$\overline{\text{RAM OUT \#6/10/14-2}}$	
34	$\overline{\text{RAM OUT \#6/10/14-3}}$	
35	GROUND	
36	$\overline{\text{RAM OUT \#7/11/15-0}}$	Rear Panel Output Connector
37	$\overline{\text{RAM OUT \#7/11/15-1}}$	
38	$\overline{\text{RAM OUT \#7/11/15-2}}$	
39	$\overline{\text{RAM OUT \#7/11/15-3}}$	
40	GROUND	
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		

Table 6-5.

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

CHAPTER 7 THE imm6-26 PROM MEMORY MODULE

The PROM Memory Module expands the instruction storage capacity of the INTELLEC® 4/MOD 40 system, using 4702A erasable Programmable Read Only Memories.

The use of the 4702A PROM permits the user to commit partially developed programs, and program segments, to semi-permanent, non-volatile storage. The PROM is ideal for simulating mask programmed ROMs, during the early phases of system testing and development. Unlike programs committed to ROM storage, those in PROM may be altered, thus eliminating the objectionable delays involved in re-ordering mask programmed components.

The 4702A is erasable by exposure to ultraviolet and may be re-programmed as often as necessary, by using the PROM Programmer Module (imm6-76) in conjunction with the INTELLEC® 4/MOD 40 system. Chapter 11 of this manual describes the process in more detail. For complete instructions, refer to the INTELLEC® 4 Operator's Manual.

The imm6-26 has sockets for as many as sixteen PROMS. Each additional memory element adds 256 X 8 bits to the system's instruction storage capacity, permitting a maximum expanded capacity of 4096 X 8.

All the required decoding logic is incorporated into the module itself, but there is no provision for an expansion in excess of 4K. If more than one module is to be used, external decoding logic must be provided.

All inputs and outputs conform in their definition to TTL standards and interface directly with components of the 7400 logic family.

The PROM Memory Module plugs directly into the universal socket on the system's mother board. All power and control connections enter the module through this connector. No special installation is necessary. The PROM program memory is selected using a switch on the system console.

FUNCTIONAL DESCRIPTION OF THE MODULE

The PROM Memory Module consists of four functional blocks, as shown in Figure 7-1. They are:

a) PROM memory bank
b) address buffer
c) chip select decoder
d) output buffers

The PROM memory bank contains the actual PROM elements. The bank will hold up to sixteen 4702As. The eight addressing inputs of each element are connected in parallel to the corresponding inputs of all other PROMs in the bank. In similar fashion, the eight data outputs of each PROM share an output bus with the outputs of all the other PROMS.

The Central Processor Module applies a twelve-line address to the module, during M_1 and M_2 of every cycle. The eight low order address lines pass through the address buffer section, and are directed to the common address bus connecting all the PROMs in the memory bank.

The eight line component of the address specifies one of 256 eight-bit locations within the PROM. But though all the PROMs are addressed simultaneously, only one will respond to the address. That is the purpose of the chip select decoder.

The chip select decoder receives the four high order bits of the address from the Central Processor Module. It translates these four bits to a 1-of-16 output, which is used to enable one of the PROM elements on the card.

The enabled PROM responds by placing the contents of the addressed location on the module's eight-line output bus.

The output buffer receives and forwards this signal to the Central Processor Module, on the eight-line $\overline{\text{MDI}}$ bus.

NOTE: The sixteen switches on the PROM Memory Module are not used in the INTELLEC® 4/MOD 40 system.

THEORY OF OPERATION

Refer to the module schematic, Figure 7-2, for the following discussion.

The input buffer section consists of four 3404 hex latches. Only four of the latches on each chip are used in the address buffer. The sixteen latches form eight cascaded pairs, with their $\overline{\text{STROBE}}$ inputs tied permanently LOW by jumper 15-16. Each pair of latches thus functions as a simple non-inverting buffer for one of the eight address lines MA_0 - MA_7. The outputs of these buffers drive the module's internal address bus.

The chip select decoder consists of two 3205 3-line to 1-of-8 decoders, labelled A1 and A10 on the module schematic. The enabling pins of these two elements are wired so that A1 is enabled by the coincidence of a $\overline{\text{PROM ENABLE}}$ at module pin #82 and a simultaneous HIGH on MAD_{11}; a $\overline{\text{PROM ENABLE}}$ at pin #82 and a simultaneous HIGH on MAD_{11} enables A10. The eight output lines of A1 therefore command chips 0-7, while those of A10 command chips 8-15.

Each of the sixteen output lines from the decoder section enables one PROM on the module. This negative-true output is applied directly to the PROM's $\overline{CS_0}$ input (pin 14).

The output buffer consists of a pair of quad gated tri-state buffers, A26 and A27. Each of the memory output lines drives the input of one of the buffers. The buffers are enabled in parallel, by the $\overline{\text{PROM ENABLE}}$ signal applied to pin #82 of the module. Their eight-line output goes directly to the Central Processor Module, via the $\overline{\text{MDI}}$ bus.

Note that this module contains some logic which is intended for use in the INTELLEC® 8 systems. Users of the four-bit processor may disregard these provisions.

UTILIZATION

This section provides information on utilization of the imm6-28 PROM Memory Module, for those who contemplate using the module outside the INTELLEC® 4/MOD 40 system.

Installation

In installing the PROM Memory Module, the user must take account of:

a) environmental extremes

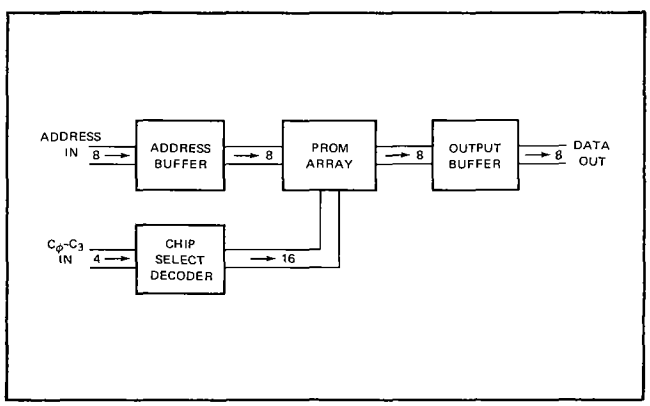


Figure 7-1. PROM Module Functional Block Diagram

b) mounting
c) electrical connections
d) power requirements
e) signal requirements
f) wiring options

ENVIRONMENT

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conductive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity is not critical to the module's operation.

MOUNTING

Avoid locating the card near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections. The result might be abnormally high noise levels, with the problems entailed, or outright failure of the card.

Dimensions of the module are 6.18 X 8.00 inches. Be sure to allow enough additional clearance to ensure adequate cooling.

The card is designed to plug directly into a standard 100-pin, double-sided PC edge connector. The connector will serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the card be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the card, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the elements on the module.

ELECTRICAL CONNECTION

All connections to the PROM Memory Module are effected by means of a standard 100-pin, double-sided PC edge connector with .125" centers. CDC #VPB 01C50E00A1 is one suitable type. Pin allocations are given on pages 85 and 86.

POWER REQUIREMENTS

The PROM Memory Module requires DC power, at the following levels:

+5 ± 5% VDC @	1.6	Amperes (max)
	1.1	Amperes (typ)
−10 ± 5% VDC @	1.6	Amperes (max)
	1.0	Amperes (typ)

Refer to the pin list for the power connections.

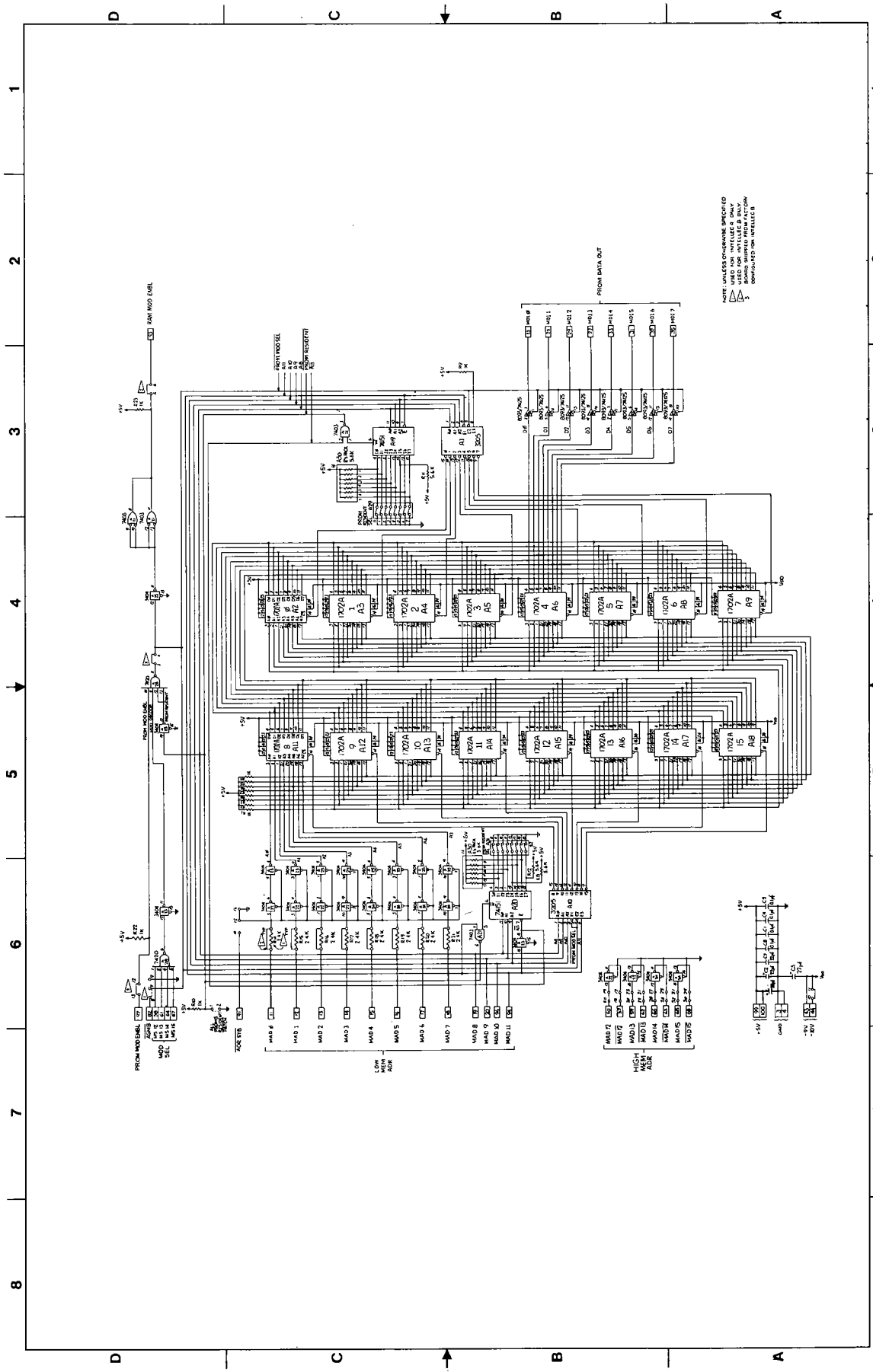


Figure 7-2. PROM Module Schematic Diagram

SIGNAL REQUIREMENTS

All inputs and outputs on the PROM Memory Module are TTL-compatible. Refer to Page 00 for pin assignments.

WIRING OPTIONS

The PROM Memory Module contains a number of optional wire jumpers. However, these are not applicable to systems that use either of the four-bit processors. Systems in this category include the INTELLEC® 4/MOD 40 system. Since the user has essentially no choice in this matter, the jumpering should be checked for conformance to the following scheme, before the module is installed in its slot:

DELETE	ADD
1-2	7-8
3-4	10-11
5-6	15-16
7-9	
14-15	

Expanding The Instruction Storage

The PROM Memory Module contains sockets for sixteen PROMs. The number of memory elements and their placement is up to the individual user. The physical location of the PROM, however, will determine its page identity, as indicated by the four high-order bits of the twelve-bit address.

Figure 7-3 shows the logical location of individual chips on the module.

PIN LIST

The following section describes connector pin allocations on the PROM Memory Module. The pins and their signal functions are listed in Table 7-1.

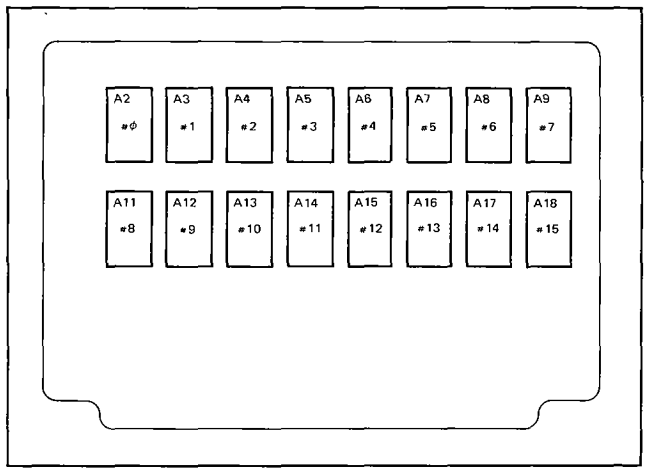


Figure 7-3. PROM Location Diagram

P1 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1		
2		
3	GROUND	
4	GROUND	
5		
6		
7		
8		
9		
10		
11	MEMORY ADDRESS 0	CPU Module
12	MEMORY ADDRESS 1	
13	MEMORY ADDRESS 2	
14	MEMORY ADDRESS 3	
15	MEMORY ADDRESS 4	
16	MEMORY ADDRESS 5	
17	MEMORY ADDRESS 6	
18	MEMORY ADDRESS 7	
19	CHIP SELECT C0	CPU Module
20	CHIP SELECT C1	
21		
22		
23	$\overline{\text{MDI 0}}$	To CPU Module
24		
25	$\overline{\text{MDI 1}}$	
26		
27	$\overline{\text{MDI 3}}$	
28		
29	$\overline{\text{MDI 2}}$	
30		
31	$\overline{\text{MDI 5}}$	
32		
33	$\overline{\text{MDI 4}}$	
34		
35	$\overline{\text{MDI 7}}$	
36		
37	$\overline{\text{MDI 6}}$	
38		
39		
40		
41		
42		
43	−10 VDC	Power Supply
44	−10 VDC	
45		
46		
47		
48		
49		
50		

Table 7-1.

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

P1 Pin List (Continued)

PIN	SIGNAL FUNCTION	DESTINATION
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		
72		
73		
74		
75		
76		
77		
78		
79		
80		
81		
82		
83		
84		
85		
86		
87		
88		
89		
90		
91		
92		
93		
94	CHIP SELECT C3	CPU Module
95		
96	CHIP SELECT C2	CPU Module
97		
98		
99	+5 VDC	Power Supply
100	+5 VDC	

Table 7-1. (Continued)

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

CHAPTER 8 THE imm4-24 DATA STORAGE MODULE

The imm4-24 Data Storage Module provides expanded working memory capacity, for the INTELLEC® 4/MOD 40 system.

The Data Storage Module has space for sixteen 4002 RAM memory elements. Four are provided with the module.

Each additional 4002 adds 80 X 4 bits to the processormodule's working memory capacity. Each memory element also contains a four-line output port, which coincidentally expands the system's I/O capacity.

Decoding logic contained within the module permits the use of as many as thirty-two 4002s in a single system. The use of two imm4-24 Data Storage Modules, each containing sixteen RAMs, provides up to 2560 X 4 bits of working storage and as many as thirty-two four-line output ports.

The Data Storage Module plugs directly into the universal socket on the system's mother board. All control and power connections enter through this connector. The module's output ports connect to the rear panel of the unit, through two connector plugs at the top of the card.

FUNCTIONAL DESCRIPTION OF THE MODULE

The Data Storage Module contains four functional sections:

a) the decoder
b) the level shifter
c) the RAM/output block
d) the RESET level shifter

The functional relationship among these sections is shown in Figure 8-1.

The decoder section receives a three-line input from the Central Processor Module, consisting of the $\overline{\text{CM-RAM}}$ command lines, 1 through 3. The decoder uses this information to selectively enable one of four memory banks within the module.

The four-line output from the decoder logic is applied to four level shifters. These translate the TTL outputs from the decoder to MOS levels suitable for driving the memory bank selection lines.

The RAM/output block contains the actual RAM elements and their associated output ports. This block consists of as many as sixteen RAMs, organized into four each. Every bank is commanded by one of the bank selection lines from the level shifter section.

Each of the four RAMs in a bank is distinguished from all the other RAMs on that line, by a two-digit binary number which prefixes the 6-bit address sent out from the Central Processor Module. The method of addressing the working memory is described in detail in Chapter 3 of the INTELLEC® 4/MOD 40 system Reference Manual.

Each RAM bank contains two 4002-1 RAMs and two 4002-2 RAMs. The two different versions are programmed in manufacture, to identify and distinguish them. Each version responds differently to the first digit of the address prefix.

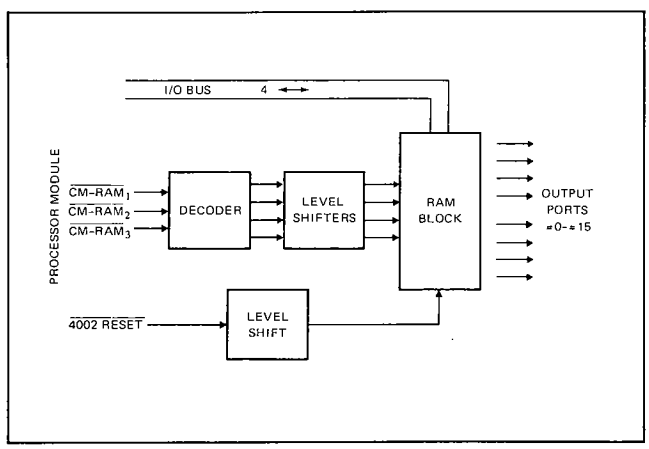


Figure 8-1. Data Storage Module Functional Block

Each chip also has an identity pin P_0, which is used as a secondary means of distinction. This pin (pin 10 of the element) is wired alternatively to the high or to the low side of the 15 Volt supply. There are thus four possible combinations of the two identifying features, as summarized in Table 8-1.

RAM Identification

RAM Version	P_0 (10)	Address
4002-1	+5 VDC	00XXXXXX
4002-1	−10 VDC	01XXXXXX
4002-2	+5 VDC	10XXXXXX
4002-2	−10 VDC	11XXXXXX

Table 8-1.

The chip addressing scheme applies both to memory reference operations and to output transactions.

The RESET level shifter section simply assures TTL compatibility. The $\overline{\text{4002 RESET}}$ from the INTELLEC® 4/MOD 40 system's Control Module is the input to this section. The output of the level shifter is used to clear the contents of all RAMs on the board. A 390-microsecond RESET pulse is necessary to ensure that all locations are cleared.

THEORY OF OPERATION

Chapter 2 of this manual describes the internal logic of the 4002 Random Access Memory and Four-Bit Output Port. The following discussion assumes familiarity with the functional operation of the RAM.

In the INTELLEC® 4/MOD 40 system, memory reference and output consist of three distinct operations. First a command line is designated (DCL), selecting one or more of the memory command lines $\overline{\text{CM-RAM}}_{0\text{-}3}$.

Next, an SRC is executed, transmitting two four-bit address bytes to the RAMs during X_2 and X_3 of the system cycle. The RAMs on the previously designated command line store and hold this address, pending the subsequent execution of an I/O or memory reference instruction.

When the processor fetches such an instruction, it enables the designated command line(s) during M_2 of the system cycle. The OPA byte of the instruction thus passes through to the control logic of the selected 4002, establishing the appropriate response to be made during the X_2 phase.

If the OPA indicates a memory read, the 4002 reacts during X_2 by placing the 4-bit contents of the addressed location on the main data bus. If a memory write is indicated, the 4002 responds during X_2 by storing the contents of the main data bus in the addressed memory location. If an output operation is indicated, the 4002 latches the contents of the main data bus and presents that data on its output lines.

Refer to the schematic of the Data Storage Module, Figure 8-2.

The decoder section of the module consists of the 3205 3-to-8 line converter, A21. $\overline{\text{CM-RAM}}_1$–$\overline{\text{CM-RAM}}_3$ are applied directly to the A_1-A_3 inputs of the decoder. The coincidence of an enabling signal to the decoder causes the decoder to activate one of eight output lines, depending upon the state of the $\overline{\text{CM-RAM}}_1$–$\overline{\text{CM-RAM}}_3$ inputs. The $\overline{\text{CM-ROM}}$ command line from the Central Processor Module is used as an enabling signal at pin #4 of the decoder, to prevent the "0" output line from being energized continuously when the other three command lines are inactive.

Wire jumpers connect four of the decoder's output lines to the inputs of four discrete two-stage level-shifters. Selection of these jumpers is the user's option. The exclusive output of each level-shifter is used to select and drive the enabling pins (pin #11) on a quad RAM bank.

The $\overline{\text{CM-RAM}}$ lines are activated at particular times in the system cycle, depending upon the kind of instruction being executed. Enabling of these lines permits the logic levels on the main data bus ($\overline{D}_0$-$\overline{D}_3$) to enter the RAM inputs and operate on the internal logic. When the SRC is executed, for example, the X_2 and X_3 contents of the bus are stored in the addressing section of the 4002 chips. Enabling of the command lines during M_2 of an I/O instruction permits the OPA code to enter the RAMs on the enabled bank, and prepare them for the impending I/O or memory reference.

The input lines of all 4002s are tied in parallel to the main data bus. The data on the bus thus passes to or from the selected RAM, according to the operation code transmitted to the RAM's internal logic. The steps involved in the execution are internal functions of the RAM and of the 4040 CPU on the Central Processor Module.

UTILIZATION

This section provides information on utilization of the imm4-24, for those who contemplate using the module outside the INTELLEC® 4/MOD 40 system.

Installation

In installing the Data Storage Module, the user must take account of:

a) environmental extremes
b) mounting
c) electrical connections
d) power requirements
e) signal requirements
f) wiring options

ENVIRONMENT

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating

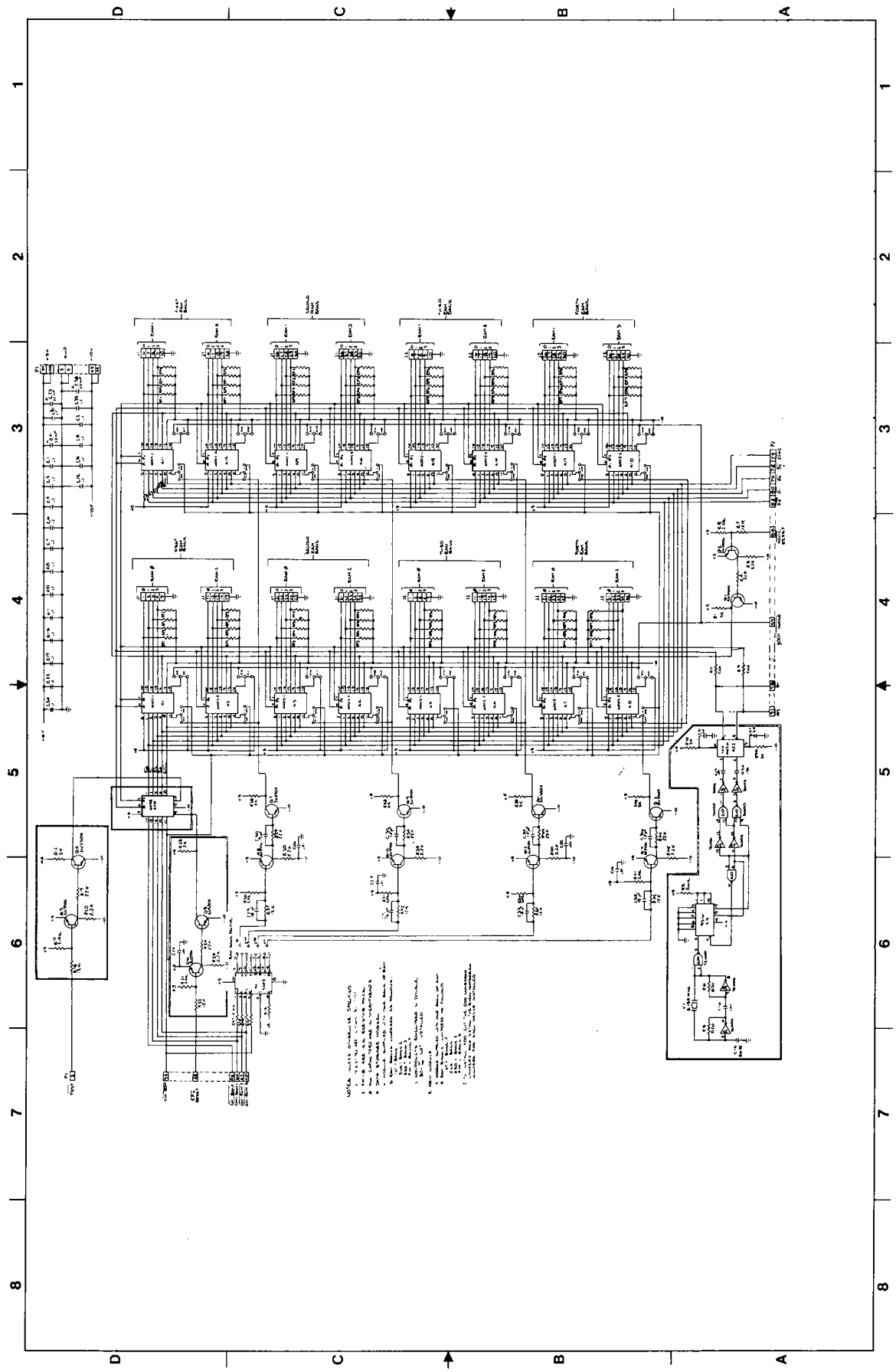


Figure 8-2. Data Storage Module Schematic Diagram

the module, giving particular attention to radiant and conductive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity is not critical to the module's operation.

MOUNTING

Avoid locating the card near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections. The result might be normally high noise levels, with the problems entailed, or outright failure of the card.

Dimensions of the module are 6.18 X 8.00 inches. Be sure to allow enough additional clearance to ensure adequate cooling.

The card is designed to plug directly into a standard 100-pin, double-sided PC edge connector. The connector will serve as a mounting, as well as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the card be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the card, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the elements on the module.

ELECTRICAL CONNECTION

All power and control connections to the Data Storage Module are effected by means of a standard 100-pin, double-sided PC edge connector, with .125" centers. CDC #VPB 01C50E00A1 is one suitable type. Pin locations on this connector are given in Table 8-3, at the end of this section.

All data inputs to the Data Storage Module pass through J1 and J2, at the top of the card. The appropriate mating connector is 3M #3432-1002. Pin allocations on J1 and J2 are given in Tables 8-4 and 8-5, at the end of this section.

In the INTELLEC® 4/MOD 40 system, both J1 and J2 are connected through 24 inches of ribbon cable to their associated input and output connectors on the rear panel of the unit.

POWER REQUIREMENTS

The Data Storage Module requires DC power, at the following levels:

+5 ±5% VDC @ 0.6 A (max)
0.3 A (typ)

-10 ±5% VDC @ 0.6 A (max)
0.3 A (typ)

Refer to pages 92-95 for the power connections.

SIGNAL REQUIREMENTS

All inputs and outputs on the Data Storage Module are at MOS or TTL levels. TTL levels conform to standard definitions. Pages 92-95 contains pin allocation data.

WIRING OPTIONS

The Data Storage Module has a number of wire link jumpers. The placement of most of these jumpers is NOT optional, for users of the INTELLEC® 4/MOD 40 system. All ODD-NUMBERED jumpers, from W1 through W63 must be installed, as shown on the module schematic.

The user does have four jumper options, which select and logically identify the RAM banks on the module. The expansion plan described below is recommended.

Expanding Data Storage

The basic Data Storage Module contains four RAM elements, but has space for as many as sixteen. If still more storage is required, a second module may be added. The addition of elements is the user's option.

The placement of RAM elements within the module determines their logical relationship to the larger memory bank. Judgement is necessary, when expanding the system to:

a) minimize the card count
b) avoid duplication of facilities
c) maintain a logical and consistent addressing structure

The plan of expansion summarized in Table 8-2 achieves all three purposes, and at the same time eliminates any functional overlap with RAM bank 0 on the Central Processor Module.

Note that two versions of the 4002 are used. The distinction is functionally important and must be observed carefully.

Figure 8-3 shows the physical location of the RAMs on the Data Storage Module, and identifies them logically.

Pin List

The following section describes connector pin allocations on the Data Storage Module. The pins and their signal functions are listed in Tables 8-3, 8-4, and 8-5.

Ckt Symbol	RAM Version	Chip I.D.	RAM Bank
MODULE #1			
A1	4002-1	0	#1
A11	4002-1	1	(A21-14 to Q8)
A2	4002-2	2	
A12	4002-2	3	
A3	4002-1	0	#2
A13	4002-1	1	(A21-13 to Q10)
A4	4002-2	2	
A14	4002-2	3	
A5	4002-1	0	#3
A15	4002-1	1	(A21-12 to Q12)
A6	4002-2	2	
A16	4002-2	3	
A7	4002-1	0	#4
A17	4002-1	1	(A21-11 to Q14)
A8	4002-2	2	
A18	4002-2	3	
MODULE #2			
A3	4002-1	0	#5
A13	4002-1	1	(A21-10 to Q10)
A4	4002-2	2	
A14	4002-2	3	
A5	4002-1	0	#6
A15	4002-1	1	(A21-9 to Q12)
A6	4002-2	2	
A16	4002-2	3	
A7	4002-1	0	#7
A17	4002-1	1	(A21-7 to Q14)
A8	4002-2	2	
A18	4002-2	3	

Table 8-2.

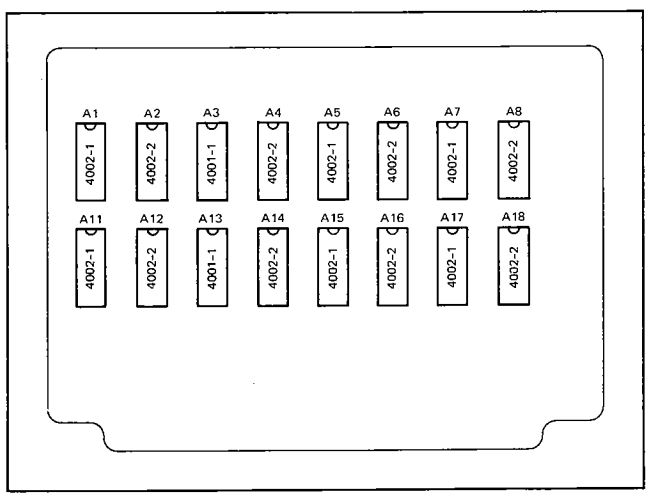


Figure 8-3. RAM Location Diagram

P1 Pin List

PIN	SIGNAL FUNCTION	DESCRIPTION
1		
2		
3	GROUND	
4	GROUND	
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43	−10 VDC	} Power Supply
44	−10 VDC	
45		
46		
47	$\overline{\text{CM-RAM 2}}$	} CPU Module
48	$\overline{\text{CM-RAM 3}}$	
49	$\overline{\text{CM-RAM 0}}$	
50	$\overline{\text{CM-RAM 1}}$	

Table 8-3.

NOTES:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

P1 Pin List (Continued)

PIN	SIGNAL FUNCTION	DESCRIPTION
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		
72	$\overline{\text{DATA 3}}$	CPU Module
73		
74		
75		
76	$\overline{\text{DATA 2}}$	CPU Module
77		
78		
79	$\overline{\text{SYNC}}$	CPU Module
80	$\overline{\text{DATA 1}}$	CPU Module
81		
82		
83	$\overline{\text{DATA 0}}$	CPU Module
84		
85		
86		
87		
88	$\overline{\text{4002 RESET}}$	CPU Module
89		
90		
01		
92		
93		
94		
95		
96		
97	$\overline{\text{Ø2}}$	CPU Module
98	$\overline{\text{Ø1}}$	
99	+5 VDC	Power Supply
100	+5 VDC	

Table 8-3. (Continued)

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

J1 Pin List

PIN	SIGNAL FUNCTION		DESCRIPTION
1	$\overline{\text{RAM OUT 0}}$		
2	$\overline{\text{RAM OUT 1}}$	BANK 1/5	Rear Panel Output Connector
3	$\overline{\text{RAM OUT 2}}$	RAM #0	
4	$\overline{\text{RAM OUT 3}}$		
5	GROUND		
6	$\overline{\text{RAM OUT 0}}$		
7	$\overline{\text{RAM OUT 1}}$	BANK 1/5	Rear Panel Output Connector
8	$\overline{\text{RAM OUT 2}}$	RAM #1	
9	$\overline{\text{RAM OUT 3}}$		
10	GROUND		
11	$\overline{\text{RAM OUT 0}}$		
12	$\overline{\text{RAM OUT 1}}$	BANK 1/5	Rear Panel Output Connector
13	$\overline{\text{RAM OUT 2}}$	RAM #2	
14	$\overline{\text{RAM OUT 3}}$		
15	GROUND		
16	$\overline{\text{RAM OUT 0}}$		
17	$\overline{\text{RAM OUT 1}}$	BANK 1/5	Rear Panel Output Connector
18	$\overline{\text{RAM OUT 2}}$	RAM #3	
19	$\overline{\text{RAM OUT 3}}$		
20	GROUND		
21	$\overline{\text{RAM OUT 0}}$		
22	$\overline{\text{RAM OUT 1}}$	BANK 2/6	Rear Panel Output Connector
23	$\overline{\text{RAM OUT 2}}$	RAM #0	
24	$\overline{\text{RAM OUT 3}}$		
25	GROUND		
26	$\overline{\text{RAM OUT 0}}$		
27	$\overline{\text{RAM OUT 1}}$	BANK 2/6	Rear Panel Output Connector
28	$\overline{\text{RAM OUT 2}}$	RAM #1	
29	$\overline{\text{RAM OUT 3}}$		
30	GROUND		
31	$\overline{\text{RAM OUT 0}}$		
32	$\overline{\text{RAM OUT 1}}$	BANK 2/6	Rear Panel Output Connector
33	$\overline{\text{RAM OUT 2}}$	RAM #2	
34	$\overline{\text{RAM OUT 3}}$		
35	GROUND		
36	$\overline{\text{RAM OUT 0}}$		
37	$\overline{\text{RAM OUT 1}}$	BANK 2/6	Rear Panel Output Connector
38	$\overline{\text{RAM OUT 2}}$	RAM #3	
39	$\overline{\text{RAM OUT 3}}$		
40	GROUND		
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			

Table 8-4.

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

J2 Pin List

PIN	SIGNAL FUNCTION		DESCRIPTION
1	$\overline{\text{RAM OUT 0}}$		
2	$\overline{\text{RAM OUT 1}}$	BANK 3/7	Rear Panel Output Connector
3	$\overline{\text{RAM OUT 2}}$	RAM #0	
4	$\overline{\text{RAM OUT 3}}$		
5	GROUND		
6	$\overline{\text{RAM OUT 0}}$		
7	$\overline{\text{RAM OUT 1}}$	BANK 3/7	Rear Panel Output Connector
8	$\overline{\text{RAM OUT 2}}$	RAM #1	
9	$\overline{\text{RAM OUT 3}}$		
10	GROUND		
11	$\overline{\text{RAM OUT 0}}$		
12	$\overline{\text{RAM OUT 1}}$	BANK 3/7	Rear Panel Output Connector
13	$\overline{\text{RAM OUT 2}}$	RAM #2	
14	$\overline{\text{RAM OUT 3}}$		
15	GROUND		
16	$\overline{\text{RAM OUT 0}}$		
17	$\overline{\text{RAM OUT 1}}$	BANK 3/7	Rear Panel Output Connector
18	$\overline{\text{RAM OUT 2}}$	RAM #3	
19	$\overline{\text{RAM OUT 3}}$		
20	GROUND		
21	$\overline{\text{RAM OUT 0}}$		
22	$\overline{\text{RAM OUT 1}}$	BANK 4	Rear Panel Output Connector
23	$\overline{\text{RAM OUT 2}}$	RAM #0	
24	$\overline{\text{RAM OUT 3}}$		
25	GROUND		
26	$\overline{\text{RAM OUT 0}}$		
27	$\overline{\text{RAM OUT 1}}$	BANK 4	Rear Panel Output Connector
28	$\overline{\text{RAM OUT 2}}$	RAM #1	
29	$\overline{\text{RAM OUT 3}}$		
30	GROUND		
31	$\overline{\text{RAM OUT 0}}$		
32	$\overline{\text{RAM OUT 1}}$	BANK 4	Rear Panel Output Connector
33	$\overline{\text{RAM OUT 2}}$	RAM #2	
34	$\overline{\text{RAM OUT 3}}$		
35	GROUND		
36	$\overline{\text{RAM OUT 0}}$		
37	$\overline{\text{RAM OUT 1}}$	BANK 4	Rear Panel Output Connector
38	$\overline{\text{RAM OUT 2}}$	RAM #3	
39	$\overline{\text{RAM OUT 3}}$		
40	GROUND		
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			

Table 8-5.

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

CHAPTER 9 THE imm4-60 INPUT/OUTPUT MODULE

The imm4-60 I/O Module is an optional plug-in module, designed to expand the input and output capacity of the basic INTELLEC® 4/MOD 40 system. It provides eight 4-line input ports and eight 4-line output ports which simulate the I/O ports normally associated with a 4001 ROM memory. All inputs and outputs are directly compatible with TTL logic components.

The addressing logic required for interface to the Central Processor Module is included on the module. The standard INTELLEC® 4/MOD 40 system can accommodate up to two of these modules, without the necessity for external addressing decoding.

Ports on the imm4-60 may be identified individually with any binary I/O address in the range of 0000 to 1111, the full addressing range of the Central Processor Module during input and output transactions. Addressing is established at the time the module is installed, by the placement of wire link jumpers on the printed circuit board. Modules will normally be shipped from the factory with input ports #4 through #11 and output ports #4 through #11 enabled, as is indicated on the schematic diagram in Figure 9-2. This eliminates any logical overlap with addressable ports on the Central Processor Module itself. However, the user is free to implement any optional addressing scheme, simply by shifting the position of individual jumpers as explained on page 100.

FUNCTIONAL DESCRIPTION OF THE I/O MODULE

The System During I/O Transactions

The Input/Output Module responds to commands that originate on the Central Processor Module of the INTELLEC® 4/MOD 40 system. It receives $\overline{IN}$ and $\overline{OUT}$ commands that select and synchronize the data transfer, and it receives a four-bit address on the chip select bus (C_0-C_3) that indicates the source or the destination of the data to be transferred. The actual data being exchanged travels between the CPU's accumulator and the addressed port via the four lines of the Central Processor Module's I/O bus (I/O_0-I/O_3).

The Central Processor Module initiates an input transaction in response to a "read ROM input" instruction (RDR) fetched from its program memory. When the processor module encounters such an instruction it is executed during the X_2 phase of that instruction cycle. The four-bit address is placed on the chip select bus. This address will automatically consist of the most significant four-bits indicated in the program's most recent "send register control" instruction (SRC); that is, the four binary digits that were contained in the even-numbered member of the index register pair specified by the instruction's operand field, at the time that the SRC was executed. Simultaneously, the processor module will activate the $\overline{IN}$ command line to the Input/Output Module. The logical function of the imm4-60 is then to decode the address presented to it, select the indicated input port, and forward the data that is present at the port during the IN strobe interval to the CPU's accumulator register, via the four lines of the I/O bus.

A "write ROM output" instruction (WRR) fetched from program memory causes the Central Processor Module to initiate an output transaction. Again, the processor executes during X_2 of the instruction cycle, by placing the current SRC pointer address on the four lines of its chip select bus. At the same time, the four-bit contents of the processor's accumulator register are made available on the four lines of the processor module's I/O bus, and the module's $\overline{OUT}$ command line is pulsed in order to synchronize the transfer. The Input/Output Module decodes the port address and forwards the $\overline{OUT}$ strobe to the selected port, causing it to latch and hold the data on the I/O bus for presentation to an external device.

Functional Description Of The I/O Module

Functionally, the Input/Output Module consists of five sections:

a) input ports
b) input select decoder
c) input gating
d) output ports
e) output select decoder

These blocks use the $\overline{IN}$, the $\overline{OUT}$, and the C_0 through C_3 chip select signals from the processor module, to develop the appropriate input or output response. The logical relationship is shown in Figure 9-1.

During input transactions, the input select decoder section uses the chip select address from the Central Processor Module to selectively activate one of the eight input ports, as shown in the diagram. Data at the inputs of the selected port is forwarded to the input gating section, where it is strobed through to the I/O bus by the $\overline{IN}$ command line from the Central Processor Module.

In an output transaction, the data to be transferred are made available to the eight output ports on the module, on the I/O bus which they share in common. The coincidence of an address on the chip select bus and an $\overline{OUT}$ command applied to the output select decoder causes the decoder to selectively strobe one of the eight output latches on the module. Data on the I/O bus at that time is recorded by the designated latch and is presented accordingly to the peripheral assigned to that port. Note that data will be available continuously at the port's outputs, until such time as it is displaced by data from a subsequent output transaction.

THEORY OF OPERATION OF THE I/O MODULE

Input

Refer to Figure 9-2. The input select decoder consists of two type 3205 Three-to-Eight Line Decoders, labeled A11 and A17 on the module schematic. The triple data inputs of these two decoders are wired in parallel to the chip select bus, lines C_0 through C_2.

Note that the fourth and most significant line of the chip select bus is wired to a negative-true enabling input on A17, and to the positive-true enabling input of A11. As a result, only one of these decoders may be active at any given time. One of the eight outputs of A17 will be pulled low whenever the chip select bus contains an address in the range of #0 to #7. An address between #8 and #15 on the other hand will cause one of A11's outputs to be activated. Thus, one of sixteen outputs will be enabled by any particular combination of states existing on the four lines of the chip select bus. Eight of these outputs are picked off by wire jumpers and are then used to control the multiplexing of inputs.

Four 8-to-4 line multiplexers, type SR 8234 are used for input multiplexing. Each of the components A1 through A4 produces an inverted four-line output whenever pin #7 or pin #9 is enabled. A low applied to pin #7 gates the input at pins #1, #6, #10, and #15 through to the outputs, while a low at pin #9 selects the data input on pins #2, #5, #11, and #14. Corresponding outputs on the four multiplexers are tied together, to form a single four-line output which is directed to the input gating logic.

In the gating logic section, the output of the multiplexer is inverted and applied to four tri-state buffers. The coincidence of an $\overline{IN}$ command and a low on one of the input port select lines generates a low at A5-3, enabling the tri-state buffer section and gating data from the addressed port through to the processor module's I/O bus.

Output

Output ports on the module are implemented by using

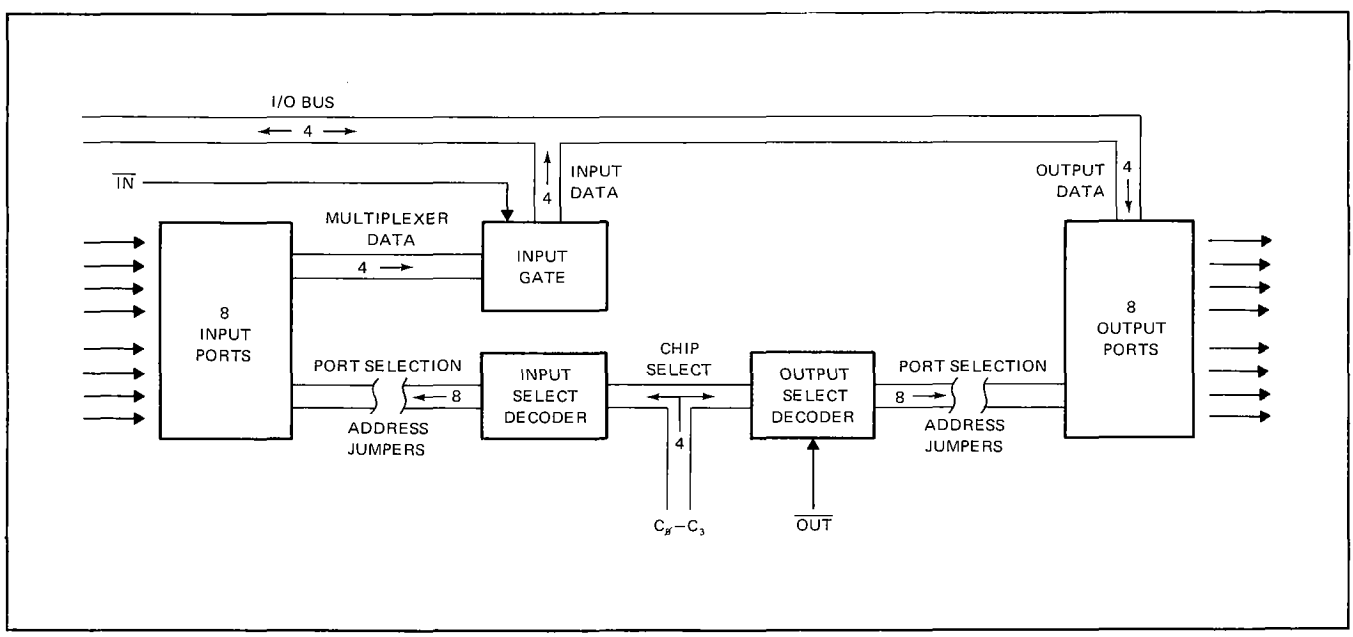


Figure 9-1. I/O Module Functional Block

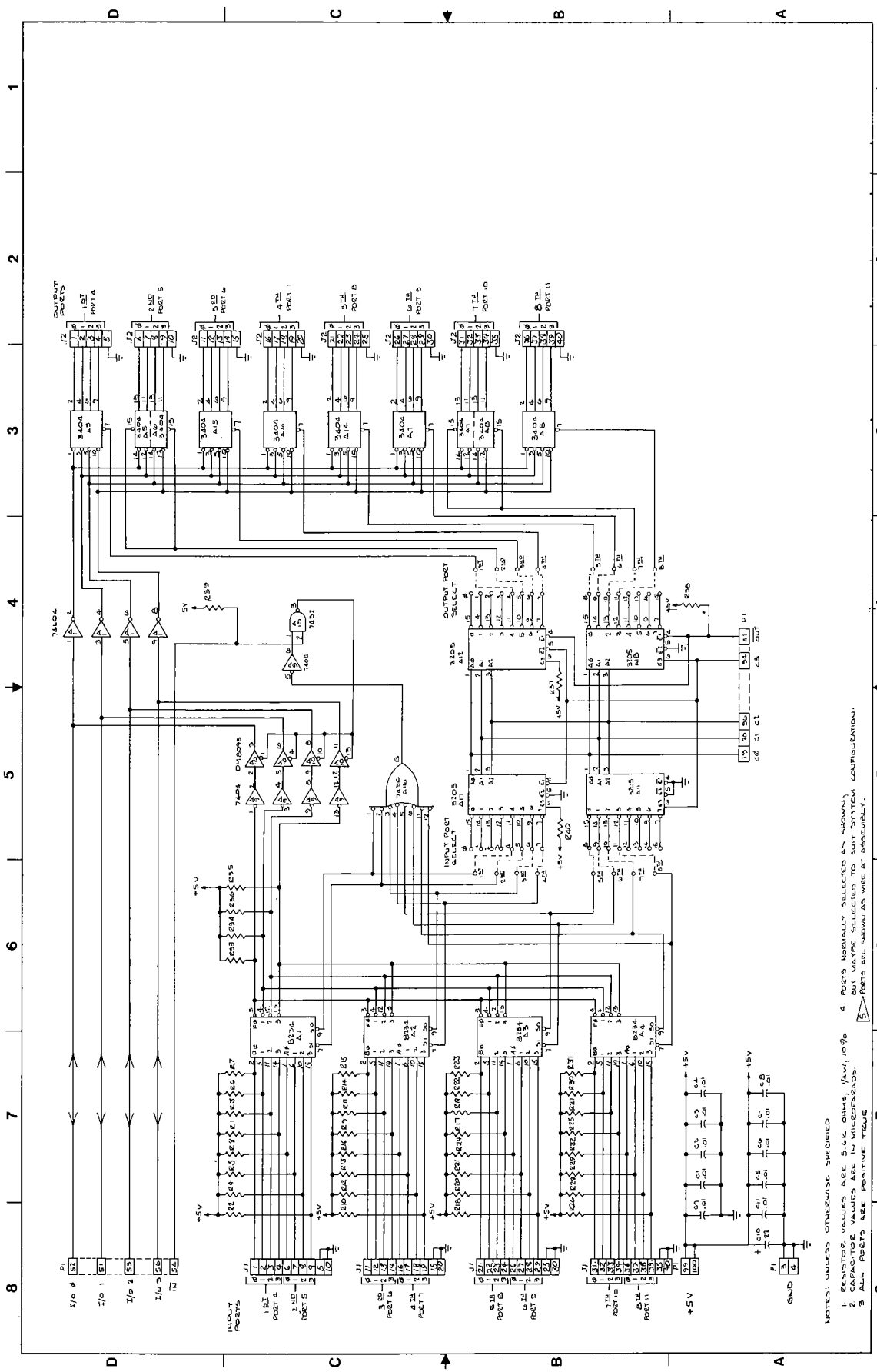


Figure 9-2. I/O Module Schematic Diagram

type 3404 hex inverting latches. Four such components are used, with individual latch sections grouped together by fours and with each group of four commanded by a single common strobe line. This provides eight individually addressable output ports.

During an output transaction, data on the I/O bus is inverted in four 7404 sections on the Input/Output Module and is bussed in common to the "D" inputs of all eight output ports on the board.

The decoding logic used for output is similar to that used during input, consisting of the two type 3205 decoders A12 and A18. In an output transaction, the coincidence of an address on the chip select bus and an $\overline{OUT}$ strobe applied to the negative-true enabling inputs of the decoders causes one of the sixteen output port select lines to go low. The negative-going edge of that transition is selected by wire jumper and coupled to the strobe input of one of the eight banked latches, causing the latch to register the data from the I/O bus and present it to the addressed peripheral.

UTILIZATION

This section provides information on utilization of the imm4-60, for those who contemplate using this module outside the INTELLEC® 4/MOD 40 system.

Installation

In installing the Input/Output Module, the user must take account of:

a) environmental extremes
b) mounting
c) electrical connections
d) power requirements
e) signal requirements
f) wiring options

ENVIRONMENT

Temperature extremes can cause instability, or result in permanent damage to the circuits on the module. Ambient temperature must therefore be maintained within the limits of 0° to 70° Centigrade. Exercise caution in locating the module, giving particular attention to radiant and conductive sources of heat. Remember that the module itself, when installed, will contribute some heat to the environment. Maintain an adequate clearance, to permit the convective dissipation of heat from the elements on the card.

Relative humidity is not critical to the module's operation.

MOUNTING

Avoid locating the card near vibrating machinery. Exposure to prolonged or violent vibration may cause fatigue or impact failure of connections. The result might be abnormally high noise levels, with the problems entailed, or outright failure of the card.

Dimensions of the module are 6.18 X 8.00 inches. Be sure to allow enough additional clearance to ensure adequate cooling.

The card is designed to plug directly into a standard 100-pin double-sided PC edge connector. The jack will serve as an electrical junction, if the environment is not too severe. Card guide slots are desirable, for the additional protection they afford. Should vibration be a problem, however, or should the card be used in a portable equipment application, an additional retaining bracket will have to be provided. When mounting the card, remember that it is desirable to orient the assembly vertically whenever possible. This optimizes convective cooling of the elements on the module.

ELECTRICAL CONNECTION

All power and control connections to the Input/Output Module are effected by means of a standard 100-pin, double-sided PC edge connector, with .125" centers. CDC #VPB 01C50E00A1 is one suitable type. Pin allocations on this connector are given in Table 9-1, at the end of this section.

All data inputs to the I/O Module are connected through J1, at the top of the card. The appropriate mating connector is 3M #3432-1002. Pin allocations on J1 are given in Table 9-2, at the end of this section.

All data outputs from the I/O Module pass through J2, at the top of the card. This connector is electrically identical to that used for inputs. Pin allocations on J2 are given in Table 9-3, at the end of this section.

In the INTELLEC® 4/MOD 40 system, both J1 and J2 are connected through 24 inches of ribbon cable to individual input and output connectors on the rear panel of the unit.

POWER REQUIREMENTS

The I/O Module requires only one source of DC power:

+5 ± 5% @ 0.9 Amperes (max)
@ 0.5 Amperes (typ)

Refer to the pin list on pages 102-105, for the power connections.

SIGNAL REQUIREMENTS

All inputs and outputs on the I/O Module are TTL compatible. Refer to pages 102-105 for pin assignments.

WIRING OPTIONS

Each input port and each output port on the imm4-60 is assigned a binary address between 0000 and 1111 (#0 to #15 decimal). The user may determine these port addresses individually, by the placement of wire jumpers on the printed circuit board. Jumpering options are shown on the schematic diagram of the module, Figure 9-2.

Note that every port may be characterized separately, by its **physical** address or by its **logical** address. Physical ports are identified by an ordinal descriptive, such as

"1st" or "5th" which always refers to the same port. Thus, the first output port on the imm4-60 always corresponds to pins #1 through #5 of the output connector J2. In like fashion, the fifth input port always corresponds to pins #21 through #25 of the input connector J1.

The port's logical address, on the other hand, is that descriptor by which the processor module identifies a particular port during input and output transactions. It corresponds to the four-bit binary address that the processor places on the chip select bus (C_0-C_3) during the X_2 phase of a "read ROM input" instruction (RDR) or a "write ROM output" instruction (WRR).

As Figure 9-2 shows, the correspondence between a port's physical address and its logical address is established by the INPUT PORT SELECT and the OUTPUT PORT SELECT strapping. The imm4-60 is normally shipped from the factory with jumpers which enable ports #4 through #11, to eliminate any overlap with ports which are physically and logically situated on the Central Processor Module in INTELLEC® 4/MOD 40 systems. It is a simple matter, however, to alter this scheme at the time of installation by replacing the standard jumpers with those of the user's own choice. To establish the logical address of the module's first input port as #1 (binary 0001), for example, you first remove the INPUT PORT SELECT jumper which connects the pad marked "1st" to the pad marked "4". Then install a link between the pad marked "1st" and that marked "1". Legends etched on the printed circuit board readily identify both INPUT PORT SELECT and OUTPUT PORT SELECT jumpers, so that it is a simple matter to verify the desired connections at the time of installation.

Pin List

The following tables give connector pin allocations on the Input/Output Module. Individual pins and their signal functions are listed in Tables 9-1, 9-2, and 9-3.

P1 Pin List

PIN	SIGNAL FUNCTION	DESTINATION
1		
2		
3	GROUND	
4	GROUND	
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19	CHIP SELECT C0	CPU Module
20	CHIP SELECT C1	
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41	$\overline{OUT}$	CPU Module
42		
43		
44		
45		
46		
47		
48		
49		
50		

Table 9-1.

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

P1 Pin List (Continued)

PIN	SIGNAL FUNCTION	DESTINATION
51	I/O 1	
52	I/O 0	CPU Module
53	I/O 2	
54	$\overline{IN}$	CPU Module
55		
56	I/O 3	CPU Module
57		
58		
59		
60		
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		
72		
73		
74		
75		
76		
77		
78		
79		
80		
81		
82		
83		
84		
85		
86		
87		
88		
89		
90		
91		
92		
93		
94	CHIP SELECT C3	
95		CPU Module
96	CHIP SELECT C4	
97		
98		
99	+5 VDC	Power Supply
100	+5 VDC	

Table 9-1 (Continued).

NOTE:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

J1 Pin List

PIN	SIGNAL FUNCTION		Port[2]	DESTINATION
1	ROM IN 0	1st	4	Rear Panel Input Connector
2	ROM IN 1			
3	ROM IN 2			
4	ROM IN 3			
5	GROUND			
6	ROM IN 0	2nd	5	Rear Panel Input Connector
7	ROM IN 1			
8	ROM IN 2			
9	ROM IN 3			
10	GROUND			
11	ROM IN 0	3rd	6	Rear Panel Input Connector
12	ROM IN 1			
13	ROM IN 2			
14	ROM IN 3			
15	GROUND			
16	ROM IN 0	4th	7	Rear Panel Input Connector
17	ROM IN 1			
18	ROM IN 2			
19	ROM IN 3			
20	GROUND			
21	ROM IN 0	5th	8	Rear Panel Input Connector
22	ROM IN 1			
23	ROM IN 2			
24	ROM IN 3			
25	GROUND			
26	ROM IN 0	6th	9	Rear Panel Input Connector
27	ROM IN 1			
28	ROM IN 2			
29	ROM IN 3			
30	GROUND			
31	ROM IN 0	7th	10	Rear Panel Input Connector
32	ROM IN 1			
33	ROM IN 2			
34	ROM IN 3			
35	GROUND			
36	ROM IN 0	8th	11	Rear Panel Input Connector
37	ROM IN 1			
38	ROM IN 2			
39	ROM IN 3			
40	GROUND			
41				
42				
43				
44				
45				
46				
47				
48				
49				
50				

Table 9-2.

NOTES:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

[2] Port number is as wired at factory.

J2 Pin List

PIN	SIGNAL FUNCTION	Port[2]	DESTINATION
1	ROM OUT 0	1st 4	Rear Panel Output Connector
2	ROM OUT 1		
3	ROM OUT 2		
4	ROM OUT 3		
5	GROUND		
6	ROM OUT 0	2nd 5	Rear Panel Output Connector
7	ROM OUT 1		
8	ROM OUT 2		
9	ROM OUT 3		
10	GROUND		
11	ROM OUT 0	3rd 6	Rear Panel Output Connector
12	ROM OUT 1		
13	ROM OUT 2		
14	ROM OUT 3		
15	GROUND		
16	ROM OUT 0	4th 7	Rear Panel Output Connector
17	ROM OUT 1		
18	ROM OUT 2		
19	ROM OUT 3		
20	GROUND		
21	ROM OUT 0	5th 8	Rear Panel Output Connector
22	ROM OUT 1		
23	ROM OUT 2		
24	ROM OUT 3		
25	GROUND		
26	ROM OUT 0	6th 9	Rear Panel Output Connector
27	ROM OUT 1		
28	ROM OUT 2		
29	ROM OUT 3		
30	GROUND		
31	ROM OUT 0	7th 10	Rear Panel Output Connector
32	ROM OUT 1		
33	ROM OUT 2		
34	ROM OUT 3		
35	GROUND		
36	ROM OUT 0	8th 11	Rear Panel Output Connector
37	ROM OUT 1		
38	ROM OUT 2		
39	ROM OUT 3		
40	GROUND		
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			

Table 9-3.

NOTES:

[1] All signal inputs and outputs are at TTL levels, unless otherwise noted.

[2] Port number is as wired at factory.

CHAPTER 10
INTELLEC® 4/MOD 40 SYSTEM MAINFRAME

The mainframe assembly of the INTELLEC® 4/MOD 40 system provides the chassis, mother board, and power supplies that unify the system.

Functionally speaking, the mainframe is very simple. Little explanation is necessary.

POWER SUPPLIES

The INTELLEC® 4/MOD 40 system uses two OEM power supplies. One supplies power of +5 ± 5% VDC, at 12 Amperes (see Figure 10-1). The other supplies −10 ± 5% VDC, at 1.8 Amperes, and +80 VDC for the PROM Programmer Module (see Figure 10-2). Supply current is more than adequate to furnish the load presented by the basic system. Figures 10-1 and 10-2 also list the jumper connections for converting from 115 VAC to 230 VAC.

CHASSIS LAYOUT

Figure 10-3 illustrates the internal layout of the INTELLEC® 4/MOD 40 system. As shown, the Central Processor Module, the Control Module, and the RAM Memory Module occupy specific dedicated locations on the mother board. Connectors J7-J19 on the other hand, are ports on a universal bus which is designed to accept optional INTELLEC® 4 modules in any combination or arrangement.

SYSTEM INTERCONNECTION

Figure 10-4 shows the connections affected on the mother board. Figure 10-5 shows cable routing within the unit.

Each of the two power supplies plugs into a designated connector on the mother board.

Two pin-to-pin cables connect J1 and J2 on the Control Module to J1 and J2 on the Control and Display Panel.

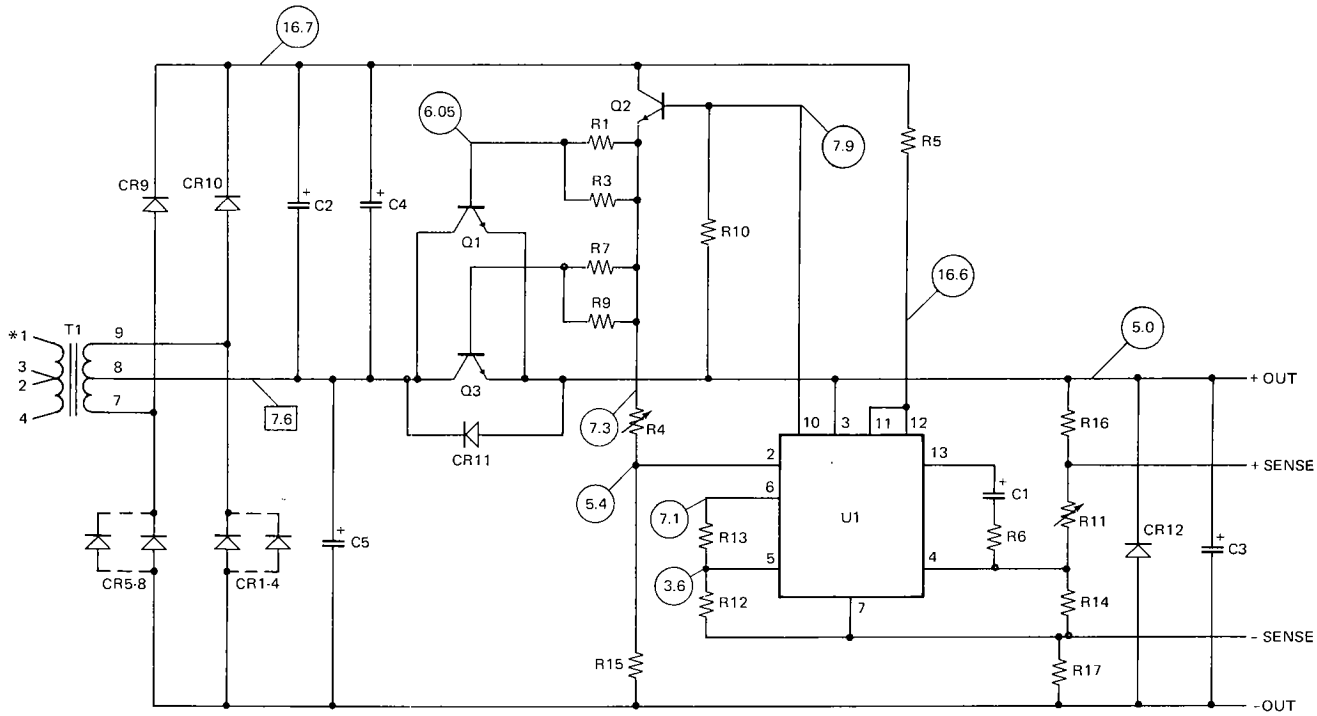
J42 on the mother board connects the Central Processor Module's TTY interface to the TTY rear panel connector (J43).

J2 on the mother board connects the Central Processor Module's output ports to the rear panel output connector.

J3 on the mother board connects the Central Processor Module's input ports to the rear panel input connector.

Ribbon cables connect the output ports on each auxiliary module to the rear panel output connector. Similar cables connect the input ports to the rear panel input connector.

The PROM Programmer Module plugs into any vacant connector on the INTELLEC® 4's universal socket. Ribbon cables connect the module's data ports to J1 of the mother board and to the PROM programming socket on the Control and Display Panel. The umbilical cable attached to the programmer module plugs into socket J34 on the chassis of the −10 VDC power supply, providing 80 VDC unregulated power and enabling signals to the module.



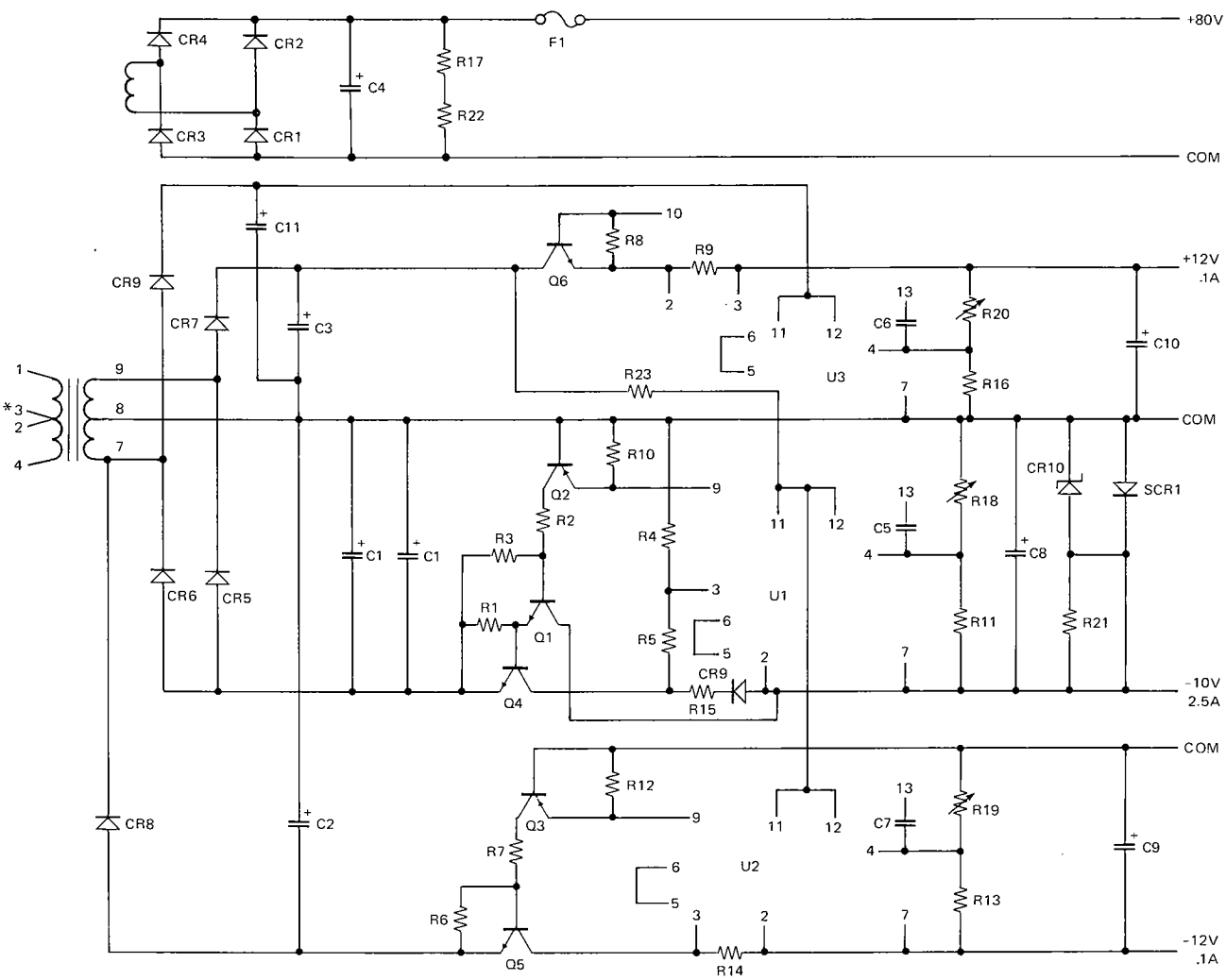
*AC CONNECTION TABLE:

For Use At:	Jumper:	Apply AC:
115 VAC	1-3, 2-4	1 & 4
230 VAC	2-3	1 & 4

PARTS LIST:

Ref. Des.	D5-120	Description
C1	.001/100	CAPACITOR, FILM
C2, 3, 4	1000/16	CAPACITOR, ELECTROLYTIC
C5	40000/15	
CR108, 12	AE3B	RECTIFIER 3A, 100V
CR9, 10, 11	AE1C	RECTIFIER 1A, 200V
R1, 3, 5, 7, 9, 16, 17	7.5Ω	RESISTOR, 1/2W, 5%, C.F.
R6, 0, 2, 13, 15	2.2K	
214	1.6K	RESISTOR, 1/4W, 2%, M.F.
R4, 11	1.5K	POTENTIOMETER, 2W, W.W.
A1, 2, 3	2N3055	TRANSISTOR POWER
U1	μA723	I.C. VOLTAGE REGULATOR
T1	11053-5	POWER TRANSFORMER
P.C.B.	11050	PRINTED CIRCUIT BOARD
CHASSIS	11051	ALUMINUM ALODINE COATED

Figure 10-1. Power Supply Schematic (±5 VDC)



*AC CONNECTION TABLE:

For Use At:	Jumper:	Apply AC:
115 VAC	1-3, 2-4	1 & 4
230 VAC	2-3	1 & 4

PARTS LIST:

Ref. Des.	CP110	Description	Ref. Des.	CP110	Description
CR1, 2, 3, 4, 7, 8	AE1C	DIODE, 1A, 200V	R1, 2, 3,6, 7,10,	1.6K	Resistor, 1/2W Carbon
CR5, 6, 9	AE3B	DIODE, 3A, 100V	11, 12, 13, 16	1.6K	Resistor, 1/2W Carbon
CR10	IN759A	Zener, 400 mv, 12V	R17, 22, 8	10K	Resistor, 1/2W Carbon
SCR1	S0303LS3	3A SCR	R4	4.7K	Resistor, 1/2W Carbon
U1, 2, 3	UA273	1C Voltage Regulator	R5, 23	270Ω	Resistor, 1/2W Carbon
Q1	2N2218A	TRANSISTOR, SIGNAL, NPN	R15	.12Ω	Resistor, 2W, WW, BWH
Q2, 3	2N2906	TRANSISTOR, SIGNAL, PNP	R9, 14	4.7Ω	Resistor, 1/2W Carbon
Q4, 5, 6	2N3055	TRANSISTOR, POWER, NPN	R21	47Ω	Resistor, 1/2W Carbon
F1	0.5A	Fuse, 3AG Fast Blow	R18, 19, 20	1500Ω	Potentiometer, 2W, WW
C1	4000/30	Capacitor, Alum., Elect.			
2, 3	470/25	Capacitor, Alum., Elect.	T1	12030A	Power Transformer
4	500/100	Capacitor, Alum., Elect.			
8, 9, 10	100/25	Capacitor, Alum., Elect.			
5, 7	.01/100	Capacitor, Mylar., Elect.			
6	.001/100	Capacitor, Alum., Elect.			

Figure 10-2. Power Supply Schematic (-10 VDC and +80 VDC)

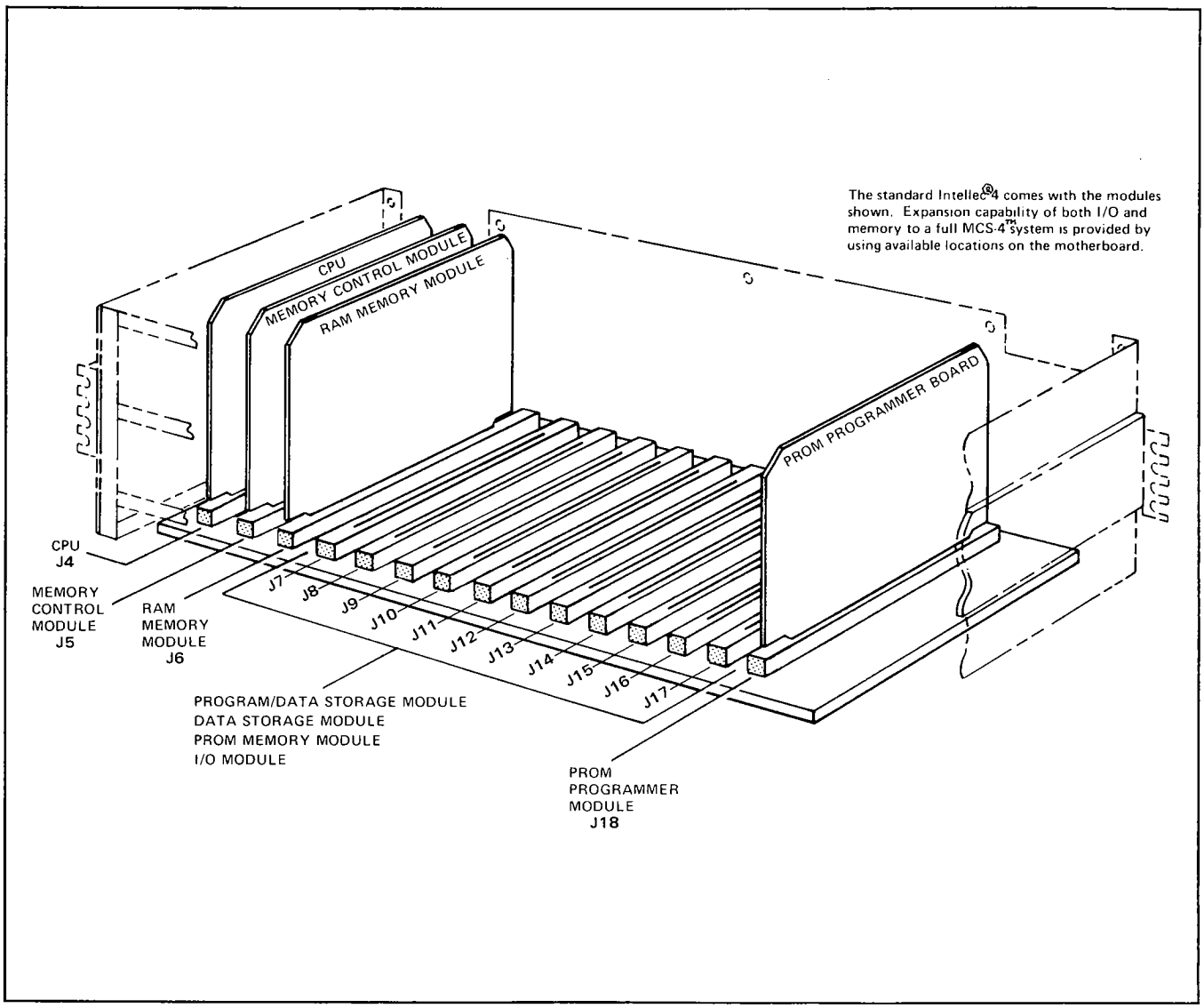


Figure 10-3. INTELLEC® 4 Chassis Layout

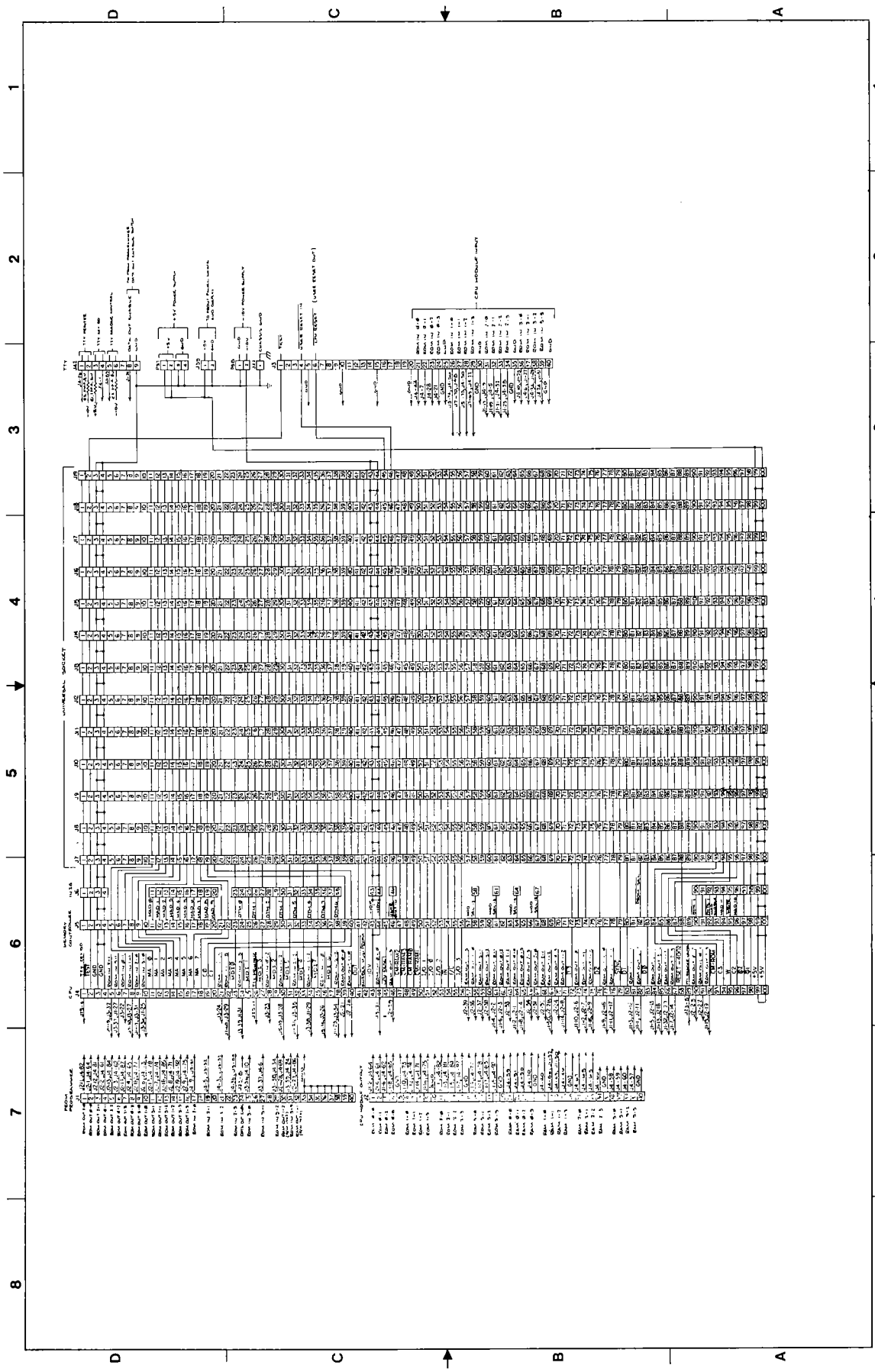


Figure 10-4. INTELLEC® 4 Mother Board Schematic

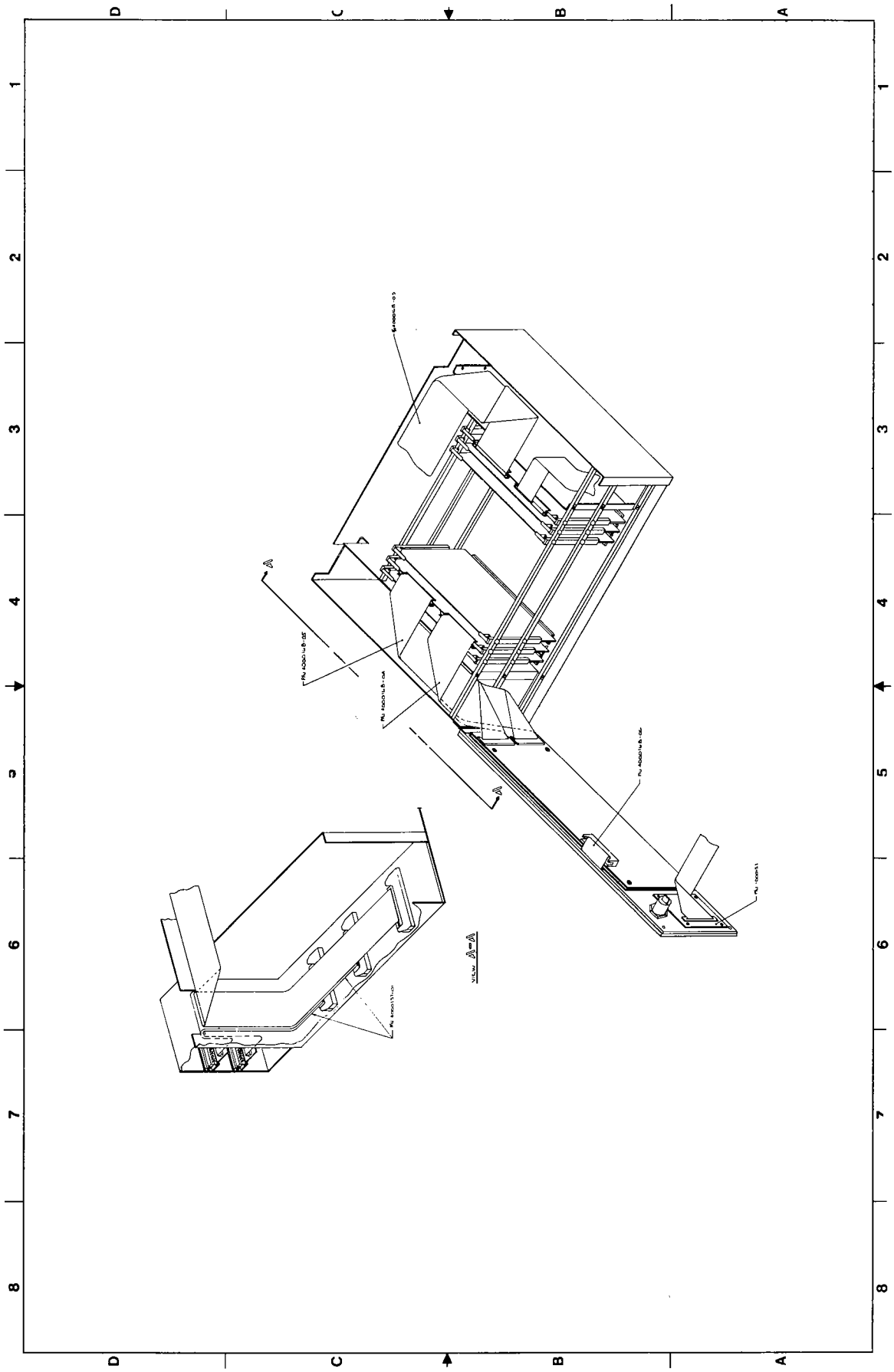


Figure 10-5. Internal Cabling

CHAPTER 11 THE imm6-76 PROM PROGRAMMER MODULE

The imm6-76 PROM Programmer Module is a standard part of the INTELLEC® 4/MOD 40 system. When used in conjunction with the INTELLEC® 4/MOD 40 System Monitor, the programmer module permits rapid, automatic loading of Intel 4602A and 4702A Programmable Read Only Memories.

The program to be transferred to a PROM is first stored in the system's program RAM memory. The PROM to be programmed is erased, if necessary, and inserted in the programming socket on the Control and Display Panel. The PRGM PROM PWR switch is turned on, and the console operator types a 'P', followed by parameters which indicate the first and the last RAM addresses to be transferred, as well as the starting address in the PROM.

The software does the rest. It transfers the eight bits of the PROM address to ROM output ports 0 and 1. It sets up the data to be written into the PROM, at ROM output ports 2 and 3. It pulses the power supply the required number of times, at the required duty cycle. And it checks the results of its programming, by reading the PROM's output through ROM input ports 2 and 3. If improper programming is indicated, the System Monitor prints an exception notice at the teletype console. This programming cycle is repeated at each of the memory locations bracketed by the initial and the terminal parameters. Complete programming involves the loading of 256 individual locations, a process which requires approximately 2 minutes. The procedure is described fully in the INTELLEC® 4 Operator's Manual.

The imm6-76 is designed for plug-in installation in the INTELLEC® 4/MOD 40 system mainframe. It makes use of existing connectors and other provisions. No special installation is necessary.

THE 4702A PROGRAMMABLE READ ONLY MEMORY

The 4702A PROM is a 256 x 8 bit electrically programmable erasable read-only memory. The 4702A is programmed by the momentary application of high amplitude pulses on selected pins of the chip. The 4702A, however, may be cleared by controlled exposure to high intensity ultraviolet. The device may be reloaded as often as desired, making it particularly suitable for use in program development.

Programming of the 4702A requires a carefully controlled sequence of operations. The safety of the chip demands that both the interelement voltages and the duty cycle of the programming pulses be maintained within specific limits. This insures against breakdown and overheating. On the other hand, insufficient power levels will lead to programming failures. An accurate balance is necessary. The PROM Programmer Module is designed to provide pulses of the correct level, duration, and frequency automatically.

The imm4-76 is designed to program the 4702A but the utility program contained in the INTELLEC® 4/MOD 40 System Monitor is not set up for the programming of 4602 and 4702 PROMs. As a result, any attempt to load 4602 or 4702 memories with the INTELLEC® 4/MOD 40 System Monitor will result in damage to the PROM. Such programming is possible, with the proper precautions, but you will have to provide your own software functions. Refer to the INTEL® MEMORY DESIGN HANDBOOK for instructions, if you plan to use the imm4-76 for this purpose.

The 4702A is shipped to the customer in a "cleared" condition; that is, with zeros in all memory locations. An internal zero-state is indicated by a HIGH on the output pins of an enabled chip. During programming ones are loaded selectively into the chip's memory locations.

A 4702A which has been programmed previously must be erased prior to reloading. Erasure is accomplished by exposing the silicon die to ultraviolet light. The device is made with a transparent quartz lid, to permit such exposure. Conventional room light, flourescent light, and sunlight have no measurable effect on data stored in the 4702A, even

after years of exposure. But the device is cleared quickly by a brief exposure to high intensity ultraviolet at a wavelength of 2537 Angstroms. The Model UVS-11 (Ultraviolet Products, Incorporated: San Gabriel, California) is a cheap and effective source for this purpose. Its accompanying filter must first be removed. The recommended integrated dose (the product of intensity and the exposure time) is 6 W-sec/cm^2. Ten minutes exposure to the UVS-11, at a distance of 1 inch, will clear the PROM completely. Avoid unnecessary or prolonged exposures, which are potentially damaging to the PROM.

—WARNING—

High intensity ultraviolet can cause serious burns. Ultraviolet radiation can also generate potentially hazardous amounts of ozone. Observe the following precautions, when using the UVS-11 to erase a PROM:

1) Never expose skin or eyes to the source directly.
2) Do not stare fixedly at an object which is under ultraviolet illumination. The light is invisible, but is nevertheless injurious to eye tissues.
3) Use the source only in a well-ventilated area.

FUNCTIONAL DESCRIPTION OF THE MODULE

An eight-line input, applied to the PROM's addressing lines, specifies the location to be programmed. Data to be written in that location is applied to the chip's eight output lines. Then address lines, data lines, the PRGM pin, and all four power lines (V_{CC}, V_{bb}, V_{gg}, and V_{DD}) are pulsed, to fix the data in location. The procedure requires about 3 milliseconds, and the cycle is repeated 32 times at each of the 256 memory locations. To prevent overheating of the 4702A, the programmer module maintains a 20% duty cycle, and it therefore takes approximately 123 seconds to program the entire chip.

To perform the required functions, the imm4-76 contains an address driver bank, a data driver bank, four electronically controlled power supplies, and a control and timing section.

The sequence of events is as follows:

1) Data to be programmed into the PROM is placed on the input lines, in complement (negative-true) form.
2) Address to be programmed is placed on the address lines, in complement form.
3) When the programming cycle begins, the following changes in the static conditions occur:
   a) V_{CC} switches from 5 to 47 Volts.
   b) V_{bb} switches from 5 to 59 Volts.
   c) V_{gg} switches from −10 to 12 Volts.
   d) V_{DD} switches from −10 to 0.6 Volts.
   e) The programming signal (PRGM) goes from 0 to 47 Volts.
   f) Address data changes from 0-5 Volts to 0-47 Volts.
4) 60 microseconds after the cycle begins, the address data is switched from its complement form to its positive-true form.
5) 155 microseconds after the cycle begins, the PRGM signal dips from 47 Volts to approximately 9 Volts.
6) 3 milliseconds later, teh PRGM signal returns to 47 Volts.
7) 3.25 milliseconds after the beginning of the cycle, all voltages and signals are switched back to their normal quiescent levels.
8) 15 milliseconds after the beginning of the first cycle, the second cycle begins.

INTERFACE TO THE INTELLEC® 4/MOD 40 SYSTEM

Note that the timing relationships above are determined by control circuitry on the PROM Programmer Module itself. The number of pulsed repetitions, however, is determined by the controlling program. The INTELLEC® 4/MOD 40 System Monitor contains a timing routine which holds the PROM programmer enabled for approximately 520 milliseconds, or 35 programming cycles, before stepping to the next memory location.

The ADDRESS IN lines on the programmer module are connected to the INTELLEC® system's ROM output ports #0 and #1. The DATA IN lines are connected internally to ROM output ports #2 and #3. The INTELLEC® 4/MOD 40 System Monitor writes into these ports when a PROM is being programmed.

When the programmer module is not actively programming a memory location, the contents of that location are available at the module's DATA OUT pins. These outputs are connected in turn to ROM input ports #2 and #3, so that the INTELLEC® 4/MOD 40 System Monitor can check the results of its programming.

The PROM Programmer Module also has two negative-true enabling inputs, which initiate the programming cycle. A LOW applied to pin #32 of the module selects a 20% programming duty cycle. This input is used when programming 4702A PROMs. A LOW applied to pin #30 selects A 2% duty cycle, used when programming 4602 and 4702 devices. In the INTELLEC 4/MOD 40 system, pin #30 is connected to the BIT #2 line of the same output port. The INTELLEC® 4/MOD 40 System Monitor controls the programmer module by writing into that port.

THEORY OF OPERATION OF THE MODULE

Refer to Figure 11-1, the schematic diagram of the PROM Programmer Module.

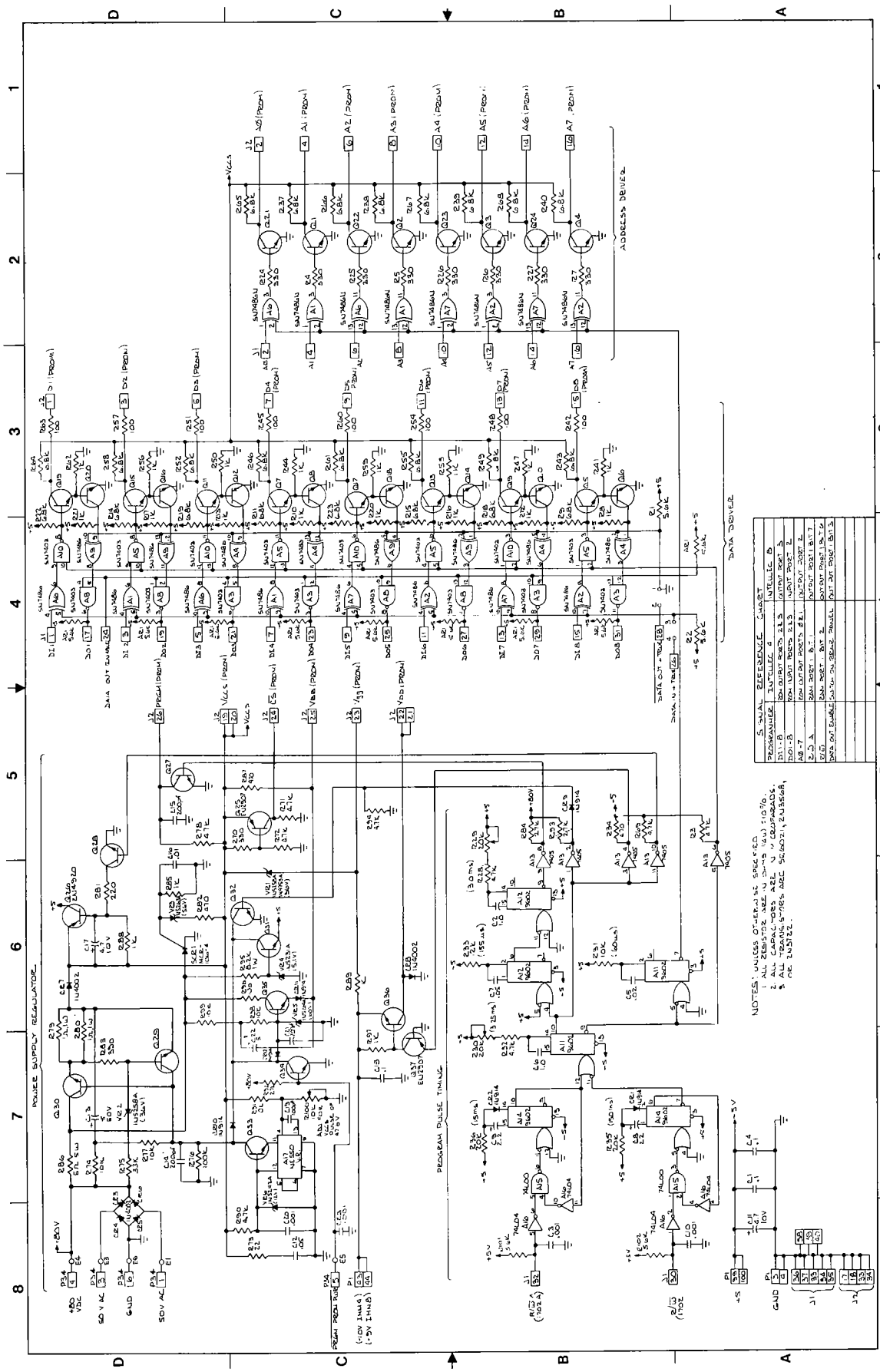


Figure 11-1. PROM Programmer Schematic Diagram

Data Distribution

The data to be programmed into the PROM originates at ROM output ports #2 and #3. The eight signal lines enter the programmer module through a ribbon cable which runs from J1 on the INTELLEC®4/MOD 40 system mother board to J1 at the top of the module. Each of the input lines is applied to one input of an XOR-gate. The alternate inputs of these eight gates are returned through a common line to the +5 Volt supply, so that each gate acts as an inverter to the incoming data.

Each of the XOR-gate outputs is directed to one input of a 7403 NAND-gate. The alternate inputs to this bank of gates are driven in common by a signal originating in the control and timing section of the module. At the appropriate time in the cycle, these inputs are permitted to swing HIGH, causing data from the XOR-gate bank to pass through to the bases of eight driver transistors: Q19, Q15, Q11, Q7, A17, A13, A9, and Q5. The signals at the collectors of these drivers are conducted out of the assembly through a ribbon cable which attaches to J2 at the top of the module. It goes from there to the programming socket on the front panel of the INTELLEC®4/MOD 40 system. This data undergoes three successive inversions, between entering and leaving the imm4-76, and the output will therefore be in complementary form, as required for the programming of the 4702A PROM.

Observe that the bases of the PROM data driver transistors are returned through pull-up resistors to the +5 Volt supply. As a result, these transistors will be conducting whenever the input NAND-gates are inhibited. Under these circumstances, the signal at each of the PROM's data pins will be applied to the base of a transistor, through a divider consisting of a 100-ohm resistor, the DC collector resistance of a driver transistor, and a 1K resistor. Transistors Q20, Q16, Q12, Q8, Q18, Q14, Q10, and Q6 amplify these eight signals and forward them to an XOR-gate bank which is used as an eight-line data inverter. The outputs of the XOR-gates are applied in turn to eight NAND-gates which have their alternate inputs tied in common to J1-24. That pin is returned to +5 Volts through a 5.6K pull-up resistor, as shown in Figure 11-1. When the programming toggle switch on the INTELLEC®system rear panel is ON, the inhibitory external ground on J1-24 is removed, permitting that pin to swing high and enabling the eight NAND-gates. The enable gates act as inverters, to produce a positive-true output. This is routed out of the assembly at J1, through a ribbon cable to J1 on the mother board, terminating eventually at ROM input ports #2 and #3 of the Central Processor Module. The INTELLEC®4/MOD 40 System Monitor reads these ports in order to determine the results of its programming.

Address data enters the module at J1, through a ribbon cable connecting it to J1 of the system's mother board. Data originating at ROM output ports #0 and #1 is therefore applied to the eight-line XOR-gate bank, shown on the right in Figure 11-1. The outputs of these gates are directed to the bases of eight driver transistors, whose outputs command the PROM address lines. Note that the alternate inputs of the XOR-gates are tied in common to a signal line from the control and timing section. This line swings LOW when the programming cycle begins. It returns to a high HIGH condition 60 microseconds later. As a result, the address forwarded to the PROM is in complementary form initially. Sixty microseconds after the programming cycle begins the address data will switch to its positive-true form, in accordance with the PROM's programming requirements.

Control and Timing

As shown in Figure 11-1, the programming cycle may be initiated by a LOW applied to pin #32 or to pin #30 of the card. The INTELLEC®4/MOD 40 System Monitor enables the pin #32 input, selecting a duty cycle of 20% (3 mS/15 mS). The pin #30 input is set up for the 2% duty cycle used to program 4602 and 4702 devices.

When a LOW is applied to pin #32 of the module, the 15 millisecond input multivibrator re-triggers itself repetitively, until the enabling signal is removed. This provides a series of positive-going excursions with a period of 15 milliseconds, which are used to trigger the 3.25 millisecond program cycle one-shot.

The output of the program cycle one-shot:

1) Complements the address to the PROM.
2) Enables the data drivers.
3) Pulses all four power supplies.
4) Triggers a 155 microsecond cascaded one-shot delay.

Sixty microseconds after the program cycle one-shot fires, the negative-going pulse output at A11-7 subsides, and the address data returns to its positive-true form.

One hundred fifty-five microseconds after the program cycle one-shot fires, A12-9-10-11-12-13-14 fires, causing the power supply to apply a 3 millisecond PRGM pulse to the PROM.

Three and a quarter milliseconds after the beginning of the programming cycle, all signals return to their quiescent levels.

The programmer module's control timing is illustrated in Figure 11-2.

Power Supply

The power supply section of the PROM Programmer Module performs the level switching functions required to program PROMs, in response to signals which are generated in the timing and control section of the module. The power supply contains a rectifier section, a voltage regulator section and six output switches. The relationship among these is shown in a simplified form, in Figure 11-3.

RECTIFIER AND REGULATOR:

The programmer module receives a 50 VAC/60 Hz input, from two 25 Volt transformers which are located on

the INTELLEC® 4/MOD 40's chassis. The secondaries of these transformers are connected in series so that their outputs are additive, and the 50 Volt output thus obtained is routed to the programmer module through J3. A full-wave bridge consisting of diodes CR3-CR6 rectifies the 50 Volt input to produce a +80 Volt DC output. The module can function with either the 50 VAC or 80 VDC input.

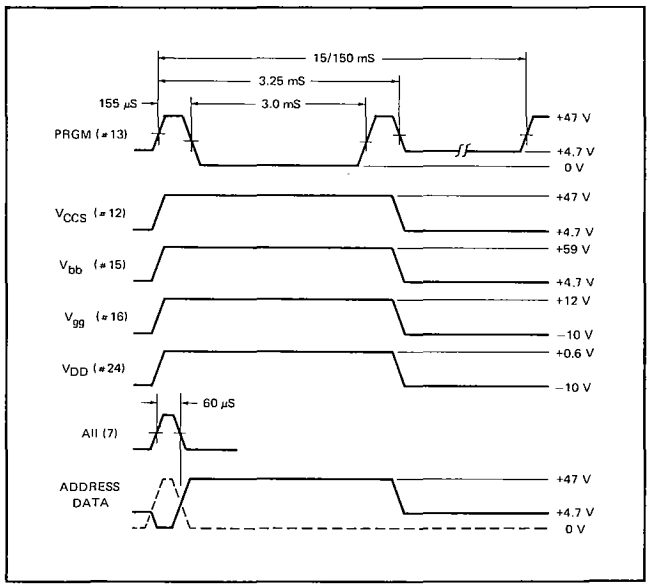


Figure 11-2. PROM Programmer Timing

The +80VDC output of the rectifier is applied to a series regulator, Q30, shown in the upper left hand corner of Figure 11-1. The output voltage at the emitter of Q30 depends upon the signal at its base. This level is determined in turn by a regulator loop which consists of an integrated voltage regulator (A17), Q33, and Q30 itself.

Figure 11-4 shows a simplified equivalent of the regulator loop. Components within the broken lines are part of the Signetics 550 monolithic voltage regulator.

The loop input is obtained from the regulator's output, through an adjustable resistive divider (R91 and R100). This level is applied to the non-inverting input of an operational amplifier which is incorporated into A17. The output of the amplifier drives a common-emitter stage, also contained within A17, and the inverted output at A17-11 is applied externally to the emitter of Q33. Q33's collector drives the base of the series regulator Q30, completing the negative feedback loop.

In a stabilized configuration such as this, the operational amplifier tends to maintain an output which results in zero error, where the error is the potential difference between the amplifier's inverting and non-inverting inputs.

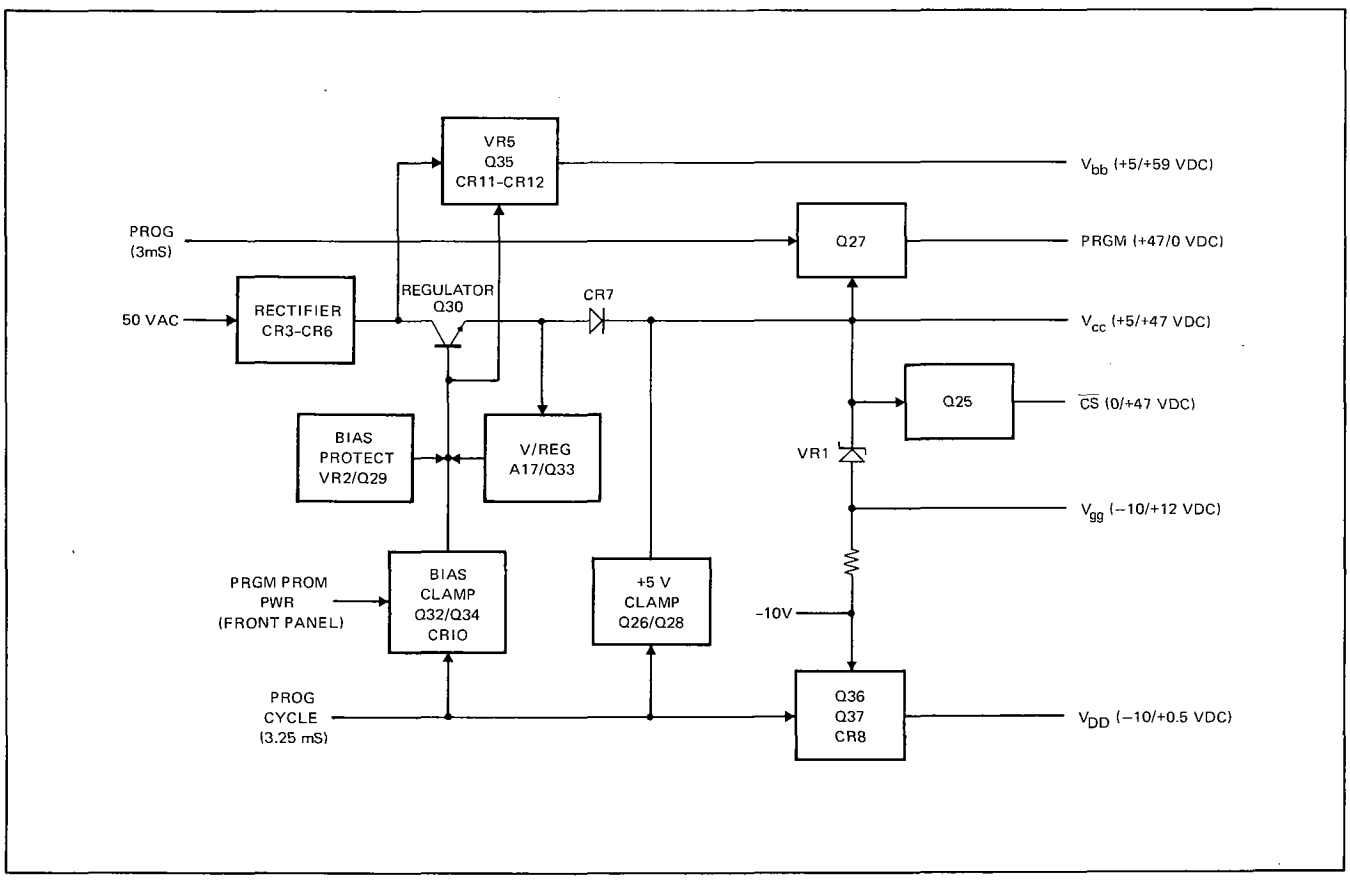


Figure 11-3. Power Supply Functional Block

Note that the inverting input is tied to the 550's internal reference (approximately 1.63 Volts). In order to obtain the desired output from the regulator, the resistive divider is adjusted for a zero error when the regulator's output is approximately +47.6 Volts.

Refer to the schematic for the PROM Programmer Module, Figure 11-1. Observe that the series regulator Q30 is protected against short-circuit overloads, by a bias protection circuit consisting of Q29 and the Zener diode VR2. Under ordinary operating conditions, Q29 will be off, and the reverse voltage applied to VR2 will be insufficient to cause this diode to conduct. In the event of a short-circuit, however, the voltage drop across Q30 will rise sharply. VR2 will begin conducting when the voltage across Q30 approaches 36 Volts, applying a forward bias to Q29. As a result, the voltage at Q29's collector will drop, clamping the base of Q30 to a relatively low level, and limiting the current output from the supply.

SCR1 is a crowbar switch, used to protect the PROM being programmed from an over-voltage condition in the supply. The normal voltage level on the V_{CCS} line (+47.6 Volts) is insufficient to cause conduction in Zener diode VR3. Should V_{CCS} rise above +56 Volts, however, the diode will conduct, forward biasing the gate of the SCR. SCR1 short-circuits the output of the rectifier, and the over-current condition blows fuse F2, interrupting AC power to the programmer module. Capacitor C16 provides an alternate gate current path, to prevent dv/dt triggering of the SCR when power is initially applied.

REGULATOR CONTROL:

Refer again to Figure 11-3, the power supply functional block. Note that the bias on Q30 is subject to the condition of a clamp. The clamp circuit consists of Q32, Q34, CR10, and associated components. These are used to switch the regulator output on and off, producing the pulses required for the programming of the PROM.

The base of Q34 is returned to the +80 Volt source, through pull-up resistor R92 (refer to Figure 11-1). Under static conditions, this transistor will conduct through CR10, clamping the base of Q30 to a low value. As a result of the low forward bias, Q30 displays a high impedance, and the output of the regulator will therefore drop to a very low value.

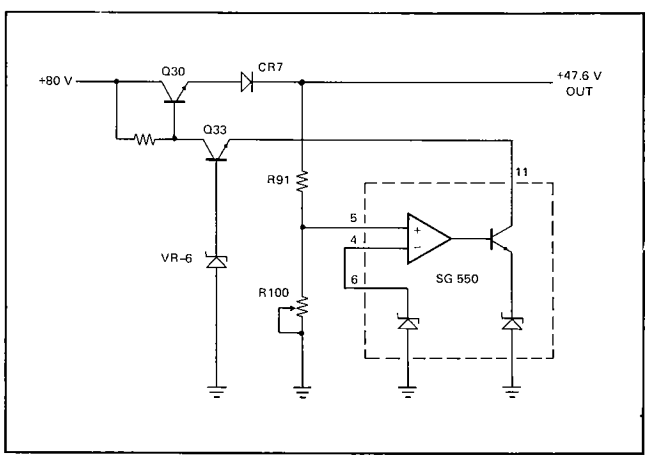


Figure 11-4. Voltage Regulator Loop: Simplified Schematic Equivalent

The PRGM PROM PWR switch is located on the Console and Display Panel of the INTELLEC® 4/MOD 40 system. Contacts of the PRGM PROM PWR switch ground the base of Q34 when that switch is turned on. This turns Q34 off, enabling the regulator.

The regulator's output remains clamped, however, by the conduction of Q32. This transistor is commanded by the control and timing section of the programmer module. The 3.25 millisecond output of the program cycle one-shot turns Q32 off at the start of the programming cycle. With both Q32 and Q34 disabled, the bias on Q30 rises to the stable level established by the characteristics of the regulator loop. The output of the regulator rises in consequence.

OUTPUT SWITCHES:

When no program cycle pulse is present, the regulator's output is at a low level. Diode CR7 is reverse biased, and the output voltage on the V_{CCS} line is determined by the clamp circuit consisting of Q26 and Q28. Under these conditions, Q26 operates in the reverse beta mode, holding V_{CCS} to approximately +4.7 Volts. When the program cycle begins, the control and timing section applies a negative-going 3.25 millisecond pulse to the base of Q28, turning that transistor off. Q26 now operates in a conventional manner, turned off by the low bias developed across R88. With the clamp removed, the V_{CCS} line is free to follow the rising output of the regulator section. CR7 conducts, and the V_{CCS} line rises to approximately +47 Volts.

Observe that the collectors of both the address drivers and the data drivers are returned to the V_{CCS} line, through their individual load resistors. Thus the normal 0 to 5 Volt logic excursion which prevails under statis conditions changes to a 0 to 47 Volt excursion during programming. This is in accord with the electrical requirements of the PROMs.

As V_{CCS} rises, Q25 goes into conduction, causing the level at the CS output to go from 0 Volts to +47 Volts.

Under static conditions, conduction through R89 holds the V_{gg} output to approximately -10 Volts. The 15 Volt drop across VR1 is not sufficient to induce an avalanche in the Zener. During programming, however, V_{CCS} rises to +47 Volts and the diode goes into conduction. As a result, V_{gg} rises to +11 Volts, approximately 36 Volts below the level on the V_{CCS} line.

The V_{DD} output is held to a static level of -10 Volts, by conduction through Q36. When programming begins a negative-going program cycle signal is applied to the emitter of Q37. The negative-going transition at its collector is coupled to the base of Q36, and Q36 turns off. CR8 conducts, causing V_{DD} to rise to about 0.6 Volts.

Under static conditions, the clamp transistor Q32 is conducting, and Q35 is turned off by the low voltage applied to its base through diode CR12. The V_{bb} output line is tied to V_{CCS} through R87, and the quiescent voltage level at this point is approximately +4.7 Volts. When the program cycle pulse turns Q32 off, VR5 conducts, and the voltage at the base of Q35 rises to the vicinity of +60 Volts. The emitter of Q35 follows this excursion and CR5 conducts, pulling V_{bb} up to a level of +59 Volts.

The PRGM line is connected to V_{CCS} through R78, and the static level at this output is approximately +4.7 Volts, at the beginning of the programming cycle, the PRGM output follows. One hundred fifty-five microseconds after the start of the cycle, the control and timing section sends a 3 millisecond program pulse to the base of Q27. This positive-going pulse turns the transistor on, and the voltage at its collector falls to approximately +9 Volts. Three milliseconds, later, the PRGM output returns to +47 Volts, where it remains until the end of the programming cycle.

UTILIZATION

This section describes the utilization of the imm4-76.

Installation

The PROM Programmer Module is designed for plug-in installation in the INTELLEC® 4/MOD 40 system. No special installation is necessary.

Plug the printed circuit board into any of the vacant connectors, J7-J19 on the system's mother board. A ribbon cable connects J1 at the top of the module to J1 on the mother board. A second ribbon cable connects J2 on the module to the programming socket on the front panel of the INTELLEC® 4/MOD 40 system.

An umbilical cable, permanently attached to the module, plugs into J34 on the system's mother board. This connection supplies AC power and enabling to the programmer module.

The toggle switch on the rear panel of the INTELLEC® 4/MOD 40 system connects the programmer's data outputs to the appropriate ROM input ports. Turn this switch OFF when you are not programming PROMs, to prevent data conflicts at input ports #2 and #3.

Refer to the INTELLEC® 4 system Operator's Manual for instructions on the programming of PROMs using the INTELLEC® 4/MOD 40 System Monitor.

POWER REQUIREMENTS

This module requires power at the following levels:

a) 50 VAC
b) +5 ± 5% VDC @ 1.0 A (max)
c) −10 ± 5% VDC @ 0.2 A (max)

The 50 VAC source shares a fuse with the -10 Volt supply in the INTELLEC® 4/MOD 40 system. This 0.5 Ampere fuse, F2, is located on the rear panel of the INTELLEC® 4/MOD 40 system.

Pin List

Connector pin allocations on the PROM Programmer Module are given in Tables 11-1, 11-2, and 11-3 and 11-4.

P1 Pin List

PIN	SIGNAL FUNCTION	PIN	SIGNAL FUNCTION
1		51	
2		52	
3	GROUND	53	
4	GROUND	54	
5		55	
6		56	
7		57	
8		58	
9		59	
10		60	
11		61	
12		62	
13		63	
14		64	
15		65	
16		66	
17		67	
18		68	
19		69	
20		70	
21		71	
22		72	
23		73	
24		74	
25		75	
26		76	
27		77	
28		78	
29		79	
30	R/$\overline{W}$ (1702)	80	
31		81	
32	R/$\overline{W}$ (1702A)	82	
33		83	
34		84	
35		85	
36		86	
37		87	
38		88	
39		89	
40		90	
41		91	
42		92	
43	-10 VDC	93	
44	-10 VDC	94	
45		95	
46		96	
47		97	
48		98	
49		99	+5 VDC
50		100	+5 VDC

Table 11-1.

J1 Pin List

PIN	SIGNAL FUNCTION
1	DATA 0 IN
2	ADDRESS 0 IN
3	DATA 1 IN
4	ADDRESS 1 IN
5	DATA 2 IN
6	ADDRESS 2 IN
7	DATA 3 IN
8	ADDRESS 3 IN
9	DATA 4 IN
10	ADDRESS 4 IN
11	DATA 5 IN
12	ADDRESS 5 IN
13	DATA 6 IN
14	ADDRESS 6 IN
15	DATA 7 IN
16	ADDRESS 7 IN
17	TEST DATA OUT 0
18	
19	TEST DATA OUT 1
20	
21	TEST DATA OUT 2
22	
23	TEST DATA OUT 3
24	
25	TEST DATA OUT 4
26	
27	TEST DATA OUT 5
28	
29	TEST DATA OUT 6
30	
31	TEST DATA OUT 7
32	
33	+5 VDC
34	+5 VDC
35	+5 VDC
36	+5 VDC
37	+5 VDC
38	+5 VDC
39	+5 VDC
40	+5 VDC
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	

Table 11-2.

J2 Pin List

PIN	SIGNAL FUNCTION
1	PROM DATA OUT 0
2	PROM ADDRESS OUT 0
3	PROM DATA OUT 1
4	PROM ADDRESS OUT 1
5	PROM DATA OUT 2
6	PROM ADDRESS OUT 2
7	PROM DATA OUT 3
8	PROM ADDRESS OUT 3
9	PROM DATA OUT 4
10	PROM ADDRESS OUT 4
11	PROM DATA OUT 5
12	PROM ADDRESS OUT 5
13	PROM DATA OUT 6
14	PROM ADDRESS OUT 6
15	PROM DATA OUT 7
16	PROM ADDRESS OUT 7
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	

Table 11-3.

J3 Pin List

PIN	SIGNAL FUNCTION
1	50 VAC (01)
2	
3	50 VAC (02)
4	+80 VDC OUT
5	$\overline{\text{PROGRAM PROM POWER}}$
6	GROUND
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	

Table 11-4.

CHAPTER 12 INSTALLATION AND INTERFACING

This chapter explains how to install the INTELLEC® 4/MOD 40 System. Detailed operating instructions, however, are outside the scope of this manual. Refer to the INTELLEC®4 System Operator's Manual for procedures.

OPERATING PRECAUTIONS

—WARNING—

When installed as directed, this instrument contains 115 Volts AC at a line frequency of 60 Hz. Under adverse conditions, contact with THIS POTENTIAL CAN KILL YOU INSTANTLY. Installation and servicing should be performed only by trained and qualified personnel, using all appropriate equipment and safety precautions.

Observe the following precautions, in the installation and use of the INTELLEC®4/MOD 40 system.

NEVER operate the system for any extended period with the cover removed. The normal air flow to the power supplies will be diverted, resulting in overheating and possible permanent damage. Brief periods of cover-off operation are permissible, for test and troubleshooting only.

The +5 VDC power supply furnished with the INTELLEC®4/MOD 40 system is rated for a maximum load of 12 Amperes. It is possible inadvertently to exceed this limit, in certain optional configurations. If you plan to use one or more optional modules, check first to be sure that the +5 VDC supply will not suffer an overload. The current drawn by each of the system components, standard and optional, is listed in the UTILIZATION section of the chapter pertaining to that module.

INSTALLATION OF THE INTELLEC®4/MOD 40 SYSTEM

The INTELLEC®4/MOD 40 system is shipped ready to use. The unit is bench-mounted and requires no special preparation.

Installation consists of:

a) power connections
b) interface connections

Power Requirements

The INTELLEC®4/MOD 40 system requires primary power of 110-120 VAC/50-60 Hz. The instrument is equipped with a standard three-prong NEMA power cord, which automatically grounds the chassis when mated with a grounded three-wire receptacle. This ground may be lifted, if special interference conditions make it necessary, at P22 on the rear panel of the instrument.

Power consumption varies with the number of optional modules, from 100 Watts for the standard system to 200 Watts for the maximum. There are two fuses on the rear panel, F1 and F2:

	AC POWER INPUT	
FUSE	115 VAC	230 VAC
F1	3 Amp.	1.5 Amp.
F2	1 Amp.	0.5 Amp.

In addition, the PROM Programmer power (+80 VDC) is fused at 0.5 Amperes (fast-blow).

Interface Requirements

In order to interface the INTELLEC®4/MOD 40 system to peripheral devices, the user must know the electrical characteristics of the I/O ports. He must also know the correct method of cabling the interface, and the techniques for minimizing or eliminating signal cross-talk. These topics are discussed below.

ELECTRICAL CHARACTERISTICS OF THE I/O PORTS

The standard INTELLEC®4/MOD 40 system is equipped with three input ports and eight output ports. An expanded system however, may contain as many as fifteen input ports and 48 output ports.

Each port consists of four unweighted TTL lines. These lines operate in the 0 Volt to +5 Volt positive zone, as is customary in TTL applications. Data and signal COMMONs are identical to power GROUND. All are at chassis potential (0 Volts).

TTL standards define a LOW logic level as one less than +0.8 Volts. A HIGH is defined as a level greater than +2.0 Volts. The INTELLEC® System's inputs and outputs all conform to this definition.

Output ports associated with the INTELLEC® 4/MOD 40 system may be ROM outputs or RAM outputs. The two kinds of ports have slightly different electrical characteristics.

Each RAM output line is driven by one section of a TI 7417 Hex Buffer. These are high-power drivers, with open-collector outputs. Their electrical characteristics depend ultimately upon the nature of the load. The recommended installation calls for a 1K "pull-up" resistor at the receiver end of each data line. A 200-ohm filter resistance will also be inserted in series with the line, as shown in Figure 12-3. Under these circumstances, the DC characteristics of each line (at the filter's output) will be approximately as shown in Table 12-1. Figures given take into account the worst-case variations in the resistance of 10% components, and the normal variations in supply voltage (±5%).

RAM Port Output Characteristics

PARAMETER*	MIN	MAX	UNIT	TEST CONDITIONS
V_{0L}		0.7	V	I_{0L} = 35 mA
V_{0H}	2.1		V	I_{0H} = -2 mA
I_{0L}		35	mA	
I_{0H}	-1.8		mA	V_{0H} = 2.4 V

*V_{0L}: Low-level output voltage
V_{0H}: High-level output voltage
I_{0L}: Low-level output current
I_{0H}: High-level output current

Table 12-1.

Each of the system's input lines terminates internally at the data input to an Intel® 8214 Dual 4-Line-To-1-Line Data Multiplexer. The electrical characteristics of these inputs are shown in Table 12-2.

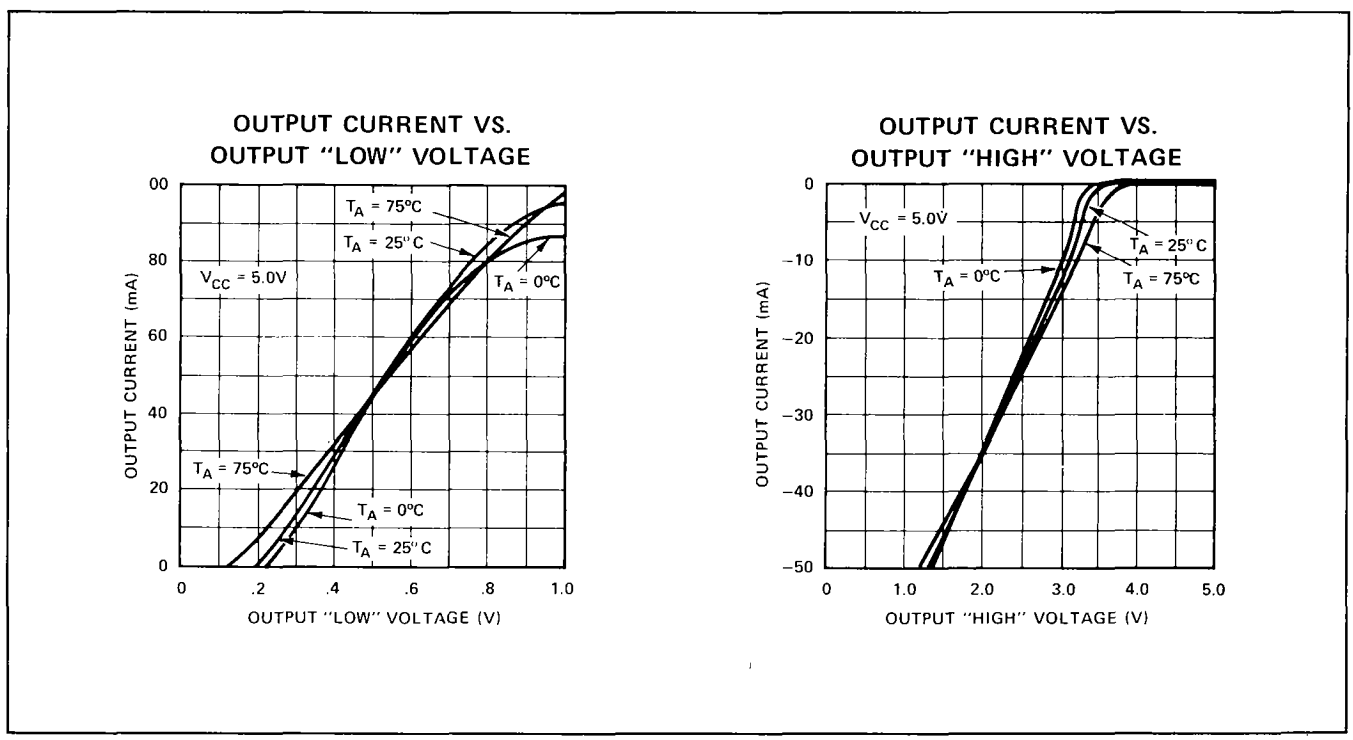


Figure 12-1. ROM Port Output Characteristics

ROM Port Input Characteristics

PARAM-ETER	MIN	MAX	UNIT	TEST CONDITIONS
V_{IL}		0.8	V	
V_{IH}	2		V	
I_{IL}		-1.6	mA	V_I = 0.4 V
I_{IJ}		40	μA	V_I = 2.4 V

Table 12-2.

One section of a TI 7404 Hex Inverter is ideal for driving each of these low-power input lines.

Note, however, that special requirements apply if the imm6-76 PROM Programmer Module is used. The data outputs of the programmer are connected internally to input ports #2 and #3 of the INTELLEC®4/MOD 40 System. The parallel load presented by these lines does not appreciably alter the DC characteristics of the ports, but it does contribute significantly to their shunt capacitance. To maintain adequate risetimes with the programmer module installed, a high-power driver such as the type 7417 must be used. The simplest solution in most cases is simply to disconnect the imm6-76 when you are not programming PROMs, by detaching the ribbon cable at J1 on the module.

Observe also that the toggle switch on the rear panel disables the data outputs of the programmer module (without physically disconnecting them). That switch must be ON when you are programming PROMs, and OFF at all other times, to prevent data conflicts at input ports #2 and #3.

I/O CABLING

All input and output lines terminate at 37-pin connectors on the rear panel of the INTELLEC®4/MOD 40 System. The number of such connectors, and their arrangement, depends upon the number of input and output ports optionally provided. Figure 12-2 shows how to identify these connectors. The appropriate mating plug is the A-MP 205210-1.

Table 12-3 gives pin allocations on the input connector used with the basic system. Table 12-4 gives pin allocations on the output connector. If optional I/O modules are included in your system, additional rear panel connectors will be provided. Allocations on these connectors will parallel those shown in the wiring tables for the basic system, but the port numbering will depend upon the particular modules used and on the module wiring options selected by the end user. Since no general guidelines can be given, these will have to be determined individually. Consult the section of this manual that deals with the module in question, for a description of the options available.

Use a twisted pair cable for the connection of each input and each output line. Distributive capacitance limits the effective length of these cables to about 12 feet. Scotchflex Woven Cable (type #3321) is recommended. This cable contains 37 individual pairs and has a 110-ohm characteristic. Any pair with a similar characteristic is acceptable.

Figure 12-3 shows the proper method of connecting the output lines. Figure 12-4 shows how to connect inputs. Note especially the use of "pull-up" resistors and the filters at the receiving end of the output lines.

The Central Processor Module provides 4 input ports and 8 output ports. Most of these ports have dedicated uses, however, in the INTELLEC 4/MOD 40 System. Table 12-5 lists the dedicated use of each I/O port on the Central Processor Module.

SIGNAL CROSS-TALK EFFECTS

With a large number of signal lines close together in parallel, some cross-talk is inevitable. The use of twisted pair cables minimizes the cross-talk components attributable to line discontinuities and to inductive coupling between adjacent lines.

The use of unbalanced lines, however, makes it impossible to cancel the effect completely. Adjacent conductors are coupled capacitively, and the impedance of the coupling varies inversely with the length of the lines and with the risetime of the signal involved.

Worst-case conditions exist when three of the four lines associated with a given port change level, simultaneously and in the same direction. Experiments show that the spurious signal generated on the fourth line may reach peak levels of 1.6 Volts under these circumstances, quite sufficient to cause data errors.

The high frequency of the spurious component makes it possible to suppress interference effectively, by using low-pass filters at the receiving end of the output lines. This is the purpose of the 200-ohm/0.001 μF combination shown in Figure 12-3.

Unfortunately, the same technique is not applicable to the input lines. Input signals must still pass through 25 inches of ribbon cable, between the rear panel connector and their destination on the internal modules. Thus an external filter would not be completely effective in cancelling cross-talk.

Further experiments, however, show that the cross-talk effect is relatively short lived. The spurious signal has virtually subsided within 150 nanoseconds of the transition that originally produced it. Since the machine cycle of the INTELLEC 4/MOD 40 System is appreciably longer than this (10.8 microseconds), any input sequence that requires more than one instruction automatically discriminates against this kind of interference.

Input Connector

PIN	SIGNAL FUNCTION
1	GROUND
2	
3	
4	$\overline{\text{CPU RESET}}$ (USER RESET OUT)
5	
6	
7	
8	
9	
10	ROM INPUT PORT 0/BIT 0
11	ROM INPUT PORT 0/BIT 1
12	STOP ACKNOWLEDGE
13	INTERRUPT ACKNOWLEDGE
14	ROM INPUT PORT 2/BIT 0
15	ROM INPUT PORT 2/BIT 1
16	ROM INPUT PORT 3/BIT 0
17	ROM INPUT PORT 3/BIT 1
18	GROUND
19	GROUND
20	GROUND
21	
22	$\overline{\text{USER RESET IN}}$
23	
24	
25	
26	
27	
28	
29	ROM INPUT PORT 0/BIT 2
30	ROM INPUT PORT 0/BIT 3
31	$\overline{\text{STOP}}$
32	$\overline{\text{INTERRUPT}}$
33	ROM INPUT PORT 2/BIT 2
34	ROM INPUT PORT 2/BIT 3
35	ROM INPUT PORT 3/BIT 2
36	ROM INPUT PORT 3/BIT 3
37	GROUND
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	

Table 12-3.

Output Connector

PIN	SIGNAL FUNCTION	
1	GROUND	
2	ROM OUTPUT PORT 0/BIT 0	
3	ROM OUTPUT PORT 0/BIT 1	
4	ROM OUTPUT PORT 1/BIT 0	
5	ROM OUTPUT PORT 1/BIT 1	
6	ROM OUTPUT PORT 2/BIT 0	
7	ROM OUTPUT PORT 2/BIT 1	
8	ROM OUTPUT PORT 3/BIT 0	
9	ROM OUTPUT PORT 3/BIT 1	
10	$\overline{\text{RAM OUTPUT PORT 0/BIT 0}}$	RAM BANK #0
11	$\overline{\text{RAM OUTPUT PORT 0/BIT 1}}$	
12	$\overline{\text{RAM OUTPUT PORT 1/BIT 0}}$	
13	$\overline{\text{RAM OUTPUT PORT 1/BIT 1}}$	
14	$\overline{\text{RAM OUTPUT PORT 2/BIT 0}}$	
15	$\overline{\text{RAM OUTPUT PORT 2/BIT 1}}$	
16	$\overline{\text{RAM OUTPUT PORT 3/BIT 0}}$	
17	$\overline{\text{RAM OUTPUT PORT 3/BIT 1}}$	
18	GROUND	
19	GROUND	
20	GROUND	
21	ROM OUTPUT PORT 0/BIT 2	
22	ROM OUTPUT PORT 0/BIT 3	
23	ROM OUTPUT PORT 1/BIT 2	
24	ROM OUTPUT PORT 1/BIT 3	
25	ROM OUTPUT PORT 2/BIT 2	
26	ROM OUTPUT PORT 2/BIT 3	
27	ROM OUTPUT PORT 3/BIT 2	
28	ROM OUTPUT PORT 3/BIT 3	
29	$\overline{\text{RAM OUTPUT PORT 0/BIT 2}}$	RAM BANK #0
30	$\overline{\text{RAM OUTPUT PORT 0/BIT 3}}$	
31	$\overline{\text{RAM OUTPUT PORT 1/BIT 2}}$	
32	$\overline{\text{RAM OUTPUT PORT 1/BIT 3}}$	
33	$\overline{\text{RAM OUTPUT PORT 2/BIT 2}}$	
34	$\overline{\text{RAM OUTPUT PORT 2/BIT 3}}$	
35	$\overline{\text{RAM OUTPUT PORT 3/BIT 2}}$	
36	$\overline{\text{RAM OUTPUT PORT 3/BIT 3}}$	
37	GROUND	

Table 12-4.

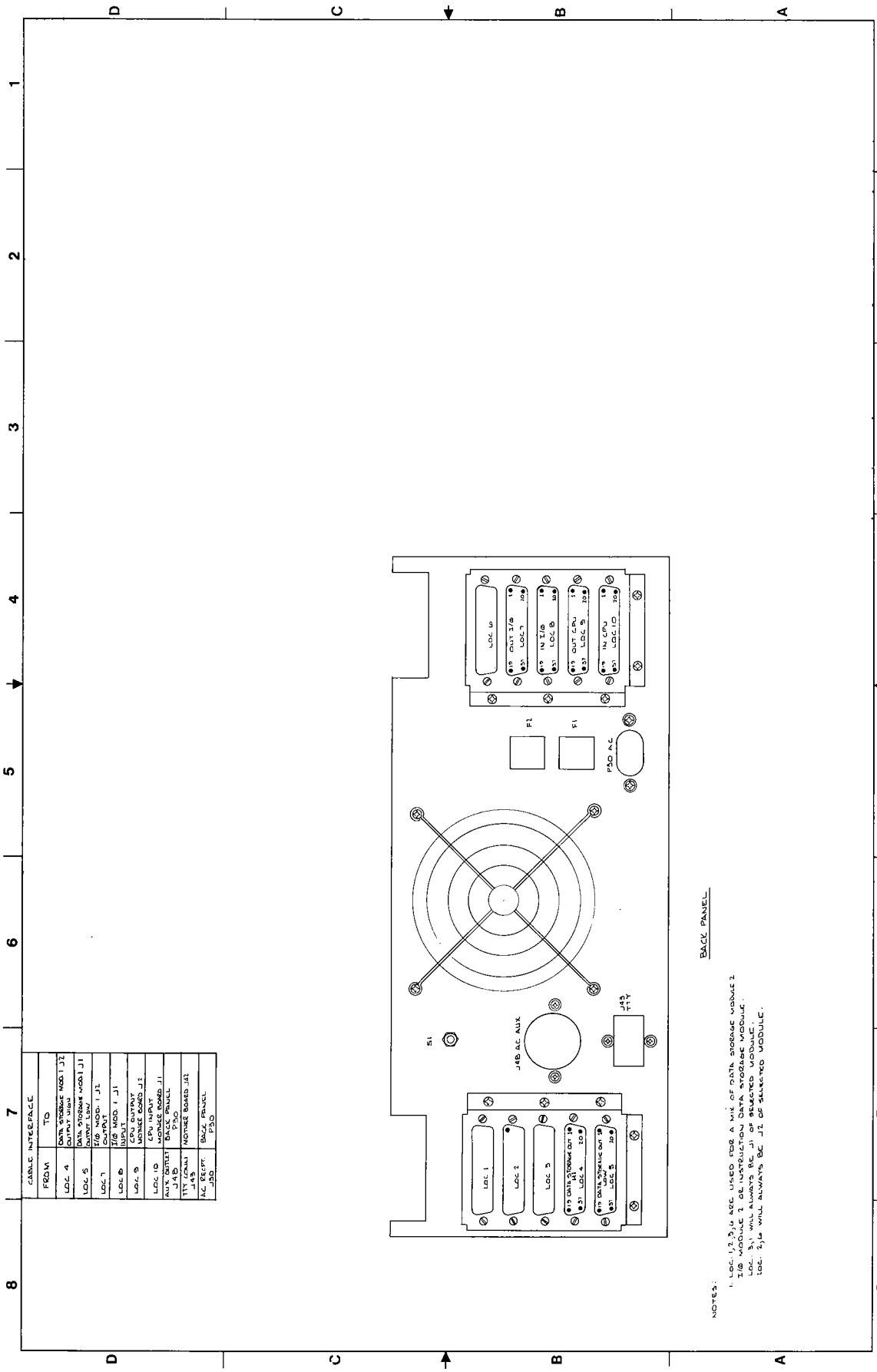


Figure 12-2. INTELLEC® 4: Rear Panel Layout

I/O Port Usage

I/O PORT	USAGE
INPUT ROM 0	Teletype Keyboard (Bit 0 only)
INPUT ROM 1	*Not available for use in INTELLEC® 4/MOD 40
INPUT ROM 2 INPUT ROM 3	PROM Program Check
OUTPUT RAM 0	Teletype Printer (Bit 0 only)
OUTPUT RAM 1	Teletype Reader (Bit 0) PRG R/$\overline{W}$ A signal for programming 1702A PROM's (Bit 1) PRG R/$\overline{W}$ signal for programming 1702 PROM's (Bit 2)
OUTPUT RAM 2	Available for use
OUTPUT RAM 3	Available for use
OUTPUT ROM 0 OUTPUT ROM 1	PROM address
OUTPUT ROM 2 OUTPUT ROM 3	PROM data

*Due to lack of connector pins

Table 12-5.

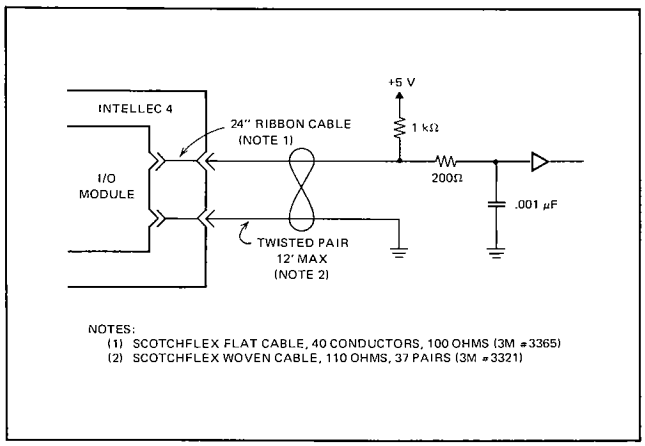


Figure 12-3. INTELLEC® 4: Output Interface

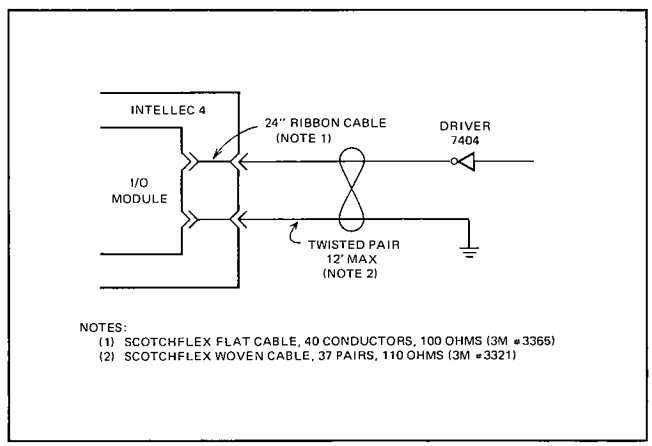


Figure 12-4. INTELLEC® 4: Input Interface

In a typical input operation, the INTELLEC® 4/MOD 40 system will first sample a test bit (flag) on the interface, to determine whether a station has data to transmit. Several cycles will elapse between the input of the control bit and the reading of the data presented to the port. Under these conditions, cross-talk is not likely to be a problem.

But if the input line is used for a control function, like the flag bit in the example just cited, the presence of spurious signals may still result in errors. Two solutions are possible.

The INTELLEC® system can be programmed to sample a control bit twice and test for continuity. A control signal may be considered valid only if it remains constant for the entire period between two successive tests.

The second solution is to use synchronized input. This method also takes advantage of the inverse time dependence of the interference effect. In this case, the INTELLEC® system transmits a control signal to the interface when it is ready to receive data. When the interface receives this signal, it responds by loading the input lines. More than 10 microseconds must elapse before the system can actually read this this port, ample time for the extraneous pulses to subside.

OTHER CONTROLS

As shown in Table 12-3, several additional control functions are available to the user at the rear panel connectors. These are:

a) $\overline{\text{USER}}$ RESET IN

b) $\overline{\text{USER}}$ RESET OUT
c) $\overline{\text{STOP}}$/STOP ACKNOWLEDGE
d) $\overline{\text{INTERRUPT}}$/INTERRUPT ACKNOWLEDGE

All are negative-true, positive-zone TTL functions. $\overline{\text{USER RESET IN}}$, $\overline{\text{TEST}}$, $\overline{\text{STOP}}$, and $\overline{\text{INTERRUPT}}$ inputs may be driven by a remote contact to GROUND. A 7404 driver, or a saturated NPN will also serve.

The $\overline{\text{USER RESET IN}}$ permits the operator to initiate a RESET of the system from a remote location. This function is duplicated on the console of the INTELLEC® 4/MOD 40 system.

The $\overline{\text{USER RESET OUT}}$ is available, where external circuitry must be cleared simultaneously with the internal RESET of the system. The output is a negative-going 500 microsecond pulse, capable of driving a unit low-power TTL load.

The $\overline{\text{STOP}}$ input enables an operator to halt the processor from a remote location. While the processor is halted, the STOP ACKNOWLEDGE status line will be high. The STOP input may also be used to re-start a processor which has been halted by a programmed machine instruction (HLT). To do so, the STOP line must be clamped low momentarily, then permitted to rise again to its normally high quiescent level. For a fuller description of how the processor responds to this input, refer to Chapter 3 of this manual.

The $\overline{\text{INTERRUPT}}$ line provides for single-level interrupt of the processor. Interruption causes an automatic subroutine jump to memory location 003, where an appropriate routine for interrupt servicing must have been stored. The INTERRUPT ACKNOWLEDGE output will go high at the end of the machine cycle in which the $\overline{\text{INTERRUPT}}$ occurs. It will remain in this state until cleared by a "branch-back and send register control" instruction is executed (BBS), indicating by implication that interrupt servicing has been completed. During an acknowledged interruption, the processor is prevented internally from responding to any activity whatsoever at its $\overline{\text{INTERRUPT}}$ input. Chapter 2 of this manual describes in detail how the 4040 CPU processes interrupt requests. The reader should also refer to the 4040 Programming Manual for a complete description of interrupt programming.

TELETYPE

While it is entirely possible to develop, load, and test a program using only the front panel controls of the INTELLEC® 4/MOD 40 system, most users will find it impractical to do so. An operator using the panel as console is denied access to all the convenience features built into the system software, including the automatic assembly and loading of programs and the de-bugging and editing aids available with the System Monitor. Manual procedures make program development an arduous and time-consuming task.

In order to take full advantage of the development software, the user must provide a teletype for console operations. The Model ASR 33 is recommended, because of its paper tape handling facilities. Perforated tape is a particularly convenient medium for the kinds of transactions involved in micro-program development. Non-volatile storage of programs, assembly, and loading are facilitated by the teletype/reader/punch combination of the ASR 33.

The system's Central Processor Module contains a built-in electrical interface which enables the input and output of data via the teletype. Three interface circuits are provided: the receiver circuit (TTY IN), the transmitter circuit (TTY PRINTER), and the reader control circuit. Only minor modifications to the teletype set are required, and these are explained fully on page 130.

The Model ASR 33 Teletype Set

In order to understand how the interface operates, one must know something about the teletype itself. A Model ASR 33 teletype set consists of:

a) a keyboard unit
b) a printer unit
c) a tape punch unit
d) a tape reader unit
e) an electrical service unit

The keyboard unit is essentially a transmitter, and the printer unit is essentially a receiver. The keyboard originates messages, by causing a time-dependent series of current interruptions, in a DC loop connecting the keyboard and one or more keyboards, printing a message in response to the pattern of interruptions in the loop current. An interruption originating at any keyboard within a loop causes a response at every printer within that loop.

The teletype set's printer contains a current option provision, enabling it to function on a nominal loop current of either 60 mA or 20 mA. The user makes the selection of loop current at the time the set is installed, by shifting the position of a wire jumper in the set's electrical service unit. Since the factory ships the ASR 33 wired for 60 mA operation, users of the INTELLEC® 4/MOD 40 system must convert the machine to 20 mA operation, before connecting the terminal into their system. Page130 describes the installation procedure.

There are two ways of connecting a pair of teletype sets, half duplex (HDX) and full duplex (FDX). In the half duplex configuration, a single loop is used. This loop passes through both the keyboard unit and the printer unit of each teletype set. A message originating at either keyboard therefore causes both printers to respond. In a half duplex configuration, only one terminal can transmit at any given time. Chaos would result if two or more keyboards attempted to transmit messages simultaneously.

The full duplex hookup uses two current loops. The first loop links the keyboard of one terminal with the printer of the second. The other loop connects the second terminal's keyboard with the printer of the first set. In this configuration, the keyboard of each set is independent of its associated printer. Thus the terminal can receive a message

on its printer while simultaneously using its keyboard for transmission.

Although it is possible to use the half duplex mode for data processing applications, this is seldom done in practice. The principal advantage in using full duplex is that the receiving terminal can "echo" the transmitter; that is, transmit the received message back to the originating terminal after a short processing delay. The delay involved is generally so brief that the terminal operator is never aware of the fact that he is working in full duplex. His printer unit will apparently respond to the keyboard just as rapidly as it would in the half duplex mode, but any transmission error or hardware malfunction will promptly reveal itself in the form of a garbled printout. This affords a highly desirable check on the operation of the entire system. The teletype interface on the Central Processor Module is of this "echoplex" type.

The tape punch unit is associated with the printer unit of the teletype set. When enabled, it punches a paper tape record of all messages received by the printer.

The tape reader unit is associated with the keyboard unit of the teletype set. The reader accepts a 1-inch paper tape, punched in ASCII code, and produces a corresponding series of current pulses in the loop it shares with the keyboard. A sprocket wheel on the reader advances the tape automatically, at the maximum TTY rate of 10 characters per second.

The teletype set's tape reader unit is often used as a convenient way to feed large volumes of data into a processing machine. The data may be prepared off-line, that is with the teletype disconnected from the processor, by using the punch unit to prepare a paper tape. A check of the printer's output enables the operator to correct any errors, before feeding the tape into the reader. Once the tape is prepared, the operator simply inserts the leader into the tape reader and pushes the lever that starts the unit.

The reader operates automatically once it is started, at the maximum transfer rate. Here, however, a hitch may develop. Segments of data on the tape may be interspersed with periods when processing action is required. If the processor were to execute these functions while the tape continued to run, it is possible that several characters might be lost irretrievably by the time the processor returned to the input/output routine. To prevent this, the processor must have some way of controlling the tape reader: of stopping the reader intermittently, and of starting it up again. That is the purpose of the reader control section of the module's teletype interface.

The tape reader advances when 115 VAC is applied to its distributor trip magnet. In order to provide the processor with some means of controlling the reader, a relay must be inserted in series with this circuit. The processor then controls the reader by controlling the power applied to the relay's pull-in coil. Page 131 contains a full description of the modifications required.

The electrical service unit is essentially a junction box, interfacing the other units to the external transmit-receive loops.

Figure 12-5 shows the functional interrelationship of the five major units in the teletype set. Observe that the mode of operation is established by shifting jumpers on a terminal block within the electrical service unit. Connecting a jumper between terminals C and D places the printer/punch in series with the keyboard/reader, enabling operation in the half duplex mode. Terminals A and F then become the points of connection for the external section of the current loop.

By deleting jumper C-D, and substituting jumpers B-C and D-E, we obtain full duplex operation. The printer and punch unit become part of the receive loop, established through terminals A and B. The transmit loop, containing the keyboard and reader units, connects to terminals E and F. (Terminals A-F are illustrative examples only, and do not correspond to actual terminals within the electrical service unit). Page 132 describes installation procedures, including the wiring required for HDX or FDX operation.

Modification of the Teletype Set

Minor modifications to the ASR 33 will be necessary, before the set can be used with the INTELLEC® 4/MOD 40 system. These include:

- Conversion from factory-wired operation, at 60 mA, to a 20 mA loop current.
- Conversion from factory-wired operation, in the half duplex mode, to full duplex.
- Provide for external control of the tape reader drive.

20 MILLIAMPERE LOOP CONVERSION

Proceed as follows, to prepare the ASR 33 for operation in a 20 mA current loop:

1) Change the current source resistor in the electrical service unit of the teletype, from 750-Ohms to 1450-Ohms. The current source resistor is a tapped resistance, and conversion is accomplished by shifting the position of a single wire lead. "Kwik-Connect" push-ons are used, and no soldering is required. Figure 12-6 shows the location of the resistor. The physical arrangement of the taps is diagrammed in Figure 12-7.
2) Change the position of a single wire on the barrier terminal strip at the rear of the electrical service unit. A screwdriver will be necessary. Figure 12-8 shows the location of the terminal strip, and Figure 12-7 shows the electrical change required. The violet wire attached to terminal #8 should be moved to terminal #9.

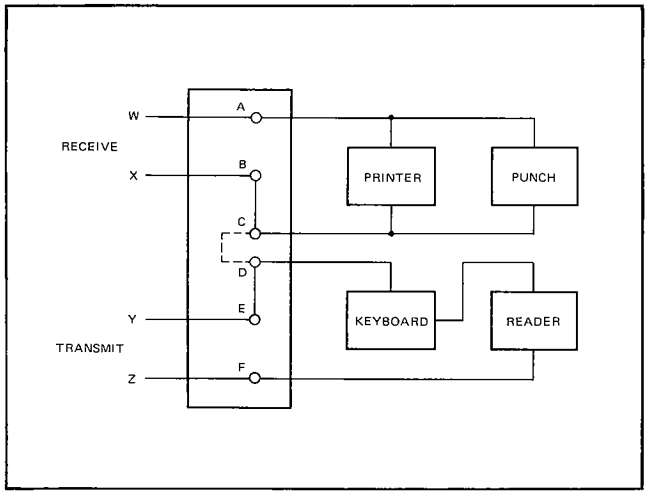


Figure 12-5. TTY: Half Duplex/Full Duplex

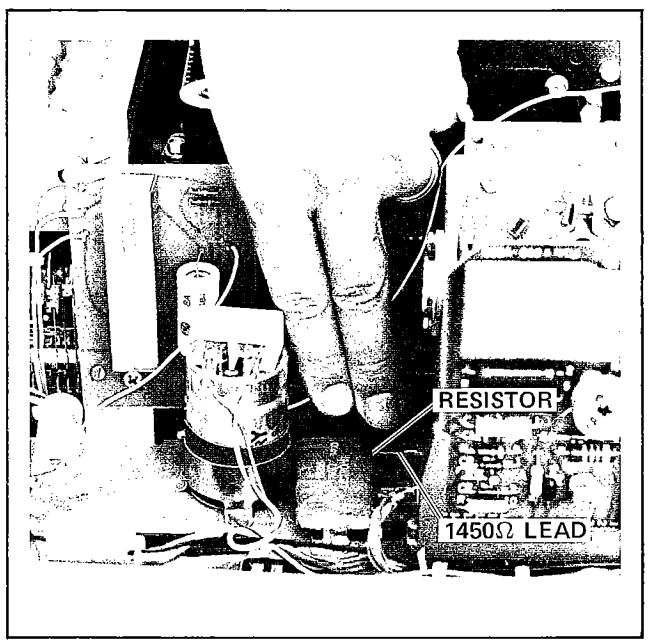


Figure 12-6. Current Source Resistor

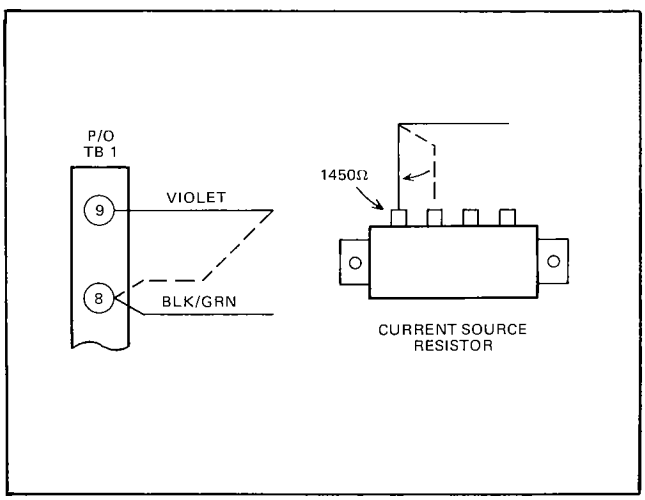


Figure 12-7. TTY: 20 mA Wiring

CONVERSION TO FULL DUPLEX OPERATION

Conversion to full duplex is accomplished simply, by shifting the position of two wires on the barrier terminal strip at the rear of the electrical service unit. A screwdriver will be required. Figure 12-8 shows the location of the terminal strip, and Figure 12-9 shows the electrical change to be accomplished. Broken lines in the diagram indicate the half duplex hookup. Solid lines indicate the wiring required for full duplex operation.

1) Move the blue/white lead from terminal #4 to terminal #5 on the strip.

2) Move the brown/yellow lead from terminal #3 to terminal #5.

READER DRIVE MODIFICATIONS

In order to provide control of the tape reader drive, you must install a relay circuit in the electrical service unit of the teletype set.

Construction of the circuit is simple. In addition to wire and a small piece of vector board, you will need the following parts:

miniature relay (Potter-Brumfield #JR-1005)	1 each
carbon comp resistor, 470-Ohm, ½ Watt	1 each
capacitor, 0.1 μF, 200 wvdc	1 each

These parts will be wired as shown in Figure 12-10.

Figure 12-11 shows how to locate and mount the auxiliary circuit in the teletype's electrical service unit, on the vertical metal tab which is just forward of the large filter capacitor. Drill the circuit board in two places. Then drill and tap the tab. Mount the auxiliary board using two machine screws.

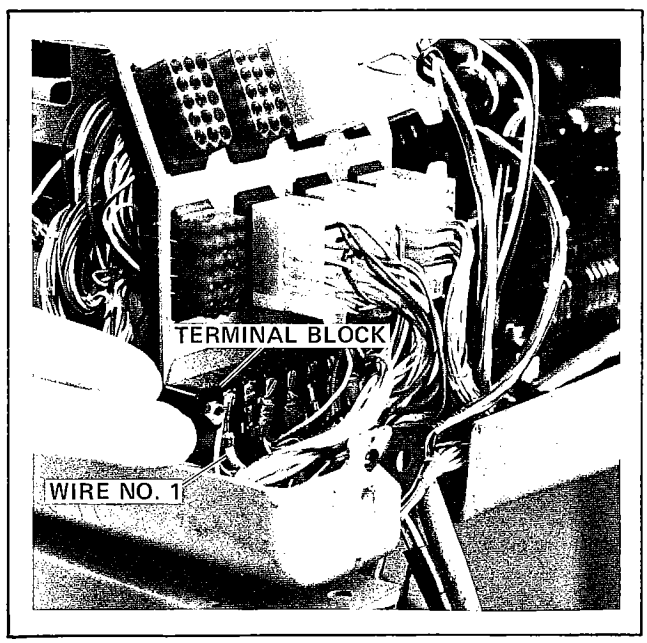


Figure 12-8. Terminal Block Location

The auxiliary circuit board is wired into the electrical service unit in three steps:

1) Remove the brown wire from the molex plug (see Figure 12-12 for location of the wire). Splice two leads into the disconnected brown wire.

2) Connect one of these leads (labeled wire "A" in Figure 12-10) to one side of the Potter & Brumfield relay contact pair.

3) Connect the remaining lead (labeled "LOCAL" in Figure 12-10) to terminal L2 of the LOCAL/LINE selector switch. Connect the other contact of the Potter & Brumfield relay to terminal L1 of the LOCAL/LINE selector switch. Figure 12-13 shows how to identify the switch terminals.

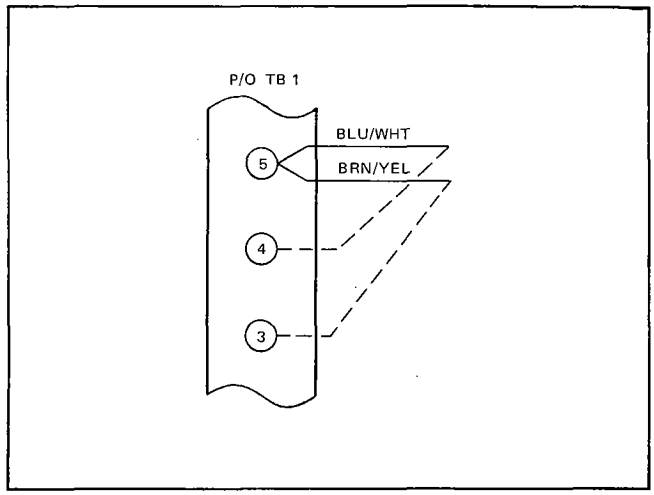


Figure 12-9. TTY: Full Duplex Wiring

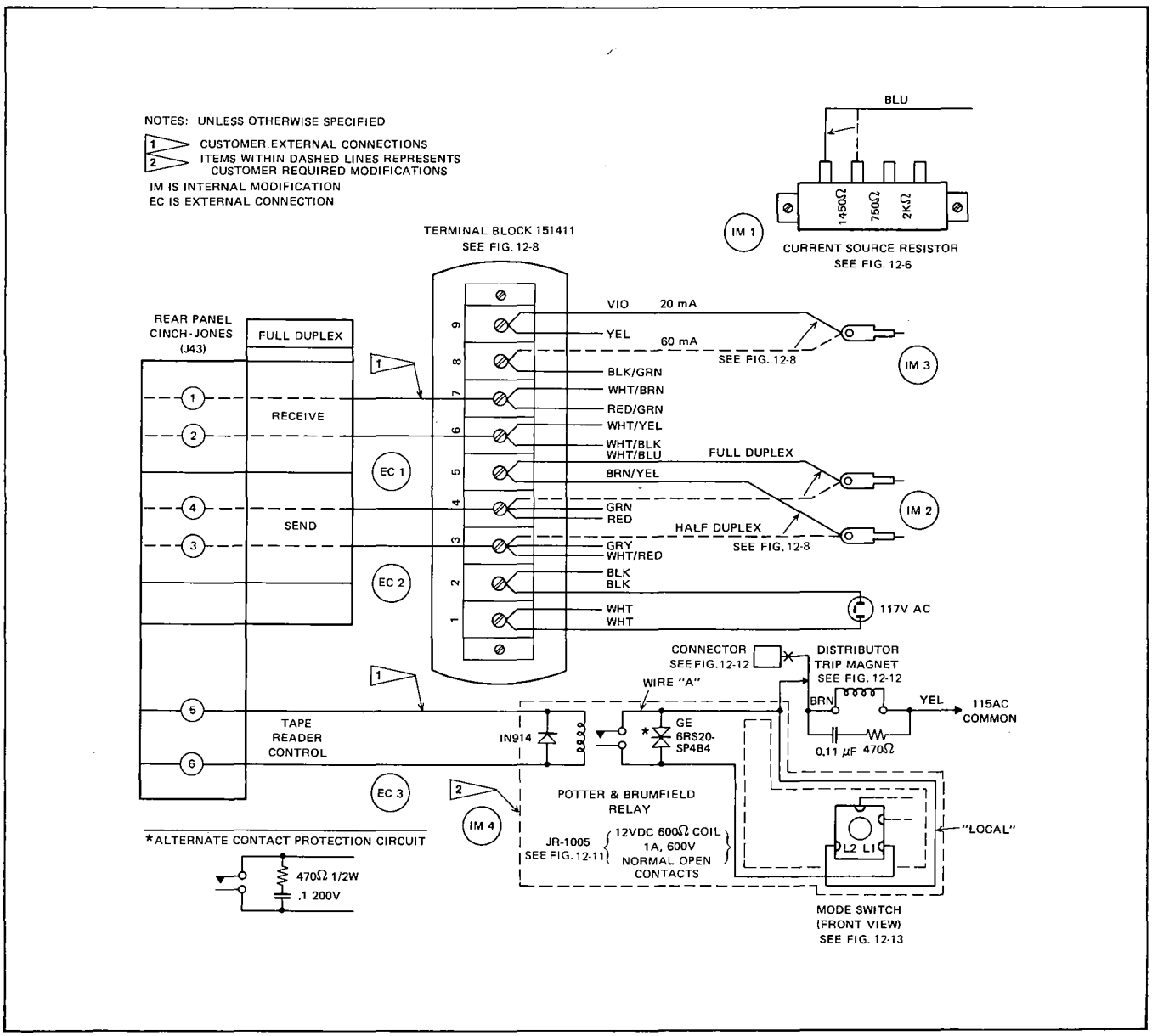


Figure 12-10. TTY Modifications

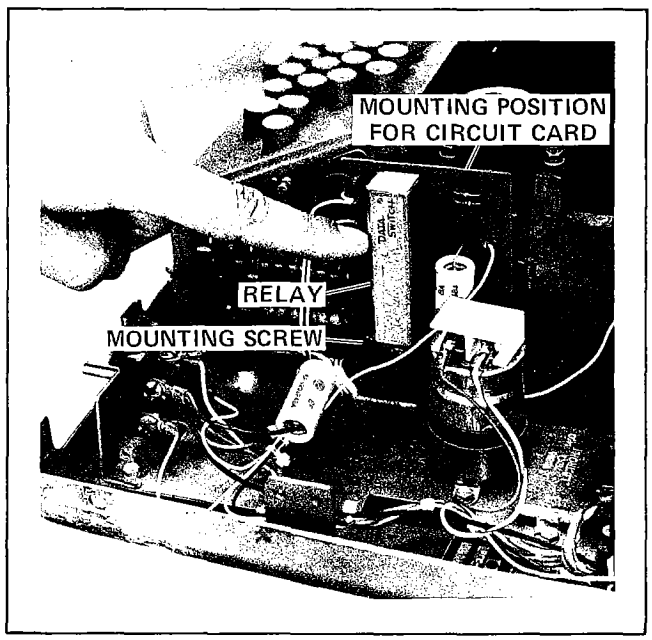


Figure 12-11. Relay Board Location

FANFOLD GUIDE INSTALLATION INSTRUCTIONS

A fanfold paper tape clip enables the ASR 33 teletype to punch fanfold paper tape. Installation of the clip is necessary to prevent the ASR 33 from jamming when punching fanfold tape. To install the clip, adhere to the following instructions:

1) Remove the PLATEN KNOB (knob used to manually shift paper).
2) Remove the ON/OFF/LOCAL knob on the front panel.
3) Remove the TELETYPE NAME PLATE.
4) Remove four screws residing under the name plate screws attach cover to teletype).
5) Remove 3 screws on back of teletype (used to attach cover to teletype).
6) Remove paper roll.
7) Lift cover off teletype.
8) Loosen screw (see A on Figure 12-14) on paper tape punch by inserting screwdriver through the back of the paper tape punch.
9) Insert paper tape clip (see B on Figure 12-14) and tighten screw.
10) Replace cover and all removed screws and knobs.

Installing the Teletype Set

With the ASR 33 modified as described in the preceding section, the teletype is connected to the INTELLEC® 4/MOD 40 system as shown in Figure 12-15. Connections for the printer and keyboard loops are made at TB1 in the teletype's electrical service unit. The reader control loop is connected directly to the auxiliary control relay, which will have been installed previously according to the directions given on page 130.

An eight-pin connector on the rear panel of the INTELLEC® 4/MOD 40 system provides for all connections to the teletype. Figure 12-2 shows the location of this connector (J43). Table 12-6 below gives pin allocations. The appropriate mating connector is Cinch-Jones #P-308-CCT-L.

Teletype Connector (J43)

PIN #	SIGNAL FUNCTION
1	PRINTER LOOP (+)
2	PRINTER LOOP (–)
3	KEYBOARD LOOP (–)
4	KEYBOARD LOOP (+)
5	READER CONTROL LOOP (+)
6	READER CONTROL LOOP (–)
7	N. C.
8	N. C.

Table 12-6.

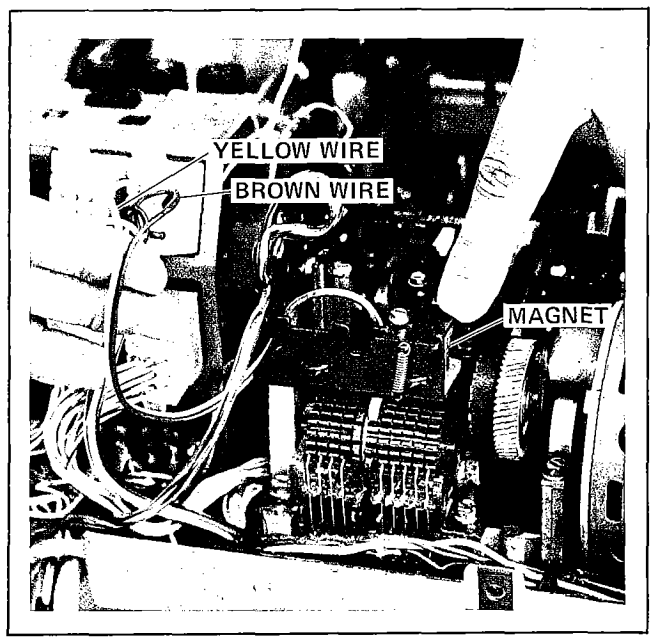


Figure 12-12. ESU Wiring

I/O Programming

Interconnection of the INTELLEC® 4/MOD 40 system and the teletype is only part of the interface. Connection enables electrical communication between processor and TTY, but it does not provide for the logic of such transfers. A software routine is required which is designed to handle the logical demands of teletype input and output. In essence, such a routine accepts bit-serial data from the teletype and assembles this data in parallel form within the processor.

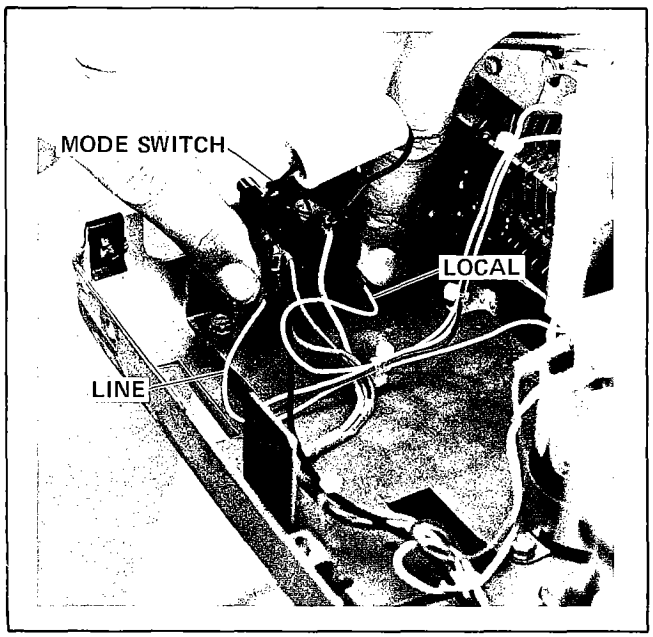


Figure 12-13. Teletype Mode Switch

Figure 12-14. Fanfold Guide Installation

Typically, the program instructs the processor to idle in a test loop, until such time as it senses the start bit (current zero) on the TTY input line. When the start bit arrives, the processor skips to an input routine, designed to draw in and assemble the serial bits of the ASCII character. With the eight-bit character fully assembled, the processor jumps to a monitor routine which determines whether the message calls for any further processor action. If so, the monitor directs a program jump to the required service routine. If not, the program jumps back to the idling loop, and waits for the next start bit to arrive. Figure 12-16 is a flowchart diagram of a typical programmed input routine.

The INTELLEC® 4/MOD 40 System Monitor contains teletype input and output programs that are suitable for use in the program development phases. Some users, however, will find it necessary to provide their own teletype I/O routines for specific applications. Such provisions are a part of the programming function and are technically outside the scope of this manual, but the programmer will not be able to write a suitable routine without having some knowledge of the teletype's requirements. For that reason, we present here some guidelines for writing a suitable program. Detailed instructions on coding will be found in the INTELLEC® 4/ system Operator's Manual, or in the MCS-40™ User's Manual.

A TYPICAL TELETYPE INPUT ROUTINE

A teletype transmits information at a maximum rate of 10 characters per second. Each character occupies a 0.1 second time frame.

Each frame contains 11 sub-intervals, periods in which the loop current may be either on or off, according to the predetermined logical profile of the character being transmitted.

The teletype normally idles in the "marking" condition; that is, with loop current on and indicating a logic "1." The current-off condition is called "spacing," and it indicates a logic "0."

The first interval in all character frames is a space, signalling that the loop is no longer idle, and that transmis-

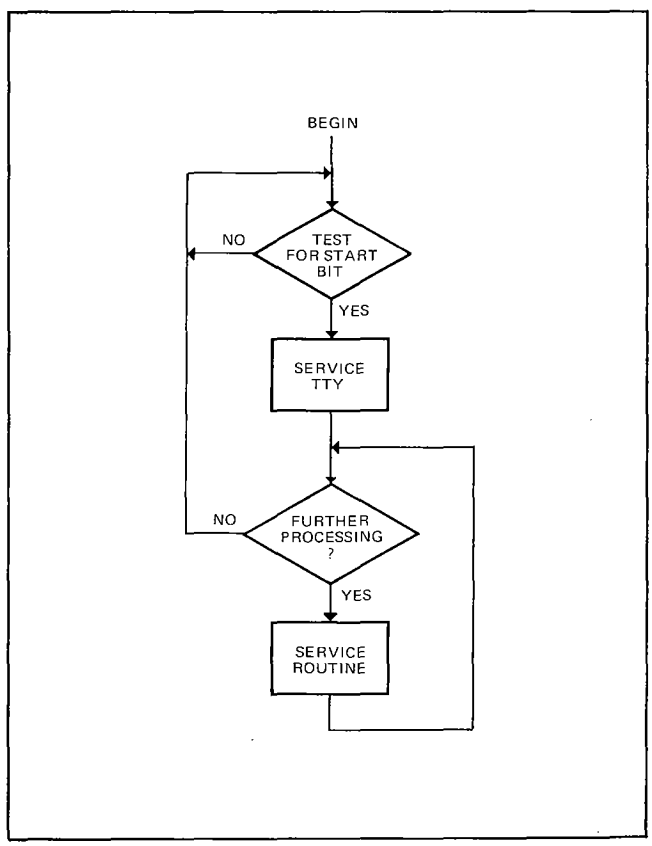


Figure 12-16. Teletype Input Routine

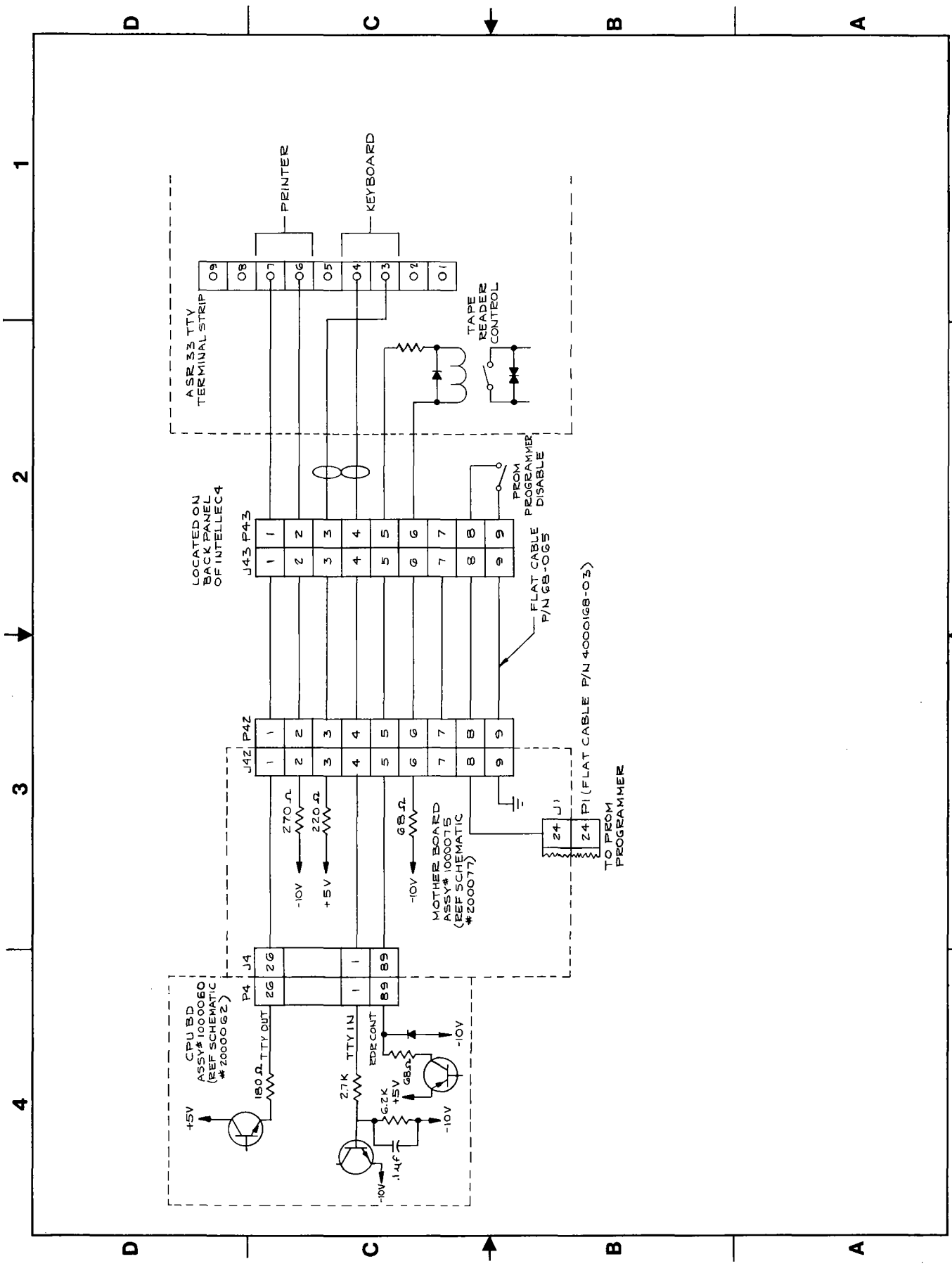


Figure 12-15. Teletype Installation

sion of a character is about to begin. This interval is known as the START bit. Its function is to synchronize transmitter and receiver. It has no significance as data.

The next seven intervals, which we shall refer to as b_1 - b_7, are reserved for intelligence transmission. The bits transmitted correspond to the 7 bits of an ASCII character. The b_8 interval contains the parity bit. (NOTE: The INTELLEC® 4/MOD 40 System Monitor ignores parity).

Interval b_9, is the STOP bit, and is always a marking interval. Like the START bit, the STOP has a control function rather than any data significance. The interval following the STOP, b_{10}, is likewise a marking bit. This is a recovery interval, prior to transmission of the next character.

The timing relationships are important, and must be observed in the program. Transmission of each bit in the TTY character frame requires 9.09 milliseconds.

The programmer implements software delays by knowing the instruction cycling time of the machine he is working with. In the case of the 4040 CPU, the basic cycle time is 10.8 μsec, with certain instructions requiring a double cycle for completion. Knowing this, and knowing that a delay of 9.09 milliseconds may be required, he can construct a nested iteration which will require approximately 842 cycles to complete. The routine might look like this, symbolically:

```
DELAY   FIM   2   50H
LOOP 1  FIM   0   DEH
LOOP 2  ISZ   1   LOOP2
        ISZ   0   LOOP2
        ISZ   2   LOOP1
        NOP
        NOP
        NOP
        BBL
```

Those unfamiliar with these mnemonics, should refer to Appendix A for explanation. This coded sequence requires 842 cycles for completion, providing a timed delay of approximately 9.09 milliseconds.

The programmer also needs to know that he can cause a mark on the teleprinter line, by writing any odd number into RAM output port 0. Writing any even number causes a space.

In similar fashion, an odd number written into the RAM 1 output port enables the tape reader. An even output turns off the drive.

Knowing this information, and knowing the limitations imposed by the instruction set of the processor chip, we can construct a routine for inputting a teletype character. This routine looks as follows, in logical outline (refer to the INTELLEC® 4 Programming Manual for an explanation of the machine instructions at your disposal).

1) Write a numerical "1" at RAM port 1, this enables the tape reader drive.
2) Initialize the working registers, by placing numerical "0" in index registers 2 and 3 and "8" in index register 4; registers 2 and 3 will be used to hold the input character, while 4 will be used as an overflow counter to determine when all eight bits have arrived.
3) Read ROM input port 0; a "1" indicates the TTY START bit.
4) If a "1" is not present, return to Step 3, otherwise continue; the program will loop between Steps 3 and 4, until such time as the START bit does arrive.
5) Delay for 4.55 milliseconds, to sample the "middle" of the teletype bit.
6) Write a numerical "0" at RAM port 1; this halts the reader, until the program determines what action may be required.
7) "Echo" the START bit, as described on page 00, by writing a numerical "0" into RAM port 0.
8) Delay for 9.09 milliseconds, to wait for the "middle" of the next bit.
9) Read ROM input port 0, to detect whether a "1" or a "0" is present.
10) Complement the accumulator, to obtain the proper logical polarity, and write the contents of the accumulator into the RAM 0 output port; this echoes the bit read in.
11) Shift right through the carry bit, placing the data bit in the carry position.
12) Transfer the contents of index register 2 to the accumulator, and again shift right through the carry bit; this places the most recent input data bit in the most significant position in the accumulator.
13) Transfer the contents of the accumulator back to index register 2.
14) Transfer the contents of index register 3 to the accumulator, and again shift right through the carry bit; this transfers any significant overflow from the preceding shift into index register 3.
15) Transfer the contents of the accumulator back to index register 3.
16) Increment index register 4; if the contents of register 4 is now zero, continue below; otherwise return to Step 8 and get the next bit.
17) Delay for 9.09 milliseconds, to wait for the middle of the STOP bit.
18) Write a numerical "1" into RAM port 0; this places a terminal "mark" on the teleprinter line.
19) Skip to the executive routine, to determine whether any processing is necessary; the executive will return control to the input routine when it is ready to receive the next character.

Upon completion of Step # 18, the input will be halted, and index registers 2 and 3 will contain the eight bits of the TTY character, as shown in Figure 12-17.

A TTY output routine will be similar in its construction to the input program just described. The programmer should now have some idea of how to develop such a program. Detailed help on programming procedures will be found in the INTELLEC® 4 system Programming Manual.

For reference, Appendix G of the INTELLEC® 4 system Operator's Manual contains the TTY input and output routines used in the INTELLEC® 4/MOD 40 System Monitor, as they appear in assembly language mnemonics. The input routine is a specialized program which ignores the parity bit, but should otherwise prove helpful for illustrative purposes.

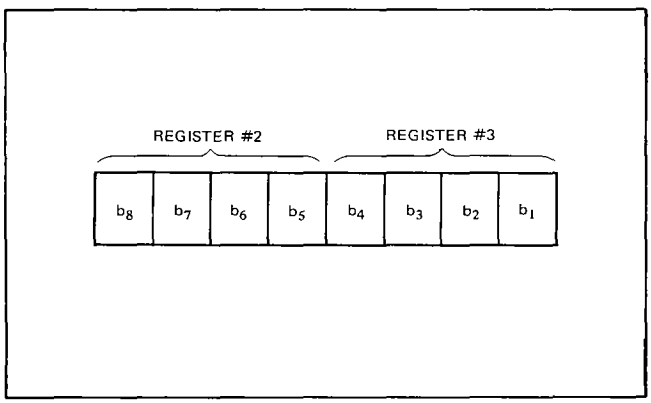


Figure 12-17. Teletype Character Input

imm4-90 HIGH-SPEED PAPER TAPE READER

Some INTELLEC® system users will find the teletype's paper tape reader too slow for their needs. The maximum transmission rate of the ASR 33 is 10 characters per second. Using the teletype's tape reader as the principal input device, it therefore takes a minimum of 100 seconds to load 1024 locations (1K) in program RAM. Loading a lengthy sequence of instructions can take 7 minutes or more.

At the customer's request, an optional paper tape reader will be supplied. Cables for the installation will be included. Users who wish to take advantage of the reader's higher transmission rates must also have at least one optional Input/Output Module in their system. The standard INTELLEC® system software is designed to accommodate the high-speed reader, when it is installed as explained on this page.

The high-speed tape reader contains an optical character sensor and a high-speed incremental drive. These enable it to transfer data at asynchronous rates in excess of 200 characters per second, at least 20 times faster than the teletype. It takes less than 30 seconds to load the entire program memory using the optional tape reader. The same evolution takes nearly 14 minutes when the teletype's reader is used. The convenience and economy of the high-speed reader will therefore be apparent in those situations that call for the loading of long programs, or frequent editing and re-assembly. In either case, the high-speed unit can shorten the development cycle appreciably.

Installing The Reader

The following items are required for installing the RRF7200 paper tape reader in the INTELLEC® 4/MOD 40 system:

PART #	DESCRIPTION	QUANTITY
0000083	Input/Output Module	1 ea.
4000237-02	I/O Cable Assembly	2 ea.
4000288-01	Reader Cable Assembly	1 ea.

Software furnished with the INTELLEC® 4/MOD 40 system is already designed to accommodate the high-speed reader. Neither the System Monitor nor the Assembler requires any modification.

MOUNTING

The high-speed paper tape reader is shipped in ready-to-use condition. The unit is fully enclosed and is designed for table-top operation. However, the user may also elect to mount the reader in a standard 19-inch modular equipment rack (8-3/4 vertical inches).

To prepare the instrument for rack-mounting, simply extract the lower four pan-head screws on the front panel, and remove the cast aluminum tape hopper. Then extract the four remaining screws, and withdraw the unit from its outer enclosure. Note that the power cord must pass through the cutout in the reader's rear panel, before the chassis will be completely free of its shroud.

Installation in the rack is the reverse of cover removal. First mount the reader chassis in its designated rack location. Then install the tape hopper. Eight 10-32 panel screws will speed removal of the machine during servicing.

POWER

Unless specified to the contrary, the reader leaves the factory wired for operation on single-phase AC power of 115 Volts (47-63 Hz). Consult the RRF7200 user's handbook for instructions on converting to 220 Volt operation. The tape reader is equipped with a three-wire power cord and a plug which grounds the chassis automatically when inserted in any standard grounded AC receptacle. Power consumption is 350 Watts (3.0 Amperes peak). The reader is fused at 2.0 Amperes, on its rear panel.

INPUT/OUTPUT PORT ASSIGNMENTS

The high-speed reader uses ROM input ports #6 and #7 for data, and ROM input port #4-BIT 3 for status ($\overline{\text{DATA READY}}$). ROM output port #4-BIT 3 is used for control ($\overline{\text{DRIVE}}$).

— NOTE —

At least one optional Input/Output Module (imm4-60) must be included in the system, in order to have the proper ports available. The standard INPUT PORT SELECT and OUTPUT PORT SELECT jumpers are in the proper position for operation of the reader.

The recommended jumpering scheme described above applies only to the latest version of the Input/Output Module, as supplied with current INTELLEC® 4 systems. These modules are readily identifiable by the production number 0000082-C, etched on their non-component side during manufacture.

Earlier versions of the Input/Output Module (0000083-A or 0000083-B) may not be used in support of the high-speed reader option without changing the reader cable.

The Input/Output Module may be installed in any of the vacant connectors J7 through J19 on the system mainframe.

INSTALLING I/O CABLES

Installation of the high-speed paper tape reader calls for the use of special I/O cables which connect the Input/Output Module to the connector boxes on the rear panel of the INTELLEC® 4/MOD 40 system. If the installation of these cables conflicts with cabling installed previously by the user or by the factory, the existing cables must first be removed.

Cable assembly #4000237-02 is installed by using two machine screws to mount the 37-pin connector in location #7 on the rear panel of the INTELLEC® 4/MOD 40 system. Figure 12-2 shows where to locate this connector. The connector on the opposite end of the cable plugs into J1 on the Input/Output Module. Route the cable carefully, and fold it flat, so that it will not interfere later with the closing of the cover.

A second cable assembly (#4000237-02) is installed in rear panel location #8. Its opposite end mates with J2 on the Input/Output Module.

CONNECTING THE READER

Before connecting the reader to the INTELLEC® 4/MOD 40 system, the Input/Output Module and its connecting cables must be installed as described above. Cabling should be checked for conformance to the specifications indicated in Table 12-7. Observe the recommended wiring scheme **exactly**, in order to ensure the proper logical interface with the INTELLEC® system software.

With the Input/Output Module installed, connect the reader to the INTELLEC® 4/MOD 40 system using the Y-cable furnished with the reader (#4000288-01):

Cable Assembly #4000288-01	J1 to Reader's rear panel connector J2 to INTELLEC's connector location #7 J3 to INTELLEC's connector location #8

High-Speed Reader Wiring

SIGNAL FUNCTION	PORT/BIT	I/O BOARD PIN #	BACK PANEL PIN #	READER PIN #
$\overline{\text{DATA TRACK 1}}$	ROM IPORT #6-0	J1-11	LOC 8 6	1
$\overline{\text{DATA TRACK 2}}$	ROM IPORT #6-1	J1-12	LOC 8 7	2
$\overline{\text{DATA TRACK 3}}$	ROM IPORT #6-2	J1-13	LOC 8 25	3
$\overline{\text{DATA TRACK 4}}$	ROM IPORT #6-3	J1-14	LOC 8 26	4
$\overline{\text{DATA TRACK 5}}$	ROM IPORT #7-0	J1-16	LOC 8 8	5
$\overline{\text{DATA TRACK 6}}$	ROM IPORT #7-1	J1-17	LOC 8 9	6
$\overline{\text{DATA TRACK 7}}$	ROM IPORT #7-2	J1-18	LOC 8 27	7
$\overline{\text{DATA TRACK 8}}$	ROM IPORT #7-3	J1-19	LOC 8 28	8
$\overline{\text{DATA READY}}$	ROM IPORT #4-3	J1-4	LOC 8 22	9
$\overline{\text{DRIVE}}$	ROM OPORT #4-3	J2-4	LOC 7 30	16
GROUND		J1-5, 10, 15, 10, -25, 30, 35, 40 J2-5, 10, 15, 20 -25, 30, 35, 40	1, 18, 19, 20, 37	11, 12, 13, 18, 24

Table 12-7.

Consult Figure 12-2 for the location of connectors on the rear panel of the INTELLEC® 4/MOD 40 system.

This completes the electrical installation of the RRF 7200 high-speed paper tape reader. Users who anticipate the need for programming this interface should refer to the directions given in I/O Programming on this page.

Functional Description of the High-Speed Reader

In its basic operation, the high-speed reader is an uncomplicated device. Since its status, control, and data functions are relatively few, interfacing is straightforward and I/O programming is greatly simplified.

The fan-folded tape being read is drawn from an input hopper on the front panel of the reader, by a sprocket wheel which engages the drive perforations in the tape. It passes over an eight-track optical sensor head and is deposited in an output hopper which is also located on the front panel of the machine.

The sprocket which drives the tape is positioned by a stepper motor, in response to signals from the reader's control electronics. Each negative-true $\overline{\text{DRIVE}}$ pulse applied externally to the reader causes the stepper motor to advance the tape incrementally, so that the next character to be read is positioned over the photo-sensor array. When the tape is in reading position the stepper motor is halted automatically, and the data at the tape address is then available in negative-true form at the reader's $\overline{\text{DATA TRACK 1}}$ through $\overline{\text{DATA TRACK 8}}$ outputs. These outputs correspond respectively to bits b_1 through b_8 of the bit-serial ASCII character from the teletype.

The $\overline{\text{DATA READY}}$ output is inhibited immediately upon receipt of the $\overline{\text{DRIVE}}$ command, and is restored as the optical sensor unit detects the next perforation on the tape's sprocket drive track. $\overline{\text{DATA READY}}$ thus signals the processor that the tape is in reading position and that the $\overline{\text{DATA TRACK}}$ outputs are valid.

An internal multivibrator establishes a minimum control delay of 3.33 milliseconds, between receipt of the $\overline{\text{DRIVE}}$ command and restoration of the $\overline{\text{DATA READY}}$ status line. This limits the reader's maximum speed to 300 characters per second. However, the minimum specific speed of the reader is 200 characters per second, and the status line in a machine that is functioning properly will be restored accordingly within 5.0 milliseconds of the $\overline{\text{DRIVE}}$ signal's leading edge. A status line which fails to come true within 5.0 milliseconds might therefore signify one of the following exceptional conditions:

a) out of tape
b) tape breakage
c) tape gate raised (loading position)
d) Drive mechanism is sluggish, or fails to respond to the $\overline{\text{DRIVE}}$ command

I/O Programming

Electrical connections between the INTELLEC® 4/MOD 40 system and the high-speed paper tape reader enable the processor to exchange signals with the reader. However, it is also necessary to have a program which is designed to handle the logic of such data transfers. The functional simplicity of the high-speed reader makes I/O programming relatively easy.

The INTELLEC® 4/MOD 40 System Monitor contains reader control routines that will be adequate for the purposes of most INTELLEC® users. Occasionally, however, a user will find it necessary to write his own programs for some specialized application. Such provisions are a part of the programming function and are technically outside the

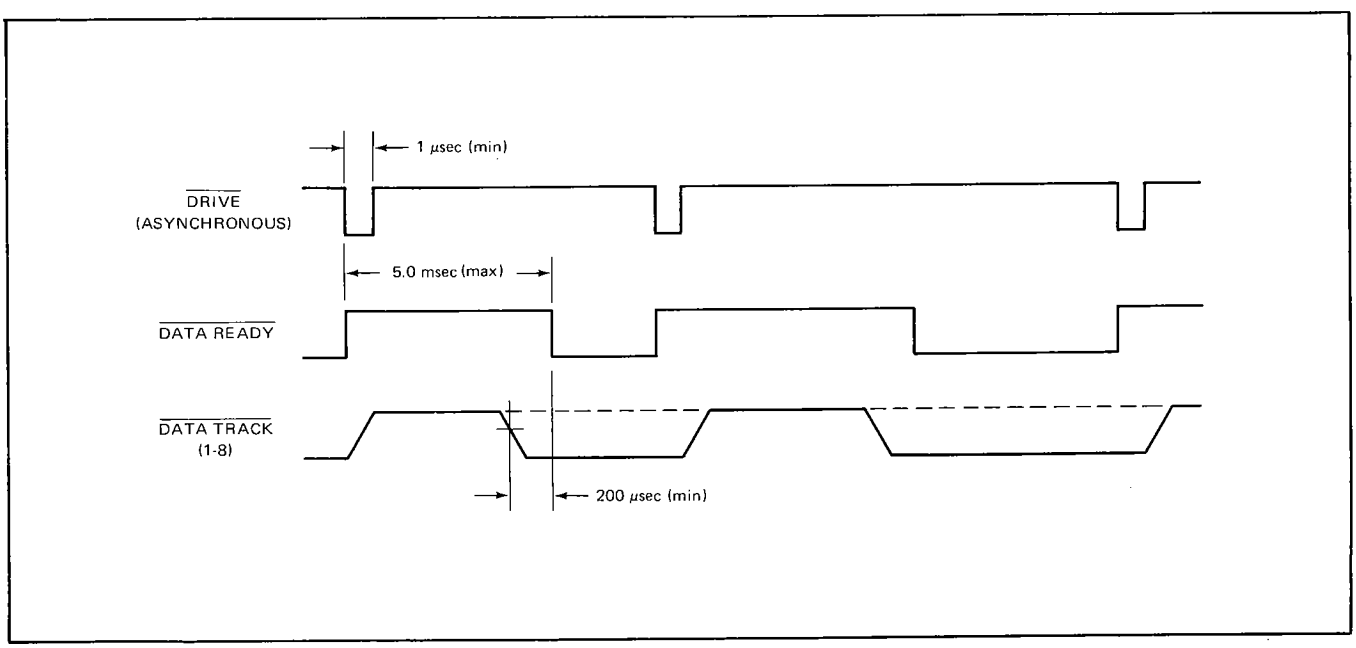


Figure 12-18. Tape Reader Timing

scope of this manual. But the programmer's ability to devise a suitable routine depends to some extent upon his knowledge of the reader's logical requirements. And because this information is not readily available elsewhere, we present here some guidelines for programming the interface. The instructions given are of a general nature only. For detailed help with the translation to machine code, consult the INTELLEC® 4 system Operator's Manual, or the MCS-40 User's Manual.

LOGIC OF THE INTERFACE

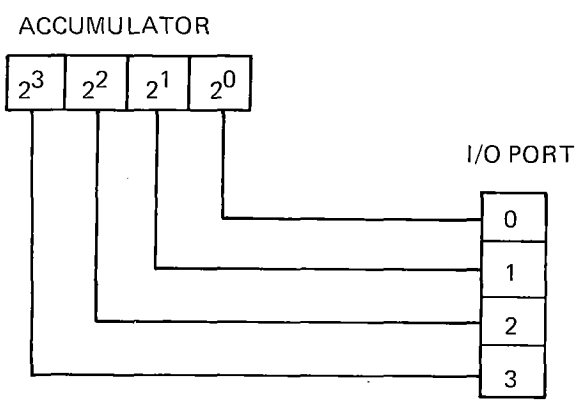
The high-speed reader may be advanced unidirectionally, from left to right, by the application of a momentary true level at its $\overline{\text{DRIVE}}$ input. Minimum duration of the $\overline{\text{DRIVE}}$ pulse is one microsecond. The reader's control logic stops the drive automatically when the tape is in position to read the next character.

Receipt of a $\overline{\text{DRIVE}}$ command promptly inhibits the $\overline{\text{DATA READY}}$, but if no malfunction exists this status line will be restored within 5.0 milliseconds. This gives the transport time to position the tape and to settle. When $\overline{\text{DATA READY}}$ again becomes true, the next character may be read. A status output which fails to come true within 465 processor cycles (approximately 5 milliseconds) could indicate one of the following exceptional conditions:

a) out of tape
b) tape breakage
c) tape guide raised (loading position)
d) drive mechanism is sluggish, or fails to respond to the $\overline{\text{DRIVE}}$ command

Data is available at the reader's eight outputs, $\overline{\text{DATA TRACK 1}}$ through $\overline{\text{DATA TRACK 8}}$, whenever the $\overline{\text{DATA READY}}$ status line is enabled. A true level on this line indicates that the tape is correctly positioned for reading and that the $\overline{\text{DATA TRACK}}$ outputs are valid.

The processor's accumulator is the principal register in all input and output transactions. Its four digits correspond one-for-one to the four lines of an addressed input or output port, with the register's least significant bits corresponding to the port's lowest numbered lines:



Reading and control are therefore accomplished by first examining and manipulating selected bits in the accumulator, then by executing an I/O command which addresses the appropriate port.

The reader's control, status, and data lines are all negative-true, and the interface is so designed that a "0" in the accumulator signifies a "true" condition at the corresponding line of an addressed port. A "1" conversely indicates a "false" condition.

The high-speed reader uses existing addressable ports on the optional I/O Module. ROM input ports #6 and #7 are used to receive the eight-line data input from the reader. $\overline{\text{DATA TRACK 1}}$ through $\overline{\text{DATA TRACK 4}}$ correspond to input port #6 (bits 0 through 3 respectively), while $\overline{\text{DATA TRACK 5}}$ through $\overline{\text{DATA TRACK 8}}$ correspond to input port #7 (bits 0 through 3 respectively). ROM input port #4-BIT 3 is allocated to status ($\overline{\text{DATA READY}}$), and ROM output port #4-3 is reserved for control ($\overline{\text{DRIVE}}$).

A TYPICAL INPUT ROUTINE

A typical routine for inputting a character from the reader will contain the following steps:

1) The Write a "0" into ROM output port #4-BIT 3. This starts the reader by pulsing the $\overline{\text{DRIVE}}$ input.
2) Write a "1" into output ROM output port #4-BIT 3, to return the $\overline{\text{DRIVE}}$ line to its quiescent condition. The width of the control pulse thus produced will be some multiple of the INTELLEC® system's basic 10.8 microsecond cycle time and will therefore be of ample duration to ensure reliable operation of the reader.
3) Up to 5.0 milliseconds will now elapse, before the $\overline{\text{DATA READY}}$ line comes true. In the interest of efficiency, other processing may be interleaved with the input program at this point, as long as control returns eventually to the input routine.
4) The reader will halt automatically, when the tape is in reading position.
5) Before reading in the character, the processor must test the reader's status by reading input port #4 and testing BIT 3. A "1" in that position indicates that the reader has not yet finished executing the previous $\overline{\text{DRIVE}}$ command. The program must idle in a test loop, repeatedly reading input #4, until Bit 3 is "0". If this does not occur within a reasonable length of time, say 20 milliseconds, jump to an exception routine designed to discover the source of the error.
6) Read input port #6 which contains the low-order four bits of the input character, and store these in a suitable index register.
7) Read input port #7 which contains the high-order four bits of the input character, and store these in an adjacent index register.

With the completion of Step #7, the designated registers will contain the eight-bit character from the reader. Another $\overline{\text{DRIVE}}$ command may be issued, as soon as this character has been duly processed.

APPENDIX A INSTRUCTION SUMMARY

This appendix provides a summary of 4004 and 4040 instructions. Abbreviations used are as follows:

Abbreviation	Description
A	The accumulator.
A_n	Bit n in the accumulator. (n=0, 1, 2, 3.)
ADDR	A read-only memory or program random-access memory address.
carry	The carry bit.
CR_n	Bit n in the COMMAND REGISTER. (n=0, 1, 2, 3.)
PC	The 12 bit Program Counter.
PCH	The high-order 4 bits of the Program Counter.
PCL	The low-order 4 bits of the Program Counter.
PCM	The middle 4 bits of the Program Counter.
RAM	Random-access memory.
REG	Any index register from 0 to 15.
Rx	Index register x. (x=0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.)
ROM	Read-only memory.
RP	Any index register pair from 0 to 14.
STK	The address stack.
$\overline{\text{value}}$	The number obtained by complementing each bit of value.
X:Y	The value obtained by concatenating the values X and Y.
[ ]	An optional field enclosed by brackets.
( )	Contents of register or memory enclosed by parentheses.
←	Replace value on left hand side of arrow with value on right hand side of arrow.

INDEX REGISTER INSTRUCTIONS

Format:

```
[LABEL:]    FIN     RP
          --or--
[LABEL:]    INC     REG
```

Code	Description	
FIN	(RP) ← ((PCH:R0:R1))	Load RP with 8 bits of ROM data addressed by register pair 0.
INC	(REG) ← (REG) + 1	Increment register REG.

INDEX REGISTER TO ACCUMULATOR INSTRUCTIONS

Format:

```
[LABEL:]    CODE    REG
```

Code	Description	
ADD	(A)←(A) + (REG) + (carry)	Add REG plus carry bit to accumulator.
SUB	(A)←(A) + (REG) + (carry)	Subtract REG from accumulator with borrow.
LD	(A)←(REG)	Load accumulator from REG.
XCH	(A)↔(REG)	Exchange contents of accumulator and REG.

ACCUMULATOR INSTRUCTIONS

Format:

```
[LABEL:]    CODE
```

Code	Description	
CLB	(A)←0, (carry)←0	Clear both accumulator and carry.
CLC	(carry)←0	Clear carry.
IAC	(A)←(A) + 1	Increment accumulator.
CMC	(carry)←$\overline{(carry)}$	Complement carry.
CMA	(A)←$\overline{(A)}$	Complement each bit of the accumulator.
RAL	A_{n+1}←A_n, (carry)←A_3, A_0←(carry)	Rotate accumulator left through carry.
RAR	A_n←A_{n+1}, (carry)←A_0, A_3←(carry)	Rotate accumulator right through carry.
TCC	(A)←0 A_0←(carry), (carry)←0	Transmit the value of the carry to the accumulator, then clear carry.
DAC	(A)←(A) − 1	Decrement accumulator.

TCS	If (carry) = 0, (A)←9_{10}
	If (carry) = 1, (A)←10_{10}
	(carry)←0
		Adjust accumulator for decimal subtract.

STC	(carry)←1	Set carry.
DAA	If (A) > 9_{10} or (carry) = 1, (A)←(A) + 6	Adjust accumulator for decimal add.
KBP	Convert A_3 A_2 A_1 A_0	Convert accumulator from 1 of n code to binary value.

IMMEDIATE INSTRUCTIONS

Format:

```
[LABEL:]    FIM     RP, DATA
          --or--
[LABEL:]    LDM     DATA
```

Code	Description	
FIM	(RP)←DATA	Load 8 bit immediate DATA into register pair RP.
LDM	(A)←DATA	Load 4 bit immediate DATA into the accumulator.

TRANSFER OF CONTROL INSTRUCTIONS

Format:

```
[LABEL:]    JCN     CN ADDR
          --or--
[LABEL:]    JIN     RP
          --or--
[LABEL:]    ISZ     REG
          --or--
[LABEL:]    JUN     ADDR
          --or--
[LABEL:]    JC      ADDR
          --or--
[LABEL:]    JNC     ADDR
          --or--
[LABEL:]    JZ      ADDR
          --or--
[LABEL:]    JNZ     ADDR
          --or--
[LABEL:]    JT      ADDR
          --or--
[LABEL:]    JNT     ADDR
```

Code	Description	
JUN	(PCH: PCM: PCL:)←ADDR	Jump to location ADDR.
JIN	(PCM: PCL)←(RP)	Jump to the address in register pair RP.
ISZ	(REG)←(REG) + 1 If result = 0, (PL)←(PL) + 2 If result≠ 0, (PCM: PCL)←ADDR	Increment REG. If zero, skip. If non-zero, jump to ADDR.
JCN	If CN true, (PCM: PCL)←ADDR If CN false, (PL)←(PL) + 2	Jump to ADDR if condition true.
JC	If carry = 1, (PCM: PCL)←ADDR If carry = 0, (PL)←(PL) + 2	Jump to ADDR if carry set.
JNC	If carry = 0, (PCM: PCL)←ADDR If carry = 1, (PL)←(PL) + 2	Jump to ADDR if carry reset.
JZ	If A = 0, (PCM: PCL)←ADDR If A ≠0, (PL)←(PL) + 2	Jump to ADDR if accumulator = 0.
JNZ	If A ≠0, (PCM: PCL)←ADDR If A = 0, (PL)←(PL) + 2	Jump to ADDR if accumulator ≠0.
JT	If TEST = 1, (PCM: PCL)←ADDR If TEST = 0, (PL)←(PL) + 2	Jump to ADDR if TEST = 1.
JNT	If TEST = 0, (PCM: PCL)←ADDR If TEST = 1, (PL)←(PL) + 2	Jump to ADDR if TEST = 0.

SUBROUTINE LINKAGE INSTRUCTIONS

Format:

```
[LABEL:]    JMS    ADDR
          --or--
[LABEL:]    BBL    DATA
```

Code		Description
JMS	(STK)←(PC), (PC)←ADDR	Call subroutine and push return address onto stack.
BBL	(PC)←(STK), (A)←DATA	Return from subroutine and load accumulator with immediate DATA.

NOP INSTRUCTION

Format:

```
[LABEL:]          NOP
```

Code		Description
NOP	----------	No operation.

MEMORY SELECTION INSTRUCTIONS

Format:

```
[LABEL:]    SRC    RP
          --or--
[LABEL:]    DCL
```

Code		Description
SRC	DATA BUS←(RP)	Contents of RP select a RAM or ROM address to be used by I/O and RAM instructions.
DCL	CR_2:CR_1:CR_0←A_2:A_1:A_0	Select a particular RAM bank.

I/O AND RAM INSTRUCTIONS

Format:

[LABEL:] CODE

Code	Description	
WRM	(RAM)←A	Write accumulator to RAM.
WMP	RAM output port←(A)	Write accumulator to RAM output port.
WRR	ROM output port←(A)	Write accumulator to ROM output port.
WPM	(PRAM)←(A)	Write accumulator to Program RAM.
WRn	RAM status character n←(A)	Write accumulator to RAM status character n (n=0, 1, 2 or 3).
RDM	(A)←RAM	Load accumulator from RAM.
RDR	(A)←ROM input port	Load accumulator from ROM input port.
RDn	(A)←RAM status character n	Load accumulator from RAM status character n (n=0, 1, 2, or 3).
ADM	(A)←(A) + (RAM) + (carry)	Add RAM data plus carry to accumulator.
SBM	(A)←(A) + $\overline{(RAM)}$ + $\overline{(carry)}$	Subtract RAM data from accumulator with borrow.

4040 ONLY INSTRUCTIONS

Format:

[LABEL:] CODE

Code	Description	
HLT	----------	Instruction execution halts until an external interrupt is received.
BBS	(PC)←(STK), DATA BUS←SRC	Return from subroutine and restore SRC address.
LCR	(A)←COMMAND REGISTER	DATA RAM bank and ROM bank status loaded into accumulator.
OR4	(A)←(A) OR (R4)	OR accumulator with index register 4.
OR5	(A)←(A) OR (R5)	OR accumulator with index register 5.
AN6	(A)←(A) AND (R6)	AND accumulator with index register 6.
AN7	(A)←(A) AND (R7)	AND accumulator with index register 7.
DB0	CR_3←0	Select ROM bank 0.
DB1	CR_3←1	Select ROM bank 1.
SB0	----------	Select index register bank 0.
SB1	----------	Select index register bank 1.
EIN	----------	Enable interrupt detection.
DIN	----------	Disable interrupt detection.
RPM	(A)←(PRAM)	Load accumulator from 4289-controlled Program RAM.

APPENDIX B EXECUTION CYCLE

DATA BUS CONTENTS DURING X2 AND X3 INSTRUCTION CYCLES

Instructions	Data Bus Contents at X2	Data Bus Contents at X3	Explanation Notes
NOP, HLT, DB0, DB1, SB0, SB1, EIN, DIN	1111	1111	
*JCN A_2, A_1	1111 1111	1111 1111	
*FIM RRR0 D_2, D_1	(RRR0)	(RRR1)	The contents of address pair RRR
SRC RR1	(RRR0)	(RRR1)	
*FIN RRR0 2nd cycle	(RRR0) 1111	(RRR1) 1111	
JIN RRR0	(RRR0)	(RRR1)	
*JUN A_3 A_2, A_1	A_3 A_3	1111 1111	
*JMS A_3 A_2, A_1	A_3 A_3	1111 1111	
INC RRRR	(RRRR)	(RRRR)+1	Content of register RRRR; content + 1 of RRRR
*ISZ RRRR A_2, A_1	(RRRR) 1111	(RRRR)+1 1111	
ADD RRRR	(RRRR)	1111	Content of register RRR
SUB RRRR	(RRRR)	1111	
LD RRRR	(RRRR)	1111	
XCH RRRR	(RRRR)	(ACC)	Content of register RRRR; the content of ACC
BBL	DDDD	1111	Data DDDD
LDM	DDDD	1111	Data DDDD
WRM, WR0, WR1, WR2, WR3, WPM, WMP, WRR	(ACC) (ACC)	111(CY) 111(CY)	Content of accumulator and CY.

*Instruction requires 16 clock periods. Top line = data bus contents during first X2 and X3 cycles, bottom line = data bus contents during second X2 and X3 cycles.

DATA BUS CONTENTS DURING X2 AND X3 INSTRUCTION CYCLES (Continued)

Instructions	Data Bus Contents at X2	Data Bus Contents at X3	Explanation Notes
RDM, RD0, RD1, RD2, RD3, ADM, SBM, RDR	(M) or (Input)	(m) or (input)	Data fetched from RAM or input.
CLB, CLC, IAC, CMC, CMA, RAL, PAR, TTC	0000	1111	
TCS	1001	1111	
STC, DAC, DCL	1111	1111	
DAA	0000 or 0110	1111	X2 depends on ACC content.
KBP	000, 0001, 0010, 0011, 0100, 1111	1111	X2 depends on ACC content.
BBS	(SRCH)	(SRCL)	SRCH is high-order four bits of SRC register.
LCR	(COM REG)	1111	
OR4	(0100)	1111	
OR5	(0101)	1111	
AN6	(0110)	1111	
AN7	(0111)	1111	
RPM	(P.M.)	(P.M.)	Program Memory Content

*Instruction requires 16 clock periods. Top line = data bus contents during first X2 and X3 cycles, bottom line = data bus contents during second X2 and X3 cycles.

INTEL CORPORATION • 3065 Bowers Ave., Santa Clara, California 95051 • (408) 246-7501