Digital Equipment Corporation
Maynard, Massachusetts

digital

PDP-15 Systems

Interface Manual

DEC-15-HOAC-D

PDP-15 SYSTEMS
INTERFACE MANUAL

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

1st Printing November 1969
2nd Printing (Rev) February 1970
3rd Printing (Rev) June, 1971

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

ILLUSTRATIONS (Cont)

TABLES

SYSTEMS REFERENCE MANUAL – Provides overview of PDP-15 hardware and software systems and options, instruction repertoire, expansion features, and descriptions of system peripherals. (DEC-15-BRZC-D)

USER'S HANDBOOK VOLUME 1, PROCESSOR – Principal guide to system hardware includes system and subsystem features, functional descriptions, machine-language programming considerations, instruction repertoire, and system expansion data. (DEC-15-H2DB-D)

VOLUME 2, PERIPHERALS – Features functional descriptions and programming considerations of peripheral devices. (DEC-15-H2DB-D)

OPERATOR'S GUIDE – Lists procedural data, including operator maintenance, for using the operator's console and the peripheral devices associated with PDP-15 Systems. (DEC-15-H2CB-D)

PDP-15/10 SYSTEM USER'S GUIDE – Features COMPACT and Basic I/O Monitor operating procedures. (DEC-15-GG1A-D)

PDP-15/20 SYSTEM USER'S GUIDE – Lists Advanced Monitor System operating procedures. (DEC-15-MG2B-D)

BACKGROUND/FOREGROUND MONITOR SYSTEM USER'S GUIDE – Lists operating procedures for the DECtape and disk-oriented Background/Foreground monitors. (DEC-15-MG3A-D)

PDP-15/10 SOFTWARE SYSTEM – Describes COMPACT software system and Basic I/O Monitor System. (DEC-15-GR1A-D)

PDP-15/20/30/40 ADVANCED MONITOR SOFTWARE SYSTEM – Describes Advanced Monitor System; programs include system monitor language, utility, and application types; operation, core organization, and input/output operations within the monitor environment are discussed. (DEC-15-MR2A-D)

PDP-15/30/40 BACKGROUND/FOREGROUND MONITOR SOFTWARE SYSTEM – Describes Background/Foreground Software System including the associated language, utility, and applications program. (DEC-15-MR3A-D)

RSX USER MANUAL – Describes the disk-oriented real time system executive language and applications.

MAINTENANCE MANUAL VOLUME 1, PROCESSOR – Provides block diagram and functional theory of operation of the processor logic; lists preventive and corrective maintenance data. (DEC-15-H2BB-D)

VOLUME 2, ENGINEERING DRAWINGS – Provides engineering drawings and signal glossary for the basic processor and options. (DEC-15-H2BB-D)

INSTALLATION MANUAL – Provides power specifications, environmental considerations, cabling, and other information pertinent to installing PDP-15 Systems. (DEC-15-H2AB-D)

ACCEPTANCE TEST PROCEDURES – Lists step-by-step procedures designed to insure optimum PDP-15 Systems operation.

PDP-15 MODULE MANUAL – Provides characteristics, specifications, timing and functional descriptions of modules used in PDP-15 Systems. (DEC-15-H2EA-D)

INTERFACE MANUAL – Provides information for interfacing devices to a PDP-15 System. (DEC-15-H0AC-D)

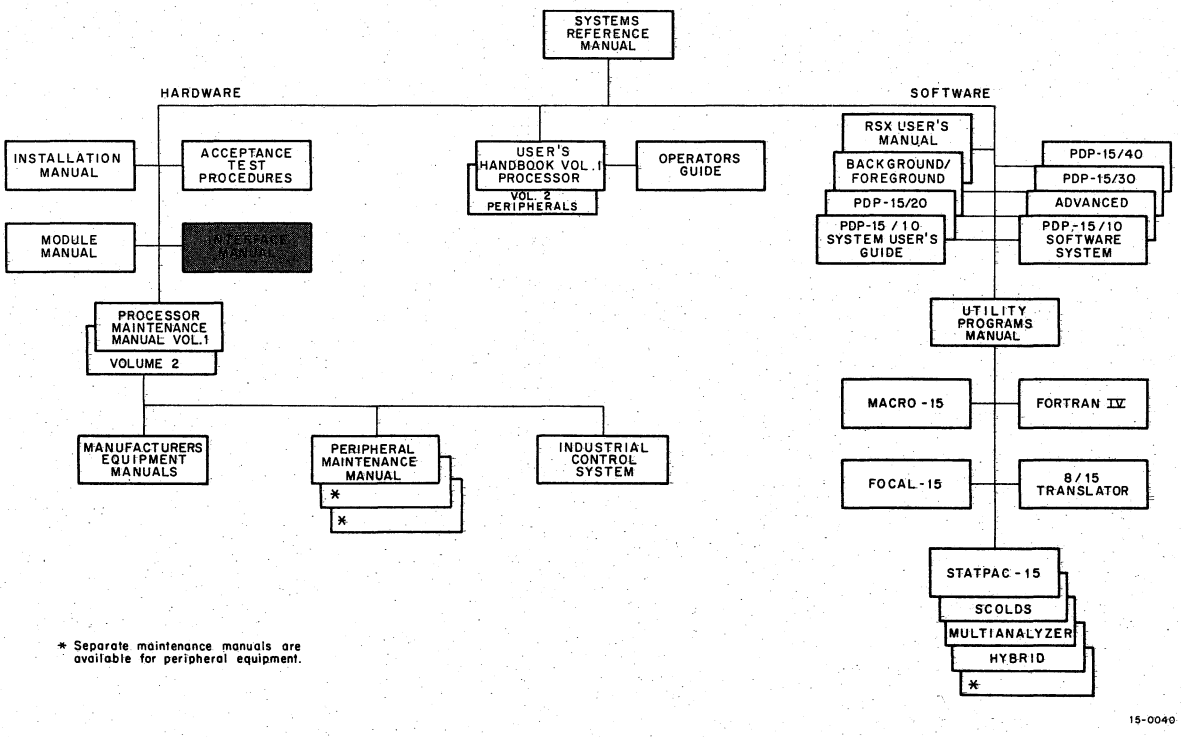
UTILITY PROGRAMS MANUAL – Provides utility programs common to PDP-15 Monitor systems. (DEC-15-YWZA-D)

MACRO-15 – Provides MACRO assembly language for the PDP-15. (DEC-15-AMZA-D)

FORTRAN IV – Describes PDP-15 version of the FORTRAN IV compiler language. (DEC-15-KFZA-D)

FOCAL-15 – Describes an algebraic interactive compiler level language developed by Digital Equipment Corporation. (DEC-15-KJZB-D)

PDP-15 FAMILY OF MANUALS



* Separate maintenance manuals are available for peripheral equipment.

CHAPTER 1
PDP-15 LOGIC SYMBOLOGY

The purpose of this chapter is to explain the organization of the PDP-15 logic diagrams to aid the logic designer or engineer involved in interfacing to a PDP-15 System. The following topics will be discussed:

a. PDP-15 System organization
b. Logic Symbology
c. Designators
d. Line connections

1.1 PDP-15 SYSTEM ORGANIZATION

All PDP-15 systems, from the basic system (PDP-15/10) through the disk-oriented background/foreground system (PDP-15/40), are organized in three main sections: Central Processor (CP), Memory (Internal Storage), and Input/Output (I/O) Processor. The designations of these three sections are as follows: KP15 Central Processor, MM15 Memory, KD15 Input/Output Processor. All PDP-15 logic diagrams have been drafted utilizing a computer controlled automated drafting system (ADS), and therefore have similar characteristics. The logic diagrams flow from left-to-right wherever possible.

Each logic print has a coordinate marking system to help locate gates. The system is as follows:

a. Numbers 1-8, from right to left, across the top and bottom margins.
b. Letters A-D, from bottom to top, on left and right margins.

A label at print coordinate A-1 (lower right corner) identifies the logic diagram. The label has three parts:

a. Print name – e.g., Indicator Strobes
b. Print size and type – e.g., D-BS which means D size, Block Schematic
c. Print number – This is section oriented; e.g., KP15-0-25 which means Print 25 of the Central Processor section.

1.2 LOGIC SYMBOLOGY

The logic modules utilized in the PDP-15 are primarily DEC M-series, which is the integrated circuit, positive logic series. The voltages used are:

Low (L) = 0V (0V -+0.4V)
High (H) = +3V (+2.4V -+3.6V)

(For specific information on operating frequencies, loading etc., see the individual module specifications.)

The logic symbology used is MIL-STD-806B. The gating symbols use small circles at the inputs of gates to indicate that a low signal activates the function. The absence of a circle indicates that a high signal activates the function. The presence or absence of a circle at the output of a gate indicates that the output is Low (L) or High (H), respectively, when the gate has been activated (its output is true). A gate's output is false if it is at a voltage different from that shown by the gate's polarity indicator (presence or absence of circle). Suffixes L or H indicate the low or high level of a signal when it is true or enabled.

1.2.1 Logic Gates

Boolean functions are symbolized as follows:

* Logical AND
\+ Logical OR (inclusive)
\- Logical negation (the vinculum is not used)

The most commonly used gating symbols are the NAND (Figures 1-1, 1-2); NOR (Figures 1-3, 1-4), and the Inverter (Figures 1-5, 1-6). Each figure shows both the symbol and a Boolean expression of the logical operation it performs.

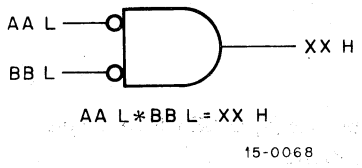


Figure 1-1 NAND Gate

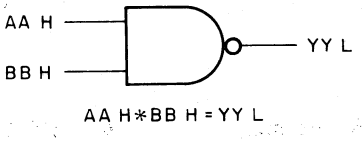


Figure 1-2 NAND Gate

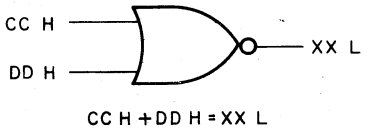


Figure 1-3 NOR Gate

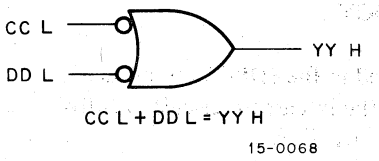


Figure 1-4 NOR Gate

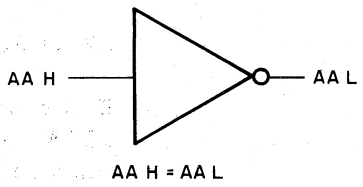


Figure 1-5 Inverter

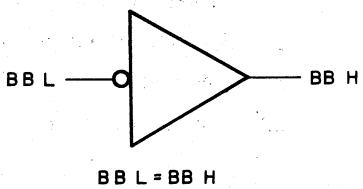


Figure 1-6 Inverter

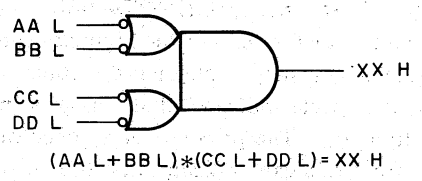


Figure 1-7 Basic Logic Relationships

Some of the logic gates are made up of combinations of the basic functions. The operations of these gates can be determined by the combined use of the basic logical relationships. (Figure 1-7.)

1.2.2 Flip-Flops

A major part of the work accomplished by the computer's logic is performed by flip-flops. The following figures illustrate the types of flip-flops used in the PDP-15 System.

The flip-flop in Figure 1-8a requires a high data input when clocked to set; the flip-flop in 1-8b requires a

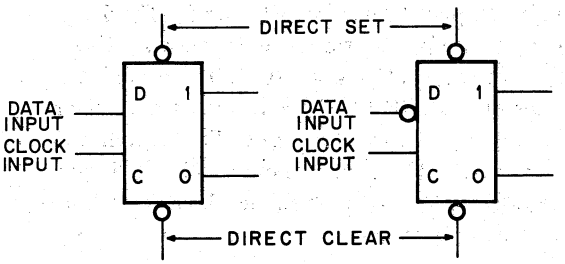


Figure 1-8 D Flip-Flop – Edge Triggered

low data input to be set. The D flip-flop will be drawn on the logic diagrams to agree with the voltage level of the data input necessary to set the flop.

Figure 1-9 shows the J-K flip-flop. The flop has direct set and reset inputs, and clock-gated set and reset inputs. If both gated inputs are high when the flop is clocked, the flop will complement.

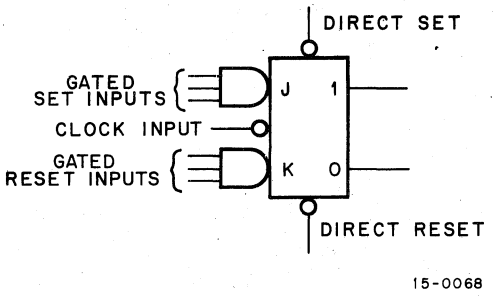
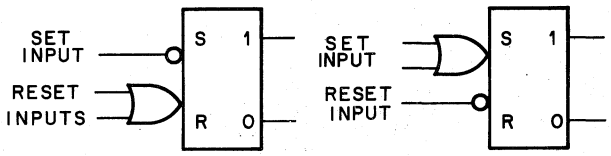


Figure 1-9 J-K Master-Slave Flip-Flop

The flip-flop in Figure 1-10a can be reset by either one of two inputs; the flip-flop in Figure 1-10b has the OR gate on its set side.



1-10a 1-10b

Figure 1-10 R-S Flip-Flop

1.2.3 Variable Clock

The PDP-15 uses variable clocks to generate timing signals and to control gating functions. Figure 1-11 shows a typical clock configuration.

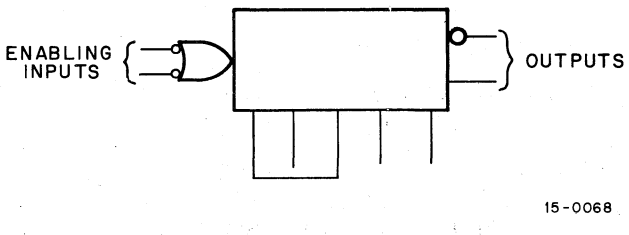


Figure 1-11 Variable Clock

1.2.4 Pulse Amplifier

A typical pulse amplifier as used in the PDP-15 is shown symbolically in Figure 1-12.

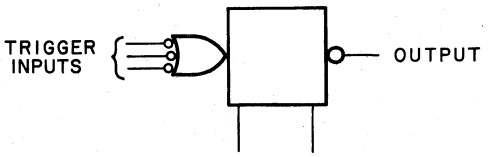


Figure 1-12 Pulse Amplifier

1.2.5 Miscellaneous Logic Symbology

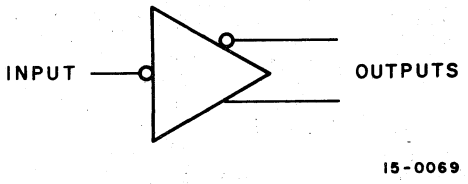


Figure 1-13 I/O Bus Receiver

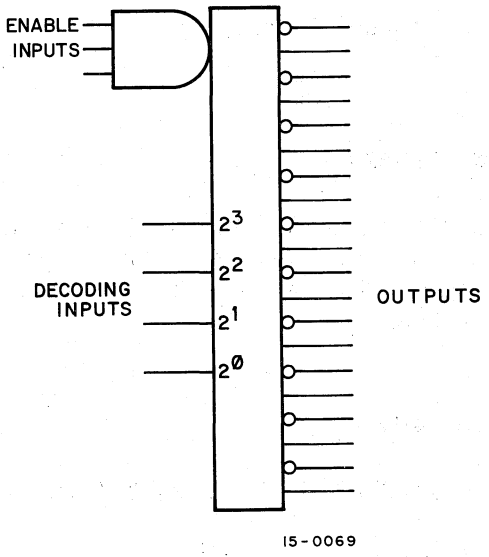


Figure 1-14 Binary to Decimal Decoder

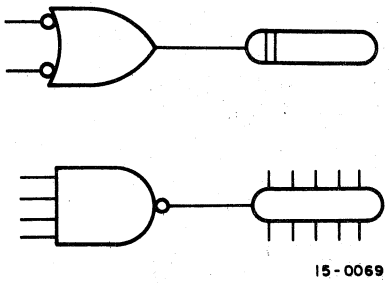


Figure 1-15 Delays

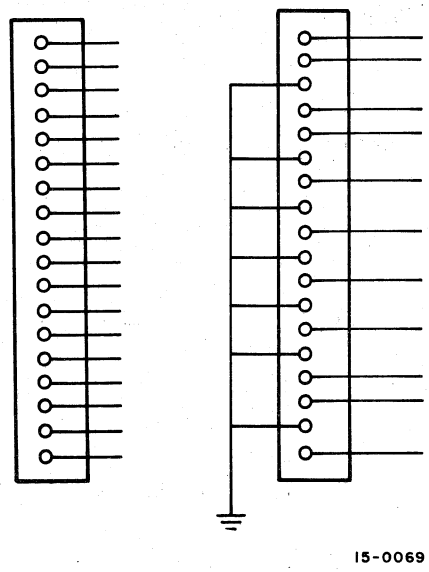


Figure 1-16 Connectors

1.3 DESIGNATORS

The logic symbology used thus far has not shown any literal designators. The following paragraphs will describe and discuss the information that accompanies the logic symbols.

1.3.1 Logic Designators

The NAND gate, Figure 1-17, contains the following information:

a. type (M112)

b. location in the computer (L18 - of applicable section - CP, I/O, or MEM)

c. input pins on specific module (P1, R1)

d. output pin (S1).

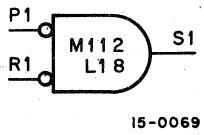


Figure 1-17 NAND Gate

The flip-flop modules also have a name associated with them, such as READ (Figure 1-18).

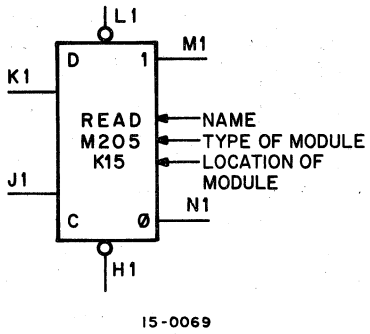


Figure 1-18 Read Flip-Flop

1.3.2 Signal Designators

All logical expressions are broken into three parts (Figure 1-19):

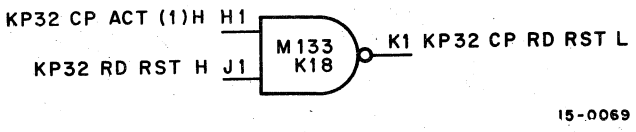


Figure 1-19 Signal Designators

a. Prefix – identifies where the signal originated (KP32 - Print 32 of the central processor).

b. Signal name – RD RST, CP ACT.

c. Polarity or voltage indicator (H or L). Note that in the case of flip-flops, a (1) or (0) is present to indicate the state of the flip-flop necessary for the desired voltage. For example, KP32 CP ACT (1) H is the 1 side high output of a flip-flop named CP ACT.

1.3.3 Signal Name Changes

When a signal originating from a flip-flop is inverted through an inverter, the signal name will be changed (Figure 1-20).

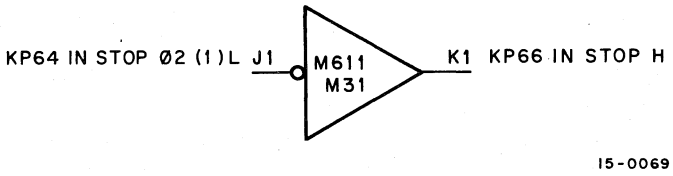


Figure 1-20 Signal Name Changes

This change in name avoids possible confusion about where the signal actually originated.

1.3.4 The Dash

Logical NOT or negation is shown by a dash (-) in front of the logical expression. It is never used in front of flip-flop designators. The gate in Figure 1-21 shows negated functions. A true output from the gate is described by the following expression:

$$\text{DEFER (1)} * \overline{\text{ISZ}} * \overline{\text{JMP}} * \overline{\text{XCT}}$$

NOTE

The vinculum is used only for explanation purposes.

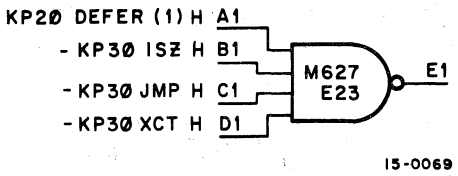


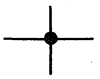
Figure 1-21 The Dash

A dash (-) within a logical expression, and not in front of it, does not indicate negation; it represents the word TO and is used in expressions that perform a transfer type function. KP32 MO - MDL H would be read as MO TO MDL.

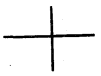
1.4 LINE CONNECTIONS

The conventions used to indicate when lines merely cross over each other, and when there is an electrical connection are as follows:

a. Dotted junctions as shown below are not used.



b. Lines that merely cross each other on the logic diagram are shown as follows:



c. A junction of signal paths, or an electrical connection is indicated as follows:

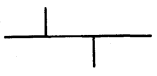


Figure 1-22a shows an example from the logic prints with lines crossing and junction points. Figure 1-22b has dots drawn at the junction points to show where junction points are, although dots are never used in the logic symbology.

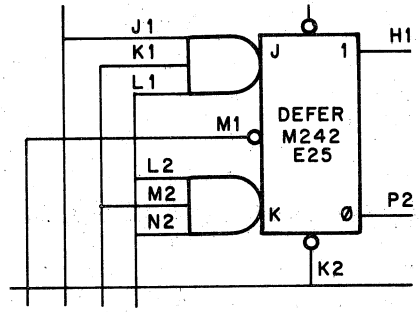


Figure 1-22a

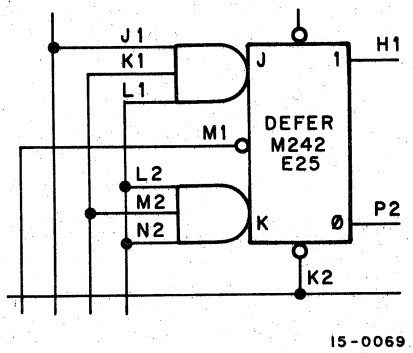


Figure 1-22b

CHAPTER 2
PDP-15 INTERFACE MODULES

This chapter provides descriptions of special modules used to interface the PDP-15 I/O Processor to peripheral devices, or PDP-15 devices to the PDP-9. Rules for wiring these modules together are given in Chapter 5.

DEC builds three series of compatible below-ground logic (the B-, R- and S-series), two series of compatible above-ground logic (K- and M-series), an extensive line of modules to interface different types of logic (W-series), a line of special purpose modules (G-series), and a line of support hardware for its module line (H-series).

With few exceptions, the DEC below-ground logic operates with logic levels of ground to -0.3V (upper level and -3.2V to -3.9V (lower level) using diode gates which draw input current at ground and supply output current at ground. Figure 2-1 shows the voltage spectrum of negative logic systems.

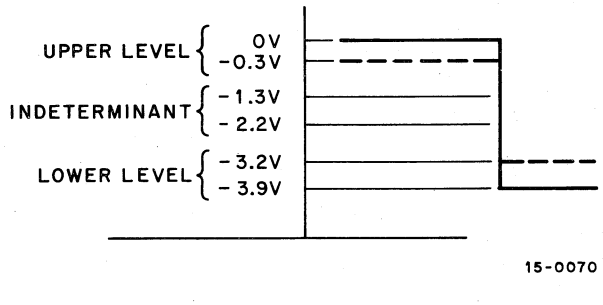


Figure 2-1 Voltage Spectrum of Negative Logic Systems

The compatible above-ground iogic generally operates with levels of ground to +0.4V, (lower level) and +2.4 to +3.6V (upper level) using TTL or TTL-compatible circuits whose inputs supply current at ground and whose outputs sink current at ground. Figure 2-2 shows their voltage spectrum.

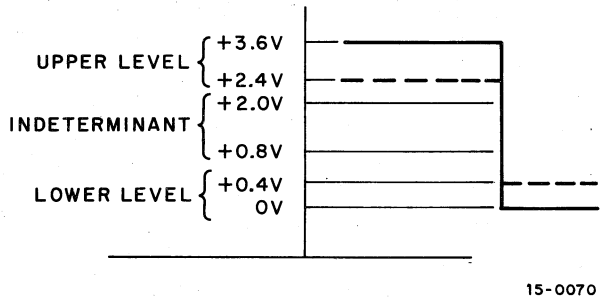


Figure 2-2 Voltage Spectrum of TTL Logic

Finally, a set of special modules designed to operate on the PDP-15 I/O bus are available. Figure 2-3 indicates the voltage spectrum that they operate in.

The DIGITAL Logic Handbook, C–105, is recommended reading for those not already familiar with the basic principles of digital logic and the type of circuits used in DEC logic modules.

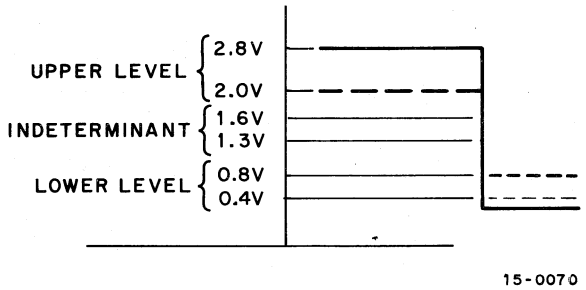


Figure 2-3 Voltage Spectrum for Positive PDP-15 I/O Bus Logic

All modules used in the interface circuits of this manual are M-series. Several are level converters designed to match positive logic to negative logic.

2.1 MEASUREMENT DEFINITIONS

Timing is measured with the input driven by a gate or pulse amplifier of the series under test and with the output loaded with gates of the same series, unless otherwise specified. Percentages are assigned as follows: 0% is the initial steady-state level, 100% is the final steady-state level, regardless of the direction of change.

Input/output delay is the time difference between input change and output change, measured from 50% input change to 50% output change. Rise and fall delays for the same module usually are specified separately.

Risetime and falltime are measured from 10% to 90% of waveform change, either rising or falling.

2.2 LOADING

Input loading and output driving are specified in "units", where one unit is 1.6 mA by definition. The inputs to low speed gates usually draw 1 unit of load. High speed gates draw 1-1/4 units or 2 mA.

2.3 MODULE DESCRIPTIONS

2.3.1 M104 Multiplexer Module

This is an M-series single-height module which contains a single multiplexer subsystem. Its equivalent circuit is shown in Figure 2-4. The inputs are standard TTL voltages and have the following input pins and loads:

Input Pin	Load (Units)
H2	2.5
S2	1
H1	6
E2	3
N1	1
F2	1-1/4
K2	68 Ω Termination
S2	1

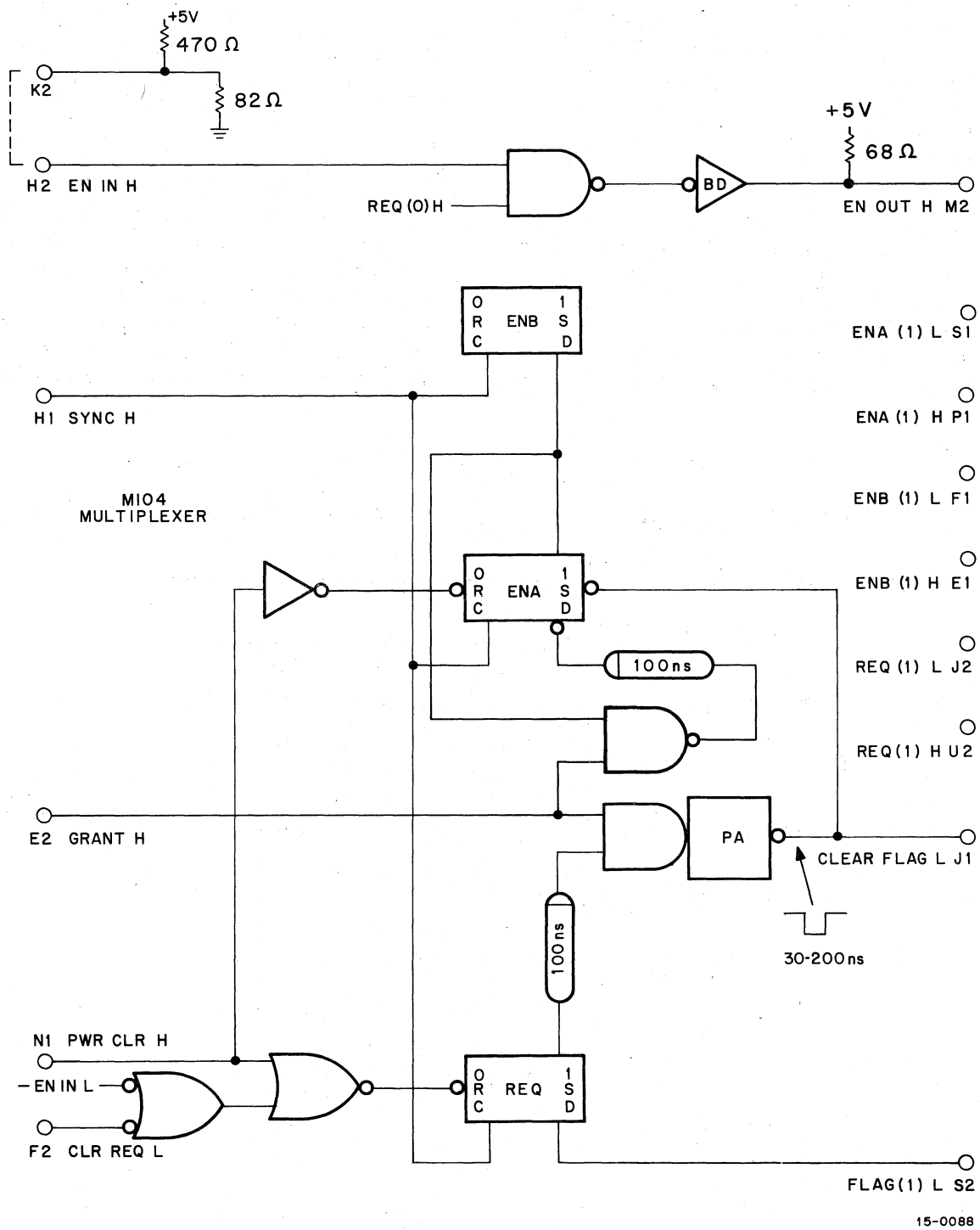


Figure 2-4 M104 Equivalent Circuit

Outputs: The output gates can drive as follows:

Output Pin	#Loads it can Drive
U2	5
J2	8
P1	9
S1	10
E1	10
F1	10
M2	PDP-15 I/O Bus Compatible (30 units)
J1	7

Power: 1 Watt

Application: The M104 module has been designed specifically for controllers of PDP-15 peripherals. It is used in all controllers which make use of the API or data channel facilities in the I/O processor. It accepts a request from the controller logic at its FLAG (1) H input and synchronizes this request to the I/O SYNC H pulses issued from the I/O processor. These pulses are fed into SYNC of the M104 and immediately set the REQ flip-flop. The REQ flip-flop can be monitored through pins J2, U2. The I/O processor responds to a request with a GRANT, and ENA is set. This flip-flop is usually used to gate any address information onto the bus; e.g., the API trap address or the word count address of the multicycle data break. Finally, the next SYNC pulse sets ENB.

The REQ flag can be reset through pin F2, called CLR RQ, by the controller logic. Pin N1 should be tied to power clear or its equivalent.

Finally, the enabling level ENABLE IN will hold REQ off if it arrives as a negative level. When REQ is set (if ENABLE IN is positive), then ENABLE OUT will go negative and the next peripheral on the bus will receive it as a negative ENABLE IN. In this way, the M104 establishes priorities among devices on the same API level or among these that use the data channel. A timing diagram is given in Figure 2-5 for the M104.

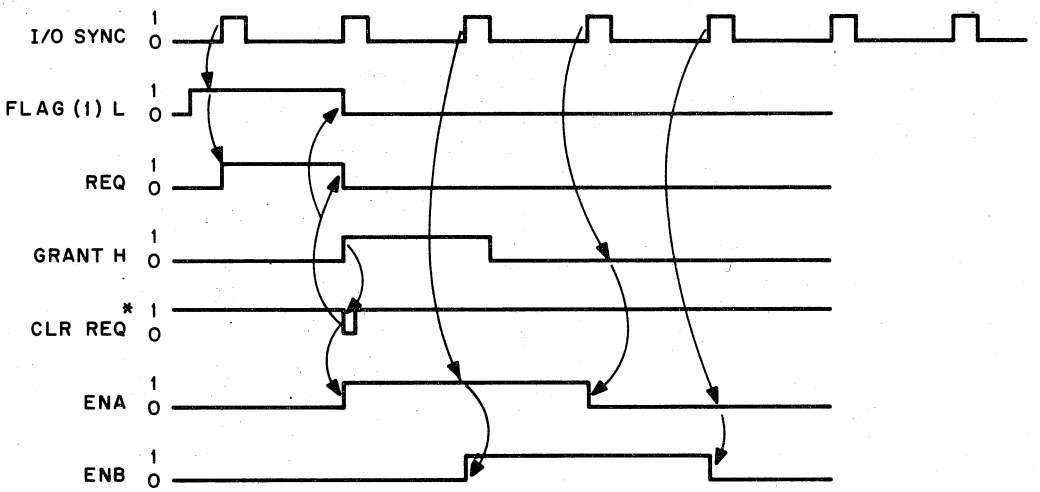


Figure 2-5 M104 Timing

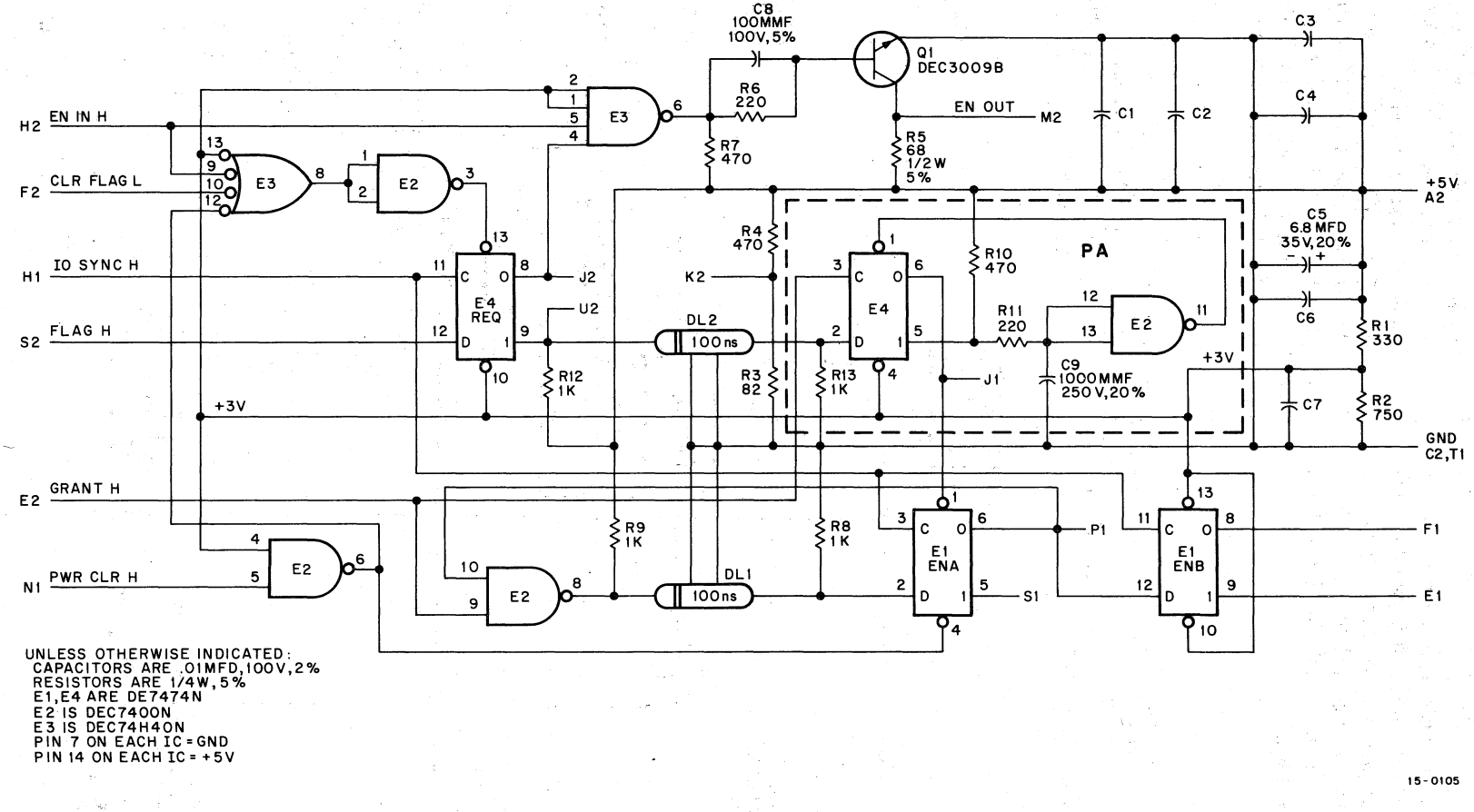


Figure 2-6 M104 Circuit Schematic

2.3.2 M194 Multiplexer Module

This is an M-series single-height module which contains a single multiplexer subsystem. It is pin-compatible with the M104 module. Inputs and outputs have identical loading and driving capabilities as the M104 module with the exceptions of pins H2 and M2. Power dissipation is 1 watt.

Application:

The M194 module has been designed for PDP-15 controllers which are to be used on the PDP-9. It is used for exactly the same reasons as the M104. Pins H2 and M2 are level converted from PDP-9 I/O bus levels (pin H2) to TTL, and then back to PDP-9 levels at M2. All other inputs and outputs are identical to those of the M104. Figure 2-7. shows the M194 equivalent circuit.

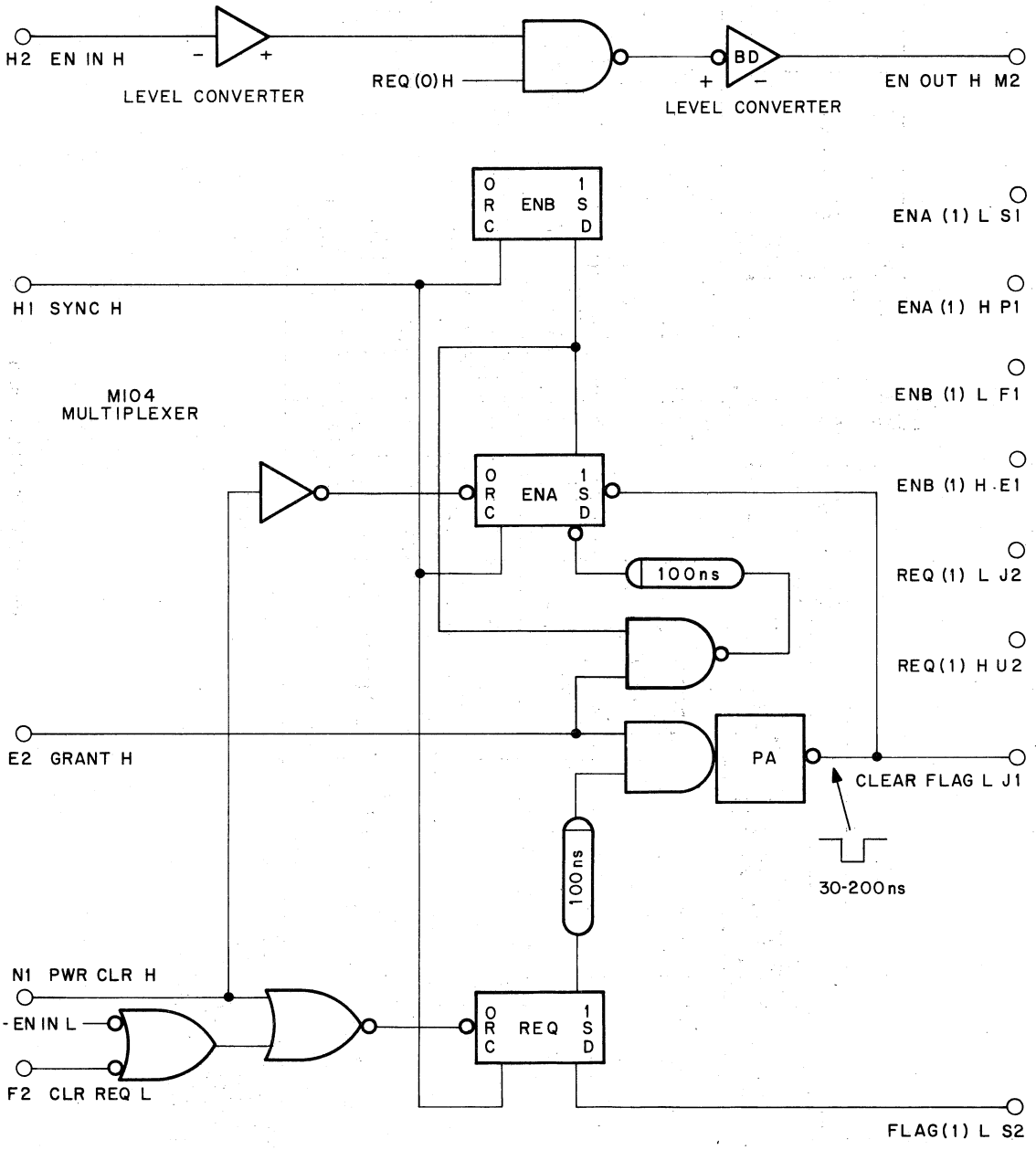


Figure 2-7 M194 Equivalent Circuit

2.3.3 M500 Negative Receiver Module

This is an M-series single-height module containing eight I/O bus receivers which can accept negative logic levels and convert them to positive levels.

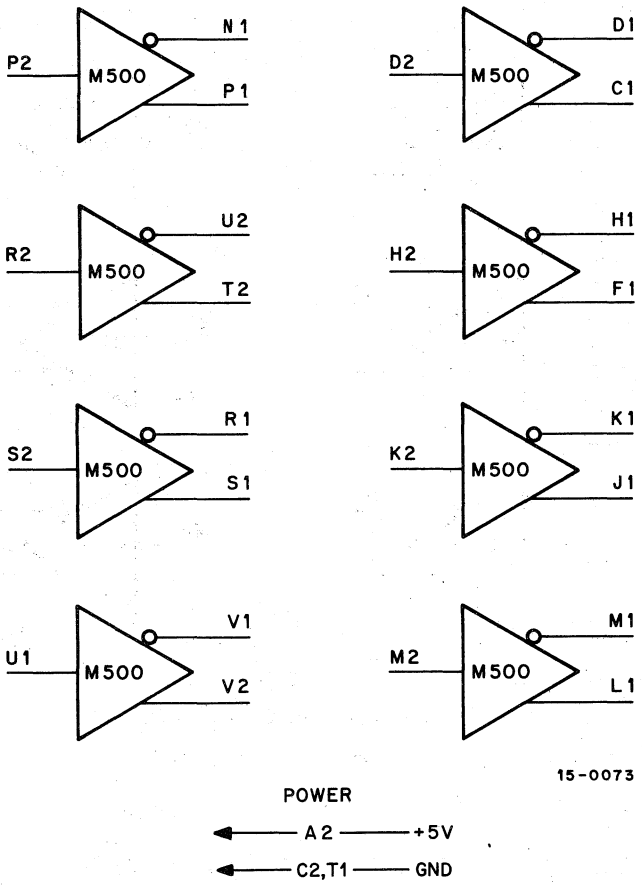


Figure 2-8 M500 Receiver Module

Each M500 receiver has a negative input clamped to 0 and -3V. The threshold switching level is -1.5V with an input current of 100 μA.

Inputs: Minimum input impedance at 0V: 30 kΩ
Maximum current load to bus: 100 μA
Inputs are standard negative logic levels of 0 and -3V.

Outputs: Fan Out Output #1: 12 units
Output #2: 11 units
Input/Output #1 delay: 50 ns
Input/Output #2 delay: 40 ns

Outputs are standard TTL logic levels.

Power Dissipation: 750 mW max from -15V
800 mW max from +5V

Application: The M500 module was designed to receive PDP-9 I/O bus signals for devices using positive logic. It provides a high input impedance.

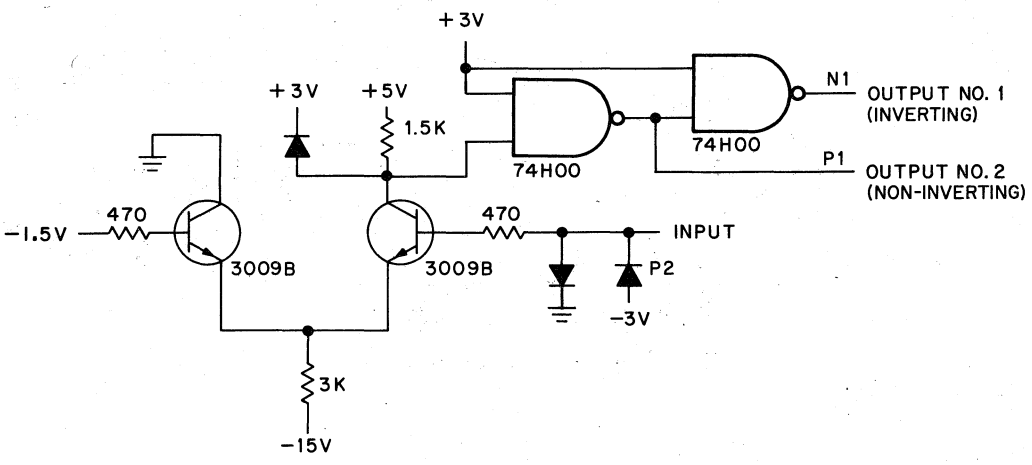


Figure 2-9 M500 Schematic

2.3.4 M510 Positive Receiver Module

The M510 module is an M-series single-height module containing eight PDP-15 I/O bus receivers.

The receiver circuit consists of a two-stage emitter follower with two TTL output buffer gates to supply both inverted and non-inverted outputs.

Inputs: Minimum input impedance: 22.5 kΩ
Maximum current load to bus: 100 μA
Inputs are standard PDP-15 I/O Positive Bus levels.

Power Dissipation: 900 mW

Outputs: Fan Out Output #1: 10 units
Output #2: 12 units
Input/Output #1 delay maximum: 50 ns
Input/Output #2 delay maximum: 60 ns
Outputs are standard M-series levels.

Application: The M510 module was designed to receive PDP-15 I/O bus signals for devices using positive logic. It provides a high input impedance which yields a switching threshold between the high and low levels of the propagated signals. This feature reduces loading and noise problems.

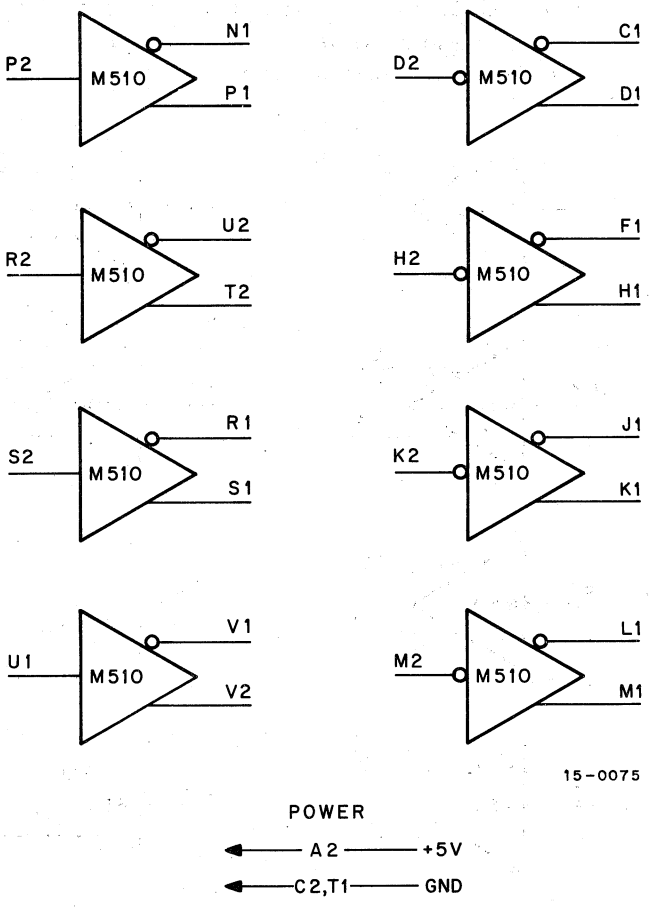


Figure 2-10 M510 Receiver Module

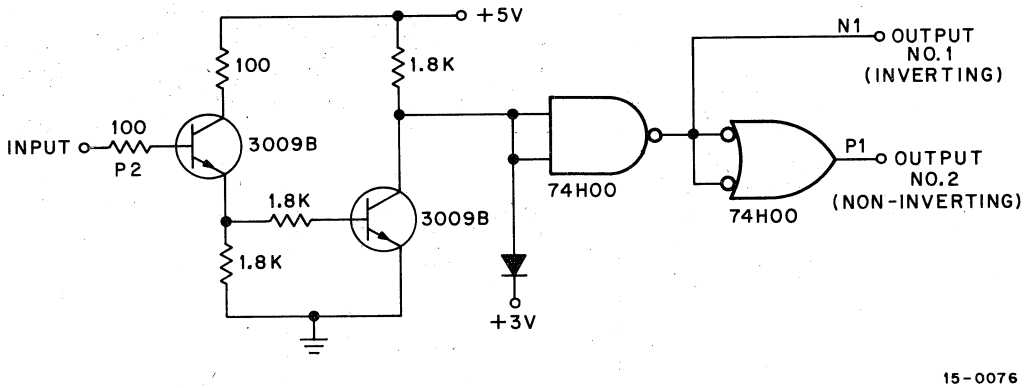


Figure 2-11 M510 Schematic

2.3.5 M622 Positive Driver Module

This is an M-series single-height module containing eight positive bus drivers.

Each driver consists of an AND/OR integrated circuit gate and a discrete component open-collector driver.

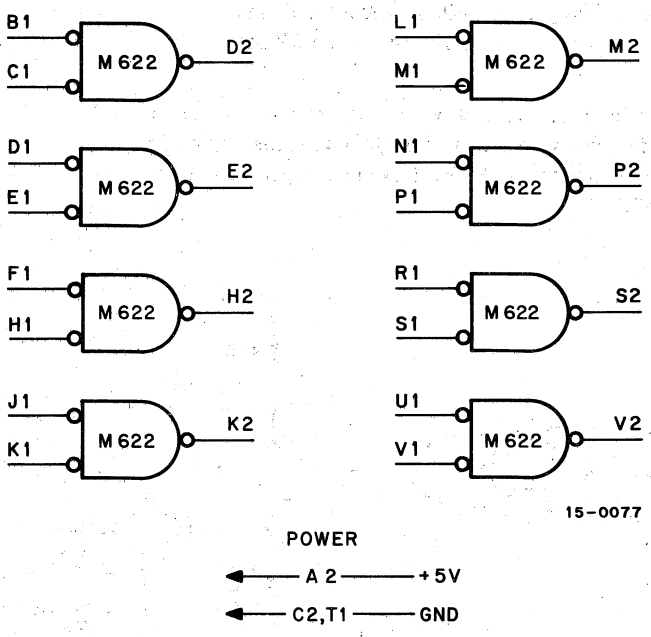


Figure 2-12 M622 Driver

Inputs: Inputs are standard TTL voltages. The input load at 0V is 1-1/4 units.

Outputs: Outputs are standard PDP-15 positive I/O bus signals.

Risetime: 15 ns at the input to the cable.

Current sink = 100 mA max. at V_{ce} sat= 0.4V

Falltime: 10 ns at the input to the cable.

Input – Output Delay = 30 ns max

Power Dissipation: 1.05W max from +5V

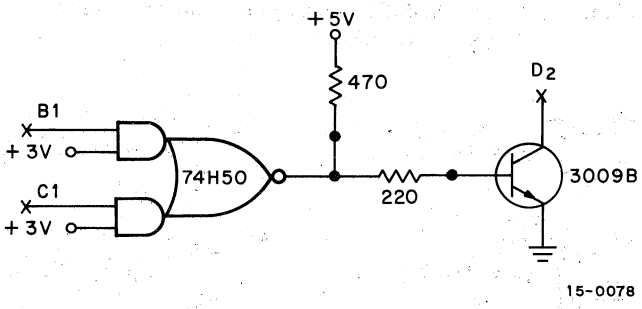


Figure 2-13 M622 Schematic

Application: The M622 module was designed specifically to drive PDP-15 I/O bus signals for devices which use positive logic modules.

2.3.6 M632 Negative Driver Module

The M632 is an M-series single-height module containing eight driver circuits. It accepts positive logic signals and converts them to negative logic levels.

Each driver consists of a TTL input gate and a negative open-collector output driver clamped to ground and -3V.

Inputs: Standard TTL levels - input current load at 0V is 1-1/4 units.

Outputs: Outputs are standard negative logic levels.

Risetime: 15 ns

Falltime: 15 ns with 1.5kΩ to -15V at output

Input – Output Delay = 50 ns max

Power Dissipation: 600 mW from -15V max
900 mW from + 5V max

Application: This driver is used to convert positive logic signals to negative logic levels that drive the PDP-9 negative I/O bus. It is pin-compatible with the M622.

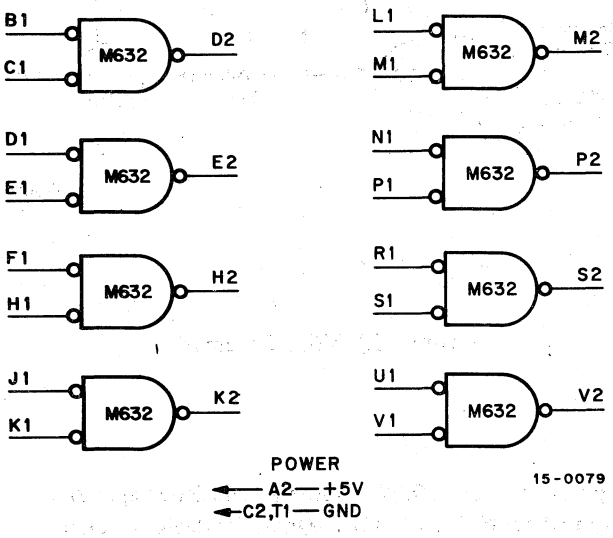


Figure 2-14 M632 Driver Module

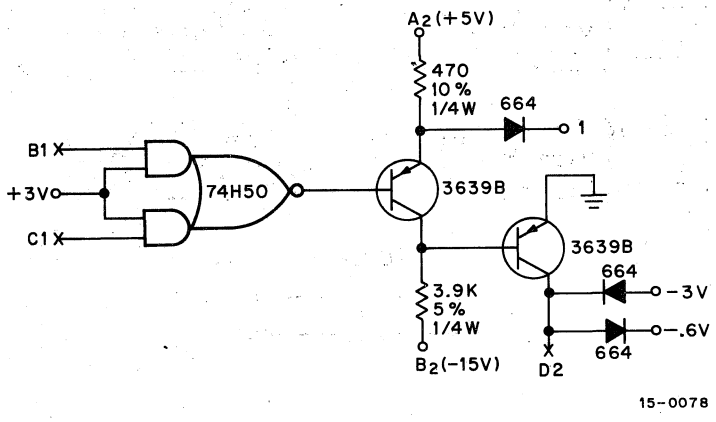


Figure 2-15 M632 Schematic

2.3.7 M909 Terminator Card

This is a standard single-height M-series board with 18 terminating resistors of 68Ω each.

All resistors are 68-1/4W 5%

GND Pins are C2 F2 J2 L2 N2 R2 U2 A1 C1 F1 K1 N1 R1 T1

Inputs: There are 18 inputs - one to each resistor.

Outputs: There are no outputs.

Power Dissipation: 1.8W max

Application: These boards should replace the output cable of the last peripheral on the positive PDP-15 I/O bus.

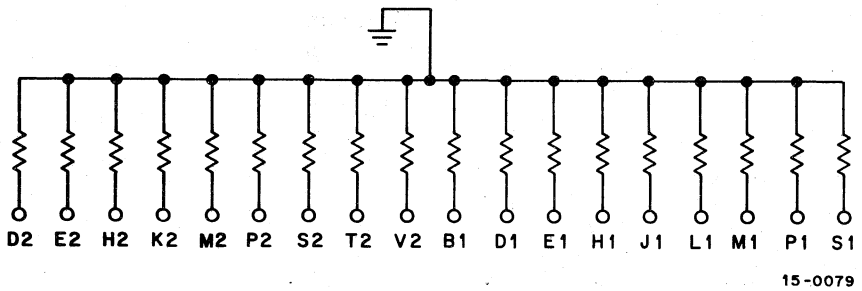


Figure 2-16 M909 Terminator Card

CHAPTER 3
INTRODUCTION TO PDP-15 INTERFACING

3.1 GENERAL PRINCIPLES

The input/output processor of the PDP-15 computer is designed to handle the information flow between the central processor, or its memory and a wide range of peripheral devices. The nature of these peripheral devices varies widely. Some, such as the typewriter, are very slow and require, or give up information at correspondingly slow rates. Others, such as the high speed disk, deal with data at a rate of millions of bits in a second. Although the information needs of such diverse devices vary, their information can be classified into three general categories: commands, data, and status. The nature of each will be illustrated with the following example.

The DECtape system is a low speed 10 (ten) track digital tape recording peripheral. A single central control handles up to eight tape transports. A transport can read or write digital information in either direction of tape movement in one of two modes, normal and continuous. It becomes immediately apparent that the computer must specify a number of DECtape parameters before beginning to read or write its data. These parameters include: the tape transport to be used, where on the tape, in which direction and in what mode the data is to be written. The information written is data and the result of the operation - such as the nature of any detected errors, (did it run out of tape) or special status functions - is called status. The computer commands DECtape to read or write data and report its status.

3.2 DATA INFORMATION

Of the three kinds of information transfers that take place in the I/O processor (commands, data and status), the command and status consist typically of one transfer each. The data transfer is usually more complex, for it involves N transfers, where N is defined by the program.

In a typical transfer of N words, the program specifies a storage (starting) address, and a count of the number of words, N, to be transferred. The storage area thus defined is referred to as a data buffer. During the transfer, the I/O processor hardware or device hardware keep track of the current address and data count until all the desired words have been transferred into the buffer. When I/O processor hardware controls the address and count, two locations in core memory are assigned to the device for this purpose. If the device is designed to manage the count and addressing it uses two of its own hardware registers. The PDP-15 has facilities for either I/O processor controlled transfers called multi cycle data breaks, or device controlled transfers called single cycle transfers.

Consider an example of an output transfer. The I/O channel takes data out of the buffer (located in core memory) in ascending sequential order and passes it on to the peripheral (e.g., a disk) in the same order. Two arithmetic operations are necessary for each transfer to the peripheral. After reading data from storage, the I/O processor (for a multi-cycle transfer) or the peripheral (for a single-cycle transfer) add a constant of +1 to the register which specifies the current address, and then add a constant of -1 to the register which specifies the word count. When the word count equals zero, the transfers are stopped. (In practice, the word count register is loaded with the 2's complement of the count, and a +1 is added until overflow.)

Input transfers are similar except data is written into storage rather than read from storage.

Some peripherals which transfer data do so at such slow rates that the high speed capability of the multi cycle or single cycle data channels is unnecessary. Examples of slow devices such as this are Teletypes and oscilloscope controls. Devices in this class "farm out" their I/O arithmetic to the central processor. This delegation of responsibility slows down the execution of the main program, however, since the program now has to compete with the peripheral for the services of the arithmetic section and the core memory where the data is stored. However, the time lost is tolerable and the hardware design of the device's interface greatly simplified. Such transfers are called program controlled transfers.

3.3 COMMAND AND STATUS INFORMATION

Command information, which was used to select the DECtape transport, the direction of data transfer and the required operation in our example, also occurs infrequently, one word at a time. This type of transfer is carried out by the PDP-15 in the same way as program controlled transfers. Usually, the program or operating system determines the commands, selects the device and, through the central processor, transfers its information to the device's command register for interpretation.

A status information transfer is the reverse. The device requests software intervention at the end of one of its operations, and the program reads a status word into the central processor under program control. The word is then analyzed by the program and further commands are issued.

3.4 INTERRUPTS

Another type of command information referenced in our examples, but not defined, pertains to the ability of a device to request software intervention after it completes an operation and needs further direction. This is a way of interrupting the program from its normal routine, and forcing it to service the device. Again, the program is slowed down while it identifies the requesting device and services it by transferring data under program control or by initializing the data channel for a more complex transfer. In order to reduce the identification period, elaborate program interrupt (PI) and automatic priority interrupt (API) hardware schemes have been devised for the PDP-15 I/O processor. These facilities allow the most urgent requests to be serviced first and as quickly as possible. They are described in detail in the PDP-15 User's Handbook.

3.5 THE PDP-15 I/O BUS SYSTEM

There are two types of bus systems in use on current digital computers (Figure 3-1): the radial system where each peripheral is connected to a physically separate cable or channel and the program identifies the device by a number on that channel; and the parallel system where one I/O cable goes to all peripherals. In the parallel system the cable leaves the I/O processor and is chained to each device on the channel in turn. This distributes the cable connections along the bus and therefore reduces cable congestion at the processor. There is one set of line drivers and receivers in the processor instead of one set per device.

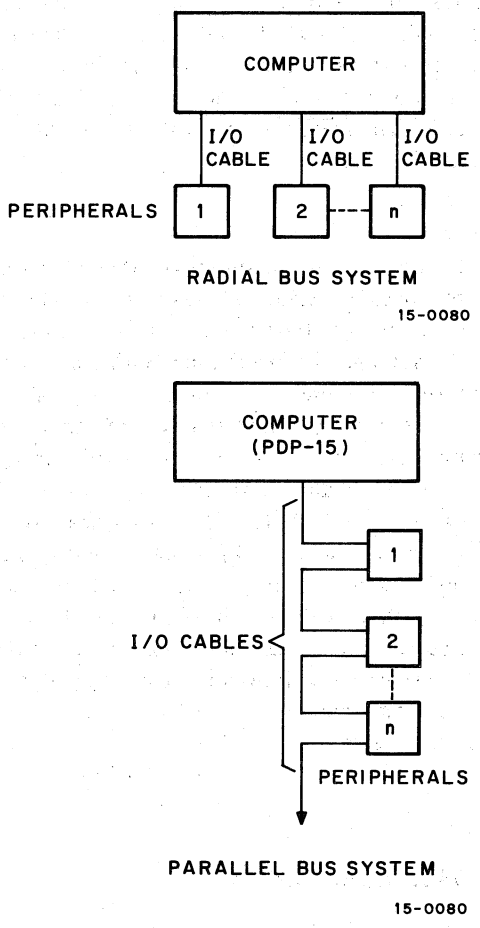


Figure 3-1 I/O Bus Schemes

The PDP-15 uses the parallel bus system where one cable is chained from device to device. Commands (other than program interrupt or automatic priority interrupt) and status are transmitted on the same lines as data, and these lines are used for both program control and data channel transfers. Data, command and status words can be up to 18 bits.

A set of control lines accompanies the data lines to identify the activity on the lines. The control lines identify data channel transfers, program controlled transfers, status or command instructions, program interrupt or API commands or miscellaneous operations such as a pulse to clear all registers prior to set up. There are a total of 72 signals in the cable.

All devices on the I/O bus can be operated concurrently by means of a time-shared multiplexing arrangement wherein the data channel and API automatically interleave services to the various subsystems; and the operating program is designed to interleave services during program-controlled data, command or status transfers.

Essential to this multiplexing system is the idea of a device number or device address. During multi-cycle data channel or API operations, the peripheral transmits its device number to the I/O processor so that the program can address the data buffer appropriate to that device. For a transfer operation, the device number specifies the locations of the word count and current address. For an API request, this number specifies the location that the program must "trap" or jump to. During single-cycle data transfers, the device specifies the core address where it wishes to transfer the current word.

A device number is also used during program-controlled command status and data transfers. In this case, the number is selected by the program through an instruction called an IOT (input/output instruction). Each device is assigned a group of such instructions which include the device number. During the execution of the instruction, the device number is transmitted to each device. Only the correct device will decode the number and carry out the intended transfer; all other devices remain inactive. This facility is called the addressable I/O bus.

All signals on the cable, except a few, which are called ENABLE signals, are broadcast to all devices in parallel. Each device passes on the signal it receives without amplification, and in a similar fashion all input signals are passed along, and the I/O processor sees a composite of all. Units that are not active transmit logical zeroes which do not cause interference between devices.

The ENABLE signal is not always passed on; it is the key to the automatic multiplexing scheme used by the data channel and API. This signal is wired in series through each device. When a device receives the ENABLE signal on API or data channel it decides whether or not it wants to communicate with the I/O processor. If communication is desired it traps ENABLE, preventing the signal from passing to the next device on the chain. If no communication is desired it merely passes the signal on as it would any other outbound signal.

3.6 SUMMARY OF THE PDP-15 I/O PROCESSOR

The PDP-15 I/O processor contains two basic subsystems; a CPU I/O controller* and a memory I/O controller*. The former handles all information transfers between the I/O bus and the CPU; this includes any program interrupts on PI or API, all program controlled transfers of data, command or status information and miscellaneous instructions such as CAF (clear all flags).

The memory I/O controller carries out the single- or multi-cycle data channel transfers of data, and includes a facility to either increment the content of a device specified location or add a number to its contents. Data in 18-bit words can move in or out of memory at speeds of 1 mHz during single-cycle transfers; 250 kHz during multi-cycle transfers.

3.6.1 The CPU I/O Controller

The PDP-15 CPU I/O controller handles all control and data transfers between the central processor and devices on the I/O bus. It uses the following basic system elements:

a. *An addressable I/O bus* which includes the shared 18-line data path, 3 IOP control signals, eight device-code lines and a line which specifies an input transfer called RD RQ L.

*Note that these subsystems are functional, and do not have any real physical location.

b. *A program interrupt request line* (PROG INT RQ L) used by the device to request program intervention. When any device selects this line, the central processor automatically traps to location 0 of bank 0, stores away important current registers under program control and proceeds to execute the contents of location 1 of bank 0.

c. *Skip and status lines* which provide a way for the computer to identify a device which causes a program interrupt. This is important since all devices share the same program interrupt line. Having trapped to location 0 on a program interrupt, the program usually jumps to a routine which issues successive IOT SKIP instructions, one for each device. The device which echoes back a pulse on its SKIP RQ L line is identified as the requesting device. The echoed pulse causes the program to skip its next instruction which usually sends the program to that device's service routine.

   The status facility of the PDP-15 issues a pulse called RD STATUS H. This is used by a device to gate its status flags onto the I/O bus where they are read into the central processor. This signal can be issued by either an IORS command or by a switch on the console (see PDP-15 User's Handbook).

d. *Automatic Priority Interrupt (API) Logic* extends the PDP-15 program interrupt capabilities by providing interrupt servicing for as many as 28 I/O devices with minimum programming and maximum efficiency. Its priority structure permits high data-rate devices to interrupt the service routines of slower devices with a minimum of system "overhead". The option permits the device service routines to access directly from hardware-generated entry points, eliminating the need for time-consuming flag searches to identify the device that is causing the interrupt.

   The option provides 32 unique channels, or entry points, for the device service routines, and 8 levels of priority. The four higher levels are for fast access to service routines in response to device-initiated service requests. Each of these levels can be multiplexed to handle up to 8 devices assigned an equal priority, up to a maximum of 28. The four lower levels are assigned to program-initiated software routines for transferring control to programs or subroutines on a priority basis. Four of the 32 channels are reserved for these software levels.

   Each device interfaced to the API option specifies (sends) its "trap address" or unique service routine entry point to the processor when granted an API break by the processor. Core memory locations 40_8 through 77_8 are assigned as these entry points. JMS or JMS I instructions contained in these locations provide linkage to the actual service routines.

   Of the 28 hardware channels, 3 are assigned internally to the paper-tape reader, real-time clock, and optional power-failure detection system.

   Each API priority takes precedence over lower API priorities, program interrupts, and the main program. The highest priority program segment interrupts lower priority program segments when activated. The data channel and real-time clock hold highest priority.

   The entire API system may be enabled or disabled by a single IOT instruction.

   The I/O bus contains 12 lines unique to the API; these include an API RQ L (request), an API GR H (grant) and an API EN H (enable) line for each of the four levels. Other I/O bus signals used by the API include I/O PWR CLR H, I/O SYNC H, and I/O ADDR 12 L-I/O ADDR 17 L. The API RQ L lines are used by the device to request an interrupt from the computer at a particular priority level. The API GR is the computer's response to an API RQ L. The API EN signal indicates to a device the status of another device on that interrupt level. (There can be as many as eight devices sharing any priority level.) I/O PWR CLR H is used to establish initial conditions, I/O SYNC H is used for timing, and the I/O ADDR 12 L to I/O ADDR 17 L specify the unique entry point or trap address to a device service routine.

3.6.2 The Memory I/O Controller

The PDP-15 memory I/O controller handles all control and data transfers between the internal core memory and peripheral devices. It uses the following basic elements to do this:

a. *The Addressable I/O Bus* which includes the shared 18-line data path, two IOP control signals (IOP 2 H and IOP 4 H) and two signals signifying the direction of a transfer; RD RQ L for a read into memory and WR RQ L for a write out of memory.

b. *The Multi-Cycle Data Channel Control Lines* which include a request line – DCH RQ L; a command to the device to post its address code – DCH GR H, and an ENABLE line – DCH EN H. The multi-cycle facility will transfer data to or from memory at speeds of up to 250 kHz in 18-bit words along 18-line data paths. The control lines also include 15 address lines to specify the address in memory where the word count is stored.

c. *The Single Cycle Data Channel Control Lines* which include the control signals used by the multi-cycle data channel, plus one other to signify a single-cycle request as opposed to a multi-cycle request. This is called SING CY RQ L. A device can operate with this facility in one of two modes: burst mode or normal mode. Burst mode allows a device to carry on back to back transfers at 1 mHz in one direction only. If the device needs to either change direction or slow down, it must drop back into normal mode which requires resynchronization at each transfer and thus has a 3 to 5 μs latency.

d. *An Increment Memory Control Line* which, when enabled, causes the I/O processor to increment the location normally specified in a multi-cycle data channel request as the word count. However, the cycle stops here. The current address is not incremented and no data is transferred. If the count overflows, the device is notified.

e. *An Add to Memory Control Line* which causes the I/O processor to add the contents of the 18 data lines set up by the device on the I/O address lines to the contents of a memory location specified by the device. Both words must be of the same sign. If the sign of the sum is different, then an overflow pulse is sent to the device.

Table 3-1
Summary of PDP-15 Input/Output Facilities

Facility	Remarks
	Data Transfers To/From Memory
Multi-Cycle Data Channel Input	Used to transfer data directly to core memory in up to 18-bit bytes at high speed (250 kHz). Cost is low but memory overhead is high.
Multi-Cycle Data Channel Output	Used to transfer data directly from memory in up to 18-bit bytes. Cost is low and memory overhead is high. Maximum speed is 188 kHz.
Add to Memory	Used to add the contents of a device register to the contents of a specified core location in 18-bit bytes. Good for signal averaging. Maximum speed is 188 kHz.
Increment Memory	This facility allows an external device to increment the content of a core location by 1. Useful for generating histograms. Maximum speed is 500 kHz.
Single-Cycle Data Channel Output	With this facility a device can transfer a burst of data from core memory at 1 mHz in 18-bit bytes. It is expensive to interface to and should be used for very high speed devices.
Single-Cycle Data Channel Input	Used to transfer a burst of data from a device to core memory at 1 mHz per 18-bit word. It is expensive to interface to and should be used for very high speed devices.
	Data Transfers To/From CPU
Addressable I/O Bus	With this facility, devices can transfer data in 18-bit bytes to or from the central processor. Cost of interfacing is minimal. A typical transfer rate is one transfer every 200 μs.
	Command and Status Transfers
Addressable I/O Bus	Command and status information can be transferred to or from the CPU in the same manner as ordinary data.
Read Status	This is a special facility designed to allow the user to monitor all vital flags in the system. Each device is assigned a bit for its flag(s), which is read onto the addressable I/O bus and into the CPU when the Read Status command is given. No two devices should use the same bit. See the PDP-15 User's Handbook for bit assignments.
Skip	The addressable I/O bus allows the computer to test the status of a flag (typically) by issuing a pulse which will echo if the addressed flag is up. Every flag that posts a program interrupt must be identifiable by the skip facility.
	Interrupts
Program Interrupt	All devices share a common program interrupt line. When a device posts an interrupt the computer is forced to location 0, bank 0, and then on to a service routine designed to identify the requesting device using the skip facility. The process requires CPU and memory overhead and takes time.
Automatic Priority Interrupt	This facility reduces the time to service a requesting device and establishes a hierarchy among devices so that important interrupts can be handled quickly and without interference.

CHAPTER 4
PDP-15 INTERFACING

4.1 INTERFACING TO THE CPU I/O CONTROLLER

In general, devices rely on the CPU for control and status information, and on memory for high speed data. An extensive monitor system running in the CPU usually commands the operation of the entire computer complex. It can initiate a peripheral such as a disk into the correct operating mode and start it when the entire system is ready. The transfer rates for these commands are low, so that the CPU can afford to provide much of the basic control logic for such transfers, thus simplifying device logic. It does this with an instruction set called IOT instructions (Figure 4-1). Control data or status information is transferred to or from the accumulator of the CPU from the I/O processor under IOT control. The I/O processor passes the data on or off the I/O bus cable to or from the selected device.

Any device which is to interface to the I/O processor for CPU communications must satisfy the needs of four elements:

a. Addressable I/O Bus

b. Program Interrupt Facility

c. Status and Skip Facilities

d. Automatic Priority Interrupt System.

Although API is a system option, any device designed for the I/O bus should always contain API logic to allow for future expansion.

4.1.1 A CPU Interface with API

As an example of an interface for a device to the CPU I/O controller, a very simple peripheral is chosen that can receive into its data buffer an 18-bit word from the accumulator of the CPU, or can transmit the states of its 18 contacts back to the accumulator (Figure 4-2).

This example has five basic components: addressable I/O bus logic, data register logic, contact sensor and transmitter logic, program interrupt skip and status logic, and API control logic.

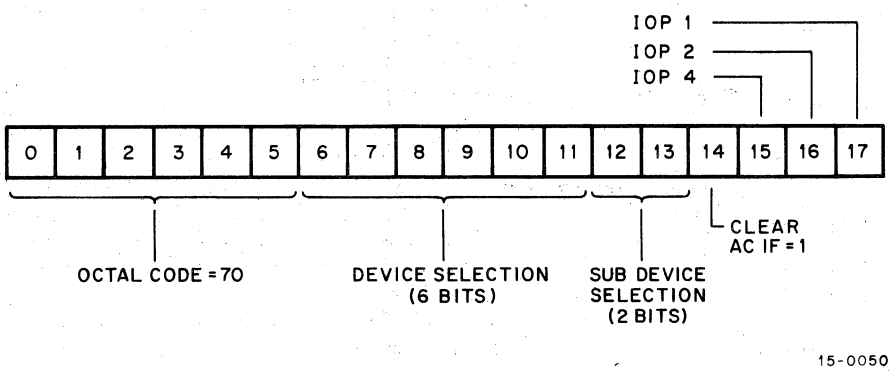


Figure 4-1 IOT Instruction Format

a. *The Addressable I/O Bus Logic* – decodes the unique device and subdevice code which originates in the IOT instruction. This device can now respond to the IOP pulses 1, 2 or 4 issued by the same IOT to transfer data or cause some control function such as "skip on flag".

The addressable I/O bus logic also contains input buffers for the 18 data lines on which data is transferred.

b. *Data Register Logic* – This 18-bit register accepts data from the addressable I/O bus. The data is strobed off the bus with one of the IOP control pulses issued by the IOT instruction.

c. *Contact Sensor and Transmitter Logic* – 18 switches are sensed by the transmitter logic, and their contents can be transferred to the 18 data lines of the addressable I/O bus logic. An IOP-2 control pulse is used to strobe the data onto the bus.

d. *Program Interrupt Skip and Status Logic* – The device notifies the computer of its need for CPU attention by posting a program interrupt. All devices share the same program interrupt line so, to determine which device caused the signal, the CPU goes through a polling routine or "skip" chain (refer to PDP-15 User's Handbook) where it issues a skip IOT for each device. The device which echoes back a SKIP RQ pulse to the processor is identified as the requesting device.

The status logic is used to notify the CPU of the status of each device. By issuing an IORS (read flags) instruction, the CPU can monitor the system status. IORS issues a pulse at IOP-2 time.

Each device should be designed to strobe the "OR" of all its interrupt flags onto a unique pre-assigned I/O bus line when the IORS IOT is issued. IORS is similar to a read IOT in that the data strobed onto the bus is loaded into the accumulator. Since all devices receive IORS simultaneously the I/O bus will see a composite of all device flags, and therefore the status of the entire computer system.

This status facility is useful for checking interrupts when API is not available on the system.

e. *The Automatic Priority Interrupt Logic* – There are four API levels assigned to peripherals on the I/O bus cable. They extend the capabilities of the program interrupt facility and save CPU time for polling. An API request at any level forces the CPU to jump or "trap" to an address specified by the device. In this way, the device is immediately identified and polling time is saved. The API logic facilitates this option for the device, provided the option is available in the I/O processor.

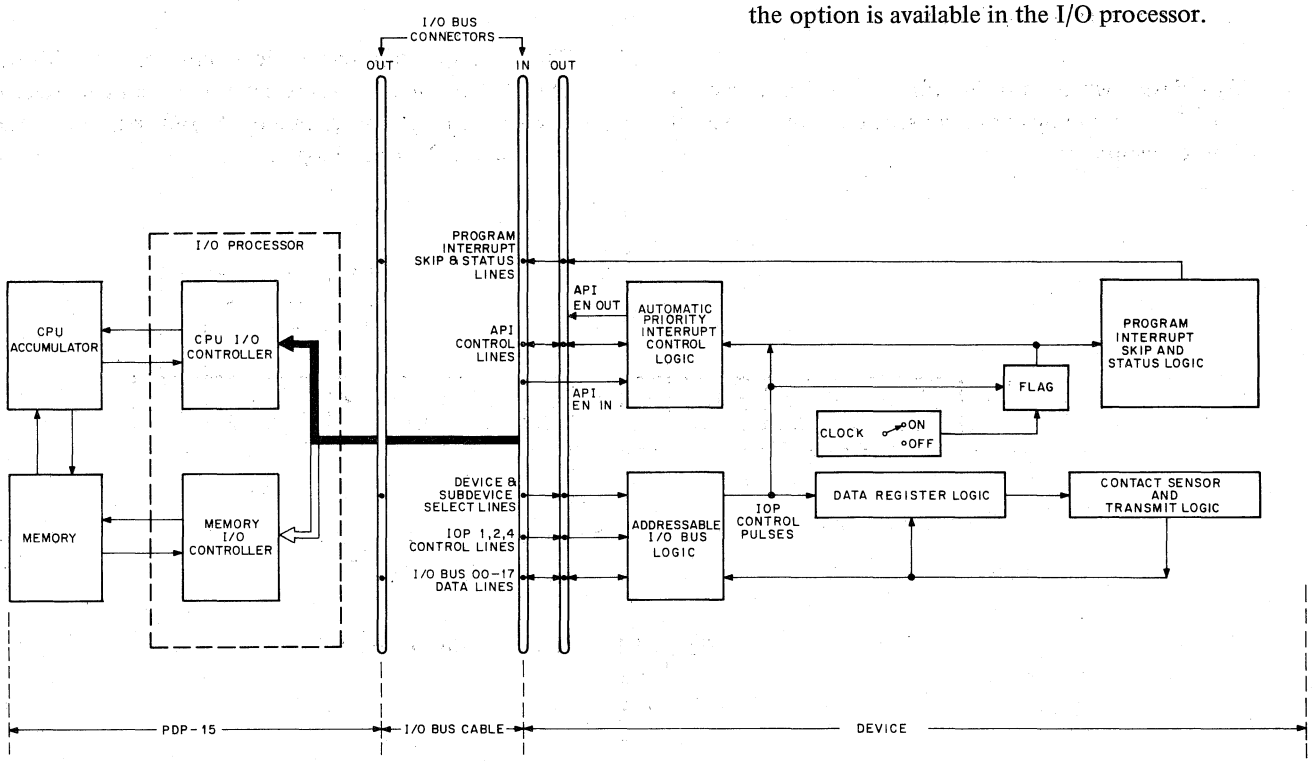


Figure 4-2 Block Diagram of CPU Device with API

4.1.2 The Logic

Figure 4-3 shows the detailed logic of each of the blocks in Figure 4-2. The following descriptions relate directly to this diagram.

This peripheral can be programmed to either read a data word from the contact sensor switches or write a word from the accumulator into the data buffer according to the whim of its programmer. A clock is provided to post interrupts or API requests at regular intervals, to which the program can respond by issuing read or write IOT's.

This device has no known practical application. It is used here because it illustrates the logic and timing necessary to mate more complex devices to the PDP-15, yet is relatively easy to understand.

The device works in the following way:

a. The contact sensors are set to whatever word is to be read.

b. The programmer selects his read or write subroutine.

c. The M401 clock is enabled.

d. The clock sets FLAG which enables the API REQ flip-flop of the M104 Multiplexer. On the next I/O SYNC H, an API 0 RQ L and a PROG INT RQ L signal are issued simultaneously. If API is available and enabled, then the CPU will trap to memory location 44. If API is not available, then a program interrupt occurs and the CPU traps to location 00. Each of the signals used by the API control logic is listed in Table 4-1, and the timing is given in Figure 4-4.

e. If the CPU responded to an API break, then it is programmed to jump to its appropriate read or write routine.

f. If the CPU could only respond to a program interrupt request (because API was unavailable or disabled), it would jump to a polling routine where it would issue the IOT SKIP instruction. When this IOT selected the device and issued IOP1, then IOT SKIP would be generated and an echo pulse called SKIP RQ L would be sent back to the computer. The CPU would be forced to skip its next instruction, and the program, recognizing this device, could jump to the read or write routine.

g. If the programmer wants to read in, then the program would issue an IOT with IOP2 selected. The IOT is decoded by the addressable I/O bus logic and an IOT READ signal strobes the contact switches onto the data lines, I/O BUS 00L to I/O BUS 17L. It also strobes a signal called RD RQ L which tells the CPU of an IN transfer.

h. If the programmer wants to write, then the program would issue an IOT with IOP4 selected. The IOT is again decoded and an IOT WRITE signal generated. This pulse strobes the data from the accumulator to the data register of this device.

i. When a transfer has been completed, an IOT CLR FLG is issued and both API RQ and FLAG are reset.

Detailed descriptions of each signal and its timing relationships to associated signals are given in Table 4-1 and Figure 4-4. These should be studied in detail before a design is attempted.

The designer should take note of the following points:

a. All I/O bus signals are received by an M510 module and transmitted by an M622 module with the exception of API EN IN and API EN OUT, which are handled by the M104.

b. PROG INT RQ and API N RQ are anded together to generate PROG INT RQ. This technique gives each device the option of using API or PI but guarantees that when API is used, it will be honoured over the interrupt. If PROG INT RQ were not gated with API N RQ, then the I/O processor may see the PI request before the API since the former is asynchronous, and service it first.

c. The logic should be designed so that if the M104 is removed from the device because its corresponding computer does not have the API option, then the PI facility will still operate. This is effected by wiring a pull up resistor of an M113 module to the input of the inverter which drives the corresponding AND gate. This forces a logical one to this gate when the M104 module is pulled.

d. It is good practice to gate the two subdevice bits into the M103 module with the device bits, when the subdevice codes are not used. This frees up remaining codes for other devices. Note that this practice was not followed in our example.

e. Make sure that all flags and registers are cleared by I/O PWR CLR, or the device may post

spurious interrupts during power up. If the device must be powered up while the computer system is on, then an internal power clear signal must be generated to avoid erroneous flagging.

f. Do not clear the interrupt flag(s) with the clear flag (J1) signal in the M104. Clear this flag(s) with an IOT. Otherwise the I/O processor may clear the flag before it acknowledges the request.

The following subroutine illustrates how this device can be programmed. It assumes for simplicity that the device clock is enabled, and that no other programs or devices are operating on the system. If any switch is up on the console switch register, then read operations are performed. Otherwise a write operation is carried out.

```
                .LOC 2000
START           CAF             /Clear all system flags
                CLA             /Clear the accumulator
                ION             /Enable the PI
                TAD (40000      /Set up the API control
                ISA              word and enable API
                JMP.            /Wait loop
                JMP START       /Return to the beginning

                .LOC 0
                000000          /PI setup
                JMP SKIP        /Go to skip chain

                .LOC 44
                JMS DATA        /API trap point

                .LOC 2020
SKIP            IOT SKIP L      /Check the flag
                HLT             /This flag was not
                                 up. Since there are no other
                                 devices operating, something
                                 is wrong.
                JMS DATA        /Go to service routine
                JMP * 0         /Go to beginning

DATA            000000          /Subroutine entry point
                CLA             /Clear accumulator
                LAS             /Load switch register
                SZA             /Skip if AC=0
                JMP READ        /Not zero, go to read
                JMP WRITE       /Zero, go to write

READ            CLA             /Clear the AC
                IOT READ        /Read data in
                DAC TEMP1       /Store it
                DBR             /Debreak and restore
                JMP * DATA      /Return

WRITE           CLA             /Clear the AC
                LAC WORD        /Get the data word
                IOT WRITE       /Write it
                DBR             /Debreak and restore
                JMP * DATA      /Return
```

This page intentionally left blank.

Table 4-1
CPU Interface With API

Signal Number	Signal Mnemonic	Input Cable Pin Number	Signal Definition	Signal Function
1	API 0 EN H (IN and OUT)	2BL1	This enable signal, one of four in the API system, is a dc level originating in the I/O processor and is daisy chained from device to device on the same level. The M104 logic in each controller can interrupt this level, cutting the level off all devices that follow it on the bus. A device receives it as API 0 EN IN H and transmits it as API 0 EN OUT H.	Each device can post a request to its API level only if the incoming API EN level is true. By posting a request, the device immediately inhibits all controllers below it on the bus by removing the enable signal to the next device. In this way, priorities on each level are established when devices request simultaneously; the first device on this level will have highest priority, the second next priority, etc.
2	I/O SYNC H	2AB1	The I/O processor clock pulse issued every microsecond. It is a 1 mHz 250-ns pulse.	This signal is used to synchronize API RQ to the I/O processor.
3	API 0 GR H	2BJ1	One of four possible signals issued by the I/O processor indicating that it grants the API request at the corresponding level, in this case level 0.	The device uses this signal to gate the address of its API level trap address onto the I/O ADDR lines by posting the API ENA flag.
4	I/O PWR CLR H	2AS1	System clear signal generated in response to 1) Power on or off; 2) CAF instruction; 3) I/O RESET key. It is a 1 mHz, 250-ns pulse.	This signal is treated as an initializing signal for all devices (controllers attached to the I/O bus). All registers are reset to "initial" status.
5 →11	SD0 H →DS5 H	2AT2 2AP2	The device and subdevice select lines decoded from I/O bits 6-13 of the IOT instruction.	These signals are decoded by the addressable I/O bus to select the device.
12	IOP 1 H	2AD1	Micro-programmable control signal which is part of an IOT instruction. Decoded from bit 17 of the IOT.	Used for I/O skip instructions to test a device flag or other control function. Cannot be used to read a device buffer register.
13	IOP 2 H	2AE1	Same as IOP 1 except it is decoded from bit 16.	Usually used to effect a transfer of data from a selected device to the processor, or to clear a device register or flag, but may be used for other control functions. May not be used to determine a skip.
14	IOP 4 H	2AH1	Same as IOP 1 H except it is decoded from bit 15.	Usually used to effect transfer of data from the CPU to the device or control. May not be used to determine a skip condition or to effect a transfer of data from a selected device to the CPU.
15→32	I/O BUS 00 L →I/O BUS 17 L	1AB1 1AV2	18 data lines which constitute the bidirectional facility for transferring data in bytes of up to 18 bits between the device and the CPU (in this case).	These data lines (I/O BUS 00 L – I/O BUS 17 L) convey data between the AC of the CPU (via the I/O processor) and the device data register. This is the most significant bit, 00.
33,34	I/O ADDR 12 L, 15 L	1BK2, 1BS2	One of fifteen lines which constiute an input bus for devices that deliver address data to the processor.	This address bus in this case is used to deliver the device's API trap address 44 during its API break.
35	API 0 RQ L	2BH1	One of four API request signals on channels 0-3. This signal is set by the device. The signal is raised by the M104 logic at I/O SYNC time only.	The device uses this signal to inform the I/O processor of its request for API priority level 0, the highest of the four.
36	PROG INT RQ L	2AL1	This signal causes the program to trap to location 000000 if the priority interrupt system is enabled. The instruction resident in location 000001 is fetched and executed.	A device delivers this level to the I/O processor to request interruption of the program in progress in order that the device be serviced by the CPU.
37	SKIP RQ L	2AJ1	The return of the signal to the I/O processor during IOP1 indicates that an IOT instruction test for a skip condition has been satisfied. The PC is subsequently incremented by one.	Used by a device to inform the program of the state of its interrupt flag.
38	RD STATUS H	2AP1	A signal issued when the CPU issues an IORS instruction or when the console switch is placed on I/O STATUS.	Used by the device to gate its status onto the I/O bus data lines (one line per status bit) which is then read into the AC of the CPU.
39	RD RQ L	2AM1	Indicates to the processor that the device is sending it a data word.	Used by the device to specify to the I/O processor an input-to-CPU data transfer is required. Its absence is interpreted as an outgoing transfer.

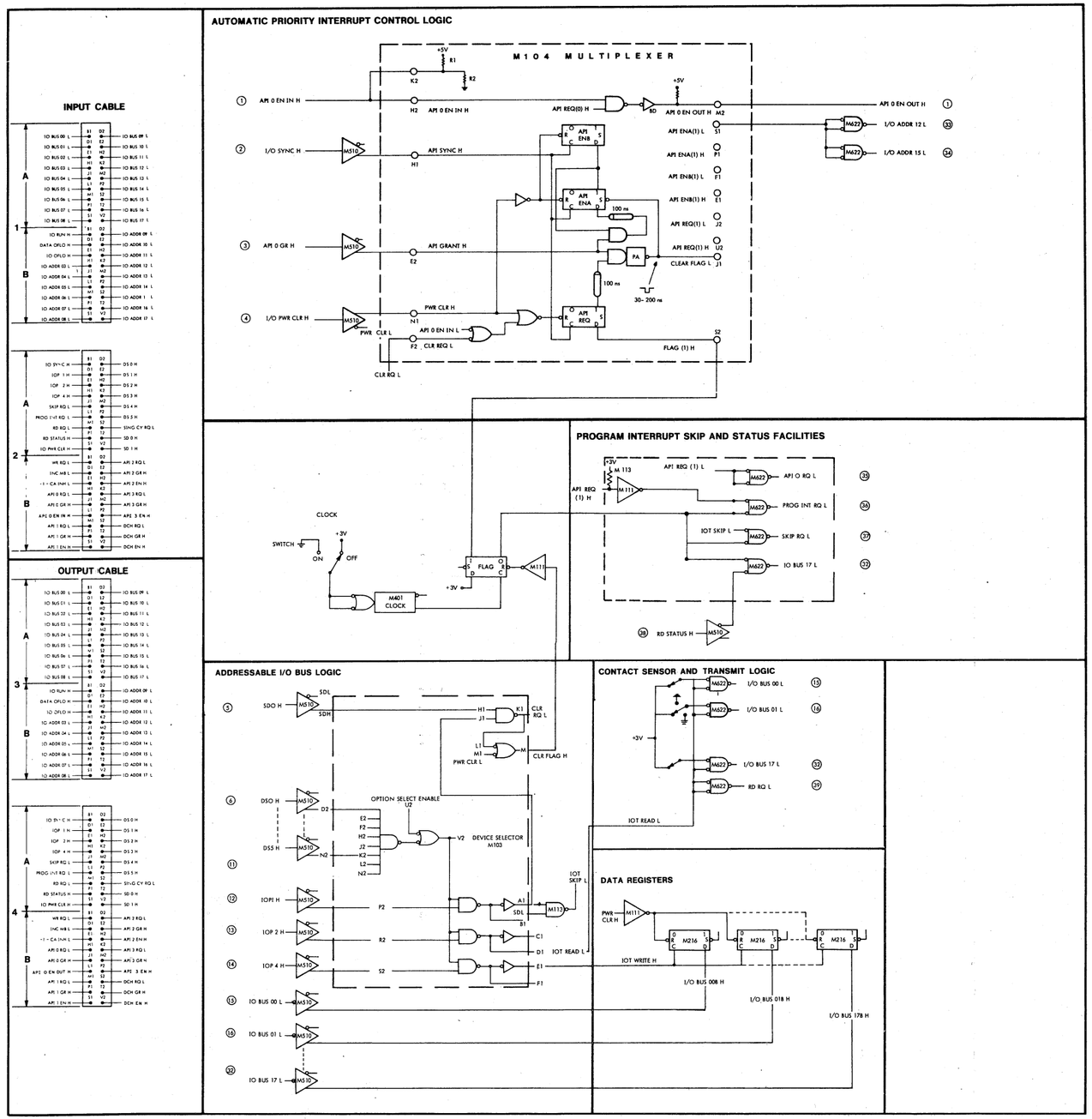


Figure 4-3 Logic Diagram of the CPU Device with API

SIGNAL NO.	SIGNAL NAME	POLARITY (VOLTS)
2	I/O SYNC H	(1) 2.5 (0) 0
	FLAG(1)H	(1) 2.5 (0) 0
34 35	API O RQ L PROG INT RQ L	(0) 2.5 (1) 0
3	API O GR H	(1) 2.5 (0) 0
32 33	API ENA (1) H I/O ADDR 12 &15 L	(1) 2.5 (0) 0
6—11	DS5 H DSO H SDO H	(1) 2.5 (0) 0
12	IOP1H (SKIP RQ L CLR RQ CLR FLAG)	(1) 2.5 (0) 0
13 39	IOP2 H RD RQ L (IOT READ)	(1) 2.5 (0) 0
14	IOP 4 H (IOT WRITE)	(1) 2.5 (0) 0
	STATES OF I/O BUS	

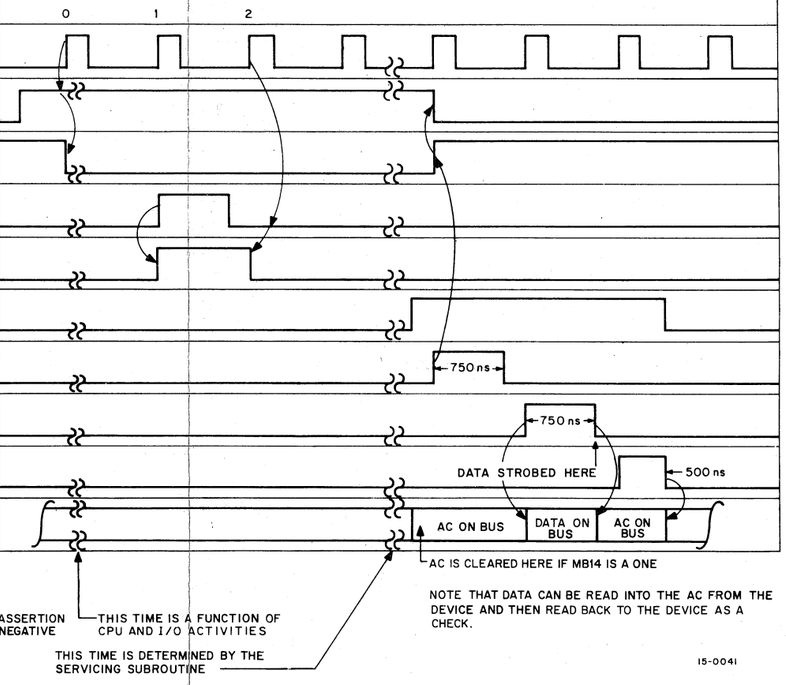


Figure 4-4 Timing Diagram of the CPU Device with API

4.2 INTERFACING TO THE MEMORY I/O CONTROLLER

The primary task of the memory I/O controller is to serve as a high speed data path between memory and storage peripherals such as disks, drums or magnetic tape. It serves this function with two facilities; a single-cycle data channel for very fast (up to 18 x 10^6 bits) transfers or a multi-cycle channel for slower (250 kHz per word) rates. However, the memory I/O controller also provides two other features: an increment memory service with which a peripheral controller can specify any PDP-15 internal memory location and increment it by one at a maximum rate of 500 kHz; and an add-to-memory feature where a peripheral can add the contents of any PDP-15 memory location to the contents of its data register at 188 kHz.

4.2.1 The Multi-Cycle Data Channel Peripheral

Any device controller designed to use the multi-cycle data channel needs six elements: the five used by a device using the CPU I/O controller plus multi-cycle data channel logic (Figure 4-5).

Each of the basic elements described for the CPU I/O controller are used in this device, with only minor differences in the logic.

Multi-Cycle Data Channel Logic is basically a single DEC module Type M104 which handles the basic control and timing signals of the multi-cycle data channel. Each device using the multi-cycle data channel is assigned two memory locations; one for its word count, and one for its current address.

These registers are identified by a device as it requests service by the control logic. Then, the memory I/O controller increments each location during the multi-cycle operation.

Figure 4-6 is a detailed logic diagram of each of the blocks in Figure 4-5. The following is a description of the functions performed by it:

a. The peripheral is initiated manually by:

(1) Setting the MODE switch to READ or WRITE

(2) Setting the data switches

(3) Setting the ENABLE switch to ON

b. When enabled, the M401 clock sets DCH FLAG which in turn enables DCH RQ of the M104. The next I/O SYNC pulse sets DCH RQ which then posts DCH RQ L back to the memory I/O controller.

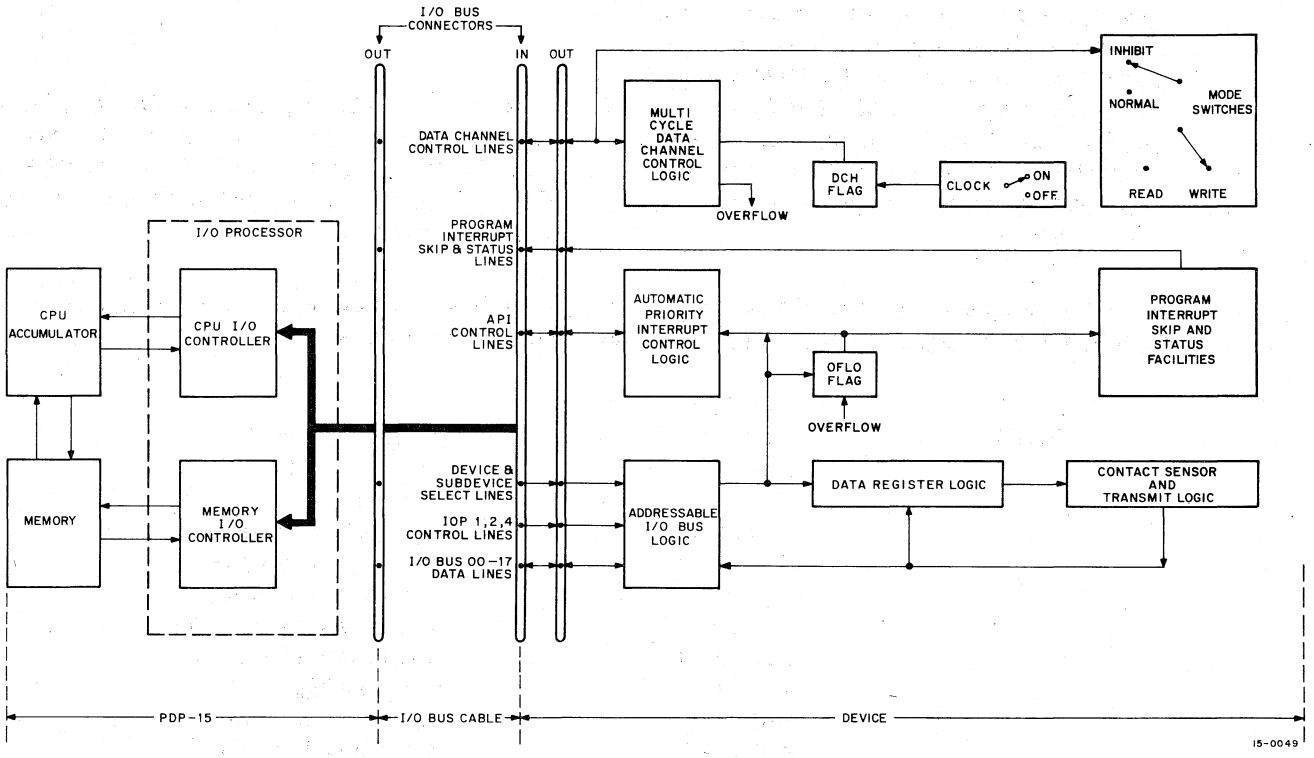


Figure 4-5 Block Diagram for Multicycle Data Channel Device

c. The I/O controller responds (some time later) with DCH GR H, setting DCH ENA of the M104 and posting I/O ADDR 13L, 14L, which signify address 30, the word count location. On the next I/O SYNC pulse the M104 flip-flop DCH EN B is set. Its purpose will be explained later.

d. The I/O controller increments location 30, which is hardware specified, and sends I/O OFLO back to the device if this location overflows (2's complement). Regardless, it increments location 31 (unless +1 →CA INH is posted during DCH ENA), the current address, and proceeds to transfer data.

e. If the MODE switch is set to READ (RD RQ L gate enabled and WR RQ L disabled), then DCH ENB (1) L will gate onto RD RQ L and signify to the I/O processor that a transfer into memory is required. The I/O processor issues a pulse on the IOP2 line. DCH EN B enables the M104 module at its optional select pin U2. IOP2 becomes IOT READ, and the data is strobed onto the data lines I/O BUS 00 L - I/O BUS 17 L, and into the location specified by the contents of memory location 31.

f. If the MODE switch is set to WRITE (WR RQ L gate enabled but the RD RQ L is not) then DCH EN B (1) L gates onto the WR RQ L line and signifies to the I/O processor that a transfer from memory is required. The I/O processor issues a pulse on the IOP4 line and DCH EN B enables the M103 module at its optional select pin U2. IOP4 becomes IOT WRITE and the data is strobed off the bus (placed there by the I/O processor) into the data register.

g. If this is the last transfer of a block, then the preset word count register (location 30) overflows (clears to zero) and an I/O OFLO H pulse sets the I/O OFLO flag. This flag then causes either an API request on level 0, or a program interrupt request. The result is the same as the previous example.

h. If the INHIBIT switch is enabled, then DCH EN A (1) L causes a signal +1 →CA INH L to inhibit incrementing of the current address location. (This feature is used by the DECtape control during a search).

i. If both WR RQ L and RD RQ L are enabled, then the machine goes into an ADD TO MEMORY mode whereby data is strobed onto the I/O bus with IOP2 (the RD RQ L part),and added to the contents of the memory location specified by the CA register. The sum is reread into the same location, and also transmitted back down the bus where it is strobed into the data register with IOP4 (WR RQ L part). If the sum is of a different sign than the two words, then a DATA OFLO pulse occurs to set the DATA OFLO flag.

Detailed descriptions of each signal and its timing relationship to associated signals are given in Table 4-2 and Figure 4-7. These should be studied in detail.

Note that all I/O bus signals are received by an M510 and transmitted by an M622 with the exception of the API EN IN, API EN OUT or DCH EN IN, DCH EN OUT, which are received and transmitted by their respective M104.

The following program will exercise the multi-cycle data channel device. It assumes that no other program or device is in operation within the system. Set the enable switch, specify read or write and go.

```
        .LOC 2050
START   CAF             /Clear the system flag and
        CLA             /the accumulator
        ION             /Turn on PI
        TAD (400000     /Set up the API control
        ISA             /word and enable API
        LAC (775777     /Set up the multi-cycle
        DAC 30          /data channel word count
        LAC (4000       /and current address to
        DAC 31          /2K8 and 4K8
        JMP.            /Loop until an interrupt occurs
        JMP START       /Go back to beginning
```

```
                .LOC 0
                000000          /PI setup
                JMP SKIP        /Go to skip chain

                .LOC 44
                JMS OFLO        /API trap point

                .LOC 2100
SKIP            IOT SKIP L      /Check the OFLO flags
                HLT             /Wrong flag - error
                JMS OFLO        /Go to overflow routine
                JMP * 0         /Go back to the beginning

OFLO            000000          /Subroutine entry point
                CLA             /Clear the AC
                CLL             /Clear the link
                IOT RDSTA       /Read the flags
                DAC TEMP        /Store this status
                IOT CLR OFLO    /Clear the flags
                RAR             /Rotate right (17→L)
                SZL             /Skip if L=0
                JMS DAFLO       /Go to data overflow routine
                CLA             /Clear the AC
                CLL             /Clear the link
                LAC TEMP        /Bring in the status word
                RTR             /Rotate twice (16→L)
                SZL             /Skip if L=0
                JMS IOFLO       /Go I/O OFLO routine
                JMP * OFLO      /Return to beginning

DAFLO           000000          /Subroutine entry point
                                /  The add to memory sum
                                /  has overflowed. This subroutine
                                /  should set up a double precision
                                /  add routine.
                DBR             /Prime API
                JM * DAFLO      /Return to beginning
```

Note also that if the +1 →CA INH had been enabled, the current address would not have changed and each word would have been transferred to the same address. This feature is useful when hunting for a specific word such as a block number on DECtape. Each number is transferred to the same location and checked against the number needed. This feature is also useful when trouble-shooting I/O problems on the multi-cycle data channel.

Note that the data could also have been read or written under program control as it was with the CPU device. This is often a useful maintenance tool.

Table 4-2
Multi-Cycle Data Channel Device

Signal Number	Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
1	DCH EN H (IN and OUT)	2BV2	This enable signal is a dc level which originates at the I/O processor and is daisy chained from device to device. The M104 logic in each device can interrupt this level, cutting the level off all devices that follow on the bus. A device receives it as DCH EN IN H and transmits it as DCH EN OUT H.	Each device can place a DCH request only if the incoming DCH EN level is true. By posting a request, the device immediately inhibits all controllers below it on the bus. In this way priorities on each level are established when devices request simultaneously.
2	I/O SYNC H	2AB1	The I/O processor clock pulse issued every microsecond. It is a 1 mHz, 250-ns pulse.	This signal is used to synchronize device control timing such as API RQ or DCH RQ to the I/O processor.
3	DCH GR H	2BT2	Issued by the I/O processor when it acknowledges a device's DCH RQ L.	The device uses DCH GR to gate the address of its word count onto the I/O ADDR lines of the I/O bus.
4	API 0 EN H (IN and OUT)	2BL1	This enable signal, a dc level originating in the I/O processor and daisy chained from device to device on the same level. The M104 logic in each controller can interrupt this level, cutting it off all devices that follow it on the bus. A device receives it as API 0 EN IN H and transmits it as API 0 EN OUT H.	Each device can post a request to its API level only if the incoming API EN level is true. BY posting a request, the device immediately inhibits all controllers below it on the bus. In this way priorities on each level are established when devices request simultaneously; the first device on this level will have highest priority; the next, etc.
5	API 0 GR H	2BJ1	One of four possible signals issued by the I/O processor indicating that it grants the API request at the corresponding level.	The device uses this signal to gate the address of its API level trap address onto the I/O ADDR lines.
6	I/O PWR CLR H	2AS1	System clear signal generated in response to 1) Power on or off; 2) CAF instruction; 3) I/O RESET key. It is a 1 mHz, 250-ns pulse.	Treated as an initializing signal for all devices (controllers attached to the I/O bus). All registers are reset to "initial" status.
7→13	DS0 H-DS5 H SD0 H	2AD2 2AT2	The six device select lines decoded from bits 6-11 of the IOT instruction, and the two sub-device select lines decoded from bits 12 and 13.	This signal together with DS1 – DS5 and SD0, SD1, is decoded by the device select logic in the controller, which responds to its unique code only.
14	IOP 1 H	2AD1	Micro-programmable control signal to effect an IOT instruction-specified operation within a device. Decoded from bit 17 of the IOT.	Used for I/O skip instructions to test a device flag or other control function. Cannot be used to read a device buffer register.
15	IOP 2 H	2AE1	Same as IOP 1 H. Decoded from bit 16. It is also issued during a multi-cycle data channel transfer into memory.	Usually used to effect a transfer of data from a selected device to the processor, or memory, or to clear a device register or flag, but may be used for other control functions. May not be used to determine a skip.
16	IOP 4 H	2AH1	Same as IOP 1 H. Decoded from bit 15. It is also issued during a multi-cycle data channel transfer out of memory.	Usually used to effect transfer of data from the CPU or memory, to the device or control. May not be used to determine a skip condition or to effect a transfer of data from a selected device to the CPU.
17,34	I/O BUS 00 L – 17 L	1AB1 – 1AV2	18 data lines which constitute the bidirectional facility for transferring data in bytes of up to 18 bits between the device and the CPU or memory.	These data lines convey data between the data register of the device and either memory or the AC of the CPU
41,35 42,36	I/O ADDR 12 L, 13 L 14 L, 15 L	1BK2 1BS2	Those lines which consititute an input bus for devices which must deliver address data to the I/O processor.	This address bus has two uses: *a.* To deliver the device's API trap address during its API break. *b.* To deliver the device's word count address during a multi-cycle DCH transfer, or an add to memory operation.

Table 4-2 (Cont)
Multi-Cycle Data Channel Device

Signal Number	Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
37	DCH RQ L	2BS2	A signal from a device to the I/O processor indicating a request for a multi-cycle data channel transfer, raised at I/O SYNC time.	This signal is interpreted by the I/O processor. It implies that some device wants to carry out a multi-cycle transfer or an increment memory, or add-to-memory.
38	+1 →CA INH L	2BE1	If the I/O processor sees this signal during multi-cycle transfers, it inhibits normal incrementing of the device's assigned current address memory location.	This facility is used by such peripherals as DECtape and magnetic tape when they search for records. It is also very useful during device checkout to prevent writing throughout memory and destroying the controlling program.
39	WR RQ L	2BB1	Indicates to the I/O processor that the device requires a transfer from memory.	The device uses this signal to inform the I/O processor that it wants a word from memory (during a multi-cycle data channel transfer).
40	RD RQ L	2AM1	Indicates to the I/O processor that the device is offering it a data word.	Used by the device to specify to the I/O processor that an input-to-CPU data transfer is required.
43	API 0 RQ L	2BH1	One of four API request signals on channels 0-3. This signal is set by the device. The signal is raised by the M104 logic at I/O SYNC time only.	The device uses this signal to inform the I/O processor of its request for service on API priority level 0, the highest of the four.
44	PROG INT RQ L	2AL1	This signal can cause the program to trap to location 000000 when no higher priority action is in progress. The instruction resident in location 000001 is fetched and executed.	A device delivers this level to the I/O processor to request interruption of the program in progress in order that the device be serviced.
45	SKIP RQ L	2AJ1	The return of the signal to the I/O processor during IOP1 indicates that an IOT instruction test for a skip condition has been satisfied. The PC is subsequently incremented by one.	Used by a device to inform the program of the state of its interrupt flag.
46	RD STATUS H	2AP1	A signal issued when the CPU issues an IORS instruction or when the console switch is placed on I/O STATUS.	Used by the device to gate its status onto the I/O bus data lines (one line per status bit) which is then read into the AC of the CPU.
47	I/O OFLO H	1BE1	This signal is issued during the first cycle of a multi-cycle data channel transfer, if the content (2's complement) of the word count assigned the currently active data channel device becomes zero when incremented.	This signal indicates to the device that the specified number of words have been transferred at the completion of the transfer in progress. It is normally used to turn off the respective device and to initiate a program interrupt or API request.
48	DATA OFLO H	1BD1	This signal is gated onto the bus by the I/O processor during the third cycle of an add-to-memory operation, when the sum (1's complement) of two like-signed numbers has an opposite sign.	This signal is used by the device to notify it an incorrect sum occurred because of overflow, during an add-to-memory operation.

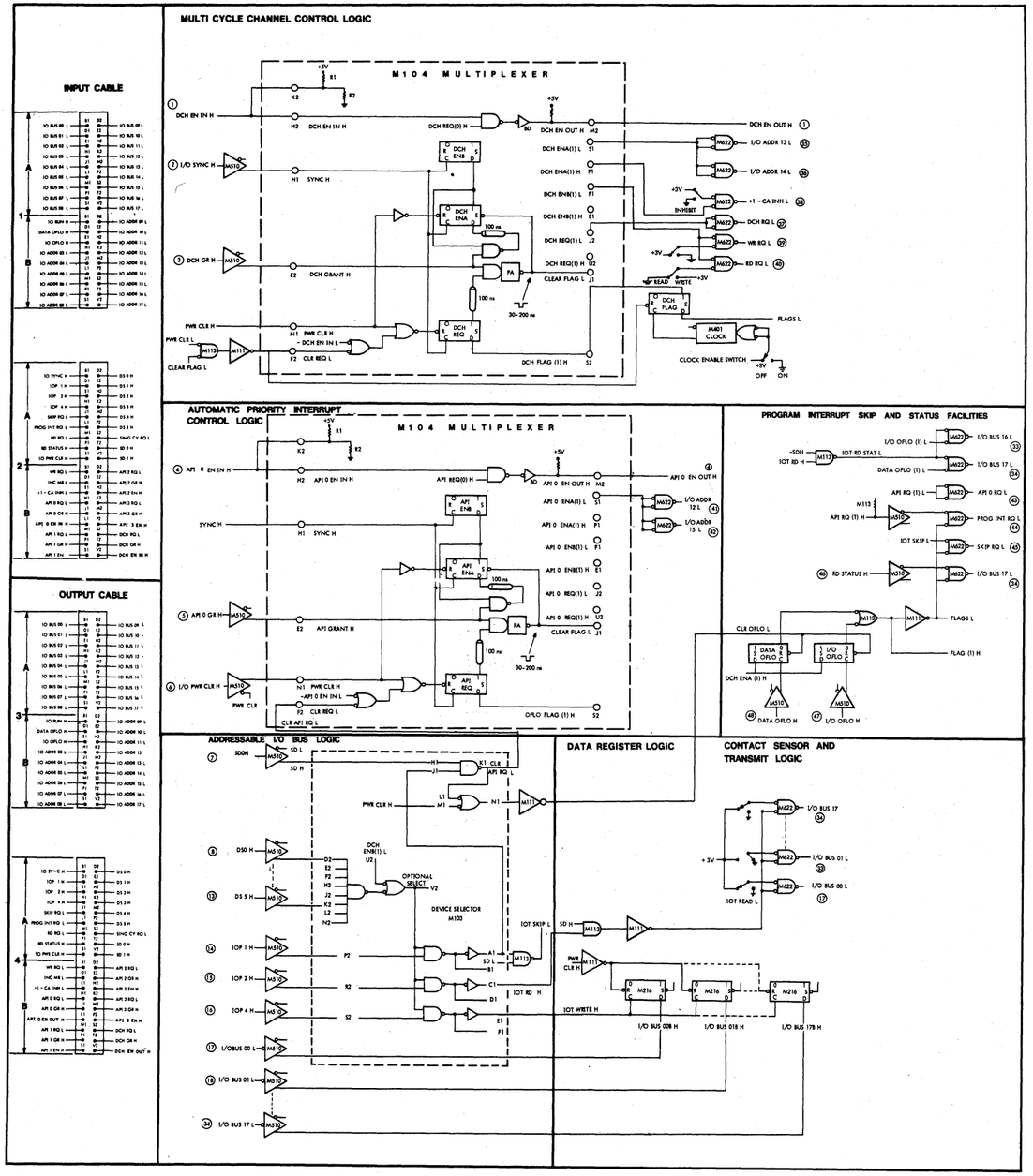


Figure 4-6 Multi-Cycle Data Channel Device Logic

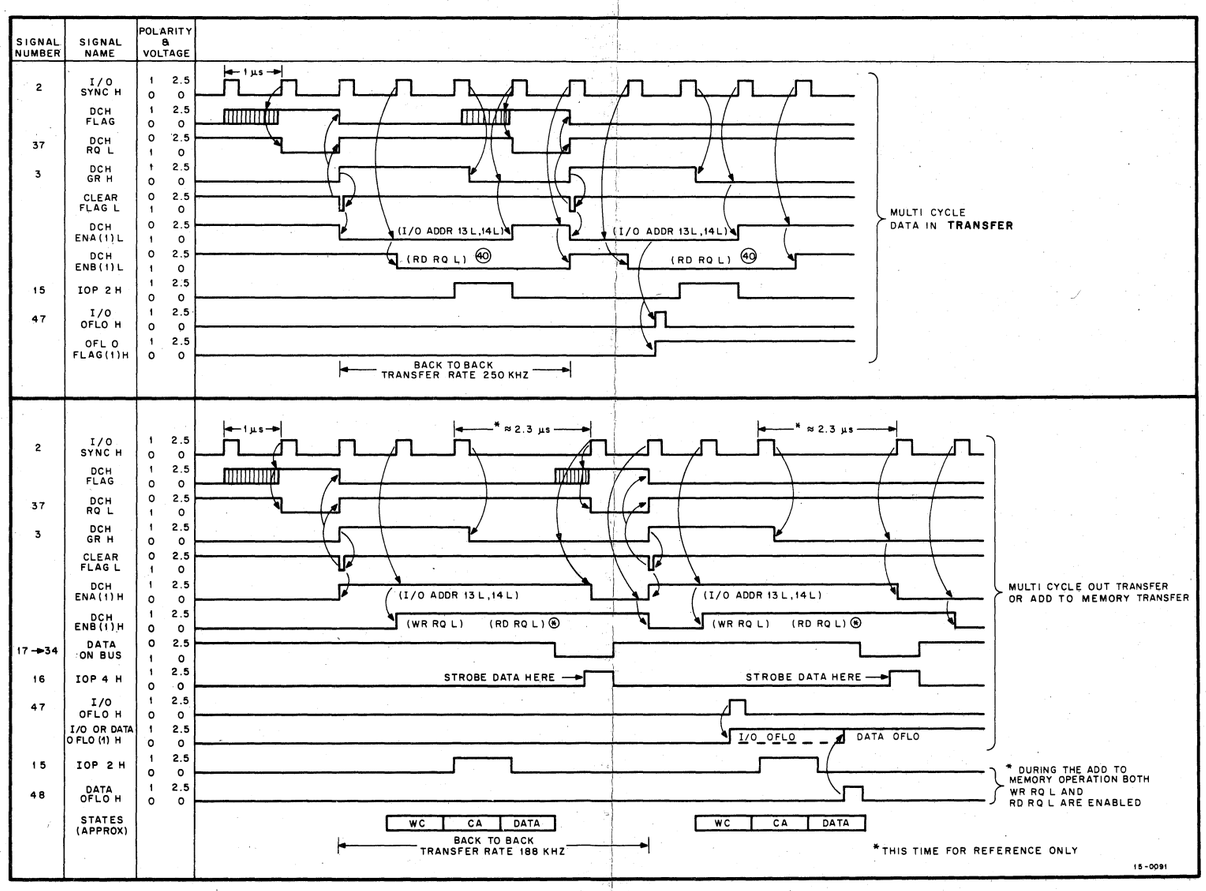


Figure 4-7 Multi-Cycle Data Channel Timing

4.2.2 Increment Memory

The basic logic needed to increment memory in any controller is very similar to the logic needed by a device which wants to transfer data on the multi-cycle data channel. It uses increment memory control logic to take care of the control and timing signals and the usual API, program interrupt, skip and status logic to handle its "bookkeeping" (Figure 4-8).

Since no data words are involved, the I/O BUS 00-17, and the data register, contact sensors and transmit logic are all absent.

Figure 4-9 is a detailed diagram of the logic for this device. The following describes a typical sequence.

a. The controller is initiated under program control by setting the CLK EN flag and enabling the M401 clock (clock logic).

b. The CLK RQ flag sets and in turn enables DCH REQ (of the increment memory control logic) to set with the next I/O SYNC H pulse.

c. DCH REQ causes a DCH RQ L line to signal the I/O processor which responds with DCH GR H, setting DCH ENA.

d. DCH ENA gates the address lines onto the I/O ADDR bus and also enables the INC MB L level. It is this control line that signals the I/O processor to go into an increment addressed memory sequence rather than a normal transfer routine.

e. If the specified location overflows (to zero), then an I/O OFLO H pulse sets CLK FLG, which then causes either an API 0 RQ L or a PROG INT RQ L to inform the program. The result is similar to previous devices. Complete timing and signal descriptions are given in Table 4-3 and Figure 4-10.

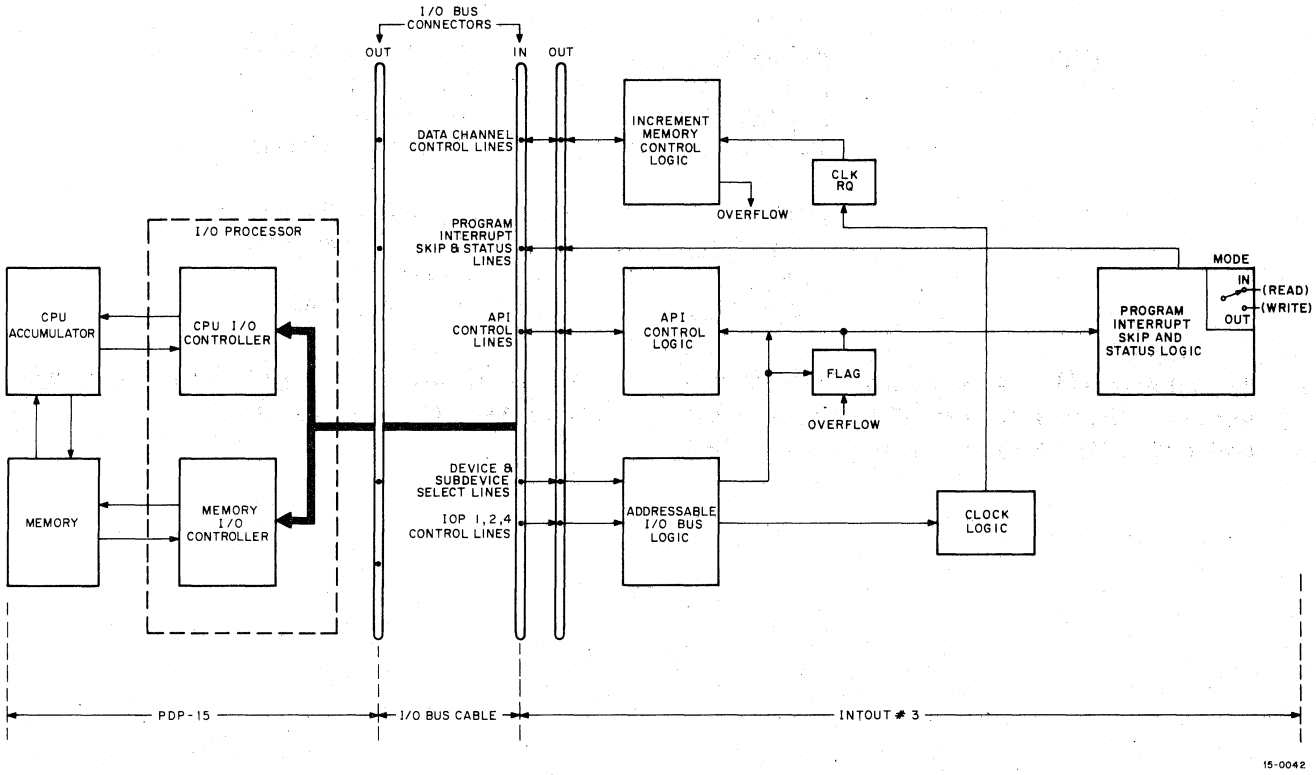


Figure 4-8 Block Diagram for Increment Memory

The following program is designed to exercise the increment memory device. Note that the computer has complete control over this device. No operator switches are used.

```
                .LOC 2150
START           CAF             /Clear all flags and
                CLA             /the accumulator
                ION             /Turn off PI
                TAD (400000     /Set up the API control word
                ISA             /and enable API.
                LAC (775777     /Set up the WC
                DAC 30          /register to overflow after 2K8.

                IOT SET CLK     /Turn on clock
                JMP.            /Wait in loop
                JMP START       /Go to beginning

                .LOC 0
                000000          /PI setup
                JMP SKIP        /Go to skip chain

                .LOC 44
                JMS OFLO        /API trap address

                .LOC 2200
SKIP            IOT SKIP L      /Check the CLK FLG
                HLT             /Wrong flag-error
                JMS CLK FLG     /Go to CLK FLG routine
                JMP * 0         /Return to beginning

CLK FLG         000000          /Subroutine entry point
                IOT CLR CLK FLG /Clear API and CLK FLG
                DBR             /Prime API
                JMP * CLK FLG   /Go to beginning
```

This device could be used as an external interval timer or an external events counter. If the address were to change (e.g., the output of an A/D converter) then a distribution curve could be plotted in memory. Note the minimal programming overhead.

This page intentionally left blank.

Table 4-3
Increment Memory Signals

Signal Number	Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
1	DCH EN H (IN and OUT)	2BV2	This enable signal is a dc level originating at the I/O processor and daisy chained from device to device. The M104 logic in each device can interrupt this level, cutting the level off all devices that follow on the bus. A device receives it as DCH EN IN H and transmits it as DCH EN OUT H.	Each device can post a DCH request only if the incoming DCH EN level is true. By posting a request, the device immediately inhibits all controllers below it on the bus. In this way priorities on each level are established when devices request simultaneously.
2	I/O SYNC H	2AB1	The I/O processor clock pulse issued every microsecond. This is a 1 mHz, 250-ns pulse.	This signal is issued to synchronize device control timing such as API RQ to the I/O processor.
3	DCH GR H	2BT2	Issued by the I/O processor when it acknowledges a device's DCH RQ L.	The device uses DCH GR to gate the address of its word count onto the I/O ADDR lines of the I/O bus.
4	API 0 EN (IN and OUT)	2BL1	This enable signal, is a dc level originating in the I/O processor and daisy chained from device to device on the same level. The M104 logic in each controller can interrupt this level, cutting off all devices that follow it on the bus. A device receives it as API 0 EN IN H and transmits it as API 0 EN OUT H.	Each device can post a request to its API level only if the incoming API EN level is true. By posting a request, the device immediately inhibits all controllers below it on the bus. In this way priorities on each level are established when devices request simultaneously.
5	API 0 GR H	2BJ1	A signal issued by the I/O processor indicating that the API request at level 0 is granted.	The device uses this signal to gate the address of its API level trap address onto the I/O ADDR lines.
6	I/O PWR CLR H	2AS1	System clear signal generated in response to 1) Power on or off; 2) CAF instruction; 3) I/O RESET key. This is a 1 mHz, 250-ns pulse.	This signal is treated as an initializing signal for all devices (controllers) attached to the I/O bus. All registers are reset to "initial" status.
7 →13	DS0 H-DS5 H →SD0 H, SD1 H	2AD2 2AV2	Six device and two subdevice select lines decoded from bit 6–13 of the IOT instruction.	This signal together with DS1-DS5 is decoded by the device select logic in the controller, which responds to its unique code only.
14	IOP 1 H	2AD1	Micro-programmable control signal part of an IOT instruction-specified operation within a device. Decoded from bit 17 of the IOT.	Used for I/O skip instructions to test a device flag or other control function. Cannot be used to initiate loading or reading a device buffer register.
15	IOP 2 H	2AE1	Same as IOP 1 H. Decoded from bit 16 of the IOT.	Usually used to effect a transfer of data from a selected device to the processor, or to clear a device register, but may be used for other control functions. May not be used to determine a skip.
16	IOP 4 H	2AH1	Same as IOP 1 H. Decoded from bit 15 of the IOT.	Usually used to effect transfer of data from the CPU to the device, or control. May not be used to determine a skip condition or to effect a transfer of data from a selected device to the CPU.
17,18 21,22	I/O ADDR 12 L, 13 L 14 L, 15 L	1BK2 - 1BS2	Several of fifteen lines which consitute an input bus for devices which must deliver address data to the I/O processor.	This address bus has two uses: *a.* To deliver the device's API trap address during its API break; *b.* to deliver the device's word count address during a multi-cycle DCH transfer, an increment memory operation, or add to memory.

Table 4-3 (Cont)
Increment Memory Signals

Signal Number	Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
19	INC MB L	2BD1	Forces the I/O processor to increment the memory location specified by the 15-bit address lines on the I/O bus.	This feature allows a device to increment memory locations in one cycle without disturbing the CPU.
20	DCH RQ L	2BS2	A signal from a device to the I/O processor indicating a request for a multi-cycle data channel transfer, or an increment memory request.	This signal is interpreted by the I/O processor that some device wants to carry out a multi-cycle transfer or an increment memory.
23	API 0 RQ L	2BH1	One of four API request signals on channels 0–3. This signal is set by the device.	The device uses this signal to inform the I/O processor of its request for API service on priority level 0, the highest of the four.
24	PROG IN RQ L	2AL1	This signal can cause the program to trap to location 000000 when no higher priority action is in progress. The instruction resident in location 000001 is fetched and executed.	A device delivers this level to the I/O processor to request interruption of the program in progress in order that the device be serviced.
25	SKIP RQ L	2AJ1	The return of the signal to the I/O processor during IOP1 indicates that an IOT instruction test for a skip condition has been satisfied. The PC is subsequently incremented by one.	Used by a device to inform the program of the state of its interrupt flag.
26	I/O BUS 17L	1AV2	Data line 17.	Used to read the status of the clock flag into the AC.
27	RD STATUS H	2AP1	A signal issued when the CPU issues an IORS instruction or when the console switch is placed on I/O STATUS.	Used by the device to gate its status onto the I/O bus data lines (one line per status bit) which is then read into the AC of the CPU.
28	I/O OFLO H	1BE1	This signal is issued during the first cycle of a multi-cycle data channel transfer or an increment memory cycle, if the content (2's complement) of the word count assigned the currently active data channel device becomes zero when incremented.	This signal indicates to the device that the specified number of words have been transferred at the completion of the transfer in progress. It is normally used to turn off the respective device and to initiate a program interrupt or API request.

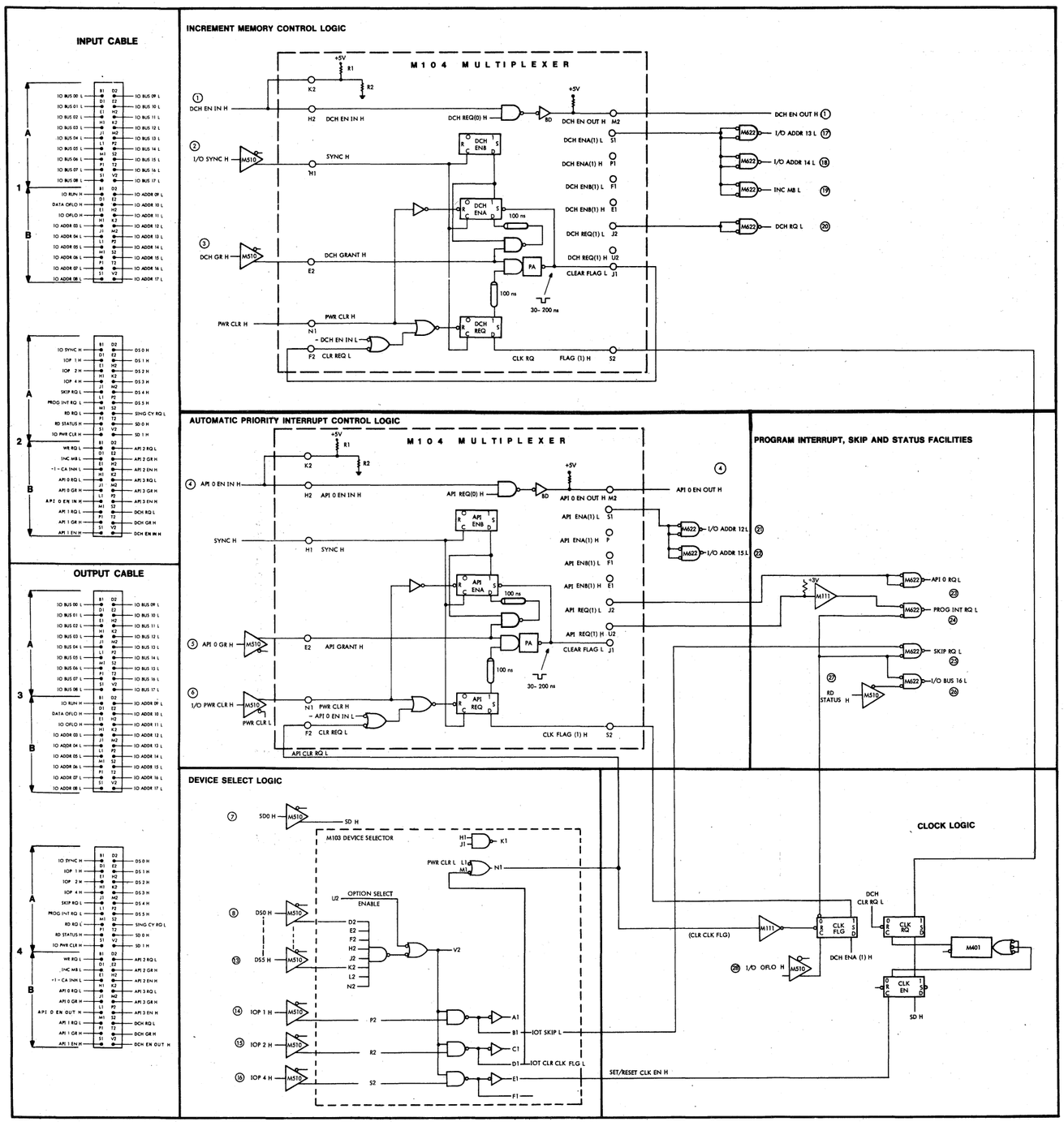


Figure 4-9 Increment Memory Device Logic

SIGNAL NUMBER	SIGNAL NAME	POLARITY & VOLTAGE	
2	I/O SYNC H	1	2.5
		0	0
	CLK RQ(1) H	1	2.5
		0	0
20	DCH RQ(1) L	0	2.5
		1	0
3	DCH GR H	1	2.5
		0	0
	CLEAR FLG L	0	2.5
		1	0
19	DCH ENA	1	2.5
		0	0
28	I/O OFLO H	1	2.5
		0	0
	CLK FLG	1	2.5
		0	0

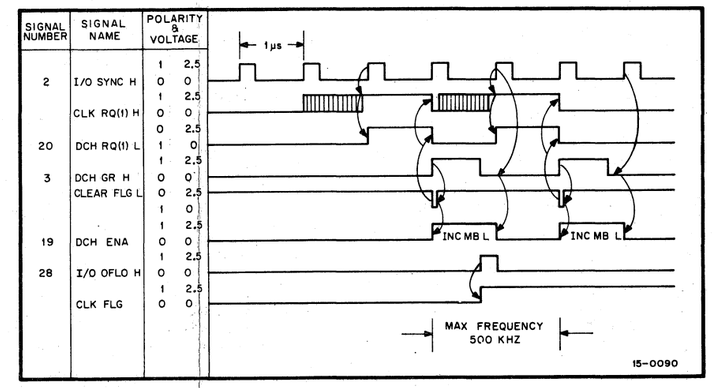


Figure 4-10 Timing Diagram for Increment Memory Devices

4.2.3 The Single-Cycle Data Channel Device

The control logic for single-cycle data channel transfers is very similar to that for multi-cycle breaks, with one major exception. The number and destination/source of transfers are parameters handled by the I/O processor and memory during multi-cycle breaks, whereas during single-cycle transfers the device must remember its own word count and specify the absolute address it must transfer to/from (rather than the location of the current address). Therefore, each single-cycle operating device will have its own word count and current address register. The time it takes to complete a transfer in this manner is reduced by two cycles. Figure 4-11 shows a block diagram of a typical device operating in single-cycle data channel mode.

A single-cycle device operates under two conditions – burst mode or normal mode. Burst mode is used when the device can transfer a word every microsecond. Typically, a slow device is double or triple buffered and transmits the data in a burst of two or three words. In burst mode, only the first word suffers the time it takes to synchronize the device with memory through the I/O processor. Then, each successive word is transferred every microsecond until the last, after which synchronization is lost. During a burst, the transfer cannot change direction, and just before the last word, the switch must be turned to normal.

In normal mode, each data channel request requires 3 to 5 μs to synchronize (unless it is the last transfer of burst mode). The transfer rate, therefore, is reduced considerably from burst mode.

A detailed logic diagram for this peripheral is given in Figure 4-12. It works in the following way:

a. An IOT instruction is issued to set the DATA MODE flag to DATA IN (IOP2 with SDO selected) or DATA OUT (IOP2 without SDO selected). The word count and current address registers are initiated (not shown). The M401 clock is switched on. The peripheral is set to burst or normal mode.

b. The DCH FLAG is set by the clock; it enables SING CY RQ of the M104, which sets on the next I/O SYNC H pulse.

c. DCH RQ L is also posted, if it is to be a transfer into the computer (DATA IN). The I/O processor begins to synchronize with memory and issues DCH GR H.

d. DCH GR H sets DCH EN A which places the contents of the current address register onto I/O ADDR lines 03 through 17. Later, when DCH GR H is reset, an address accepted or data accepted pulse is generated which should be used to increment the device's current address and word count register.

e. If the word count register in the device overflows signaling that the last transfer has been completed, the register must issue an overflow pulse to set the OFLO flag and cause an API or program interrupt request in the usual way.

f. If the I/O processor has sensed a DATA OUT transfer:

(1) It places the content of the memory location specified by the current address onto the I/O BUS 00 – 17 lines; then

(2) It issues an IOP4 which passes through the W103 because DCH ENA (1) L has asserted its option select line. IOP4 strobes the data off the lines into the device data register.

g. If the I/O processor has sensed a DATA IN transfer, then DCH ENA (1) H together with DATA MODE (1) H gate the state of the switch register onto the data lines. The I/O processor then strobes them into memory.

h. If the peripheral is in normal mode, then the DCH FLAG, DCH RQ are reset when DCH ENA is set. In burst mode they are inhibited from resetting.

i. In burst mode the peripheral must count the number of DCH GR H's, and on the rising edge of the last, it must reset to normal mode. The M104 unripples in the usual way after the last transfer. In the timing diagram of Figure 4-13b, it is assumed that four transfers occur.

The single-cycle device requires the same programming effort as the multi-cycle devices. Most of the work comes in setting up the word count and current address registers (not shown) or setting modes. It is faster and uses less I/O processor time, however.

```
                .LOC 2250
START           CAF             /
                CLA             /       Same as previous devices
                ION             /
                TAD (400000

                ISA
                LAC (WC)        /Set up the
                IOT SET WC      /WC, CA
                LAC (CA)        /registers and the
                IOT SET CA      /mode;
                IOT SET MODE    /then wait in
                JMP.            /a loop

                .LOC 0
                000000          /
                JMP SKIP        /

                .LOC 44         /
                JMS OFLO        /

                .LOC 2300
SKIP            IOT SKIP L      /       Same as previous devices
                HLT
                JMS OFLO        /
                JMP * 0         /
OFLO            000000
                IOT CLR OFLO    /
                DBR             /
                JMP * OFLO      /
```

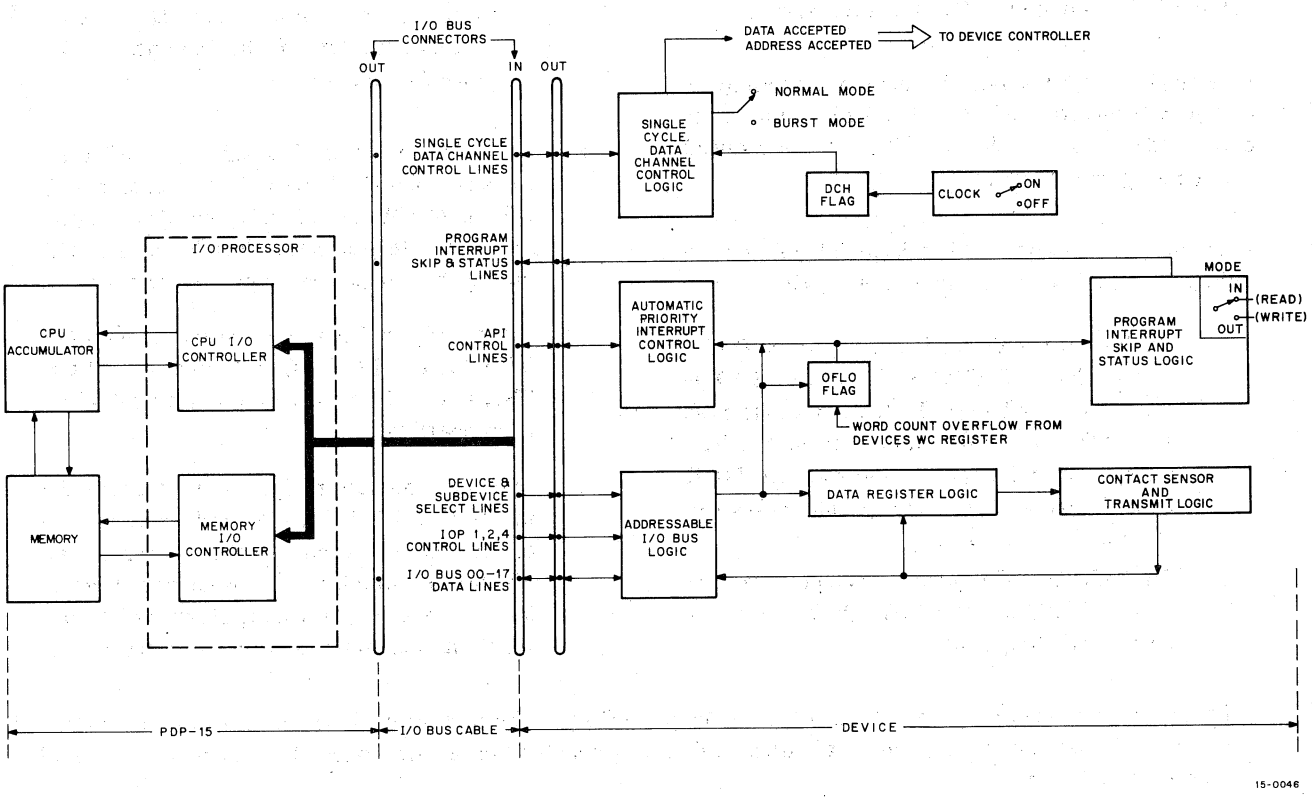


Figure 4-11 Block Diagram of Single-Cycle Data Channel Device

This page intentionally left blank.

Table 4-4
Single-Cycle Data Channel Interface

Signal Number	Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
1	DCH EN H	2BV2	This enable signal is a dc level originating at the I/O processor and daisy chained from device to device. The M104 logic in each device can interrupt this level, cutting the level off all devices that follow on the bus. A device receives it as DCH EN IN H and transmits it as DCH EN OUT H.	Each device can post a DCH request only if the incoming DCH EN level is true. By posting a request, the device immediately inhibits all controllers below it on the bus. In this way priorities on each level are established when devices request simultaneously.
2	I/O SYNC H	2AB1	The I/O processor clock pulse issued every microsecond; a 1 mHz, 250-ns pulse.	This signal is used to synchronize device control timing such as API RQ to the I/O processor.
3	DCH GR H	2BT2	Issued by the I/O processor when it acknowledges a device's DCH RQ L.	The device uses DCH GR to set ENA which gates the memory address onto the I/O ADDR lines of the I/O bus.
4	API 0 EN	2BL1	This enable signal, is a dc level originating in the I/O processor and daisy chained from device to device on the same level. The M104 logic in each controller can interrupt this level, cutting the level off all devices that follow it on the bus. A device receives it as API 0 EN IN H and transmits it as API 0 EN OUT H.	Each device can post a request to its API level only if the incoming API EN level is true. By posting a request, the device immediately inhibits all controllers below it on the bus. In this way priorities on each level are established when devices request simultaneously.
5	API 0 GR H	2BJ1	One of four possible signals issued by the I/O processor indicating that it grants the API request at the corresponding level, 0.	This device uses this signal to gate the address of its API level trap address onto the I/O ADDR lines.
6	I/O PWR CLR H	2AS1	System clear signal generated in response to: 1) Power on or off; 2) CAF instruction; 3) I/O RESET key; a 1 mHz, 250-ns pulse.	This signal is treated as an initializing signal for all devices (controllers) attached to the I/O bus. All registers are reset to "initial" status.
7 13	DS0 H-DS5 H SD0 H-SD1 H	2AD2 2AV2	Six device and two subdevice select lines decoded from IOT instruction bits 6-13.	These signals are decoded by the device select logic in the controller, which responds to its unique code only.
14	IOP 1 H	2AD1	Micro-programmable control signal; part of an IOT instruction-specified operation within a device. Decoded from bit 17 of the IOT.	Used for I/O skip instructions to test a device flag or other control function. Cannot be used to read a device buffer register.
15	IOP 2 H	2AE1	Same as IOP 1 H. Decoded from bit 16 of the IOT.	Usually used to effect a transfer of data from a selected device to the processor or memory, or to clear a device register, but may be used for other control functions. May not be used to determine a skip.
16	IOP 4 H	2AH1	Same as IOP 1 H, and it is also issued during an out-going single-cycle transfer. Decoded from bit 15 of the IOT.	Usually used to effect transfer of data from the CPU of memory to the device, or for control. May not be used to determine a skip condition or to effect a transfer of data from a selected device to the CPU.
	I/O BUS 00 L – 17 L	1AB1 – 1AV2	Eighteen data lines which constitute the bi-directional facility for transferring data in bytes of up to 18 bits between the device and either the CPU or memory.	These data lines (I/O BUS 00 L – I/O BUS 17 L) convey data between the AC of the CPU and a selected device information buffer register or, the bus buffer of the I/O processor and a selected device buffer register during data channel operations.
35–49	I/O ADDR 03 L – 17 L	1BH1 – 1BV2	Fifteen lines which constitute an input bus for devices which must deliver address data to the I/O processor.	This address bus has two uses: *a.* To deliver the device's API trap address during its API break. *b.* To deliver the device's word count address during a multi-cycle DCH transfer, an increment memory operation, add to memory, or single-cycle break.
50	SING CY RQ L	2AS2	Indicates when a device wants to carry out a single-cycle data transfer to memory.	The device uses this line to request from the I/O processor a single-cycle transfer. If a DCH RQ L signal is sent with it, the I/O processor responds to an input (to computer) transfer; otherwise, it determines an output transfer.

Table 4-4 (Cont)
Single-Cycle Data Channel Interface

Signal Number	Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
51	DCH RQ L	2BS2	A signal from a device to the I/O processor, when posted with a single cycle request, shows that an input transfer must be effected.	This signal is interpreted by the I/O processor in this way: if a single-cycle request (SING CY RQ L) is posted, then it and DCH RQ L are ANDed to inform the I/O processor that a transfer into memory is to be effected in single-cycle mode; otherwise, the I/O processor assumes an outgoing single-cycle transfer is required.
52	API 0 RQ L	2BH1	One of four API request signals on channels 0–3. This signal is set by the device at I/O SYNC time.	The device uses this signal to inform the I/O processor of its request for service on API priority level 0, the highest of the four.
53	PROG INT RQ L	2AL1	This signal can cause the program to trap to location 000000 when no higher priority action is in progress. The instruction resident in location 000001 is fetched and executed.	A device delivers this level to the I/O processor to request interruption of the program in progress in order that the device be serviced.
54	SKIP RQ L	2AJ1	The return of the signal to the I/O processor during IOP1 indicates that an IOT instruction test for a skip condition has been satisfied. The PC is subsequently incremented by one.	Used by a device to inform the program of the state of its interrupt flag.
55	RD STATUS H	2AP1	A signal issued when the CPU issues an IORS instruction or when the console switch is placed on I/O STATUS.	Used by the device to gate its status onto the I/O bus data lines (one line per status bit) which is then read into the AC of the CPU.
56	SD 1 H	2AV2	Same as SD0 H - Decoded from bit 13 of the IOT.	Same as SD0 H.

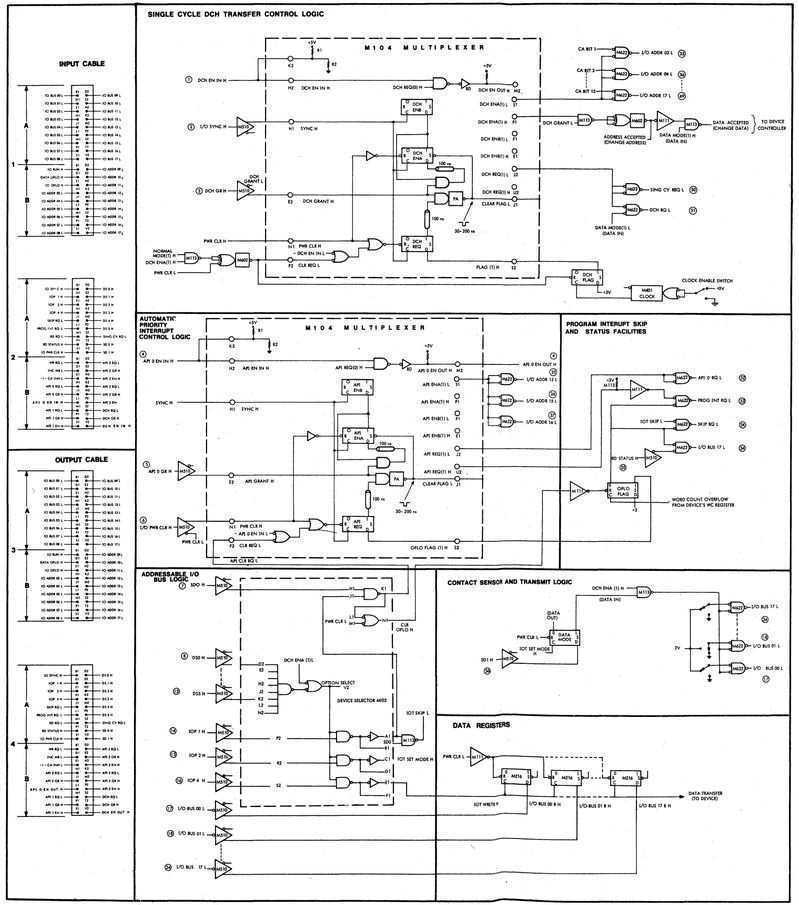
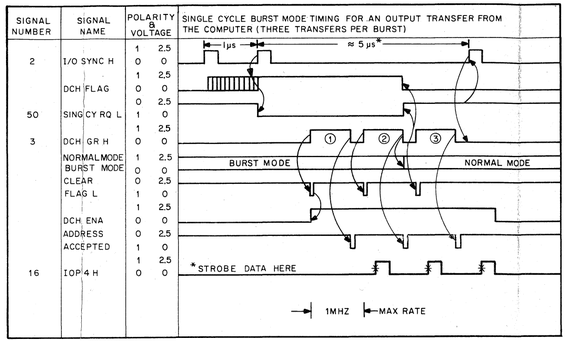
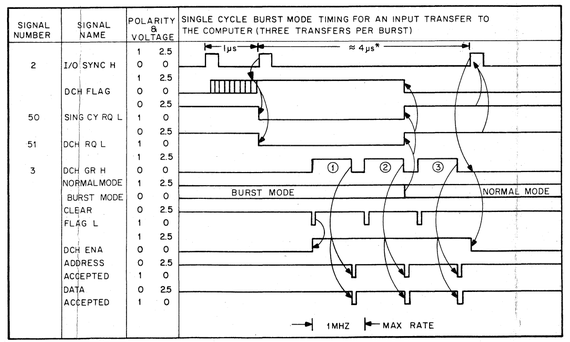


Figure 4-12 Logic Diagram of Single-Cycle Data Channel

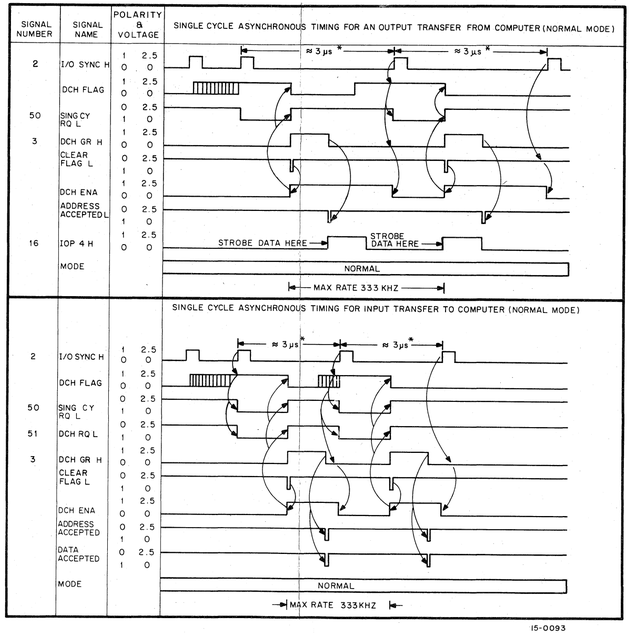




*NOTE THESE TIMES FOR REFERENCE ONLY

15-0092

Figure 4-13a Timing Diagram for Single-Cycle Device in Normal Mode



15-0093

*NOTE THESE TIMES FOR REFERENCE ONLY

Figure 4-13b Single-Cycle Asychronous Timing

4.3 SYSTEM PRIORITY STRUCTURES

There are three classes of priorities within the PDP-15 system:

a. Class 1 is established if the I/O processor and the central processor simultaneously request a memory cycle. Under this condition, the I/O processor is served first; this is necessary to prevent the CPU from shutting out the I/O.

b. Class 2 occurs within the I/O processor itself. The five subsystems which use the I/O processor, the data channel, real time clock, API, PI and IOT instructions, are ordered from data channel to IOT.

Classes 1 and 2 are discussed in more detail in the PDP-15 System Reference Manual and the User's Handbook.

c. Class 3 is the priority structure associated with devices on the I/O bus which use the data channel or the API.

There are four API levels, with three associated control signals (API RQ, API EN and API GR); they are ordered from the highest level 0 to the lowest level 3. Any device of a higher level not only takes priority over another of a lower level, but actively interrupts that level during its service routine (see User's Handbook), thus making nested interrupts possible.

Each API level can have up to eight devices using it, although the system software allots only 28 memory locations for this function. If any two devices on the same level request interrupts simultaneously, the device closest to the computer on the I/O bus is serviced first. Figure 4-14 illustrates this process. If device #1 and the device #2 post REQ flags together, the API 0 EN level (for example) to device #2 is disabled by the REQ flag of device #1 through the AND gate of its M104 Multiplexer. The disabled API 0 EN flag grounds the direct clear of the REQ flip-flop in device #2, and inhibits it until the API of device #1 is serviced.

If any device in the chain does not use API or data channel, the respective ENABLE lines must be jumped from the input to the output cable.

The data channel devices operate the same way on only one ENABLE line. Up to eight devices can use the data channel, four of which can be multi-cycle devices and eight can be single-cycle devices. The restriction on cycle devices is a software constraint explained in Chapter 6. Because of the priority scheme on the data channel, latency sensitive devices should be placed closest to the I/O processor.

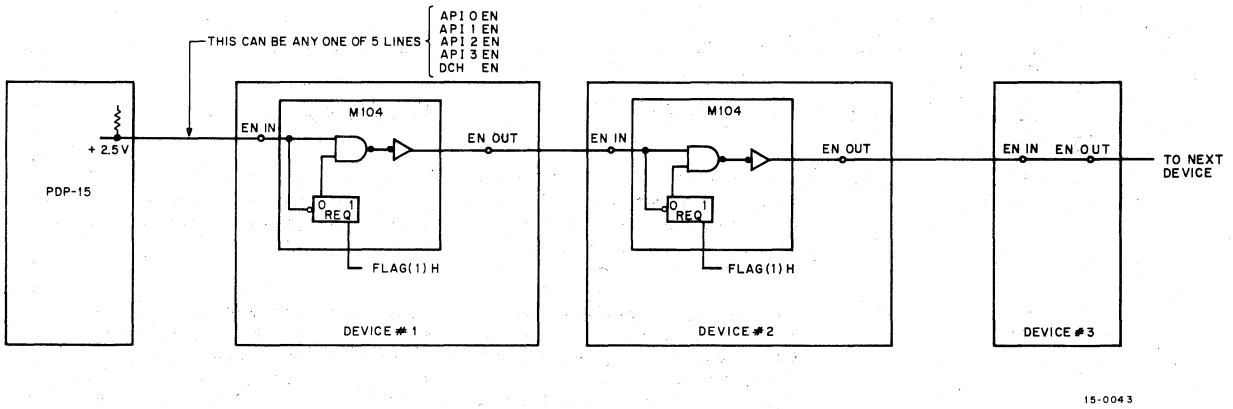


Figure 4-14

4.4 SUMMARY OF I/O BUS FUNCTIONS

Table 4-5 describes I/O Bus signals and their functions.

Table 4-5
Summary of I/O Bus Signal Functions

Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
API 0 EN H	2BL1	This enable signal, one of four in the API system, is a dc level originating in the I/O processor and daisy chained from device to device on the same level. The M104 logic in each controller can interrupt this level, cutting the level off all devices that follow it on the bus. A device receives it as API 0 EN IN H and transmits it as API 0 EN OUT H.	Each device can post a request to its API level only if the incoming API EN level is true. By posting a request the device immediately inhibits all controllers below it on the bus. In this way priorities on each level are established when devices request simultaneously.
API 1 EN H	2BS1	Same as API 0 EN H	Same as API 0 EN H
API 2 EN H	2BH2	Same as API 0 EN H	Same as API 0 EN H
API 3 EN H	2BP2	Same as API 0 EN H	Same as API 0 EN H
API 0 GR H	2BJ1	One of four possible signals issued by the I/O processor indicating that it grants the API request at the corresponding level.	The device uses this signal to gate the address of its API level trap address onto the I/O ADDR lines.
API 1 GR H	2BP1	Same as API 0 GR H	Same as API 0 GR H
API 2 GR H	2BE2	Same as API 0 GR H	Same as API 0 GR H
API 3 GR H	2BM2	Same as API 0 GR H	Same as API 0 GR H
API 0 RQ L	2BH1	One of four API request signals on channels 0 – 3. This signal is set by the device at I/O SYNC time.	The device uses this signal to inform the I/O processor of its request for API priority level 0, the highest of the four.
API 1 RQ L	2BM1	Same as API 0 RQ L	Request API priority level 1.
API 2 RQ L	2BD2	Same as API 0 RQ L	Request API priority level 2.
API 3 RQL	2BK2	Same as API 0 RQ L	Request API priority level 3.
DATA OFLO H	1BD1	This signal is gated onto the bus by the I/O processor during the third cycle of an add-to-memory operation when the sum (1's complement) of two like-signed numbers has an opposite sign.	This signal is used by the device to notify it when an incorrect sum occurs, because of overflow during an add-to-memory operation.
DCH EN H	2BV2	This enable signal is a dc level originating at the I/O processor and daisy chained from device to device. The M104 logic in each device can interrupt this level, cutting the level off all devices that follow on the bus. A device receives it as DCH EN IN H and transmits it as DCH EN OUT H.	Each device can post a DCH request only if the incoming DCH EN level is true. By posting a request, the device immediately inhibits all controllers below it on the bus. In this way priorities are established when devices request simultaneously.
DCH GR H	2BT2	Issued by the I/O processor when it acknowledges a device's DCH RQ L.	The device uses DCH GR to gate the address of its word count onto the I/O ADDR for 3-cycle transfers and gates memory address during 1-cycle transfers.
DCH RQ L	2BS2	A signal from a device to the I/O processor indicating either a request for a multi-cycle data channel transfer or, when posted with a single-cycle request, showing that an input transfer must be effected. The table below shows how the two functions relate.	This signal is interpreted by the I/O processor in in two ways: If it is present without a single-cycle request, it implies that some device wants to carry out a multi-cycle transfer, an increment memory, or add to memory.

Table 4-5 (Cont)
Summary of I/O Bus Signal Functions

Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
		DCH RQ L / SING CY RQ L / FUNCTION 0 / 0 0 / 1 — Single-Cycle Transfer Out 1 / 0 — Multi-Cycle Transfer (In or Out) 1 / 1 — Single-Cycle Transfer In	If a single-cycle request is also posted, then the two signals are ANDed to inform the I/O processor that a single-cycle transfer into memory is to be effected. Otherwise, the I/O processor assumes an outgoing single-cycle transfer is required.
DS0 H	2AD2	The first of six device select lines decoded from bit 6 of the IOT instruction.	This signal together with DS1 – DS5 is decoded by the device select logic in the controller, which responds to its unique code only.
DS1 H	2AE2	The second of the six device select lines.	See DS0 H
DS2 H	2AH2	The third of the six device select lines.	See DS0 H
DS3 H	2AK2	The fourth of the six device select lines.	See DS0 H
DS4 H	2AM2	The fifth of six device select lines.	See DS0 H
DS5 H	2AP2	The sixth of six device select lines.	See DS0 H
INC MB L	2BD1	Forces the I/O processor to increment the contents of the memory location specified by the 15-bit address lines on the I/O bus.	This feature allows a device to increment memory locations in one cycle without disturbing the CPU.
I/O ADDR 03 L	1BH1	One of fifteen lines which constitute an input bus for devices which must deliver address data to the processors.	This address bus has two uses: *a.* To deliver the device's API trap address during its API break. *b.* To deliver the device's word count address during a multi-cycle DCH transfer, an increment memory operation, or to add to memory. To deliver an absolute address during single-cycle transfers.
I/O ADDR 04 L	1BJ1	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 05 L	1BJ1	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 06 L	1BM1	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 07 L	1BP1	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 08 L	1BS1	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 09 L	1BD2	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 10 L	1BE2	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 11 L	1BH2	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 12 L	1BK2	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 13 L	1BM2	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L

Table 4-5 (Cont)
Summary of I/O Bus Signal Functions

Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
I/O ADDR 14 L	1BP2	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 15 L	1BS2	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 16 L	1BT2	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O ADDR 17 L	1BV2	Similar to I/O ADDR 03 L	Similar to I/O ADDR 03 L
I/O BUS 00 L	1AB1	The first of 18 data lines which constitute the bidirectional facility for transferring data in bytes of up to 18 bits between the device and either the CPU or memory. This is the MSB.	These data lines (I/O BUS 00 L through I/O BUS 17 L) convey data between *a.* The AC of the CPU and a selected device information buffer register or *b.* the bus buffer of the I/O processor and a selected device buffer register during data channel operations.
I/O BUS 01 L	1AD1	Data line two	See I/O BUS 00L
I/O BUS 02 L	1AE1	Data line three	See I/O BUS 00 L
I/O BUS 03 L	1AH1	Data line four	See I/O BUS 00 L
I/O BUS 04 L	1AJ1	Data line five	See I/O BUS 00 L
I/O BUS 05 L	1AL1	Data line six	See I/O BUS 00 L
I/O BUS 06 L	1AM1	Data line seven	See I/O BUS 00 L
I/O BUS 07 L	1AP1	Data line eight	See I/O BUS 00 1
I/O BUS 08 L	1AS1	Data line nine	See I/O BUS 00 L
I/O BUS 09 L	1AD2	Data line ten	See I/O BUS 00 L
I/O BUS 10 L	1AE2	Data line eleven	See I/O BUS 00 L
I/O BUS 11 L	1AH2	Data line twelve	See I/O BUS 00 L
I/O BUS 12 L	1AK2	Data line thirteen	See I/O BUS 00 L
I/O BUS 13 L	1AM2	Data line fourteen	See I/O BUS 00 L
I/O BUS 14 L	1AP2	Data line fifteen	See I/O BUS 00 L
I/O BUS 15 L	1AS2	Data line sixteen	See I/O BUS 00L
I/O BUS 16 L	1AT2	Data line seventeen	See I/O BUS 00 L
I/O BUS 17 L	1AB2	Data line eighteen This is the LSB	See I/O BUS 00 L

Table 4-5 (Cont)
Summary of I/O Bus Signal Functions

Signal Mnemonic	Connector Pin Number	Signal Definition	Signal Function
I/O OFLO H	1BE1	This signal is issued during the first cycle of a multi-cycle data channel transfer or an increment memory cycle, if the content (2's complement of the word count assigned the currently active data channel device becomes zero when incremented.	This signal indicates to the device that the specified number of words have been transferred at the completion of the transfer in progress. It is normally used to turn off the respective device and to initiate a program interrupt or API request.
IOP 1 H	2AD1	Microprogrammable control signal part of an IOT instruction-specified operation within a device. Decoded from bit 17 of the IOT.	Used for I/O skip instructions to test a device flag or other control function. Cannot be used to read a device buffer register. In general, a designer should be wary of using IOP pulses for multiple purposes. Never clear and skip on a flag, with the same IOT, for example!
IOP 2 H	2AE1	Same as IOP 1 H and it is also issued during a multi-cycle data channel transfer into memory. Decoded from bit 16 of the IOT.	Usually used to effect a transfer of data from a selected device to the processor or memory, to clear a device register, but may be used for other control functions. May not be used to determine a skip.
IOP 4 H	2AH1	Same as IOP 1 H and it is also issued during a multi- or single-cycle data channel transfer out of memory. Decoded from bit 15 of the IOT.	Usually used to effect transfer of data from the CPU or memory to the device or control. May not be used to determine a skip condition or to effect a transfer of data from a selected device to the CPU.
I/O PWR CLR H	2AS1	System clear signal generated in response to: 1. Power on or off 2. CAF instruction 3. I/O RESET key 1 mHz, 250-ns pulse width	This signal is treated as an initializing signal for all devices (controllers) attached to the I/O bus. All registers are reset to "initial" status.
I/O RUN H	2BB1	This level becomes high when the CPU is running.	Can be used to disable a device if the CPU stops.
I/O SYNC H	2AB1	The I/O processor clock pulse issued every microsecond; 1 mHz, 250-ns pulse width.	This signal is used to synchronize device control timing such as API RQ and DC H RQ to the I/O processor.
PROG INT RQ L	2AL1	This signal can cause the program to trap to location 000000. The instruction resident in location 000001 is fetched and executed.	A device delivers this level to the I/O processor to request interruption of the program in progress in order that the device be serviced.
RD RQ L	2AM1	Indicates to the processor that the device is sending it a data word.	Used by the device to specify to the I/O processor an input-to-CPU data transfer is required.
RD STATUS H	2AP1	A signal issued when the CPU issues an IORS instruction or when the console switch is placed on I/O STATUS.	Used by the device to gate its status onto the I/O bus data lines (one line per status bit) which is then read into the AC of the CPU.
SD0 H	2AT2	The first of two subdevice select lines decoded from bit 12 of the IOT instruction.	This signal and DS1 H can be decoded by the device for mode selection.
SD1 H	2AV2	Same as SD0 H except if it is decoded from bit 13 of the IOT instruction.	Same as SD0 H
SING CY RQ L	2AS2	Indicates when a device wants to carry out a single-cycle data transfer to memory.	The device uses this line to request from the I/O processor a single-cycle transfer. If a DCH RQ signal is sent with it, then the I/O processor responds to an input (to computer) transfer. Otherwise it determines an output transfer.
SKIP RQ L	2AJ1	The return of the signal to the I/O processor during IOP^1 indicates that an IOT instruction test for a skip condition has been satisfied. The PC is subsequently incremented by one.	Used by a device to inform the program of the state of its interrupt flag.
WR RQ L	2BB1	Indicates to the I/O processor that the device requires a transfer from memory during a multi-cycle data channel.	The device uses this signal to inform the I/O processor that it wants a word from memory (during a multi-cycle data channel transfer).
+1 → CA INH L	2BE1	If the I/O processor sees this signal during multi-cycle transfers, it inhibits normal incrementing of the device's assigned current address memory location.	This facility is used by such peripherals as DECtape and magnetic tape when they search for records. It is also useful during device checkout.

CHAPTER 5
WIRING PRACTICES

Noise within a digital system can be minimized by following simple wiring rules and using good working practices. The following paragraphs detail some of the common sources of noise and outline some rules which, if followed, will eliminate most problems.

5.1 WIRING RULES

DEC uses both fast and slow TTL logic in its modules. The slow 74-series modules have typical rise and fall times of under 8 ns. The high speed 74H-series gates show typical rise and fall times of under 5 ns. At the frequencies contained in these edges, the effects of inductance, mutual inductance, and capacitance as well as transmission line properties of the wiring become effective. Noise is generated on single lines by ringing during a rising or falling edge, and between lines through the coupling of mutual inductance and capacitance. Unwanted pulses generated by ringing and crosstalk can endanger input gates if they exceed the allowable input of 5.5V (Zener breakdown), or falsely trigger gates and flip-flops if they reach the threshold or band X region between 0.8V and 2.0V.

5.1.1 Single Line Waveform Degradations

The ringing problem in a single line can be reduced if the line is shortened or the loading is increased since a line with only a single load is more susceptible to ringing problems than one which is fully loaded. Further, a short line which has less inductance and capacitance is less vulnerable to ringing than a longer line. If it is impossible to shorten a line which has a noise problem, then it should be considered a transmission line and properly terminated. Table 5-1 shows recommended wire lengths for fast and slow gates under single and full load.

5.1.2 Termination Technique

When lines longer than the lengths recommended in Table 5-1 must be used, they should be terminated to ensure clean transitions and fast settling times. Figure 5-1 shows one way that this can be done. To handle the current drain of the termination and gate loading, a module which uses a 74H40 gate or equivalent which will drive 60 mA in the 0 state, is suggested. This module has 10 ns rise and fall times, and a very low high-state output impedance.

The line can be terminated with a 120Ω resistor to +5V and a 180Ω to ground (Figure 5-1). The 120Ω resistor to +5V draws at a maximum of 44 mA, and the 180Ω resistor to ground does not allow the 120Ω resistor to pull the high-state line potential any more than +3V. The gate does not have to supply any current to the resistors during its high-state. Even when terminated, 36 in. is the recommended maximum length of a wire with either fast or slow gates.

Gate loading in this scheme is limited to 10 units for reasons shown below.

Table 5-1
Recommended Maximum Wiring Lengths
With Single Wires

TTL Series Gates	Maximum Line Lengths	
	1 Unit Load	Max Unit Load
74 Series (Slow)	21 in.	26 in.
74 Series (Fast)	11 in.	18 in.

10 unit load = 1.6 mA max drain x 10 = 16 mA
Standard 120Ω ±5% resistor 5V/114 = 44 mA
74H40 = 60 mA drive at 0V Total = 60 mA

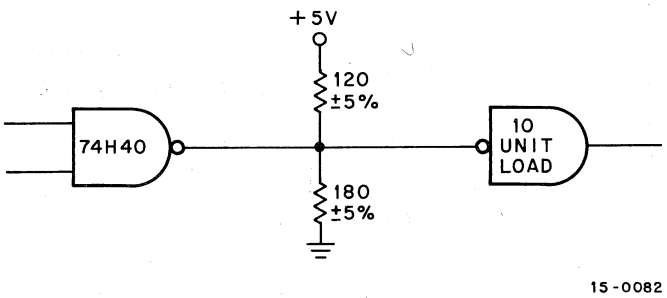


Figure 5-1 Terminating Long Wires

It is good engineering practice to examine every signal in a system for possible waveform degredations.

5.1.3 Crosstalk in Parallel Logic Lines

The present trend toward fast transition times and densely packed wiring adds to the probability of crosstalk noise in digital systems. This phenomenon varies with the high and low speed gates, the frequencies at which the gates are operating, the length of the wires involved, and even the relative positions of drivers and receivers. Crosstalk occurs as a result of coupling between parallel lines through their mutual inductance and capacitance. The coupling and noise injected from active to quiescent circuits increases as lines run parallel to each other over greater distances. Those gates which are faster respond to a greater bandwidth of noise and so are more vulnerable. Higher signal frequencies couple more readily than lower frequencies; and the relative position of the driver to the receiver on either line affects the magnitude of crosstalk. If both receivers and drivers of the two circuits are at the same end of the lines, as shown in Figure 5-2a, then capacitive and inductive crosstalk components cancel, and one becomes dominant. Alternatively, if the drivers and receivers oppose each other as in Figure 5-2b, then the components add, and the situation is more precarious.

Another variable to consider is the state of the quiescent or receiving circuit. Since the output impedance of the driving circuits is different for each state, it would be expected that crosstalk would also vary. Table 5-2 summarizes the recommended limits of parallel line lengths for digital circuits under the conditions outlined above. The effect of loading is also considered.

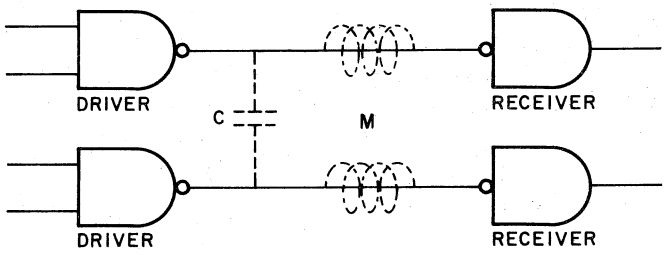


Figure 5-2a

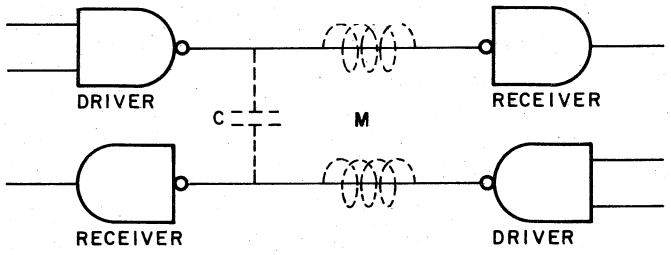


Figure 5-2b

Figure 5-2 Capacitance and Inductive Crosstalk Components

Table 5-2
Parallel Line Length Limits

Noise Sensitivity	Environment	Loading in Units	Limitations (Inches)	
			Slow Gates	Fast Gates
Zener Breakdown Danger (1,2)	Freq < 100 kHz	min max	36 >36	18 >36
	Freq 1 mHz	min max	33 >36	6 14
Noise Injection to "Band X" Region (3) Quiescent Line (High) (4)	Lines running same direction	min max	18 >36	>36 >36
	Lines running opposite direction	min max	5 5	5 5
Noise Injection to "Band X" Region – Quiescent Line (Low)	Lines running same direction	min	21	9
	Lines running opposite direction	min	12	6
	Average of both directions	max	26	10

(1) Zener Breakdown Endangerment is the potential breakdown of a receiver gate input diode when the reverse voltage at the input is greater than 5.5V. Induced noise reaching this potential results in the same endangerment.

(2) The noise generated when endangering Zener breakdown is frequency sensitive and not line-reversing sensitive.

(3) "Band X" Region is the voltage range between a high state and a low state at which the threshold level can be situated at any point in this range. The specification range is between 0.8V and 2.0V.

(4) The quiescent line is the line that the noise is induced on. The (low) or (high) designates its static level.

5.2 PROPERTIES OF #30 AWG WIRE

The propogation delay of typical #30 AWG wiring (used extensively in DEC systems) is 1.5 ns/ft (4.5 ns/m). Typical wiring has a characteristic impedance of 120Ω and TTL voltage transitions are 2.5 to 4.0V in amplitude, so that the current available at the end of the wire for rising waveforms is 20 to 30 mA until reflections propogate, regardless of the source current available.

5.3 THE GROUND SYSTEM

5.3.1 DC Ground System

A good dc ground system is essential to reliable logic operation. The following system is recommended when the H900 Mounting Panel series is used.

a. Bus the ground pins C2, T1 in each module row with 933 horizontal bussing strip.

b. Bus the power pins A2 in each module with the same type of strip.

c. Tie the ground bus strips to chassis ground about every 2 in., using solid wire with spaghetti insulation.

d. Wirewrap all grounded pins together for each vertical module row.

This ground mesh will form a stable box for satisfactory logic performance. Figure 5-3 illustrates the system.

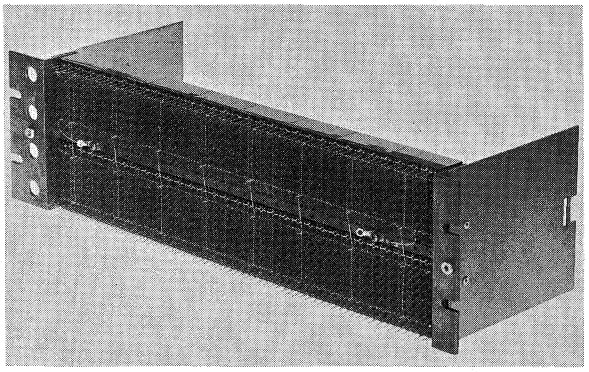


Figure 5-3 The Ground System

5.3.2 AC Ground System

To keep electrical noise and potential differences under control, the following ground system is recommended although other grounding methods may prove adequate.

When two cabinets are joined together, they should be bonded together electrically by running a #4 gage conductor or several copper mesh straps between the two cabinets. Ordinary stranded #4 gage wire is adequate for this purpose, although #4 gage welding cable (extra flexible stranding) may be preferred by some.

Upon installation, the purchaser should supply a good earth ground connection to the central processor through #4 gage copper wire or equivalent. In general, an adequate earth ground is provided by a steel beam of a building frame or a large water pipe. Auxiliary units such as the line printer and card reader should be grounded to their associated control cabinets with #4 gage copper wire (#6 wire can be used in this case if desired).

The type of the earth ground necessary depends on the use of the system. A system involving a digital-analog interface usually requires that the digital system ground be connected to the analog system ground at a single point, often at the analog-digital interface. A good ground connection is usually required in these cases. In small systems where no analog interface is involved, the grounding provided by a large electrical conduit may be adequate, although electrical conduit systems often are connected together poorly in terms of a low resistance path to ground. In large systems, additional connections to earth ground may also be advisable. All of these ground connections are additions to (not substitutes for) the ground leads carried along through the various signal buses (memory, I/O multiplexer and channel) and the ground conductors contained in the power (main) cables. The green wire in the power cable must also be returned to ground, usually through the conduit of the electrical distribution system.

In general, ground conductors should follow the path of the data buses through the system (i.e., in parallel to the memory buses, the I/O bus, the channel bus, etc.), see Figure 5-4.

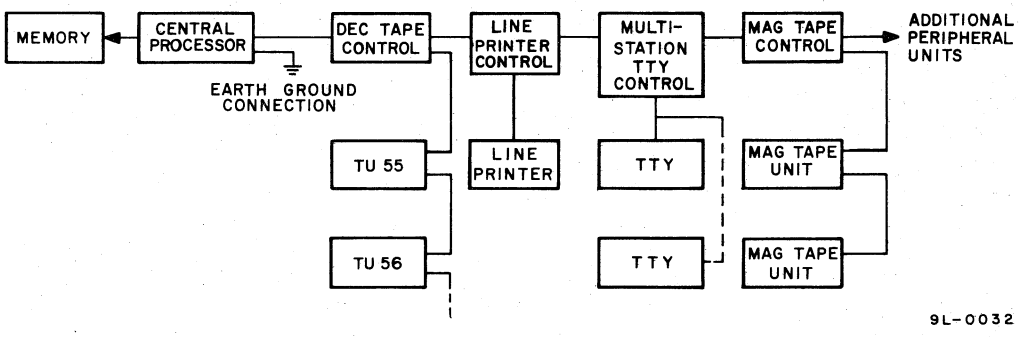


Figure 5-4 Typical Ground Mesh System

5.4 CABLES IN DIGITAL SYSTEMS

The wiring rules of the previous section restrict single wires to 36 in., when terminated, and certain parallel runs to only 5 in. Signals to be transmitted 75 ft down the PDP-15 bus, a set of parallel lines and therefore subject to all of the problems inherent in such wires, present a special problem.

It is possible to construct a cable which will allow pulses to travel 75 ft at 1 mHz, yet not interfere with neighboring lines. This is done by establishing a tactical geometry of wires within the cable to isolate sensitive signals from critical transitions and setting up twisted pair alternate grounding. It uses special insulation, well designed drivers and receivers, and, finally, termination techniques.

5.4.1 The PDP-15 Positive I/O Bus Cable

There are three types of cable assemblies which can be used on a PDP-15 system. The positive cable, called the BC09B, is designed to run between PDP-15 positive logic devices. The other two, the BC09A and BC09C are used when negative logic PDP-9 peripherals are connected to the system. In this section, the BC09B will be described. Chapter 6 describes the uses of all three.

5.4.1.1 Cable Geometry – The BC09B cable (the same cable as the BC09A and BC09C assemblies use) contains 36 pairs of twisted wires, each with a characteristic impedance of 68Ω. The wires are distributed in three concentric layers shown in Figure 5-5. The inside layer A contains 6 pairs, the middle layer B, 12 pairs, and the outer layer C, 18 pairs. Each layer is twisted independently around the center core of the cable. Layers A and B are twisted to the right, and layer C to the left.

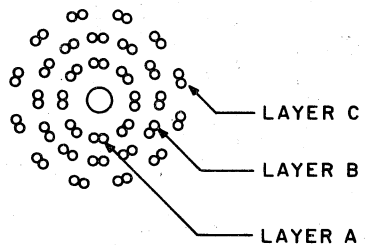


Figure 5-5 Wire Distribution

The PDP-15 I/O bus consists of two of these cables terminated at either end by one set of double-height double sided cards. The signals are distributed in the two cables in the following manner.

CABLE 1

LAYER A	I/O SYNC H, IOP1 H, IOP2 H, IOP4 H, SKIP RQ L, PROG INT RQ L.
LAYER B	DS0 H – DS5 H, SD0 H, SD1 H, RD RQ L, RD STATUS H, I/O PWR CLR H, SING CY RQ L.
LAYER C	I/O BUS 00 L – I/O BUS 17 L

CABLE 2

LAYER A	WR RQ L, INC MB L, + 1 → CA INH L, API 0 RQ L, API 0 GR H, API 0 EN H
LAYER B	API 1 RQ L, API 1 GR H, API 1 EN H, API 2 RQ L, API 2 GR H, API 2 EN H, API 3 RQ L, API 3 GR H, API 3 EN H, DCH RQ L, DCH GR H, DCH EN H.
LAYER C	I/O RUN H, I/O OFLO H, DATA OFLO H, I/O ADDR 03 L – I/O ADDR 17 L.

With this distribution of signals in the two cables, the maximum amount of crosstalk picked up under normal operating conditions in any given wire is .45V, with an average of about .22V. The cable is described in more detail in Chapter 6.

5.4.1.2 Cable Connections and Terminations – Each device on the PDP-15 I/O bus (positive) has an input and an output cable. The two cards must be placed next to each other and short wires run horizontally to connect the output pins to the input pins (all except the ENABLE signals that are being used). For the positive bus, the last device on the line should replace its output receptacle with a 68Ω Terminator Card, Type M909.

5.4.1.3 Cable Drivers and Receivers – Signals are driven down the PDP-15 I/O bus by specially designed type M622 modules, and are received by high input impedance type M510 modules. The specifications for each are given in Chapter 2.

When DEC logic is not used in the interface, it is important that the drivers and receivers used match the specifications of the M622 and M510 modules.

5.5 GENERAL WIRE INFORMATION

5.5.1 Wire Types

The selection of the correct wire type is the primary requirement for a properly made connection. There are two basic types of insulated wire used at Digital.

Solid wire consists of a single copper conductor, usually plated with a coating of nickel, silver, or tin. Solid wire is easy to use because it is easy to strip, form, and it stays in place once formed. Its disadvantages are that it breaks easily if bent sharply or if subject to constant flexing or vibration. Solid wire, therefore, should be used only on fixed panels or assemblies, and never used in cables or to connect two different assemblies which may move or vibrate with respect to each other.

Stranded wire consists of a series of small solid wires twisted together. There are seven or nineteen separate conductors in a typical stranded wire. It has the advantage of being very flexible and difficult to break and, therefore, is ideal for cables, patch cords, and for making connections between moving assemblies. Stranded wire's disadvantages are that it requires more ties and clamps to hold the wire in place and more care in handling, as the strands may become spread.

5.5.2 Wire Insulation

There are three basic types of insulation used at Digital: PVC (poly-vynal-chloride), TEF (teflon, type E) and KYNAR.

PVC is a rubber-like compound with fair mechanical strength, low cost and excellent stripping characteristics. The major disadvantage of PVC is that it melts easily and tends to draw from the heat source. For these reasons, PVC should never be used in a soldered connection.

Teflon is a greasy feeling plastic compound with good mechanical properties, fair stripping characteristics, and excellent resistance to heat. Its major disadvantages are low cut-through resistance and a tendency to cold-flow. For these reasons, teflon is preferred for soldered connections, but care must be exercised so that it is not pulled tightly across sharp edges. The cold-flow characteristic will cause the insulation to move away from the point of pressure. Careless routing of teflon wire may result in exposed conductors after an indeterminant period of time.

KYNAR is a plastic-like compound with good mechanical properties, average stripping characteristics, and good resistance to heat. It has much higher resistance to cut-through than teflon and doesn't cold-flow. KYNAR must be used where wires may be pulled over sharp edges (i.e., hand wire-wrap).

The qualities of these types of insulation are summarized in Table 5-3.

Table 5-3
Properties of Insulation Types

Property	PVC	Teflon	KYNAR
Resistance to Soldering Temperature	Poor	Excellent	Good
Mechanical Strength	Fair	Good	Good
Resistance to Cut-Through	Average	Poor	Good
Resistance to Cold-Flow	Average	Poor	Good
Ease of Stripping	Excellent	Fair	Average

5.5.3 Wire Insulation Stripping

The correct stripping of insulation is one of the most important steps in making an acceptable connection. The insulation must be completely removed, yet the conductor or conductors must not be nicked or cut.

Using the proper tools makes stripping insulation relatively easy. Diagonal cutters or knives are not the proper tools. The use of either of these tools will usually result in a damaged conductor.

The correct tool to use is a wire stripper. There are two preferred types available. One type has a fixed adjustment with a different size opening for each wire size (see Figure 5-6). It is virtually impossible to nick the conductor if the proper size opening is used. The second and more common type of wire stripper has one opening and an adjustable set screw, see Figure 5-7. This type of cutter is easy to use but requires constant checking to insure undamaged conductors. With either type of tool, it is essential to regularly check that the conductor is not being nicked. This is easily done by stripping the wire twice and checking the conductor for damage at the first point of stripping (see Figure 5-8).

A properly stripped piece of wire has no nicks, cuts or frayed insulation. Under no circumstances can nicks reduce the area of a conductor by more than 10%. Stranded wire containing 19 conductors may have a maximum of 1 broken conductor; stranded wire containing 7 conductors may have no broken conductors.

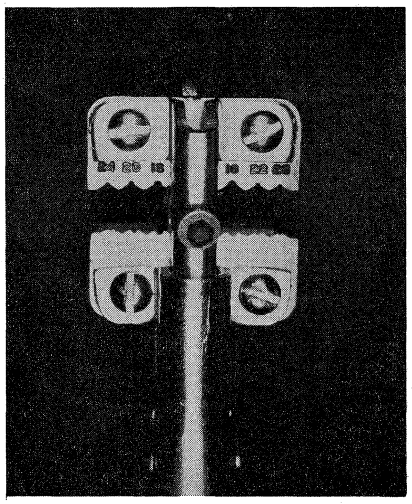


Figure 5-6 Non Adjustable Wire Stripper

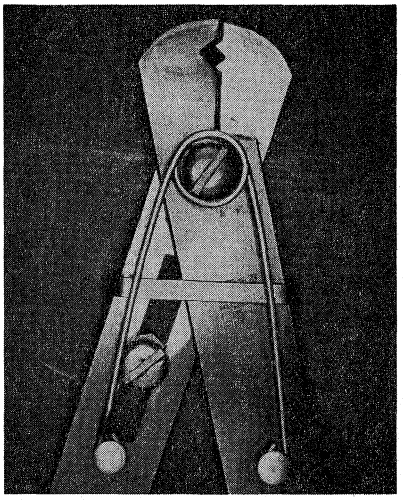


Figure 5-7 Adjustable Wire Stripper (Use With Caution)

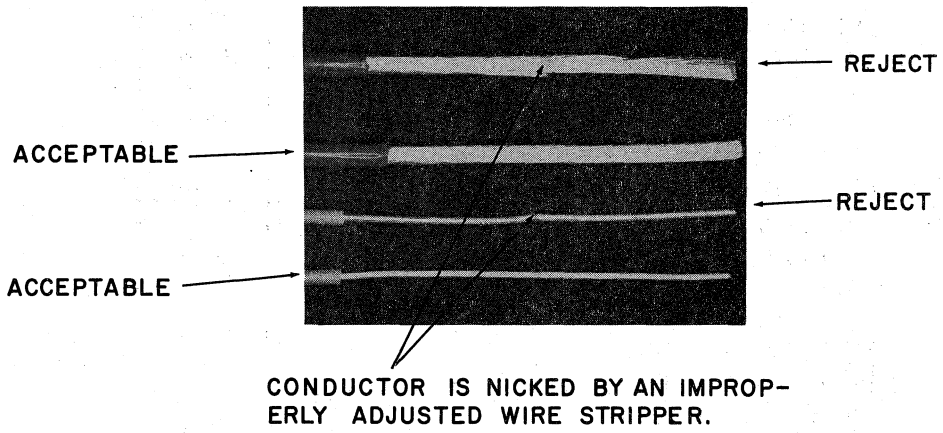


Figure 5-8 Stripping Damage

5.5.4 Wire Data

The standard wire sizes, colors, and insulation types presently used at DEC are given in Table 5-4. The table contains the AWG (American Wire Gauge) size, the DEC part number, the colors stocked, the number of strands in the stranded types, and the typical use of each. Some wire types are available as a twisted pair; that is, two different color wires of the same size wound together. This is very handy for ac-power wiring and special uses. Most part numbers have a two digit number at the end.

All hook-up wire (excluding twisted pairs) will carry the two digit class code "91", the five digit body code and a two digit modifier.

The first digit of the modifier designates the base color (0 = black, 1 = brown ..... 9 = white), the second digit designates the color of the tracer stripe. A solid colored wire is designated by both digits being the same such as "–66" since (it is impossible to have a blue wire with a blue stripe, etc.).

Digit	Color
0	Black
1	Brown
2	Red
3	Orange
4	Yellow
5	Green
6	Blue
7	Violet
8	Gray
9	White

EXAMPLE: 91-07586-92 White wire with a red stripe

Table 5-4
Solid and Stranded Wires

Stranded Wire (Teflon Type 'E' Insulation)				
Size	Part Number	Color	Strands	Note
#10	91-07390-00	BLK	19	
	91-07390-22	RED	19	
	91-07390-66	BLU	19	
	91-07390-99	WHT	19	
#14	91-07370-00	BLK	19	
	91-07370-11	BRN	19	
	91-07370-22	RED	19	
	91-07370-33	ORG	19	
	91-07370-44	YEL	19	
	91-07370-55	GRN	19	
	91-07370-66	BLU	19	
	91-07440-04	BLK/YEL	19	Twisted Pair
	91-07440-29	RED/WHT	19	Twisted Pair
	91-07440-09	BLK/WHT	19	Twisted Pair
	91-07440-03	BLK/ORG	19	Twisted Pair
	91-07440-35	ORG/GRN	19	Twisted Pair
	91-07440-06	BLK/BLU	19	Twisted Pair
#18	91-07360-00	BLK	19	
	91-07360-11	BRN	19	
	91-07360-22	RED	19	
	91-07360-33	ORG	19	
	91-07360-44	YEL	19	
	91-07360-55	GRN	19	
	91-07360-66	BLU	19	
	91-07360-99	WHT	19	

Table 5-4 (Cont)
Solid and Stranded Wires

Stranded Wire (Teflon Type 'E' Insulation)				
Size	**Part Number**	**Color**	**Strands**	**Note**
	91-07430	RED/WHT	19	Twisted Pair
#22	91-07350-44	YEL	7	
	91-07350-55	GRN	7	
	91-07350-66	BLU	7	
	91-07350-77	WHT/VIO	7	
	91-07350-90	WHT/BLK WHT/BLK	7	
	91-07350-91	BRN WHT/BRN	7	
	91-07350-92	RED	7	
	91-07350-93	ORG	7	
	91-07350-94	YEL	7	
	91-07350-95	GRN	7	
	91-07350-96	BLU	7	
	91-07350-97	VIO	7	
	91-07350-98	GRY	7	
	91-07420-40	YEL/BLK	7	Twisted Pair
	91-07420-25	RED/GRN	7	Twisted Pair
	91-07420-29	RED/WHT	7	Twisted Pair
	91-07420-84	GRY/YEL	7	Twisted Pair
	91-07420-15	BRN/GRN	7	Twisted Pair
	91-07420-05	BLK/GRN	7	Twisted Pair
	91-07420-21	RED/BRN	7	Twisted Pair
	91-07420-26	RED/BLU	7	Twisted Pair
	91-07420-39	ORG/WHT	7	Twisted Pair
#26	91-07636-00	BLK	7	
	91-07636-11	BRN	7	
	91-07636-22	RED	7	
	91-07636-33	ORG	7	
	91-07636-44	YEL	7	
	91-07636-55	GRN	7	
	91-07636-66	BLU	7	
	91-07636-91	WHT/BRN	7	
	91-07636-92	WHT/RED	7	
	91-07636-93	WHT/ORG	7	
	91-07636-94	WHT/YEL	7	
	91-07636-95	WHT/GRN	7	
	91-07678-98	WHT/GRY	7	Twisted Pair

Solid Wire				
Size	**Part Number**	**Color**	**Insulation**	**Note**
#24	91-07470-10	BLK	TEF(E)	
	91-07470-22	RED	TEF(E)	
	91-07470-33	ORG	TEF(E)	
	91-07470-44	YEL	TEF(E)	
	91-07470-55	GRN	TEF(E)	
	91-07470-66	BLU	TEF(E)	
	91-07470-99	WHT	TEF(E)	
	91-07586-44	YEL	TEF(E)	Wire Wrap Machine Use only 50,000 ft drums

Table 5-4 (Cont)
Solid and Stranded Wires

Stranded Wire (Teflon Type 'E' Insulation)				
Size	**Part Number**	**Color**	**Insulation**	**Note**
	91-07497-09	BLK/WHT	TEF(E)	Twisted Pair
	91-07688-55	GRN	KYNAR	ECO's Wire Wrap
	91-07688-66	BLU	KYNAR	Hand Wrap
	91-07688-99	WHT	KYNAR	GND Lug Wrap Connection
#26	91-07694-44	YEL	KYNAR	Wire Wrap Machine
	91-07694-55	GRN	KYNAR	ECO's Wire Wrap
	91-07694-66	BLU	KYNAR	Hand Wrap
	91-07694-99	WHT	KYNAR	GND Lug Connection
#30	91-05740-44	YEL	KYNAR	Wire Wrap Hand Machine
	91-05740-55	GRN	KYNAR	ECO's Wire Wrap
	91-05740-66	BLU	KYNAR	Machine Wrap Repairs
	91-05740-99	WHT	KYNAR	GND Lug Connections

Certain colors have been assigned to standard wire applications in all Digital products. They are summarized in Table 5-5.

Table 5-5
Wire Color and Application Chart

Use		Color	Note
AC Power Cord	Hot	Black	All twisted together with a solid covering
	Neutral	White	
	Machine Ground	Green	
Internal Machine AC Power	Hot	Red	Twisted Pair
	Neutral	White	
	Machine Ground	Black	Also DC Ground
PDP-8, 8/S, 9, 9/L, 10, A,B,G,R, S and W Series Modules	+10V Fixed Power	Red	
	+10V Marginal Power	Orange	
	Ground	Black	
	-15V Fixed Power	Blue	
	-15V Marginal Power	Green	
8/I, 8/L	+15 or +30	Orange	
	+5	Red	
	-15	Blue	
	-30	Green	
	Ground	Black	
	Auto-Tap	Blk & Wht	Twisted Pair
Hand or machine wire wrap		Yellow	
Wire-wrap ground lug connections		White	
Wire-wrap ECO's		Green	
Wire-Wrap Corrections		Blue	Done at time of wrapping

5.6 SOLDERLESS CONNECTIONS

A solderless connection is a mechanical technique for fastening a wire to a post of a lug. Two basic types of solderless connections are used at Digital; crimped and wire-wrap. Acceptable connections can only be obtained by the correct use of the proper tools.

5.6.1 Crimped Connections

'Crimped' connectors come in many sizes and shapes. Typical crimped connectors are shown in Figure 5-9.

In all these connectors, a metal body is compressed around the conductor. To guarantee a reliable joint it is absolutely essential that the proper size wire, connector, and tool always be used, see Table 5-6.

The major cause of unreliable crimped connections is too loose a crimp. A loose crimp occurs when hand tools are used which can be released by the operator before the joint is fully made. For this reason, only ratchet type or air driven hand tools should be used. Figure 5-10 shows the characteristics of acceptable and unacceptable crimped connections.

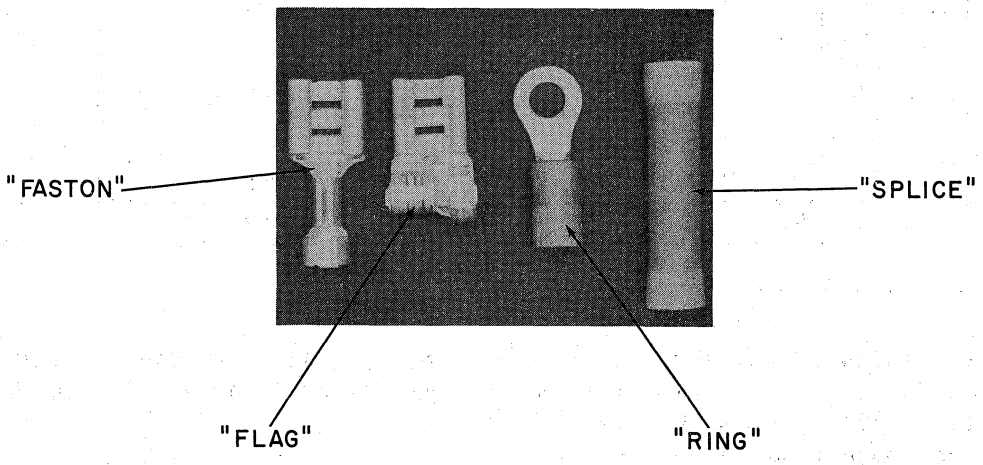


Figure 5-9 Crimped Connectors

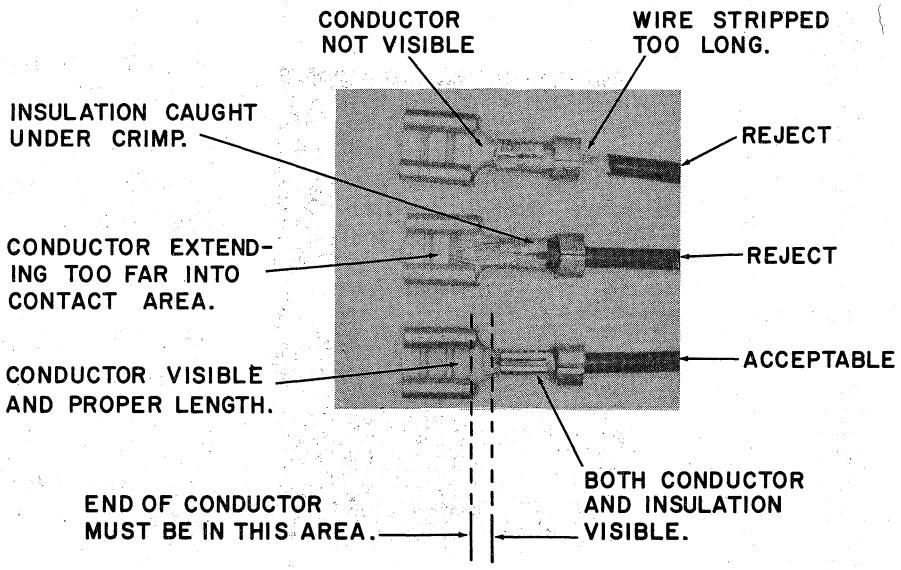


Figure 5-10 Acceptable and Unacceptable Crimped Connections

Table 5-6
Wire Size, Connector Type and Tools To Use For Crimped Connections

Connector Type	Wire Size	Screw Size	Mfg. Part Number	DEC Part No.	Tool	Pull Test (Minimum,Pounds)
FASTON	#10–#14	NA	AMP-4150	90-07669	AMP-47417	60
	#18–#22	NA	AMP-62025-1	90-06997	AMP-47417	15
MALE-FEMALE ADAPTER	#10–#14	NA	ARC-LESS 300H21A	90-07925	ARC-LESS 100P1	60
RING	#10	#10	ARC-LESS 50368 (Yellow)	90-07926		
	#14	#6	ARC-LESS 50325 (Blue)	90-07927	ARC-LESS 100P1	60
		#10	ARC-LESS 50364 (Blue)	90-07928		
	#18–#22	#6	ARC-LESS 50321 (Red)	90-07929	ARC-LESS 100P1	15
		#10	ARC-LESS 50360 (Red)	90-07930		
SPLICE	#18	NA	AMP-34071	90-06702	AMP-59250	15
FLAG	#14–#18	NA	AMP-60102-2	90-07931	AMP-47417	15

Stripping length for all connectors above is 1/4 inch.

5.6.1.1 Insulating Crimped Connections – In many applications it is necessary to place insulation around the end of the crimped connection. The most commonly used insulation is heat-shrinkable spaghetti (shrinky). Before putting a 'shrinky' over a crimped connection, carefully inspect the joint. Heat must be applied carefully or the 'shrinky' will discolor and crack. The correct tool to use is a heat gun made by Master Appliance Corp., Model HG 501 lg. The heat gun should be held four to six inches from the work.

The spaghetti will be fully shrunk in about 4 seconds, with the cover on the side of the heat gun fully open. Figure 5-11 shows properly and improperly shrunk spaghetti.

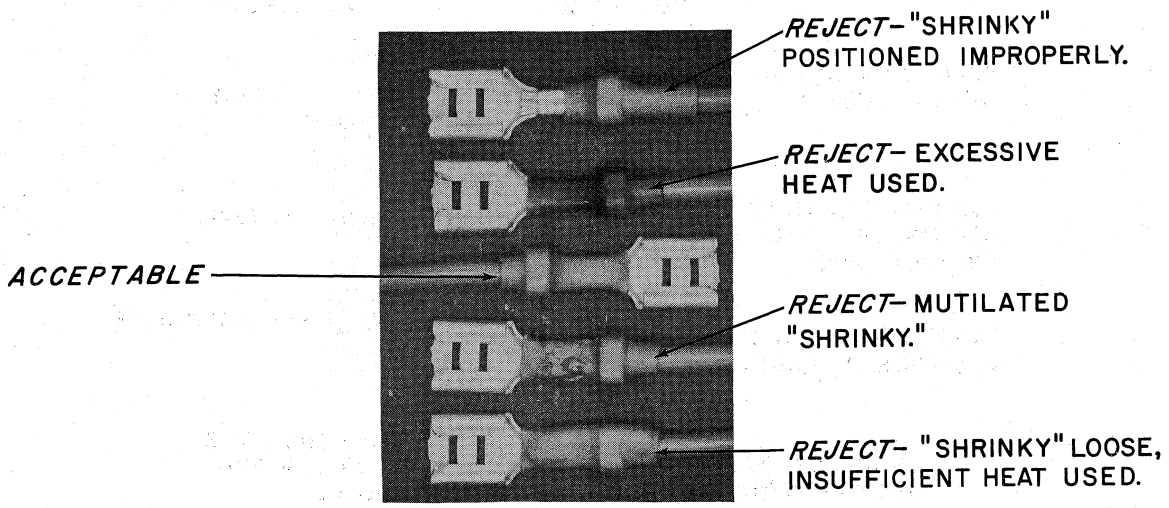


Figure 5-11 Properly and Improperly Shrunk Spaghetti

5.6.1.2 Tool Maintenance – Tool maintenance is very important for insuring properly made crimped connections. It is essential to test the quality of the crimped joints routinely. All tooling should be checked at least monthly or every 10,000 joints, whichever occurs first. To test the tooling, crimp four samples of each size lug, two on the largest and two on the smallest size wire and perform a pull test. All the joints must meet the minimum pull test requirement which are spelled out in the list in Table 5-6.

5.6.2 Wire Wrap

Wire wrapping is used for fairly small (#24 to #30) solid wire in repetitive applications. It evolved as the only economical way to easily make the large number of connections required in today's computers.

A wire-wrapped joint is made by tightly wrapping a bare conductor around a rectangular post. The post has sharp edges which cut into the wire as it is wrapped. In a properly made joint, the pressures at the edges of the post are high enough to cause a cold weld to take place. For every complete turn around the post, four welds are made, one at each corner. In a typical joint, the wire makes six full turns and is therefore fastened to the post at twenty-four places.

Figure 5-12 illustrates the terminology used in wire wrapping.

5.6.2.1 Wire Wrap Requirements

a. The most important factor in making reliable wire-wrap joints is the use of the correct tools. Always use the bits and sleeves outlined in Table 5-7.

b. Check the condition of the wire-wrap post prior to making the connection. It must meet the following requirements:

(1) No more than a 15° twist.
(2) No more than 10° from the vertical.
(3) No solder build-up over .003 inches.

c. The wrap shall meet the following requirements:

(1) 4-1/2 to 6 turns of bare wire for #24 wire (.031 x .062 post)*
7 to 9 turns of bare wire for #24, #30 wire (.025 x .025 post)*
(2) 1/2 to 2 turns of insulated wire at beginning of wrap.
(3) Spacing between wraps; maximum 1/2 the stripped wire diameter.
(4) No overlap of bare wire, one conductor on top of another. The first wrap of insulated wire may overlap the last turn of uninsulated wire in a wrap below.
(5) The last turn of a third level wrap must not extend into the tapered part of the post.
(6) The 'end tail' may not extend more than one wire diameter beyond the wrapped joint.

*Number of turns see Table 5-7.

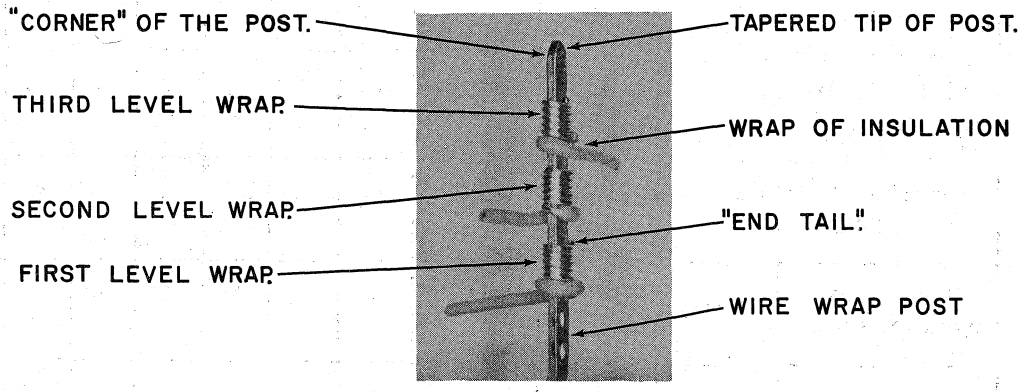


Figure 5-12 Wire Wrap Terminology

d. The wire used shall meet the following requirements:

(1) Stripped length – See Table 5-7.

(2) Stripped conductor must be free from any scrapes or nicks. The portion of a conductor which has been removed from a post may never be reused.

(3) The insulation must be free from nicks or cuts.

(4) The insulation must be 'KYNAR'. (Teflon acceptable for auto-wrapping.)

(5) The color code below should be used for all wire wrapping.

Ground Lugs	– White
Other Hand Wrap	– Blue
Automatic Machine Wrap	– Yellow

Table 5-7
Chart of Tools, Wire Size, and Wrapping Requirements for Connectors

Wire Gauge	Connector Type	Stripping Length (Inches)	Gardener Denver Bit #	Gardener Denver Sleeve #	Turns of Wire		Turns of Insulation		Pull Test Requirements (lbs)		Gardener Denver Unwrapping Bit
					Min	Max	Min	Max	Min	Max	
	144 Pin (H800W) Part #12-02244 *	1-1/2	26263	18840	4-1/2	6	1/2	2	7	35	500130-L-E
	18 Pin (H802) Part #12-02625 *	1-1/2	26263	18840	4-1/2	6	1/2	2	7	35	500130-L-E
#24	288 Pin (H803) Part #12-05348 **	1-1/2	505415	502129	5	7	1/2	2	4	30	500130-L-E
	36 Pin (H807) Part #12-09123 **	1-1/2	505415	502129	5	7	1/2	2	4	30	500130-L-E
	144 Pin (H800W) Part #12-02244 *	1-5/8	26263	18840	5	7	1/2	2	6	35	500130-L-E
	18 Pin (H802) Part #12-02625 *	1-5/8	26263	18840	5	7	1/2	2	6	35	500130-L-E
#26	288 Pin (H803) Part #12-05348 **	1-5/8	505415	502129	6	8	1/2	2	4	30	500130-L-E
	36 Pin (H807) Part #12-09123 **	1-5/8	505415	502129	6	8	1/2	2	4	30	500130-L-E
	288 Pin (H803) Part #12-05348 **	1-1/32	504221	500350	7	9	1/2	2	4	30	505084-L 505244-L-R
#30	36 Pin (H807) Part #12-09123 **	1-1/32	504221	500350	7	9	1/2	2	4	30	505084-L 505244-L-R

*Cross Section of Pin .031 x .062 inches
**Cross Section of Pin .025 x .025 inches

e. The routing of the wire must meet the following requirements:

(1) The wire must be positioned such that subsequent routing of the wire does not tend to unwrap the joint.

(2) Wires should be routed for the shortest practical length.

(3) Wires must start and end on the same level. 'Level jumping' should not be allowed.

(4) Allow enough slack when routing a wire around an unused post to allow a wire wrap tool to be placed over the post without damaging the wire passing by.

f. All repairs, and rework should be performed in the following manner:

(1) A wire-wrap joint should never be forced to lower level.

(2) A wire-wrap joint should never be reused.

(3) When changing a wire-wrap joint, either strip a new section of the wire or replace it entirely.

(4) Always use the correct color KYNAR wire.

(5) When working on a panel containing modules, be sure a drain wire is connected between an air-driven tool and the chassis ground, and that AC-powered tools are operated from an isolation transformer.

5.6.2.2 Wire-Wrap Post Location Standard – It is necessary to be able to locate the 'from' and 'to' end of any wire-wrap 'run'. The 'from' and 'to' is indicated by a series of six letters and numbers such as 1A24F1. The first number identifies the bay or cabinet (1A24F1). Since most small systems have only one bay the first digit is usually not listed. The first letter identifies the horizontal row (A24F1). Refer to Figure 5-13. The rows are lettered starting with 'A' from the top of the bay. The second and third digits identify the vertical column or 'slot' in a particular row (A24F1). There may be as many as 40 or as few as 12 'slots' in a row. The second letter identifies the pin (wire-wrap post) in a given row and column (A24F1). There are 18 letters used: A through V. The letters G, I, O, and Q are not used. There are twice as many 30 gauge pins as 24 gauge pins in a slot. The last digit (1A24F1) identifies the left or right pin group in a 30 gauge socket (see Figure 5-14). A '1' indicates the left hand group and '2' the right hand group. This last digit is not listed for a 24 gauge connector.

A typical 'run' of 24 gauge wire would be:

From A24F to C13R (thus, a 24 gauge wire would run from row A, slot 24, pin F, to row C, slot 13, pin R).

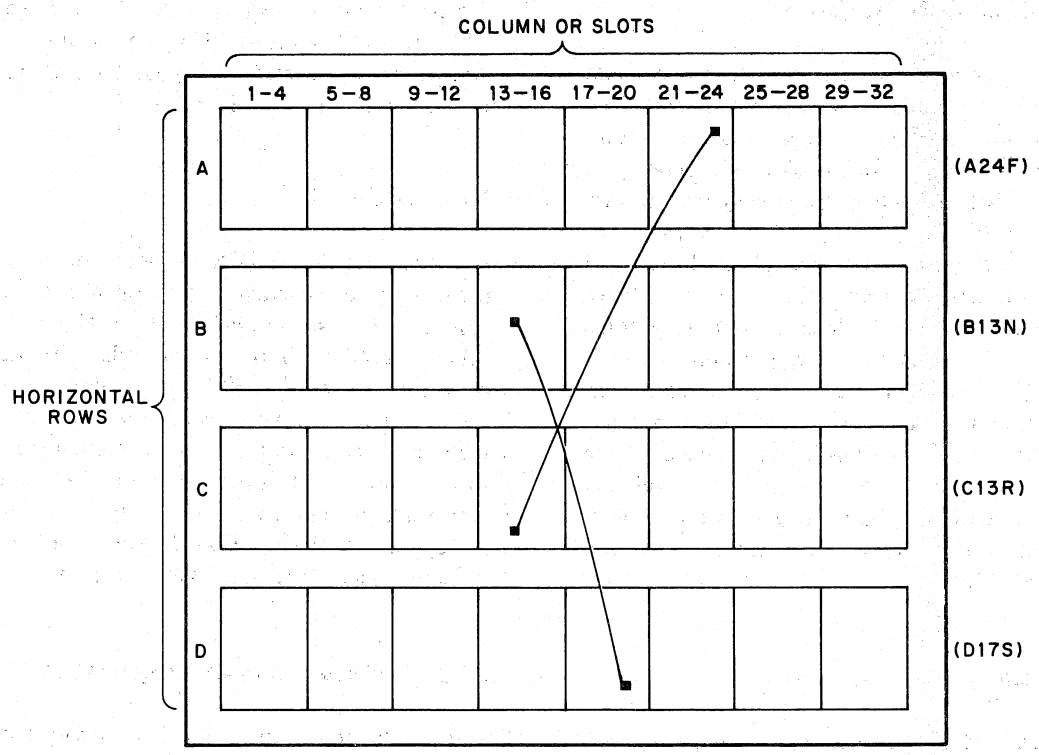


Figure 5-13 Wire Wrap Post Location Standard 12-0020

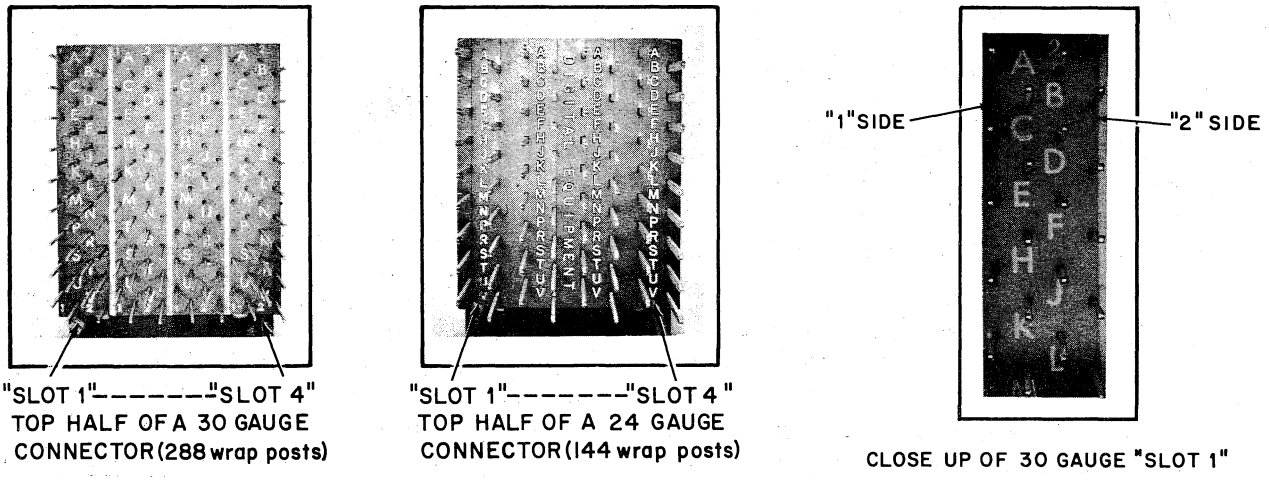


Figure 5-14 Examples of 30 and 24 Gauge Connectors

A typical run of 30 gauge wire would be:

From B13N1 to D17S2 (thus, a 30 gauge wire would run from row B, slot 13, pin N1 to row D, slot 17, pin S2).

Every fifth slot, starting at slot 5, is identified by a piece of red spaghetti pushed over pin A1. It is fairly easy, therefore, to locate a particular slot by counting from the nearest red marker.

5.7 SOLDERED CONNECTIONS

In a soldered connection, a metal alloy is melted, flows around, and bonds to all parts of the joint. The most commonly used alloy or solder is a 60% lead, 40% tin mixture. This type of solder is simply called 'sixty-forty'. Solder melts at a relatively low temperature, about 375° F. In a properly made joint, the solder actually forms a chemical bond to all the metal surfaces it contacts. A clean surface is essential to assure a good chemical bond. Since all surfaces become contaminated by exposure to air, a cleaning agent is necessary. These cleaning agents are called fluxes. Most solder contains the flux. The flux is released from the core of the solder as heat is applied. Since the flux melts at a lower temperature than solder, it cleans the surfaces of the joint just before the solder flows. Rosin flux is the only acceptable type of flux for hand soldering.

5.7.1 Soldering Irons

The heat required to melt the solder is usually generated by an electrical soldering iron. Many soldering irons have a built in thermostat which automatically regulates the temperature of the tip.

The temperature required to solder a joint is the same, regardless of its size. A larger connection does, however, require a large soldering iron. The 'size' of a soldering iron is measured by its wattage rating; a higher wattage soldering iron is required for a larger connection.

The iron used most often at DEC is a Weller, 48W, TCP-1 with a Weller Model PU-1 power unit. This unit has a thermostat control, safety holder, isolation transformer and tip cleaner, all in one unit.

5.7.2 Solder

It is essential that only the proper type of solder be used. Digital has standardized on a 63/37 (60/40), rosin-core wire solder made by Alpha Metals, Inc. This solder is available in five pound rolls. The wire solder is the same diameter as #20 wire. This solder contains a fairly mild flux, which is adequate for soldering all reasonably clean joints. Solder containing stronger flux should never be used. Flux is actually a mild corrosive and should always be cleaned from the joint. Always check the label on the roll of solder. Never use any solder which is marked acid core!

5.7.3 Soldered Connection Requirements

a. The solder in a joint is necessary for electric

continuity, not mechanical strength. Prior to soldering, a joint must be wrapped tightly and all wires must contact at least three corners of the lug or post. The joint must be tight enough to keep the wire from moving while the solder is cooling (see Figure 5-15).

b. The wire must be dressed in such a manner as to minimize strain on the solder joint.

c. All wires on a joint must contact the post or lug. Do not wrap layers of wire over other wires (see Figure 5-16). The lower layer may not solder and will be impossible to inspect.

5.7.4 Use of Stranded Wire (See Figure 5-17)

a. Stranded wire must be 'tinned' before use. Tinning involves coating and impregnating the wire with solder.

b. Dipping the end of the wires in a solder pot is the easiest way to 'tin' a large number of ends. Always clean the residue (dross) off the surface of the molten solder before immersing the wire.

c. When tinning, do not allow the solder to run up under the insulation. This tends to trap flux under the insulation which may eventually cause corrosion.

d. Tin only that portion of the wire which is going to be in the joint. Tinning too much of the wire makes it stiff and decreases its flexibility.

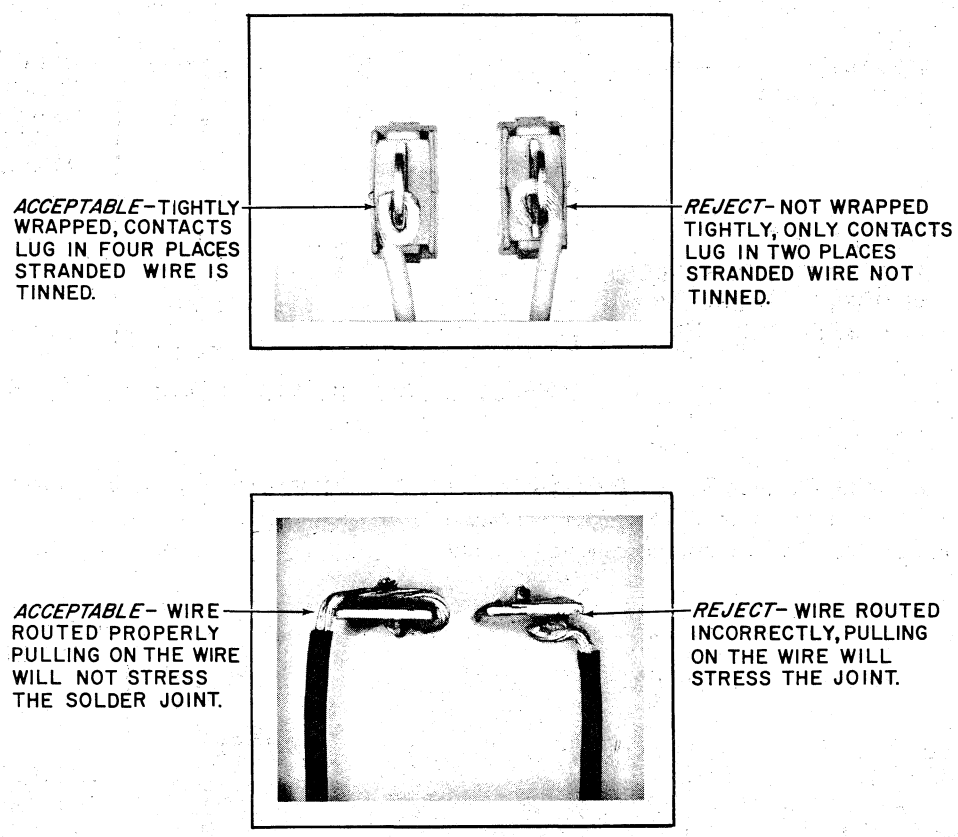


Figure 5-15 Example of Wire Joint Before Soldering

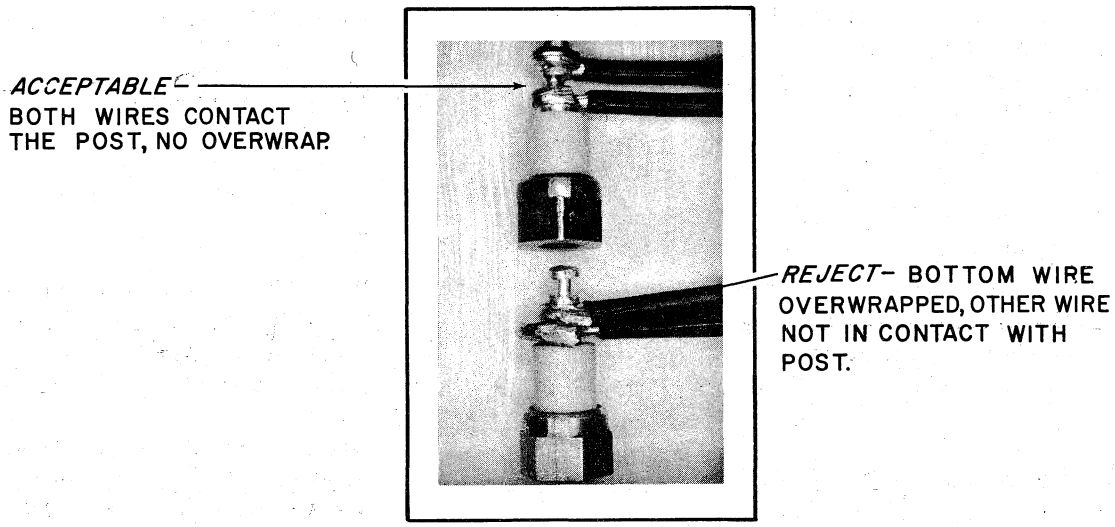


Figure 5-16 Example of Soldering Two Wires on a Post

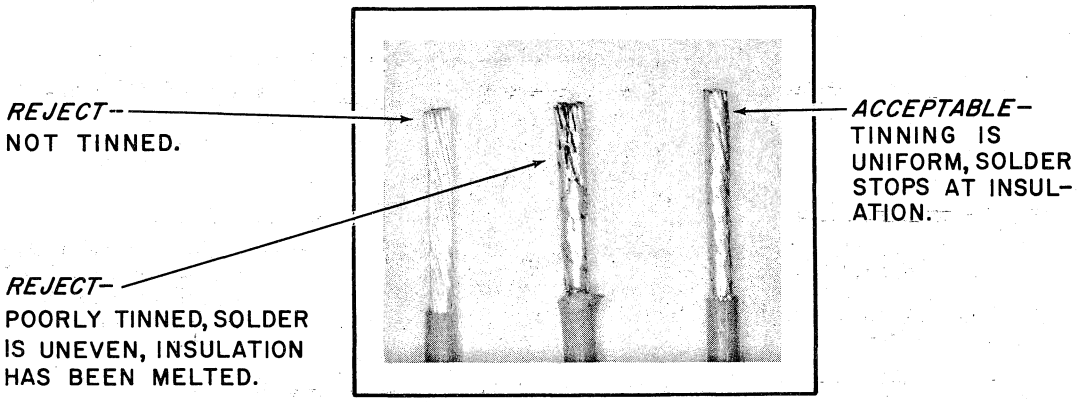
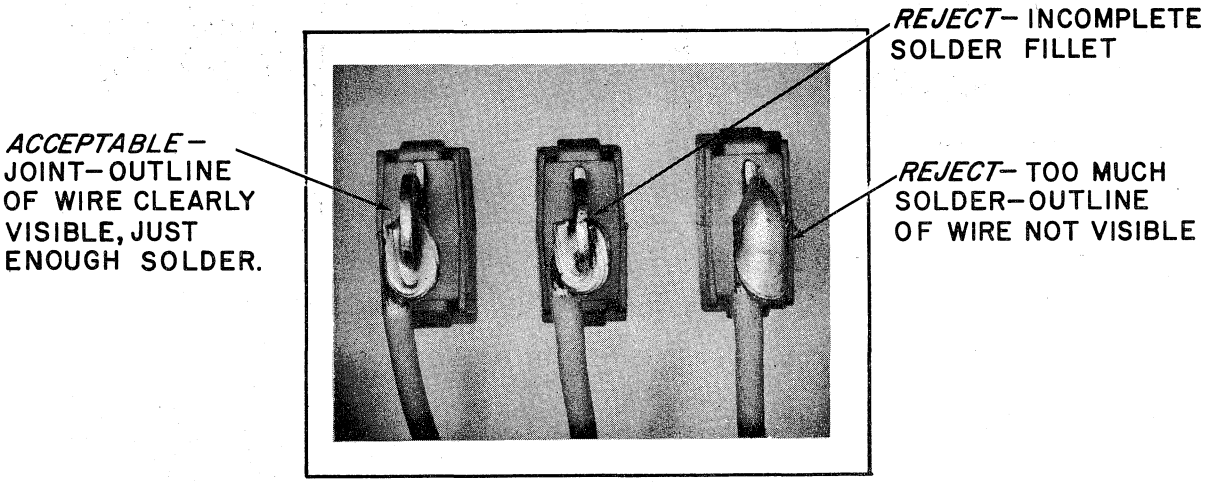
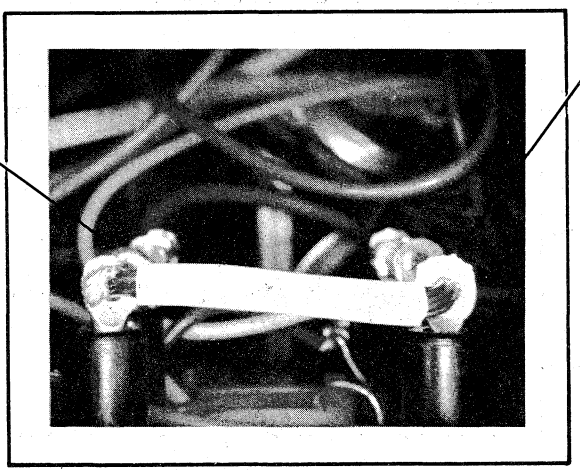


Figure 5-17 Uses of Stranded Wire

Acceptable Joint Characteristics – The quality of a solder can be easily determined by a simple visual inspection. Photos of characteristics of acceptable and defective solder joints are shown in the following photographs.

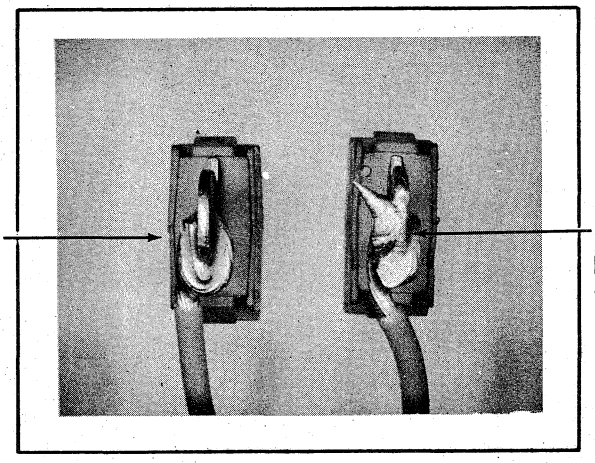


REJECT—INSUFFIENT SOLDER—CAUSED BY USING TOO SMALL A SOLDERING IRON ON A LARGE (#10) WIRE.



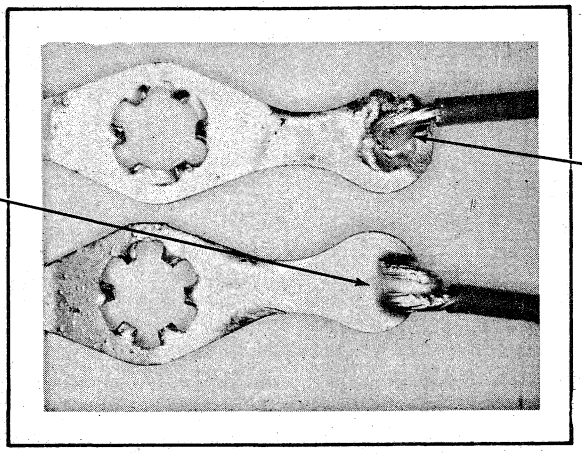
REJECT—INSUFFICIENT SOLDER.

ACCEPTABLE—SOLDER SMOOTH AND SHINY.

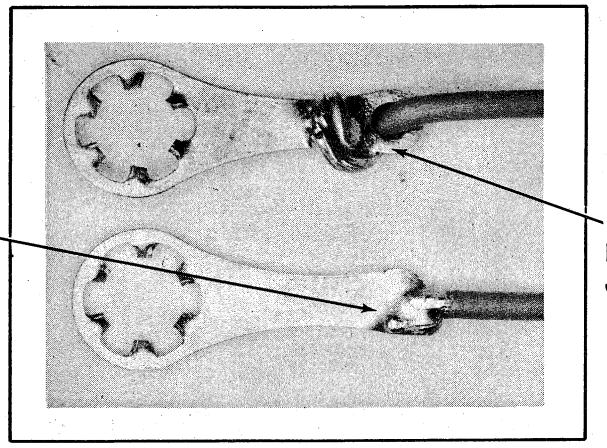


REJECT—SOLDER PEAK, IRON REMOVED TOO SOON

ACCEPTABLE

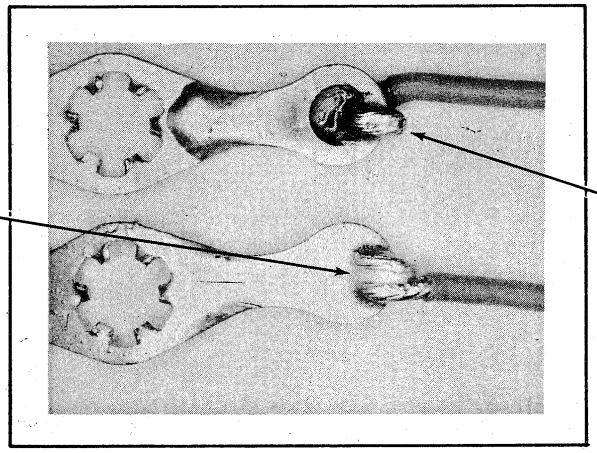


REJECT—COLD SOLDER JOINT, POUROUS JOINT.



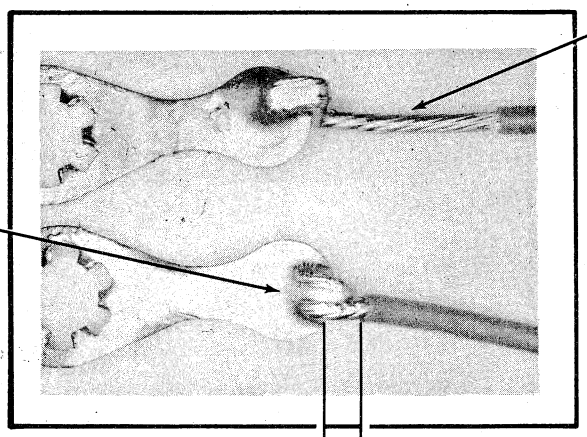
ACCEPTABLE– INSULATION NOT ENTRAPPED.

REJECT– INSULATION RUNS INTO SOLDER JOINT.



ACCEPTABLE– JOINT CLEAN, CAN BE EASILY INSPECTED.

REJECT– EXCESSIVE FLUX–JOINT CAN'T BE INSPECTED.



REJECT– WIRE STRIPPED TOO LONG.

ACCEPTABLE EXPOSED CONDUCTOR LESS THAN 1/8".

1/8" MAX.

The PDP-15 I/O bus system consists of a level converter, two types of cables and a line of peripherals. The level converter, called the DW15 Bus Adapter, changes the positive bus levels of the PDP-15 into negative bus levels (B-, R-, W-series) of the PDP-9. Since the two I/O buses (PDP-9, PDP-15) are compatible in all but single-cycle or direct memory access operations, most PDP-9 devices can be operated on the PDP-15 through the DW-15. The two types of cables are necessary to interconnect:

a. positive logic devices (BC09B)

b. negative logic devices (BC09A)

Peripherals added to a PDP-15 bus system will effect its electrical characteristics, its decoding, its timing, latency and priorities.

These parameters are studied in this chapter.

6.1 THE DW15 BUS ADAPTER

The DW15 converts positive PDP-15 bus levels to negative PDP-9 bus levels so that certain PDP-9 devices will operate on the PDP-15. To the PDP-15 I/O bus, the DW15 is just another positive device, with Type BC09B input and output cables. The positive output cable attaches to the next positive device. However, the adapter acts as a T-connector, for it also sprouts two BC09A negative Output Cables. These attach to the nearest negative peripheral. Succeeding negative peripherals are interconnected with two standard Type BC09A PDP-9 I/O bus cables. Figure 6-1 illustrates the interconnection scheme. The DW15 automatically gives API or DCH positive logic devices priority over the negative logic devices by removing the ENABLE level of the negative bus whenever a request comes up on the positive bus.

6.2 PDP-15 I/O BUS CABLES AND CABLE ASSEMBLIES

There are two types of cable assemblies used in PDP-15 systems; each type has a different set of connector boards, but the same BC09 cable. The BC09A cable assembly is used for interconnecting negative logic devices and the BC09B cable assembly interconnects positive logic devices (Figure 6-1).

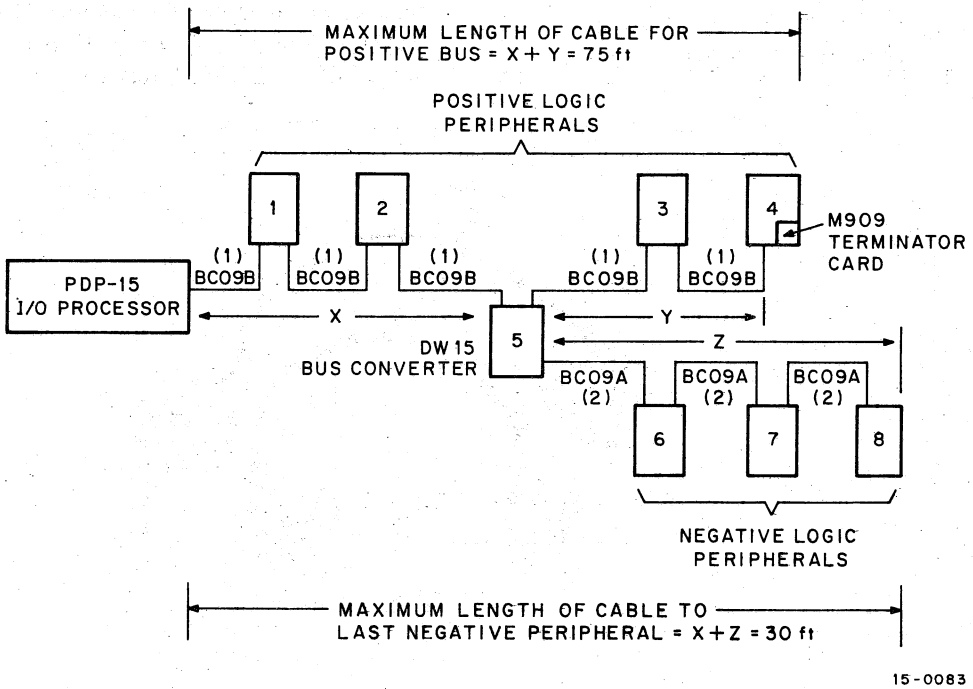


Figure 6-1 The I/O Bus Cable System

6.2.1 The BC09 Cable Characteristics

Table 6-1 summarizes the physical characteristics of the cable, and Table 6-2 lists the electrical characteristics. The electrical values shown are estimates, and where such parameters are critical, it is recommended that the user perform his own measurements.

Table 6-1
BC09 Cable Specifications

36-Pair Cable	Specifications
Each Pair	24 AWG, 7-strand tinned copper Type B 600V 105°C per MIL-W-16878D
Twin	1.0 in. LHL
Cable	
Core	1 of filler
First Layer	6 Pair 5.4 in. RHL
Second Layer	12 Pair 5.4 in. RHL
Outer Layer	18 Pair 7.5 in. LHL
Type	.001 in. Mylar tape spiral wrap 25: Overlap
Jacket	.035 in. wall black 802 C Polyvinylchloride
O.D.	Finished Nominal O.D. .580 in. ± .020 in.

Table 6-2
BC09 Electrical Characteristics

Characteristic	Specification
Characteristic Impedance	68Ω
Delay	1.8 ns/ft
Risetime	45 ns in 100 ft
Capacitance	25 pF/ft

6.2.2 BC09A Assembly

The BC09A Cable Assembly consists of one length of BC09 cable with two Type W850 male connectors (Figure 6-2). The W850 connectors consist of two double-height FLIP CHIP boards. The size and weight of the cable and its connectors require that retaining blocks be used to fasten the W850 connectors to the Type H800 female connector block of the 1943 panels. These retaining blocks, designated as H003 and H004 kits, are shown in Figure 6-3. The H003 spans one H800 block and allows two cables to be plugged into it. The H004 spans two blocks and will receive four cables. All the necessary hardware and installation instructions are furnished with each kit.

The W850 connector has two diode-clamping circuits terminating the lines. The circuit schematic is shown in Figure 6-4. All input W850 connectors must have a normal power connection to pin B (-15V); output connectors need not. Each cable draws 400 mA of current.

Two type BC09A cables are needed to complete a bus connection between two negative devices. General rules for connecting devices with the cables are given in the PDP-9, PDP-9/L Interface Manual.

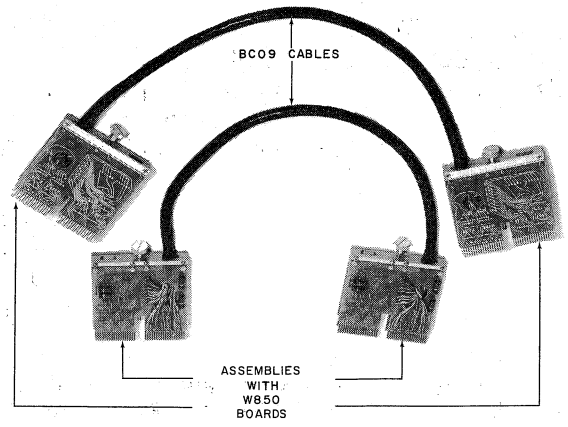
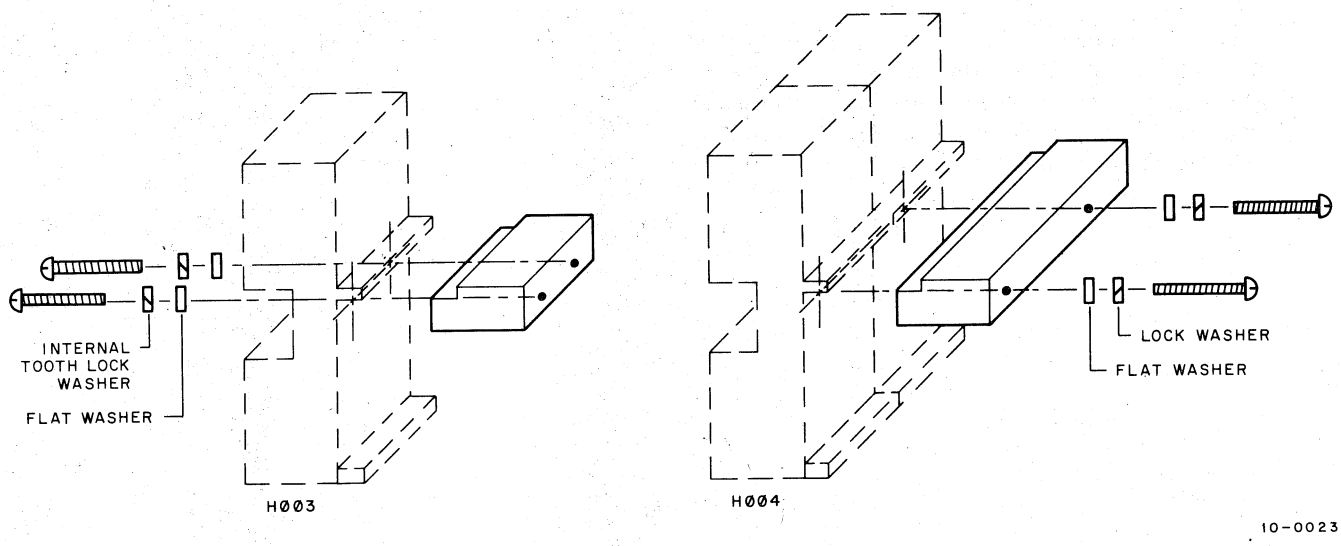
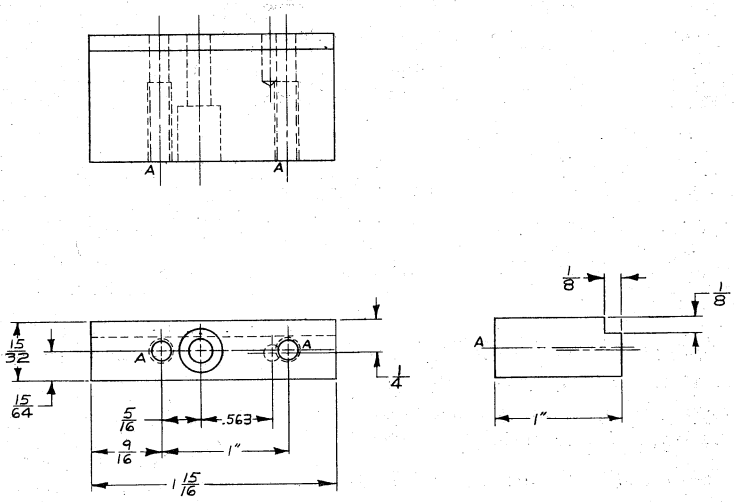


Figure 6-2 BC09A Cable Assembly



H003, H004 Connector Retaining Block
Kits for 1943 Panels



Retaining Block for H911 Panel

Figure 6-3 Retaining Block Kits

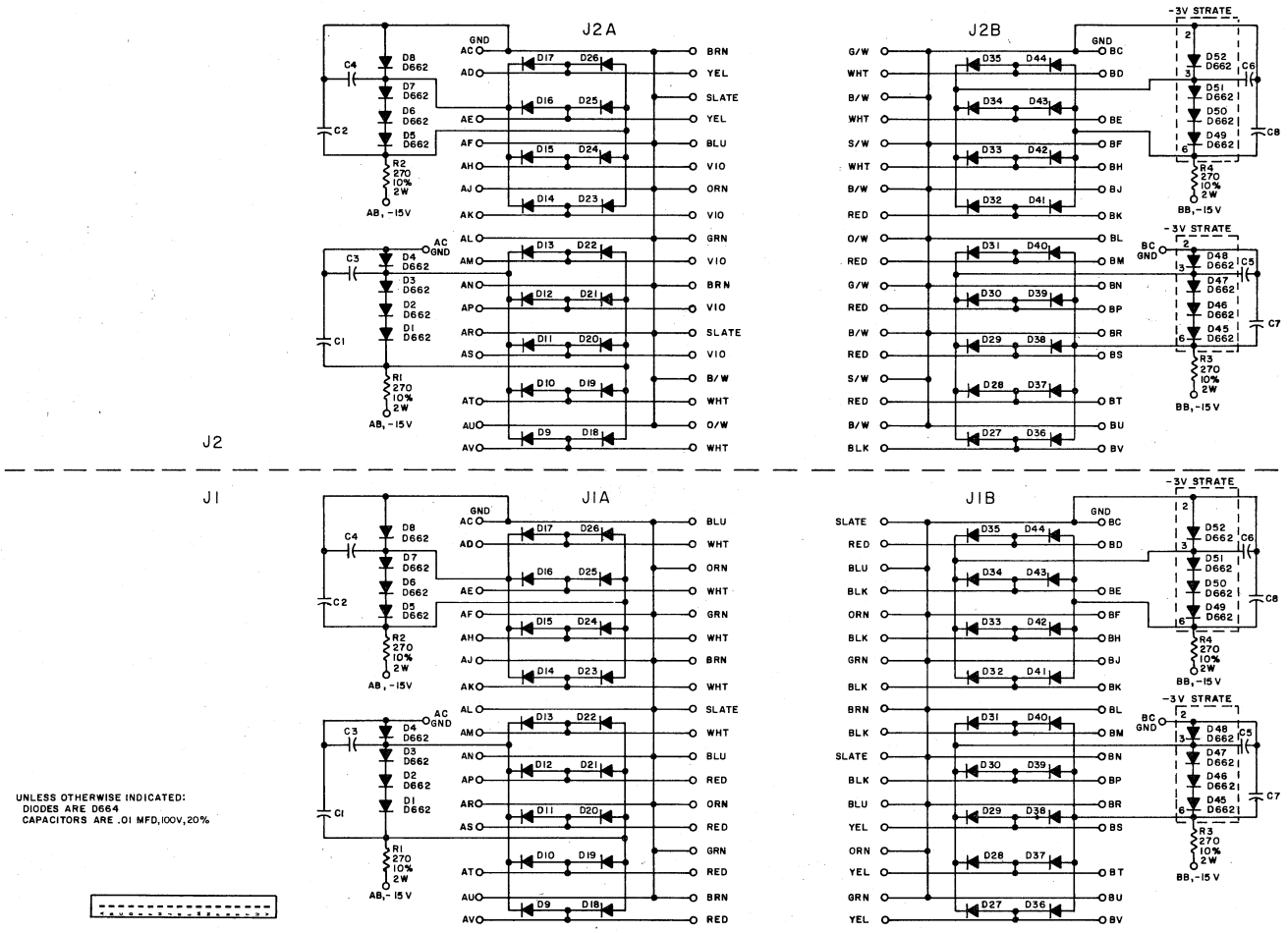


Figure 6-4 I/O Connector W850

6.2.3 BC09B Cable Assembly

This cable assembly is made up of two BC09 cables with two M912 connectors at either end and some associated hardware. It uses the retaining block shown in Figure 6-3 for the H911 panel. Only one BC09B assembly (Figure 6-5) is needed to interconnect two positive logic devices.

Figure 6-6 shows the circuit schematic of the M912 connector.

6.2.4 Cable Lengths

The maximum length of the positive I/O bus is 75 ft, and the combined length of the positive bus before the DW15 and the negative bus after the DW15 cannot exceed 30 ft. Figure 6-1 illustrates these constraints.

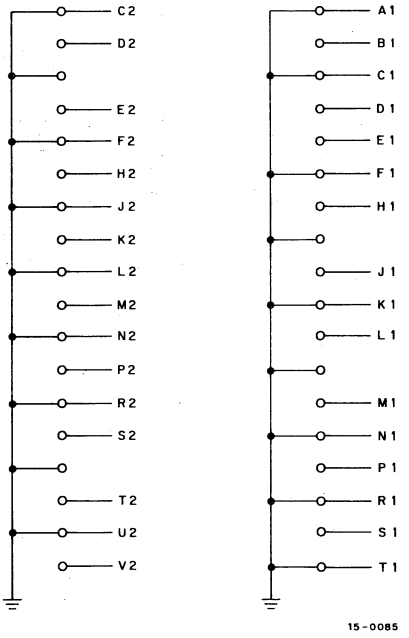


Figure 6-6 M912 PDP-15 I/O Bus Card

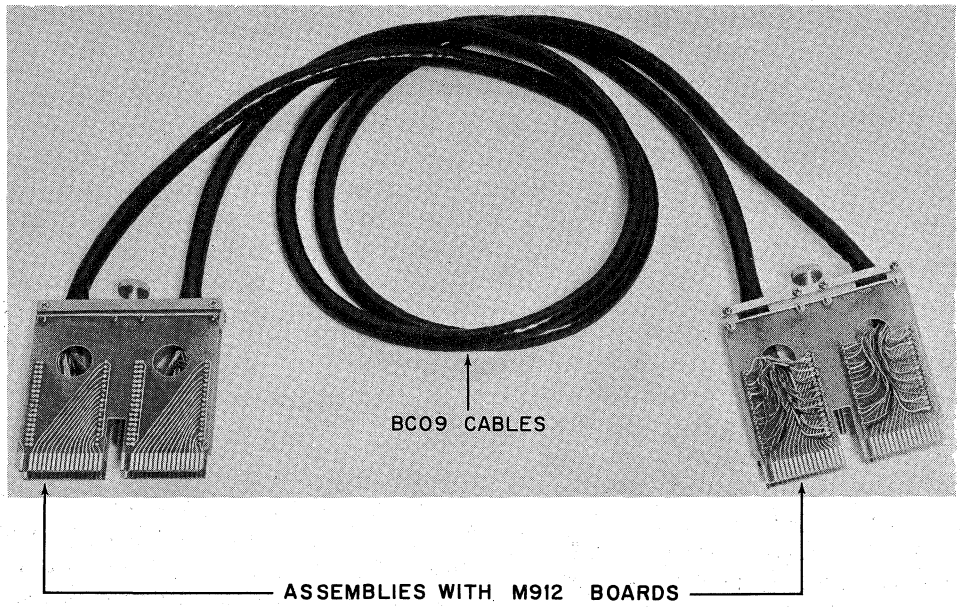


Figure 6-5 BC09B Cable Assembly

6.3 ADDING PERIPHERALS TO THE I/O BUS

When another peripheral is added to a PDP-15 system, it affects not only the electrical characteristics of the I/O bus, but the rest of the peripherals, the speed of the entire I/O complex, the software operating system; and the memory map. How these parameters are affected is described in the following paragraphs.

6.3.1 The Electrical Characteristics of the Bus

The positive I/O bus will electrically support 50 devices which accept an input cable and pass on an output cable; if each device draws no more than 40 μA from a signal through its M510 receiver. Further, 16 devices can be added to the negative portion of the bus. However, there are certain other system constraints which could limit the number of certain types of devices.

6.3.2 Device Select Codes

Each PDP-15 peripheral is assigned a set of unique device select codes which are decoded from the IOT instruction. There are 2^8, or 256, such codes available on the system, and it is vital that no two peripherals have the same code. Obviously, two identical peripherals, such as Type TC15 DECtape Controllers, cannot be added unless one is first modified to change its device select code. This also means that the system software (device handlers) must be changed to recognize the new codes. A designer adding a special peripheral to a PDP-15 system must be careful that the device codes selected are not used by a present or planned peripheral.

Table 6-3 shows all codes presently used by standard peripherals. Since there are a limited number of codes, the peripherals on the system must acknowledge this constraint. It is recommended that the designer refer to the codes used by each device on his system as listed in its respective manual, before committing himself to a design.

6.3.3 Addresses

Each peripheral which uses the API or the multi-cycle data channel facility is assigned unique core locations for their trap address (API) or word count and current address (data channel). These addresses are recognized by the system software according to Tables 6-4, 6-5. The designer must be very careful that the addresses he selects for his special system do not violate the rules of his software operating system.

There are 28 addresses assigned for API devices, and 8 for multi-cycle data channel devices. This restricts the number of API devices to 28, and multi-cycle data channel devices to 4. This is another constraint on the number of devices which may be attached to the I/O bus.

6.3.4 Timing Constraints

The number of devices which use any one of the API levels cannot exceed eight; if this number is exceeded there may be timing errors due to the time it takes the enabling level to propogate through the cable and the M104 modules in each device. This same constraint applies to data channel devices. Only eight devices which use the data channel (single- or multi-cycle) can operate on any one PDP-15 system.

6.3.5 System Latency and Priorities

A design engineer may wish to know if the system will respond quickly enough to his peripheral when it posts a request for transfer, an error condition or an alarm. This question can be answered only when the designer can indicate the state of the other peripherals (e.g., Is the A/D converter transferring and possibly blocking out the device?); the time it takes the operating system to respond; and how long the peripheral can wait before the information is lost. Table 6-6 shows some figures for a typical PDP-15 system. Table 6-7 lists some of the latency figures which assume in each case that the device in question is not interfered with by any other device. It is interesting to note that latencies sometimes overlap in a system. Note that latency is defined as the time between a request for data transfer and the completion of the transfer.

6.3.6 Cable Runs and Terminations

The PDP-15 I/O bus originates at locations MN02 and MN03 in the PDP-15 I/O processor and chains its way from device to device to interconnect all peripherals.

Table 6-3
Assigned PDP-15 IOT Device Selection Codes

00 1 RT Clock 2 Prog. Interrupt 4 RT Clock	10 AFC-15 UDC-15	20 AFC-15 UDC-15	30 VT15 Graphic Processor	40 LT19 Line 1,2,3,4 Teleprinter LT15A	50	60	70 DEC Disk RF15
01 Perforated-Tape Reader PC15	11 Analog-to-Digital or Digital-to-Analog Converter	21 Relay Buffer DR09A	31 VT15 Graphic Processor	41 LT19 Line 1,2,3,4 Keyboard or LT15A	51	61	71
02 Perforated-Tape Punch PC15	12 A/D or D/A Converter	22 DB98 & 99 Interprocessor buffers	32 Power Fail Graf pen	42 LT19 Line 5,6,7,8 Teleprinter	52	62	72 DEC Disk RF15
03 1 Keyboard 2 Keyboard 4 IORS	13 A/D Converter	23	33 1 33 KSR Skip 2 Clear All Flags 4 DBR,DBK	43 LT19 Line 5,6,7,8 Keyboard	53	63 Disk Pack RP15	73 Mag Tape Control TC59D
04 Teleprinter	14 AM03 & AM09 SD0, 1 SYS I SD2, 3 SYS II	24 Incremental Plotter Control XY15	34	44 LT19 Line 9,10,11,12 Teleprinter	54	64 Disk Pack RP15	74 Mag Tape Control TC59D
05 Displays VP15	15	25 DP09A Data Communication	35	45 LT19 Line 9,10,11,12 Keyboard	55 Automatic Priority Interrupt KA15	65 Automatic Line Printer LP15	75 DECtape Control TC15
06 VP15	16	26 DP09A Data Communication	36	46 LT19D Line 13,14,15 Teleprinter LT15A	56	66 Automatic Line Printer LP15	76 DECtape Control TC15
07 Display and Light Pen VP15BL	17 Memory Protection KM15, KT15	27 Memory Parity MP15	37	47 LT19D Line 13,14,15 Keyboard LT15A	57	67 Card Reader Type CR03B	77 Bank Addressing Instructions

Table 6-4
API Addresses

Channel Number (Octal)	Trap Address	Standard Device	Suggested Priority Level	I/O ADDR Bits 12–17
0	40	Software channel 0	4	100 000
1	41	Software channel 1	5	100 001
2	42	Software channel 2	6	100 010
3	43	Software channel 3	7	100 011
4	44	TC15 DECtape	1	100 100
5	45	TC59 Magtape	1	100 101
6	46	Not assigned	1	100 110
7	47	Not assigned	1	100 111
10	50	PC15 High Speed Paper Tape	2	101 000
11	51	KW15 Real Time Clock	3	101 001
12	52	KF15 Power Fail option	0	101 010
13	53	MP15 Memory Parity option	0	101 011
14	54	VP15 Display	2	101 100
15	55	CR03B Card Reader	2	101 101
16	56	LP15 C/F Line Printer	2	101 110
17	57	AD15 Analog Subsystem	0	101 111
20	60	DB09A Interprocessor Buffer	3	110 000
21	61	Not assigned	3	110 001
22	62	DP09A Dataphone	2	110 010
23	63	RF15 DECdisk	1	110 011
24	64	RP15 Disk Pack	1	110 100
25	65	XY15 Plotter	2	110 101
26		Not assigned		110 110
27		Not assigned		110 111
30		Not assigned		111 000
31		Not assigned		111 001
32	72	LT19/LT15A Keyboard	3	111 100
33	73	LT19/LT15A Printer	3	111 101
34	74	TC15 DECtape Control	3	111 110
35	75	DP09 Dataphone	2	111 111

Table 6-5
Data Channel Addresses

Device	Word Count	Current Address
Interprocessor Buffers DB99, 98	22,24	23,25
Not presently assigned	26	27
DECtape TC15	30	31
Magtape TC59	32	33
Not presently assigned	34	35
RF15	36	37

Table 6-6
Worst Case Latency, PDP-15 Peripherals

Device	Maximum Transfer Rate	Allowable Latency (in μs)	Worse-Case Latency in this system (in μs)
RP15 Disk Pack	130,000* word/second	14	8.5
RS09 DECdisk	62,000 words/second	16	12
TU56 DECtape	5,000 words/second	200	17
LP15C Line Printer	1,000 lines/minute	40**	36
LP15F Line Printer	356 lines/minute	not applicable	not applicable
Real-Time Clock	1,000 cycles/second	1,000	61
PC15 Paper Tape System	300 characters/second (reader) 50 characters/second (punch)	3,333	62 + subroutine
CR03B Card Reader	200 cards/minute	3,750	190
KSR 35 Teletype	10 characters/second	100,000	250

*The RP15 is double-buffered with two 36-bit registers. Two 18-bit words are transferred back-to-back on the I/O bus.

**The LP15C transfers two 18-bit words every 40 μs.

Table 6-7
Worst Case Latency Figures (μs)

Multi-Cycle Data Out (From Memory)	7
Multi-Cycle Data In	6
Add to Memory	7
Increment Memory	4
Single-Cycle Data Out (Normal Mode)	5
Single-Cycle Data In (Normal Mode)	5

Each cable enters a free standing cabinet from the bottom, and connects devices from bottom to top. The cabinets are about 6 ft high so that about 15 ft of cable per free-standing cabinet, or 9 ft per cabinet if they are bolted together, should be anticipated if more than three different devices are to be interconnected within a cabinet.

The last device on the positive I/O bus must plug a Type M909 68Ω Terminator Card into the slots usually assigned to its output I/O bus cable. Figure 6-7 shows the I/O bus drive/termination scheme.

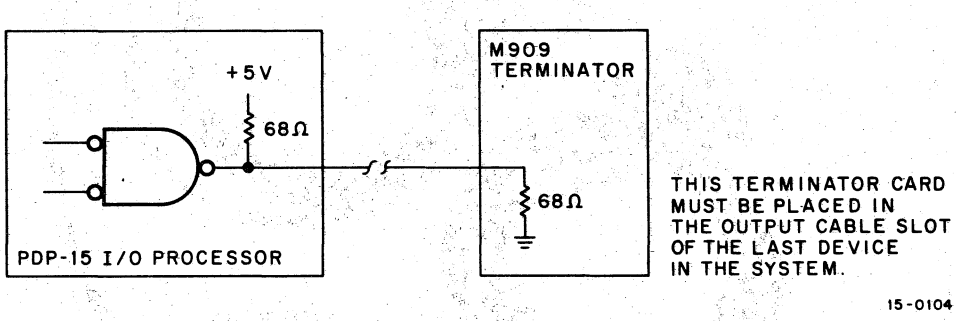


Figure 6-7 The PDP-15 I/O Bus Drive/Termination Scheme

6.4 DESIGNING PDP-15 DEVICES TO OPERATE ON THE PDP-9

Most PDP-15 peripherals will operate from a PDP-9 through its I/O bus provided the positive PDP-15 I/O bus signals are converted to the corresponding negative PDP-9 levels. The only exceptions are single-cycle data channel devices, which have no PDP-9 equivalents.

All peripherals designed with DEC logic must use M510 receivers and M622 drivers. In order to simplify the level converting needed to operate such peripherals on the PDP-9, two level converting modules (pin compatible with the M510 and M622) called the M500 and M632, respectively, are available. These modules conform to PDP-9 I/O bus requirements, and convert negative output signals to positive levels, and positive input (drive to computer) signals to negative PDP-9 I/O bus levels. They are described completely in Chapter 2.

The M104 module is then replaced with an M194 which converts the EN IN and EN OUT levels to their appropriate voltages.

All of these modules are described in Chapter 2.

Any PDP-15 device which is placed on the PDP-9 I/O bus must be cabled with either a BC09B cable assembly if it is connected to other PDP-15 devices, or with a special cable called the BC09C if it is connected to regular PDP-9 devices.

The BC09C cable assembly is a hybrid of the BC09A and the BC09C cables. It has a BC09B(M912) termination at one end and two BC09A(W850) terminations on the other. Two BC09 cables make up the rest of the assembly.

The BC09B end of the BC09C cable is plugged into the appropriate slot of the converted PDP-15 device. The two BC09A terminations then go to the neighboring PDP-9 device.

Figure 6-8 shows the BC09C cable, and Figure 6-9 illustrates how it should be connected.

In summary, to convert a PDP-15 device to operate on the PDP-9, it must first not use the single cycle data break and then must submit to the following operations:

1) Replace all M510 receivers with M500 receivers.
2) Replace all M622 drivers with M632 drivers.
3) Replace all M104 multiplexers with M194 multiplexers. Use BC09B cables to interconnect converted PDP-15 devices, and BC09C cable assemblies to interconnect converted PDP-15 devices and PDP-9 devices.

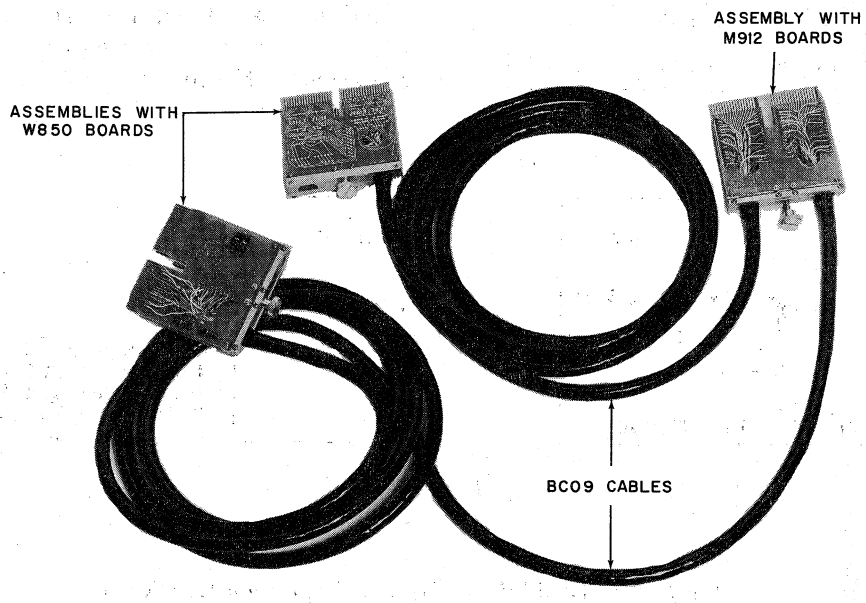


Figure 6-8 The BC09C Cable

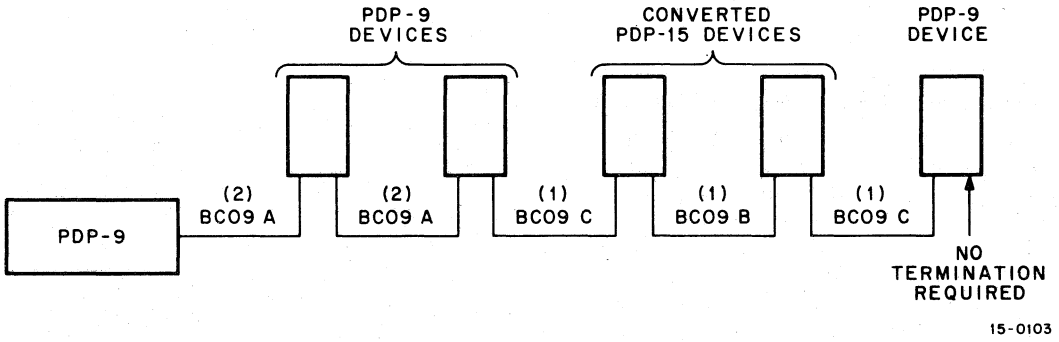


Figure 6-9 Cabling Converted PDP-15 Devices to the PDP-9 I/O Bus

Digital Equipment Corporation
Maynard, Massachusetts

digital

printed in U.S.A.