ALLEN-BRADLEY
A Rockwell International Company

Programming and Operations Manual

Mini-PLC-2 Programmable Controller

(Cat. No. 1772-LN1, -LN2, -LN3)

Price: $25.00

Because of the variety of uses for this equipment and because of the differences between this solid state equipment and electromechanical equipment, the user of and those responsible for applying this equipment must satisfy themselves as to the acceptability of each application and use of the equipment. In no event will Allen-Bradley Company be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The illustrations, charts and layout examples shown in this manual are intended solely to illustrate the text of this manual. Because of the many variables and requirements associated with any particular installation, Allen-Bradley Company cannot assume responsibility or liability for actual use based upon the illustrative uses and applications.

No patent liability is assumed by Allen-Bradley Company with respect to use of information, circuits, equipment or software described in this text.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Section 1
INTRODUCTION

1.0 OVERVIEW

The Bulletin 1772 Mini-PLC-2 Programmable Controller is a rugged, solid state programmable controller that consists of the Mini-PLC-2 Processor (Cat. No. 1772-LN1, -LN2 or -LN3) and the 1771 I/O Family of racks and modules. Refer to Figure 1-1.

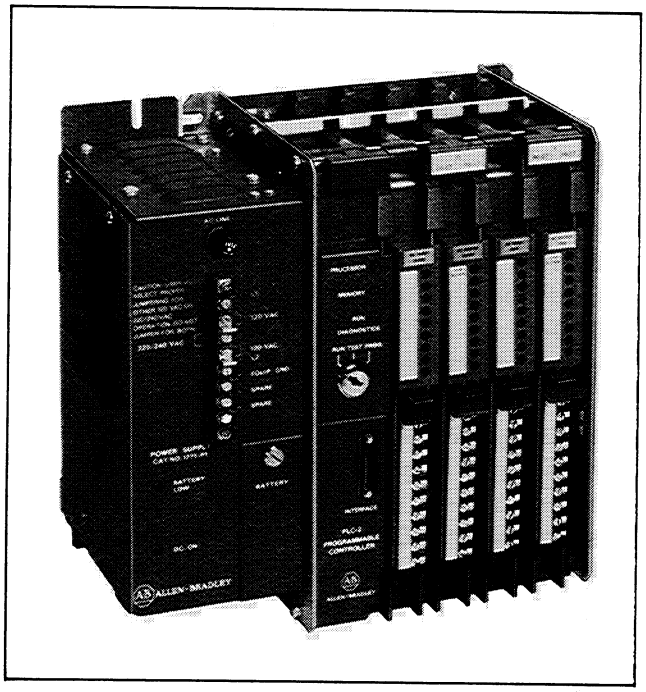


FIGURE 1-1 — Mini-PLC-2 Programmable Controller

With a user-written program and appropriate I/O modules, the Mini-PLC-2 Controller can be used to control many types of industrial applications such as:

- Process control
- Material handling
- Palletizing
- Measurement and gaging
- Pollution control and monitoring

The Mini-PLC-2 Processor has a Read/Write CMOS memory that stores User Program instructions, numeric values and I/O device status.

The User Program is a set of instructions in a particular order that describes the operations to be performed and the operating conditions. It is entered into memory, instruction by instruction, in a ladder diagram format display from the keyboard of the Industrial Terminal (Cat. No. 1770-T1, -T2 or -T3). Some ladder diagram symbols closely resemble the relay symbols used in hardwired relay control systems.

During program operation, the Mini-PLC-2 Processor continuously monitors the status of input devices and, based on User Program instructions, either energizes or de-energizes output devices. Because the memory is programmable, the User Program can be readily changed if required by the application.

In addition to ON/OFF control, the Mini-PLC-2 Controller can perform additional functions such as:

- Timing/Counting operations
- Arithmetic (+, -) operations
- Arithmetic (x, ÷) operations (1772-LN3 Processor)
- Data comparisons
- Block Transfer (1772-LN3 Processor Module)
- I/O Forcing
- Data Highway and RS-232-C interfacing

When the Industrial Terminal (Cat. No. 1770-T1, -T2 or -T3) and the Mini-PLC-2 Programmable Controller are used together or with additional peripheral devices, application data can be recorded or displayed using a variety of peripheral functions:

- Report generation
- Contact histogram (the ON/OFF history of a bit in memory)
- Hard-copy printout of the User Program or the complete memory

- Recording/loading/verifying the User Program using magnetic tape

1.1 LADDER DIAGRAM LOGIC

PC ladder diagram logic closely resembles hardwired relay logic. Hardwired relay control systems require electrical continuity to turn output devices ON and OFF. For example, the relay diagram in Figure 1-2 shows that limit switch LS1 and relay contact CR2 must be closed to energize relay coil CR4.

Similarly, in each rung of ladder diagram program, logic continuity is needed to energize or de-energize the output instruction, and ultimately the output device. For example, the ladder diagram rung in Figure 1-3 shows the input devices and the output device with their respective Data Table bit addresses. The bit addresses correspond to the location of the I/O devices wired to the I/O modules. When the two input instructions are logically TRUE, or the bits in memory are ON, logic continuity is established. This causes the output instruction to be TRUE and the output device to be turned ON.

1.2 MEMORY STRUCTURE

The Data Table of the Mini-PLC-2 Processor memory is made up of an arrangement of storage points called bits (BInary digiTs). A bit is the smallest unit of memory and can store information as a "1" or a "0" (Figure 1-4). When a "1" occupies a bit, that bit is ON; when a "0" occupies a bit, that bit is OFF.

A group of 8 bits forms a single byte. Two bytes, or 16 bits, make up one word. All Data Table words are identified by their word address which is a 3-digit octal number. The octal numbering system is explained in Section 13.

Similarly, each bit in a word is identified by a two-digit number using the octal numbering system. The memory bits are numbered 00 through 07 and 10 through 17, with the most significant bit 17 (MSB) at the left and the least significant bit 00 (LSB) at the right.

A specific bit in Data Table can be identified by combining the 3-digit word address and 2-digit bit number to form the bit address, such as 03012 or 030/12. The difference depends on the Industrial Terminal used. The 1770-T1 and -T2 display the 5-digit bit address centered above the instruction symbol. The 1770-T3 displays the bit address by placing the 3-digit word address above and the 2-digit bit number below the instruction symbol. Programming examples will be illustrated using the 1770-T3 display. However, both displays will be illustrated in the Instruction Summary Table at the end of each programming section in this manual.

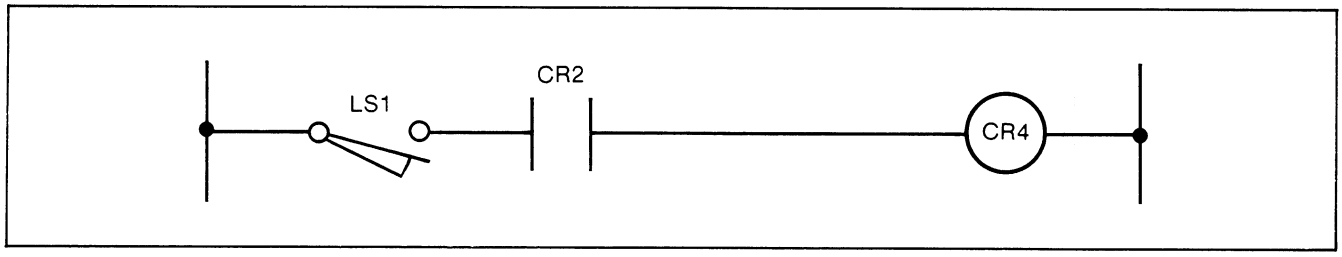


FIGURE 1-2 — Relay Diagram

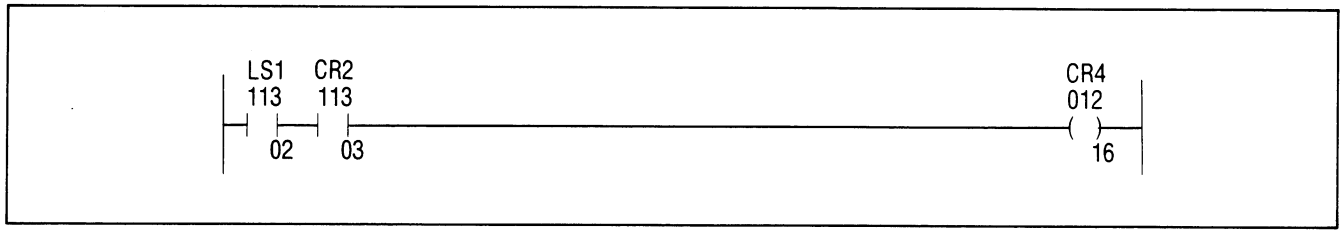


FIGURE 1-3 — Ladder Diagram Rung

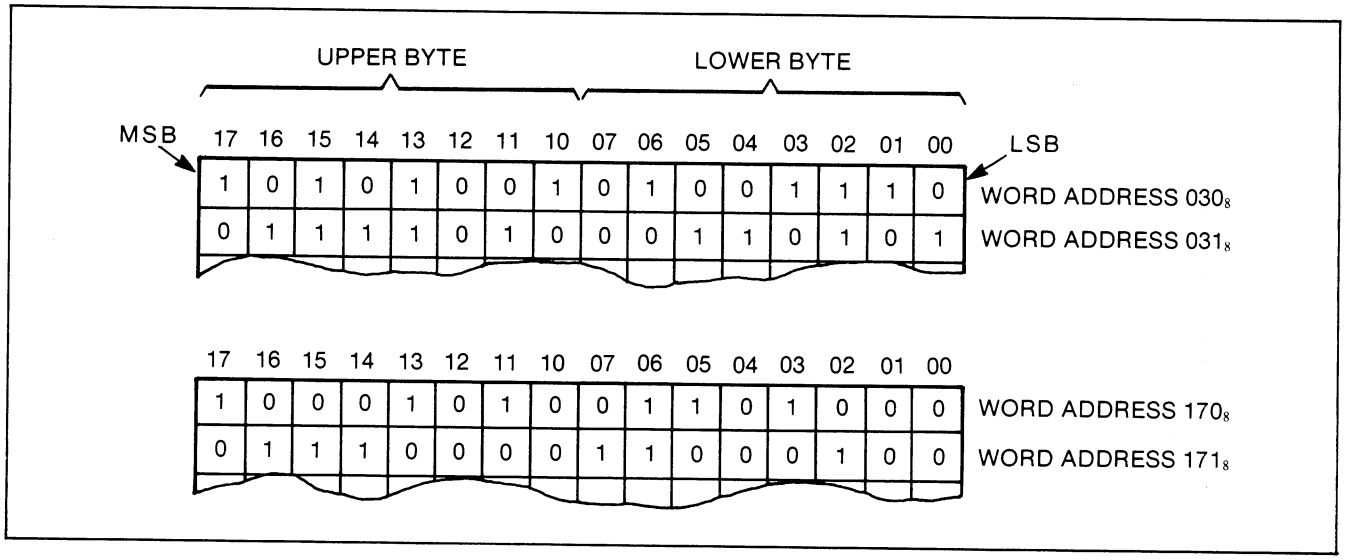


FIGURE 1-4 — Memory Word Structure

1.3 MEMORY ORGANIZATION

The Mini-PLC-2 Processors have either a 512 word memory size (Cat. No. 1770-LN1) or a 1024 word memory size (Cat. No. 1772-LN2 or -LN3). These memory words are organized by their word addresses and are divided into the following major areas. The size of each area can be varied within limits to suit user needs, but the total cannot exceed the Processor memory size.

- Data Table
- User Program
- Message Storage (if used)

The Data Table stores the information needed in the execution of the User Program such as the status of input and output devices, timer/counter Preset and Accumulated values, bit/word storage, etc. Any instruction in the User Program can address any word or bit in the Data Table except in the Processor Work Areas.

The Data Table is factory configured to 128 words. The words reserved for timers and counters can be decreased to any even-number value down to 48 words so that storage capacity for User Program and/or messages can be increased. Figures 1-5 shows the organization of a factory configured Data Table.

The User Program follows the Data Table in memory. The User Program is the logic that controls the machine operation. The logic consists of instructions that are programmed in ladder diagram format. Each instruction requires 1 word of memory.

Message Storage area begins after the END statement of the User Program. This area stores the alphanumeric characters of the messages. Two characters can be stored in one word.

For a detailed description of memory, refer to Publication 1772-700, the Organization and Structure of the Mini-PLC-2 Memory.

1.4 HARDWARE/PROGRAM INTERFACE

The Processor monitors input conditions and controls output devices according to a user-entered program. The interface between hardware and program occurs in the Input/Output Image Table.

1.4.1 Image Table

The primary purpose of the Input Image Table is to duplicate the status (ON or OFF) of the input devices wired to input module terminals. If an input device is ON (closed), its corresponding Input Image Table bit is ON ("1"). If an input is OFF (open), its corresponding Input Image Table bit is OFF ("0"). Input Image Table bits are MONITORED by User Program instructions.

The primary purpose of the Output Image Table is to control the status (ON or OFF) of the output devices wired to output module terminals. If an Output Image Table bit is ON ("1"), its corresponding output device is ON (energized). If a bit is OFF ("0"), its corresponding output device is OFF (de-energized). Output Image Table bits are CONTROLLED by User Program instructions.

1.4.2 Instruction Address

Instruction addresses in the Input/Output Image Table have a dual role. The 5-digit bit address references both an I/O Image Table

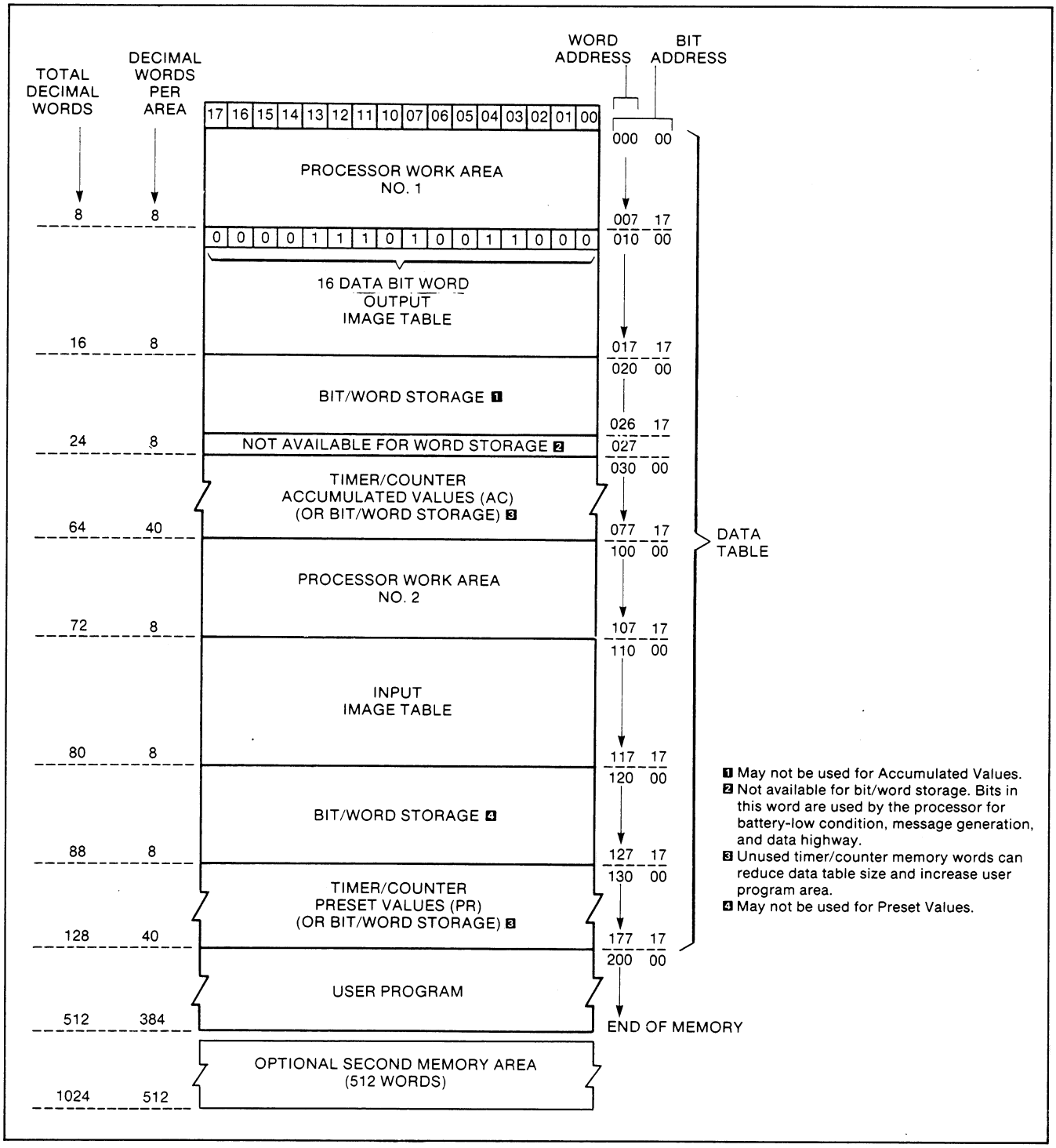


FIGURE 1-5 — Data Table Organization, Factory Configured

address and a hardware location. An I/O Image Table word corresponds to two I/O modules located in a Module Group in the I/O Rack. Both are represented by the upper 3 digits of the 5-digit bit address. The lower byte of the I/O Image Table word corresponds to an I/O module in the left slot of the Module Group. The upper byte corresponds to an I/O module in the right slot of the Module Group. See Figure 1-6. The remaining two digits represent the bit number in the I/O Image Table word and the terminal number in the Module Group. Each 5-digit bit address in the I/O Image Table directly relates to an I/O module terminal as shown in Figure 1-7.

1.4.3 Fundamental Operation

The hardware-program interface is illustrated in Figure 1-8 by showing the operational relationship between the input and output devices, the Input/Output Image Table and the User Program.

When an input device connected to terminal 112/12 is closed, the input module circuitry senses a voltage. The ON condition is reflected in the Input Image Table bit 112/12. During the Program scan, the Processor examines bit 112/12 for an ON (1) condition. If the bit is ON (1), the EXAMINE ON instruction is logically TRUE. A TRUE condition is displayed as an intensified instruction. A path of logic continuity is established and causes the rung to be TRUE. The Processor then sets Output Image Table bit 013/06 to ON (1). The Processor turns ON terminal 013/06 during the next I/O scan and the output device wired to this terminal becomes energized. When the rung condition is TRUE, the output instruction is intensified.

When the input device wired to terminal 112/12 opens, the input module senses no voltage. The OFF condition is reflected in the Input Image Table bit 112/12. During the program scan, the Processor examines bit 112/12 for an ON (1) condition. Since the bit is OFF (0), logic continuity is not established and the rung is FALSE. The Processor then sets Output Image Table bit 013/06 to OFF (0). In the next I/O scan, the Processor turns OFF terminal 013/06 and the output device wired to this terminal is turned OFF.

1.5 COMPATIBILITY

The Mini-PLC-2 Processors are compatible with the Industrial Terminal, the Data Highway and RS-232-C interfacing. The 1771-LN3 Processor is compatible with Bulletin 1771 Block Transfer I/O modules.

1.5.1 Industrial Terminal Compatibility

The Mini-PLC-2 Controller can be programmed using the Industrial Terminal (Cat. No. 1770-T1, T2, or T3). It can also be programmed with the PLC-2 Program Panel (Cat. No.1772-T1) or with the combination of the PLC Program Panel with PLC-2 Program Panel Adapter (Cat. No. 1774-TA and 1772-T4).

This manual will illustrate programming examples using the Industrial Terminal. The first edition of this manual, dated January 1980, illustrates programming examples using the (obsolete) PLC-2 Program Panel.

1.5.2 Data Highway Compatibility

The Mini-PLC-2 Controller can be connected to the Allen-Bradley Data Highway using the Communication Adapter Module (Cat. No. 1771-KA). Data Highway messages and the Data Highway Communication zone of User Program must reference only the addresses within the user-configured Data Table.

1.5.3 Block Transfer Compatibility

The Mini-PLC-2 Processor module (Cat. No. 1772-LN3) can be programmed to communicate with intelligent Bulletin 1771 I/O modules having Block Transfer capability. These include the Thermocouple, Analog Input, Analog Output, Encoder/Counter, PD, etc. modules. Section 11.2 of this manual covers Block Transfer Programming.

1.5.4 User Program Compatibility

User Programs written for the Mini-PLC-2/15, PLC-2/20 or PLC-2/30 should not be loaded into the Mini-PLC-2 Controller unless the User Program is compatible in the following areas:

- The Data Table is 128 words or less.
- The words of memory used do not exceed 512 for the 1772-LN1 Processor; or 1024 for the 1772-LN2 or -LN3 Processor.

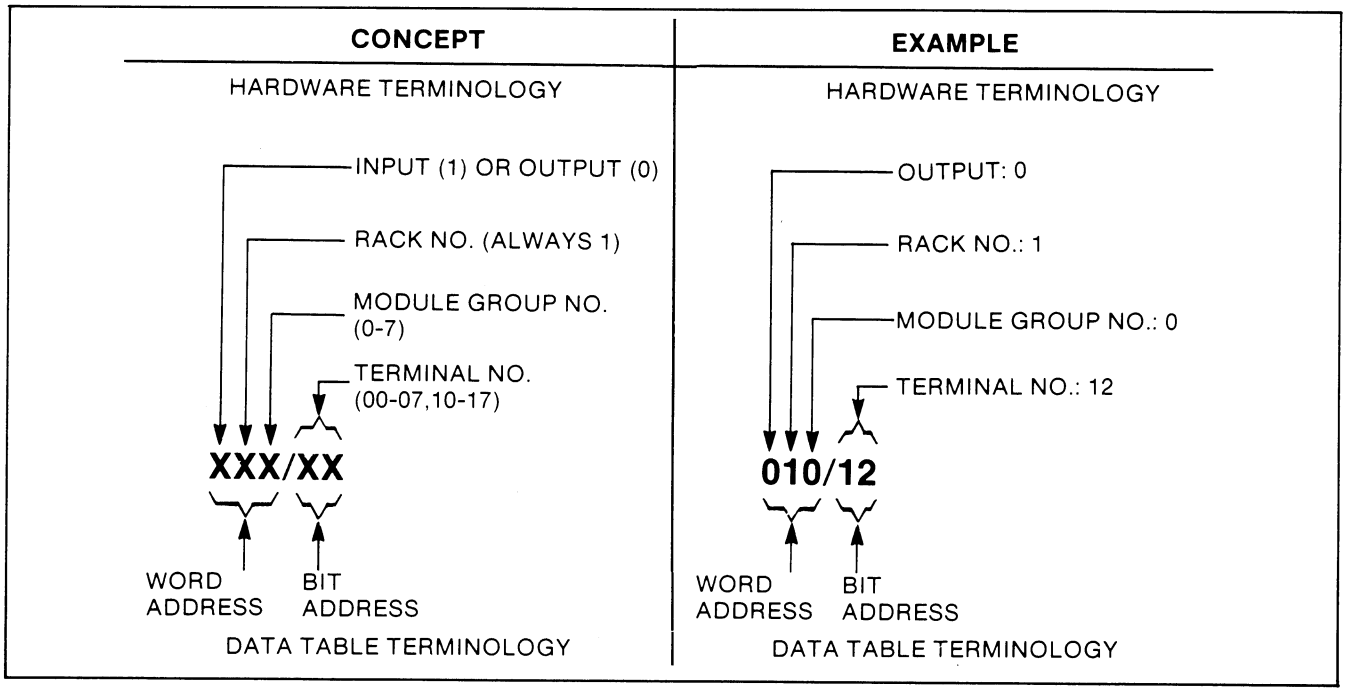


FIGURE 1-6 — Instruction Address Terminology

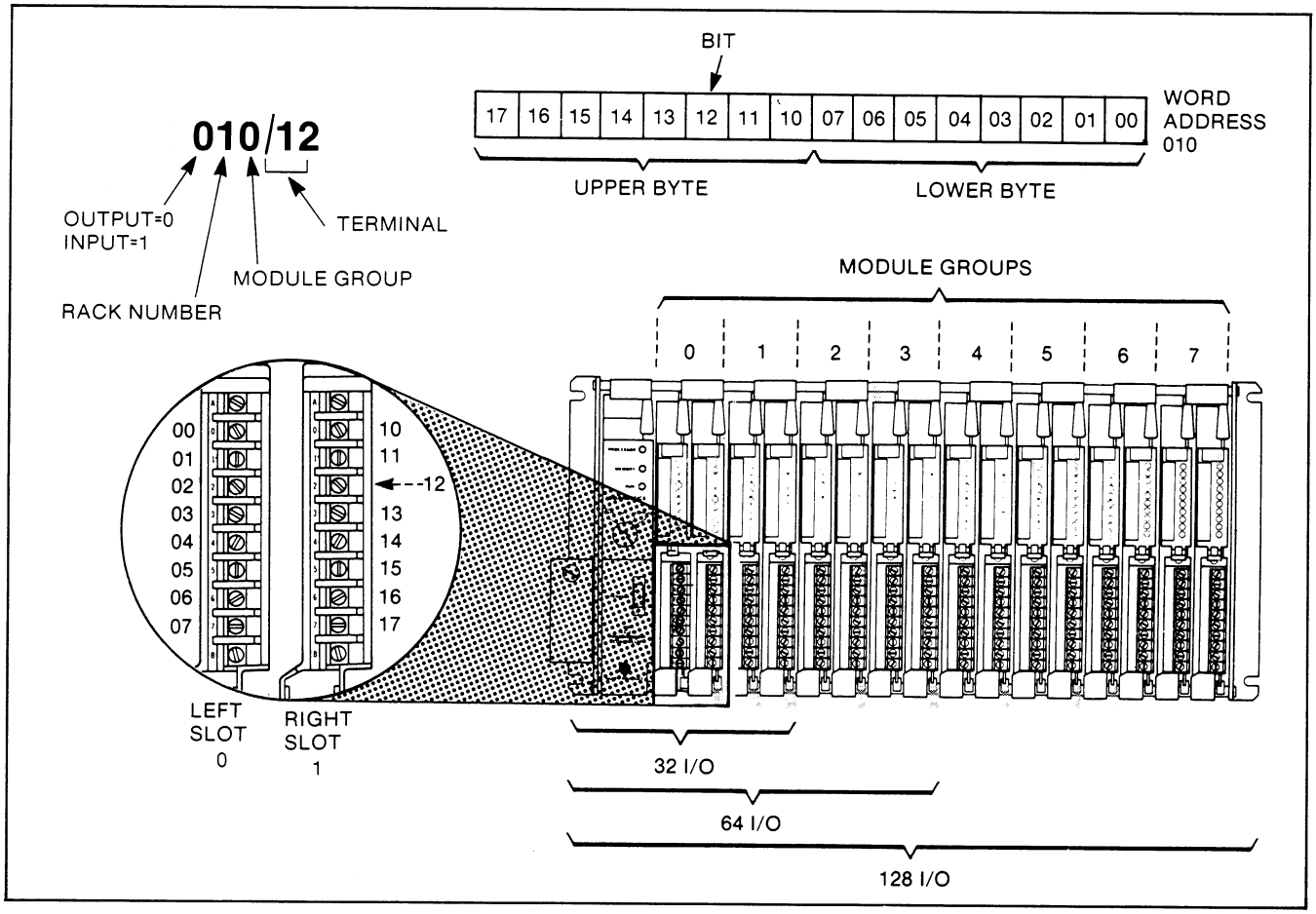


FIGURE 1-7 — Bit Address to Hardware Relationship

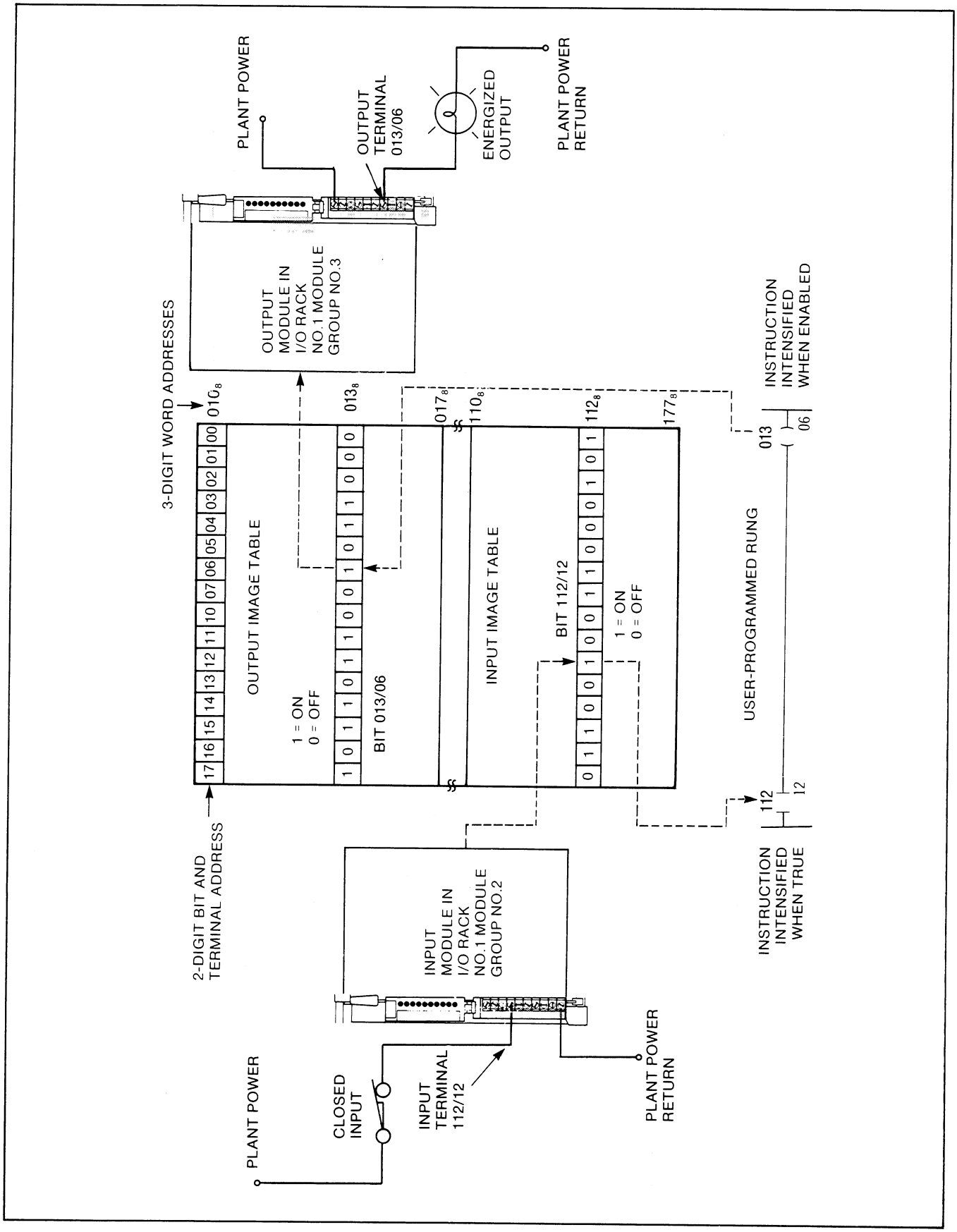


FIGURE 1-8 — Hardware-Program Interface

- The instruction set contains compatible instructions: block instructions and subroutine programming are absent; MULTIPLY or DIVIDE instructions can only be entered in the 1771-LN3 Processor; divide by 0 is compatible as stated in Section 6.4.

The Industrial Terminal will prevent the loading of memory from tape or data cartridge if the Data Table size or the number of memory words exceed the capacity of the Processor.

If a User Program has a compatible Data Table and memory capacity but contains incompatible instructions, it could be loaded into a Mini-PLC-2 Controller. When using a 1772-LN3 Processor, the incompatible instructions would be displayed on a 1770-T3 Industrial Terminal when the Processor was in PROGRAM mode. Upon switching to TEST or RUN mode, the Processor would fault before outputs could be energized. The message "PROCESSOR FAULT" would appear at the top of a blank screen. Upon switching back to PROGRAM mode, the Industrial Terminal would be initialized to the Mode Selection display. Options then available include:

a) Remove the incompatible User Program using the Total Memory Clear function [CLEAR MEMORY][9][9]

b) Salvage the User Program by replacing the incompatible instructions with equivalent programming that the Mini-PLC-2 can handle.

If a User Program with an incompatible instruction set were to be loaded into a 1772-LN3 Processor having a 1770-T1 or -T2 Industrial Terminal, ERR messages would appear randomly throughout the program when the Processor was in PROGRAM mode. They would be located adjacent to any instruction that the Processor was not capable of handling. Upon switching to TEST or RUN mode, the Processor would fault before outputs could be energized. The options available would be the same as those stated above. The 1770-T1 Industrial Terminal would display the message "COMMUNICATION FAULT-CHECK CABLES FIRST." Two messages would be alternately displayed by the 1770-T2 Industrial Terminal: "PROCESSOR FAULT" and INVALID INSTRUCTION ENCOUNTERED.

Incompatible instructions would be ignored by a 1772-LN1 or -LN2 Processor. The display would depend on the Industrial Terminal. A 1770-T1 or -T2 would display "ERR" messages in place of incompatible instructions. The 1770-T3 would display incompatible instructions although they could not executed.

1.6 SUPPORT DOCUMENTATION

The following support documents contain additional information regarding Mini-PLC-2 Controller components.

- Mini-PLC-2 Programmable Controller Assembly and Installation Manual (Publication 1772-820): contains necessary information on installation, assembly, maintenance and troubleshooting
- PLC-2 Family Support Documentation Manual (Publication 1772-803-1): contains useful information on memory organization, Data Table expansion, system features, wiring, module keying and various features of Mini-PLC-2 Controller components and 1771 I/O.

Section 2
HARDWARE CONSIDERATIONS

2.0 GENERAL

This section will only describe the hardware features of the Mini-PLC-2 Programmable Controller that are used when inputting or debugging the User Program. For information on installation, start-up, troubleshooting, etc, refer to Publication 1772-820, the Mini-PLC-2 Programmable Controller Assembly and Installation Manual.

2.1 MODE SELECT SWITCH

The Mini-PLC-2 Processor has a three-position keylock Mode Select Switch (Figure 2-1) that places the Processor in one of three operating modes:

- PROGRAM
- TEST
- RUN

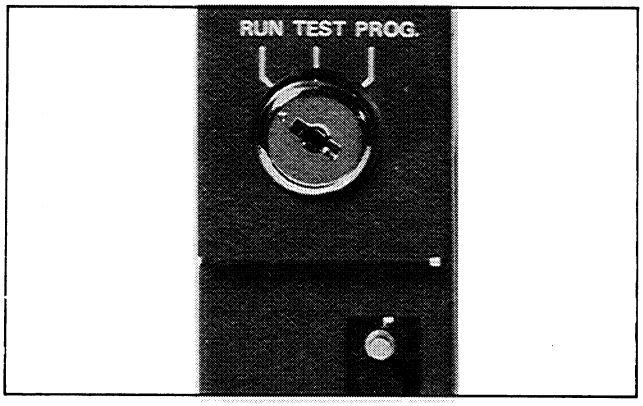


FIGURE 2-1 — Mode Select Keyswitch

2.1.1 PROG Position

This switch position places the Processor in the PROGRAM mode. User Program instructions are entered in this mode. They can be entered from the Industrial Terminal, or entered from the Digital Cassette Recorder (Cat. No. 1770-SA) or the Data Cartridge Recorder (Cat. No. 1770-SB) when connected to the Industrial Terminal. All outputs are de-energized in this switch position and the machine controlled by the Mini-PLC-2 will not operate.

2.1.2 TEST Position

This switch position places the Processor in the TEST mode. The User Program is tested under simulated operating conditions. Inputs are active and recognized by the Processor, but user output devices are not energized. All outputs are disabled in this switch position. Changes to the User Program are NOT permitted, but Data Table values can be changed using the On-Line Data Change function or the Bit Manipulation function.

2.1.3 RUN Position

This switch position places the Processor in the RUN mode. The User Program will be executed and outputs are controlled by the program. Changes to the User Program are not permitted, but Data Table values can be changed using the On-Line Data Change function or the Bit Manipulation function.

This is the only switch position that allows removal of the Mode Select Switch key.

2.2 PROCESSOR DIAGNOSTIC INDICATORS

Indicators on the front panel of the Processor Module aid in analyzing controller status. (Refer to Figure 2-2.) During operation in any mode, the Processor continuously monitors its own status through checks on timing and data parity. In addition, the Processor receives a signal from the power supply if user AC power goes low for longer than one-half cycle.

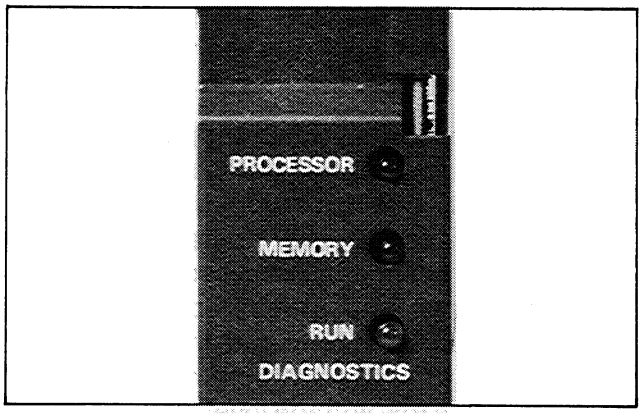


FIGURE 2-2 — Diagnostic Indicators

2.2.1 PROCESSOR Indicator

This red indicator illuminates if the Processor is unable to scan the User Program and Data Table. It is normally OFF. When ON, the Processor has stopped communication with I/O modules. If this occurs, the Last State Switch determines the status of energized controller outputs. (Refer to paragraph 2.4.)

Reset may be attempted for this type of fault by changing the Mode Select Switch to the PROGRAM mode, then back to RUN. Reset may also be accomplished by cycling line power to the system power supply or by reloading the User Program.

2.2.2 MEMORY Indicator

This red indicator illuminates if the Processor detects loss of User Program, a discrepancy in memory data, or a parity error. It is normally OFF.

The Processor stops communication with I/O modules if this type of fault is detected. The Last State Switch determines the status of controller outputs if this fault occurs. (Refer to paragraph 2.4.)

This error may be reset by turning the Mode Select Switch to the PROG position then back to RUN, by cycling line power to the system power supply, or by reloading the User Program.

2.2.3 RUN Indicator

This green indicator illuminates when the Processor is operating with the Mode Select Switch in the RUN mode. When this indicator is ON, controller outputs are enabled. This also implies that no Processor-related fault has been detected.

This indicator turns OFF in the RUN mode if the system power supply detects that voltage on the user AC line has dropped below 98V or 196V for 120V or 220/240V operation, respectively. In this event, the Processor disables all output devices and stops receiving input module data. This prevents the Processor from storing input data which might be inaccurate due to a low voltage level. The indicator also goes OFF when a fault occurs.

In the event of user AC power failure, the restart of the Processor is automatic with recovery of the line to the normal voltage range.

2.3 POWER SUPPLY DIAGNOSTIC INDICATORS

The system Power Supply (Figure 2-3) has two diagnostic indicators on the front panel: The BATTERY LOW and DC ON indicators.

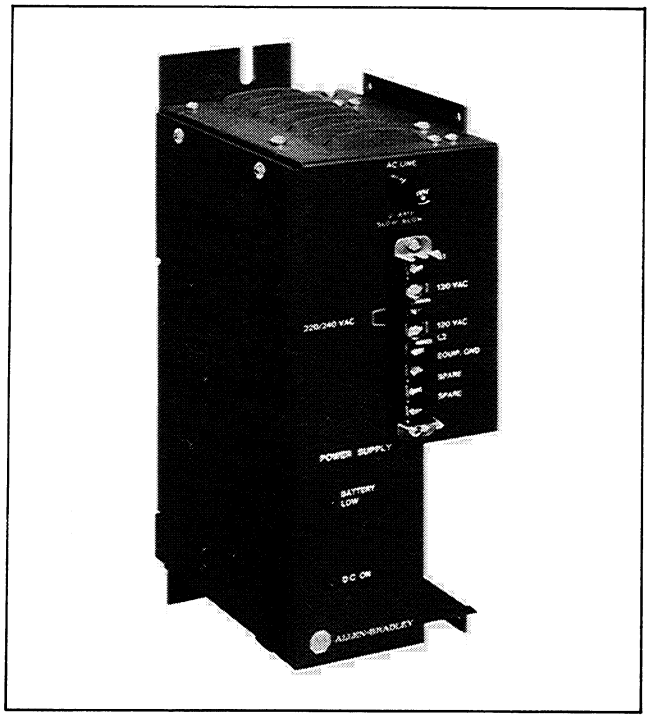


FIGURE 2-3 — System Power Supply (Without Battery Pack)

2.3.1 BATTERY LOW Indicator

When the batteries for memory backup are low, this red indicator flashes ON and OFF. The Battery Low Bit, bit 027/00, will cycle ON and OFF when a battery-low condition is detected and the Mode Select Switch is in the RUN or TEST position. Programming Techniques can be used to examine this bit and to control some type of alerting device when a battery-low condition exists. The battery will continue to provide memory backup for about one week after the indicator begins to flash.

2.3.2 DC ON Indicator

The DC ON is a red indicator that monitors the 5.1V DC line to the logic circuitry in the

Processor, Processor memory and I/O modules. It is ON when 5.1V DC is present. If the line drops below 5.1V DC, the indicator turns OFF and the controller shuts down.

2.4 SWITCH GROUP ASSEMBLY

When the Processor detects an internal fault, communication with the I/O rack is terminated. The last state of output terminals in this situation is user-selectable by the settings on the Switch Group Assembly. It is located on the left side of the I/O chassis backplane. (Refer to Figure 2-4.) Switch #1 must be set to determine output response to a Processor fault. Switch numbers 2-8 are not used. There are two switch settings:

- ON — Outputs remain in their last state, energized or de-energized, when a fault is detected.
- OFF — Outputs are de-energized when a fault is detected.

WARNING: Switch #1 should be set to OFF for most applications. This allows the Processor to turn controlled devices OFF when a fault is detected. If this switch is set ON, machine operation can continue after fault detection and damage to equipment and/or injury to personnel could result.

2.5 INDUSTRIAL TERMINAL

The Industrial Terminal (Cat. No. 1770-T1,-T2 or -T3) can be used to program the Mini-PLC-2.

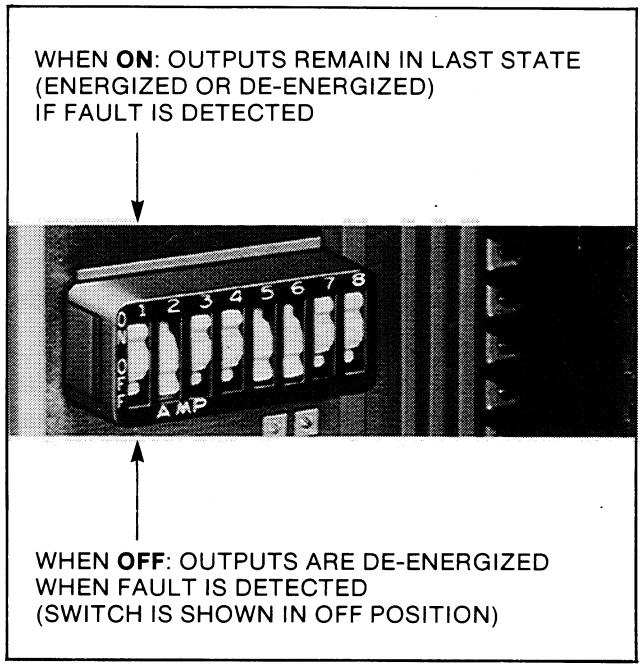


FIGURE 2-4 — Switch Group Assembly

2.5.1 Hook-Up

Perform the following steps to connect the Industrial Terminal to the Mini-PLC-2 Processor: See Figure 2-5.

1. Plug the AC power cord of the Industrial Terminal into a grounded AC outlet.

2. Connect one end of the PLC-2 Program Panel Interconnect Cable (Cat. No. 1772-TC) to Channel A on the rear of the Industrial Terminal.

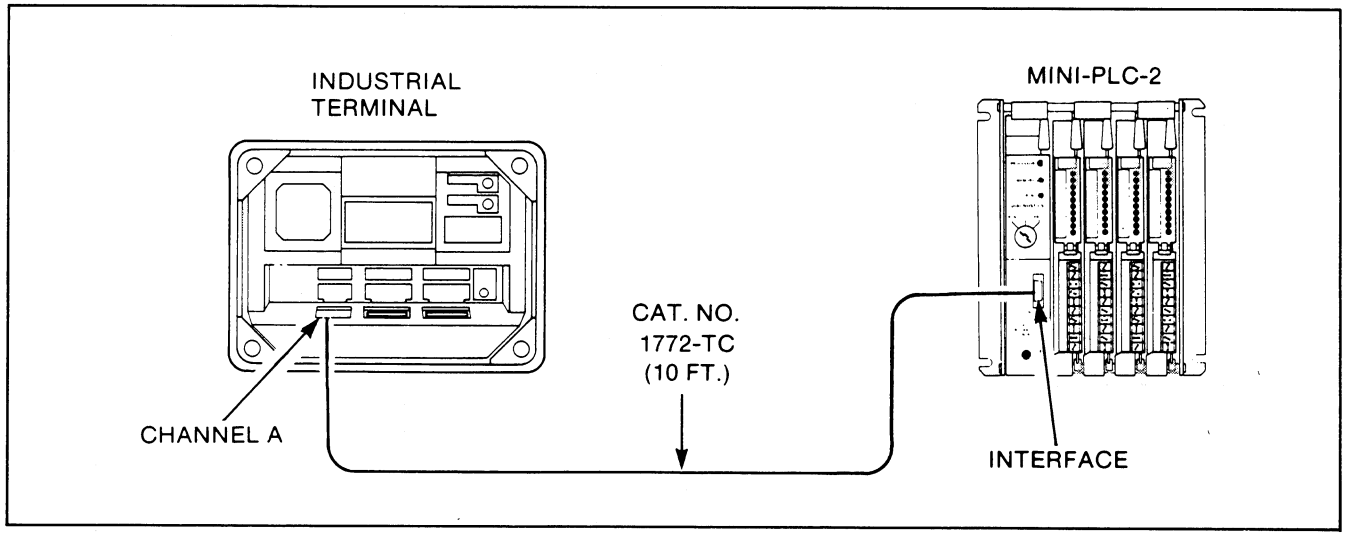


FIGURE 2-5 — Mini-PLC-2 Connection Diagram

3. Connect the other end of the cable to the socket labeled INTERFACE on the front of the Mini-PLC-2 Processor.

4. Insert the PLC-2 Keytop Overlay (Cat. No. 1770-KCA), Figure 2-6, on the Keyboard Module (Cat. No. 1770-FDC).

5. Turn the power switch on the front of the Industrial Terminal to the ON position. Mode Select display will appear.

For additional information on the Industrial Terminal, refer to the Industrial Terminal System User's Manual, Publication No. 1770-805

2.5.2 Mode Selection and Initialization of the Industrial Terminal

When the Industrial Terminal is turned ON or when communication between the Industrial Terminal and Processor is interrupted for any reason, the Mode Selection display will appear. See Figure 2-7 for the 1770-T1 or -T2 display and Figure 2-8 for the 1770-T3 display. Any of the following occurrences can cause

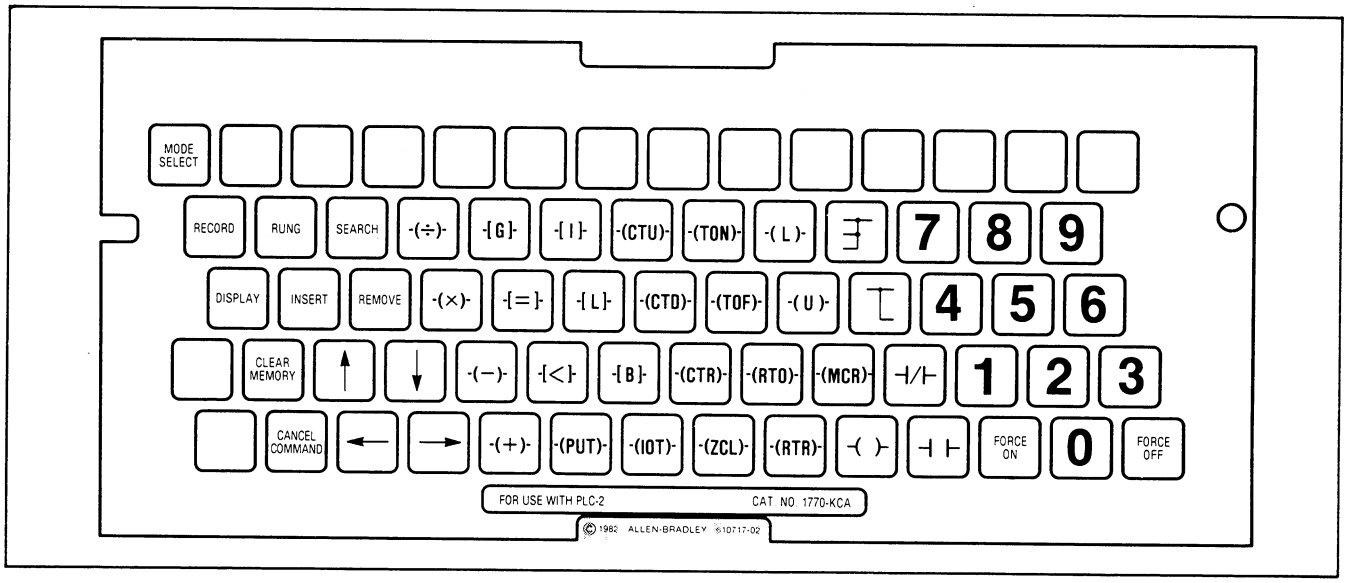


FIGURE 2-6 — PLC-2 Keytop Overlay (Cat. No. 1770-KCA)

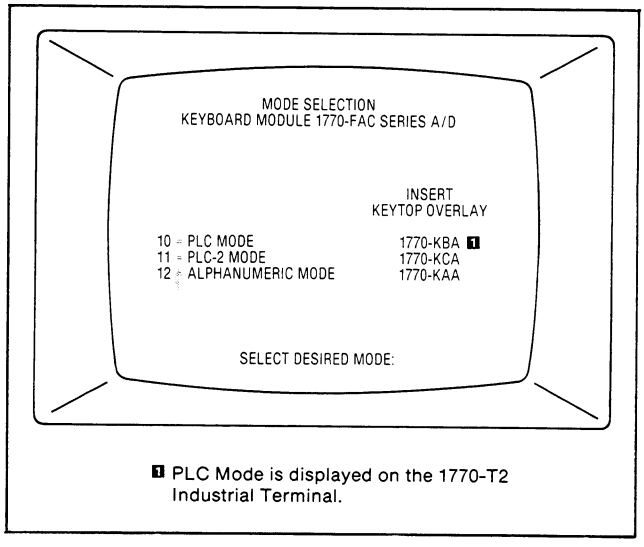


FIGURE 2-7 — Mode Selection Display 1770-T1 or T2 Industrial Terminal

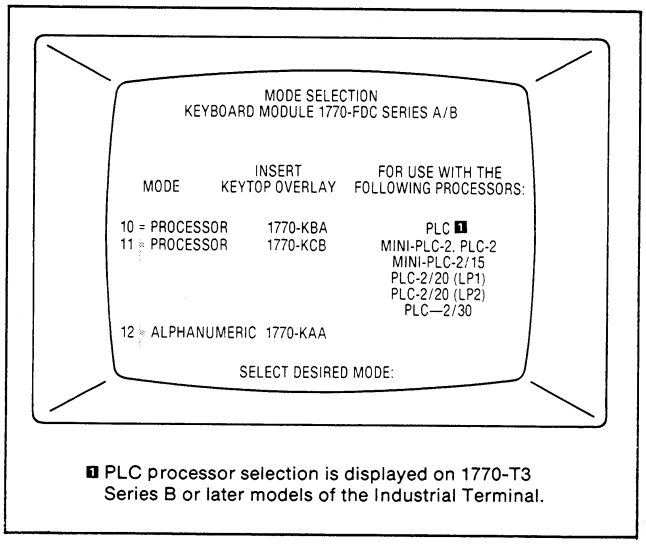


FIGURE 2-8 — Mode Selection Display 1770-T3 Industrial Terminal

an interruption in communication and initialization of the Industrial Terminal

- Pressing the [MODE SELECT] Key
- Loss of power to the Processor or Industrial Terminal
- Disconnecting the Program Panel Interface Cable

To initialize the Industrial Terminal, one of the operating modes shown in Figure 2-7 or 2-8 must be chosen. If the Industrial Terminal is connected to a PLC-2 Family Processor, the Processor type (i.e. Mini-PLC-2, Mini-PLC-2/15, PLC-2/20, PLC-2/30) will be intensified.

To enter Ladder Diagram (PLC-2) mode, press [1][1] on the PLC-2 Keytop Overlay (Cat No. 1770-KCA).

When the Industrial Terminal is to be used as an alphanumeric data terminal, insert the Alphanumeric Keytop Overlay (Cat. No. 1770-KAA) and press [1][2]. Operation of the Industrial Terminal with the Alphanumeric Keytop Overlay is described in Section 10.3, Report Generation.

2.5.3 Keytop Overlay

The Mini-PLC-2 Processor should be programmed using the PLC-2 Keytop Overlay (1770-KCA) shown in Figure 2-6. All keys in this overlay are functional with the 1770-T1, -T2 or -T3 Industrial Terminal. The functions of the keys will be described in detail starting with Section 3 of this manual.

The PLC-2 Family Keytop Overlay (1770-KCB) can be used with any of the Processors in the PLC-2 Family. It should be used with the 1770-T3 Industrial Terminal when programming either the Mini-PLC-2/15 or the PLC-2/30. This overlay contains keys for some functions that are not possible with the Mini-PLC-2. When any one of these keys is pressed, the message "FUNCTION NOT AVAILABLE WITH THIS PROCESSOR" or "INVALID KEY" will appear on the screen.

Section 3
RELAY-TYPE INSTRUCTIONS

3.0 GENERAL Programmable controllers have many of the capabilities of hardwired relay control systems. Control functions similar to those available with relays are provided by the following relay-type instructions:

- Examine instructions
- Output instructions
- Branch instructions

3.1 EXAMINE INSTRUCTIONS

There are two Examine instructions:

- ON -| |-
- OFF -|/|-

The Examine instructions can examine the status of bits in any Data Table area except for Processor Work Areas. When an EXAMINE ON or EXAMINE OFF instruction is given an address in the I/O Image Table, the instruction can indirectly examine the status of a corresponding I/O device. The status of the I/O Image Table bit will be a 1 or 0 reflecting the ON or OFF condition, respectively, of the I/O device. The I/O device and the I/O Image Table bit have the same address. (See Hardware/Program Interface, Section 1.4).

The condition of the instruction can be either TRUE or FALSE depending on the status of the examined bit. If the Image Table bit is in the desired state, the instruction is TRUE. The TRUE-FALSE conditions of the Examine instructions are as follows:

- The EXAMINE ON instruction is TRUE when the addressed memory bit is a 1, meaning that the corresponding I/O device or bit is ON.
- The EXAMINE ON instruction is FALSE when the addressed memory bit is a 0, meaning that the corresponding I/O device or bit is OFF.
- The EXAMINE OFF instruction is TRUE when the addressed memory bit is a 0, meaning that the corresponding I/O device or bit is OFF.
- The EXAMINE OFF instruction is FALSE when the addressed memory bit is a 1, meaning that the corresponding I/O device or bit is ON.

The EXAMINE ON and EXAMINE OFF instructions are illustrated in an example rung in Figures 3-1 and 3-2, respectively.

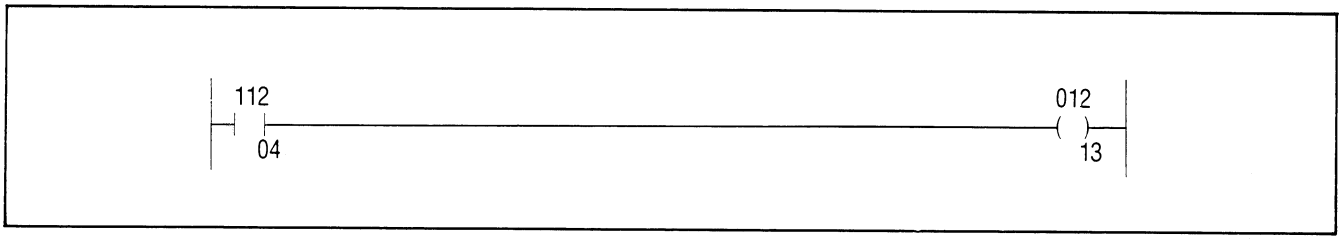


FIGURE 3-1 — EXAMINE ON Instruction

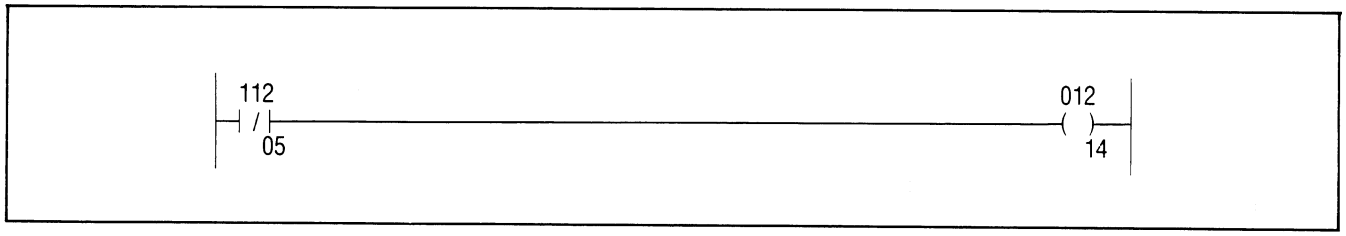


FIGURE 3-2 — EXAMINE OFF Instruction

3.2 OUTPUT INSTRUCTIONS

The output instructions set an addressed memory bit to "1" (ON) or reset it to "0" (OFF). An Output Image Table bit, as a "1" or "0", can cause an output device to be turned ON or OFF, respectively.

Output instructions are programmed at the end of the ladder-diagram rungs. Only one output instruction can be programmed on each rung. The output instruction will be performed only if the Condition (input) instructions preceding it provide a path of logical continuity (Figure 3-3).

These output instructions are:

- ENERGIZE -( )-
- LATCH -(L)-
- UNLATCH -(U)-

These instructions are used to set memory bits ON or OFF in any area of the Data Table, excluding the Processor Work Areas. Generally, they should NOT be assigned Input Image Table addresses because Input Image Table words are reset by the I/O scan.

3.2.1 OUTPUT ENERGIZE Instruction

The OUTPUT ENERGIZE instruction tells the Processor to turn an addressed memory bit ON when rung conditions are TRUE. This memory bit will determine the ON or OFF status of an output device when addressed to an output terminal. This instruction can also be used to turn ON a storage bit for later use in the program.

The OUTPUT ENERGIZE instruction tells the Processor to turn the addressed memory bit OFF when rung conditions go FALSE. Refer to Figure 3-4.

The OUTPUT ENERGIZE instruction can be programmed unconditionally for some types of specialized programming. Its use should be limited to storage bits for these special purposes. An unconditional OUTPUT ENERGIZE instruction (Figure 3-5) causes the output instruction to remain energized continuously. This is not advisable in output device programming for safety reasons, because the device cannot be turned OFF by program logic. Care should be taken not to inadvertently enter an unconditional output instruction.

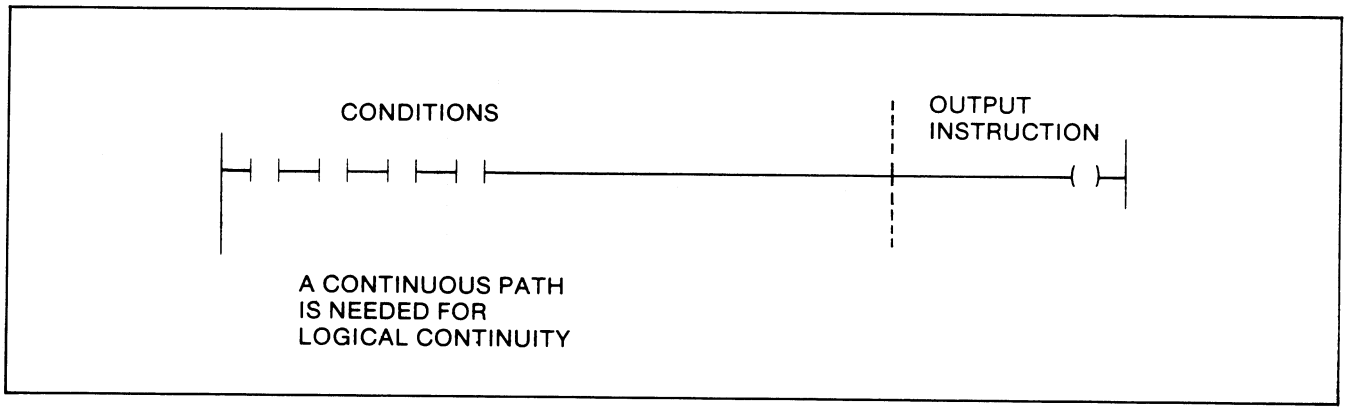


FIGURE 3-3 — Ladder-Diagram Rung

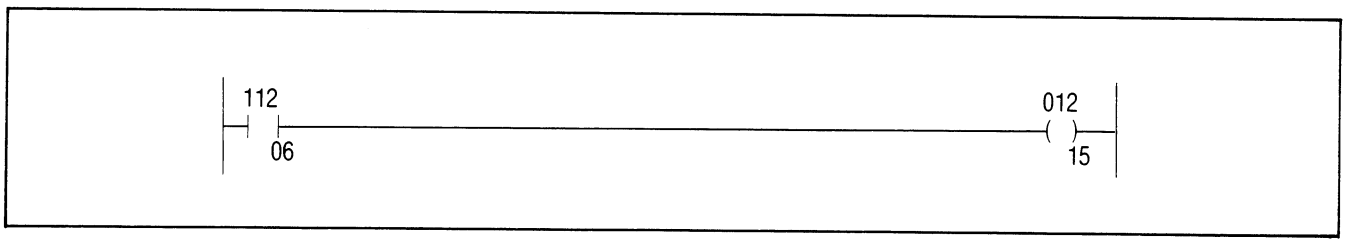


FIGURE 3-4 — OUTPUT ENERGIZE Instruction

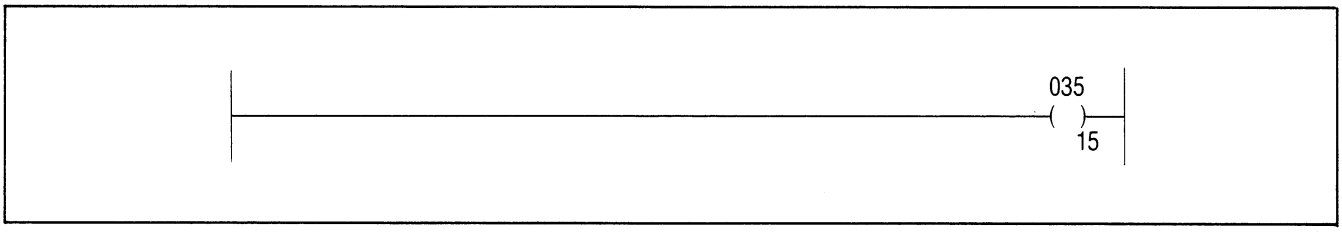


FIGURE 3-5 — Unconditional OUTPUT ENERGIZE Instruction

3.2.2 OUTPUT LATCH and UNLATCH Instructions

There are two output instructions that are termed "retentive." These instructions are:

- OUTPUT LATCH
- OUTPUT UNLATCH

These instructions are usually used as a pair for any bit address they control.

The OUTPUT LATCH instruction (L) is somewhat similar to the OUTPUT ENERGIZE instruction. The OUTPUT LATCH instruction tells the Processor to set an addressed memory bit ON when the rung condition is TRUE. Unlike the OUTPUT ENERGIZE instruction, the OUTPUT LATCH instruction is "retentive." This means that once the rung condition goes FALSE, the latched bit remains ON until reset by an OUTPUT UNLATCH instruction. If power is lost but Processor back-up battery is maintained, all latched bits will remain ON. Outputs associated with the latched bits will be OFF with the power OFF. However, they will turn ON immediately when power is restored.

The OUTPUT UNLATCH instruction (U) is used to turn OFF a memory bit that has been latched ON. The OUTPUT UNLATCH instruction addresses the same memory bit that has been latched ON (Figure 3-6). When the rung condition for the OUTPUT UNLATCH instruction goes TRUE, the addressed memory bit is reset to zero (OFF). Refer to Figure 3-7. The OUTPUT UNLATCH is also "retentive." This means that once the rung condition goes FALSE, the unlatched bit remains OFF until reset by an OUTPUT LATCH instruction.

When the Mode Select Switch is changed from the RUN position, the last LATCH or UNLATCH instruction continues to control the addressed memory bit, but the output device is de-energized by the Processor. When the Mode Select Switch is turned back to RUN a latched output device will be energized.

The OUTPUT LATCH and UNLATCH instructions, when entered, automatically set the bit to OFF. The bit can be initially preset ON by entering the number [1] immediately after the address. The ON or OFF condition will be displayed below the instruction when the Processor is in the PROGRAM mode (Figure 3-8). When the Mode Select Switch is turned to the RUN position, the addressed memory bit and output device, if latched ON, will

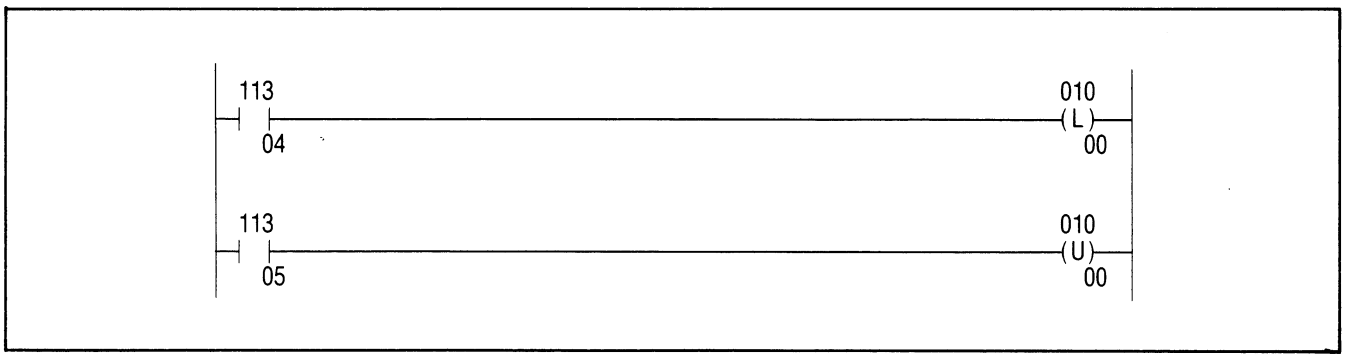


FIGURE 3-6 — LATCH/UNLATCH Instructions

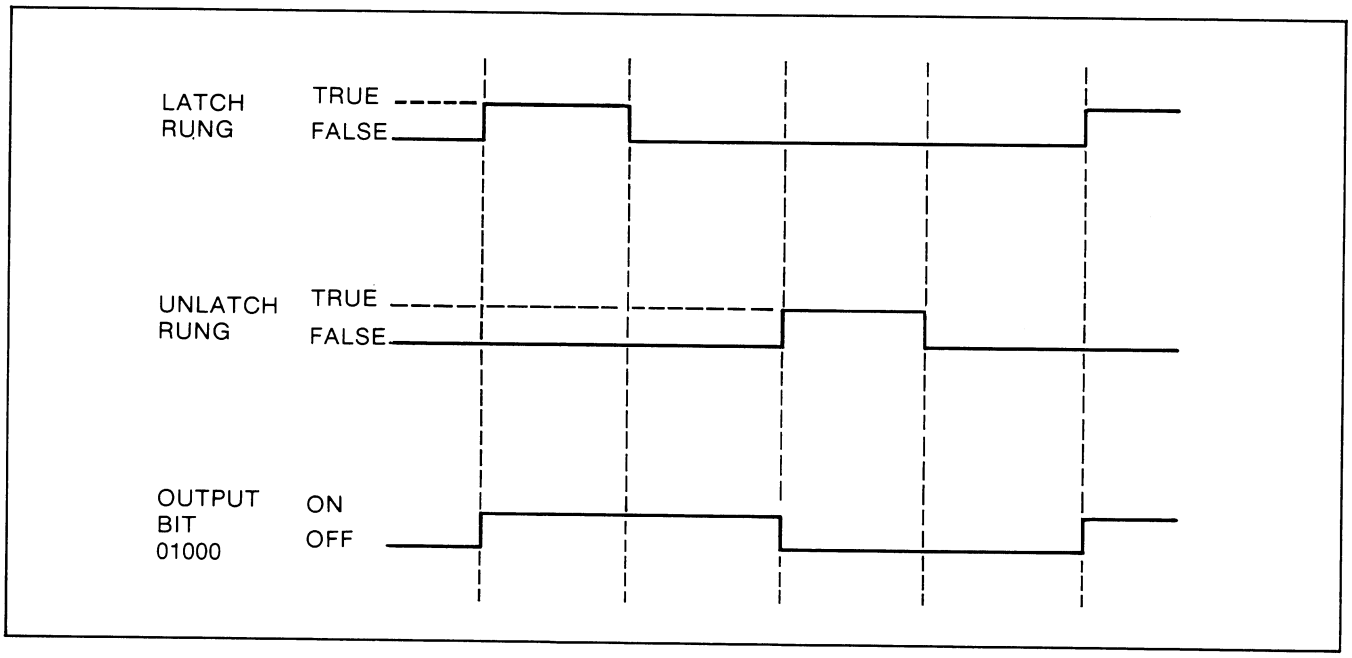


FIGURE 3-7 — LATCH/UNLATCH Timing Diagram

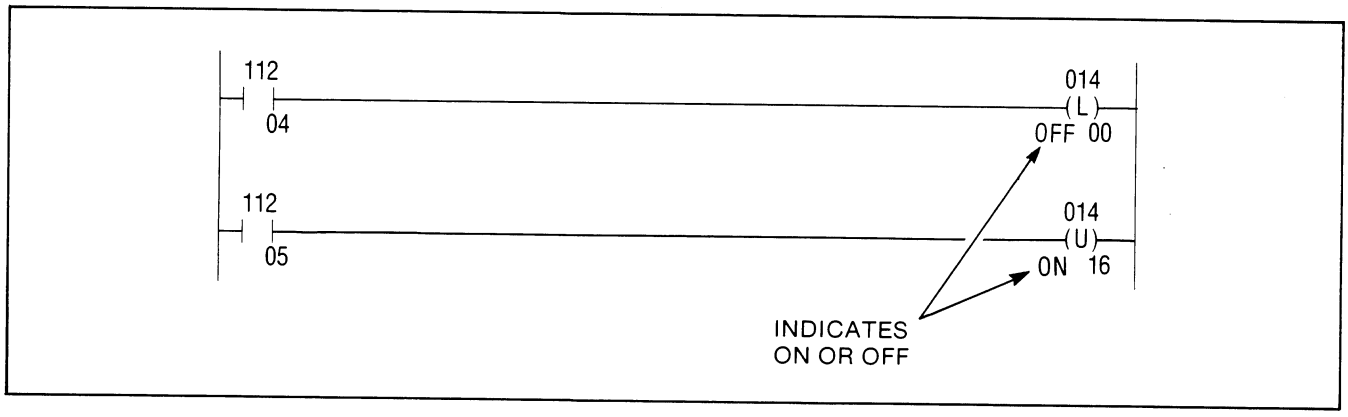


FIGURE 3-8 — LATCH/UNLATCH Indication

immediately be energized, regardless of the rung condition.

WARNING: Do not preset a bit ON controlled by LATCH/UNLATCH instructions if it controls potentially hazardous machine motion. If the bit is preset ON by the LATCH/UNLATCH instructions, the output device controlled by that bit is energized immediately when the Mode Select Switch is turned to the RUN position. Injury to personnel near the machine could result.

Both LATCH and UNLATCH instructions can be programmed unconditionally. This programming technique is generally used with storage bits and should not be used to control output devices.

3.3 BRANCH INSTRUCTIONS

The branch instructions allow more than one combination of input conditions to energize an output device (Figure 3-9).

There are two branch instructions:

- BRANCH START
- BRANCH END

BRANCH START — This instruction begins each parallel logic branch of a rung. The BRANCH START is programmed immediately

before the first instruction of each parallel logic path.

BRANCH END — This instruction completes a set of parallel branches. The BRANCH END is entered after the last instruction of the last branch to end a set of parallel branches.

Branch instructions must be entered in the correct order for proper logic function. The only limitation is that a "nested" branch (a branch within a branch) cannot be programmed directly (Figure 3-10). A total of seven (7) branches can be programmed in one rung and properly displayed on the Industrial Terminal.

3.4 PROGRAMMING RELAY-TYPE INSTRUCTIONS

All relay-type instructions are entered from the Industrial Terminal Keyboard with the Processor in the PROGRAM mode. When a relay type instruction is initially entered, it will appear intensified on the screen to indicate the cursor's present position. When a bit address is required, the instruction will blink to indicate information is needed to complete the instruction. The default bit address, 010/00, is displayed automatically with the instruction. (NOTE: The term "default" simply means that data is to be added.) A reverse-video character cursor is positioned at the first digit. This cursor indicates where information is needed and moves to the next digit as information is entered. When all information is entered, the instruction stops blinking and remains intensified until the next instruction is entered.

Refer to Table 3-1 for a complete summary of relay-type instructions.

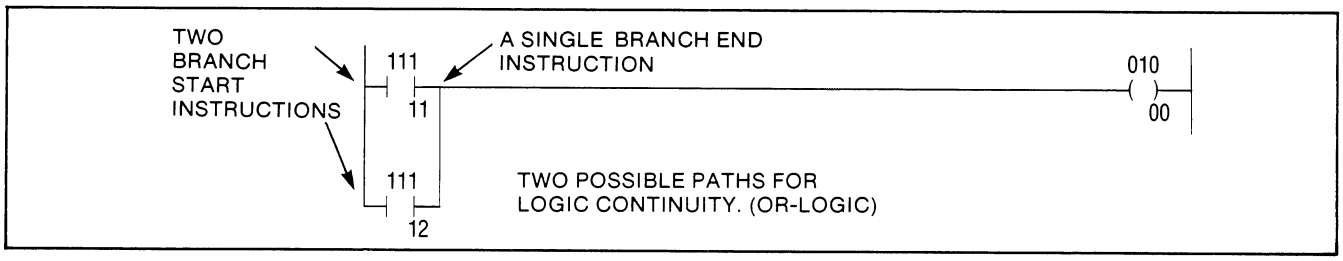


FIGURE 3-9 — Branching Instructions

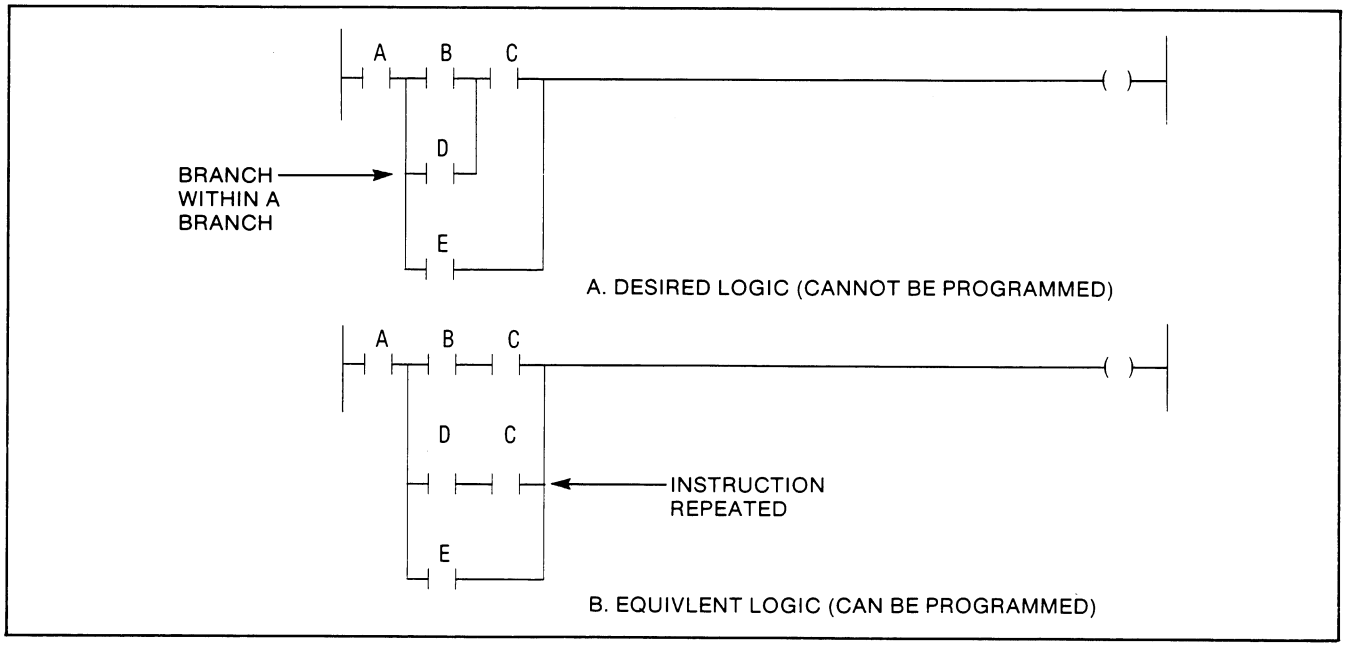


FIGURE 3-10 — Nested Branching vs. Proper Programming

TABLE 3-1 — Relay Type Instructions

Note: Examine and Output addresses, XXX/XX, can be assigned to any location in the Data Table, excluding the Processor Work Areas and as noted below.

KEYTOP SYMBOL	INSTRUCTION NAME	-T1, -T2 DISPLAY	1770-T3 DISPLAY	DESCRIPTION
-| |-	EXAMINE ON	XXXXX -| |-	XXX -| |- XX	When the addressed memory bit is ON, the instruction is TRUE.
-|/|-	EXAMINE OFF	XXXXX -|/|-	XXX -|/|- XX	When the addressed memory bit is OFF, the instruction it TRUE.
-( )-	ENERGIZE	XXXXX -( )-	XXX -( )- XX	[1] When the rung is TRUE, the addressed memory bit is set ON. If the bit controls an output device that output device will be ON.
-(L)-	OUTPUT LATCH	XXXXX -(L)- ON or OFF	XXX -(L)- ON XX or OFF	[1] When the rung is TRUE, the addressed memory bit is latched ON and remains ON until it is unlatched. The OUTPUT LATCH instruction is initially OFF when entered, as indicated below the instruction. It can be preset ON by pressing a [1] after entering the bit address. An ON will then be indicated below the instruction in PROGRAM mode.
-(U)-	OUTPUT UNLATCH	XXXXX -(U)- ON or OFF	XXX -(U)- ON XX or OFF	[1] When the rung is TRUE, the addressed bit is unlatched. If the bit controls an output device that device is deenergized. ON or OFF will appear below the instruction indicating the status of the bit in PROGRAM mode only.
	BRANCH START			This instruction begins a parallel logic path and is entered at the beginning of each parallel path.
	BRANCH END			This instruction ends two or more parallel logic paths and is used with BRANCH START instructions.

[1] These instructions should not be assigned Input Image Table addresses because Input Image Table words are reset by the I/O scan.

Section 4
TIMER AND COUNTER INSTRUCTIONS

4.0 GENERAL

Timer and Counter instructions are output instructions internal to the Processor. They provide many of the capabilities available with timing relays and solid state timing/counting devices. Usually conditioned by Examine instructions, timers and counters keep track of timed intervals or counted events according to the logic continuity of the rung. Up to 40 internal timers and/or counters can be programmed.

Each Timer or Counter instruction has two 3-digit values associated with it, and thus requires two words of Data Table memory. These 3-digit values are:

- Accumulated (AC) Value — Stored in the Accumulated Value area of the Data Table starting at word address 030_8. For timers, this is the number of timed intervals that have elapsed. For counters, this is the number of events that have been counted.
- Preset (PR) Value — Stored in a Preset Value area of the Data Table, always 100_8 words greater than its corresponding AC Value. This value is entered into memory. The Preset value is the number of timed intervals or events to be counted. When the Accumulated value equals the Preset value, a status bit is set ON and can be examined to turn ON or OFF an output device.

The Accumulated and Preset values are stored in the Data Table in 3-digit BCD (Binary Coded Decimal) format. BCD numbers can range from 000 to 999 when stored in the lower 12 bits of a memory word (Figure 4-1). Each BCD digit is represented by a group of 4 bits. The arrangement of "1's" and "0's" in a group of 4 bits corresponds to a decimal number from 0 to 9. For more information on numbering systems, refer to Section 13.

The remaining 4 bits in a word (bits 14-17) are not used to form a BCD number. In the Accumulated Value word, they are used as status bits. In the Preset Value word, they are not used and are available for internal storage. With .01 second timers, these bits are used for internal timing functions and cannot be used for storage. For more information on bit storage, refer to Section 8.3.4.

4.1 TIMER INSTRUCTIONS

A timer counts elapsed time-base intervals (1.0, 0.1 or .01 seconds) and stores this count in its Accumulated Value word. When timing is complete (when AC = PR), bit 15 is either set ON or OFF depending on the type of timer instruction. For all timers, bit 17 is set ON when rung conditions are TRUE and is set OFF when they are FALSE. Both status bits are located in the Accumulated Value word as shown in Figure 4-2.

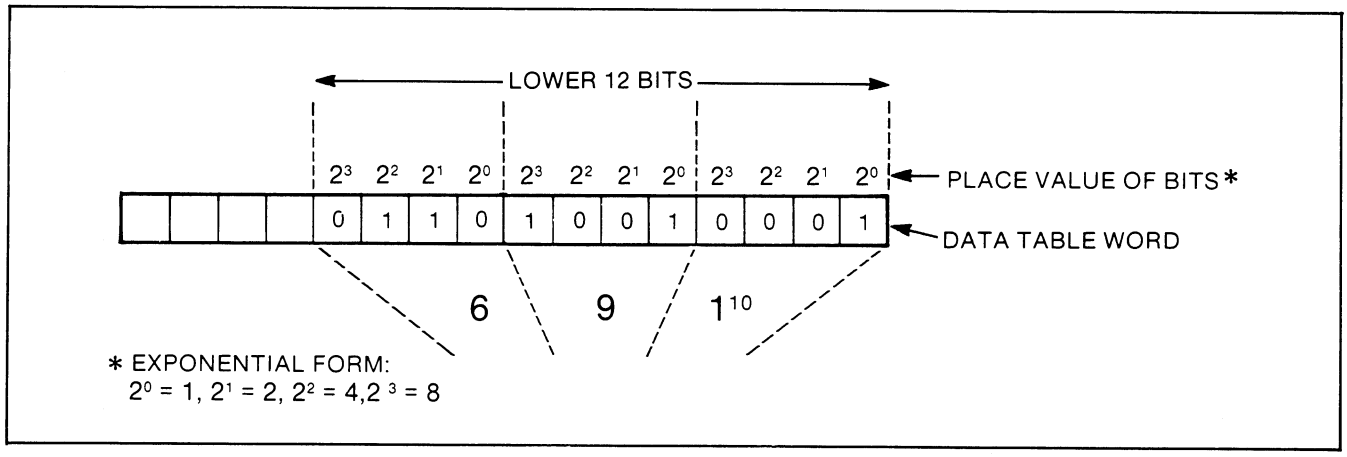


FIGURE 4-1 — BCD Format

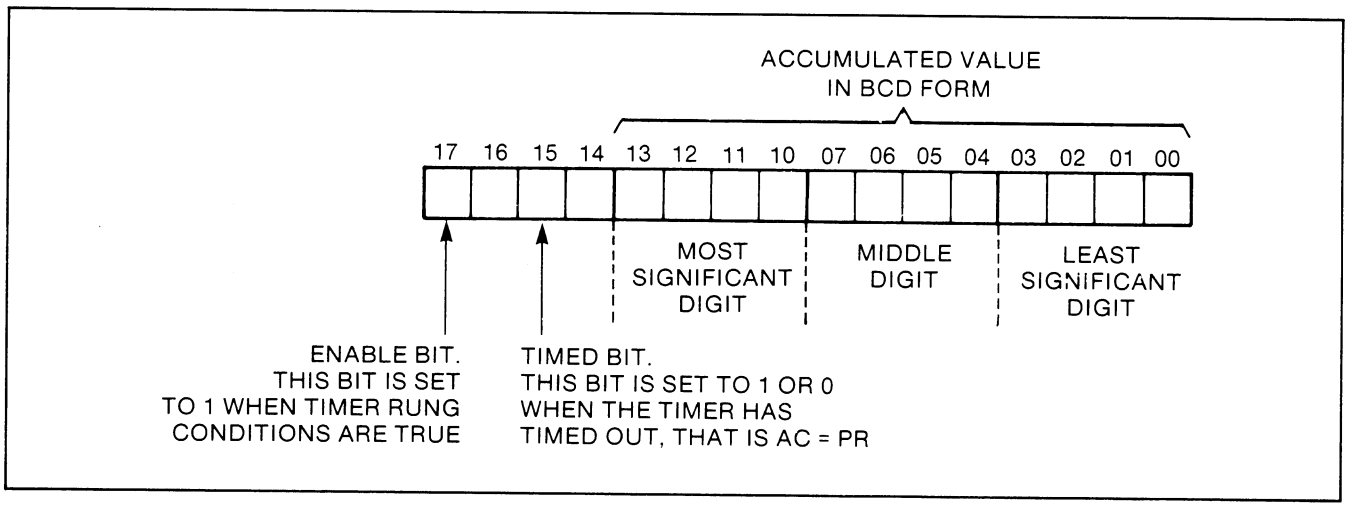


FIGURE 4-2 — Timer Accumulated Value Word

The four timer instructions available with the Mini-PLC-2 Controller are:

- TIMER ON-DELAY -(TON)-
- TIMER OFF-DELAY -(TOF)-
- RETENTIVE TIMER -(RTO)-
- RETENTIVE TIMER RESET -(RTR)-

The timers differ in the way they set and reset status bits, respond to rung logic continuity and reset the Accumulated Value. They are similar in time base selection. One of the following time bases must be selected when entering the instruction.

- 1.0 second
- 0.1 second
- 0.01 second (10 milliseconds)

Bit 16 of the timer Accumulated value word reflects the time base. It will go ON and OFF at the selected time base rate. Therefore, do not use bit 16 of a timer instruction in User Program as an output or storage bit.

4.1.1 TIMER ON-DELAY Instruction

The TIMER ON-DELAY instruction (TON) can be used to turn a device ON or OFF once an interval is timed out.

Refer to Figure 4-3. When the rung condition for a TIMER ON-DELAY instruction becomes TRUE, the timer begins to count time-base intervals. The "Enable" bit, bit 17, is set ON whenever the rung condition is TRUE and the timer is enabled (rung 1). As long as the rung condition remains TRUE, the timer increments its Accumulated value for each interval. When the Accumulated value equals the programmed Preset value, the timer stops incrementing its Accumulated value and sets the "timed" bit, bit 15, of this word ON. Bit 15 is then used to turn an output device ON or OFF, as a condition for program logic (rung 2).

Whenever the rung condition for the TON instruction goes FALSE, the Accumulated value is reset to 000 and bits 15 and 17 of that word are reset to zero. The Accumulated value and status bits are also reset when the Mode Select Switch is turned to the PROG position or when there is a loss of power.

4.1.2 TIMER OFF-DELAY Instruction

The TIMER OFF-DELAY instruction (TOF) can be used to turn a device OFF or ON after a timed interval. Like the other timer instructions, the TOF instruction counts time-base intervals which are stored in its Accumulated Value word. The TOF instruction, however, varies from the other instructions in significant ways.

Refer to Figure 4-4. The TIMER OFF-DELAY instruction begins to time an interval as soon as its rung condition goes FALSE. The Enable bit, bit 17, goes FALSE when the timer begins (rung 1). As long as its rung condition remains FALSE, the TOF instruction continues to time, until the Accumulated Value equals the Preset

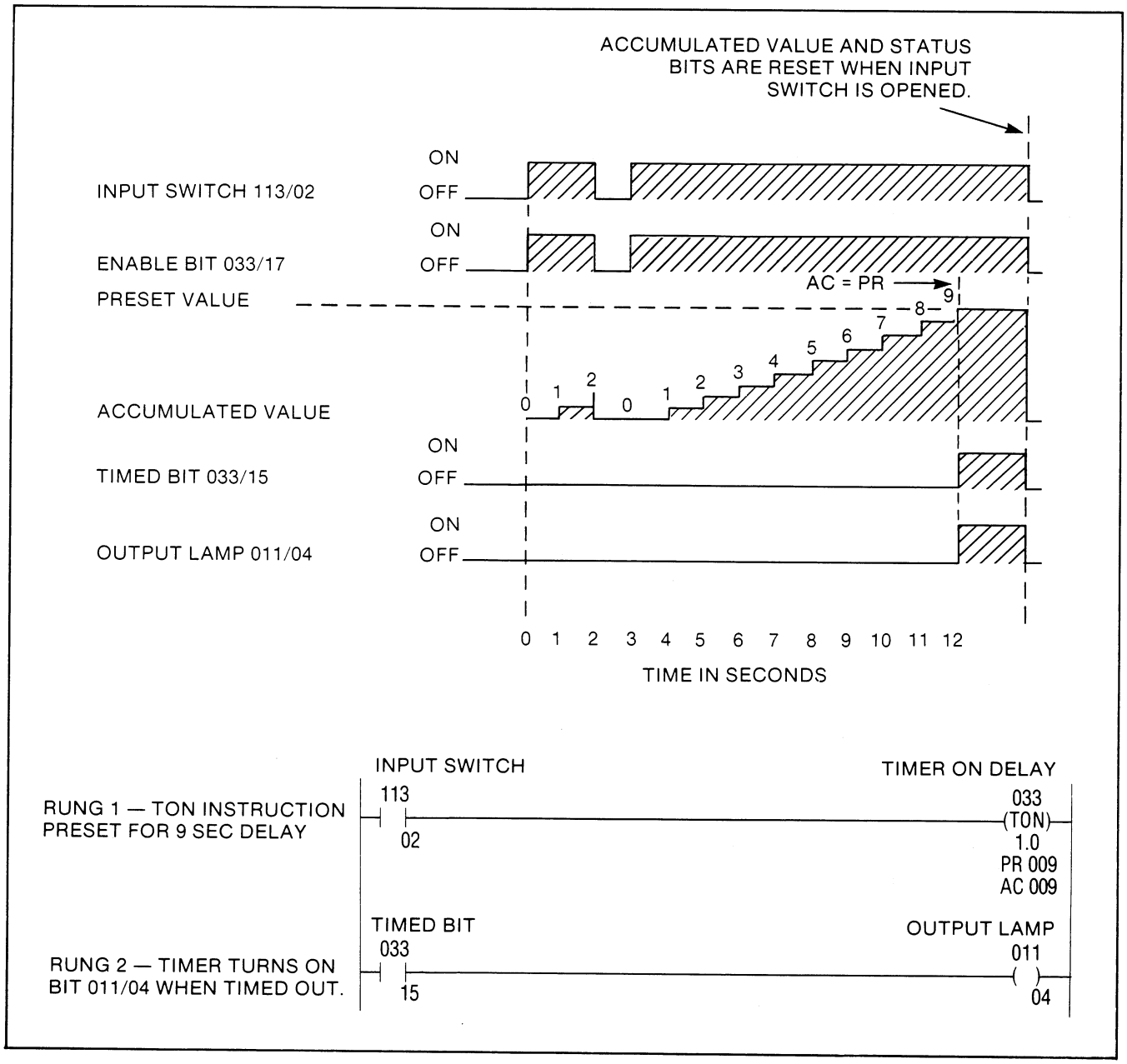


FIGURE 4-3 — TIMER ON-DELAY Timing Diagram & Programming

Value. When the TOF instruction times out, bit 15 is set to zero (OFF) which turns OFF the output (rung 2).

The Accumulated Value is reset to 000 and bit 15 is set ON when the rung condition again goes TRUE. The next timed interval begins when the rung condition goes FALSE.

Bit 17, the enabled bit, is controlled by the logic continuity of the rung. When the rung is TRUE, bit 17 is set to ONE (ON); when it is FALSE, bit 17 is set to zero (OFF).

4.1.3 RETENTIVE TIMER Instruction

The RETENTIVE TIMER instruction (RTO), like the TON instruction, can be used to turn a device ON or OFF once a programmed Preset value is reached.

Unlike the TIMER ON-DELAY instruction, the RETENTIVE TIMER instruction retains its Accumulated value when any of the following conditions occur:

- Rung condition goes FALSE.

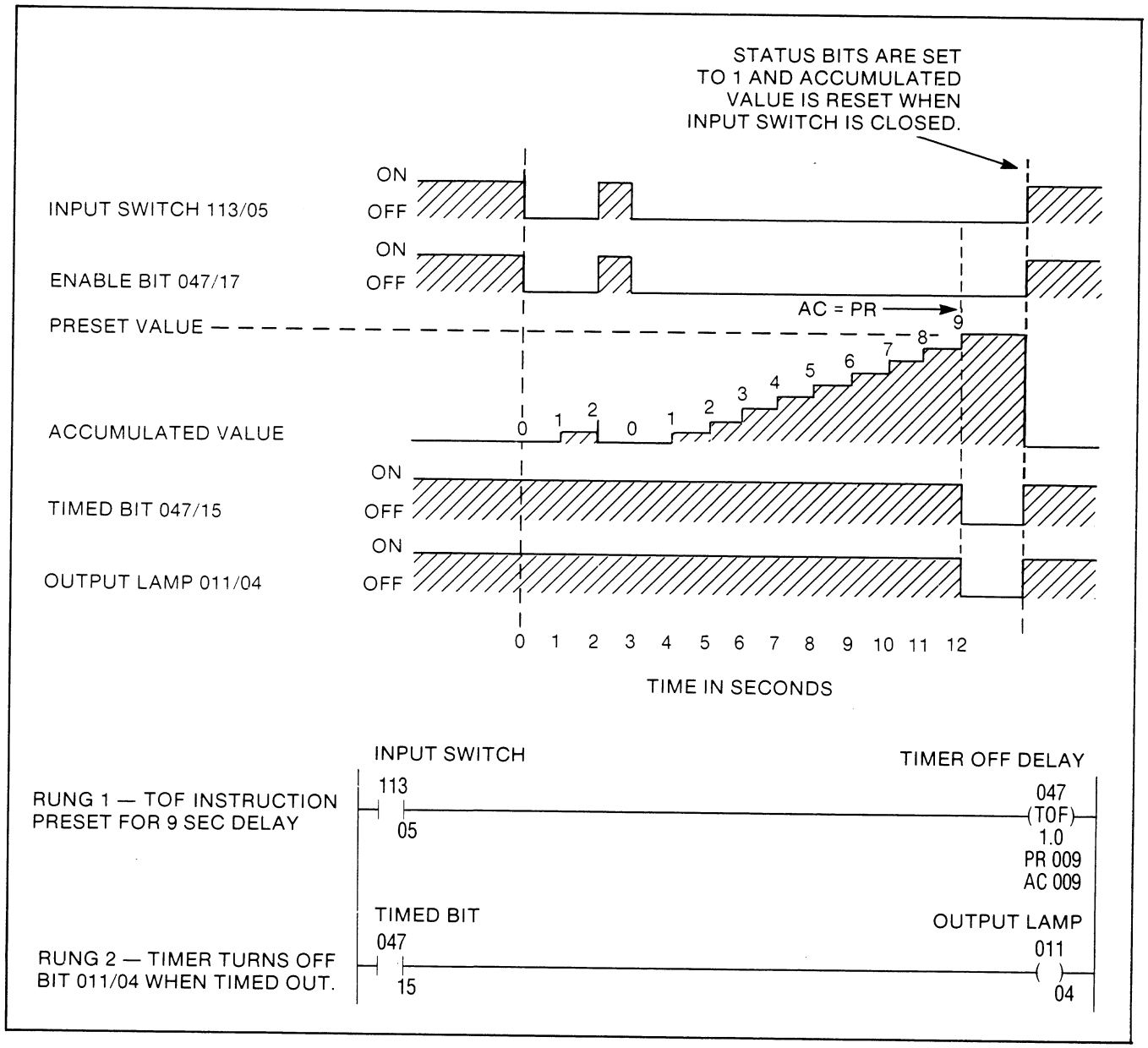


FIGURE 4-4 — TIMER OFF-DELAY Timing Diagram & Programming

- The Mode Select Switch is changed to the PROG position.
- A power outage occurs provided memory backup power is maintained.

Refer to Figure 4-5. When the rung condition goes TRUE, the enabled bit (bit 17) is set ON and the timer starts counting time-base intervals (rung 1). Any time the rung goes FALSE, bit 17 is set OFF but the Accumulated value is retained. When the timer times out, the timed bit (bit 15) is set ON which turns ON an output (rung 2).

By retaining its Accumulated value, the RTO instruction measures the cumulative period during which the rung condition is TRUE. Because this timer retains its Accumulated value, it must be reset by a separate instruction, the RETENTIVE TIMER RESET (RTR) instruction (rung 3).

4.1.4 RETENTIVE TIMER RESET Instruction

The RETENTIVE TIMER RESET instruction (RTR) is used to reset the Accumulated value and timed bit of the Retentive Timer to zero. This instruction is given the same word address

as its corresponding RTO instruction as shown in figure 4-5. When the rung condition goes true, the RTR instruction resets the Accumulated value and status bits of the RTO instruction to zero.

4.1.5 Timer Accuracy for 10 msec Timers

The accuracy of the 10 msec timer is related to nominal scan time. When scan times are 9 msec or less, the 10 msec timer is accurate to

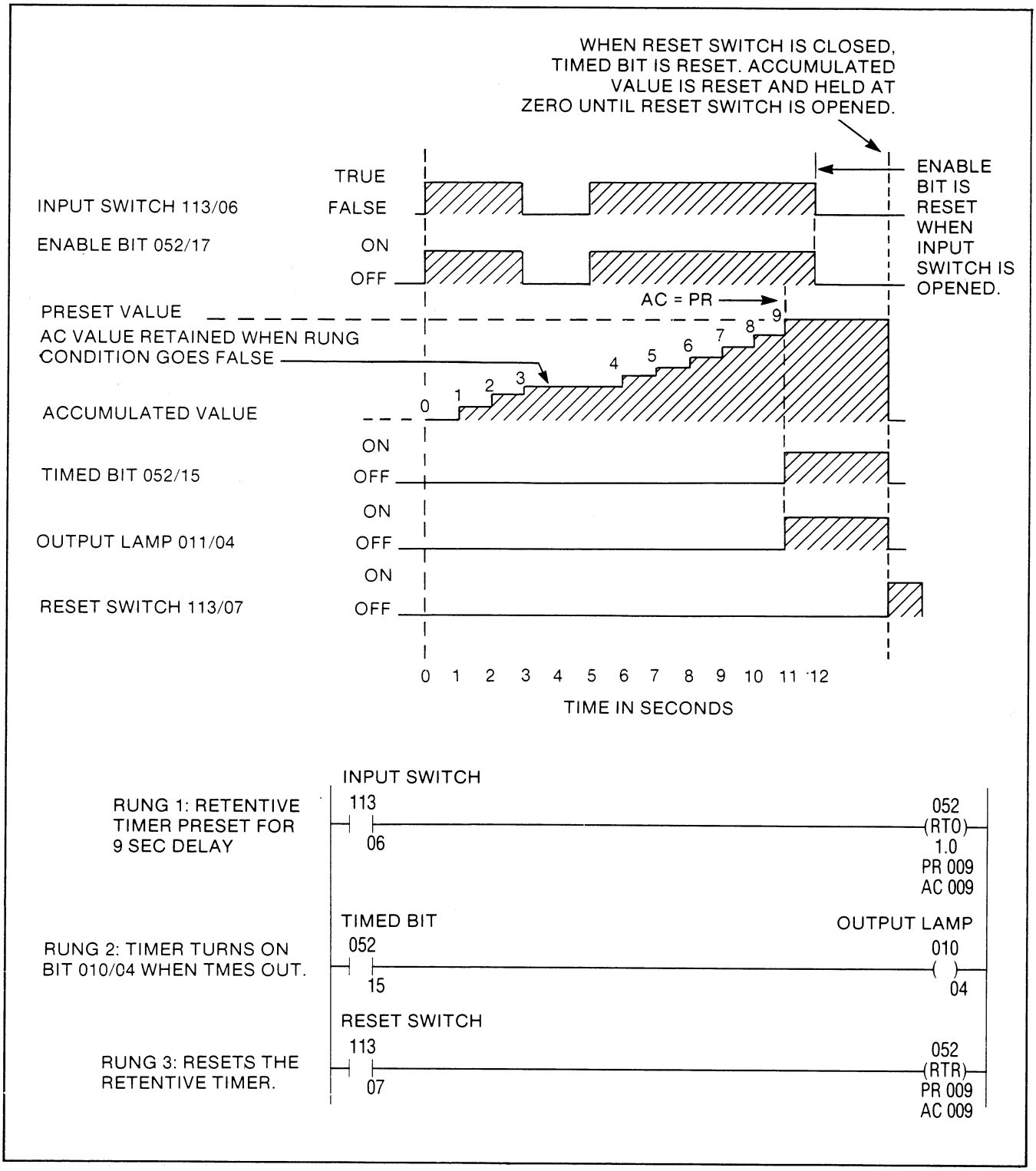


FIGURE 4-5 — RETENTIVE TIMER with RESET Timing Diagram & Programming

plus or minus one time base (±10 msec). When the scan time is greater than 9 msec, accuracy of ±10 msec can be achieved through special programming techniques described in Programming 0.01-Second Timers, Section 11.4.

4.2 COUNTER INSTRUCTIONS

Three types of counter instructions are available with the Mini-PLC-2 Controller. They are:

- UP-COUNTER -(CTU)-
- COUNTER RESET -(CTR)-
- DOWN-COUNTER -CTD)-

A counter counts the number of events that occur and stores this count in its Accumulated Value word. The remaining four bits in the Accumulated Value word are used as status bits. See Figure 4-6.

- Bit 14 is the Overflow/Underflow bit. It is set to 1 when the AC value of the CTU instruction exceeds 999 or the AC value of the CTD instruction goes below 000.
- Bit 15 is set to 1 when a count has been reached or exceeded, that is, when the AC value is ⩾ PR value.
- Bit 16 is the Enable bit for a CTD instruction. It is set ON when the rung condition is TRUE.
- Bit 17 is the Enable bit for a CTU instruction. It is set ON when the rung condition is TRUE.

Counter instructions differ from Timer instructions because they have no time-base. They count one event each FALSE-to-TRUE transition of the rung condition.

4.2.1 UP-COUNTER Instruction

The UP-COUNTER instruction (CTU) increments its Accumulated value for each FALSE-to-TRUE transition of the rung condition. Because only the FALSE-to-TRUE transition causes a count to be made, the rung condition must go from TRUE to FALSE and back to TRUE before the next count is registered. The CTU instruction retains its Accumulated value when:

- Mode Select Switch is changed to the PROG position.

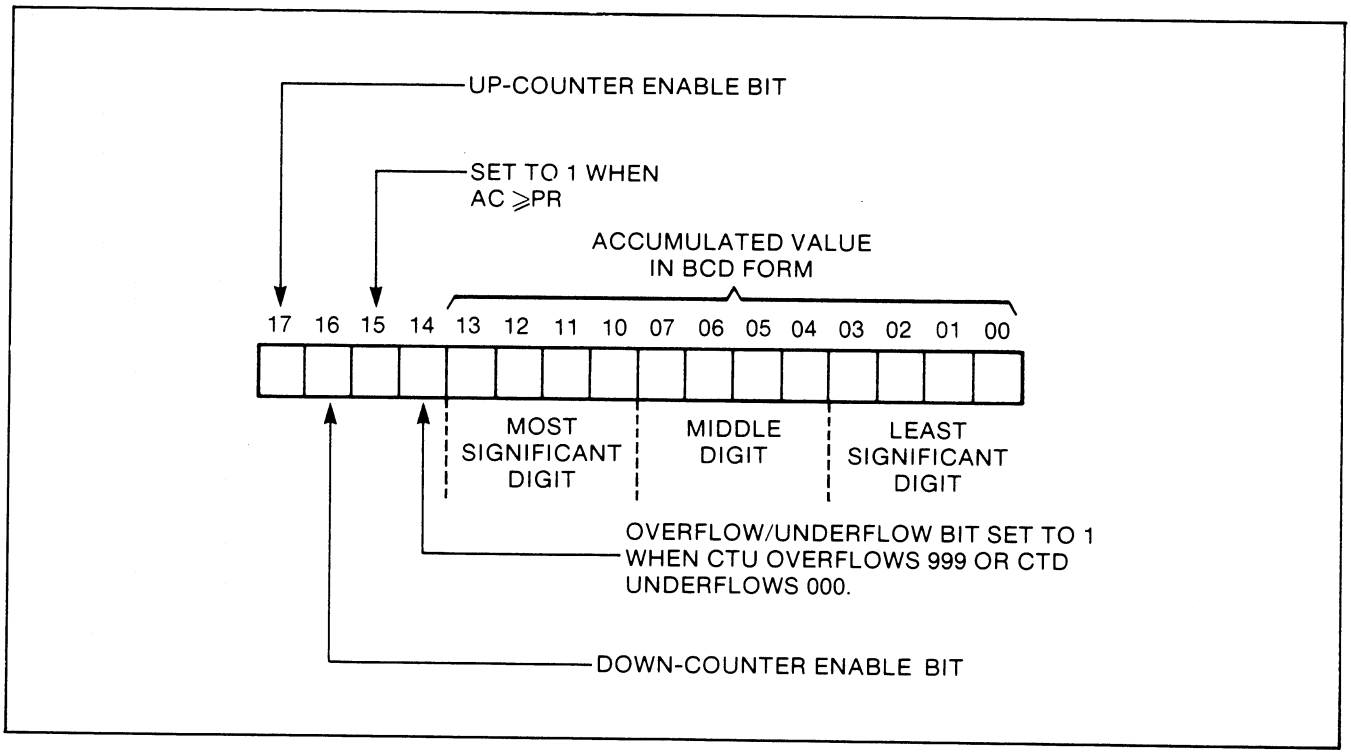


FIGURE 4-6 — Counter Accumulated Value Word

- The rung condition goes FALSE.
- A power outage occurs provided memory backup power is maintained.

Refer to Figure 4-7. Each time the CTU rung goes TRUE, bit 17, the Enable bit, is set ON (rung 1). When the Accumulated value reaches the Preset value, the Count Complete bit, bit 15, is set ON (rung 2). Unlike a timer, the CTU instruction continues to increment its Accumulated value after the Preset value has been reached. If the Accumulated value goes above 999, bit 14 is set ON to indicate an overflow condition and the CTU continues up-counting from 000 (rung 3). Bit 14 stays ON until the counter is reset. Bit 14 can be examined to cascade counters for counts greater than 999 (Section 4.3). Because this counter retains its Accumulated value, it must be reset by a separate instruction, the COUNTER RESET (CTR) instruction.

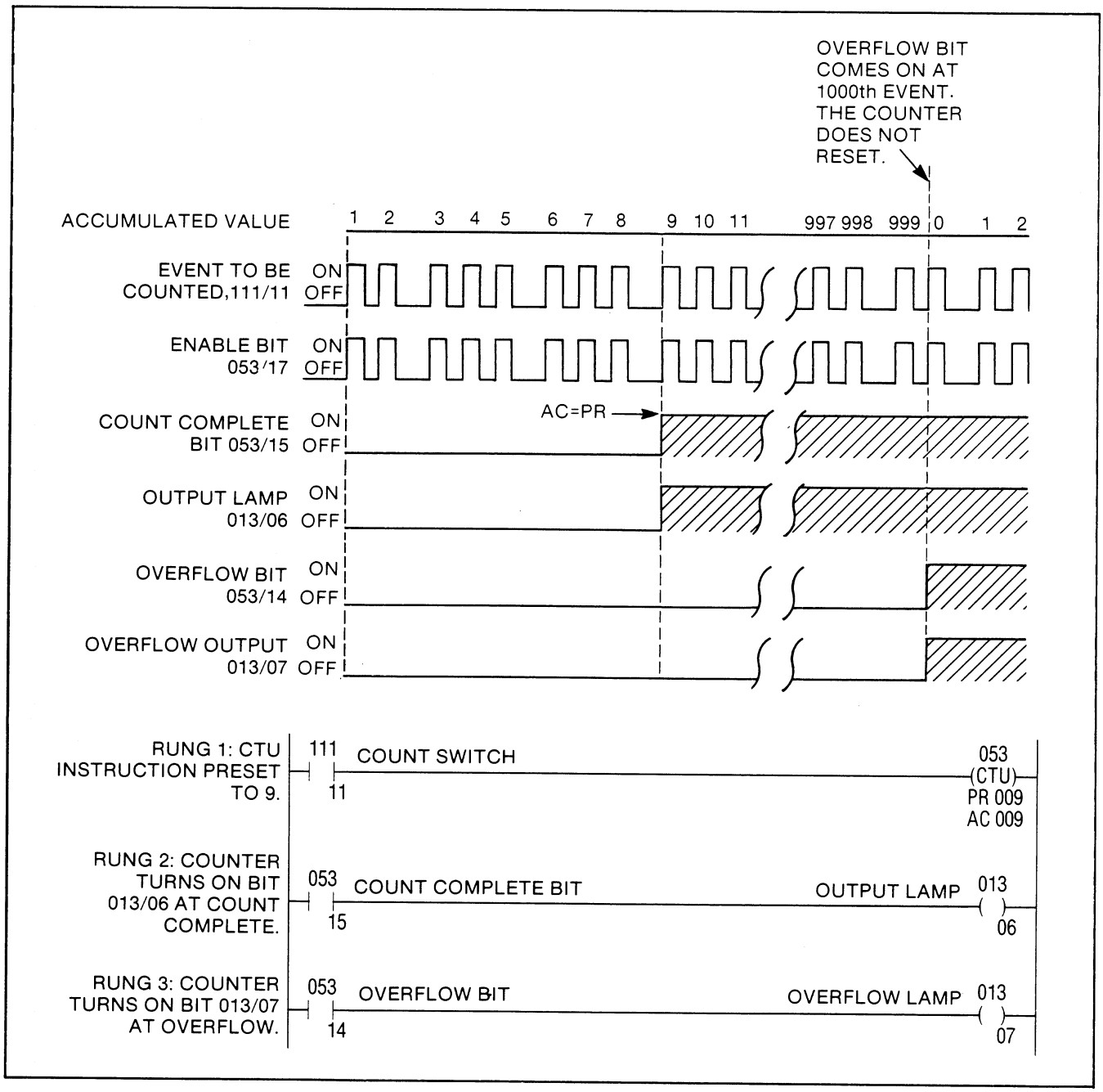


FIGURE 4-7 — UP-COUNTER Diagram & Programming

4.2.2 COUNTER RESET Instruction

The COUNTER RESET instruction (CTR) is an output instruction that resets the CTU Accumulated value and status bits to zero when the reset rung goes TRUE.

Refer to Figure 4-8. The counter operates in the same manner as described for the CTU instruction, with the addition of the Reset instruction (rung 3).

In this example, the reset pushbutton is pressed after count 12. When the pushbutton is released, the next event starts the sequence at count 1.

The CTR instruction is given the same word address as the CTU instruction. The Preset and Accumulated values are automatically displayed when the word address is entered.

4.2.3 DOWN-COUNTER Instruction

The DOWN-COUNTER instruction (CTD) subtracts one from its Accumulated value for each FALSE-to-TRUE transition of its rung condition. Because only the FALSE-to-TRUE transition causes a count to be made, the rung condition must go from TRUE to FALSE and back to TRUE before the next count is registered.

The CTD instruction Accumulated value is retained when:

- Mode Select Switch is changed to the PROG position.
- The rung condition goes FALSE.
- A power outage occurs provided memory backup power is maintained.

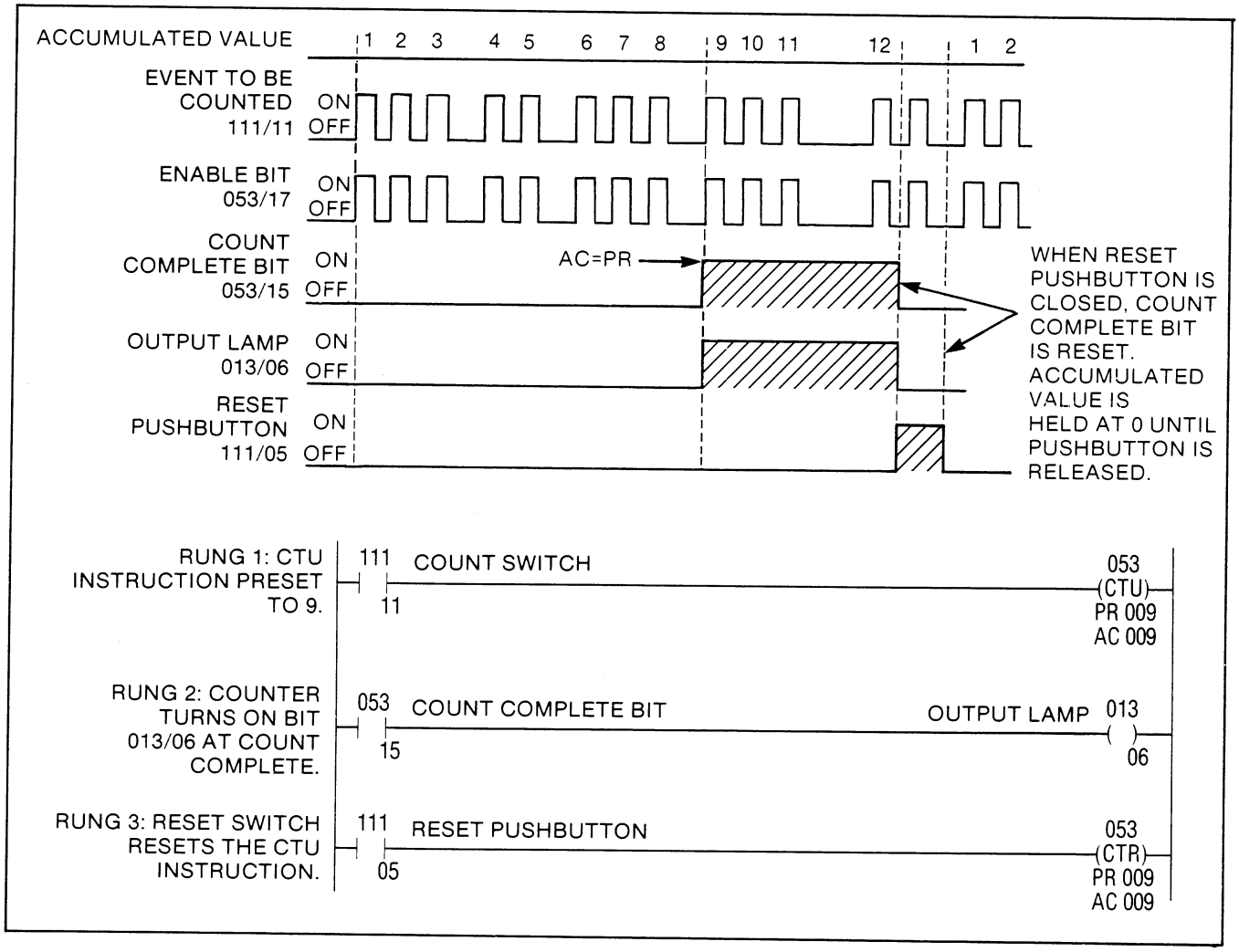


FIGURE 4-8 — UP-COUNTER with RESET Diagram & Programming

Each time the CTD instruction rung goes TRUE, bit 16, the Enable bit, is set ON. When the Accumulated value is greater than or equal to the Preset value, bit 15 is set ON. When the Accumulated value goes below 000, bit 14 is set ON to indicate an underflow condition and the CTD instruction continues down-counting from 999.

Normally, the DOWN-COUNTER instruction is paired with the UP-COUNTER instruction to form an up/down counter, using the same word address, AC value and PR value as shown in Figure 4-9.

NOTE: Bit 14 of the Accumulated Value word is set ON when the Accumulated value either "overflows" or "underflows." Because of this, bit 14 may require monitoring in some applications. When a DOWN-COUNTER Preset is set to 000, Underflow bit 14 will not be set ON when the count goes below zero.

When used alone, the CTD instruction's Accumulated value may need to be "reset" in the program to its original value (usually a value other than 000). For this reason, a GET/PUT transfer (described in Section 5.1) rather than a CTR instruction is usually used to load a value in the CTD instruction's Accumulated Value word.

4.3 CASCADING TIMERS OR COUNTERS

An individual timer or counter can time or count up to 999 intervals or events. By "cascading" two or more timers or counters, the timing or counting capability within the program can be increased beyond three digits.

To cascade timers or counters, each timer or counter is assigned a different word address (Figure 4-10). The status bit of the first timer (bit 15) changes status each time the Preset value is reached. The status bit of a counter (bit 14) is set ON each time a counter overflows. The status bit of the timer or counter is then used to increment the second timer or counter and reset the first to 000.

4.4 PROGRAMMING TIMER AND COUNTER INSTRUCTIONS

Timer and Counter instructions are programmed from the Industrial Terminal keyboard with the Processor in the PROGRAM mode. Allowable addresses are 030_8 through 077_8.

Timer instructions are programmed by entering the word address of the Accumulated value, a time base and a Preset value. With the RTO instruction, an Accumulated value can also be entered. The time base of 1.0, 0.1 or 0.01 second is entered as [1][0], [0][1], or [0][0] respectively.

Counter instructions are programmed by entering the word address of the Accumulated value, a Preset value, and if desired, an Accumulated value. Press [CANCEL COMMAND] if no Accumulated value is desired.

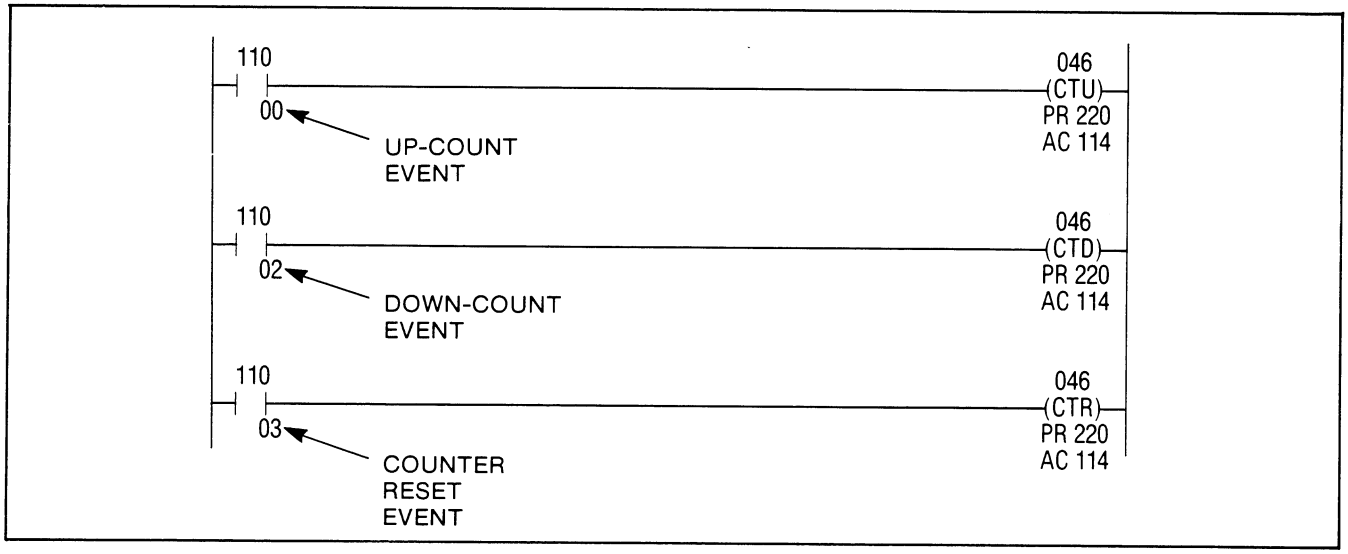


FIGURE 4-9 — UP/DOWN COUNTER Example

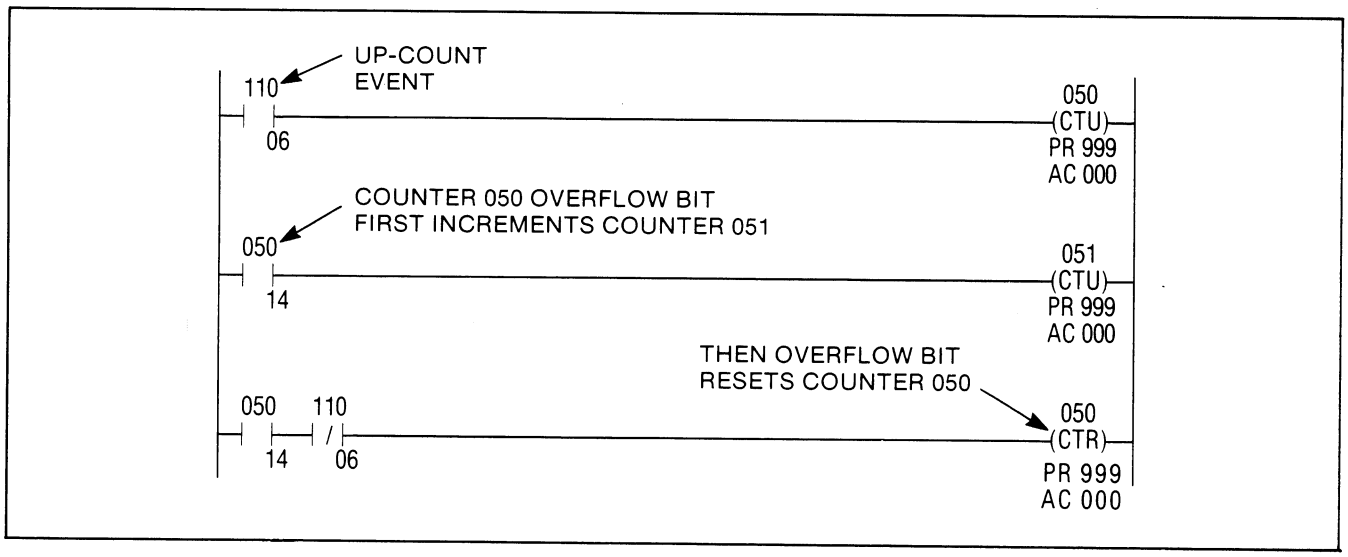


FIGURE 4-10 — Cascading Counters Example

When entered, these instructions will be displayed as intensified and blinking. The default word address above the instruction will have a reverse-video cursor positioned at the first digit. The instruction will continue to blink until all the data is entered.

Refer to Table 4-1 and 4-2 for a complete summary of Timer and Counter instructions.

TABLE 4-1 — Timer Instructions

Note: The Timer word address, XXX, is assigned to the timer Accumulated area of the Data Table.

The time base, TB, is user-selectable and can be 1.0, 0.1 or 0.01 second. Preset values YYY and Accumulated values ZZZ can vary from 000 to 999.

KEYTOP SYMBOL	INSTRUCTION NAME	DISPLAY	DESCRIPTION
-(TON)-	TIMER ON DELAY	XXX -(TON)- TB PR YYY AC ZZZ	When the rung is TRUE, the timer begins to increment the Accumulated Value at a rate specified by the time base. When the rung is FALSE, the timer resets the Accumulated Value to 000.
-(TOF)-	TIMER OFF DELAY	XXX -(TOF)- TB PR YYY AC ZZZ	When the rung is FALSE, the timer begins to increment the Accumulated Value. When the rung is TRUE, the timer resets the Accumulated Value to 000.
-(RTO)-	RETENTIVE TIMER	XXX -(RTO)- TB PR YYY AC ZZZ	When the rung is TRUE, the timer begins to increment the Accumulated Value. When FALSE, the Accumulated Value is retained. It is reset only by the RTR instruction.
-(RTR)-	RETENTIVE TIMER RESET	XXX -(RTR)- PR YYY AC ZZZ	XXX — Word address of the retentive timer it is resetting. YYY, ZZZ — Preset and Accumulated Values are automatically entered by the Industrial Terminal. When the rung is TRUE, the Accumulated Value and status bit are reset to zero.

TABLE 4-2 — Counter Instructions

Note: The Counter word address, XXX, is assigned to the counter Accumulated areas of the Data Table. Preset values YYY and Accumulated values ZZZ can vary from 000-999.

KEYTOP SYMBOL	INSTRUCTION NAME	DISPLAY	DESCRIPTION
-(CTU)-	UP COUNTER	XXX -(CTU)- PR YYY AC ZZZ	Each time the rung goes TRUE, the Accumulated Value is incremented one count. The counter will continue counting after the Preset Value is reached. The Accumulated Value can be reset by the CTR instruction. The Accumulated Value "Overflow" bit is bit 14.
-(CTR)-	COUNTER RESET	XXX -(CTR)- PR YYY AC ZZZ	XXX - Word address of the CTU it is resetting. Preset and Accumulated Values are automatically entered by the Industrial Terminal. When the rung is TRUE, the CTU Accumulated Value and status bits are reset to 000.
-(CTD)-	DOWN COUNTER	XXX -(CTD)- PR YYY AC ZZZ	Each time the rung goes TRUE, the Accumulated Value is decreased one count. The Accumulated Value "Underflow" bit is bit 14. The Enable bit is bit 16.

Section 5
DATA MANIPULATION INSTRUCTIONS

5.0 GENERAL

The Data Manipulation instructions are used to transfer or compare data that is stored in Data Table words and bytes. There are six Data Manipulation instructions:

- GET
- PUT
- LES
- EQU
- GET BYTE
- LIMIT TEST

The GET and PUT instructions are used together to transfer 16 bits of data from one word location in the Data Table to another word location. Data can be in the form of 3-digit Binary Coded Decimal numbers.

The LES and EQU instructions compare data such as 3-digit numberic values in BCD format using the first 12 bits of a Data Table word (Figure 5-1). This 3-digit value can be a decimal number ranging from 000 to 999.

The GET BYTE and LIMIT TEST instructions compare 3-digit values in octal format using eight bits (one byte) of a Data Table word (Figure 5-2). This 3-digit value is an octal number ranging from 000_8 to 377_8. Note that two 3-digit values can be stored in a word: one in the upper byte (bits 10-17) and one in the lower byte (bits 00-07).

A Data Manipulation instruction can address any word in the Data Table excluding the Processor Work Areas.

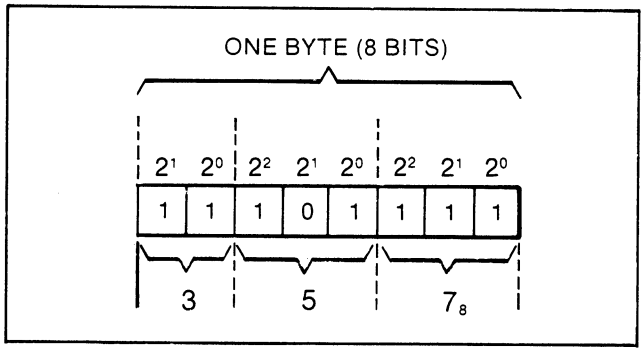


FIGURE 5-2 — Octal Representation

5.1 DATA TRANSFER INSTRUCTIONS

There are two Data Transfer instructions. They are:

- GET -[G]-
- PUT -(PUT)-

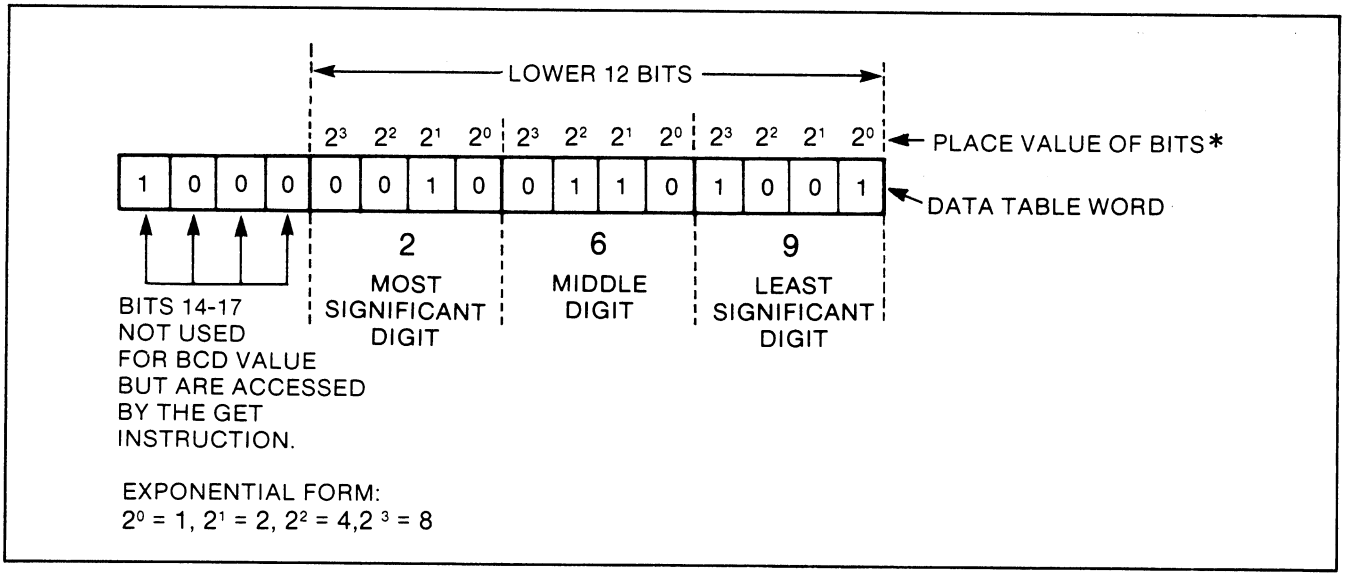


FIGURE 5-1 — BCD Word Format

5.1.1 GET Instruction

GET instructions -[G]- are programmed in the condition area of the ladder diagram rung. When the rung containing the GET/PUT instructions goes TRUE, the data (16 bits) in the word address of the GET instruction is duplicated and transferred to the word address of the PUT instruction. (Figure 5-3).

If the word addressed by a GET instruction already contains data, the lower 12 bits of the data are displayed automatically beneath the instruction after the word address is entered. Entry of new data such as a BCD value writes over the data previously stored in the addressed word.

Although each Data Table word can store data such as one BCD value, the word address can be assigned to more than one GET instruction in the same program. This allows the program to perform several different functions with the same data.

The GET instruction is NOT a "condition" that determines rung logic continuity. When the Processor is in the RUN or TEST mode, the GET instruction is always intensified regardless of rung logic continuity. This does not mean that data transfer will occur. Data transfer occurs only when the rung is TRUE.

The GET instruction can be programmed either at the beginning of a rung or with one or more Condition instructions preceding it. Condition instructions, however, should not be programmed after a GET instruction. When one or more Condition instructions precede the GET instruction, they determine whether the rung is TRUE or FALSE. Parallel branches of GET instructions cannot be programmed unless they are paired with a LES or EQU instruction.

5.1.2 PUT Instruction

The PUT instruction (PUT) is an output instruction. It receives 16 bits of data from its corresonding GET instruction and stores the data at its address as shown in Figure 5-3. A PUT instruction can have the same address as other instructions in the program. For example, a PUT instruction having the same address as a counter Preset will change the counter Preset value to that transferred from the GET instruction. See Figure 5-4.

The lower 12 bits of transferred data are displayed in BCD beneath the PUT instruction.

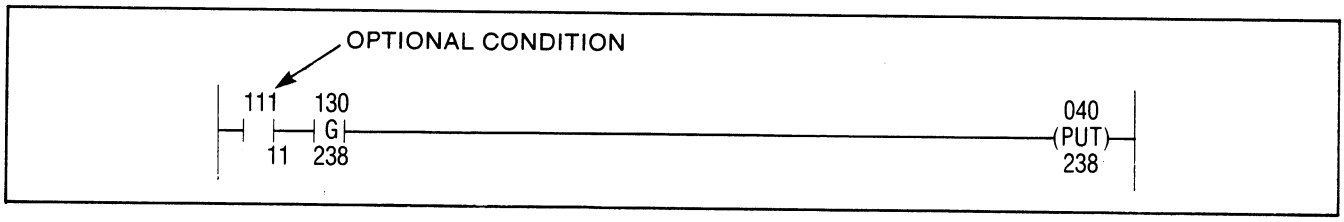


FIGURE 5-3 — GET and PUT Instructions

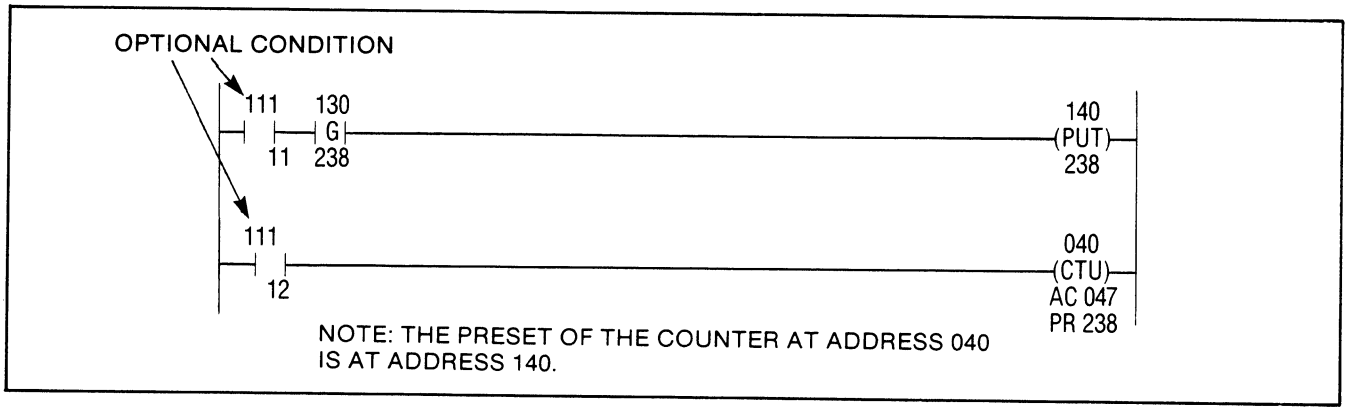


FIGURE 5-4 — Changing a Counter Preset

Bits 14-17 are not displayed but are transferred. While the rung is TRUE, any change in the data of the GET instruction also changes the data of the PUT instruction. However, the PUT instruction is retentive, which means that while the rung is FALSE, any change in the data of the GET instruction does not change the data of the PUT instruction. Also, during a power loss, the data is retained.

5.2 DATA COMPARISON INSTRUCTIONS

The Data Comparison instructions are:

- LESS THAN -[<]-
- EQUAL TO -[=]-
- GET BYTE -[B]-
- LIMIT TEST -[L]-

Data Comparison operations differ from Data Transfer operations because Data Table values are not transferred. Instead, the values at different word locations are compared.

Data Comparison instructions operate with either BCD values or octal values. With the LES and EQU instructions, only 12 bits of a word (the BCD values) are compared. Bits 14-17 are not compared. With the GET BYTE and LIMIT TEST instructions, 8 bits (one byte) of a word are compared.

5.2.1 LES and EQU Instructions

The LES (less than) and EQU (equal to) instructions, [<] and [=] are used with the GET instruction to perform data comparisons. They compare BCD values and are programmed in the condition area of the ladder diagram rung.

A GET/LES or GET/EQU pair of instructions forms a single condition for logic continuity. Alone or with other conditions, each pair can be used to energize an output device or other output instruction. In all cases, the GET instruction must be programmed before the LES or EQU instruction. If other conditions are also programmed, they should be entered before the GET instruction or after the LES or EQU instruction.

Data comparisons are made by comparing a changing BCD value to a reference BCD value. The reference value need not be fixed. The following types of data comparisons of BCD values can be made:

- Less than
- Greater than
- Equal to
- Less than or equal to
- Greater than or equal to

LESS THAN — A less-than comparison is made with the GET/LES pair of instructions. The BCD value of the GET instruction is the changing value. It is compared to the BCD value of the LES instruction which is the reference value (Figure 5-5). When the GET value is less than the LES value, the comparison is TRUE and logic continuity is established.

GREATER THAN — A greater-than comparison is also made with the GET/LES pair of instructions. This time the GET instruction BCD value is the reference and the LES instruction BCD value is the changing value.

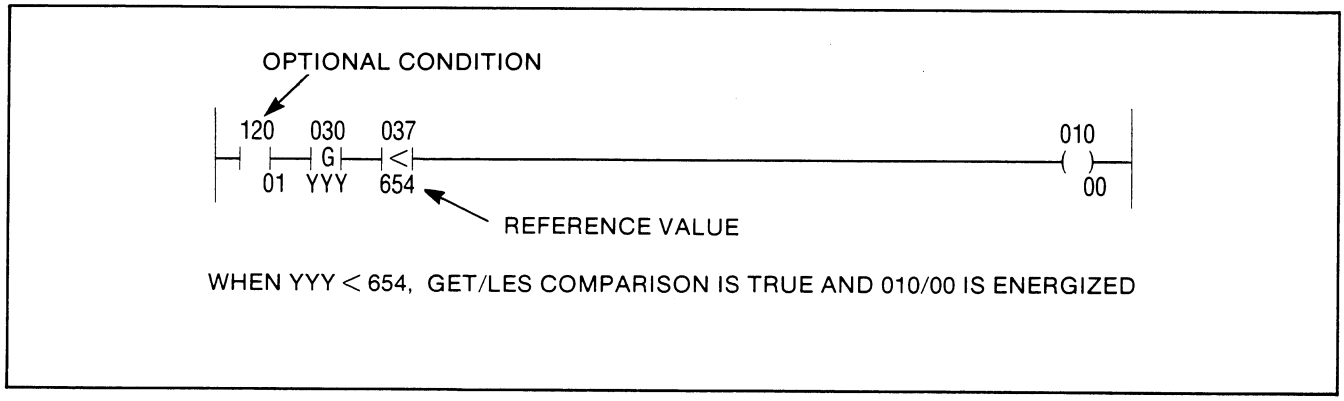


FIGURE 5-5 — LESS THAN Comparison

The LES value is compared to the GET value for a greater-than condition (Figure 5-6). When the LES value is greater than the GET value, the comparison is TRUE and logic continuity is established.

EQUAL TO — An equal-to comparison is made with the GET and EQU instructions (Figure 5-7). The GET value is the changing variable and is compared to the reference value of the EQU instruction for an equal to condition. When the GET value equals the EQU value, the comparison is TRUE and logic continuity is established.

LESS THAN OR EQUAL TO — This comparison is made using the GET, LES and EQU instructions. The GET value is the changing value. The LES and EQU instructions are assigned a reference value (Figure 5-8). When the GET value is either less than or equal to the value at the LES and EQU instructions, the comparison is TRUE and logic continuity is established.

NOTE: Only one GET instruction is required for a parallel comparison. The LES and EQU instructions are programmed on parallel branches.

GREATER THAN OR EQUAL TO — This comparison is made using the GET, LES and EQU instructions. The GET value is assigned a reference value. The LES and EQU values are changing values that are compared to the GET value (Figure 5-9). When the LES and EQU values are greater than or equal to the GET value, the comparison is TRUE and logic continuity is established.

NOTE: Only one GET instruction is required for this parallel comparison. The LES and EQU instructions are programmed on parallel branches.

5.2.2 GET BYTE and LIMIT TEST Instructions

The GET BYTE and LIMIT TEST instructions [B] and [L] are used together to compare an octal value to upper and lower limits that are

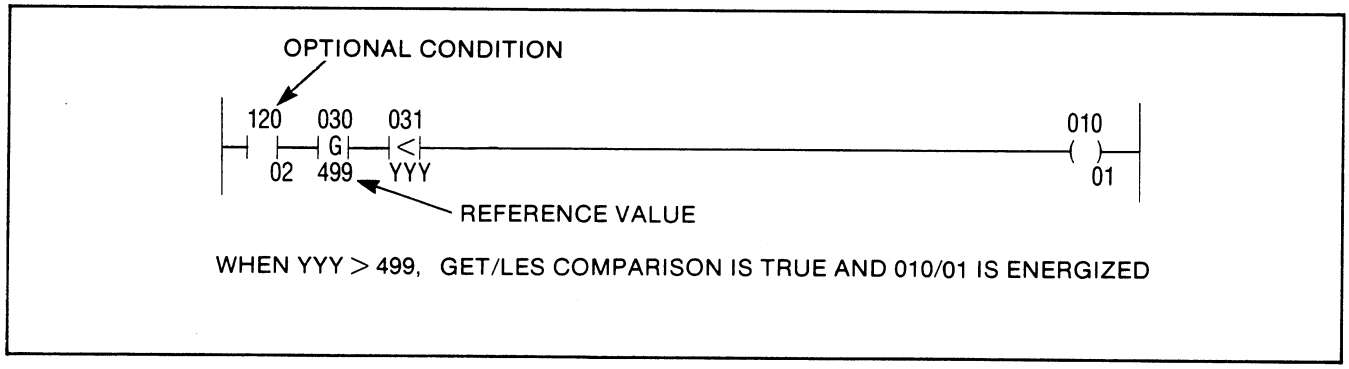


FIGURE 5-6 — GREATER THAN Comparison

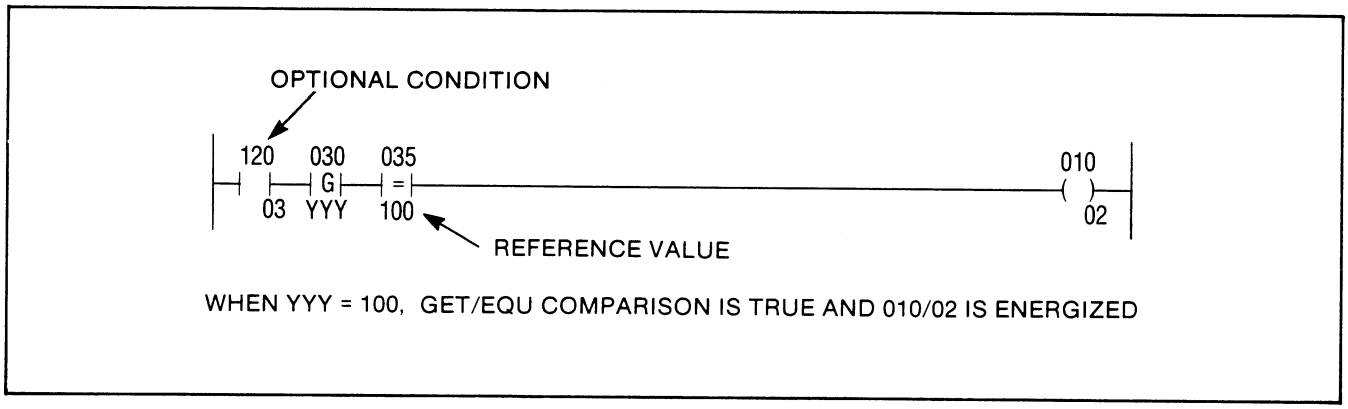


FIGURE 5-7 — EQUAL TO Comparison

also octal values. These values can range from 000_8 to 377_8.

The GET BYTE and LIMIT TEST instructions are programmed in the condition area of the ladder diagram rung. Together they form a single condition for logic continuity. Condition instructions can be programmed before the GET BYTE instruction or after the LIMIT TEST instruction but not between them.

The GET BYTE instruction addresses either the upper or lower byte of a Data Table word. A "1" is entered after the word address for an upper byte; a "0" is entered for the lower byte.

The LIMIT TEST instruction addresses one Data Table word that stores both the upper and lower limits. The upper limit is stored in the upper byte and the lower limit is stored in the lower byte. The upper byte of word 045_8 would be addressed as 0451. See Figure 5-10.

The Processor makes a duplicate of the upper or lower byte of the word addressed by the GET BYTE instruction. The octal value stored at that byte is then compared to the upper and lower octal values of the LIMIT TEST instruction. If the GET BYTE value is equal to or between the LIMIT TEST values, the comparison is TRUE and logic continuity is established.

5.3 PROGRAMMING DATA MANIPULATION INSTRUCTIONS

The Data Manipulation instructions are programmed from the Industrial Terminal keyboard with the Processor in the PROGRAM mode. When entered, they are displayed intensified and blinking. The default word address above the instruction will have a reverse-video cursor positioned at the first digit. The instruction will continue to blink until all information is entered.

Refer to Table 5-1 for a summarized description of these instructions.

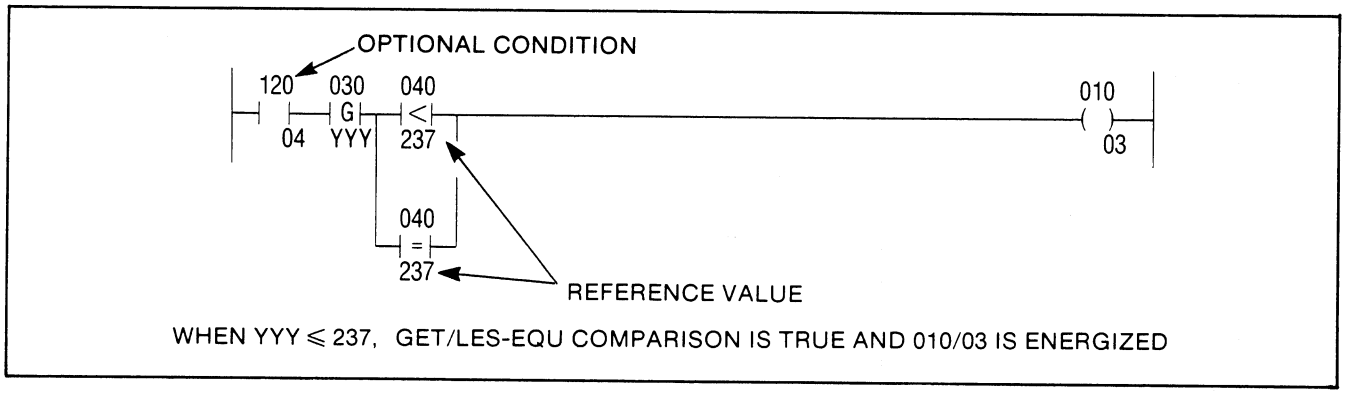


FIGURE 5-8 — LESS THAN or EQUAL TO Comparison

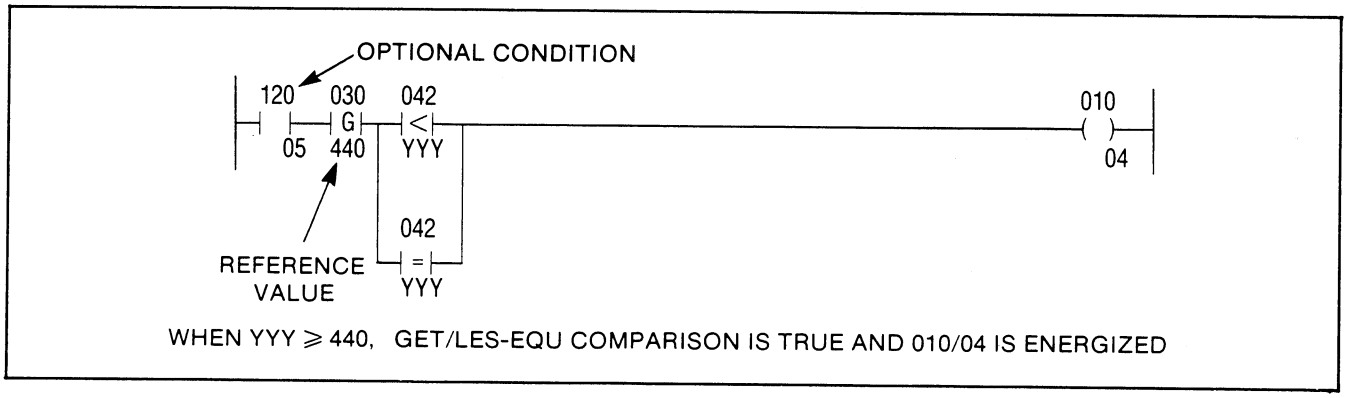


FIGURE 5-9 — GREATER THAN or EQUAL TO Comparison

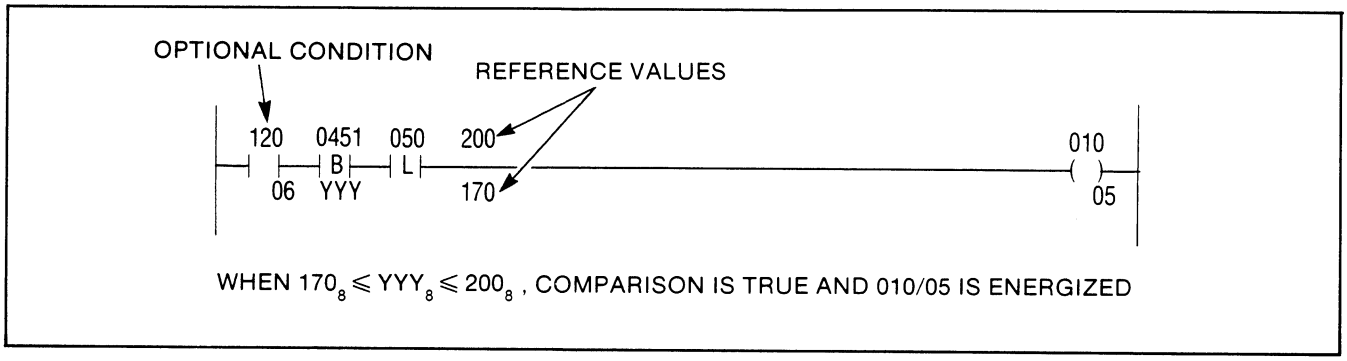


FIGURE 5-10 — GET BYTE/LIMIT TEST Comparison

TABLE 5-1 — Data Manipulation Instructions

Note: Data Manipulation instructions operate upon BCD values and/or 16 bit data in the Data Table. The word address XXX is displayed above the instruction; the BCD value or data operated upon YYY is displayed beneath it. The BCD value is stored in the lower 12 bits of the word address and can be any value from 000 to 999, except as noted.

KEYTOP SYMBOL	INSTRUCTION NAME	DISPLAY	DESCRIPTION
-[G]-	GET	XXX -[G]- YYY	The GET instruction is used with other Data Manipulation or Arithmetic instructions. When the rung is TRUE, all 16 bits of the GET instruction are duplicated and the operation of of the instruction following it is performed.
-(PUT)-	PUT	XXX -(PUT)- YYY	The PUT instruction should be preceded by the GET instruction. When the rung is TRUE, all 16 bits at the GET instruction address are transferred to the PUT instruction address.
-[<]-	LESS THAN	XXX -[<]- YYY	The LESS THAN instruction should be preceded by a GET instruction. 3-Digit BCD values at the GET and LESS THAN word addresses are compared. If the logic is TRUE, the rung is enabled.
-[=]-	EQUAL TO	XXX -[=]- YYY	The EQUAL TO instruction should be preceded by a GET instruction. 3-digit BCD values at the GET and EQUAL TO word addresses are compared. If equal, the rung is enabled.

TABLE 5-1 — Data Manipulation Instructions (cont.)

KEYTOP SYMBOL	INSTRUCTION NAME	DISPLAY	DESCRIPTION
-[B]-	GET BYTE	XXXD -[B]- YYY	D - Designates the upper or lower byte of the word. 1 = upper byte, 0 = lower byte. YYY - Octal value from 000_8 to 377_8 is stored in the upper or lower byte of the word address. The GET BYTE instruction should be followed by a LIMIT TEST instruction. A duplicate of the designated byte is made and compared with the upper and lower limits of the LIMIT TEST instruction.
-[L]-	LIMIT TEST	XXX AAA -[L]— BBB	AAA - Upper limit of LIMIT TEST, an octal value from 000_8 to 377_8 BBB - Lower limit of LIMIT TEST, an octal value from 000_8 to 377_8. The LIMIT TEST instruction should be preceded by a GET BYTE instruction. Compares the value at the GET BYTE instruction with the values at the LIMIT TEST instruction. If found to be between or equal to the limits, the rung is enabled.

Section 6
ARITHMETIC INSTRUCTIONS

6.0 GENERAL

The Mini-PLC-2 Processor can be programmed to perform arithmetic operations with two 3-digit BCD values using a set of Arithmetic instructions. These output instructions are:

- ADD -(+)-
- SUBTRACT -(–)-
- MULTIPLY -(x)-(x)- (1772-LN3 Processor)
- DIVIDE -(:)-(:)- (1772-LN3 Processor)

The two 3-digit BCD values to be computed are stored in two GET instruction words. The GET instructions, programmed in the condition area of the ladder diagram rung, should be followed by the Arithmetic instruction. Other condition instructions, if used, should be programmed before the GET instructions.

The Arithmetic instructions are programmed in the output position of the ladder diagram rung. They are assigned either one or two Data Table words to store the computed result, depending on the arithmetic operation performed. The ADD and SUBTRACT instructions use one Data Table word to store the result. The MULTIPLY and DIVIDE use two Data Table words to store the result.

The computed result is stored in BCD format in the lower 12 bits of the Arithmetic instruction word (Figure 6-1). Two of the remaining bits (bits 14 and 16) are used to indicate overflow and underflow conditions.

6.1 ADD INSTRUCTION

The ADD instruction (+) tells the Processor to add the two values stored in the GET words. The sum is then stored at the ADD instruction word address. When the sum exceeds 999, the overflow bit (bit 14) in the ADD instruction word is set ON (Figure 6-2). In the RUN or TEST mode, the overflow condition is displayed on the Industrial Terminal screen as a "1" preceding the sum.

NOTE: If an overflowed value (4 digits) is used for subsequent comparisons or other arithmetic operations, inaccurate results could occur. The Processor performs arithmetic and data manipulation operations with 3-digit BCD values, only.

6.2 SUBTRACT INSTRUCTION

The SUBTRACT instruction (–) tells the Processor to subtract the second GET word value from the first GET word value (Figure 6-3).

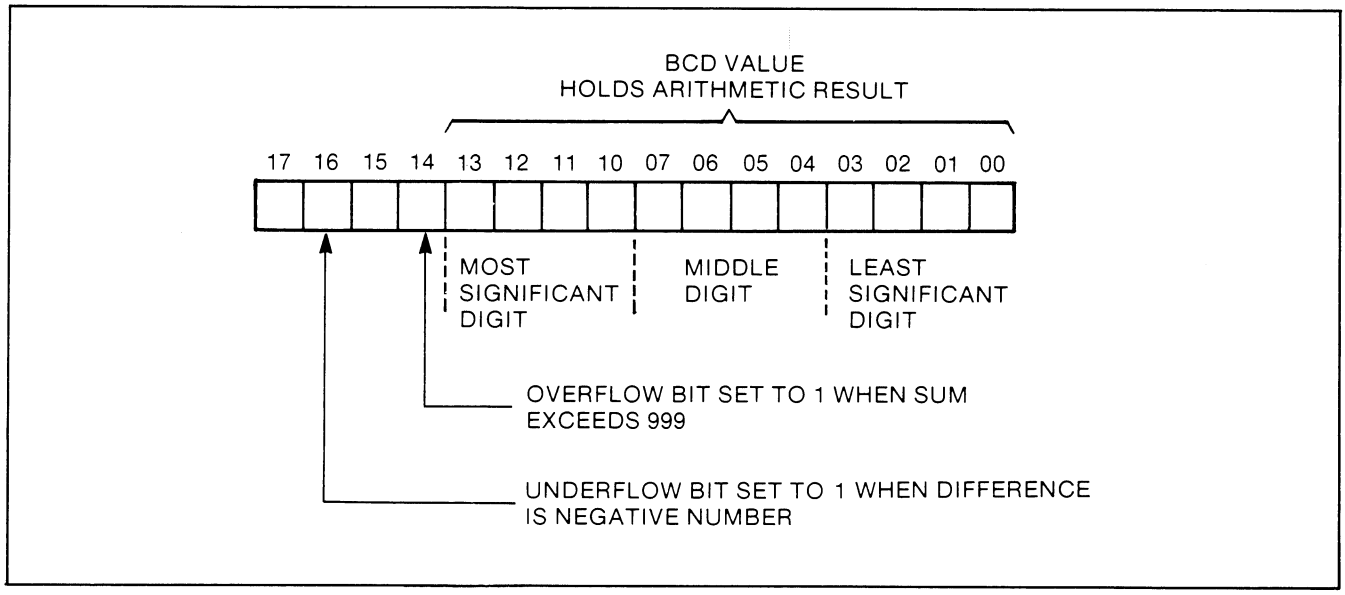


FIGURE 6-1 — Arithmetic Instruction Word

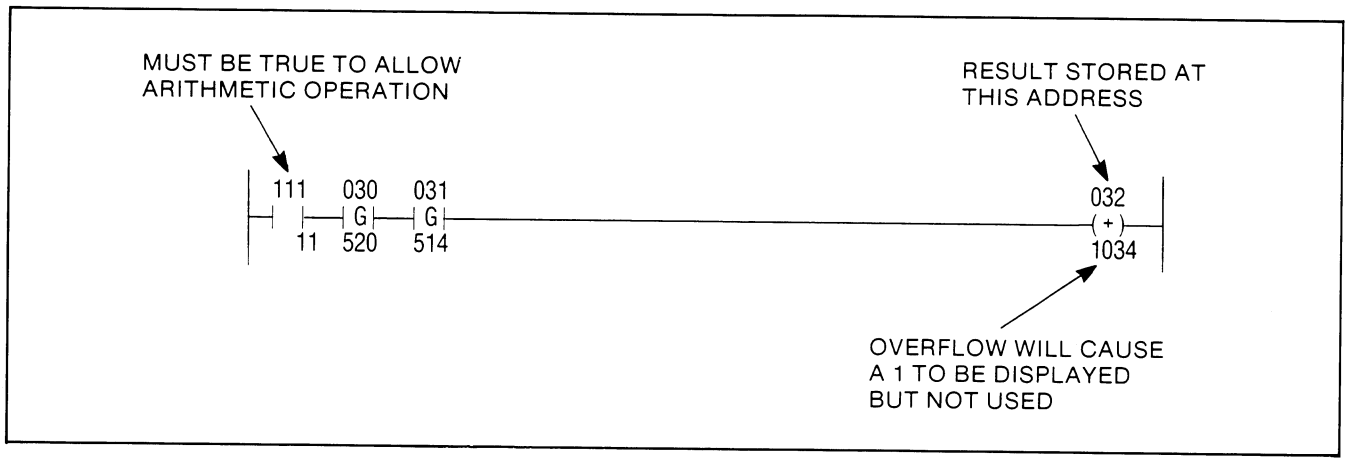


FIGURE 6-2 — ADD Instruction

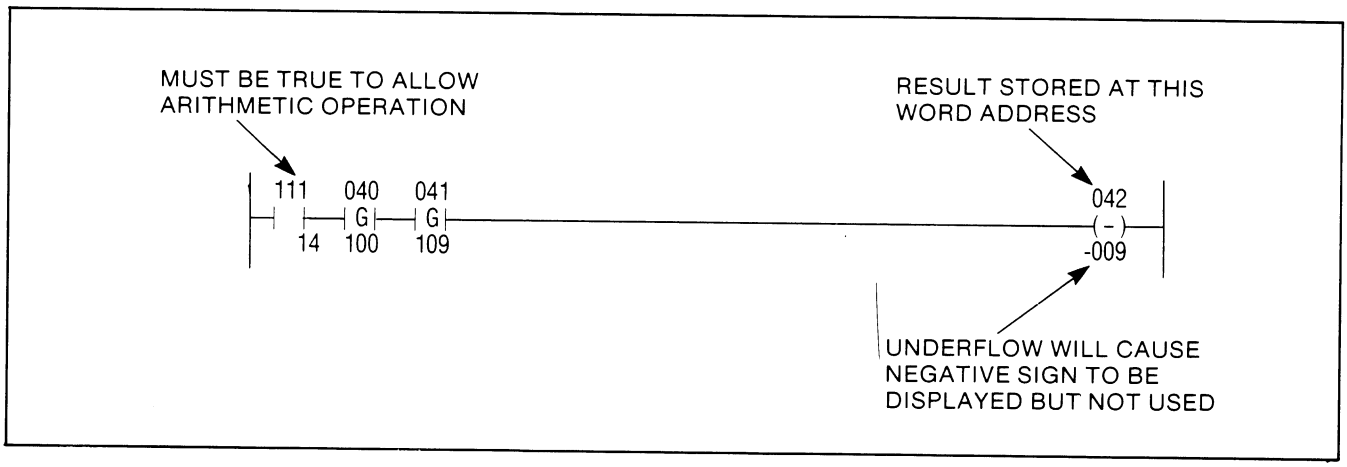


FIGURE 6-3 — SUBTRACT Instruction

The difference is then stored at the Data Table word addressed by the SUBTRACT instruction.

If the difference is a negative number, the underflow bit of the SUBTRACT word (bit 16) is set ON. In the RUN or TEST mode, the negative sign will appear on the Industrial Terminal screen preceding the difference.

NOTE: If a negative BCD value is used for subsequent operations, inaccurate results could occur. The Processor only compares, transfers and computes the absolute BCD value.

6.3 MULTIPLY INSTRUCTION (1772-LN3 Processor Module)

The MULTIPLY (X) instruction tells the Processor to multiply the two BCD values stored at the GET instruction words. The result is then stored in two Data Table words addressed by the MULTIPLY instruction (Figure 6-4).

For ease of programming, two consecutive Data Table words should be chosen to store the product. If the product is less than 6 digits, leading zeros will appear in the product.

6.4 DIVIDE INSTRUCTION (1772-LN3 Processor Module)

The DIVIDE instruction (:) tells the Processor to divide the first GET instruction value by the second GET instruction value. The result is stored in two Data Table words addressed by the DIVIDE instruction (Figure 6-5). Usually two consecutive Data Table words are chosen to store the quotient for ease of programming.

The quotient is rounded off and expressed as a decimal number. The decimal point is auto-

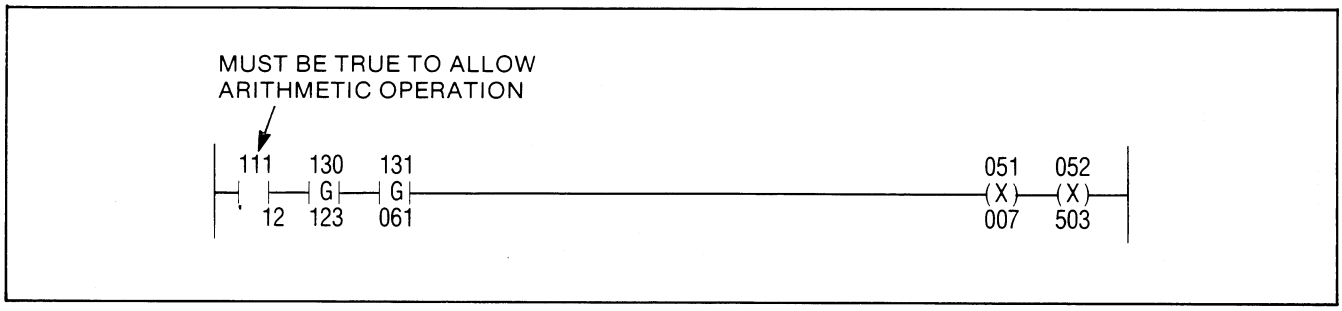


FIGURE 6-4 — MULTIPLY Instruction

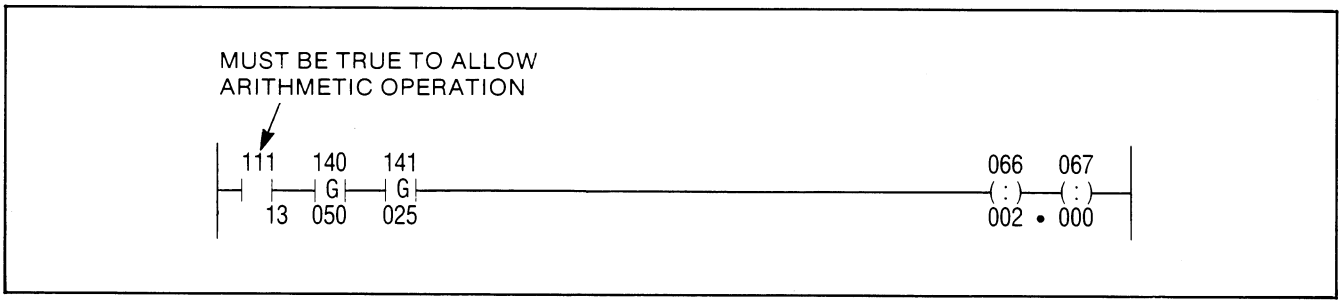


FIGURE 6-5 — DIVIDE Instruction

matically inserted between the two DIVIDE instruction values by the Industrial Terminal. Leading and trailing zeros in the quotient are also entered automatically by the Industrial Terminal.

Although division by 0 is undefined mathematically, the division of a number including zero by 0 will give the result of 999.999. (This differs from the PLC-2/20 and PLC-2/30 where 0÷0 = 1.000.)

6.5 PROGRAMMING ARITHMETIC INSTRUCTIONS

Arithmetic instructions are programmed from the Industrial Terminal keyboard with the Mini-PLC-2 Processor in the PROGRAM mode. When entered, these instructions will be intensified and blinking. The default word address above the instuction will have a reverse-video cursor positioned at the first digit. The instruction will continue to blink until the word address is entered.

Refer to Table 6-1 for a summarized description of these instructions.

TABLE 6-1 — Arithmetic Instructions

Note: Arithmetic instructions operate on BCD values in the Data Table. The word address XXX is displayed above the instruction; the BCD value YYY which is the result of the arithmetic operation is displayed beneath it. The BCD value is stored in the lower 12 bits of the word address and can be any value from 000 to 999.

KEYTOP SYMBOL	INSTRUCTION NAME	DISPLAY	DESCRIPTION
-(+)-	ADD	XXX -(+)- YYY	The ADD instruction is an output instruction. It is always preceded by two GET instructions which store the BCD values to be added. When the sum exceeds 999, bit 14 is set to 1, and a 1 is displayed in front of the result YYY.
-(-)-	SUBTRACT	XXX -(-)- YYY	The SUBTRACT instruction is an output instruction. It is always preceded by two GET instructions. The value in the second GET address is subtracted from the value in the first. When the difference is negative, bit 16 is set to 1, and a minus sign is displayed in front of the result YYY.
-(X)-	MULTIPLY (1772-LN3 Processor)	XXX XXX -(X)—(X)- YYY YYY	The MULTIPLY instruction is an output instruction. It is always preceded by two GET instructions which store the values to be multiplied. Two word addresses are required to store the 6 digit product.
-(÷)-	DIVIDE (1772-LN3 Processor)	XXX XXX -(:)—(:)- YYY.YYY	The DIVIDE instruction is an output instruction. It is always preceded by two GET instructions. The value of the first is divided by the value of the second. Two word addresses are required to store the 6 digit quotient. Its decimal point is placed automatically by the Industrial Terminal.

Section 7
OUTPUT OVERRIDE AND I/O UPDATE INSTRUCTIONS

7.0 GENERAL

Programming instructions may be needed for certain applications requiring output overrides or I/O updates. They are:

- MASTER CONTROL RESEST Instruction
- ZONE CONTROL LAST STATE Instruction
- IMMEDIATE INPUT Instruction
- IMMEDIATE OUTPUT Instruction

7.1 OUTPUT OVERRIDE INSTRUCTIONS

The two output instructions that can be used to override a group of outputs are:

- MASTER CONTROL RESET -(MCR)-
- ZONE CONTROL LAST STATE -(ZCL)-

These instructions are similar to a hardwired Master Control Relay in that they can affect a group of outputs in the User Program. The MCR and ZCL instructions, however, are NOT a substitute for a hard-wired relay, which provides emergency stop capabilities for all I/O devices.

WARNING: A PC system should not be operated without a hard-wired Master Control Relay and Emergency Stop switches to provide emergency I/O power shutdown. Emergency Stop switches can be monitored but should not be controlled by the User Program. These devices should be wired as described in the Mini-PLC-2 Assembly and Installation Manual (Publication 1772-820). The purpose of these devices is to guard against damage to equipment and/or injury to personnel.

The MCR and ZCL instructions control the zoned outputs differently:

- MCR — When FALSE, all nonretentive outputs within the MCR zone are de-energized or turned OFF.
- ZCL — When FALSE, the outputs within the ZCL zone are held in their last state: either ON or OFF.

To override a group of output devices, two MCR or ZCL instructions are required: one to begin the zone and one to end the zone (Figure 7-1). The Start Fence begins the zone and is always programmed with a set of input conditions. The End Fence ends the zone and must be programmed unconditionally.

When the MCR or ZCL Start Fence is TRUE, all outputs within the zone are controlled by their respective rung conditions. When the MCR or ZCL Start Fence is FALSE, the outputs within the zone are controlled by the MCR or ZCL Start Fence as stated above.

WARNING: MCR or ZCL zones must not be overlapped or nested. Each zone must be separate and complete. Common outputs must not be shared between MCR zones. Overlapping MCR or ZCL zones could result in unpredictable machine operation with possible damage to equipment and/or injury to personnel.

Sharing common outputs in more than one ZCL zone is permitted, provided that only one ZCL zone is enabled at a time. Common outputs can be examined in more than one MCR or ZCL zone.

7.2 I/O UPDATE INSTRUCTIONS

Two instructions used to accelerate the update of I/O data during the execution of the User Program are:

- IMMEDIATE INPUT -[I]-
- IMMEDIATE OUTPUT -(IOT)-

These instructions are used to transfer critical I/O data ahead of the normal scan sequence. The status of inputs is made immediately available to User Program and output decisions are accelerated to the output devices.

The IMMEDIATE I/O instructions are usually used where I/O modules interface with I/O devices that operate in a shorter period than the Processor scan time. These may include TTL-logic or fast response input or output devices.

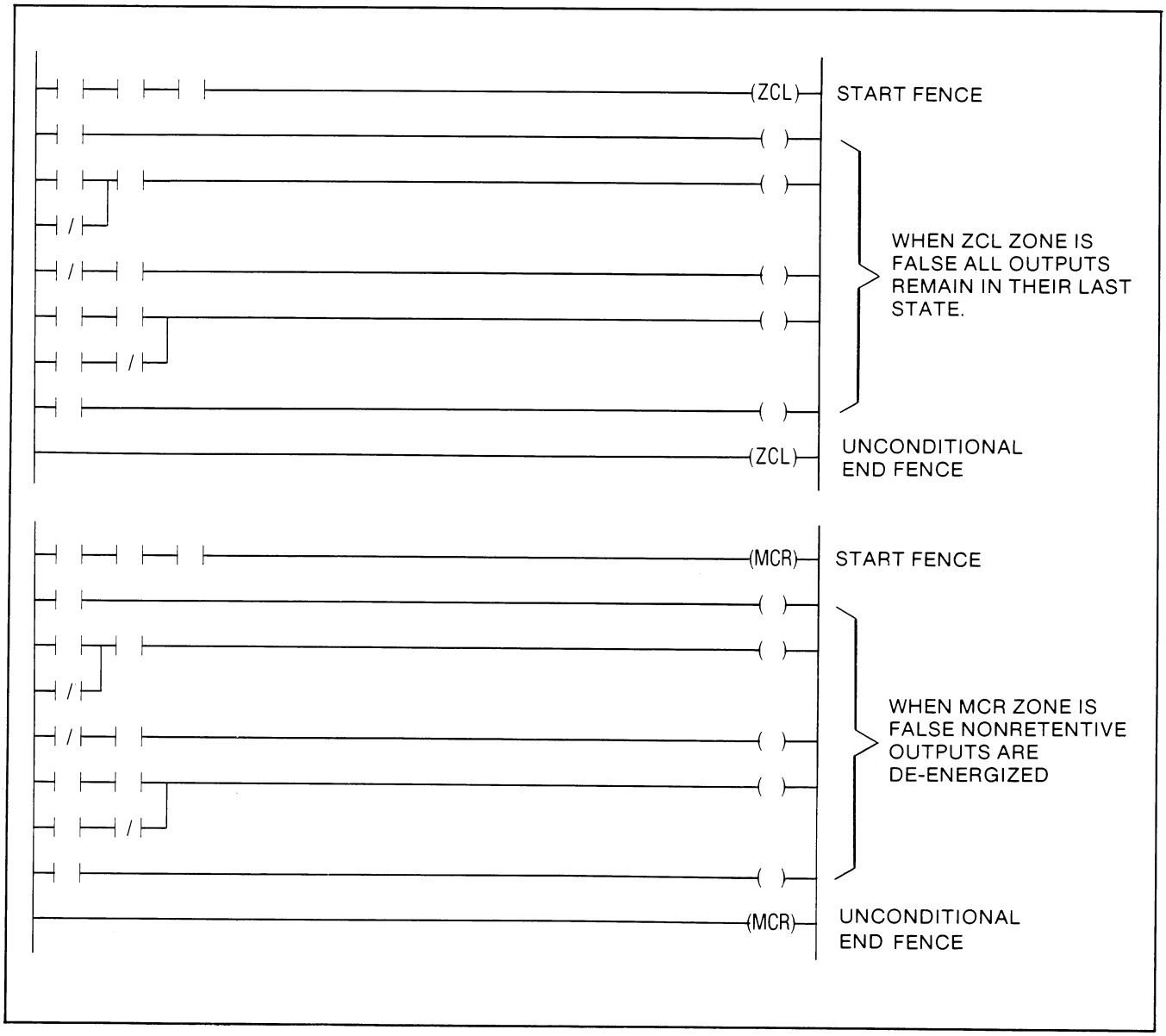


FIGURE 7-1 — MCR and ZCL Zone Programming

Most electromechanical devices have a response time longer than the Processor scan time. Thus, data to and from these devices need not be updated ahead of the normal I/O scan.

7.2.1 Scan Sequence

The Mini-PLC-2 Processor scan sequence can be divided into 2 parts (Figure 7-2):

- I/O Scan
- Program Scan

Upon power up, the Processor begins the scan sequence with the I/O scan. During the I/O scan, data from input modules is transferred to the Input Image Table. Data from the Ouput Image Table is transferred to the output modules.

After completing the I/O scan, the Processor begins the program scan. Here, all User Program instructions are generally scanned and executed in the order they were entered.

The I/O scan and program scan are performed one after the other. The time required to complete both scans is typically 23 msec for 900 instructions.

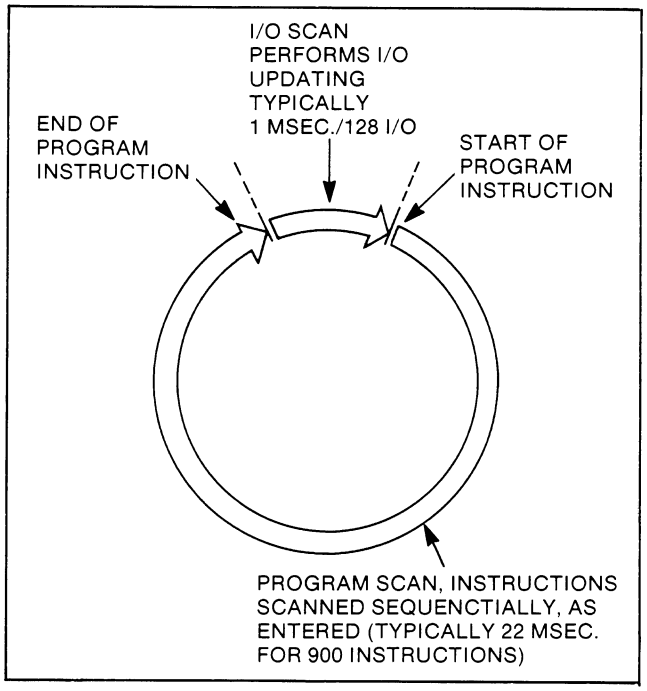


FIGURE 7-2 — Scan Sequence

Typically 23 msec may pass before I/O data is updated in a 1K system. The purpose of IMMEDIATE I/O instructions is to interrupt the program scan to update a word of critical input data or to transfer a word of critical data from the Output Image Table to the module in advance of the normal update sequence.

7.2.2 IMMEDIATE INPUT Instruction

The IMMEDIATE INPUT instruction [I] updates one word of the Input Image Table data in advance of the normal scan sequence (Figure 7-3). The Image Table word represents one Module Group in the I/O Chassis.

The IMMEDIATE INPUT instruction is programmed in the condition area of the ladder-diagram rung. The IMMEDIATE INPUT instruction can be considered as always TRUE; it is always executed whether or not other rung conditions allow logic continuity.

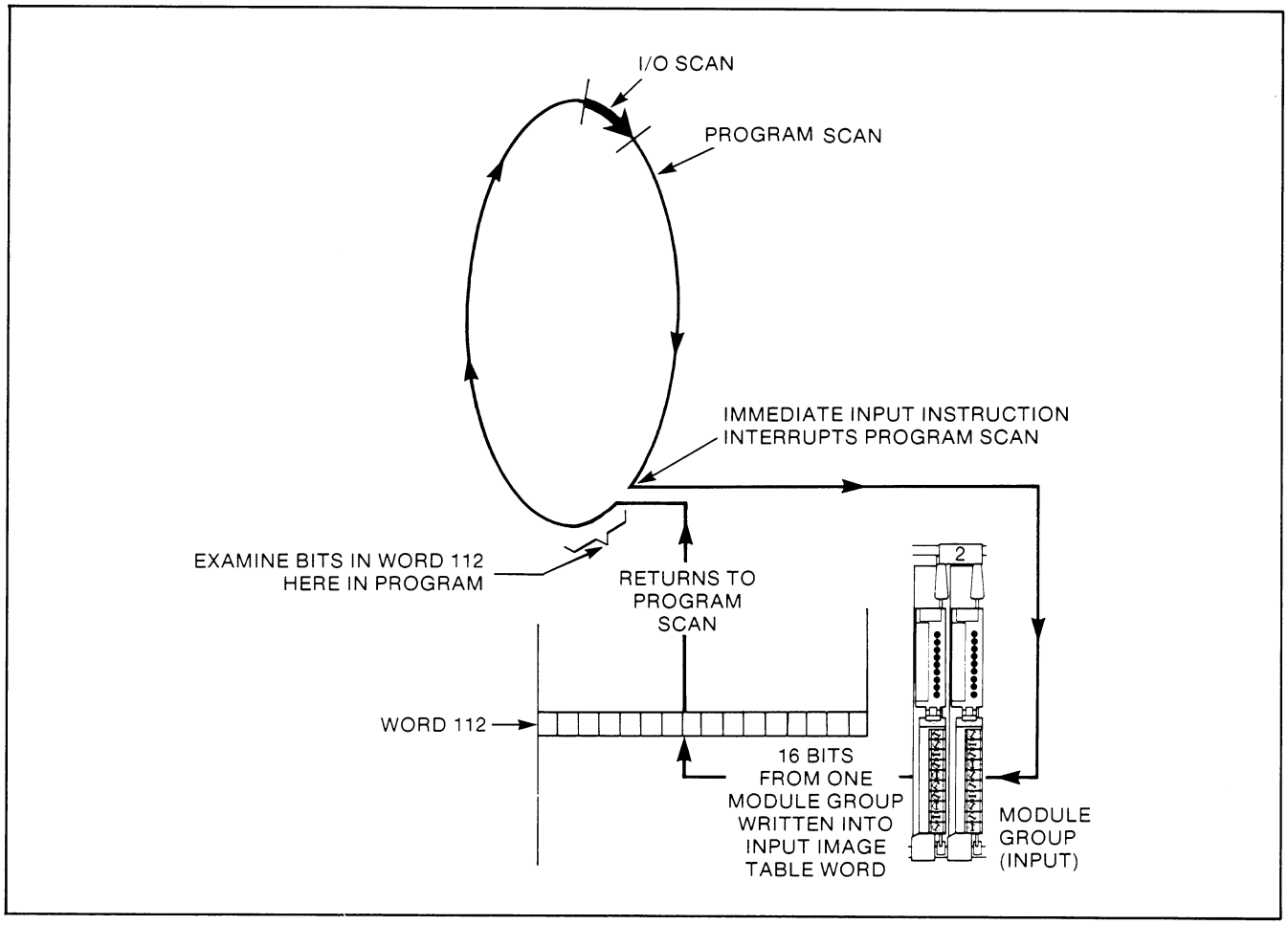


FIGURE 7-3 — IMMEDIATE INPUT Instruction

Program the IMMEDIATE INPUT instruction only when necessary. The need depends on both the response time of the input devices and modules being used, and on the position in the program of the rungs examining these inputs. It is best to program the IMMEDIATE INPUT instruction just before input instructions addressed to the applicable Module Group are examined.

7.2.3 IMMEDIATE OUTPUT Instruction

The IMMEDIATE OUTPUT instruction (IOT) updates one Module Group with data from one Output Image Table word ahead of the normal scan sequence (Figure 7-4).

The IMMEDIATE OUTPUT instruction is programmed as an output instruction in the ladder-diagram rung. This instruction is executed when rung conditions allow logic continuity. Unconditional programming can also be used to cause the Module Group to be updated during each program scan.

Program the IMMEDIATE OUTPUT instruction only when necessary. This depends on the response time of output modules and devices, and on the position of the rungs addressing the applicable Module Group.

The IMMEDIATE OUTPUT instruction should be programmed just after the rungs that con-

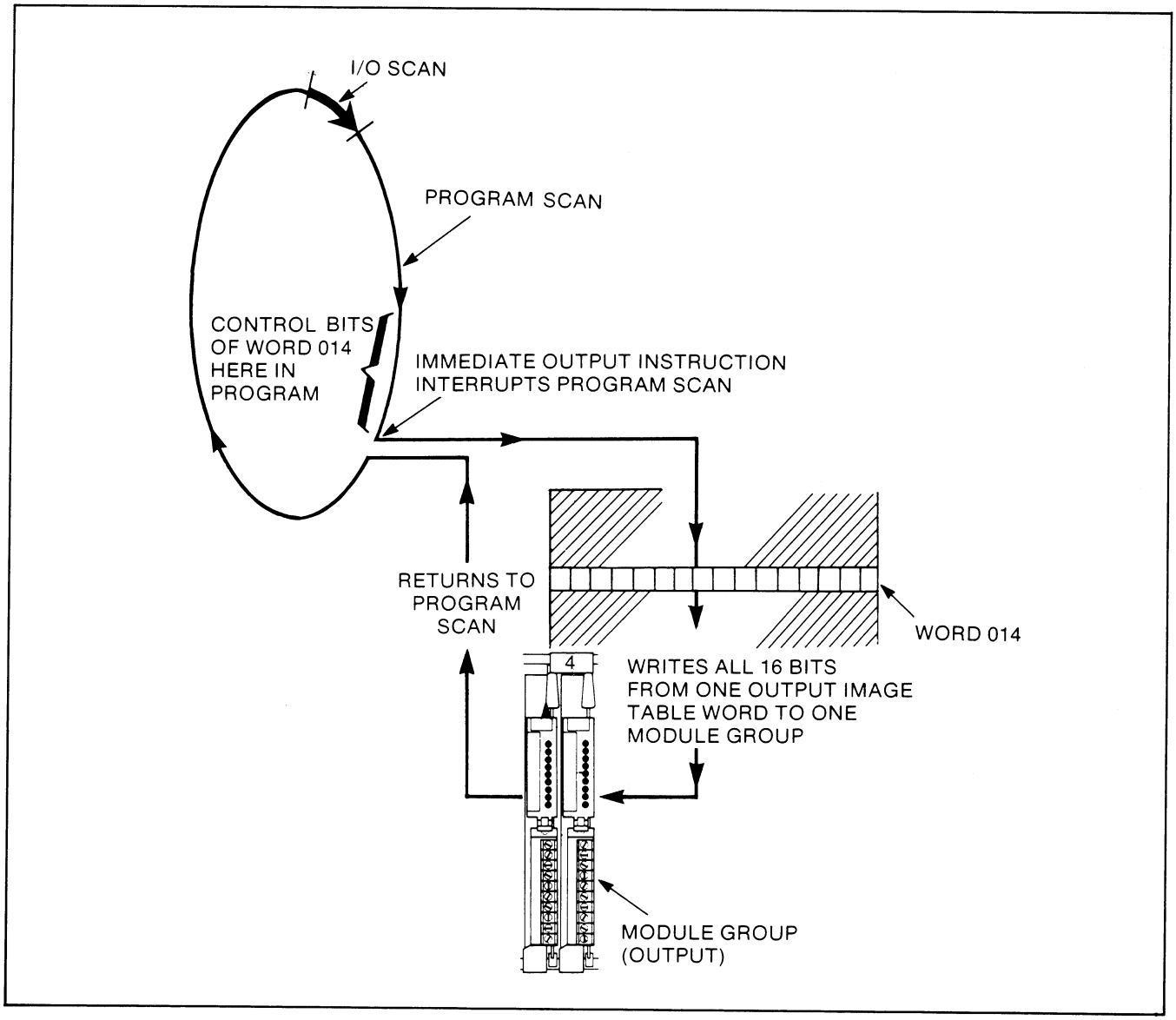


FIGURE 7-4 — IMMEDIATE OUTPUT Instruction

trol the bits in the addressed Output Image Table word.

In PC applications, this instruction only gives a slight advantage when entered near the end of the program scan; since the output data will soon be updated in the I/O scan. This instruction is best applied when entered near the middle of the User Program.

7.3 PROGRAMMING OUTPUT OVERRIDE AND I/O UPDATE INSTRUCTIONS

Instructions are programmed from the Industrial Terminal keyboard with the Processor in PROGRAM mode. When entered, they will be displayed as intensified and blinking with the reverse-video cursor positioned on the first digit of the default word address. The instruction will continue to blink until the word address is entered.

Refer to Table 7-1 for a summarized description of these instructions.

TABLE 7-1 — Output Override and I/O Update Instructions

Note: The MCR and ZCL boundary instructions have no word address.

The word addresses XXX of the IMMEDIATE INPUT and OUTPUT instructions are limited to the Input and Output Image Tables respectively.

KEYTOP SYMBOL	INSTRUCTION NAME	DISPLAY	DESCRIPTION
-(MCR)-	MASTER CONTROL RESET	-(MCR)-	Two MCR instructions are required to control a group of outputs. The first MCR instruction is programmed with input conditions to begin the zone. The second MCR instruction is programmed unconditionally to end the zone. When the first MCR rung is FALSE, all outputs within the zone, except those forced ON, latched ON, or any other retentive output will be de-energized. Do not overlap MCR zones, or nest with ZCL zones. Do not share common outputs between MCR zones.
-(ZCL)-	ZONE CONTROL LAST STATE	-(ZCL)-	Two ZCL instructions are required to control a group of outputs. The first ZCL instruction is programmed with input conditions to begin the zone. The second ZCL instruction is programmed unconditionally to end the zone. When the first ZCL rung is FALSE, outputs in the zone will remain in their last state. Do not overlap ZCL zones, or nest with MCR zones.
-[I]-	IMMEDIATE INPUT	XXX -[I]-	Processor interrupts program scan to update Input Image Table with data from the corresponding module group. It is updated before the normal I/O scan and executed each program scan.
-(IOT)-	IMMEDIATE OUTPUT	XXX -(IOT)-	When rung is TRUE, Processor interrupts program scan to update module group with data from corresponding Output Image Table word address. It is updated before the normal I/O scan and executed each program scan when the rung is TRUE. Can be programmed unconditionally.

Chapter 8
WRITING THE USER PROGRAM

8.0 GENERAL

The basic tools for writing the User Program include the programming instructions and an understanding of Processor operation as described in Section 1.4, Hardware/Program interface. Although approaches to and methods of writing programs that control machine operation vary, there are some guidelines that should be followed.

8.1 DEVELOPING THE PROGRAM

The first step in developing the User Program is to establish an operating sequence for input and output devices. The sequence must be evaluated to determine what the devices must do, what the conditions must be and the order in which they must operate.

After evaluating the operating sequence, the action of the different devices should be described in proper sequence with proper conditions for energizing each output device. This description is then used to develop the ladder diagram program. If a process diagram exists, it can be used as an aid in developing a ladder diagram program.

8.2 SAMPLE PROGRAM

The way a ladder diagram program is developed is best described by a simple example.

The application is one of separating good parts from bad parts. Figure 8-1 shows a part moving along a conveyor belt. Each part will trip a series of limit switches and will be sorted according to its size. The desired part size is 1.0" ±0.1".

If a part trips 2LS but not 3LS, the part is greater than or equal to 0.9" and less than or equal to 1.1". Because it is a good part, a storage bit (3SB) is latched ON. When the part trips 4LS, SOL1 is energized which moves the swingarm actuator, directing the part onto the good part conveyor.

If the part trips both or neither 2LS and 3LS, the part is too large or too small. When either condition occurs, a storage bit (4SB), 4CR is latched ON. Although the part will trip 4LS, it will continue along and trip 5LS, which energizes S0L2. The swingarm actuator will direct the part into the bad part bin. Each time a part enters the bad part bin, a counter is incre-

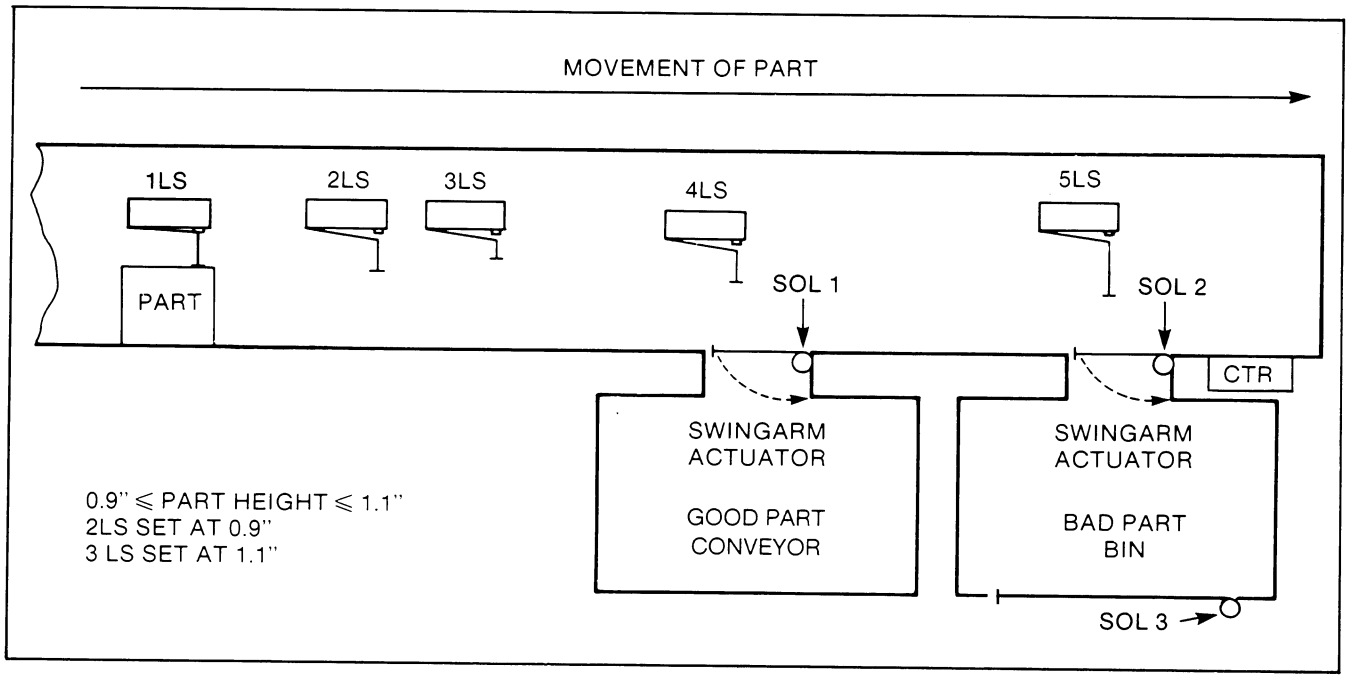


FIGURE 8-1 — Conveyor Belt Example

mented. When the bin is full (count complete), SOL3 is energized, which opens the bottom of the bin long enough to empty it. The counter will then reset automatically. Each time a new part enters the conveyor belt, 1LS is tripped which unlatches the storage bits and begins a new cycle.

The conveyor motor can be started or stopped with pushbutton START or STOP switches. Motor starter, MS1, controls the conveyor motor. A watchdog timer is used to monitor the folow of parts. If parts should become jammed causing a delay between 1LS and 4LS, the timer will time-out and turn OFF the conveyor motor. Another watchdog timer detects if a part becomes jammed beneath 4LS or 5LS. A conveyor RUN indicator and a parts JAM indicator allow remote observation of the conveyor operation. Additional documentation (not shown) would include a Power Distribution schematic showing a hardwired master control relay and emergency stop switches.

The logic can be written as a PC ladder diagram program (Figure 8-2). Data Table addresses are assigned to the hardwired devices. (Table 8-1). The ladder diagram should be developed by analyzing the logic required to operate the machine. A rung by rung description of the logic follows.

Rung 1 — This rung provides 3-wire control of the conveyor motor with jam detection for automatic shut down.

Rung 2 — The auxiliary contact of the motor starter is monitored to provide a conveyor RUN indication.

Rung 3, 4, 5, 6 — The part trips the first limit switch and unlatches storage bits 1 - 4 to begin a new cycle.

Rungs 7 — The first limit switch enables a Retentive Timer which is latched by the timer Enable bit. A jam condition is detected if the timer times out.

Rung 8 — Limit switch 4 (or the START pushbutton) resets the timer. If reset prior to 5 seconds, no jam has occurred between 1LS and 4LS. A jam beneath 4LS or to the right of it is not detected by this rung.

Rung 9 — A part passing 2LS latches SB1 if the height $\geq$ 0.9 inch. SB1 remains unlatched if the height $<$ 0.9 inch.

Rung 10 — A part passing 3LS latches SB2 if the height $>$ 1.1 inch. SB2 remains unlatched if the height $<$ 1.1 inch.

Rung 11 — A part within tolerance latches SB3.

Rung 12 — A part out of tolerance latches SB4.

Rung 13 — A good part at 4LS actuates SOL1 with swingarm actuator to direct the part to the good part conveyor.

Rung 14 — A bad part at 5LS actuates SOL2 with swingarm actuator to direct the part to the bad part bin.

Rung 15 — SOL2 increments the Up-counter, one count for each bad part.

TABLE 8-1 — Data Table Addresses for Hardwired Devices

Input Device	Address
STOP Pushbutton	112/00
START Pushbutton	112/01
Motor Starter Auxiliary	112/02
Limit Switch (1LS)	112/03
Limit Switch (2LS)	112/04
Limit Switch (3LS)	112/05
Limit Switch (4LS)	112/06
Limit Switch (5LS)	112/07
Output Device	
Motor Starter (MS1)	014/00
Conveyor RUN Indicator	014/01
Good Part Solenoid (SOL1)	014/02
Bad Part Solenoid (SOL2)	014/03
Bin Dump Solenoid (SOL3)	014/04
JAM Detect Indicator	014/05
Internal Functions	
Storage Bit 1 (SB1)	012/01
Storage Bit 2 (SB2)	012/02
Storage Bit 3 (SB3)	012/03
Storage Bit 4 (SB4)	012/04
Storage Bit 5 (SB5)	012/05
Retentive Timer, Watchdog	050
Timer, Bin Dump	051
Timer, Watchdog	052
Counter	060

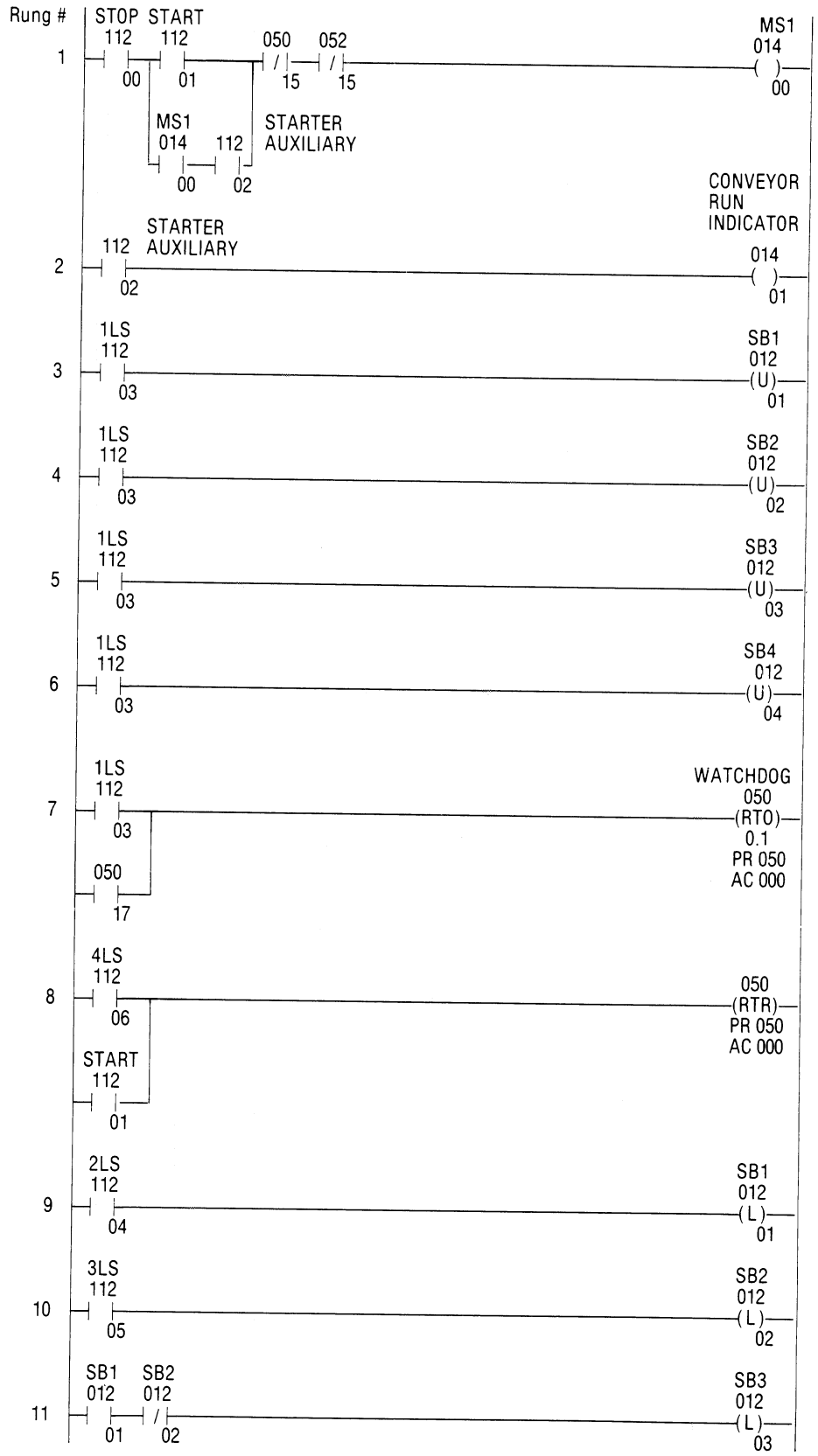


FIGURE 8-2 — Ladder Diagram Progam

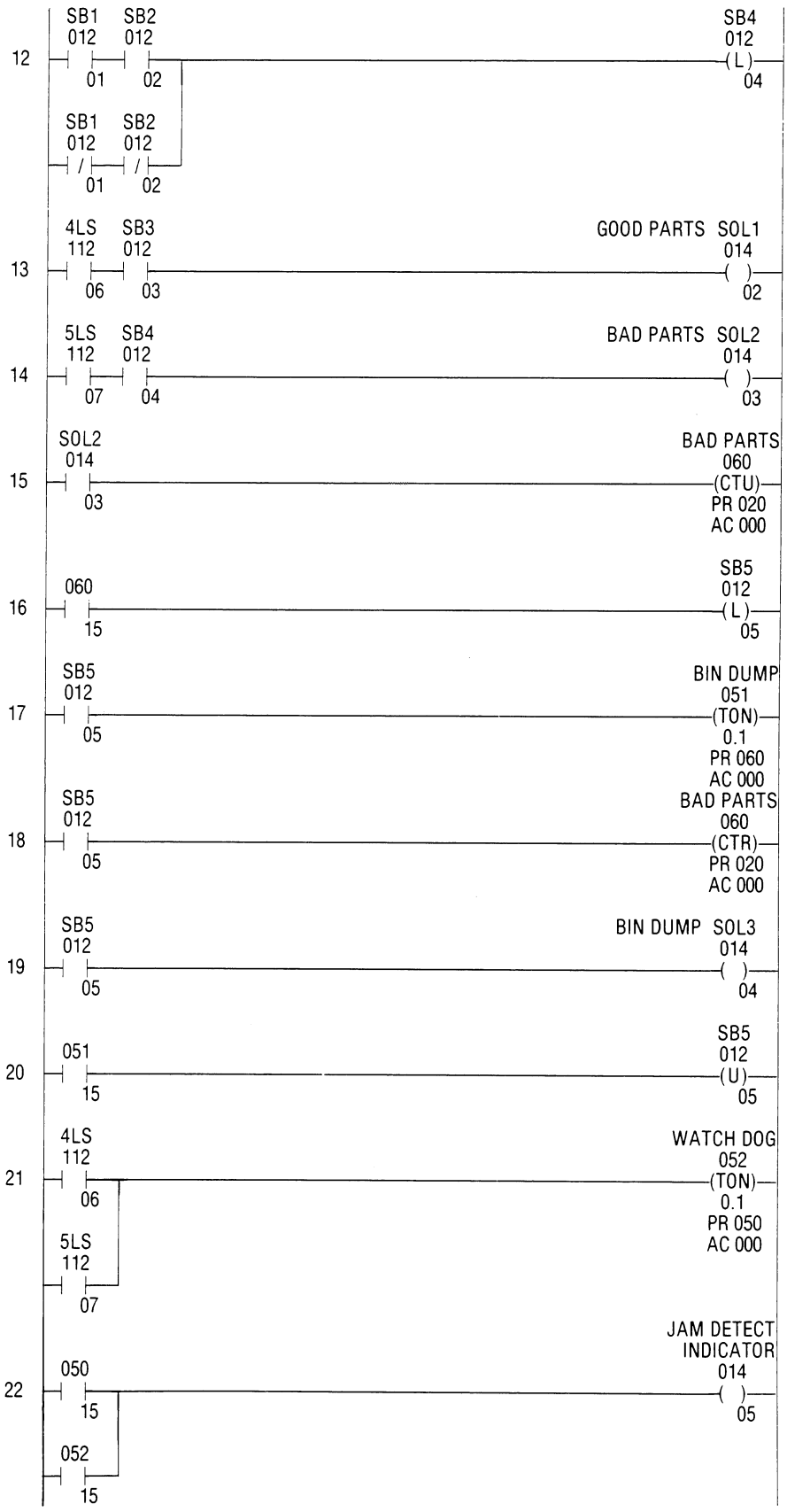


FIGURE 8-2 — Ladder Diagram Program (cont.)

Rung 16 — When AC = PR = 20, the Count Complete bit latches SB5.

Rung 17 — SB5 starts a timer to maintain a 6 second bin dump.

Rung 18 — SB5 resets the bad part counter.

Rung 19 — SB5 actuates SOL3 to dump the bad part bin by gravity feed.

Rung 20 — The Timed-out bit of timer 051_8 unlatches SB5 which in turn resets the timer.

Rung 21 — 4LS or 5LS enables the watchdog times 052_8. If 4LS or 5LS is held closed by a jam, this timer will time out.

Rung 22 — Timed-out bits are monitored to provide a JAM indication to the operator.

8.3 DEVELOPING THE DATA TABLE

The factory configured Data Table shown in Section 1, Figure 1-5, should be used as a guide when developing the Data Table. Determining the number of words needed and assigning addresses is a procedure that requires care and attention to detail.

The Data Table should be roughed-out in advance but formally developed as the User Program is being written. Each Data Table word and bit address and its function should be logged as it is assigned.

8.3.1 Data Table Documentation Forms

The Data Table documentation forms presented at the end of this section can be reproduced or revised as needed. They include two general types:

1) Data Table Map to describe the Data Table as a whole. Publication 5045.

2) Data Table Bit and Word Assignment Sheets to log and describe the function of assigned addresses. Publication 5046 and 5047.

An example showing how each form is used accompanies the description.

DATA TABLE MAP (128-WORD)

This form can be used to log the bit status of a word and to describe the function of groups of related words within a 128-word Data Table section.

The lower two digits of the 3-digit word addresses are pre-numbered in the left-hand column. The bit numbers, 00-17, complete the 5-digit bit address. The starting word address can be written once for the entire 64 word column.

For example, Figure 8-3 shows a completed portion of the Data Table Map. The left-hand column represents the addresses 200/00 through 277/17 because a "2" is written in the starting word address blank at the top of the column.

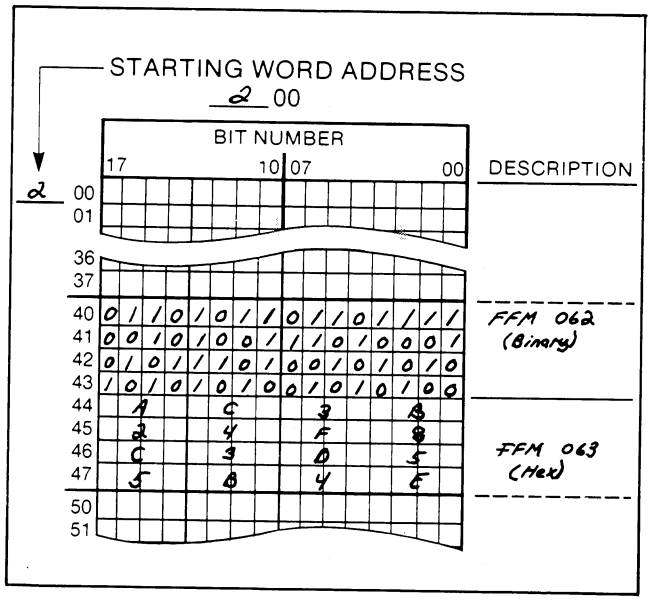


FIGURE 8-3 — Example of a Data Table Map

DATA TABLE WORD ASSIGNMENTS (64 WORD)

This form can be used to write functional descriptions of word addresses used in the Data Table for word storage, timers and counters, etc.

The form is divided into two 32-word columns. The words can be numbered consecutively through the entire 64 words. Or, the right-hand column can be numbered 100_8 greater than the left-hand column to conveniently track Accumulated and Preset values. In either case, the lowest digit of the 3-digit word address is pre-numbered, 0-7.

For example, a portion of the Data Table Word Assignment Sheet is shown in Figure 8-4. It

illustrates timer and counter functional descriptions for Accumulated values starting at word address 030_8 and Preset values starting at 130_8. An "03" and "13" were written into the left-hand and right-hand word address boxes, respectively.

DATA TABLE BIT ASSIGNMENTS

This form can be used to log the function of input, output and storage bits.

Similar to the Word Assignment Sheet, the Bit Assignment Sheet is divided into two 2-word columns. The words can be numbered consecutively. Or, the right-hand column can be numbered 100_8 greater than the left-hand column for the convenient logging of input, output and/or storage bits having the same module group number. The bit numbers are pre-numbered, 00-17.

For example, a portion of the Data Table Bit Assignment Sheet is shown in Figure 8-5. It illustrates logging the input devices associated Module Group 2 and the storage bits of the corresponding storage word 012_8 (complement of word 112_8). Word addresses "012" and "112" have been entered into corresponding word address boxes in the left- and right-hand columns, respectively. The 3-digit word address is entered once for all 16 bits.

8.3.2 I/O Assignment Considerations

Once the description of the application is complete, Data Table bit addresses can be assigned to the input and output devices wired to the Controller. The 5-digit bit address directly corresponds to the location of each I/O device with respect to the Rack number (always 1), Module Group and Terminal number. Bit addresses cannot be assigned arbitrarily to I/O devices because bit addresses are hardware-related. Review Section 1.4, Hardware/Program Interface, if necessary. Analog modules and other intelligent I/O modules use word addresses rather than 5-digit bit addresses. Refer to the module Users Manual for more information on addressing and wiring.

The installer and programmer should work closely together to determine the best placement of the I/O modules within the I/O chassis. To simplify installation and troubleshooting procedures, it may be desirable to group like modules together. Also, module locations should be assigned to minimize electrical noise radiation from AC lines. It is helpful to document I/O assignments on a form such as Publication 5039 found at the end of this section.

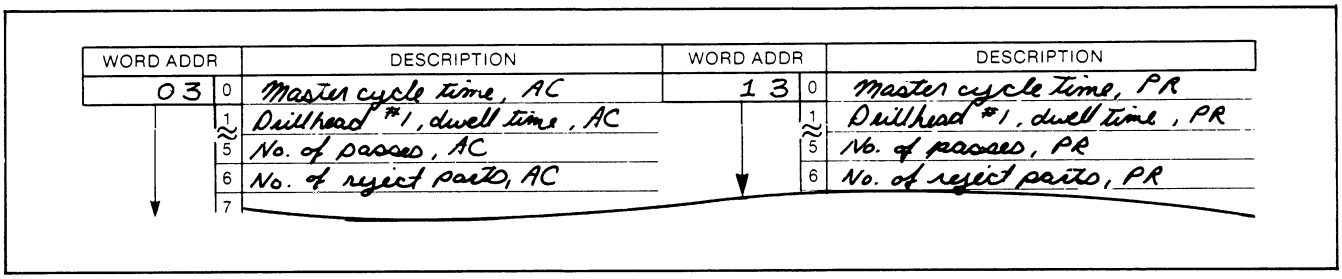


FIGURE 8-4 — Example of Data Table Word Assignments

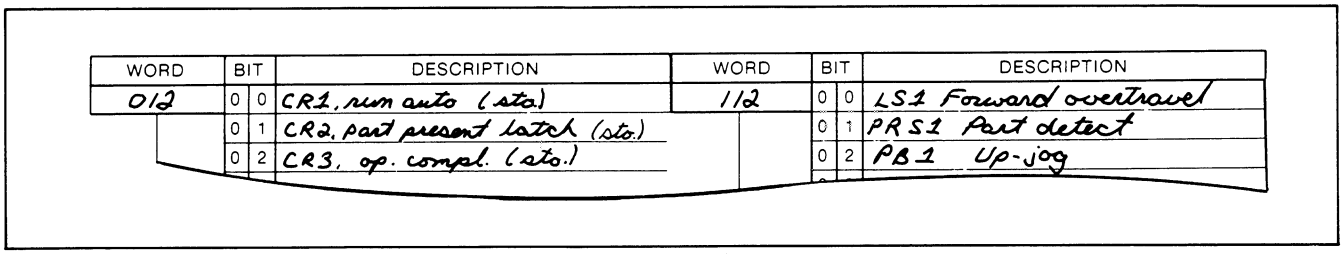


FIGURE 8-5 — Example of Data Table Bit Assignments

This form can be used as follows. The titles of the modules can be written in the spaces immediately below the Module Group labels. The spaces immediately to the left of the LED indicators and the screw terminations can be used to identify the I/O devices and to label the wire numbers connecting the devices to the terminals, respectively.

Recommendations for I/O wiring and module placement can be found in Publication 1772-820, the Mini-PLC-2 Assembly and Installation Manual.

8.3.3 Timer/Counter Assignment Considerations

Timers and counters require two Data Table word addresses, one for the Accumulated value, the other for the Preset value. The instruction address is the address where the Accumulated value is stored. Timer and counter instructions can be assigned Data Table addresses beginning at word address 030_8 through 077_8. The Preset value is located at the word address 100_8 greater than the Accumulated value word address. Timer and counter addresses and descriptions should be entered on Data Table Word Assignment Sheets.

If Block Transfer programming is used, the addresses of the pair of GET instructions must be entered into the timer/counter areas of the Data Table starting at word 030_8. These instructions also use two Data Table word addresses, one at an address 100_8 greater than the other. Each Block Transfer rung will decrease by one (1) the number of available equivalent timers and/or counters from the maximum quantity of 40. For more information on Block Transfer, see Section 11.2.

8.3.4 Bit/Word Storage Considerations

Bit/word storage addresses can be located in all areas of the Data Table excluding the Input Image Table and Processor Work areas. Bit and word storage addresses should be chosen carefully to conserve memory. The following recommendations for bit and word storage should be considered:

- Bits 14-17 of a timer or counter Preset word can be used for bit storage, provided data is NOT transferred to the Preset word by a GET/PUT transfer, or the time base of the timer is NOT .01 second.
- Unused Data Table words in the timer/counter areas can be used for bit/word storage. To conserve memory, use both the Accumulated and corresponding Preset words for storage.
- Output Image Table words can be used for storage when the corresponding Input Image Table words are used for input modules (Block Transfer modules excluded). However, when there is a vacant module group or slot in the I/O Chassis, do not use the corresponding I/O Image Table words for storage. Reserve these words for future system expansion.
- Unused Input Image Table words can not be used for storage. They are cleared to zero during each I/O scan.
- Word 027_8 should not be used for storage or control of output devices. Many of the bits are used by the Processor for control functions.

The number of bit/word storage addresses will depend on User Program requirements and is sometimes difficult to estimate in advance.

8.4 Sizing the Data Table

The Data Table is factory configured to 128 words. The Data Table can be reduced in size to 48 words by reducing the number of words available for timers, counters and equivalent word storage in the Accumulated and Preset value areas. If less than 40 equivalent timers/counters are assigned, the size of the Data Table should be reduced to allow additional memory to be used for User Program instructions. Up to 80 instructions can be added to User Program, two instructions for each equivalent timer/counter not used. See Figure 8-6.

After completing the User Program and logging all addresses on Data Table Assignment Sheets, the highest address assigned will determine the size of the Data Table.

The size of the Data Table also can be computed. The computation is made using the

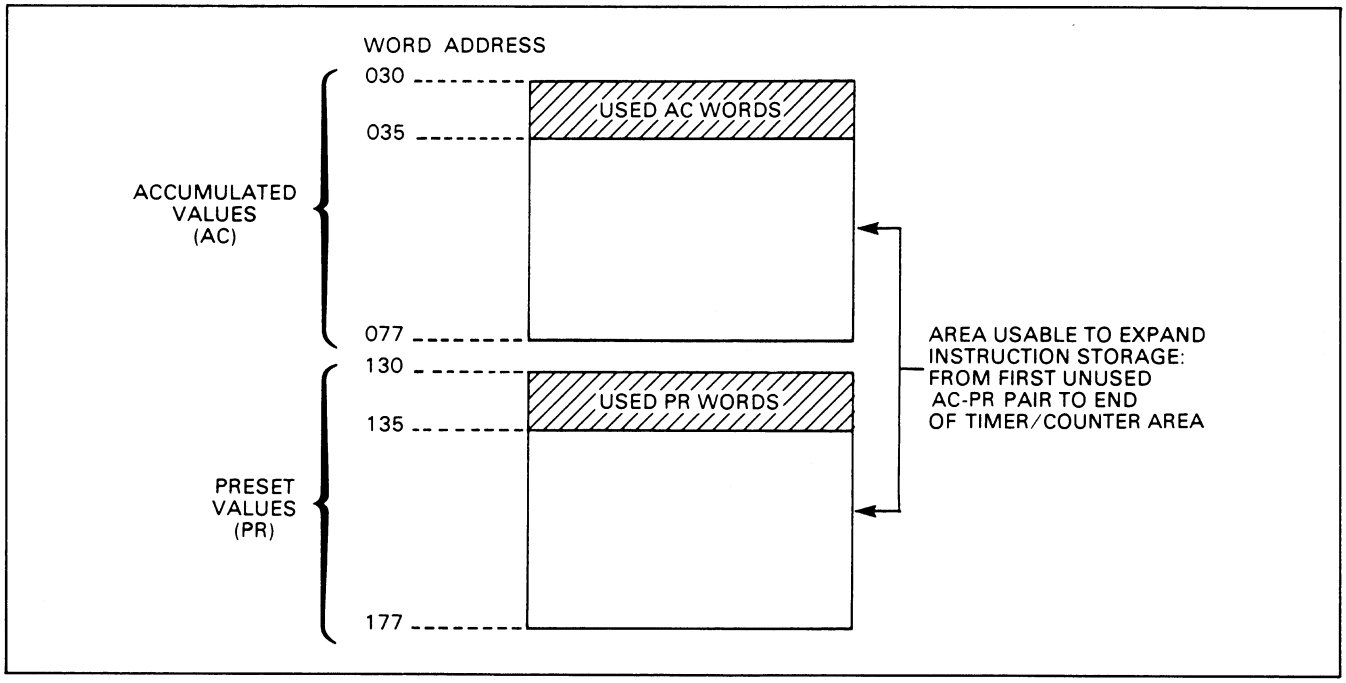


FIGURE 8-6 — Data Table Adjusted for Additional User Program

following formula:

$$ET = T + C + IS/2$$

where ET = # equivalent timers/counters, T = # timers, C = # counters and IS = # storage words. Any partially assigned or unassigned storage word from the highest assigned AC-PR pair to the beginning of the timer/counter area must be considered as an assigned storage word.

The number of equivalent timers/counters is used to adjust the Data Table in Processor memory before entering the User Program. The Data Table Adjustment procedure is described in Section 9.1.

8.5 PROGRAM RECOMMENDATIONS

The program recommendations listed below for constructing a ladder diagram rung should be considered:

Note: Special considerations are given for MULTIPLY and DIVIDE instructions. The rung size limitations exist because of the Industrial Terminal screen size.

- Only one Output instruction can be programmed in a rung.
- Generally, program only one rung to energize an output device to simplify troubleshooting and maximize safety.
- Up to 12 Condition instructions in series can be programmed in a rung; up to 11 if the output is a MULTIPLY or DIVIDE instruction.
- When the desired number of series Condition instructions exceeds the horizontal limit of the screen (Figure 8-7a), use a storage bit to make two rungs (Figure 8-7b).
- Up to 7 parallel branches can be programmed in a rung.

8.6 CURRENT RECORD

A hard-copy printout and/or a tape recording of Data Table, User Program and messages should be made after the machine operation is working as desired. Before reproducing Processor memory content, set all Data Table values to start-up conditions. Record a total memory dump that includes the start-up Data Table values.

The Data Table Assignment Sheets, hard-copy printout and/or the tape recording constitute the current record of the machine operation. If subsequent changes are required, all back-up records should be kept up to date.

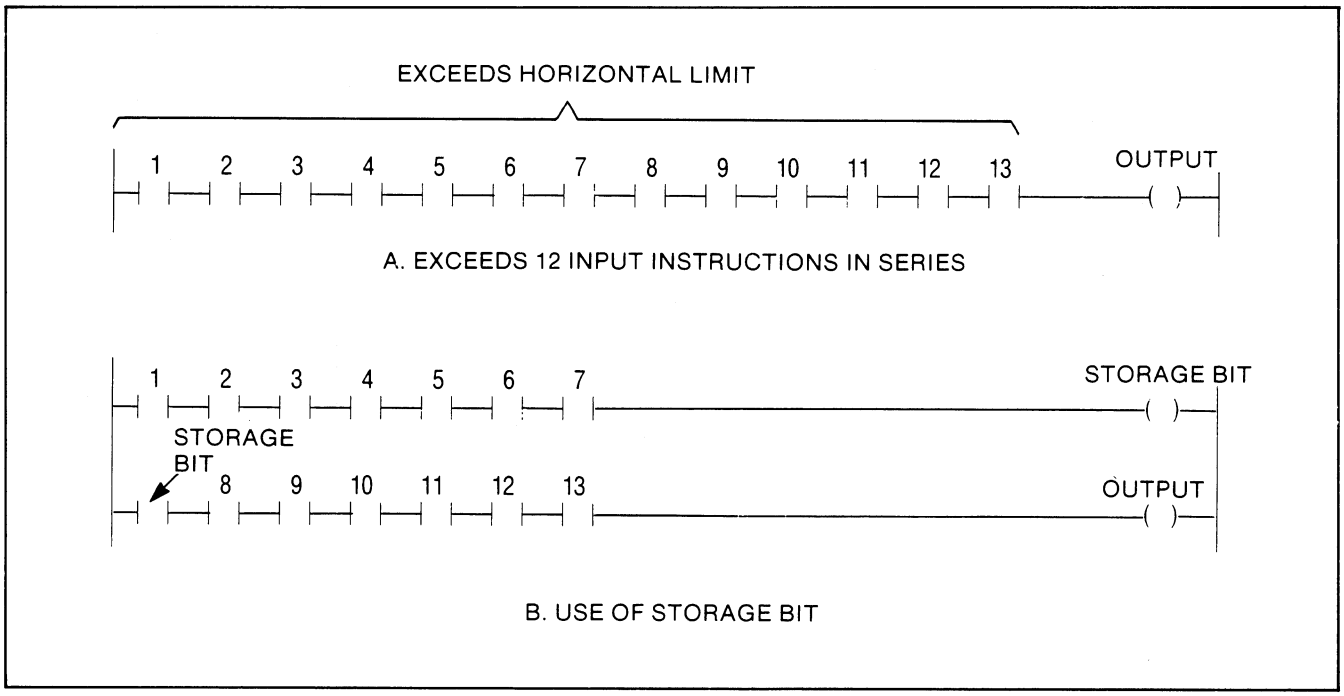


FIGURE 8-7 — Storage Bit Example

ALLEN-BRADLEY
Programmable Controller
DATA TABLE MAP
(128-WORD)
(Publication 5045 — February, 1982)

PAGE _____ OF ____
ADDRESS _____ TO ____

PROJECT NAME _____ PROCESSOR _____

DESIGNER _____ DATA TABLE SIZE _____

STARTING WORD ADDRESS
____00

	BIT NUMBER 17 10 07 00	DESCRIPTION
____ 00		
01		
02		
03		
04		
05		
06		
07		_____
10		
11		
12		
13		
14		
15		
16		
17		_____
20		
21		
22		
23		
24		
25		
26		
27		_____
30		
31		
32		
33		
34		
35		
36		
37		_____
40		
41		
42		
43		
44		
45		
46		
47		_____
50		
51		
52		
53		
54		
55		
56		
57		_____
60		
61		
62		
63		
64		
65		
66		
67		_____
70		
71		
72		
73		
74		
75		
76		
77		_____

STARTING WORD ADDRESS
____00

	BIT NUMBER 17 10 07 00	DESCRIPTION
____ 00		
01		
02		
03		
04		
05		
06		
07		_____
10		
11		
12		
13		
14		
15		
16		
17		_____
20		
21		
22		
23		
24		
25		
26		
27		_____
30		
31		
32		
33		
34		
35		
36		
37		_____
40		
41		
42		
43		
44		
45		
46		
47		_____
50		
51		
52		
53		
54		
55		
56		
57		_____
60		
61		
62		
63		
64		
65		
66		
67		_____
70		
71		
72		
73		
74		
75		
76		
77		_____

ALLEN-BRADLEY
Programmable Controller
DATA TABLE WORD ASSIGNMENTS
(64-WORD)
(Publication 5046 — February, 1982)

PAGE ______ OF ____
ADDRESS ______ TO ____

PROJECT NAME ____________________

DESIGNER ____________________

PROCESSOR ____________________

DATA TABLE SIZE ____________________

WORD ADDR	DESCRIPTION	WORD ADDR	DESCRIPTION
0		0	
1		1	
2		2	
3		3	
4		4	
5		5	
6		6	
7		7	
0		0	
1		1	
2		2	
3		3	
4		4	
5		5	
6		6	
7		7	
0		0	
1		1	
2		2	
3		3	
4		4	
5		5	
6		6	
7		7	
0		0	
1		1	
2		2	
3		3	
4		4	
5		5	
6		6	
7		7	

Comments ____________________

ALLEN-BRADLEY
Programmable Controller
DATA TABLE BIT ASSIGNMENTS
(Publication 5047 — February, 1982)

PAGE _____ OF ____
ADDRESS _____ TO ____

PROJECT NAME _____ PROCESSOR _____

DESIGNER _____ DATA TABLE SIZE _____

WORD	BIT	DESCRIPTION	WORD	BIT	DESCRIPTION
	0 0			0 0	
	0 1			0 1	
	0 2			0 2	
	0 3			0 3	
	0 4			0 4	
	0 5			0 5	
	0 6			0 6	
	0 7			0 7	
	1 0			1 0	
	1 1			1 1	
	1 2			1 2	
	1 3			1 3	
	1 4			1 4	
	1 5			1 5	
	1 6			1 6	
	1 7			1 7	
	0 0			0 0	
	0 1			0 1	
	0 2			0 2	
	0 3			0 3	
	0 4			0 4	
	0 5			0 5	
	0 6			0 6	
	0 7			0 7	
	1 0			1 0	
	1 1			1 1	
	1 2			1 2	
	1 3			1 3	
	1 4			1 4	
	1 5			1 5	
	1 6			1 6	
	1 7			1 7	

Comments _____

ALLEN-BRADLEY
Cleveland, Ohio 44143

Bulletin 1771 I/O Chassis

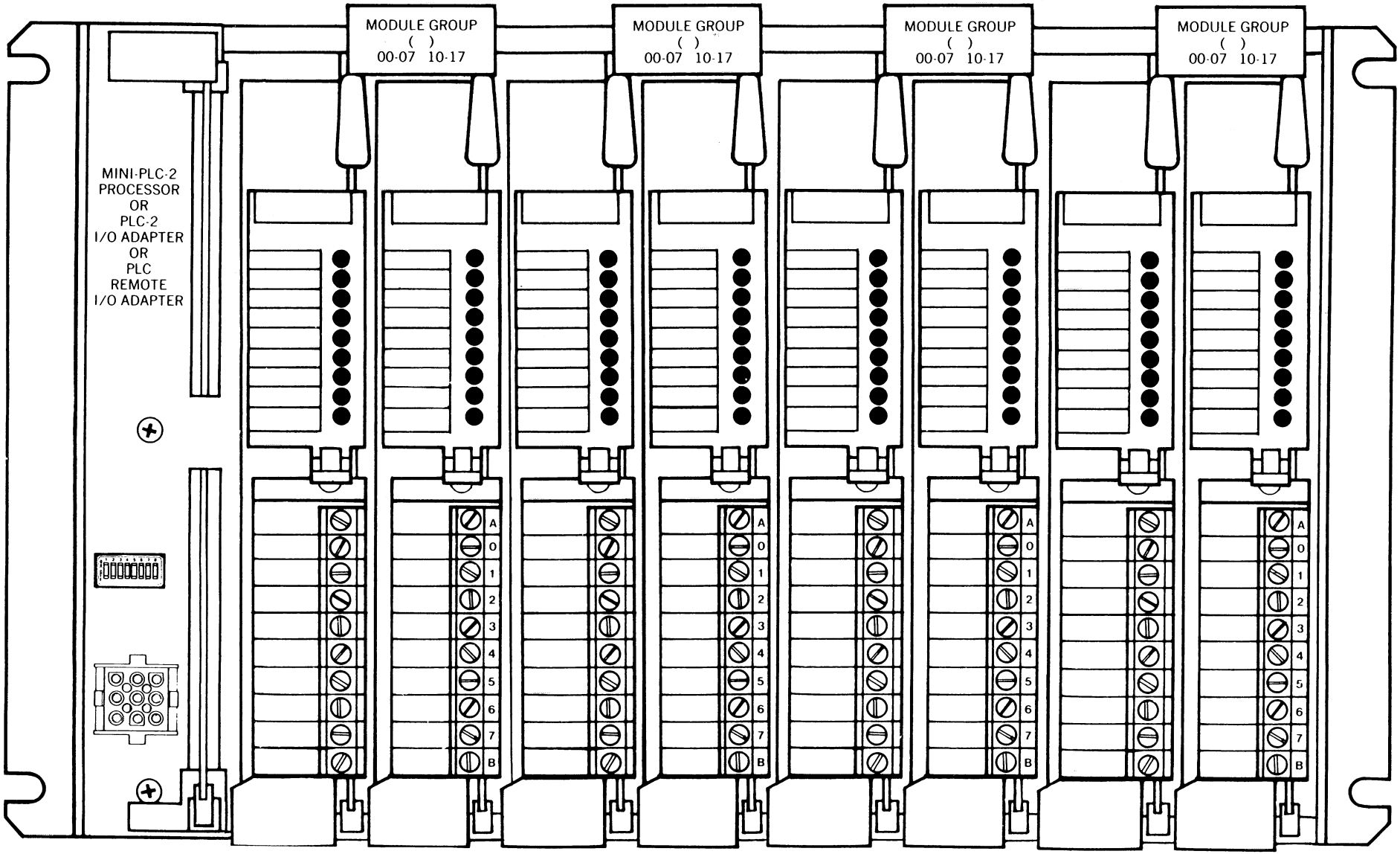
CONNECTION DIAGRAM ADDRESSING

(Publication 5039 — September, 1980)

PAGE _____ OF _____

DATE _____

PROJECT NAME _____ DESIGNER _____



Section 9
OPERATING INSTRUCTIONS

9.0 GENERAL

This section contains the operating instructions that are used to move through the program and perform a variety of functions. The instructions are grouped by function and are summarized in Tables 9-1 through 9-6 at the end of this section. They include:

- Data Table Adjustment (Table 9-1)
- Addressing
- Editing Functions (Table 9-2)
- Directories (Table 9-3)
- Search Functions (Table 9-4)
- Troubleshooting Aids (Table 9-5)
- Clear Memory Functions (Table 9-6)

9.1 DATA TABLE ADJUSTMENT

After the size of the Data Table has been determined as described in Section 8.4, the SEARCH 50 function is used to adjust the Data Table. The following display will appear when the [SEARCH] [5][0] keys are pressed in PROGRAM mode.

DATA TABLE ADJUSTMENT

Number of Input/Output Racks	2
Number of Timers/Counters	040
Data Table Size	128

NOTE: The default value of 2 I/O racks will be displayed. It is not a user-entered value for the Mini-PLC-2.

The Data Table is factory configured to 128 words for 1 I/O rack and 40 timers/counters. The Data Table can be reduced in 2-word increments to a minimum of 48 words if no equivalent timers/counters are selected.

The number of equivalent timers/counters to be entered is prompted by a reverse video cursor. When this number has been entered, the Industrial Terminal will compute and enter the Data Table size.

Anytime the Data Table is reduced in size, the Processor searches for instructions in those areas. If an instruction exists in an area to be deleted, the change will not be allowed and the following message will be displayed: "INSTRUCTION EXISTS IN DELETED AREA." To display the rung that is preventing the change, press [SEARCH]. At that time, the decision can be made whether to keep or delete the instruction.

Press [CANCEL COMMAND] to terminate the Data Table Adjustment display.

The instructions for adjusting the Data Table are summarized in Table 9-1.

9.1.1 Memory Layout Display (1770-T3 Industrial Terminal)

The SEARCH 54 function displays a diagram of the areas of memory including the Data Table, User Program, Message Area and the unused memory. The number of words in each area is indicated in decimal numbers.

Press [SEARCH][5][4] to initiate this display, and press [CANCEL COMMAND] to terminate this display.

9.2 ADDRESSING

The ladder diagram instructions are entered with the Processor in the PROGRAM mode. When entered, they are displayed as intensified and blinking to indicate cursor position and that information is needed.

When entering addresses and data, the reverse-video character cursor can be manipulated to the left and right using the [←] and [→] keys to make corrections. The character cursor cannot be moved to the left past the first digit. If the character cursor is moved off the instruction address to the right, the instruction will be entered. It will stop blinking but will remain intensified until the next instruction is pressed or the instruction cursor is moved to the right.

Any time a digit being entered is not within the proper limits, the message "DIGIT OUT OF RANGE" will be displayed. The cursor will

TABLE 9-1 — Data Table Adjustment

FUNCTION	MODE	INDUSTRIAL TERMINAL	KEY SEQUENCE	DESCRIPTION
Data Table Adjustment	PROGRAM	Any [1]	[SEARCH] [5][0] [Numbers]	Enter the number of equivalent timers/counters. The Industrial Terminal displays the size of the reduced Data Table.
Processor Memory Layout	Any	1770-T3	[SEARCH] [5][4]	Displays the number of words in the Data Table Area, User Program Area, Message Area and unused memory.
Either			[CANCEL COMMAND]	To terminate.

[1] 1770-T1, -T2 or -T3

remain in the same position until a valid digit is entered.

9.3 EDITING

Changes to an existing program can be made through a variety of editing functions when the Processor is in PROGRAM mode. Instructions and rungs can be added or deleted; addresses, data and bits can be changed; a rung left incomplete due to an interruption while programming can be located and corrected. The Editing instructions are summarized in Table 9-2.

9.3.1 Inserting an Instruction

Only non-output instructions can be inserted in a rung. There are two ways of doing this.

One way is to press the key sequence [INSERT] [Instruction] [Key sequence of Address]. The new instruction will be inserted after the cursor's present position. If an instruction is to be entered at the beginning of a rung, the cursor must be positioned on the output instruction of the previous rung. However, if the cursor is on the END statement, the instruction will be inserted in the position preceding the cursor.

The other way to insert an instruction is to press the key sequence [INSERT] [←] [Instruction] [Key sequence of Address]. The new instruction will be inserted before the cursor's present position.

If, at any time, the memory is full, the instruction cannot be entered and a "MEMORY FULL" message will be displayed.

9.3.2 Removing an Instruction

Only non-output instructions can be removed from a rung. Output instructions can be removed only by removing the complete rung.

To remove an instruction, place the cursor on the appropriate instruction and press the key sequence [REMOVE] [Instruction]. If the wrong instruction is pressed, an "INSTRUCTIONS DO NOT MATCH" message will be displayed.

Note: Bit values and data of word instructions are not cleared. However, the Input Image Table bits will be rewritten during the next I/O scan except if the instruction had been removed in a forced ON condition. The force function would prevail until removed.

9.3.3 Inserting a Rung

A rung can be inserted anywhere within a program by pressing [INSERT][RUNG] and entering the instructions. The cursor can be positioned anywhere in the previous rung. The new rung will be inserted after the rung which contains the cursor. If it is necessary to remove a newly entered instruction, the rung must be completed first. If the cursor is on the END statement, the [INSERT][RUNG] keys need not be used. The rung can be entered just as in initial program entry.

TABLE 9-2 — Editing Functions

FUNCTION	MODE	INDUSTRIAL TERMINAL	KEY SEQUENCE	DESCRIPTION
Inserting a Condition Instruction	PROGRAM	Any ■	[INSERT] [Instruction] [Address] or [INSERT] [←] [Instruction] [Address]	Position the cursor on the instruction that will precede the instruction to be inserted. Then press key sequence. Position the cursor on the instruction that will follow the instruction to be inserted. Then press key sequence.
Removing a Condition Instruction	PROGRAM	Any ■	[REMOVE] [Instruction]	Position the cursor on the instruction to be removed and press the key sequence.
Inserting a rung	PROGRAM	Any ■	[INSERT] [RUNG]	Position the cursor on any instruction in the preceding rung and press the key sequence. Enter instructions. Editing is prevented until Output is entered.
Removing a rung	PROGRAM	Any ■	[REMOVE] [RUNG]	Position the cursor anywhere on the rung to be removed and press the key sequence. NOTE: Only addresses corresponding to OUTPUT ENERGIZE, LATCH and UNLATCH instructions are cleared to zero.
Change data of a word instruction	PROGRAM	Any ■	[INSERT] [Data]	Position the cursor on the word whose data is to be changed. Press the key sequence.
Change the address of a word instruction	PROGRAM	Any ■	[INSERT] [First Digit] [←] [Address]	Position cursor on a word instruction with data and press [INSERT]. Enter first digit of the first data value of the instruction. Then use the [←] and [→] keys as needed to cursor to the word address or data. Enter the appropriate digits.
Replace an Instruction or change Address of Instructions without Data	PROGRAM	Any ■	[Instruction] [Address]	Position cursor on the instruction to be replaced or whose address is to be changed. Press the key sequence.

TABLE 9-2 — Editing Functions (cont.)

FUNCTION	MODE	INDUSTRIAL TERMINAL	KEY SEQUENCE	DESCRIPTION
On-Line Data Change	RUN	Any [1]	[SEARCH] [5][1] [Data]	Position cursor on the word whose data is to be changed. Press key sequence. Cursor keys can be used.
			[INSERT]	Press [INSERT] to enter new data into memory.
			[CANCEL COMMAND]	To terminate On-Line Data Change
All Editing Functions	As Applicable	Any [1]	[CANCEL COMMAND]	Aborts the operation at the current cursor position.
[1] 1770-T1, -T2 or -T3				

If, by chance, the rung was inserted in the wrong position, it must be completed (press [-( )-][CANCEL COMMAND]) before it can be removed.

If, at any time, the memory is full, a "MEMORY FULL" message will be displayed and more instructions will not be accepted.

9.3.4 Removing a Rung

Removing a rung is the only way an output instruction can be removed. Any rung, except the last one containing the END statement, can be removed.

To remove a rung, position the cursor anywhere on that rung and press [REMOVE] [RUNG].

Note: Only bits corresponding to OUTPUT ENERGIZE, LATCH or UNLATCH instructions addresses are cleared to zero. All other word and bit addresses are not cleared when a rung is removed.

9.3.5 Changing Data of a Word Instruction

The data of any word instruction, except the Arithmetic and PUT instructions, can be changed in the PROGRAM mode without removing and re-entering the instruction. This is done by positioning the cursor on the appropriate word instruction and pressing [INSERT][Data Digits]. When the last digit of the data is entered, the function is terminated and the data is entered into memory. The function can also be terminated and entered into memory before the last digit is entered by pressing [CANCEL COMMAND]. Also, once the first digit has been entered, the [→] and [←] keys can be used to cursor to any digit in the address or value to make a correction.

9.3.6 Replacing an Instruction or Changing the Address of an Instruction Without Data

To replace one instruction with another, place the cursor on the instruction. Then press [Instruction] [Key Sequence of the Address]. This procedure also can be used when changing the address of an instruction that does not contain data.

9.3.7 On-line Data Change

The lower 12 bits of a word or word instruction excluding Arithmetic and PUT instructions can be changed while the Processor is in the RUN or TEST mode. This is done by positioning the cursor on the appropriate instruction and pressing [SEARCH][5][1]. The message "ON-LINE DATA CHANGE, ENTERING DIGITS" will be displayed near the bottom of the screen. The new digits will be displayed to the right of the message as they are entered. Use the [→] and [←] cursor control keys as needed. After the new data is displayed, press [INSERT] to enter the data into memory.

To terminate this function, press [CANCEL COMMAND].

WARNING: When the address of an instruction whose data is to be changed duplicates the address of other instructions in User Program, the consequences of the change for each instruction should be thoroughly explored beforehand. This is to guard against unexpected machine operation which could result in damage to equipment and/or injury to personnel.

9.4 DIRECTORIES (1770-T3 Industrial Terminal)

Directories have been developed as an aid in using the Industrial Terminal. They list the several functions common to a single multi-purpose key such as the [SEARCH] key. The directories are summarized in Table 9-3.

The Help directory, accessed by pressing the [HELP] key gives a master list of directories and the key sequence to access them. Three other directories that can be accessed from the Help directory are:

- Control functions by pressing [SEARCH] [HELP]
- Record functions by pressing [RECORD] [HELP]
- Clear memory functions by pressing [CLEAR MEMORY][HELP]

The other directories listed in the Help directory cannot be accessed. Certain functions listed in the Control function and Record function directories are not available with the Mini-PLC-2 Processor. If the keys to select any one of these are pressed, the Industrial Terminal will issue a "FUNCTION NOT AVAILABLE WITH THIS PROCESSOR" message.

9.5 SEARCH FUNCTIONS

The Industrial Terminal can be used to search the User Program for a specific instruction or address, the first or last rung, the first or last instruction of a rung or for an incomplete rung using the [SEARCH] key as part of the key sequence. In addition, the Industrial Terminal allows either a single rung or multiple rungs to be displayed. The Search instructions are summarized in Table 9-4.

9.5.1 Search for First Rung

The first rung of the program can be located from any point within the program in any mode of operation by pressing [SEARCH][↑]. This positions the cursor on the first instruction of the program.

TABLE 9-3 — Directories

FUNCTION	MODE	INDUSTRIAL TERMINAL	KEY SEQUENCE	DESCRIPTION
Help Directory	Any	1770-T3	[HELP]	Displays a list of the keys that are used with the [HELP] key to obtain further directories.
Control Function Directory	Any	1770-T3	[SEARCH] [HELP]	Provides a list of all control functions that use the [SEARCH] key.
Record Function Directory	Any	1770-T3	[RECORD] [HELP]	Provides a list of functions that use the [RECORD] key.
Clear Memory Directory	PROGRAM	1770-T3	[CLEAR MEMORY] [HELP]	Provides a list of all functions that use the [CLEAR MEMORY] key.
All Directory Functions	As Applicable	1770-T3	[CANCEL COMMAND]	To terminate

TABLE 9-4 — Search Functions

FUNCTION	MODE	INDUSTRIAL TERMINAL	KEY SEQUENCE	DESCRIPTION
Locate first rung of program	Any	Any [1]	[SEARCH] [↑]	Positions cursor on the first instruction of the program.
Locate last rung of program	Any	Any [1]	[SEARCH] [↓]	Positions cursor on the TEMPORARY END instruction, if present, or the END statement.
Locate first instruction of current rung	PROGRAM	Any [1]	[SEARCH] [←]	Positions cursor on first instruction of the current rung.
Locate output instruction of current rung	Any	Any [1]	[SEARCH] [→]	Positions cursor on the output instruction of the current rung.
Locate rung without an output Instruction	Any	1770-T3	[SHIFT] [SEARCH]	Locates any rung left incomplete due to an interruption in programming.
Locate specific instruction	Any	Any [1]	[SEARCH] [Instruction keys] [Address]	Locates instruction searched for. Press [SEARCH] to locate the next occurrence of instruction.
Locate specific word address	Any	Any [1]	[SEARCH] [8] [Address]	Locates this address in the program (excluding -| |- and -|/|- instructions). Press [SEARCH] to located the next occurence of this address.
Single rung display	Any	Any [1]	[SEARCH] [DISPLAY]	Displays the first rung of a multiple rung display. Press key sequence again to view multiple rungs.

[1] 1770-T1, -T2 or -T3

9.5.2 Search for Last Rung

The last rung of the program (END statement) can be located from any position in the program by pressing [SEARCH][↓]. The cursor will stop at the TEMPORARY END instruction, if present, or the END statement. With the 1770-T3 Industrial Terminal, if the cursor was on the TEMPORARY END instruction, the END statement can be reached by pressing the [SEARCH][↓] keys again. With the 1770-T1 or -T2 Industrial Terminal, the temporary END statement must be removed in order to locate the END statement.

9.5.3 Search for First Instruction of a Rung

With the Processor in the PROGRAM mode, the first instruction of the rung containing the cursor can be located by pressing [SEARCH] [←]. If not in PROGRAM mode, the cursor will move off the screen to the left. To bring it back, press the [→] key. The cursor is displayed by blinking the instruction.

9.5.4 Search for Output Instruction of a Rung

With the Processor in any mode, the output instruction of the rung containing the cursor can be located by pressing [SEARCH][→].

9.5.5 Search For Incomplete Rung (1770-T3 Industrial Terminal)

In the event that an interruption in programming occurred and a rung was inadvertently left without an Output instruction, this rung can be located by pressing the [SHIFT] [SEARCH] keys. The Processor can be in any mode.

9.5.6 Search For Specific Instruction and Specific Address

The procedures for finding a specific instruction or an address are similar. Any instruction in User Program can be located by pressing [SEARCH] [Instruction][Key Sequence of Address]. Any address (excluding those associated with EXAMINE ON and EXAMINE OFF instructions) can be located by pressing the keys [SEARCH][8][Key Sequence of Address]. The address entered is the word address. For the OUTPUT ENERGIZE, LATCH and UNLATCH instructions, the Industrial Terminal will locate all of the bit addresses associated with the word address.

The message "SEARCH FOR" and the entered key sequences will be displayed at the bottom of the screen. The message "EXECUTING SEARCH" will appear temporarily. The 1770-T3 Industrial Terminal will begin to search for the address and/or instruction from the cursor's position. It will look past the TEMPORARY END boundary to the END statement. Then it will continue searching from the beginning of the program to the point where the search began.

A 1770-T1 or -T2 Industrial Terminal will not look past the temporary END statement. It will continue searching from the beginning of the program to the point where it began the search

If found, the rung containing the first occurrence of the address and/or instruction will be displayed as well as the rungs after it. If the [SEARCH] key is pressed again, the next occurrence of the address and/or instruction will be displayed. When it cannot be located or all addresses and/or instructions have been found, a "NOT FOUND" message will be displayed at the bottom of the screen.

This function can be terminated at any time by pressing [CANCEL COMMAND]. All other keys are ignored during the search.

9.5.7 Single Rung Display

Upon power-up, a multiple rung display appears on the screen. A single rung can be viewed by pressing [SEARCH][DISPLAY]. To return to the multiple rung display, press [SEARCH][DISPLAY] again.

9.6 TROUBLESHOOTING AIDS

The following troubleshooting aids are useful during starting-up and when troubleshooting a system:

- Bit Manipulation and Monitor Functions (1770-T3 Industrial Terminal)
- FORCE ON and FORCE OFF Functions
- TEMPORARY END Instruction (1770-T3 Industrial Terminal)
- ERR Message Display

The Troubleshooting aids are summarized in Table 9-5.

9.6.1 Bit Manipulation and Monitor (1770-T3 Industrial Terminal)

Bit Monitor allows the status of all 16 bits of any Data Table word to be displayed. Bit Manipulation allows the status of the displayed bits to be selectively changed or forced, and is useful in setting initial conditions in the data of word instructions.

BIT MONITOR

Bit Monitor can function when the Processor is in any mode. By pressing the key sequence [SEARCH][5][3][Key Sequence of Word Address], the status of all 16 bits of the desired word will be displayed. While the cursor is in the word address field, the [→] and [←] keys can be used to change address digits.

The status of the 16 bits in the next highest or next lowest word address also can be displayed by pressing the [↓] or [↑] keys, respectively. Bit Monitor also can display the status of Force conditions, if any. See 9.6.2 below.

TABLE 9-5 — Troubleshooting Aids

FUNCTION	MODE	INDUSTRIAL TERMINAL	KEY SEQUENCE	DESCRIPTION
Bit Monitor	Any	1770-T3	[SEARCH] [5][3] [Address]	Displays the ON/OFF status of all 16 bits at specified word address and corresponding force conditions if they exist.
			[↑] or [↓]	Displays the status of 16 new bits at the next lowest or highest word address, respectively.
Bit Manipulation	PROGRAM or TEST	1770-T3	[SEARCH] [5][3] [Address]	Displays the ON/OFF status of all 16 bits at specified word address and corresponding force conditions if they exist.
			[→] or [←]	Moves cursor to the bit to be changed.
			[1] or [0]	Enter a "1" to set bit ON or a "0" to set bit OFF.
			See FORCING below	Forcing or removing forces from input bits or output devices.
Either of above			[CANCEL COMMAND]	To terminate.
FORCE ON or FORCE OFF instruction	TEST or RUN	Any [1]	[FORCE ON] [INSERT] or [FORCE OFF] [INSERT]	Position the cursor on the Image Table bit or bit instruction to be forced ON or OFF and press the key sequence. The input bit or output device will be forced ON or OFF. [2]
Removing a FORCE ON or FORCE OFF instruction	TEST or RUN	Any [1]	[FORCE ON] [REMOVE] or [FORCE OFF] [REMOVE]	Position the cursor on the Image Table bit or bit instruction whose force ON is to be removed and press the key sequence.
Removing all FORCE ON instructions	TEST or RUN	Any [1]	[FORCE ON] [CLEAR MEMORY]	Position cursor anywhere in program and press key sequence.
Removing all FORCE OFF instructions	TEST or RUN	Any [1]	[FORCE OFF] [CLEAR MEMORY]	Position the cursor anywhere in program and press key sequence.
Forced Address Display	Any	1770-T3	[SEARCH] [FORCE ON] or [SEARCH] [FORCE OFF]	Displays a list of the bit addresses that are forced ON or OFF. The [SHIFT] [↓] and [SHIFT] [↑] keys can be used to display additional forces.

TABLE 9-5 — Troubleshooting Aids (cont.)

FUNCTION	MODE	INDUSTRIAL TERMINAL	KEY SEQUENCE	DESCRIPTION
Any of the above			[CANCEL COMMAND]	To terminate
Inserting a Temporary END Statement	PROGRAM	1770-T1, -T2	[INSERT] [-|/|-] [8]	Position the cursor on the instruction that will precede the temporary END statement. Press the key sequence and END will be displayed on the screen. The remaining rungs will not be displayed or scanned.
Inserting a TEMPORARY END Instruction	PROGRAM	1770-T3	[INSERT] [T.END]	Position the cursor on the instruction that will precede the TEMPORARY END instruction and press the key sequence. TEMPORARY END will be displayed on the screen. The remaining rungs, although displayed and accessible, are not scanned.
Removing a Temporary END Statement	PROGRAM	1770-T1, -T2	[REMOVE] [-|/|-]	Position cursor on END statement and press key sequence. If temporary, it can be removed. If it cannot be removed, it is an END of program statement.
Removing a TEMPORARY END Instruction	PROGRAM	1770-T3	[REMOVE] [T.END]	Position cursor on the TEMPORARY END instruction and press key sequence.

[1] 1770-T1, -T2, or -T3

[2] When in TEST mode, the Processor will hold outputs OFF regardless of attempts to force them ON.

BIT MANIPULATION

Bit Manipulation can function when the Processor is in Program mode. When in TEST mode, the User Program may override the bit status in the next scan.

The [←] and [→] keys can be used to cursor over to any bit. With the cursor on the desired bit, its status can be changed by pressing the [1] or [0] key. Bit Manipulation also allows the forcing of Image Table bits as described in 9.6.2 below.

To terminate this function, press [CANCEL COMMAND].

WARNING: If it is necessary to change the status of any Data Table bit, be sure that the consequences of the change are thoroughly understood beforehand. If not, unpredictable machine operation could occur directly or indirectly as a result of changing the bit status. Damage to equipment and/or injury to personnel could occur.

9.6.2 FORCE ON and FORCE OFF Functions

The Force functions are used to selectively force an input bit or output device ON or OFF. The Processor must be in the TEST or RUN mode.

The Force functions determine the ON/OFF status of input bits and output devices by overriding the I/O scan. An input bit can be forced ON or OFF regardless of the actual state of the corresponding input device. However, forcing an output terminal will cause the corresponding output device to be ON or OFF regardless of the rung logic or the status of the Output Image Table bit.

From 1 to 16 bits of an Input Image Table word and from 1 to 16 terminals of an Output Module Group can be forced ON or OFF separately or in combination.

NOTE: When in TEST mode, the Procesor will hold outputs OFF regardless of attempts to force then ON, even though the output bit instruction will be intensified.

USING THE 1770-T1, -T2 or 1770-T3 (Series A Rev A) INDUSTRIAL TERMINAL

The same force, ON or OFF, can be applied to any of the bits within a word. However, if a bit in a different Input Image Table word, or a terminal in a different Output Module Group is forced the same way, all previous forces in that word or module group are instantly removed. For example, if any terminal in Output Module Group 013 is forced OFF, and another terminal such as 012/01 is then forced OFF, all force conditions in the Output Module Group 013 will be removed.

Forcing functions can be applied in Ladder Diagam display by placing the cursor on the desired Examine or Energize instruction. After positioning the cursor, any one of the following key sequences can be used for placing or removing a forced condition:

- [FORCE ON][INSERT]
- [FORCE OFF][INSERT]
- [FORCE ON][REMOVE]
- [FORCE OFF][REMOVE]

USING THE 1770-T3 (Series A Rev. B or later) INDUSTRIAL TERMINAL

Simultaneous forcing of bits in different Input Image Table words or terminals in different Output Module Groups is prevented by the 1770-T3 Series A Rev B or later model Industrial Terminal. If attempted, one of the following messages would appear.

SIMULTANEOUSLY FORCING BITS ON (OR OFF) IN TWO INPUT IMAGE TABLE WORDS IS NOT ALLOWED

SIMULTANEOUSLY FORCING BITS ON (OR OFF) IN TWO OUTPUT MODULE GROUPS IS NOT ALLOWED

A bit that is already forced, cannot be forced in the opposite mode. For example, if bit 012/03 is forced ON and an attempt is made to force it OFF, the following message will be displayed.

BIT ALREADY FORCED. EXISTING FORCE MUST BE REMOVED

Forcing functions can be applied using the 1770-T3 Industrial Terminal in either of two ways using: 1) Bit Manipulation/Monitor display of an I/O word or b) Ladder Diagram display of User Program. By pressing the key sequence [SEARCH][5][3][Key sequence of Address], the bit status and force status of the 16 corresponding input bits or output terminals of the desired word can be displayed. The [→] and [←] keys can be used to cursor over to the desired bit. Or, in the Ladder Diagram display, forcing can be applied by placing the cursor on an Examine or Energize instruction.

ALL MODELS

When in TEST mode, the Processor will hold outputs OFF regardless of attempts to force them ON even though the output bit instructions will be intensified.

In every mode except the PROGRAM mode, the ON or OFF status of a forced bit will appear beneath the bit instruction in the rung. In all Processor modes, a "FORCED I/O" message will be displayed near the bottom of the screen when bits are forced ON or OFF.

Note: The ON or OFF status of OUTPUT LATCH/UNLATCH instructions is also displayed below the instruction. However, this is displayed only in PROGRAM mode.

All Force ON or all Force OFF functions can be removed at once in Ladder Diagram Display by pressing either of the following key sequences:

- [FORCE ON][CLEAR MEMORY]
- [FORCE OFF][CLEAR MEMORY]

All force functions will be removed immediately if any of the following conditions should occur: the Industrial Terminal or Processor is disconnected or loses AC power; the [MODE SELECT] key is pressed; or a terminal of a different Output Module Group or a bit in a different Input Image Table word is forced (excluding 1770-T3 Series A Rev B or later).

WARNING: When an energized output is being forced OFF, keep personnel away from the machine area. Accidental removal of Force functions will instantly turn ON the output device. Injury to personnel could result.

9.6.3 Forced Address Display (1770-T3 Industrial Terminal)

A complete list of bit addresses that are forced ON and OFF can be displayed by the Industrial Terminal. Either of the following key sequences can be used.

- [SEARCH][FORCE ON]
- [SEARCH][FORCE OFF]

If all the bits forced ON or OFF cannot be displayed at one time, the [SHIFT] [↓] and [SHIFT][↑] keys can be used to display additional forced bits.

To terminate this display, press [CANCEL COMMAND].

9.6.4 TEMPORARY END Instruction

The TEMPORARY END instruction (or temporary END statement) can be used to test or debug a program up to the point where it is inserted. It acts as a program boundary because instructions below it in User Program are not scanned or operated upon. Instead, the Processor immediately scans the I/O Image Table followed by User Program from the first instruction to the TEMPORARY END instruction, or temporary END statement.

USING THE 1770-T3 INDUSTRIAL TERMINAL

When the TEMPORARY END instruction is inserted, the rungs below it, although visable and accessible, are not scanned. Their content can be edited, if desired. The displayed section of User Program made inactive by the TEMPORARY END instruction will contain the message "INACTIVE AREA" in the lower right-hand corner of the screen.

The TEMPORARY END instruction can be inserted in either of two ways:

a) Cursor to the last rung of the User Program to be kept active. Position the cursor on the output instruction. Press [INSERT][→][T.END]

b) Cursor to the first rung of the User Program to be made inactive. Position the cursor in the first instruction in the rung. Press [INSERT] [←][T.END].

To remove this instruction, position the cursor on it and press [REMOVE][T.END].

To enter a rung after the T.END instruction, place the cursor on the T.END instruction and press [INSERT][RUNG]. Then enter the new rung.

Although more than one TEMPORARY END instruction can be inserted, no rungs will be executed beyond the TEMPORARY END instruction closest to the beginning of the program. The TEMPORARY END instruction uses one word of User Program.

USING THE 1770-T1 OR -T2 INDUSTRIAL TERMINAL

The temporary END statement is similar in function to the TEMPORARY END instruction described above with the following exceptions. When the temporary END statement is inserted, the rungs below it are not shown (nor are they scanned) and no rungs can be entered below it. The temporary END statement will look just like the normal END statement.

The temporary END statement can be inserted in two ways:

a) Cursor to the last rung of the User Program to be kept active. Position the cursor on the output instruction. Press [INSERT][-|/|-][8].

b) Cursor to the first rung of User Program to be made inactive. Position the cursor on the first instruction in the rung. Press [INSERT] [←][-|/|-][8].

To remove this instruction, position the cursor on the END statement and press [REMOVE] [-|/|-]. If it is temporary, it will be removed and the subsequent program instructions will be displayed.

9.6.5 ERR Message for an ILLEGAL OPCODE

An illegal opcode is an instruction code that the Processor does not recognize. It will cause the Processor to fault and will be displayed as an ERR message in the ladder diagram rung in which it occurs. The 4-digit hex value of the illegal opcode is displayed above the ERR message by the 1770-T3 Industrial Terminal. The 1770-T1 or -T2 Industrial Terminal will display the ERR message without the hex value.

If an illegal opcode should occur, the rung containing it can be compared with the equivalent rung in a hard copy printout of the program. A decision must be made either to replace the error with its correct instruction, see paragraph 9.3.6 Replacing an Instruction, or to remove it. The ERR message due to an illegal opcode cannot be removed directly. Instead, remove and replace the entire rung as described in paragraphs 9.3.3 and 9.3.4 Inserting and Removing a Rung. The cause of the problem should be identified and corrected in to correcting the ERR message.

9.7 CLEARING MEMORY

The option of clearing the Data Table, User Program and Messages is available with various CLEAR MEMORY functions. The Clearing Memory instructions are summarized in Table 9-6.

9.7.1 Data Table Clear (1770-T3 Industrial Terminal)

Part or all of the Data Table can be cleared by pressing [CLEAR MEMORY][7][7], entering a start and end word address, and then pressing [CLEAR][MEMORY] again. The Data Table will be cleared between and including these two word addresses.

9.7.2 User Program Clear (1770-T3 Industrial Terminal)

Part or all of the User Program can be cleared by pressing [CLEAR MEMORY][8][8]. The User Program will be cleared from the cursor position to the TEMPORARY END instruction, or the END statement. Neither the Data Table nor Messages are cleared.

9.7.3 Partial Memory Clear

Part of the User Program and the Messages can be cleared by pressing [CLEAR MEMORY] [9][9]. The User Program and Messages are cleared from the cursor position to the end of memory. None of the bits in the Data Table are cleared.

9.7.4 Total Memory Clear

The complete memory can be cleared by pressing [SEARCH][↑] to position the cursor on the first instruction of the program and then pressing [CLEAR MEMORY][9][9]. This resets all the Data Table bits to zero. A total memory clear should be done before entering the User Program.

TABLE 9-6 — Clear Memory Functions

FUNCTION	MODE	INDUSTRIAL TERMINAL	KEY SEQUENCE	DESCRIPTION
Data Table Clear	PROGRAM	1770-T3	[CLEAR MEMORY] [7][7] [Start Address] [End Address] [CLEAR MEMORY]	Displays a start address and an end address field. Start and end word addresses determine boundaries for Data Table clearing. Clears the Data Table within and including addressed boundaries.
User Program Clear	PROGRAM	1770-T3	[CLEAR MEMORY] [8][8]	Clears User Program from the position of the cursor to the END statement or TEMPORARY END instruction. Does not clear Data Table or Messages.
Partial Memory Clear	PROGRAM	Any [1]	[CLEAR MEMORY] [9][9]	Clears User Program and messages from position of the cursor to end of memory. Does not clear Data Table.
Total Memory Clear	PROGRAM	Any [1]	[SEARCH] [↑] [CLEAR MEMORY] [9][9]	Position the cursor on the first instruction of the program. Clears total memory (Data Table, User Program and Messages).

[1] 1770-T1, -T2 or -T3

Section 10
PERIPHERAL FUNCTIONS INCLUDING REPORT GENERATION

10.0 GENERAL

There are several functions that can be performed with a Mini-PLC-2 and the Industrial Terminal. The functions include:

- Contact Histogram
- Report Generation
- Cassette Recorder Dump and Load
- Data Cartridge Recorder Dump and Load (1770-T3 Industrial Terminal)
- Ladder Diagram Dump
- Total Memory Dump (1770-T3 Industrial Terminal)

Except for the contact histogram and report generation, the remaining functions require the use of a peripheral device connected to Channel C of the Industrial Terminal.

10.1 BAUD RATE SETTING

The baud rate for Channel C must be set to match the baud rate of the peripheral device when a peripheral device other than the Digital Cassette Recorder (Cat. No. 1770-SA) or Digital Cartridge Recorder (Cat. No. 1770-SB) is used. The baud rate is the number of bits per second sent to/from Channel C. The baud rate for Channel C can be set in one of two ways:

- Setting switches 1, 2 and 3 of the Switch Group Assembly on the Industrial Terminal's main logic board (Table 10-1).
- Pressing [RECORD][n] and a number from 2 to 8 on the Industrial Terminal (Table 10-2).

A baud rate entered via the keyboard will override the default setting of the Switch Assembly Group if initially set to some other (often used) baud rate.

USING THE 1770-T3 INDUSTRIAL TERMINAL

Channel C must be ON to receive input from a peripheral device. Channel C is initially ON. It can be turned OFF by pressing [RECORD][9] and ON by pressing [RECORD][9] again. The ON/OFF status of Channel C and the baud rate will be displayed at the bottom of the screen when setting the baud rate using the [RECORD][n] keys where $2 \leqslant n \leqslant 8$.

TABLE 10-1 — Switch Group Settings

SWITCH			BAUD RATE
1	2	3	
Down	Down	Down	110
Down	Down	Up	300
Down	Up	Down	600
Down	Up	Up	1200
Up	Down	Down	2400
Up	Down	Up	4800
Up	Up	Down	9600

TABLE 10-2 — Key Sequence for Setting Baud Rate

KEY SEQUENCE	BAUD RATE
[RECORD][2]	110
[RECORD][3]	300
[RECORD][4]	600
[RECORD][5]	1200
[RECORD][6]	2400
[RECORD][7]	4800
[RECORD][8]	9600

USING THE 1770-T1 OR -T2 INDUSTRIAL TERMINAL

Channel C must be ON to receive input from a peripheral device. Channel C is initially OFF. It can be turned ON by setting the baud rate using the [RECORD][n] keys. Another way of turning ON channel C is as follows:

- Change to the Alphanumeric Keytop Overlay, 1770-KAA.
- Press [1][2], Mode Selection on the Alphanumeric Keytop Overlay.
- Press [4], the Channel C Alphanumeric Mode Option.
- Press [1] for ON ([0] for OFF).

Pressing the [1] key returns the display to the Alphanumeric Mode Option which displays

the ON/OFF status of Channel C. Pressing the [Mode Select] key will terminate the Alpha-numeric Mode Option and will turn OFF Channel C.

10.2 CONTACT HISTOGRAM

The Contact Histogram function displays the ON/OFF history of a specific memory bit. This can be monitored on the Industrial Terminal and can also be printed by a peripheral printer. If a peripheral device is used, the baud rate for Channel C of the Industrial Terminal must be set.

Any Data Table bit, excluding the Processor Work Areas, can be accessed by the Contact Histogram command. The status of the bit (ON or OFF) and the length of time the bit remained ON or OFF (in hours, minutes and seconds) will be displayed. The seconds are displayed to within 00.01 second (10 msec.) resolution.

There are two operating modes for the contact histogram, shown in Table 10-3:

- Continuous: It is accessed by pressing [SEARCH][6]. Once started, the histogram is displayed continuously until stopped.
- Paged: It is accessed by pressing [SEARCH][7]. The histogram is displayed one page at a time by user command.

After pressing [SEARCH][6] or [SEARCH][7], enter the bit address to be monitored.

After pressing [DISPLAY], the data of the histogram will be displayed on every other line with 5 frames of data per line. Each frame of data contains the ON or OFF status and the length of time in hours, minutes and seconds [read between the dash (-) symbols] in the format shown in Figure 10-1.

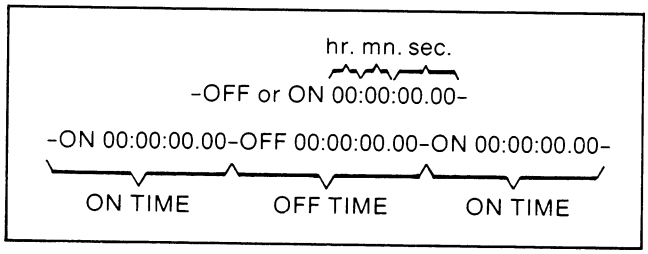


FIGURE 10-1 — Contact Histogram Display

If the bit is changing states faster than can be printed or displayed, a buffer is maintained to store these changes. If the buffer becomes

TABLE 10-3 — Contact Histogram Functions

FUNCTION	MODE	KEY SEQUENCE	DESCRIPTION
Continuous Contact Histogram	RUN or TEST	[SEARCH][6] [Bit Address] [DISPLAY]	Provides a continuous display of the ON/OFF history of the addressed bit in hours, minutes and seconds. Obtain a hardcopy printout of contact histogram by connecting a peripheral device to Channel C and selecting proper baud rate before entering indicated key sequence.
Paged Contact Histogram	RUN or TEST	[SEARCH][7] [Bit Address] [DISPLAY]	Displays 11 lines ON/OFF history of the addressed bit in hours, minutes and seconds.
		[DISPLAY]	Displays the next 11 lines of contact histogram. Obtain a hard copy printout of contact histogram by connecting peripheral device to Channel C and selecting proper baud rate.
Either	RUN or TEST	[CANCEL COMMAND]	To terminate.

full, all monitoring stops and a "BUFFER FULL" message will be displayed. Subsequent changes in the ON-OFF status of the device are lost until the histogram function finishes printing out or displaying the data in the buffer. Then a BUFFER RESET message will be displayed and the histogram function will resume.

The Industrial Terminal screen can display up to 11 lines of data at one time. In the continuous mode, the screen will automatically display a new page of data when the screen is full.

In the paged mode, 11 lines will fill the screen and stop. Subsequent changes are stored in the buffer until [DISPLAY] is pressed. The data stored in the buffer will then be displayed, one page at a time.

To terminate the contact histogram, press [CANCEL COMMAND].

10.3 REPORT GENERATION

Report Generation is a function of the Industrial Terminal. The 1770-T1 orT2 Industrial Terminal can generate up to 6 messages while the 1770-T3 Industrial Terminal (1770-FD Keyboard Series A Rev. B or later) can generate up to 70 messages. Report Generation is performed in the PLC-2 mode. (The Alphanumeric mode converts the Industrial Terminal into a peripheral device.) Messages can contain ASCII and graphic characters and variable Data Table information. Messages are stored in Processor memory after the END statement.

Messages can be entered into memory from either the Industrial Terminal or a peripheral device connected to Channel C of the Industrial Terminal. If the Industrial Terminal is used, one of two keytop overlays can be used, depending on whether graphic characters are desired (Figure 10-2):

- Alphanumeric Keytop Overlay (Cat. No. 1770-KAA)
- Alphanumeric/Graphics Keytop Overlay (Cat. No. 1770-KAB)

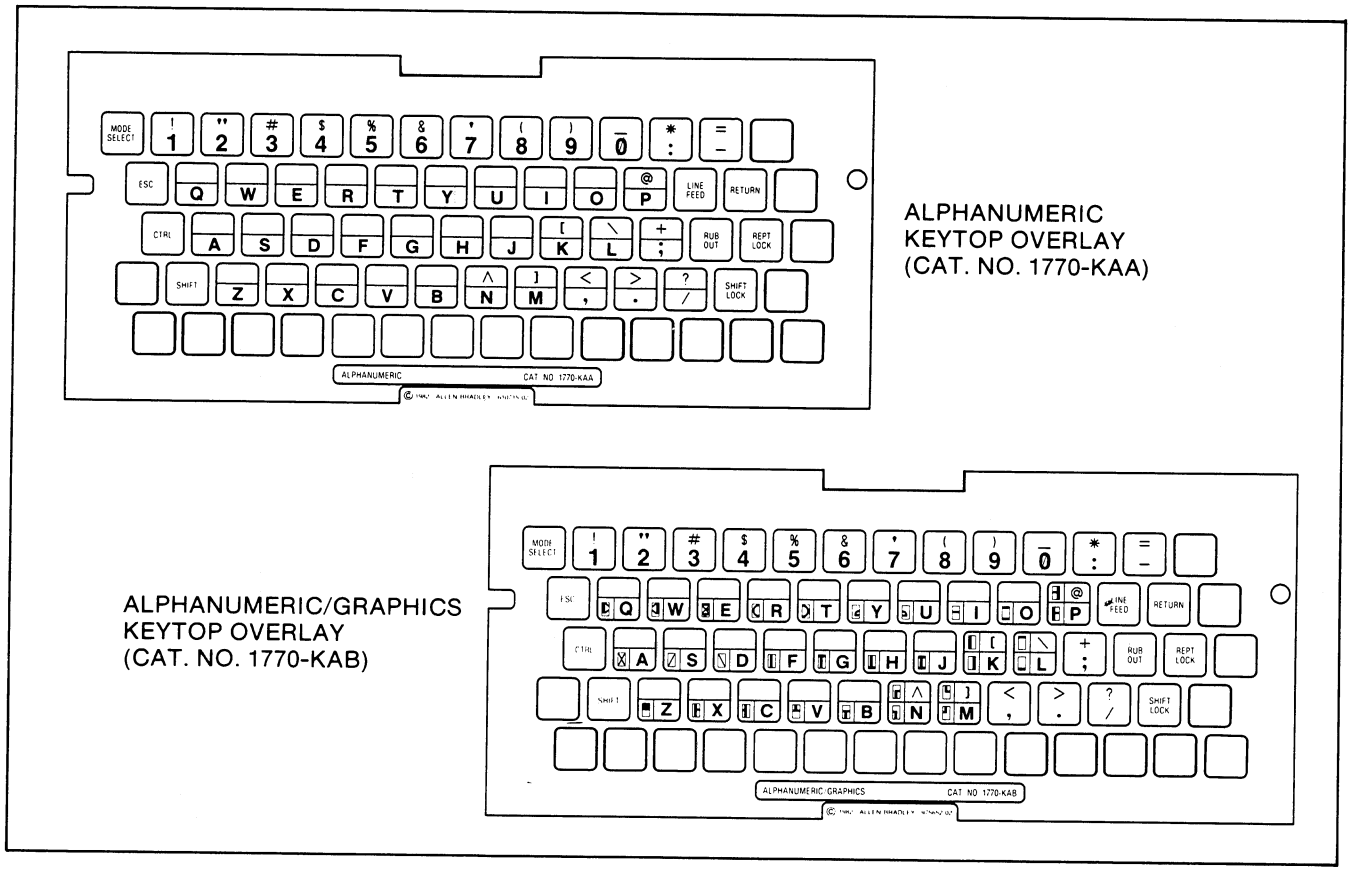


FIGURE 10-2 — Alphanumeric Keytop Overlays

The messages can be manually displayed or printed on the Industrial Terminal or peripheral device by a key sequence each time a message is desired. They can also be activated through program control by programming specific Data Table bits in the ladder diagram program (Automatic Report Generation, Section 10.3.2).

10.3.1 Report Generation Commands

The report generation function is entered by pressing [RECORD][DISPLAY] on the PLC-2 Keytop Overlay (Cat. No. 1770-KCA). A prompt CHANGE TO ALPHANUMERIC KEYTOP OVERLAY will be given. Or, set the baud rate. Then enter the desired report generation command from the peripheral device.

There are 5 report generation commands used to store, print, report and delete messages and to display an index of existing messages. These are summarized in Table 10-4.

MESSAGE STORE

Accessible only in the PROGRAM mode, this command is used to enter messages in memory. The Message Store command is accessed by pressing [M][S][,] [message number] [RETURN]. Valid message numbers are 1-6 for the 1770-T1, -T2 or -T3 Rev A Industrial Terminal. Valid message numbers for the 1770-T3 Industrial Terminal, Rev B or later are 1-6, 010-017, 110-117, 210-217, 310-317, 410-417, 510-517, 610-617 and 710-717.

TABLE 10-4 — Report Generation Commands

COMMAND	KEY SEQUENCE	DESCRIPTION
Enter Report Generation Function	[RECORD][DISPLAY] or set baud rate [message codes]	Puts Industrial Terminal into Report Generation Function. Same (entered from a peripheral device).
Message Store	[M][S][,][message number] [RETURN]	Stores message in Processor memory. Use [ESC] to end message.
Message Print	[M][P][,][message number][RETURN]	Prints message exactly as entered.
Message Report	[M][R][,][mesage number] [RETURN]	Prints message with current Data Table values or bit status.
Message Delete	[M][D][,][message number][RETURN]	Removes message from Processor memory.
Message Index	[M][I][RETURN]	Lists messages used and the number of words in each message.
Automatic Report Generation	[SEARCH][4][0] or [M][R][RETURN]	Allows messages to be printed through program control. Same (entered from a peripheral device).
Exit Automatic Report Generation	[ESC] or [CANCEL COMMAND] ■1	Terminates Automatic Report Generation. Same (entered for a periperal device).
Exit Report Generation Function	[ESC] or [CANCEL COMMAND] ■1	Returns to ladder diagram display. Same (entered from a peripheral device).

■1 [CANCEL COMMAND] can only be used if the function was entered by a command from a peripheral device.

After pressing the key sequence, a READY FOR INPUT message is presented as a prompt to enter the desired message. Any subsequent keys pressed then become part of the message.

While entering a message, each key pressed except the [SHIFT] [CTRL] [ESC] or [RUB OUT] key, generates a code that is stored in one byte of memory. This includes ASCII and graphic characters as well as other keys such as [LINE FEED], [RETURN] or the [SPACE] keys. The [RUB OUT] key is not stored in memory. The [SHIFT] and [CTRL] keys and the next character in the sequence are stored together in one byte of memory.

Messages can be entered which when reported will give the current value of a Data Table word or byte or the ON/OFF status of a Data Table bit by using the delimiters shown in Table 10-5. The desired delimiter is entered before and after the bit, byte or word address. The delimiter is used to tell the Industrial Terminal to print the current status or value of the bit, byte, or word at the address. As many addresses as needed can be entered consecutively by sharing the same delimiter, such as *XXX*XXX*XXX*.

As an example, suppose it was desired to report the output condition, On or OFF, of a device SR6, during each cycle of machine operation. Delimiters would be used to denote the output address 013/05, and the cycle counter Accumulative value (stored at 030_8). The desired message, SR6 is (ON or OFF) in CYCLE (XXX), would be entered into memory with the following keystrokes:

[S][R][6][ ][I][S][ ][*][0][1][3][0][5][*][ ][I][N][ ][C][Y][C][L][E][ ][*][0][3][0][*][.][ESC].

The message entry must be terminated with the escape ([ESC]) key. Until [ESC] is pressed, all key strokes become part of the message. Pressing [ESC] again will return to ladder diagram display. Pressing [CANCEL COMMAND] on the PLC-2 Family Keytop Overlay will also terminate the Message Store command and return to ladder diagram display if a peripheral device was used to enter report generator mode.

When entering a message, there are several keys and special Industrial Terminal control codes that are used to move through the display and perform a variety of functions (Tables 10-6 and 10-7). For example, graphic capability can be accessed by the control code, [CTRL] [P] [5] [G]. In addition, standard ASCII control codes can be used with the Industrial Terminal (Table 10-8). These codes, although not displayed, can be interpreted

TABLE 10-5 — Address Delimiters

DELIMITER FORMAT	EXPLANATION	MESSAGE REPORT FORMAT
XXX	Enter 3-digit word address between delimiters. (1770-T1, -T2 or -T3)	Displays BCD value at assigned word address.
#XXX#	Same (1770-T3)	Same
XXX1 or *XXX0*	Enter 3-digit word address and a "1" for upper byte or a "0" for lower byte between delimiters.	Displays the octal value of byte at assigned address.
XXXXX	Enter 5-digit bit address between delimiters. (1770-T1, -T2, -T3)	Displays the ON or OFF status of the assigned bit address.
^XXXXX^	Same (1770-T3)	Same

and acted on by a peripheral device connected to Channel C.

The Industrial Terminal screen size is an 80 x 24 format: 80 columns across by 24 lines down. An example message using graphic and alphanumeric characters is shown in Figure 10-3.

The control code, [CTRL] [P] [Column #] [;] [Line] [A], should be used for cursor positioning to conserve memory when possible. For example, [CTRL] [P] [3] [9] [;] [9] [A] uses 3 words of memory, storing CRTL P in one byte and each remaining character in one byte. If the cursor had been at column 0, line 0 and normal space and line feed commands were used, it would have taken 24 words of memory to accomplish the same thing! Note that the column and line numbers begin at zero rather than one.

TABLE 10-6 — Alphanumeric/Graphic Keytop Definitions

KEY	FUNCTION
[LINE FEED]	Moves the cursor down one line in the same column.
[RETURN]	Returns the cursor to the beginning of the next line.
[RUB OUT]	Deletes the last character or control code that was entered.
[REPT LOCK]	Allows the next character that is pressed to be repeated continuously until [REPT LOCK] is pressed again.
[SHIFT]	Allows the next key pressed to be a shift character.
[SHIFT LOCK]	Allows all subsequent keys pressed to be shift characters until [SHIFT] or [SHIFT LOCK] is pressed.
[CTRL]	Used as part of a key sequence to generate a control code.
[ESC]	Terminates the present function.
[MODE SELECT]	Terminates all functions and returns the Mode Select display to the screen.
Blank Yellow Keys	Space keys. They move the cursor one position to the right.

MESSAGE PRINT

Accessible in any mode, the Message Print command is used to print the contents of a message to verify it. This command is accessed by pressing [M] [P] [,] [message

TABLE 10-7 — Industrial Terminal Control Codes

CONTROL CODE KEY SEQUENCE	FUNCTION
[CTRL][P] [Column #][;] [Line #][A]	Positions the cursor at the specified column and line number. [CTRL][P][A] will position the cursor at the top left corner of the screen.
[CTRL][P][F]	Moves the cursor one space to the right.
[CTRL][P][U]	Moves the cursor one line up in the same column.
[CTRL][P][5][C]	Turns cursor ON.
[CTRL][P][4][C]	Turns cursor OFF.
[CTRL][P][5][G]	Turns ON graphics capability.
[CTRL][P][4][G]	Turns OFF graphics capability.
[CTRL][P][5][P]	Turns Channel C Outputs ON.
[CTRL][P][4][P]	Turns Channel C Outputs OFF.
[CTRL][I]	Horizontal tab that moves the cursor to the next preset 8th position.
[CTRL][K]	Clears the screen from cursor position to end of screen and moves the cursor to the top left corner of the screen.
KEY SEQUENCE	**ATTRIBUTE [1]**
[CTRL][P][0][T]	Attribute 0 = Normal Intensity
[CTRL][P][1][T]	Attribute 1 = Underline
[CTRL][P][2][T]	Attribute 2 = Intensify
[CTRL][P][3][T]	Attribute 3 = Blinking
[CTRL][P][4][T]	Attribute 4 = Reverse Video

[1] Any three attributes can be used at one time using the following key sequence: [CTRL][P][Attribute #][;][Attribute #][;][Attribute #][T]

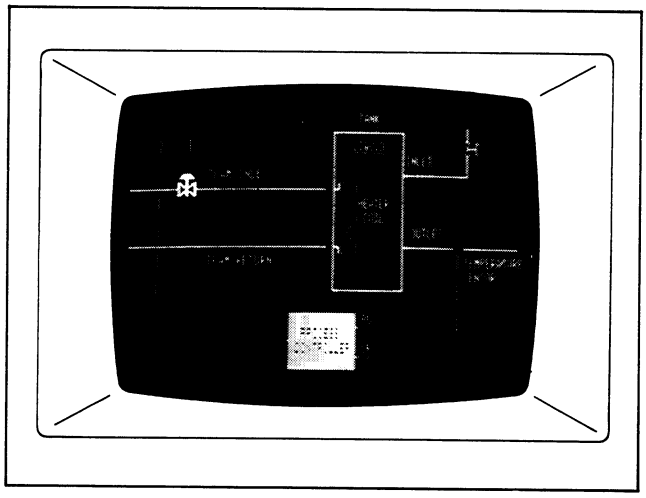


FIGURE 10-3 — Example Graphic/Alphanumeric Message

number] [RETURN]. Valid message numbers are listed under MESSAGE STORE.

In the example, the message print command would give the following:

SR6 IS *01305* IN CYCLE *030*.

The Message Print command is self-terminating. [ESC] or [CANCEL COMMAND] can be used to return to Ladder Diagram display.

MESSAGE REPORT

Accessible in any mode, the Message Report command is used to print a message with the current Data Table value or bit status that corresponds to an address between the delimiters. This command is accessed by pressing [M] [R] [,] [message number] [RETURN]. Valid message numbers are listed under MESSAGE STORE.

In the example, the Message Report command would give the following: (eg. bit 013/05 is ON and counter 030_8 Accumulated value is 5)

SR6 IS ON IN CYCLE 005

The Message Report command is self-terminating. When [ESC] or [CANCEL COMMAND] is pressed, ladder diagram operation will resume.

MESSAGE DELETE

Accessible only in PROGRAM mode, the Message Delete command is used to clear messages from memory. This command is accessed by pressing [M] [D] [,] [message number] [RETURN]. Valid message numbers are listed under MESSAGE STORE.

The Message Delete command cannot be terminated before completion. It will self-terminate after the message has been cleared from memory and a MESSAGE DELETED prompt will be printed. [ESC] or [CANCEL COMMAND] can be used to return to ladder diagram display.

MESSAGE INDEX

Accessible in any mode, the Message Index command prints a list of the message numbers used and the amount of memory (in words) used for each message. In addition, the number of unused memory words available will be listed.

The Message Index command is accessed by pressing [M] [I] [RETURN]. This command cannot be terminated before completion. It will self-terminate after the list is completed. To return to ladder diagram display, press [ESC] or [CANCEL COMMAND].

10.3.2 Automatic Report Generation

Messages can be printed through program control "automatically" by energizing specific message request bits using OUTPUT LATCH and OUTPUT UNLATCH instructions.

Automatic report generation can be accessed in the TEST or RUN modes by pressing [SEARCH] [4] [0] on the 1770-KCA overlay or by pressing [M] [R] [RETURN] on the 1770-KAA overlay. It can also be activated automatically upon initialization of the 1770-T3 Industrial Terminal by setting parity switches 4 and 5 UP on the Industrial Terminal's main logic board (Figure 10-4).

Once automatic report generation is activated, the message request bits are scanned by the Industrial Terminal for a 0-to-1 transition. Each time one of the request bits goes TRUE, the corresponding message will be printed automatically.

Messages 1-6 use bits 10-15 of word 027_8 as message request bits. For the 1770-T3 Industrial Terminal, Rev B or later, all other messages use control bits in a user-defined set of

message control words. These two catagories will be discussed separately below.

Automatic report generation can be terminated by pressing [ESC]. To return to ladder diagram display, press [ESC] again. Pressing [CANCEL COMMAND] will also terminate automatic report generation and return to ladder diagram display if automatic report generation was entered by a command from a peripheral device.

MESSAGES 1-6

The upper byte of word 027_8 is used to control messages 1-6. Bit 027/10 is the request bit for message number 1, bit 027/11 is the request bit for message number 2 and so on. Bit 027/16, the Busy bit is set ON when any of messages 1-6 are requested and will remain ON until all requested messages have been printed. Once all messages are generated, bit 027/17 will stay ON for 300 ms and is then set OFF.

TABLE 10-8 — ASCII Control Codes

CONTROL [1]	ASCII CODE	DISPLAY [2]	MNEMONIC NAME
CTRL 0 [3]	$^{N}_{U}$	NUL	NULL
CTRL A [3]	$^{S}_{H}$	SOH	START OF HEADER
CTRL B [3]	$^{S}_{X}$	STX	START OF TEXT
CTRL C [3]	$^{E}_{X}$	ETX	END OF TEXT
CTRL D	$^{E}_{T}$	EOT	END OF TRANSMISSION
CTRL E	$^{E}_{Q}$	ENQ	ENQUIRE
CTRL F	$^{A}_{K}$	ACK	ACKNOWLEDGE
CTRL G	$^{B}_{L}$	BEL	BELL
CTRL H	$^{B}_{S}$	BS	BACKSPACE
CTRL I	$^{H}_{T}$	HT	HORIZONTAL TAB
CTRL J	$^{L}_{F}$	LF	LINE FEED
CTRL K	$^{V}_{T}$	VT	VERTICAL TAB
CTRL L	$^{F}_{F}$	FF	FORM FEED
CTRL M	$^{C}_{R}$	CR	CARRIAGE RETURN
CTRL N	$^{S}_{O}$	SO	SHIFT OUT
CTRL O	$^{S}_{I}$	SI	SHIFT IN
CTRL P	$^{D}_{L}$	DLE	DATA LINK ESCAPE
CTRL Q	$^{D}_{1}$	DC1	DEVICE CONTROL 1
CTRL R	$^{D}_{2}$	DC2	DEVICE CONTROL 2
CTRL S	$^{D}_{3}$	DC3	DEVICE CONTROL 3
CTRL T	$^{D}_{4}$	DC4	DEVICE CONTROL 4
CTRL U	$^{N}_{K}$	NAK	NEGATIVE ACKNOWLEDGE
CTRL V	$^{S}_{Y}$	SYN	SYNCHRONOUS IDLE
CTRL W	$^{E}_{B}$	ETB	END OF TRANSMISSION BLOCK
CTRL X	$^{C}_{N}$	CAN	CANCEL
CTRL Y	$^{E}_{M}$	EM	END OF MEDIUM
CTRL Z	$^{S}_{B}$	SUB	SUBSTITUTE
ESCAPE	$^{E}_{C}$	ESC	ESCAPE
CTRL ,	$^{F}_{S}$	FS	FILE SEPARATOR
CTRL -	$^{G}_{S}$	GS	GROUP SEPARATOR
CTRL .	$^{R}_{S}$	RS	RECORD SEPARATOR
CTRL /	$^{L}_{F}$	US	UNIT SEPARATOR
DELETE	$^{D}_{T}$	DEL	DELETE

[1] Some ASCII control codes are generated using non standard keystrokes.

[2] Will be displayed when Control Code Display option is set ON in Alphanumeric mode, only. (Not in Report Generation mode).

[3] Valid key in Report Generation mode for 1770-FD Keyboard Series B Rev A or later.

FIGURE 10-4 — Parity Switch

ADDITIONAL MESSAGES (1770-T3 Industrial Terminal, Rev B or later)

Bits from eight consecutive user-selected words are used to control the 64 additional messages.

The eight message control words are determined by establishing a 2-word message in memory, called message 0. Message 0 is stored as follows:

[M] [S] [,] [0] [RETURN]

A prompt, MESSAGE CONTROL WORDS (3 DIGITS REQUIRED): will be printed. The word address of the first message control word must be entered. The Industrial Terminal will calculate and display the ending word address. The set of Message Control words can be located anywhere in the Data Table except Processor Work areas and Input Image Table. Once the first word address is entered, the Industrial Terminal will also display a Table which shows the message numbers associated with each message control word (Table 10-9).

The upper byte of each message control word contains the request bits for eight messages. There is an easy way to determine the message number from the bit which requests it. The three right-most digits in the bit address are coded to the message number. For example, if message number 312 were of interest, bit 12 of the third message control word would request message 312. See Figure 10-5.

TABLE 10-9 — Example Message Control Word-Message Number Relationship

CONTROL WORDS [1]	MESSAGE NUMBERS
170	010-017
171	110-117
172	210-217
173	310-317
174	410-417
175	510-517
176	610-617
177	710-717

[1] This table assumes user selected message control words begin at 170_8.

Unlike messages 1-6 which share a common Done bit (027/17), the additional 64 messages each have a separate Done bit. After a particular message has been printed, the Done bit is set until the User Program resets the request bit. Done bits are located in the lower byte of the message control words. Figure 10-6 shows this relationship. For example, if 124/15 is the request bit for a message, the Done bit is located at 124/05, 10_8 (one byte) below the request bit.

The Message Print command is valid for message 0. It will print out the message control word addresses in tabular form such as shown in Table 10-9. If the location of the message control file is to be changed or if message 0 is no longer needed, it can be deleted with the Message Delete command and re-entered at any time.

WARNING: Message control words should not be used for any other purpose. If a message control word is assigned to an Output Image Table address, be sure that neither slot of the corresponding module group contains an output module. Otherwise, a Request or Done bit would turn ON an output terminal in either module. Unexpected machine operation could result with possible damage to equipment and/or injury to personnel.

CONTROL WORD NUMBER	CONTROL WORD ADDRESS	MESSAGE NUMBERS
0	170	010-017
1	171	110-117
2	172	210-217
3	173	310-317
4	174	410-417
5	175	510-517
6	176	610-617
7	177	710-717

THE CONTROL WORD ADDRESSES ARE USER SELECTED.

MESSAGE NUMBER 3XX HAS A MESSAGE REQUEST BIT AT ADDRESS 173/XX. MESSAGE REQUEST BIT 173/XX, WHEN ENABLED, WILL ACTIVATE MESSAGE NUMBER 3XX WHERE XX ARE BIT NUMBERS $00\text{-}17_8$.

FIGURE 10-5 — Bit Address-Message Number Relationship

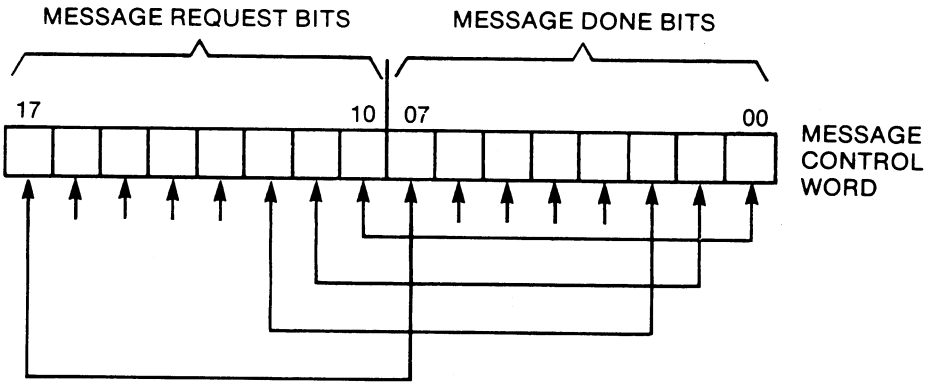


FIGURE 10-6 — Message Request Bit-Done Bit Relationship

EXAMPLE PROGRAMMING

Using LATCH and UNLATCH instructions, automatic report generation can easily be programmed to handle multiple or simultaneous message requests. Simultaneous requests are handled by a priority system, the lower the message number, the higher the priority.

Figure 10-7 shows a sample program that can be used to activate each message. When the event occurs which requests the message, the request bit is latched. After the event has ocurred and the message is printed (the Done bit comes ON), the request bit is unlatched.

10.4 DIGITAL CASSETTE RECORDER

The Digital Cassette Recorder (Cat. No. 1770-SA) is a peripheral device that connects to Channel C of the Industrial Terminal. It is used to dump memory onto tape, to load memory from tape and to verify memory.

10.4.1 Dumping Memory to Cassette Tape

The Cassette Dump command is used to dump (record) the contents of the Data Table, User Program and Messages onto a cassette tape. Although accessible in any mode, it is recommended that the dump be performed only in the PROGRAM mode because Data Table values are constantly changing in other modes.

To dump the complete memory onto the cassette tape, position the cursor on the first rung. The Cassette Dump command is then activated by pressing [RECORD][0] on the

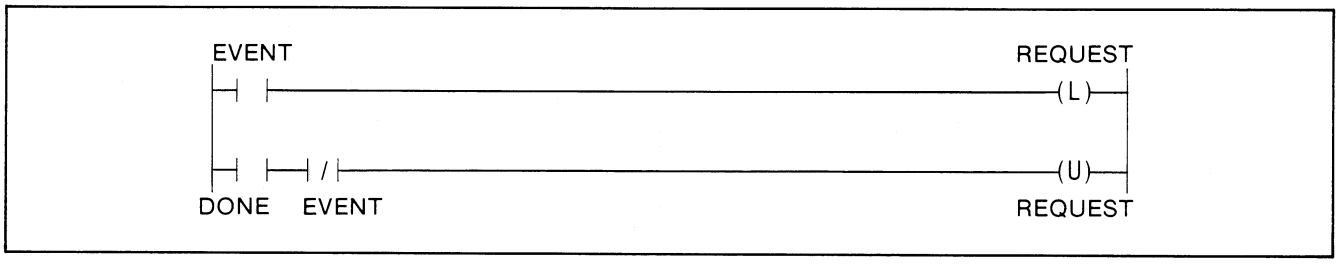


FIGURE 10-7 — Example Program to Request a Message

PLC-2 Overlay, and by pressing [RECORD ON TAPE] on the Cassette Recorder.

As memory is being recorded, the Industrial Terminal will count and display the number of Data Table words and Program words that were recorded on tape.

The Cassette Dump command is self-terminating. At completion, the content on the tape will be verified automatically by comparison with the Processor memory content unless the operation is terminated by pressing [CANCEL COMMAND].

10.4.2 Loading Memory from Cassette Tape

Loading the Processor memory from cassette tape can be done only in PROGRAM mode. The Data Table must be configured to the size which will match the Data Table of the taped program. Set the Data Table size as described is Section 9.1 Data Table Adjustment. If the size of the Data Table on tape is not immediately available and the Processor is configured differently, the load operation will abort automatically. The Industrial Terminal will display the Data Table configuration contained on the tape along with a prompt to configure the Processor Data Table.

The Cassette Load command is accessed by pressing [RECORD][0] on the PLC-2 Overlay and by pressing either [READ FROM TAPE] or [PLAY] on the Cassette Recorder. To load the complete memory, rewind the tape to the beginning of the program.

As memory is being loaded, the number of Data Table words and Program words will be counted and displayed. When loading is complete, the Processor memory content will be verified automatically with the content on tape unless the cassette function is terminated by pressing [CANCEL COMMAND].

10.4.3 Automatic Verification

This command can be accessed immediately after dumping or loading memory to/from the cassette tape to verify that an error-free transfer was made. The Processor must be in the PROGRAM mode to verify the Data Table.

This command is accessed by first pressing [REWIND] and then either [READ FROM TAPE] or [PLAY] on the Cassette Recorder. During verification, the number of Data Table words and Program words will be counted and displayed.

Once verification is complete, the number of program errors and whether the Data Table was verified will be displayed. The Automatic Verification command will self-terminate when complete. If program errors exist, they can be displayed and located by the procedure in paragraph 10.4.5 unless the cassette function is terminated by pressing [CANCEL COMMAND].

10.4.4 Program Verification

Accessible in any mode, this command is used to verify the User Program and messages in memory with the version on the cassette tape, or vice versa. Although the Data Table size and configuration are checked, the Data Table values are not verified.

This command is accessed by pressing [RECORD][1] on the PLC-2 Overlay and by pressing either [READ FROM TAPE] or [PLAY] on the Cassette Recorder. Rewind the tape to the beginning of the program beforehand.

When verification is complete, the command will self-terminate and display the number of

program discrepencies, if any. If discrepencies are found, either the tape can be re-recorded using the memory dump procedure, or the Processor memory can be corrected using the procedure in paragraph 10.4.5.

10.4.5 Displaying and Locating Errors

During automatic or program verification, the Processor will identify discrepancies between memory content and the content on the cassette tape. By pressing [SEARCH][9] on the PLC-2 Overlay, the number of program and Data Table discrepancies found and whether or not the Data Table was verified will be displayed. Up to 19 discrepancies can be detected.

Each program discrepancy can be searched for and located by pressing [SEARCH] and a number from [0][1] to [1][9]. Each time a discrepancy is searched for, the rung containing it will be displayed with the cursor positioned on the instruction that doesn't match the corresponding instruction on tape. A hard copy printout of the tape program is required for visual comparison. If the Processor memory is in error, it can be corrected using the editing procedure described in Section 9.3.

This function can be terminated at any time by pressing the [CANCEL COMMAND] key.

10.5 DATA CARTRIDGE RECORDER

The Data Cartridge Recorder (Cat. No. 1770-SB) is a peripheral device used for program storage and retrieval. It connects to Channel C of the Industrial Terminal and uses a magnetic data cartridge tape to record (dump), load and verify Processor memory.

The Data Cartridge Recorder can be operated from the 1770-T3 Industrial Terminal keyboard. It can also be operated in the same manner as a Digital Cassette Recorder (Cat. No. 1770-SA) using both the recorder control panel and the Industrial Terminal Keyboard. In either case, the baud rate switch in the Data Cartridge Recorder must be set to 1200.

It should be noted that when a data cartridge tape is inserted and the recorder is ON, the recorder will automatically rewind the tape to correct tape tension. This process should not be confused with the dump, load or verify operation.

Remote operation of the Data Cartridge recorder from the Industrial Terminal Keyboard is discussed in the following paragraphs. For operation in the same manner as a Digital Cassette Recorder, refer to paragraph 10.4.

10.5.1 Dumping Memory to Data Cartridge Tape

Data Table, User Program and messages can be recorded onto a data cartridge tape and the transfer verified by a single command from the Industrial Terminal. The Processor should be in PROGRAM mode to ensure that the Data Table values are not changing.

Once the cursor is positioned on the first instruction in User Program, the Cartridge Dump command is initiated by pressing [RECORD] [SHIFT] [B].

As memory content is being recorded on tape, the Industrial Terminal will count and display the number of User Program and Data Table words.

After memory content has been recorded, the tape is automatically rewound and the content verified with the content in memory to be sure that no discrepencies occuried during the recording operation. During verification, the number of User Program and Data Table words are again counted and displayed.

Once verification is complete, a message stating the number of discrepencies between Processor memory and tape content, if any, will be displayed. If one or more discprepencies are found, the entire recording operation should be repeated.

This command can be aborted at any time by pressing [CANCEL COMMAND].

10.5.2 Loading Memory From Data Cartridge Tape

Processor memory can be loaded from a data cartridge tape and the transfer verified automatically by pressing [RECORD] [SHIFT] [A] on the Industrial Terminal Keyboard. The Processor must be in PROGRAM mode.

The Data Table must be configured to the size which will match the Data Table of the taped program. Set the Data Table size as described is Section 9.1 Data Table Adjustment. If the size of the Data Table on tape is not immediately available and the Processor is configured differently, the load operation will abort automatically. The Industrial Terminal will display the Data Table configuration contained on the tape along with a prompt to configure the Processor Data Table.

The number of Use Program and Data Table words are counted and displayed while memory content is loaded and again during verification. After verification, a message displays the number of discrepencies found, if any. Instructions in memory that don't match corresponding instructions on the data cartridge tape can be located and displayed using the procedure described in paragraph 10.4.5, Displaying and Locating Errors. The discrepencies can be corrected if a hard copy printout of the program is available showing the correct instructions. Otherwise erase the entire memory (put the cursor on the first instruction and press [CLEAR MEMORY] [9][9]) and repeat the memory loading procedure.

This command can be aborted at any time by pressing [CANCEL COMMAND].

10.5.3 Data Cartridge Verification

This command is used to verify User Program and messages in Processor memory with the content in data cartridge tape, or vice versa. Although the Data Table size and configuration are checked, the Data Table content is not verified.

With the Processor in any mode, verification can be done by pressing the keys [RECORD] [SHIFT] [C] on the Industrial Terminal Keyboard. The number of User Program and Data Table words are counted and displayed while tape content and memory content are being compared. When verification is complete, the number of discrepencies, if any, is displayed. If discrepancies are found, either the tape can be re-recorded using the Dumping Memory procedure in paragraph 10.5.1, or the Processor memory can be corrected using the procedure in paragraph 10.4.5.

The verification process can be aborted at any time by pressing [CANCEL COMMAND].

10.6 LADDER DIAGRAM DUMP

Accessible in any mode, the Ladder Diagram Dump command is used to print out a hard copy of the User Program using a peripheral printer that is connected to Channel C.

After setting the baud rate for the printer, this command is accessed by pressing the keys [SEARCH][4][4] on the PLC-2 Overlay. The printout will begin from the current rung, allowing all or part of the program to be printed.

When the printout is complete, this command is automatically terminated. This command can be terminated before completion by pressing [ESC] on the peripheral device or [CANCEL COMMAND] on the PLC-2 Overlay.

10.7 TOTAL MEMORY DUMP

The Total Memory Dump command is accessible in the PROGRAM mode only. It is used to print out a hard copy of the Data Table, User Program and Messages using a peripheral printer connected to Channel C.

After setting the baud rate for the printer, this command is accessed by pressing the keys [SEARCH][4][5] on the PLC-2 Overlay and will print out the complete memory, regardless of cursor position (1770-T3, only).

The Data Table will be printed in hexadecimal. The bit pattern for each Data Table word will be as shown in Figure 10-8. In each row, the 4-digit octal word address is the address where the left-most hex value is stored. For example, the hex values ECCB16 and 024C16 are stored in word addresses 0020_8 and 0025_8, respectively. For more information on the hexadecimal numbering system, refer to Section 13, Numbering Systems.

The Data Table printout will be followed by the User Program in ladder diagram and block format. The messages will be printed out and identified by number.

DATA TABLE								
WORD ADDR	DATA							
0010	26C1	A4FF	952B	F073	D572	43CE	FFFF	300F
0020	ECCB	9A00	4721	002F	5101	024C	312B	AC0B
"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"
0177	2EC4	6F6D	ABCD	1C2D	4FGC	D10D	21F6	5BA2
"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"

FIGURE 10-8 — Data Table Printout in Hexadecimal

When the printout is complete, this command is automatically terminated. The Total Memory Dump Command can be terminated prior to completion by pressing [ESC] on the peripheral printer or [CANCEL COMMAND] on the PLC-2 Overlay.

Section 11
SPECIAL PROGRAMMING TECHNIQUES

11.0 GENERAL

There are several programming techniques that offer versatile control of the process or machine operation. They include:

- Scan Counter
- Block Transfer (1772-LN3 Processor Module)
- One-Shot
- Programming 0.01-Second Timers

11.1 SCAN COUNTER

The scan counter is a programming technique that can be used where timing may be an important consideration. For example, it can be used in diagnostics to compute and compare average scan times of an operation with predicted scan times, and to detect if repeated requests for an operation are being ignored due to a malfunction.

The scan counter is programmed using two rungs as shown in Figure 11-1. The first rung contains optional condition instructions and a CTU instruction whose Preset value equals the number of scans to be counted. The second rung is started with a BRANCH END instruction to open the rung and is ended with the same CTU instruction.

When the first rung goes TRUE, bit 17 of the CTU instruction is set to 1 and the CTU instruction Accumulated value will increment one count. The second rung will never go TRUE because it is an open rung. It is used to reset the CTU instruction Enable bit (bit 17) to zero so the CTU instruction can increment its count during the next scan. The CTU instruction will continue to increment past the Preset value unless it is reset by a CTR instruction.

11.2 BLOCK TRANSFER (1772-LN3 Processor Module)

Block Transfer can be performed with the 1772-LN3 Processor and any Bulletin 1771 I/O Module with Block Transfer capability.

11.2.1 Introduction

Block Transfer is a programming technique used to transfer up to 64 16-bit words of data in one scan from I/O modules to the Data Table and vice versa. It is used with intelligent 1771 I/O modules such as the Analog, Thermocouple, or Encoder/Counter modules which have this capability. Block Transfer can be compared with single transfer programming in which only one word of data is transferred per scan.

Block Transfer can be performed as a Read, Write or Bidirectional operation, depending on the I/O Module being used. An input module uses the Block Transfer Read operation, an output module uses the Block Transfer Write operation and a Bidirectional module can use either or both the Read and Write operations. During a Read operation, data is read into the Processor's memory from the

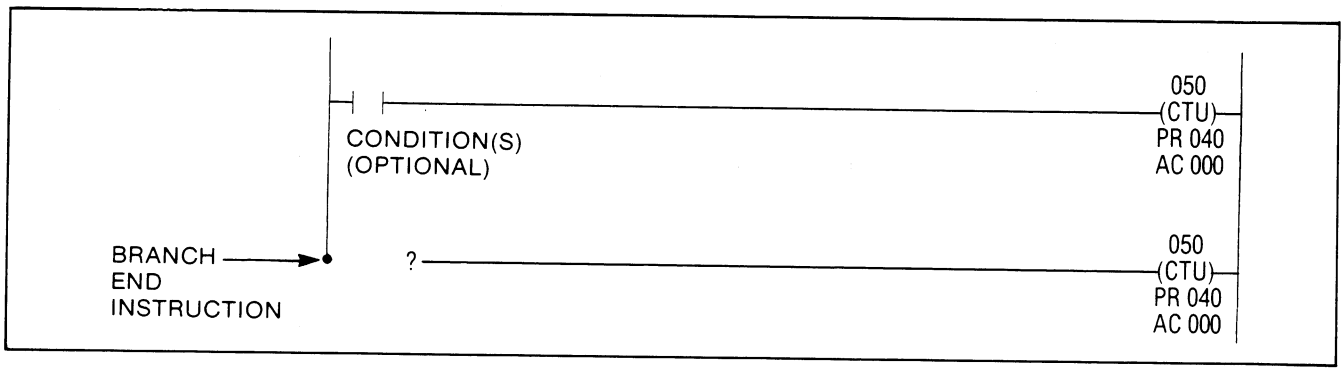


FIGURE 11-1 — Scan Counter

input module. During a Write operation, data is written to the output module from the Procesor's memory.

The number of words transferred in one scan can range from 1 to 64, depending on the I/O module being used. The time required to perform Block Transfer depends on the number of words being transferred.

The Processor uses two I/O Image Table bytes to communicate with Block Transfer modules. The byte corresponding to the module's address in the Output Image Table (Control byte) contains the Read or Write bit for initiating the transfer of data. The byte corresponding to the module's address in the Input Image Table (Status byte) is used to signal the completion of the transfer.

Whether the upper or lower byte of the I/O Image Table word is used depends on the position of the module in the Module Group. When in the lower slot, the lower byte is used and vice versa. See Figure 11-2. For double slot modules, the slot number must always be entered as a "0" because the overlapping of module groups is not permitted.

When Block Transfer is requested in the ladder diagram program, the output part of the next I/O scan is interrupted (Figure 11-3). If the I/O module is ready for Block Transfer, the transfer will be performed. When the transfer of data is complete, the Read bit (or Write bit) is set in the corresponding Input Image Table byte. The processor then continues its normal scan.

The I/O module may not be ready for Block Transfer 100% of the time. For a short time, the I/O module must perform internal checks on its own operation. If Block Transfer is requested when the I/O module is not ready, the Read or Write bit in the corresponding Input Image Table byte (done bit) is held at zero. The Processor will then continue with its normal I/O scan.

A Block Transfer read or write operation is requested by only one rung of program. A bidirectional Block Transfer is requested by two program rungs. Other support rungs, described later, may be necessary to support the Block Transfer operation. Of these support rungs, buffering data must be programmed to ensure validity of Block Transfer data.

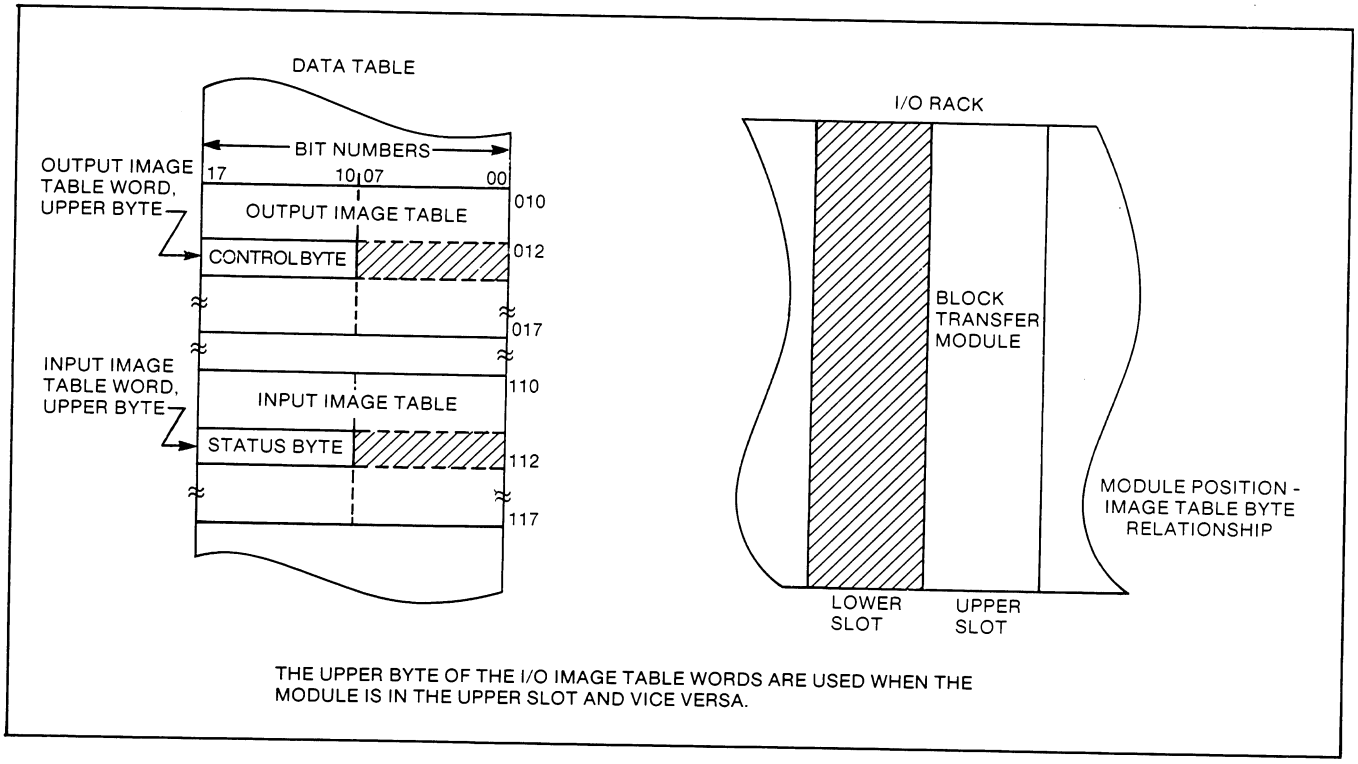


FIGURE 11-2 — Module Position/Image Table Byte Relationship

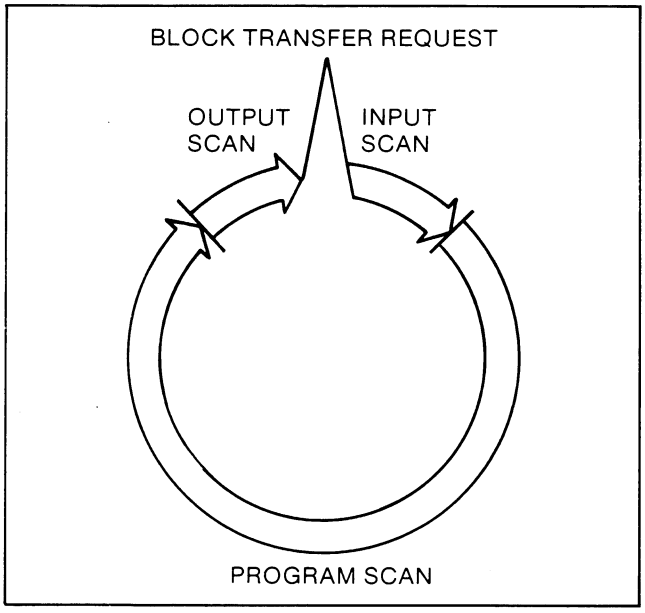


FIGURE 11-3 — Block Transfer Request

11.2.2 Block Transfer Rungs

The Block Transfer rung must be programmed in a certain format (Figure 11-4). It consists of condition instructions that are optional, two GET instructions and an OUTPUT ENERGIZE instruction.

FIRST GET INSTRUCTION

The first GET instruction (Figure 11-4) is used to identify the rack location of the block transfer module. The location must be stored in the first available address in the timer/counter Accumulated value area of the Data Table, starting with 030_8. When more than one Block Transfer module is used, consecutive addresses must be assigned in this area.

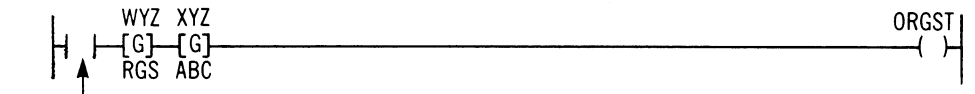
As its "data," the first GET instruction stores the location of the Block Transfer module by R=Rack, G=Module Group and S=Slot number. When Block Transfer is performed, the Processor searches the timer/counter Accumulated value area for a match of the module's Rack, Group and Slot number.

SECOND GET INSTRUCTION

The second GET instruction (Figure 11-4) must be assigned an address in the timer/counter Preset area of the Data Table, 100_8 words above the first GET address.

As its "data," the second GET instruction stores a Data Table address that designates the beginning of the area reserved for Block Transfer data. During a read operation, data is loaded into consecutive word locations starting with the designated address. During a write operation, data is sent to the module from consecutive word locations starting with the designated address.

When reserving an area for Block Transfer data, an appropriate address must be selected



WYZ = First available T/C Address (Accumulated Area)
XYZ = T/C Address 100_8 higher than WYZ (Preset Area).
RGS = Location of Block Transfer module where R=Rack, G=Module Group and S=Module Slot.
S is a zero for the left slot and a one for the right slot. For a 2-slot module, S is always zero.
ABC = Starting address where data is transferred to/from.
ORGST = OUTPUT ENERGIZE initiates Block Transfer where O=Output byte, R=Rack, G=Module Group, S=Module Slot, T=6 for write operation; 7 for read operation

FIGURE 11-4 — Block Transfer Rung

to ensure that a) none of the Block Transfer data can overrun word 077_8 or the upper boundary of the Data Table, word 177_8; (if so, the transfer is prevented) and b) that none of the Block Transfer data can write over addresses assigned to other functions. The reserved area begins with the starting address of the Block Transfer data (ABC in Figure 11-4) and requires consecutive word storage locations equal to the number of words to be transferred.

OUTPUT ENERGIZE INSTRUCTION

The OUTPUT ENERGIZE instruction (Figure 11-4) is used to initiate Block Transfer. It is given an address that indicates the module location and the type of Block Transfer operation. The first digit of the address is always "0" for output byte, even though an input or output Block Transfer module can be used. The next three digits identify the module location by Rack, Group and Slot. The last digit is either a "6" for a write operation or a "7" for a read operation. When Block Transfer is successfully completed, the Read bit (or Write bit) is set in the corresponding Input Image Table byte.

PROGRAMMING EXAMPLE

An example rung for performing a Block Transfer Read operation and the Data Table areas used by the rung are shown in Figure 11-5. The following parameters have been chosen for the example:

Module Address	121
Storage word containing the Module Address	030
Address of first word of Block Transfer data	060
Storage word containing address of first word of data	130

The module is located in Module Group 2, Slot 1. Therefore, the Control and Status bytes corresponding to the module's address in the Output and Input Image Tables are at word address 012_8 and 112_8 (upper bytes) respectively. The address of the Read bit and Done bit is 012/17 and 112/17, respectively.

During the program scan when input switch 113/02 is closed, the rung is enabled and Read bit 012/17 is set to "1." In the next scan of the Output Image Table, the upper byte data of word address 012 is sent to the module. The module responds that it is ready for transfer. The Processor interrupts the Output Image Table scan and starts searching the timer/counter Accumulated area of the Data Table. It finds the Module Address 121_8 in word address 030_8, and the address of the first word of Block Transfer data 060_8 in word address 130_8. The Processor then transfers the data from the module to Data Table addresses 060_8 through 067_8. When the number of words to be transferred is not programmed, the module will transfer its maximum (default) number which in this example was eight.

BIDIRECTIONAL BLOCK TRANSFER

A bidirectional operation requires one rung for a read operation and one rung for a write operation. Consecutive addresses in the timer/counter area of the Data Table should be selected for the GET instructions (Figure 11-6). For example, the first GET instruction of both rungs should be assigned consecutive word addresses such as 040_8 and 041_8. Both will have the same "data" to identify the module location at address 130_8.

The second GET instruction of both rungs will be assigned addresses 100_8 words above the first GET instructions. As "data," they will store the starting address for Block Transfer data such as 050_8 and 060_8.

The OUTPUT ENERGIZE instruction is addressed for a read operation in one rung and a write operation in the other rung. To let the Processor know whether a read or write operation is to be performed, bit 16 or 17 of the first GET instructions must be set ON. This can be done by programming an OUTPUT ENERGIZE or OUTPUT LATCH instruction unconditionally. For example, bit 16 of word 040_8 is set ON to indicate a write operation. Bit 17 of word 041_8 is set ON to indicate a read operation.

11.2.3 Support Rungs

There are additional techniques that can be used to support the Block Transfer operation:

- Loading Zeros

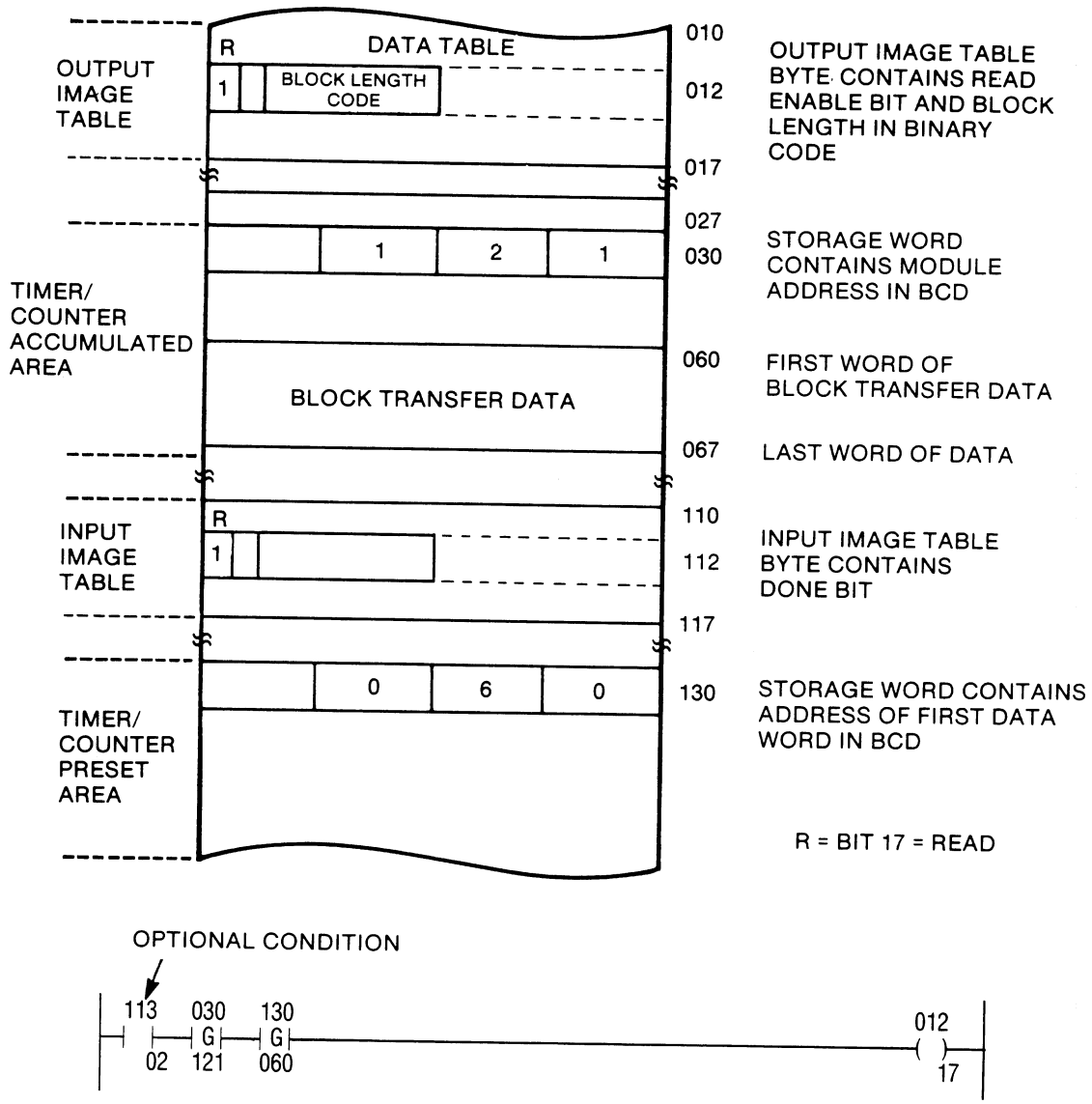


FIGURE 11-5 — Data Table Locations for a Block Transfer Read Operation

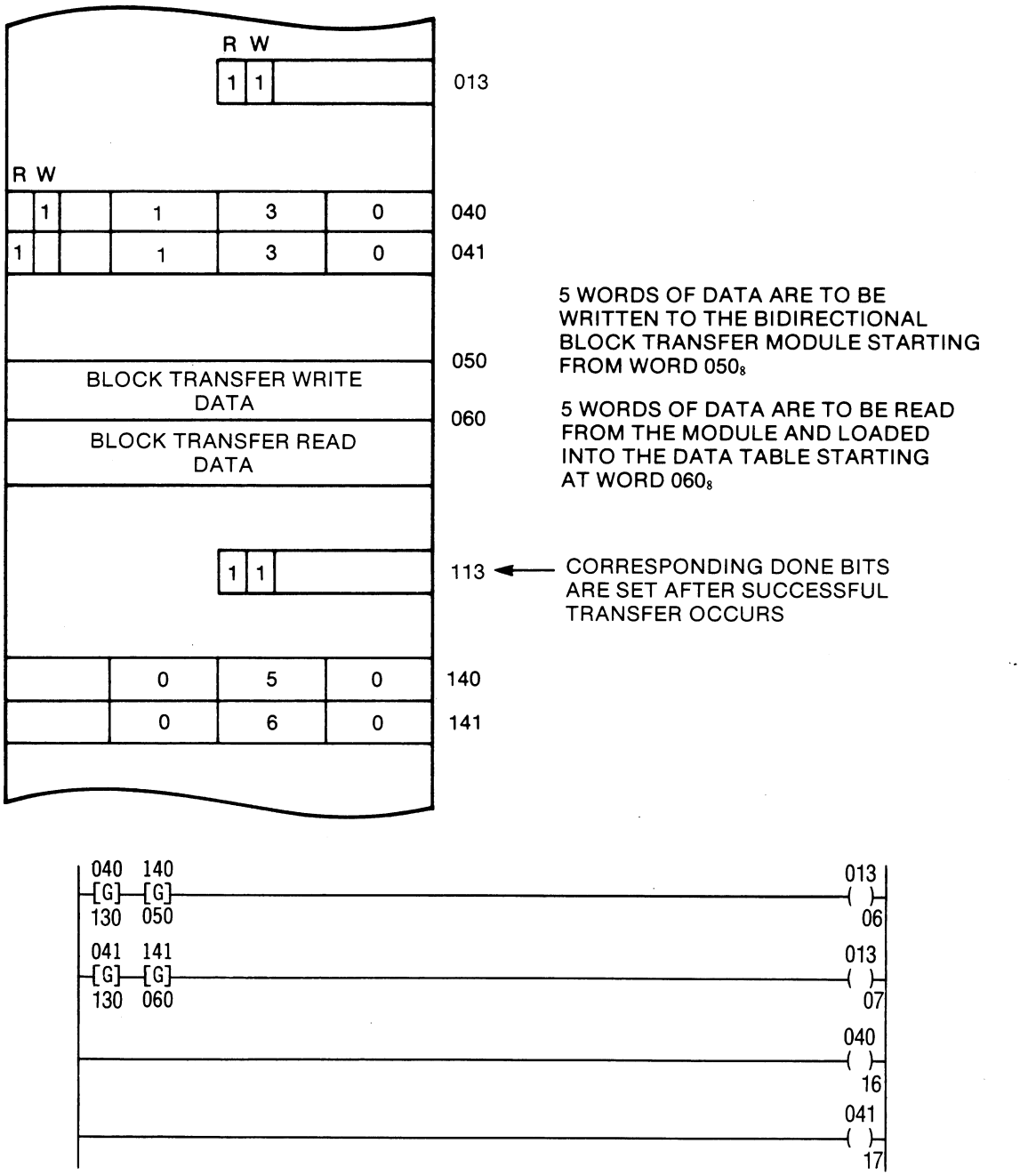


FIGURE 11-6 — Data Table Locations for Bidirectional Block Transfer

- Setting the number of words to be transferred
- Buffering data

Of these support rungs, buffering data must be programmed to ensure the Block Transfer data is valid. Other techniques, such as an IMMEDIATE OUTPUT instruction or a Scan Monitor, can also be programmed. The IOT instruction is used to request Block Transfer more than once per scan by assigning it the word address corresponding to the module's location. A Scan Monitor is used to monitor the number of scans that have occurred between each Block Transfer operation. For programming information on the Scan Monitor, refer to the respective User's Manual for the Block Transfer module.

LOADING ZEROS

When programming Block Transfer rungs, the first address(es) in the timer/counter Accumulated area of the Data Table starting at address 030_8 must be used to store the location(s) of the Block Transfer module(s). After the last module location is stored, the next consecutive address should be loaded with zeros. This serves as a boundary to prevent the Processor from searching the remaining timer/counter Accumulated value addresses where it could by chance, find a BCD value identical to the Rack, Module Group and Slot number of a Block Transfer module.

The boundary word data bits can be set to zero manually using Bit Manipulation [SEARCH][5][3], or by GET/PUT transfer. The GET/PUT transfer can be programmed by assigning the GET and PUT instructions to the address immediately following the last Block Transfer Data Address (Figure 11-7). The value of the GET instruction can be set to 000 when programmed.

SETTING THE NUMBER OF WORDS TO TRANSFER

Each Block Transfer module has a default value that specifies the maximum number of words that can be transferred. Either the default value or some lesser value can be selected. For Bidirectional Block Transfer, use the default value of the module. Because the default value can vary from one kind of module to another, consult the appropriate module documentation.

When a lesser number of words are desired, the number of words to be transferred must be set when programming the Block Transfer support rungs. The number of words is stored in the upper or lower Output Image Table byte

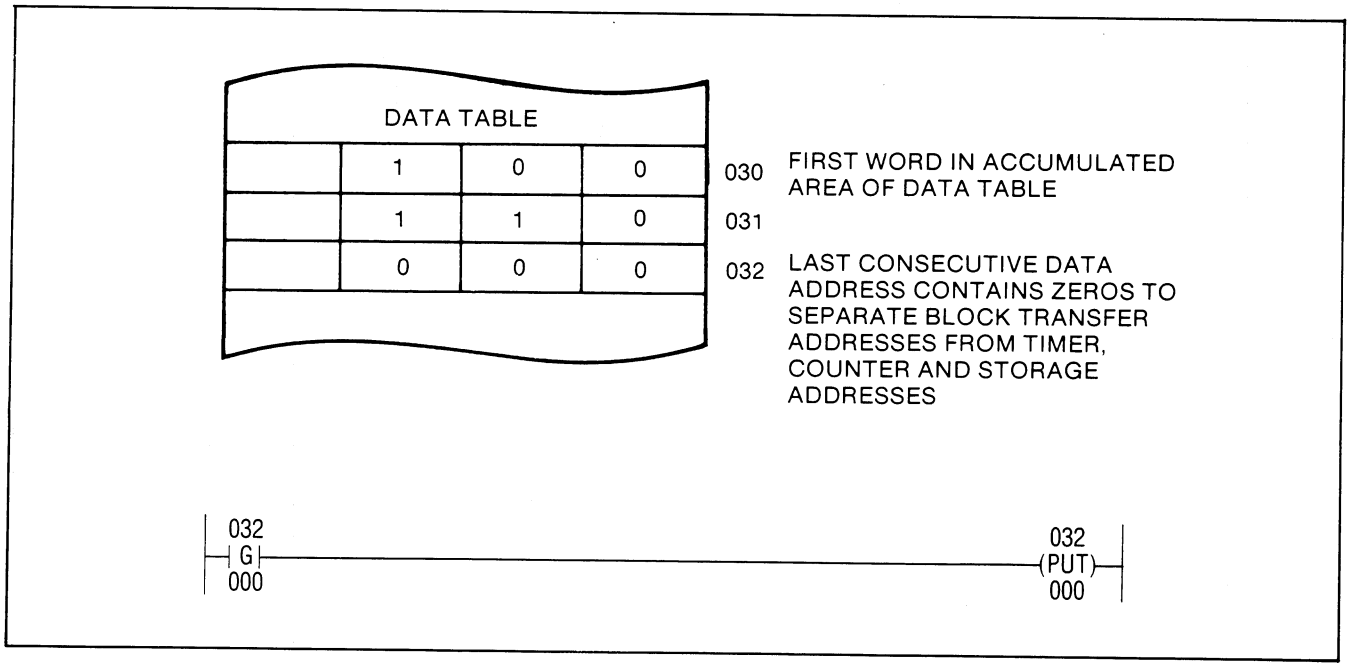


FIGURE 11-7 — Loading Zeros

that corresponds to the module's location. The appropriate bits must be set ON to specify a binary value that equals the number of words to transfer (Figure 11-8). These bits can be set ON by programming unconditional OUTPUT ENERGIZE or OUTPUT LATCH instructions.

For example, 5 words can be transferred by setting bits 00 and 02 high unconditionally in Output Image Table word 014. The binary equivalent of 5, as stated in the look-up table, is 000101.

If the number of words to be transferred or the location of Block Transfer data must change at different times, special programming techniques must be used. Refer to the appropriate documentation for the Block Transefer module for these programming techniques.

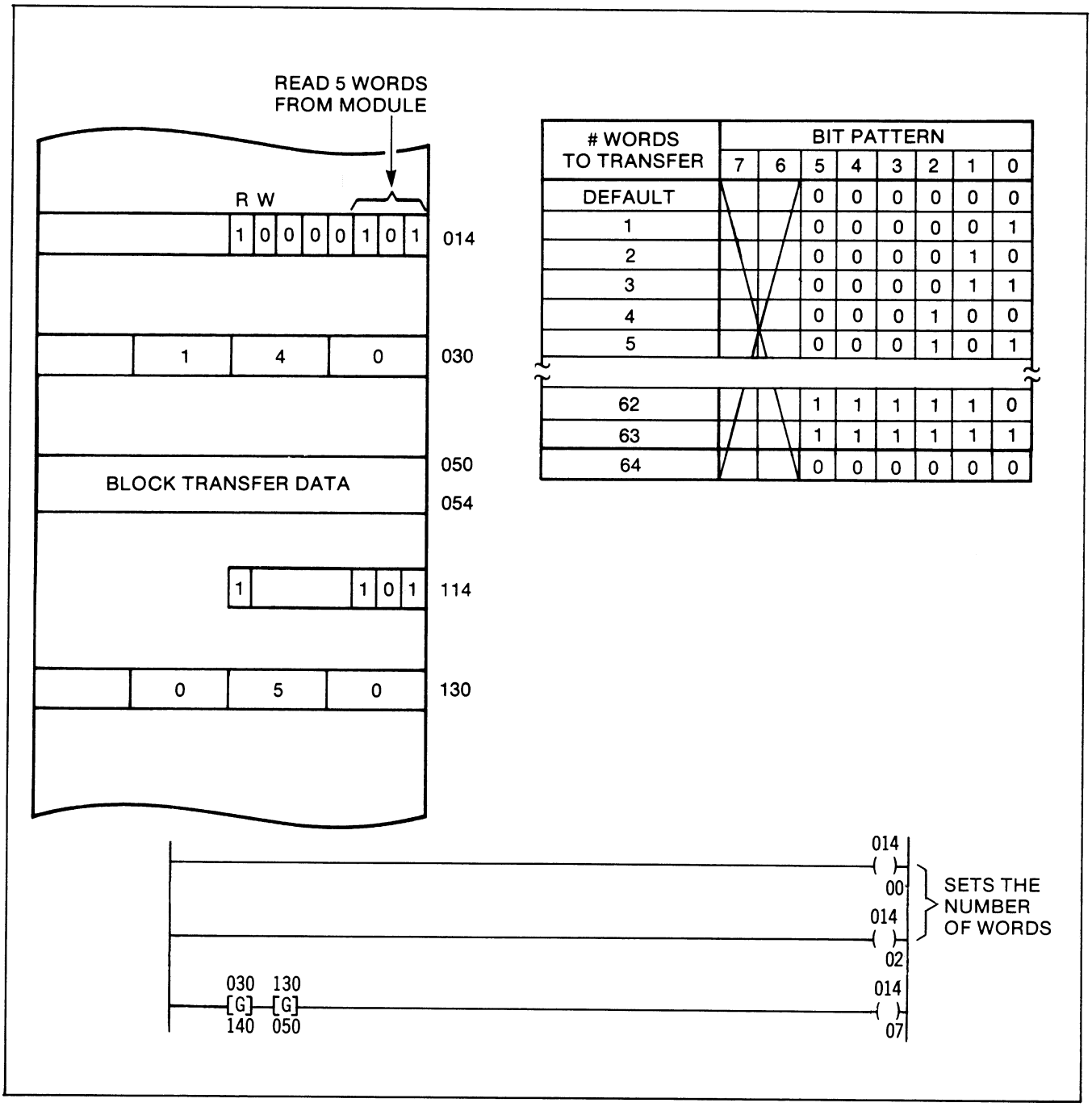


FIGURE 11-8 — Setting the Number of Transfer Words

BUFFERING DATA

The purpose of Block Transfer data buffering is to allow the data to be validated before it can be used. Data that is read from the Block Transfer module and transferred to Data Table locations must be buffered. Data that is written to the module need not be buffered because Block Transfer modules perform this function internally.

Transferred data is buffered to ensure that both the transfer and the data are valid. As an example, readings from an open-circuited temperature sensor (invalid data) could have a valid transfer from an analog input module to the Data Table. The Processor examines data-valid and/or diagnostic bits contained in the transferred data to determine whether or not the DATA is valid. The Read bit (or Write bit) is set in the Input Image Table byte if the TRANSFER is valid.

The data-valid and/or diagnostic bits differ for each Block Transfer module. Some modules set one or both for the entire group of words transferred, while others set a data-valid diagnostic bit in each word. Refer to the respective User's Manual for the Block Transfer module to determine the correct usage of the diagnostic and/or data valid bit(s).

One technique for bufferring data is to store the transferred data in a temporary storage location. If the data is valid, it is immediately transferred to another storage location in the Data Table where it can be used. If invalid, it is not transferred but written over in the next transfer. Another technique uses only one storage location. This technique prevents invalid data from being operated upon by preconditioning the rungs that would transfer data out of storage one word at a time.

Data can be moved from storage word-by-word using GET/PUT transfers. Generally, this method is used when one diagnostic bit is contained in each word. Diagnostic bits are examined as conditions for enabling the GET/PUT transfers.

The example in Figure 11-9 shows the memory map and ladder diagram rungs for buffering 3 words of data that are read from the Block Transfer module. Two storage locations are used. The data is read and buffered in the following sequence:

1. When Rung 3 goes TRUE, bit 014/07 (the Read bit) will be turned ON and Block Transfer will be requested. This latches ON storage bit 010/00 in Rung 4.

2. Block Transfer will be requested during the program scan. The transfer will be performed during an interruption of the next I/O scan. Data from the module will be loaded into words 050_8-052_8. When Block Transfer is complete, the Read bit 114/07 is set in the Input Image Table byte. This indicates Block Transfer was successfully performed. The Processor then continues with the I/O scan and program scan.

3. During the Program scan, Rung 1 will be TRUE because bit 010/00 is still latched ON and bit 114/07 is ON because Block Transfer was performed. This will turn bit 010/02 ON. In Rung 2, bit 010/00 is then unlatched.

4. In Rung 5, bit 010/02 is still ON and a diagnostic bit is examined to ensure the data read from the module is valid. Assuming the data is valid, the diagnostic bit will be ON and the data will be transferred from word 050_8 to 150_8. In rungs 5 and 6, the data in words 051_8 and 052_8 will be transferred to words 151_8 and 152_8 if the diagnostic bits are ON.

11.3 ONE-SHOT

The one-shot programming technique is used for certain applications to set a bit ON for one scan only. There are two types of one-shots that can be programmed:

- Leading Edge
- Trailing Edge

11.3.1 Leading Edge One-Shot

A leading edge one-shot is used to set a bit ON for one scan when its input condition has made a FALSE-to-TRUE transition. The FALSE-to-TRUE transition represents the leading edge of the input pulse. The programming for a leading edge one-shot is shown in Figure 11-10.

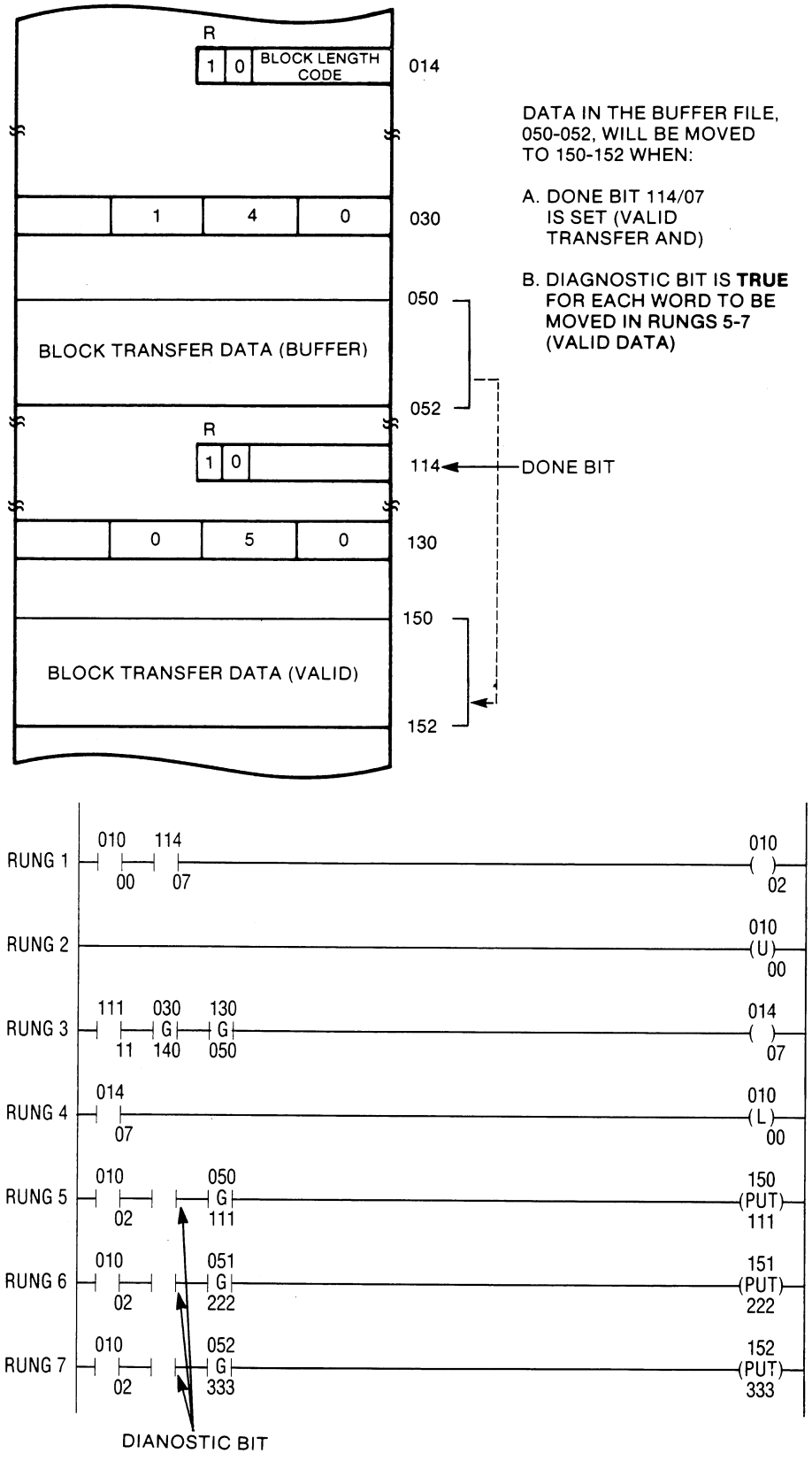


FIGURE 11-9 — Buffering Data

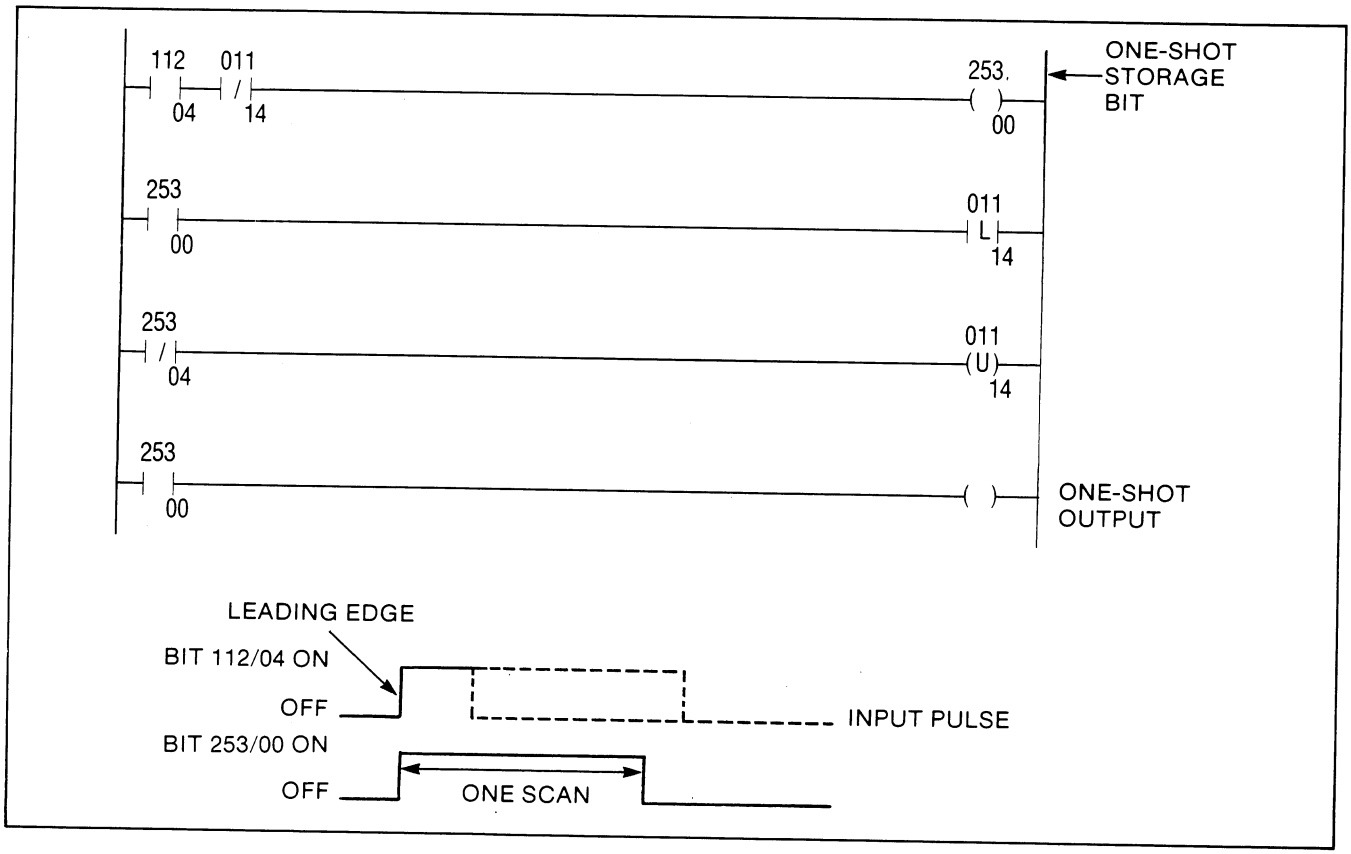


FIGURE 11-10 — Leading Edge One-Shot

When bit 112/04 makes a FALSE-to-TRUE transition, the one-shot bit (bit 053/00) is set ON for one scan. The length of time bit 112/04 remains ON does not affect the one-shot bit due to the next two rungs. Bit 011/14 will be latched ON when bit 112/04 is ON or bit 011/14 will be unlatched when bit 112/04 is OFF. During the next scan, either set of conditions will prevent bit 053/00 from being set ON. The one-shot bit is set ON for another scan only when bit 112/04 makes another FALSE-to-TRUE transition.

11.3.2 Trailing Edge One-Shot

A trailing edge one-shot is used to set a bit ON for one scan when its input condition has made a TRUE-to-FALSE transition. The TRUE-to-FALSE transition represents the trailing edge of the input pulse. Programming for a trailing edge one-shot is shown in Figure 11-11.

When bit 112/04 goes TRUE, bit 011/14 is latched ON. As soon as bit 112/04 makes a TRUE-to-FALSE transition, the one-shot bit (bit 053/00) is set ON and bit 011/14 is unlatched. Bit 053/00 will remain ON for only one scan. The input bit 112/04 must go TRUE and FALSE to set the one-shot bit ON for another scan.

11.4 PROGRAMMING 0.01-SECOND TIMERS

11.4.1 Introduction

The Bulletin 1772 Mini-PLC-2 programmable controller permits the user to enter On Delay Timer (TON), Off Delay Timer (TOF) and Retentive Timer (RTO) Instructions with a 0.01-second time base. These are also referred to as 10 millisecond (10-msec) timers.

Timers with a 10-msec time base provide the user with greater timing resolution and accuracy than is possible with a 0.1-second time base. Ten-msec timers are used when time delays from 0.02 to 9.99 seconds are required.

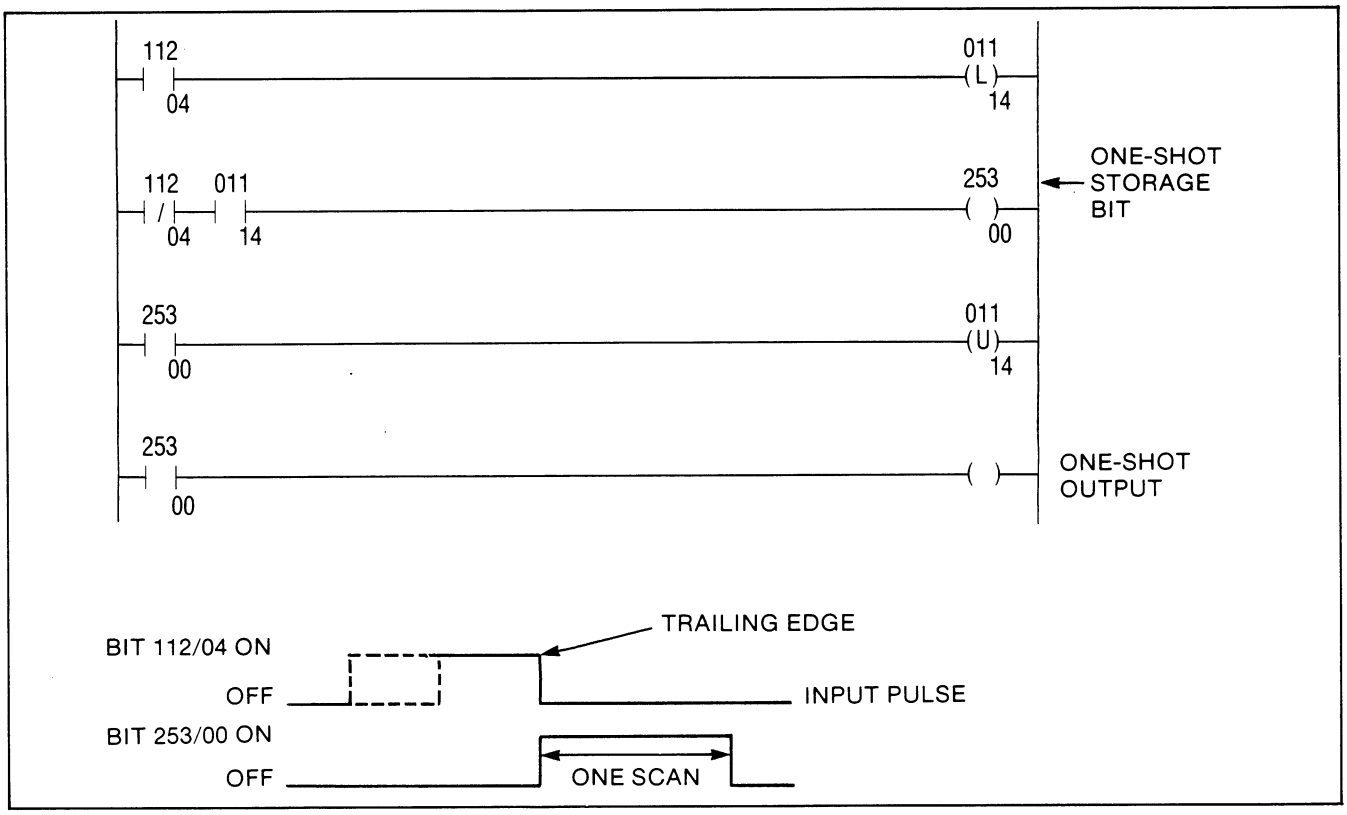


FIGURE 11-11 — Trailing Edge One-Shot

11.4.2 Time Base Selection When a timer Instruction is entered into a program, the programmer must specify:

- Timer word address
- Time base
- Preset value
- Accumulated value (for RTO only)

Note that the user selection of preset value and time base are closely related. The Processor executes the time-delay functions by incrementing the timer accumulated value one unit for each time base unit that elapses. In other words, the preset value represents a specific number of increments of the time base.

Note, however that the preset value is **not** an absolute length of time. For example, if the present value is 010, the time delay will be:

- 10 seconds if a 1.0-second time base is entered
- 1.0 second if a 0.1-second time base is entered
- 0.10 second if a 0.01-second (10-msec) time base is entered

The smaller the time base, the larger the preset value must be to obtain the same time delay. To obtain a 5-second time delay, the program could contain:

- 1.0-second time base and preset = 005
- 0.1-second time base and preset = 050
- 0.01-second time base and preset = 500

11.4.3 Timer Accuracy

Given any preset value, a Mini-PLC-2 controller timer is accurate to within one interval of its time base. Specifically, the timed interval does not exceed the preset interval, but it may be as much as 1 time-base unit shorter than the preset. These examples illustrate this accuracy:

- TON: Time base 1.0-second; preset value 100. This time interval will be greater than 99 seconds, but less than or equal to 100 seconds as given in the equation below.

99 seconds $<$ TON timed out $<$ 100 seconds

- TON: Time base 0.1-second; preset value 100. This time interval will be greater than 9.9 seconds, but less than or equal to 10.0 seconds as given in the equation below.

9.9 seconds < TON timed out < 10.0 seconds

- TON: Time base 10-msec; preset value 100. This time interval will be greater than 0.99 seconds, but less than or equal to 1.00 second as given in the equation below.

0.99 seconds < TON timed out < 1.00 second

Note that special programming techniques are required to use the 10-msec timer in a program. These techniques are explained in Scan Time and Program Execution, 11.4.6 and 11.4.7 below.

Programmed timers examine internal timing pulses of the Mini-Processor. (Refer to figure 11-12.) A change in the state of this internal clock causes the timer to increment its accumulated value. Note, however that the timing pulses are continuous and are **only** examined by the Mini-Processor when a timer Instruction is being executed in the program. As figure 11-12 shows, when the Mini-Processor initially examines this internal clock, the clock may have just changed state or may be just about to change state. This variable makes it possible for the inacuracy of up to 1 time-base increment.

Note that these timing accuracies refer only to internal Mini-Processor operation. That is, these intervals refer to the length of time which occurs between the moment that a timer is initialized (bit 17 set) and the moment that a timed interval is complete (bit 15 set). Other factors add to this timer inaccuracy. Chief among these are the response time of the actual hardware devices controlled and monitored by the Mini-PLC-2 controller. (Refer to Hardware and Processor Considerations, 11.4.5 below.)

The user is urged not to overspecify timing accuracy. In many applications timing within 0.1 second will provide accuracy comparable to, or better than, typical electomechanical timing relays. In general, these rules may be applied:

- For delays of 99 to 999 seconds, use the 1.0-second time base
- For delays of 2.00 to 99.9 seconds, use a 0.1-second time base
- For delays of 0.02 to 2.00 seconds, use the 10-msec time base

When time delays are incorporated in a program to provide a warmup or initializing period, or to prevent the simultaneous application of power to high-current devices, inaccuracies on the order of 50 to 250 msec are probably insignificant. The 1.0- or 0.1-second time bases are more than adequate for these uses. Applications for the 10-msec timer are discussed below.

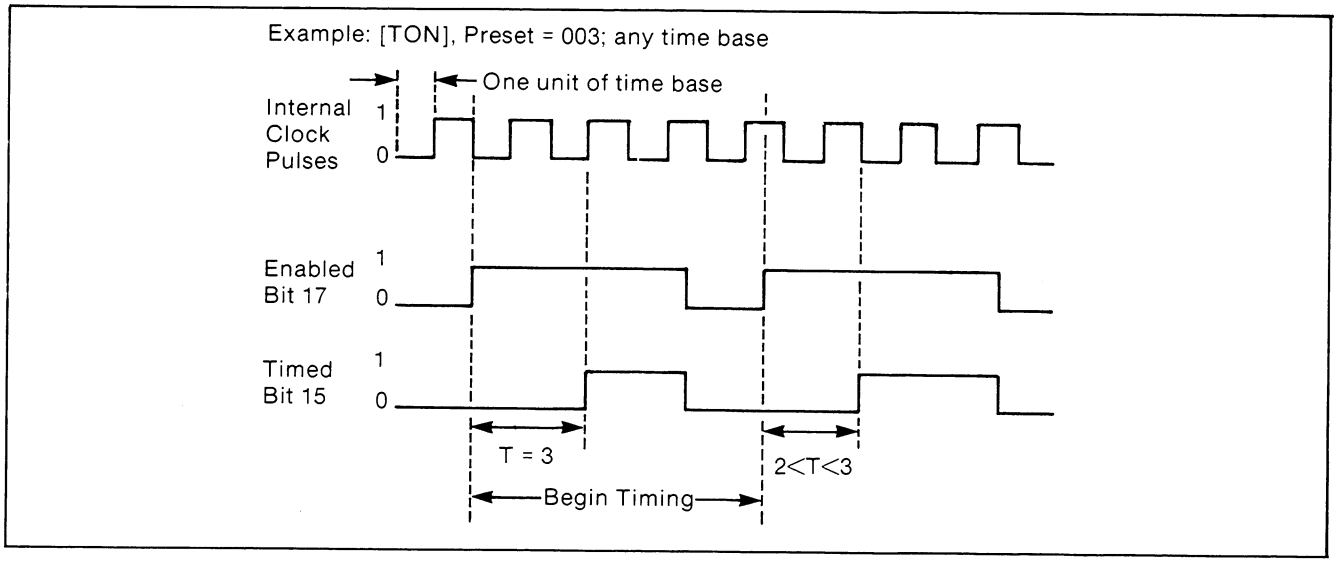


Figure 11-12 — Timing Diagram

11.4.4 Typical Applications

In general, 10-msec timers are used for these functions:

- Monitor events on a high-speed assembly or transfer line, such as that used in canning and bottling machines
- Generate short-duration pulses for accurate positioning control

For example, on a bottling or canning line, photoelectric sensors or electromagnetic proximity switches can be used to detect the movement of bottles. Each time a bottle passes a detector, an On Delay or Off Delay timer can be started. The next bottle down the line will turn the sensor on (or off), thereby resetting the timer. Once the second bottle is past the sensor, the timer is started again. If the bottles are moving too slowly, or if a bottle is missing, the timer will time out. The timed bit in the Data Table of the Mini-PLC-2 controller can be programmed to set off an alarm, or to stop the machine until the problem is corrected.

With the speeds encountered on a typical high-speed bottling machine, a timer with a 0.1-second time base would probably be too slow for this application. By computing the minimum bottle travel speed, the maximum time between bottles could be determined. The time in 10-msec increments could then be entered as the timer preset.

As another typical example, 10-msec timers could also be used to operate sorting mechanisms for high speed machines. Two methods could be used.

In the first method, the sort mechanism could be energized, for example, 60 msec after a REJECT is sensed by a particular sensor.

In the second method, the REJECT sense switch could immediately apply a 40-msec pulse to the sort mechanism. In this case, the pulse is just long enough for the mechanism to pull only one rejected bottle off the line.

Yet another example for the generating of short-duration pulses can also be found in machine tools and similar applications requiring accurate positioning control. Typically, 10-msec timers are used to generate one short duration pulse, or a series of pulses, when a limit switch or proximity switch detects end of travel, depth reached, or similar data. Detection that machining depth has been reached could, for example, generate a 130-msec pulse to the motor reverse circuit, thus plugging or braking the spindle with great accuracy.

Programmable control offers additional advantages in these applications. For example, consider a bottling machine capable of filling and capping 12-ounce and 16-ounce bottles. The larger bottles may move more slowly, or the spacing between bottles may be different. Detection of 16-ounce bottles could cause the Mini-PLC-2 Processor to GET different timer preset values and PUT them into monitoring and sorting timers such as those discussed above.

Changing the timer presets in this manner also enables the programmer to fine-tune the system without physically adjusting the locations of detection devices:

11.4.5 Hardware and Processor Considerations

When considering use of the 10-msec timer, the user must consider other timing factors, both within the programmable controller and in the hardware devices:

- Every input device requires a length of time to change state. Photoelectric devices and electromagnetic proximity switches typically operate in the range of 3 to 50 msec. Mechanical switches and magnetic control relays can require longer times for operation.
- Some Input Modules may provide a slight delay resulting from the input filter time constant (typically 10-25 msec).
- The execution of each program Instruction requires a certain length of time. Instruction execution times are discussed below in relation to program scan time computation.
- Scan time (I/O scan + program scan) depends on the number and type of Instructions, as discussed below. Incorporation of Immediate Input and Immediate Output Instructions can compensate for the length of scan time.

- DC Output Modules typically respond in 1 to 5 msec. AC Output Modules respond in 3 to 10 msec, depending on the instantaneous value of the AC wave when the turn-on signal is applied.
- User output devices may take 50 to 100 msec or longer to operate after current is applied. Inductive loads, or devices with substantial surge suppression circuitry, may also have longer response time.

Each of the items discussed above will have an impact on the actual time delay obtained from a programmable timer. Selection of fast-response indput devices is the responsibility of the user, and is beyond the scope of this document. For selection of suitable I/O Modules contact your local Allen-Bradley representative for further assistance.

The remainder of this section will discuss programming techniques which the programmer must use to effectively program the 10-msec timer. The required programming is based on 2 concepts:

- Scan time
- Sequential program scan

11.4.6 Scan Time

The Mini-PLC-2 Processor performs an I/O scan and then a program scan, in sequence. Scan time is the **sum** of the times required for both these scans. (Note that the Processor does not scan unused memory, nor does it scan that portion of the memory used to store Messages.)

During an I/O scan, the Processor examines Output Image Table bits, and updates or corrects the ON/OFF signals applied to the Output Modules. It also examines the ON/OFF signals from the Input Modules and updates the ON/OFF status of the corresponding Input Image Table bits.

During a program scan, the Processor scans each instruction in the Program, one at a time. It executes Examine and Branching Instructions, but it executes Ouptut Instructions only if the rung is TRUE. The sequential nature of this scan is discussed further below.

Scan time cannot be specified exactly for all processors because each user program is different. The length of the scan time depends on both the number and the type of Instructions the program contains. (Actual scan time computation is discussed below.) For purposes of discussion, scan time is generally assumed to be about 25 msec, though in practice it will range from about 15 msec to 50 msec, or more in certain extreme cases.

11.4.7 Program Execution

The second consideration for 10-msec timer programming is the sequential nature of the program scan. The Processor executes one program Instruction at a time. After it executes an Instruction, it cannot examine that Instruction again until the next scan of memory. With respect to timer Instructions, particularly, the Processor cannot increment the accumulated value except when it is executing that Instruction.

Furthermore, the only states of any memory bits that affect the execution of any single Instruction are the states those bits have at the instant the Processor executes the Instruction. If a bit changes state after the Instruction is executed, the change of state will **not** affect the instruction until it is executed the next time.

For example, suppose one program Instruction is Examine On 110/13. If the device is open, the Processor will detect an OFF signal from the Input Module during the I/O scan, and will clear (reset to "0") the corresponding Input Image Table bit. After the I/O scan, the program scan begins.

Suppose, in this case, that the input device wired to a terminal at address 110/13 closes when the program scan begins. The corresponding Image Table bit will remain "0" (device is open) until the next I/O scan after the current program scan is finished, or until the Processor executes an Immediate Input Instruction addressed to word 110.

The Processor can also update a timer only at the instant it is executing that timer instruction. Remember that an integral timing clock (see Section 11.4.3) puts out pulses for the 1.0-second, 0.1-second and 10-msec timers. When 1.0-second and 0.1-second timers are used in a program, the timing pulses are always longer than the Processor scan time. No special

programming is required; these timers will not miss a timing pulse.

Timing pulses for the 10-msec time base, however, are usually shorter than the program scan time. Since the Processor can only increment a timer while it is executing that Instruction, the 10-msec timer could miss one or more timing pulses on each program scan. The solution is to instruct the Processor to execute the timer Instruction often enough that it will not miss a pulse.

11.4.8 Programming Compensation

In order to compensate for the length of the scan time and to assure accurate timing, 10-msec timer programming must be repeated several places in the program.

A typical program using the total memory can nominally be assumed to have a scan time of less than 30 msec. (See Section 11.4.9.) In such a program, enter the same timer rung at 3 different places in the program: once near the beginning of the program, once near the middle, and once near the end. The Processor will update the timer accumulated value each time it scans that timer Instruction. Refer to Figure 11-13 and note the following:

- The rung must be identical each time it is used: the same Examine Instructions to condition the rung, the same timer word address, the same time base, and the same preset value.
- Use this technique only for 0.01-second timers. (The program scan is fast enough to assure accurate operation of the 1.0-second and 0.1-second timers with only one timer rung per program.)

Multiple entry of the timer rung will help to assure that the accuracy of the timer accumulated value is within the accuracy limits discussed above. Additional programming techniques can help to assure that output devices controlled by the timer are energized or de-energized after as precise a time delay as possible. The programmer may want to include:

- Multiple entries of rungs which examine the timed bit of the timer to condition an Output Energize Instruction.
- Immediate Input Instructions to help assure that the timer is enabled as quickly as possible after the external event occurs.
- Immediate Output Instructions to help assure that the output device is energized/de-energized as quickly as possible after the Mini-Processor sets the Output Image Table bit to "1" or clears it to "0".

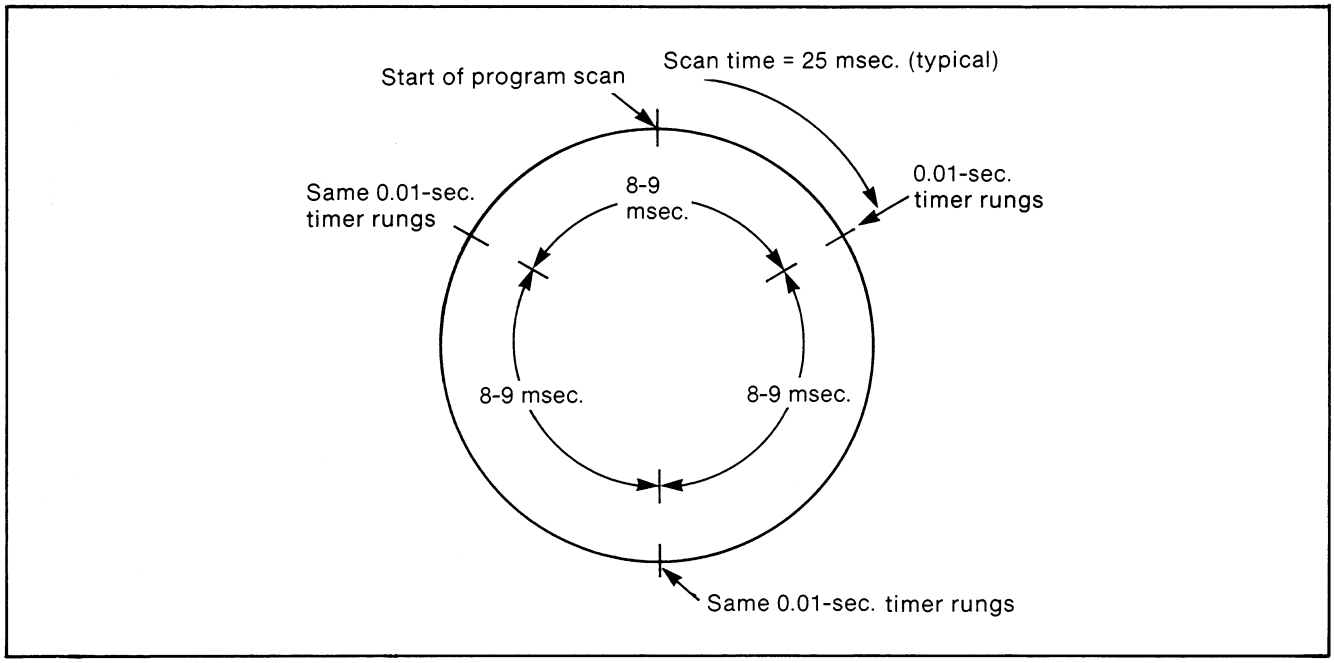


Figure 11-13 — Typical Timing Diagram for 0.01-Second Timer

Example 10-msec timer rungs are shown in Figure 11-14. In Rung No. 1, the Immediate Input Instruction precedes the Examine Instruction addressed to a bit in Input Image Table word 110. When used near the middle or end of a program, the Immediate Input Instruction helps to assure that the Processor will be executing the Instruction based on accurate data.

In Rung No. 2, the Output Energize Instruction conditioned by the timer bit should also be repeated in the program. When used near the beginning or middle of the program, the Immediate Output Instruction addressed to Output Image Table word 014 will help to assure that the Output Module will respond quickly to timer cycling.

By repeating the timer rung and related rungs, the programmer can assure that the Processor will update timer accumulated values more frequently than the timing pulses change state. As shown in Figure 11-13 repetition within 8 or 9 msec will be adequate for this purpose.

11.4.9 Program Scan Time Computation

In order to evaluate programming needs, the user may wish to calculate the approximate scan time. An exact computation is not practical, but a reasonable approximation can be obtained using the approximate execution times listed in Section 12, Table 12-1. Enter the 10-msec timer rungs 3 times per 1K (1024) words of program, then compute the scan time. A sample computation follows:

Assume the Processor is using a 128-word Data Table and has 1024 words of memory. If all memory words are used, the program will contain 896 Instructions. A program of this size might typically have the following distribution:

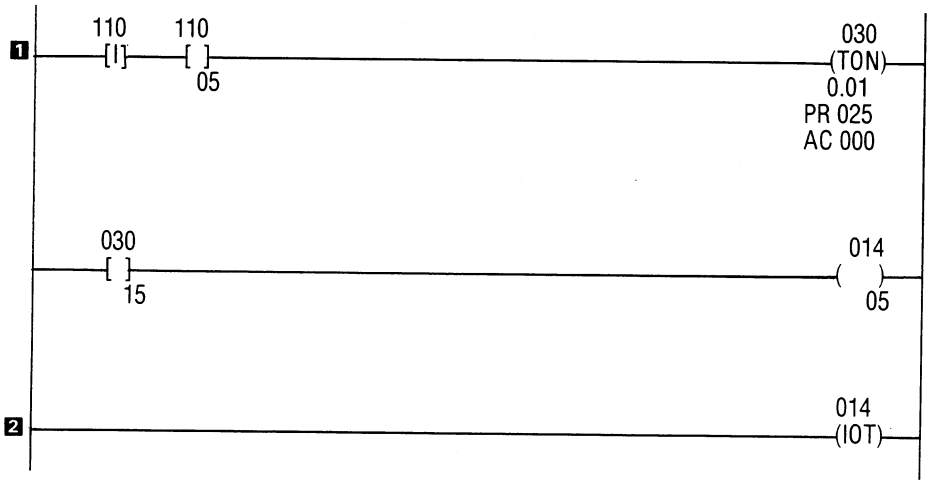
- 546 Instructions x 18 μsec = 9.8 msec
- 306 Instructions x 28 μsec = 8.6 msec
- 44 Instructions x 83 μsec = 3.7 msec
- Total (rounded): = 22 msec

The I/O scan time adds approximately 1.0 msec + 1 msec

Program Panel interaction requires about 3.0 msec + 3 msec

Total: 26 msec

If a 10-msec timer is used in a program of this duration, the timer rung and related rungs must occur at least 3 times in the program at evenly spaced intervals. For longer programs, it may be necessary to repeat the timer instruction and related rungs several more times to assure that timing is accurate.



Repeat these rungs 3 or more places in the program.

1 When used near the beginning of the program, the [I] instruction can be omitted.

2 When used near the end of the program, this rung can be omitted.

Figure 11-14 — Example 0.01-Second Programming

SECTION 12
SCAN TIME AND INSTRUCTION EXECUTION TIMES

12.0 GENERAL

In order for the Processor to implement the User Program, it must evaluate the action that it must take based on monitoring the status of input conditions. In addition, it must control the status of output devices in accordance with the program logic. Every instruction in User Program requires execution time. Execution times vary greatly depending upon the instruction, the amount of data to be operated on and whether the instruction is FALSE or TRUE.

12.1 SCAN TIME

Scan time is the amount of time it takes the Processor to monitor and update inputs and outputs, and to execute instructions in memory in accordance with User Program. The scan is performed serially, first the I/O Image Table is updated, (other parts of the Data Table are not scanned), then the User Program.

Scan time is typically 23 msec for a program of 900 instructions. The scan time is increased by approximately 3 msec when the Industrial Terminal is connected to the Processor. When the Communication Adapter Module or the PLC-2 Family/RS-232-C Interface Module, 1771-KA or 1771-KG, respectively, is connected to the Processor, the scan time is increased by approximately 5 msec.

12.1.1 Determination of Average Scan Time (1772-LN3 Processor)

Using the scan counter described in Section 11.1, a brief program can be written to display average scan time (See Figure 12-1). For

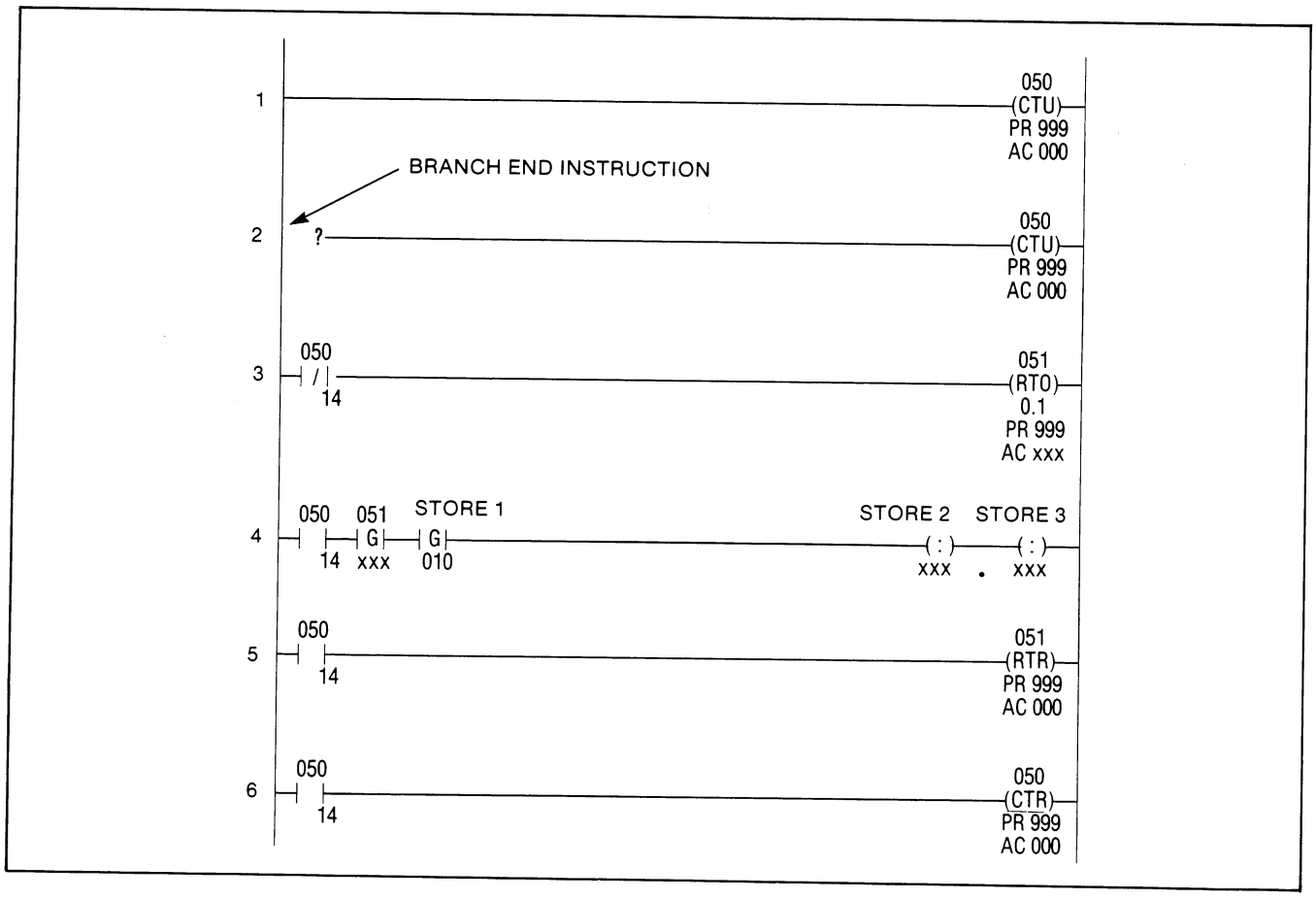


FIGURE 12-1 — Scan Time Program

example, the time required to complete 1000 scans can be monitored using a timer and GET/PUT transfer instructions. With a time base of 0.1 seconds, the average scan time can be read directly in milliseconds.

The scan time is displayed as the timer Accumulated value beneath the GET or PUT instruction.

NOTE: The displayed average scan time includes both the I/O scan time of approximately 1 msec and the User Program scan time.

WARNING: The lower limit of input device cycle time should not be less than the scan time of the Processor. If so, incorrect input data could be used during program execution. This could cause unpredictable machine operation and possibly cause damage to equipment and/or injury to personnel.

Critical inputs can be monitored and critical outputs can be controlled in an accelerated manner using the I/O Update Instructions described in Section 7.2.

12.2 WATCHDOG TIMER

The Processor contains an internal watchdog timer that sets an upper limit for the scan time. If the scan time should exceed the limit, a Processor fault would occur and the Processor would shut down. The watchdog timer is factory set at 180 MS.

12.3 INSTRUCTION EXECUTION TIMES

Execution times in microseconds for each instruction are presented in Table 12-1. Two execution times are given: when the instruction is FALSE, and an average execution time when the instruction is TRUE. For some instructions, the execution time will depend on other circumstances described below.

12.3.1 MULTIPLY and DIVIDE Instructions (1772-LN3 Processor)

The execution times for these instructions depend on the values of the numbers being computed. Generally, the larger the number, the longer the execution time.

12.3.2 Instructions Within a ZCL Zone

In addition to the ZCL instruction itself, when the rung containing the ZCL instruction is FALSE (i.e., outputs are controlled by the zone), the time required to scan each instruction between the Start Fence and End Fence is 15 microseconds.

12.4 BLOCK TRANSFER PROGRAMMING

In addition to the time required to execute the instruction in the program scan, the I/O scan is also interrupted while data is transferred. The delay is proportional to the number of words transferred and may vary from one kind of module to another. Consult the I/O Module User's Manual.

Table 12-1 Instruction Execution Times

	Approximate Execution Time per scan (in microseconds)	
Instruction Name	**Instruction FALSE**	**Instruction TRUE (avg)**
EXAMINE ON, EXAMINE OFF	17	17
OUTPUT ENERGIZE	28	28
OUTPUT LATCH, OUTPUT UNLATCH	22	27
GET	—	28
PUT	22	36
EQUAL	36	36
LESS THAN	41	41
GET BYTE	—	15
LIMIT TEST	28	28
COUNTER RESET	23	35
RETENTIVE TIMER RESET	23	43
TIMER ON-DELAY	45	80
RETENTIVE TIMER ON-DELAY	46	81
TIMER-OFF DELAY	60	102
UP COUNTER	63	105
DOWN COUNTER	82	100
ADD	22	56
SUBTRACT	26	77 for (+) or 100 for (-)
MULTIPLY	35	230-520 (See 12.3.1)
DIVIDE	35	162-775 (See 12.3.1)
MASTER CONTROL RESET	31	31
ZONE CONTROL LAST STATE	45 (See 12.3.2)	35
BRANCH START	24	24
BRANCH END	26	26
IMMEDIATE INPUT	92	92
IMMEDIATE OUTPUT	22	120

Section 13
NUMBERING SYSTEMS

13.0 GENERAL

There are four numbering systems used with Allen-Bradley programmable controllers. They are:

- Decimal
- Octal
- Binary
- Hexadecimal

These numbering systems differ by their number sets and place values.

13.1 DECIMAL NUMBERING SYSTEM

The decimal numbering system uses a number set made up of ten digits: the numbers 0 through 9. All decimal numbers are composed of these digits. The value of a decimal number depends on the digits used and the place value of each digit.

Each place value in a decimal number represents a power of ten (Figure 13-1), starting with 10^0. (Any number raised to the zero power is equal to 1) The value of a decimal number is determined by multiplying each digit by its corresponding place value and adding these numbers together.

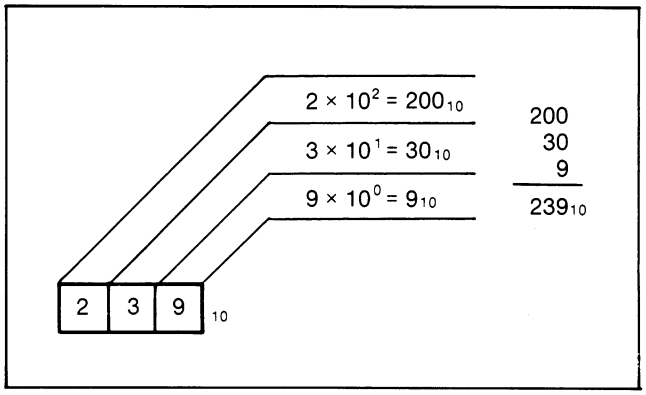


FIGURE 13-1 — Decimal Numbering System

13.2 OCTAL NUMBERING SYSTEM

The octal numbering system is used to address word and bit locations in the Data Table. Its number set is composed of eight digits: the numbers 0 through 7.

Each digit in an octal number has a certain place value, represented by a power of eight (Figure 13-2).

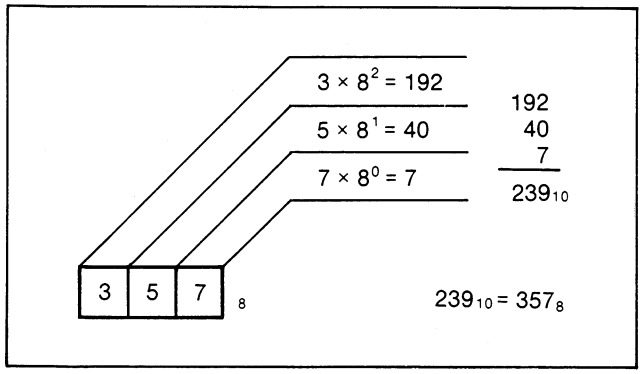


FIGURE 13-2 — Octal Numbering System

The decimal value of an octal number is computed by multiplying each octal digit by its place value and adding these numbers together.

13.3 BINARY NUMBERING SYSTEM

The binary numbering system is a number set that consists of two digits: the numbers 0 and 1. All information in memory is stored as an arrangement of 1's and 0's.

Each digit in a binary number has a certain place value expressed as a power of two (Figure 13-3). The decimal equivalent of a binary number is computed by multiplying each binary digit by its corresponding place value and adding these numbers together.

By grouping several binary digits together, values can be formed to represent decimal, hexadecimal or octal numbers.

13.3.1 Binary Coded Decimal

Binary Coded Decimal (BCD) uses an arrangement of 12 binary digits to represent a 3-digit decimal number from 000 to 999 (Figure 13-4). Each group of 4 binary digits is used to represent a decimal number from 0 to 9. The

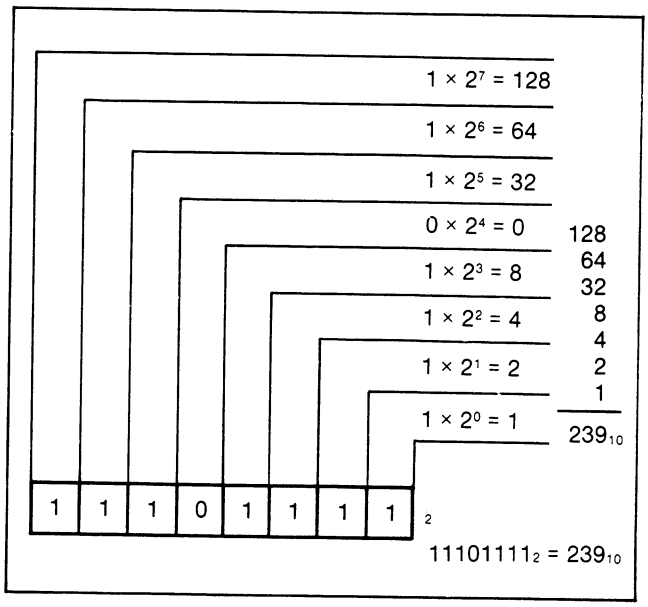


FIGURE 13-3 — Binary Numbering System

place values for each group of 4 digits are 20, 21, 22 and 23 (Table 13-1).

The decimal equivalent for a group of 4 binary digits is determined by multiplying the binary digit by its corresponding place value and adding these numbers.

13.3.2 Octal Representation

Octal Representation is an arrangement of 8 bits (one byte) to represent a 3-digit octal number from 000_8 to 377_8 (Figure 13-5). The 8 bits are separated into three groups: 2 bits, 3 bits and 3 bits.

TABLE 13-1 — BCD Representation

PLACE VALUE				DECIMAL EQUIVALENT
2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

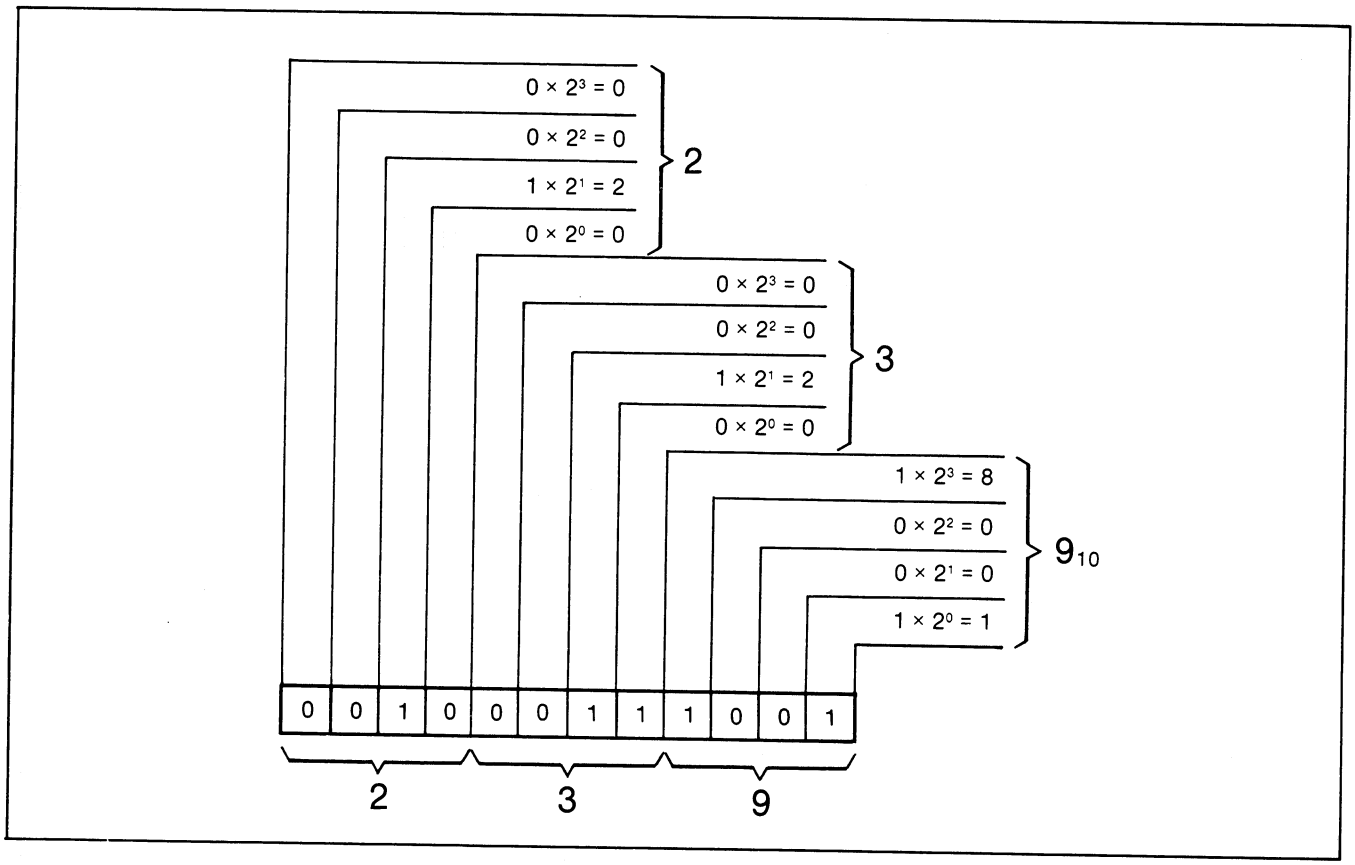


FIGURE 13-4 — Binary Coded Decimal

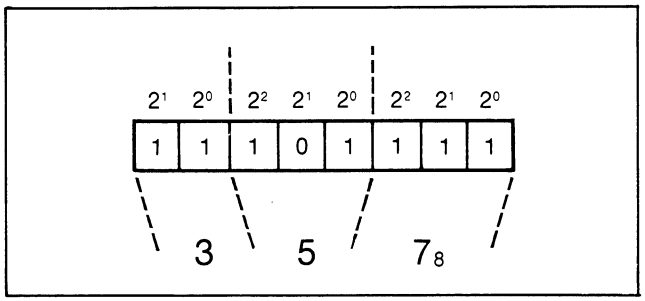


FIGURE 13-5 — Octal Representation

The octal number for each group of bits is determined by multiplying the binary digit by its corresponding place value and adding these numbers together. (Table 13-2).

TABLE 13-2 — Octal Representation

PLACE VALUE			OCTAL EQUIVALENT
2^2 (4)	2^1 (2)	2^0 (1)	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

13.4 HEXADECIMAL NUMBERING SYSTEM

The hexadecimal numbering system has a number set of 16 digits: the numbers 0-9 and the letters A-F (Table 13-3). The letters A-F represent the decimal numbers 10-15 respectively.

A hexadecimal number can be converted to a decimal number by multiplying the hexadecimal digit by its corresponding place value (Figure 13-6).

FIGURE 13-3 — Numbering System Conversion Chart

HEXA-DECIMAL	BINARY	DECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

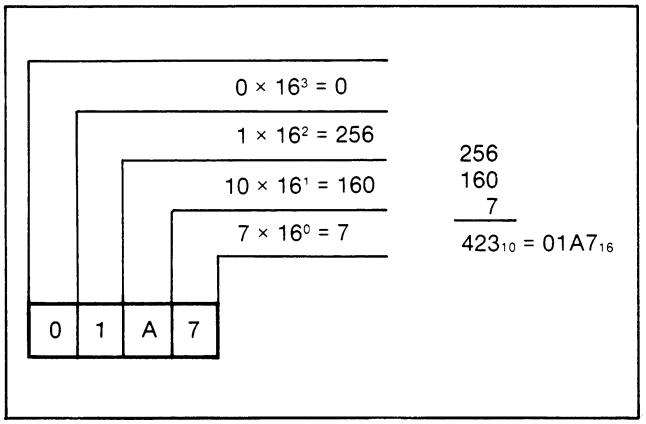


FIGURE 13-6 — Example of Hexadecimal to Decimal Conversion

Because each hexadecimal digit represents 4 binary digits, it is easy to convert a hexadecimal number to a binary number. This is done by writing out the 4-bit binary pattern for each hexadecimal digit (Figure 13-7).

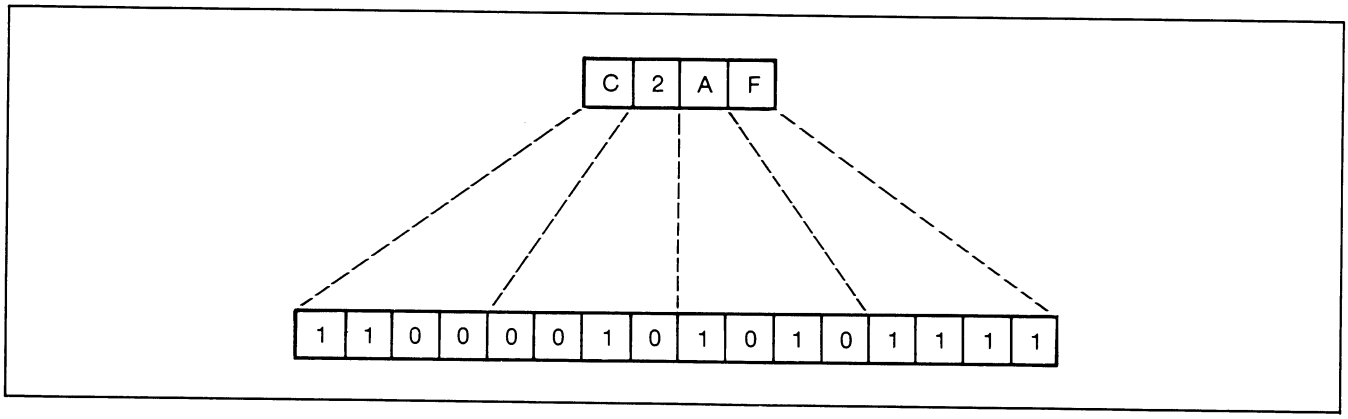


FIGURE 13-7 — Example of Hexadecimal to Binary Conversion

INDEX

ALLEN-BRADLEY
A Rockwell International Company

Industrial Computer Group — PC Division
747 Alpha Drive, Cleveland, Ohio 44143

Publication 1772-6.8.4 — December, 1983
Supersedes Publication 1772-821 — December, 1983

P/N 955094-60