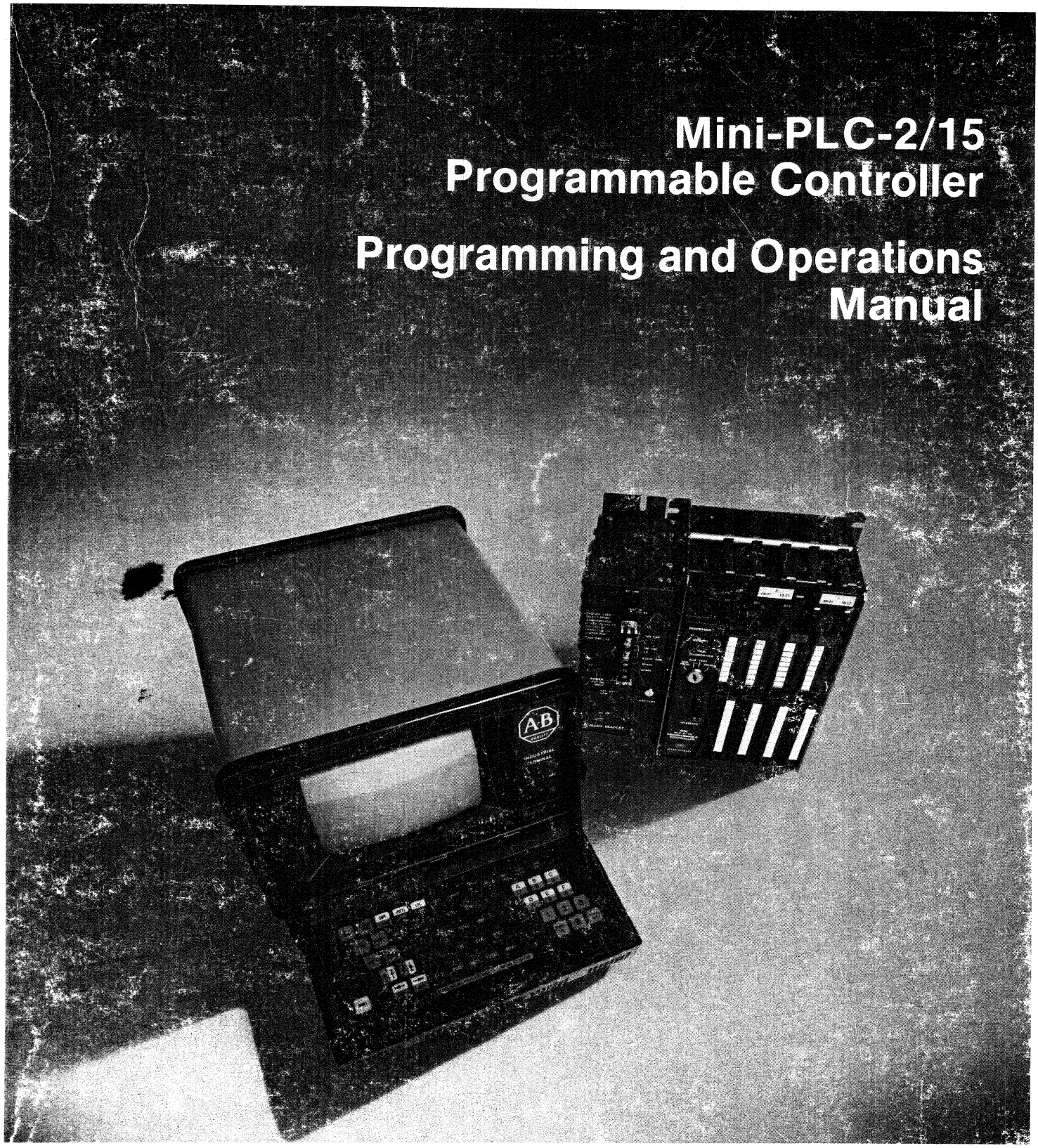
ALLEN-BRADLEY

1772-804

Mini-PLC-2/15 Programmable Controller

Programming and Operations Manual



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.

Reproduction of the content of this manual, in whole or in part, without written permission of the Allen-Bradley Company is prohibited.

© 1982 Allen-Bradley Company

TABLE OF CONTENTS

Section/ Paragraph	Title	Page
1	**INTRODUCTION**	
1.0	Overview	1-1
1.1	Ladder Diagram Logic	1-2
1.2	Memory Structure	1-3
1.3	Memory Organization	1-3
1.4	Hardware/Program Interface	1-7
1.4.1	Image Table	1-7
1.4.2	Instruction Address	1-7
1.4.3	Fundamental Operation	1-7
1.5	Compatibility	1-10
1.5.1	Industrial Terminal Compatibility	1-10
1.5.2	Data Highway Compatibility	1-10
1.5.3	Manual Applicability Using the 1770-T1 or -T2 Industrial Terminal	1-11
1.6	Support Documentation	1-11
2	**HARDWARE CONSIDERATIONS**	
2.0	General	2-1
2.1	Mode Select Switch	2-1
2.1.1	PROG Position	2-1
2.1.2	TEST Position	2-1
2.1.3	RUN Position	2-1
2.1.4	RUN/PROG Position	2-1
2.2	User EPROM System	2-2
2.2.1	Memory Write Protect	2-2
2.3	Run-Time Errors	2-2
2.4	Processor Diagnostic Indicators	2-3
2.4.1	MEMORY Indicator	2-3
2.4.2	PROCESSOR Indicator	2-3
2.4.3	RUN Indicator	2-4
2.5	Power Supply Diagnostic Indicators	2-4
2.5.1	BATTERY LOW Indicator	2-4
2.5.2	DC ON Indicator	2-4
2.6	Switch Group Assembly	2-4
2.7	Industrial Terminal	2-4
2.7.1	Hook-Up	2-5
2.7.2	Mode Selection and Initialization of the Industrial Terminal	2-5
2.7.3	Keytop Overlay	2-7
2.8	Mini Processor Transport Cable	2-7
3	**RELAY-TYPE INSTRUCTIONS**	
3.0	General	3-1
3.1	Examine Instructions	3-1
3.2	Output Instructions	3-2
3.2.1	OUTPUT ENERGIZE Instruction	3-2
3.2.2	OUTPUT LATCH and UNLATCH Instructions	3-3
3.3	Branch Instructions	3-4
3.4	Programming Relay-Type Instructions	3-5

Section/ Paragraph	Title	Page
4	**TIMER AND COUNTER INSTRUCTIONS**	
4.0	General	4-1
4.1	Timer Instructions	4-1
4.1.1	TIMER ON-DELAY Instruction	4-2
4.1.2	TIMER OFF-DELAY Instruction	4-2
4.1.3	RETENTIVE TIMER Instruction	4-3
4.1.4	RETENTIVE TIMER RESET Instruction	4-4
4.1.5	Timer Accuracy for 10 ms Timers	4-4
4.2	Counter Instructions	4-6
4.2.1	UP-COUNTER Instruction	4-6
4.2.2	COUNTER RESET Instruction	4-7
4.2.3	DOWN-COUNTER Instruction	4-8
4.3	Cascading Timers or Counters	4-9
4.4	Programming Timer and Counter Instruction	4-9
5	**DATA MANIPULATION INSTRUCTIONS**	
5.0	General	5-1
5.1	Data Transfer Instructions	5-2
5.1.1	GET Instruction	5-2
5.1.2	PUT Instruction	5-2
5.2	Data Comparison Instructions	5-2
5.2.1	LES and EQU Instructions	5-3
5.2.2	GET BYTE and LIMIT TEST Instructions	5-4
5.3	Programming Data Manipulation Instructions	5-5
6	**ARITHMETIC INSTRUCTIONS**	
6.0	General	6-1
6.1	ADD Instruction	6-1
6.2	SUBTRACT Instruction	6-2
6.3	MULTIPLY Instruction	6-2
6.4	DIVIDE Instruction	6-2
6.5	Programming Arithmetic Instructions	6-3
7	**OUTPUT OVERRIDE AND I/O UPDATE INSTRUCTIONS**	
7.0	General	7-1
7.1	Output Override Instructions	7-1
7.2	I/O Update Instructions	7-1
7.2.1	Scan Sequence	7-2
7.2.2	IMMEDIATE INPUT Instruction	7-3
7.2.3	IMMEDIATE OUTPUT Instruction	7-3
7.3	Programming Output Override and I/O Update Instructions	7-4
8	**JUMP INSTRUCTIONS AND SUBROUTINE PROGRAMMING**	
8.0	General	8-1
8.1	Subroutine Area	8-1
8.2	LABEL Instruction	8-1
8.2.1	LABEL Instruction Characteristics	8-3
8.2.2	Causes of Run-Time Errors	8-3
8.3	JUMP Instruction	8-3
8.3.1	JUMP Instruction Characteristics	8-3

Section/ Paragraph	Title	Page
8.3.2	Causes of Run-Time Errors	8-3
8.4	JUMP TO SUBROUTINE Instruction	8-3
8.4.1	JUMP TO SUBROUTINE Instruction Characteristics	8-4
8.4.2	Causes of Run-Time Errors	8-5
8.5	RETURN Instruction	8-5
8.5.1	RETURN Instruction Characteristics	8-5
8.5.2	Causes of Run-Time Errors	8-5
8.6	Multiple JUMPS to the Same LABEL Instruction	8-5
8.7	Subroutine Programming Considerations	8-5
8.8	Programming Jump/Subroutine Instructions	8-6
9	**FILES AND DATA MONITOR MODE**	
9.0	General	9-1
9.1	File Concepts	9-1
9.1.1	File Definition	9-1
9.1.2	File Planning	9-2
9.1.3	File Instructions	9-2
9.1.4	Programming File Instructions	9-9
9.1.5	File Instruction Run-Time Errors	9-9
9.2	Data Monitor Mode	9-9
9.2.1	Accessing the Data Monitor Mode	9-9
9.2.2	Data Monitor Display	9-9
9.2.3	Cursor Controls	9-11
9.2.4	Data Monitoring Procedures	9-12
9.2.5	Entering and Changing Data	9-12
10	**DATA TRANSFER FILE INSTRUCTIONS**	
10.0	General	10-1
10.1	FILE-TO-FILE MOVE Instruction	10-1
10.1.1	Instruction Operation	10-1
10.1.2	FILE-TO-FILE MOVE Instruction Characteristics	10-2
10.1.3	Programming the FILE-TO-FILE MOVE Instruction	10-2
10.2	WORD-TO-FILE MOVE and FILE-TO-WORD MOVE Instructions	10-3
10.2.1	Instruction Operation	10-3
10.2.2	WORD-TO-FILE MOVE and FILE TO WORD MOVE Instruction Characteristics	10-4
10.2.3	Programming the WORD-TO-FILE MOVE and FILE-TO-WORD MOVE Instructions	10-4
11	**SEQUENCER INSTRUCTIONS**	
11.0	General	11-1
11.1	SEQUENCER OUTPUT Instruction	11-2
11.1.1	Sequencer Output Analogy	11-2
11.1.2	Instruction Operation	11-2
11.1.3	Masking Output Data	11-3
11.1.4	SEQUENCER OUTPUT Instruction Characteristics	11-4
11.1.5	Programming the SEQUENCER OUTPUT Instruction	11-4
11.2	SEQUENCER INPUT Instruction	11-6
11.2.1	Instruction Operation	11-6
11.2.2	Masking	11-7

Section/ Paragraph	Title	Page
11.2.3	SEQUENCER INPUT Instruction Characteristics	11-7
11.2.4	Programming the SEQUENCER INPUT Instruction	11-8
11.3	SEQUENCER LOAD Instruction	11-9
11.3.1	Instruction Operation	11-9
11.3.2	SEQUENCER LOAD Instruction Characteristics	11-10
11.3.3	Programming the SEQUENCER LOAD Instruction	11-10
12	**WRITING THE USER PROGRAM**	
12.0	General	12-1
12.1	Developing the Program	12-1
12.2	Sample Program	12-1
12.3	Developing the Data Table	12-5
12.3.1	Data Table Documentation Forms	12-5
12.3.2	I/O Assignment Considerations	12-7
12.3.3	Timer/Counter Assignment Considerations	12-8
12.3.4	Bit/Word Storage Considerations	12-9
12.3.5	File Storage Considerations	12-9
12.3.6	Sequencer Table Storage Considerations	12-9
12.4	Sizing the Data Table	12-9
12.5	Program Recommendations	12-10
12.6	Current Record	12-11
13	**OPERATING INSTRUCTIONS**	
13.0	General	13-1
13.1	Data Table Configuration	13-1
13.1.1	Memory Layout Display	13-1
13.2	Addressing	13-2
13.3	Help Directories	13-2
13.4	Search Functions	13-2
13.4.1	Search for First and Last Rung and User Program Boundaries	13-2
13.4.2	Search for First or Last Instruction in a Rung	13-3
13.4.3	Search for Incomplete Rung	13-3
13.4.4	Search for Specific Instruction and Specific Word Address	13-3
13.4.5	Single Rung Display	13-3
13.4.6	Remote Mode Select	13-3
13.5	Editing	13-3
13.5.1	Inserting an Instruction	13-4
13.5.2	Removing an Instruction	13-4
13.5.3	Inserting a Rung	13-4
13.5.4	Removing a Rung	13-4
13.5.5	Changing Data of a Word or Block Instruction	13-4
13.5.6	Changing the Address of a Word or Block Instruction	13-5
13.5.7	Replacing an Instruction or Changing the Address of an Instruction without Data	13-5
13.5.8	On-Line Data Change	13-5
13.6	On-Line Programming	13-5
13.6.1	Data Initialization Key	13-6
13.6.2	On-Line Programming Procedures	13-6
13.6.3	General Rules	13-9

Section/ Paragraph	Title	Page
13.7	Troubleshooting Aids	13-9
13.7.1	Bit Manipulation and Monitor	13-9
13.7.2	Force ON and Force OFF Functions	13-10
13.7.3	Forced Address Display	13-11
13.7.4	TEMPORARY END Instruction	13-11
13.7.5	ERR Message for an Illegal Opcode	13-11
13.8	Clearing Memory	13-12
13.8.1	Data Table Clear	13-12
13.8.2	User Program Clear	13-12
13.8.3	Partial Memory Clear	13-12
13.8.4	Total Memory Clear	13-12
14	**PERIPHERAL FUNCTIONS**	
14.0	General	14-1
14.1	Baud Rate Setting	14-1
14.2	Contact Histogram	14-1
14.3	Report Generation	14-3
14.3.1	Report Generation Commands	14-3
14.3.2	Automatic Report Generation	14-8
14.4	Digital Cassette Recorder	14-11
14.4.1	Dumping Memory to Cassette Tape	14-11
14.4.2	Loading Memory From Cassette Tape	14-11
14.4.3	Automatic Verification	14-12
14.4.4	Program Verification	14-12
14.4.5	Displaying and Locating Errors	14-12
14.5	Data Cartridge Recorder	14-12
14.5.1	Dumping Memory to Data Cartridge Tape	14-13
14.5.2	Loading Memory from Data Cartridge Tape	14-13
14.5.3	Data Cartridge Verification	14-13
14.6	Ladder Diagram Dump	14-14
14.7	Total Memory Dump	14-14
15	**SPECIAL PROGRAMMING TECHNIQUES**	
15.0	General	15-1
15.1	Scan Counter	15-1
15.2	Block Transfer	15-1
15.2.1	Introduction	15-1
15.2.2	Basic Operation	15-2
15.2.3	BLOCK TRANSFER Instructions	15-3
15.2.4	BLOCK TRANSFER READ or WRITE Instruction Characteristics	15-5
15.2.5	Causes of Run-Time Errors	15-5
15.2.6	Programming BLOCK TRANSFER READ and WRITE Instructions	15-5
15.2.7	Multiple Reads of Different Block Lengths from One Module	15-5
15.2.8	Defining the Block Transfer Data Address Area	15-8
15.2.9	Buffering Data	15-8
15.2.10	Bidirectional Block Transfer	15-9
15.3	One Shot	15-14

Section/ Paragraph	Title	Page
15.3.1	Leading Edge One-Shot	15-14
15.3.2	Trailing Edge One-Shot	15-14
15.4	Manual Restart	15-15
16	**SCAN TIME AND INSTRUCTION EXECUTION TIME**	
16.0	General	16-1
16.1	Scan Time	16-1
16.1.1	Determination of Average Scan Time	16-1
16.2	Watchdog Timer	16-1
16.3	Instruction Execution Times	16-2
16.3.1	MULTIPLY and DIVIDE Instructions	16-2
16.3.2	Addresses Above 3778	16-2
16.3.3	Instructions within a ZCL Zone	16-2
16.3.4	BLOCK TRANSFER Instructions	16-4
16.3.5	SEQUENCER Instructions	16-4
16.3.6	FILE-TO-FILE MOVE Instruction	16-4
17	**NUMBERING SYSTEMS**	
17.0	General	17-1
17.1	Decimal Numbering System	17-1
17.2	Octal Numbering System	17-1
17.3	Binary Numbering System	17-1
17.3.1	Binary Coded Decimal	17-2
17.3.2	Octal Representation	17-3
17.4	Hexadecimal Numbering System	17-3

LIST OF FIGURES

Figure	Title	Page
1-1	Mini-PLC-2/15 Programmable Controller	1-1
1-2	Relay Diagram	1-2
1-3	Ladder Diagram Rung	1-2
1-4	Memory Word Structure	1-3
1-5	User Memory Representation	1-4
1-6	Data Table Organization, Factory Configured	1-5
1-7	Data Table Organization, Field Expandable	1-6
1-8	Instruction Address Terminology	1-8
1-9	Bit Address to Hardware Relationship	1-8
1-10	Hardware-Program Interface	1-9
1-11	ERR Message for Invalid Display of Processor Memory	1-10
2-1	Mode Select Keyswitch	2-1
2-2	Location of EPROM and Access Door	2-2
2-3	Diagnostic Indicators	2-3
2-4	System Power Supply (Without Battery Pack)	2-4
2-5	Switch Group Assembly	2-5
2-6	Mini-PLC-2/15 Connection Diagram	2-5
2-7	PLC-2 Family Keytop Overlay (Cat. No. 1770-KCB)	2-6
2-8	Mode Selection Display	2-6

Figure	Title	Page
3-1	EXAMINE ON Instruction	3-1
3-2	EXAMINE OFF Instruction	3-1
3-3	Ladder-Diagram Rung	3-2
3-4	OUTPUT ENERGIZE Instruction	3-2
3-5	Unconditional OUTPUT ENERGIZE Instruction	3-3
3-6	LATCH/UNLATCH Instructions	3-3
3-7	LATCH/UNLATCH Timing Diagram	3-4
3-8	LATCH/UNLATCH Indication	3-4
3-9	Branching Instructions	3-5
3-10	Nested Branching vs. Proper Programming	3-5
4-1	BCD Format	4-1
4-2	Timer Accumulated Value Word	4-2
4-3	TIMER ON-DELAY Timing Diagram & Programming	4-3
4-4	TIMER OFF-DELAY Timing Diagram & Programming	4-4
4-5	RETENTIVE TIMER with RESET Timing Diagram & Programming	4-5
4-6	Counter Accumulated Value Word	4-6
4-7	UP-COUNTER Diagram & Programming	4-7
4-8	UP-COUNTER with RESET Diagram & Programming	4-8
4-9	UP/DOWN COUNTER Example	4-9
4-10	Cascading Counters Example	4-10
5-1	BCD Word Format	5-1
5-2	Octal Representation	5-1
5-3	GET and PUT Instructions	5-2
5-4	Changing a Counter Preset	5-3
5-5	LESS THAN Comparison	5-3
5-6	GREATER THAN Comparison	5-4
5-7	EQUAL TO Comparison	5-4
5-8	LESS THAN or EQUAL TO Comparison	5-5
5-9	GREATER THAN or EQUAL TO Comparison	5-5
5-10	GET BYTE/LIMIT TEST Comparison	5-6
6-1	Arithmetic Instruction Word	6-1
6-2	ADD Instruction	6-2
6-3	SUBTRACT Instruction	6-2
6-4	MULTIPLY Instruction	6-3
6-5	DIVIDE Instruction	6-3
7-1	MCR and ZCL Zone Programming	7-2
7-2	Scan Sequence	7-3
7-3	IMMEDIATE INPUT Instruction	7-4
7-4	IMMEDIATE OUTPUT Instruction	7-5
8-1	Representative Subroutine Area	8-2
8-2	LABEL Format	8-2
8-3	JUMP Format	8-2
8-4	JUMP to LABEL Operation	8-4
8-5	JUMP TO SUBROUTINE Format	8-4
8-6	RETURN Format	8-5
8-7	JUMP TO SUBROUTINE LABEL Operation	8-6
8-8	Multiple JUMPS to LABEL in User Program	8-7

Figure	Title	Page
8-9	Multiple JUMPS to LABEL in Subroutine Area	8-7
8-10	Multiple JUMPS to LABEL and Multiple Return Paths	8-8
8-11	Subroutine Programming	8-9
9-1	File Structure	9-1
9-2	File Instruction Format	9-3
9-3	Example of an Externally Indexed File Instruction	9-3
9-4	Example of an Internally Indexed File Instruction	9-4
9-5	Complete Mode Operation	9-5
9-6	Status Bits for Complete Mode	9-5
9-7	Distributed Complete Mode Operation	9-6
9-8	Status Bits for Distriuted Complete Mode	9-7
9-9	Incremental Mode Operation	9-8
9-10	Status Bits for Incremental Mode	9-8
9-11	Example Hexadecimal Data Monitor Display of a File Instruction	9-10
10-1	FILE-TO-FILE MOVE Operation	10-1
10-2	FILE-TO-FILE MOVE Format	10-2
10-3	FILE-TO-FILE MOVE Example Rung	10-3
10-4	WORD-TO-FILE MOVE Operation	10-3
10-5	FILE-TO-WORD MOVE Operation	10-4
10-6	WORD-TO-FILE MOVE Format	10-5
10-7	FILE-TO-WORD MOVE Format	10-5
10-8	WORD-TO-FILE MOVE Example Rung	10-6
11-1	Sequencer Table	11-1
11-2	Sequencer Table Format in the Data Table	11-2
11-3	Sequencer Output Analogy	11-3
11-4	Masking Transfer Data	11-4
11-5	SEQUENCER OUTPUT Format	11-5
11-6	SEQUENCER OUTPUT Example Rung	11-6
11-7	Control of Sequencer Outputs	11-6
11-8	Example Binary Data Monitor Display	11-7
11-9	SEQUENCER INPUT Format	11-8
11-10	SEQUENCER INPUT Example Rung	11-9
11-11	SEQUENCER LOAD Format	11-11
11-12	SEQUENCER LOAD Example Rung	11-11
12-1	Conveyor Belt Example	12-1
12-2	Ladder Diagram Progam	12-3
12-3	Example of a Data Table Word Map	12-6
12-4	Example of a Data Table Map	12-6
12-5	Example of Data Table Word Assignments	12-7
12-6	Example of Data Table Bit Assignments	12-7
12-7	Example of Sequencer Table Bit Assignments	12-8
12-8	Storage Bit Example	12-10
14-1	Contact Histogram Display	14-2
14-2	Alphanumeric Keytop Overlays	14-3
14-3	Example Graphic/Alphanumeric Message	14-8
14-4	Parity Switch Location	14-9
14-5	Bit Address-Message Number Relationship	14-10

Figure	Title	Page
14-6	Message Request Bit-Done Bit Relationship	14-10
14-7	Example Program to Request a Message	14-11
14-8	Data Table Printout in Hexadecimal	14-14
15-1	Scan Counter	15-1
15-2	Module Position-Image Table Byte Relationship	15-2
15-3	Block Transfer Request	15-2
15-4	Block Transfer Format	15-4
15-5	Data Table Locations for a BLOCK TRANSFER READ	15-6
15-6	Programming Multiple Reads from One Module	15-7
15-7	Defining the Data Address Area	15-8
15-8	Buffering Data	15-10
15-9	Data Table Locations for Bidirectional Block Transfer	15-12
15-10	Leading Edge One-Shot	15-14
15-11	Trailing Edge One-Shot	15-15
15-12	Manual Restart Example Programming	15-16
16-1	Scan Time Program	16-2
17-1	Decimal Numbering System	17-1
17-2	Octal Numbering System	17-1
17-3	Binary Numbering System	17-2
17-4	Binary Coded Decimal	17-2
17-5	Octal Representation	17-3
17-6	Example of Hexadecimal to Decimal Conversion	17-4
17-7	Example of Hexadecimal to Binary Conversion	17-4

LIST OF TABLES

Table	Title	Page
3-1	Relay-Type Instructions	3-7
4-1	Timer Instructions	4-11
4-2	Counter Instructions	4-12
5-1	Data Manipulation Instructions	5-7
6-1	Arithmetic Instructions	6-4
7-1	Output Override and I/O Update Instructions	7-6
8-1	Jump/Subroutine Instructions	8-10
9-1	Modes of Instruction Operation	9-4
9-2	Accessing the Data Monitor Mode	9-10
9-3	Field Cursor Commands and Scrolling	9-11
9-4	Digit Cursor Commands	9-12
9-5	Paging and Specified Paging Commands	9-12
9-6	Data Entry Commands	9-13
10-1	File Instructions	10-7
11-1	Sequencer Instructions	11-12
12-1	Data Table Addresses for Hardwired Devices	12-2
13-1	Data Table Configuration	13-12
13-2	Help Directories	13-13
13-3	Search Functions	13-14

Table	Title	Page
13-4	Editing Functions	13-15
13-5	Troubleshooting Aids	13-16
13-6	Clear Memory Functions	13-17
14-1	Switch Group Settings	14-1
14-2	Key Sequence for Setting Baud Rate	14-1
14-3	Contact Histogram Functions	14-2
14-4	Report Generation Commands	14-4
14-5	Address Delimiters	14-5
14-6	Alphanumeric/Graphic Keytop Definitions	14-6
14-7	Industrial Terminal Control Codes	14-6
14-8	ASCII Control Codes	14-7
14-9	Example Message Control Word-Message Number Relationship	14-9
15-1	Timer/Counter-Block Transfer Analogy	15-4
15-2	Block Transfer Instructions	15-13
16-1	Instruction Execution Times	16-3
17-1	BCD Representation	17-3
17-2	Octal Representation	17-3
17-3	Numbering System Conversion Chart	17-3

Section 1
INTRODUCTION

1.0 OVERVIEW

The Bulletin 1772 Mini-PLC-2/15 Programmable Controller is a rugged, solid state programmable controller that consists of the Mini-PLC-2/15 Processor (Cat. No. 1772-LV) and the 1771 I/O Family of racks and modules. Refer to Figure 1-1.

With a user-written program and appropriate I/O modules, the Mini-PLC-2/15 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-2/15 Processor has a Read/Write memory that stores User Program instructions, numeric values and I/O device status. It also has an optional EPROM back-up memory feature. This semi-permanent memory automatically reprograms the RAM memory when the Processor is restarted after a power failure (battery back-up lost as well) or if a memory parity error should cause a Processor fault.

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 and functional block format display from the keyboard of the Industrial Terminal (Cat. No. 1770-T3). Some ladder diagram symbols closely resemble the relay symbols used in hardwired relay control systems. The functional block displays are an easy method of programming and monitoring complex instructions.

During program operation, the Mini-PLC-2/15 Processor continuously monitors the status of

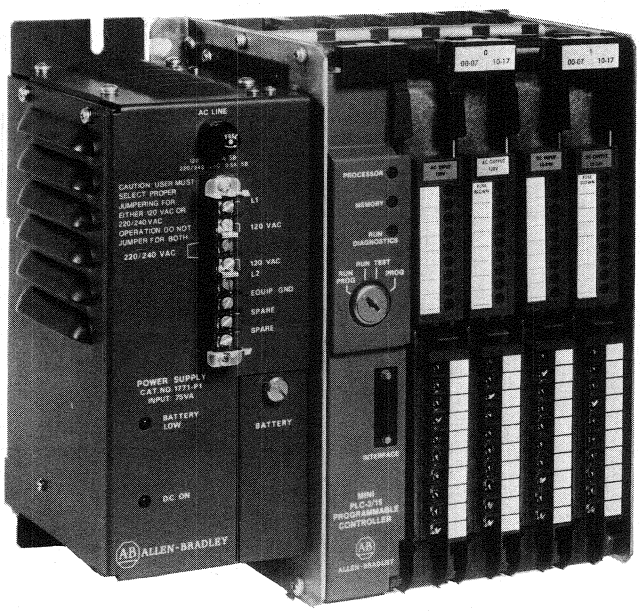


FIGURE 1-1 — Mini-PLC-2/15 Programmable Controller

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-2/15 Controller can perform additional functions such as:

- Timing/Counting operations
- Arithmetic (+, –, x, ÷) operations
- Word and File moves
- Data comparisons
- Block Transfers
- Jump/Subroutine operations
- Sequencer operations
- Complete I/O Forcing
- On-Line programming
- Data Highway and modem capability

When the Industrial Terminal (Cat. No. 1770-T3) and the Mini-PLC-2/15 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 two 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.

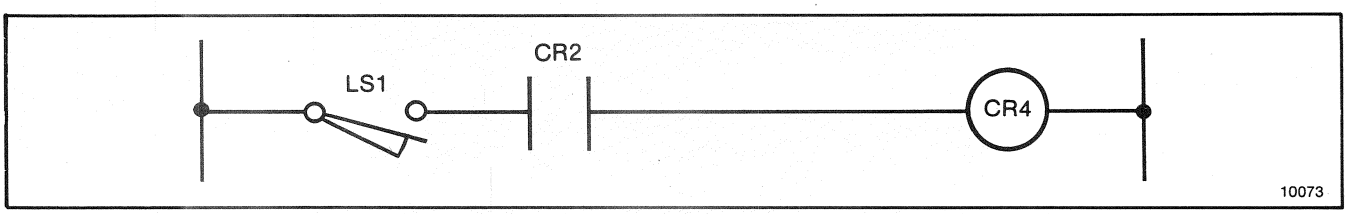


FIGURE 1-2 — Relay Diagram

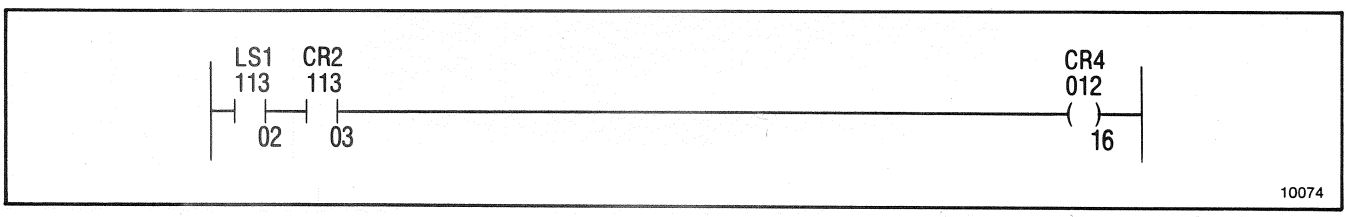


FIGURE 1-3 — Ladder Diagram Rung

1.2 MEMORY STRUCTURE

The Data Table of the Mini-PLC-2/15 Processor 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: either a 3 or 4-digit number using the octal numbering system.

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 (MSB) 17 at the left and the least significant bit (LSB) 00 at the right.

A specific Data Table bit can be identified by combining the word address and bit number to form the bit address, such as 030/12 or 1701/16. The bit address of an instruction is shown by writing the word address above the instruction and the bit number below it.

1.3 MEMORY ORGANIZATION

The Mini-PLC-2/15 Processor (Cat. No. 1772-LV) has a memory capacity of 2048 (2K) memory words. These memory words are organized by their word address and are divided into the following major areas shown in Figure 1-5. The size of each can be varied to suit user needs.

- Data Table
- User Program
  - Main Program
  - Subroutine Area (if used)
- 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 and file data, 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. It can be field expanded to 1920 words or decreased to 48 words through the Industrial Terminal keyboard. Figures 1-6 and 1-7 show the organization of factory configured and field expandable Data Tables, respectively.

The User Program is the logic that controls the machine operation. The logic consists of instructions that are programmed in ladder diagram format. Most instructions require 1

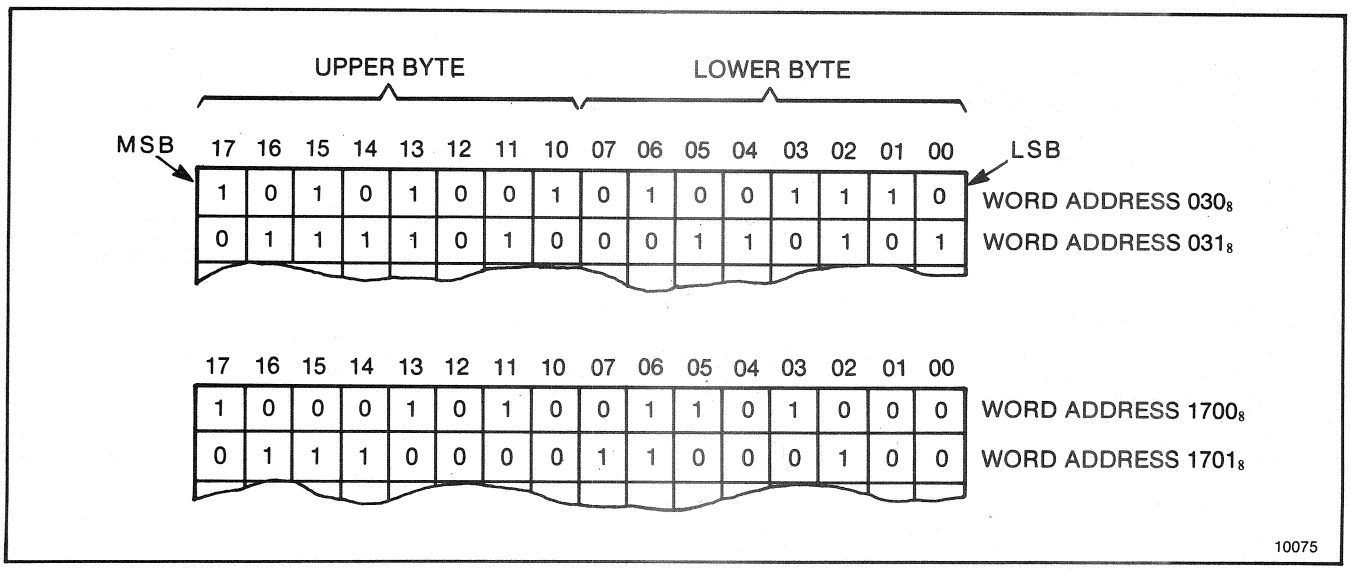


FIGURE 1-4 — Memory Word Structure

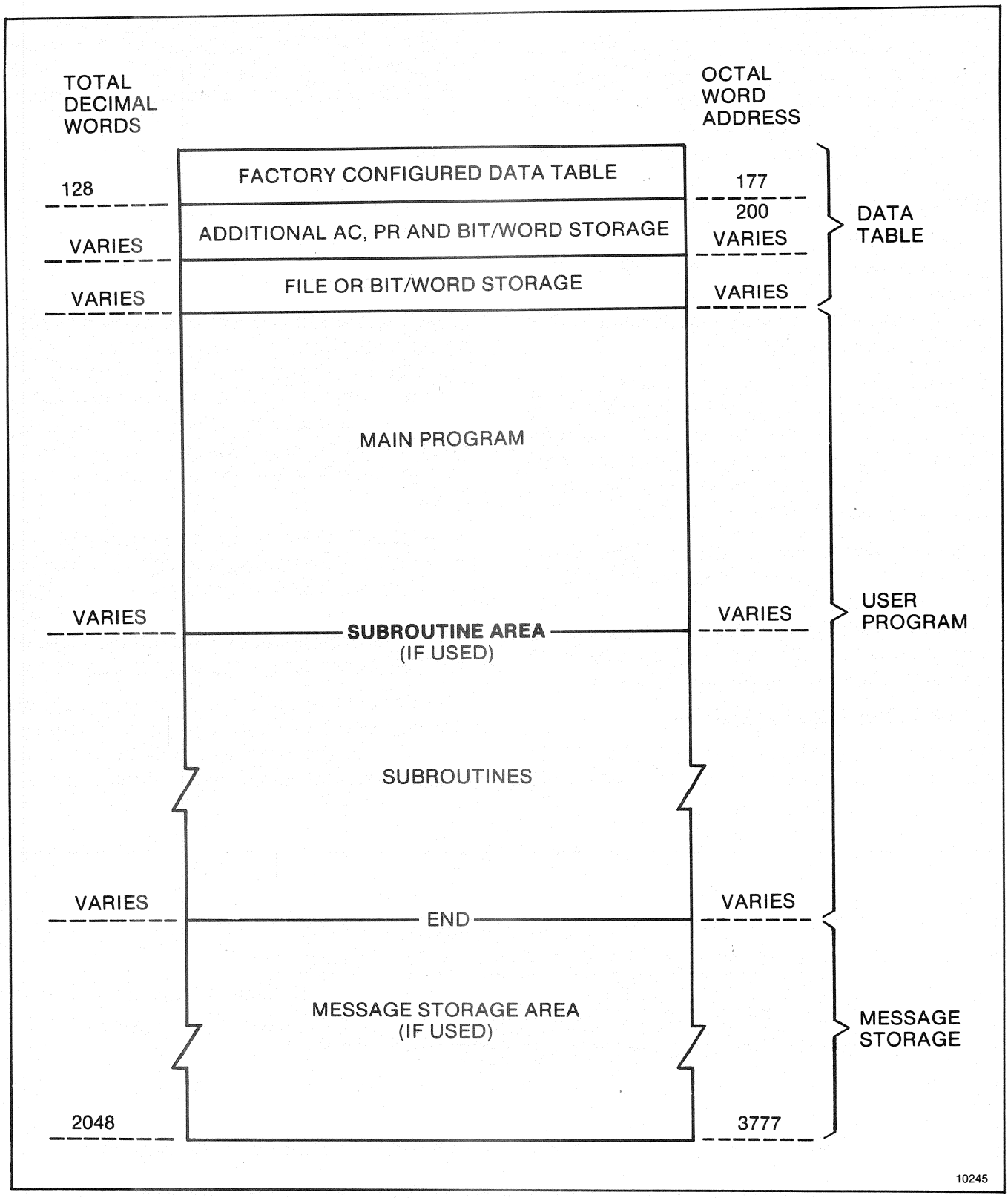
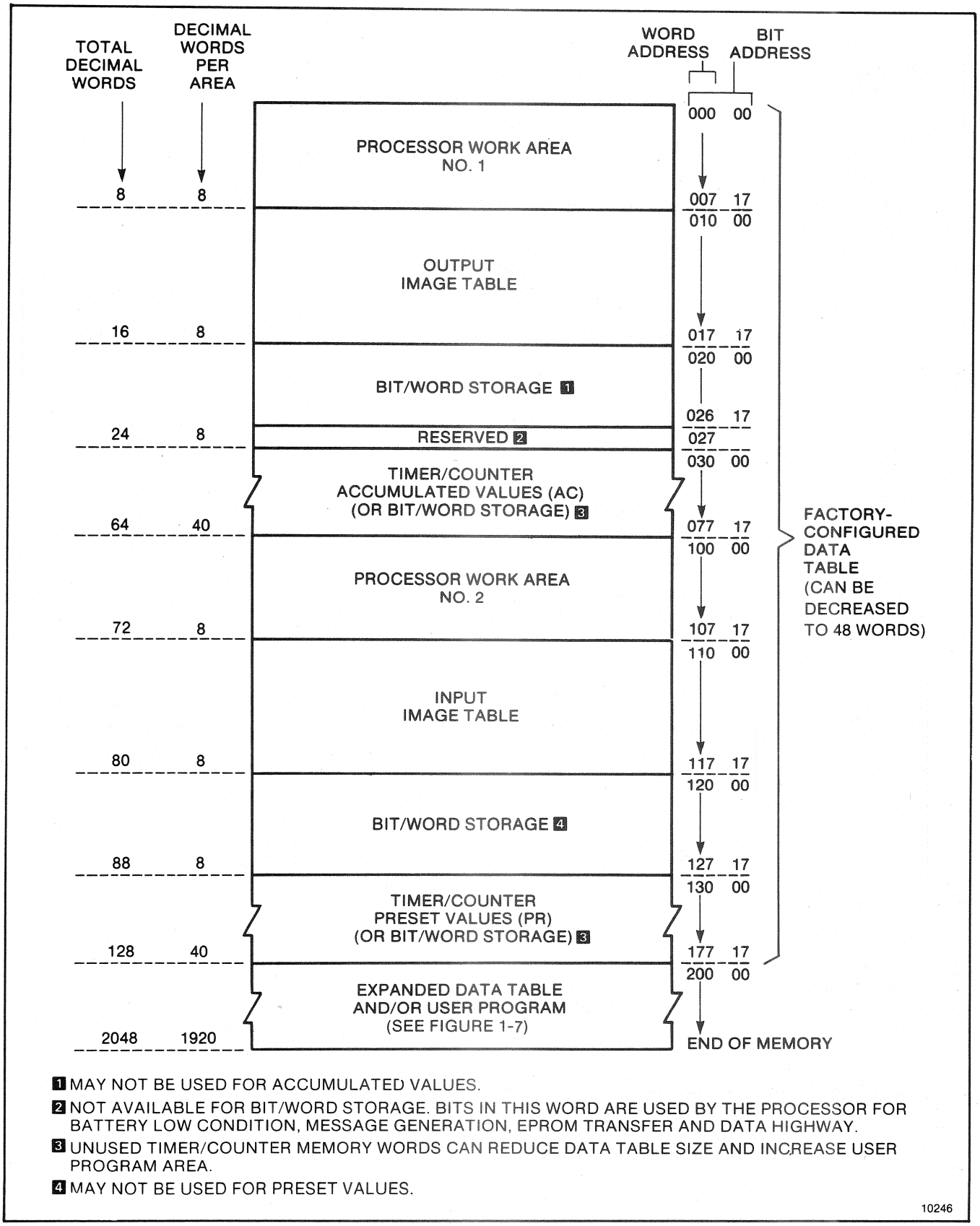


FIGURE 1-5 — User Memory Representation



1 MAY NOT BE USED FOR ACCUMULATED VALUES.

2 NOT AVAILABLE FOR BIT/WORD STORAGE. BITS IN THIS WORD ARE USED BY THE PROCESSOR FOR BATTERY LOW CONDITION, MESSAGE GENERATION, EPROM TRANSFER AND DATA HIGHWAY.

3 UNUSED TIMER/COUNTER MEMORY WORDS CAN REDUCE DATA TABLE SIZE AND INCREASE USER PROGRAM AREA.

4 MAY NOT BE USED FOR PRESET VALUES.

10246

FIGURE 1-6 — Data Table Organization, Factory Configured

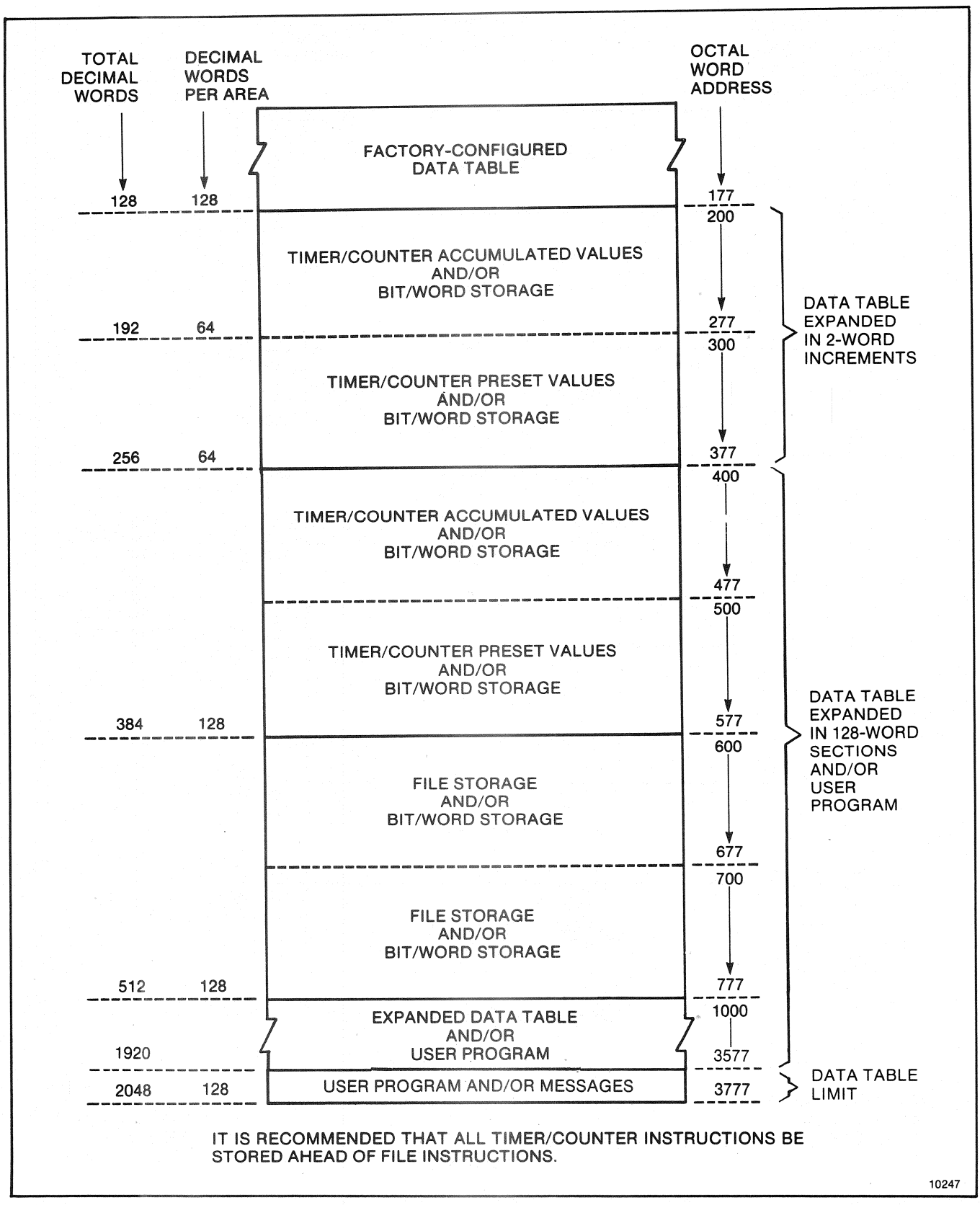


FIGURE 1-7 — Data Table Organization, Field Expandable

word of memory, but some block instructions can require up to eight (8) memory words. Unless specified elsewhere, instructions require 1 word of memory when their address is 377 or less, and 2 words of memory when their address is 400 or greater.

The User Program contains the Main Program and Subroutine Area. The Main Program follows the Data Table in memory. It generally contains the majority of User Program instructions. In the absence of a Subroutine Area, the Main Program is the User Program.

The Subroutine Area contains the instructions of one or more sections of program that are often repeated. They are written once and placed in the Subroutine Area where they can be used as often as needed. They may also include special sections of program which can be called upon only when needed. They are placed in the Subroutine Area to decrease nominal scan time. The boundary between the Subroutine Area and the Main Program serves as an end-of-program statement for the Main Program. The end of the Subroutine Area is the end of the User Program.

The 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 1772-912, Memory Organization and Structure of Mini-PLC-2/15 Processors.

1.4 HARDWARE/PROGRAM INTERFACE

It is important to understand how machine data, sensed by the input modules is used by the Processor to turn output devices ON or OFF. The Hardware-program interface occurs in the Input/Output Image Table.

1.4.1 Image Table

The primary Purose 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 address references both an I/O Image Table 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 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 1-8. The remaining two digits represent the bit address 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 1-9.

1.4.3 Fundamental Operation

The hardware-program interface is illustrated in Figure 1-10 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 113/12 is closed, the input module circuitry senses a voltage. The ON condition is reflected in the Input Image Table bit 113/12. During the Program scan, the Processor examines bit 113/12 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 012/06 to ON (1). The Pro-

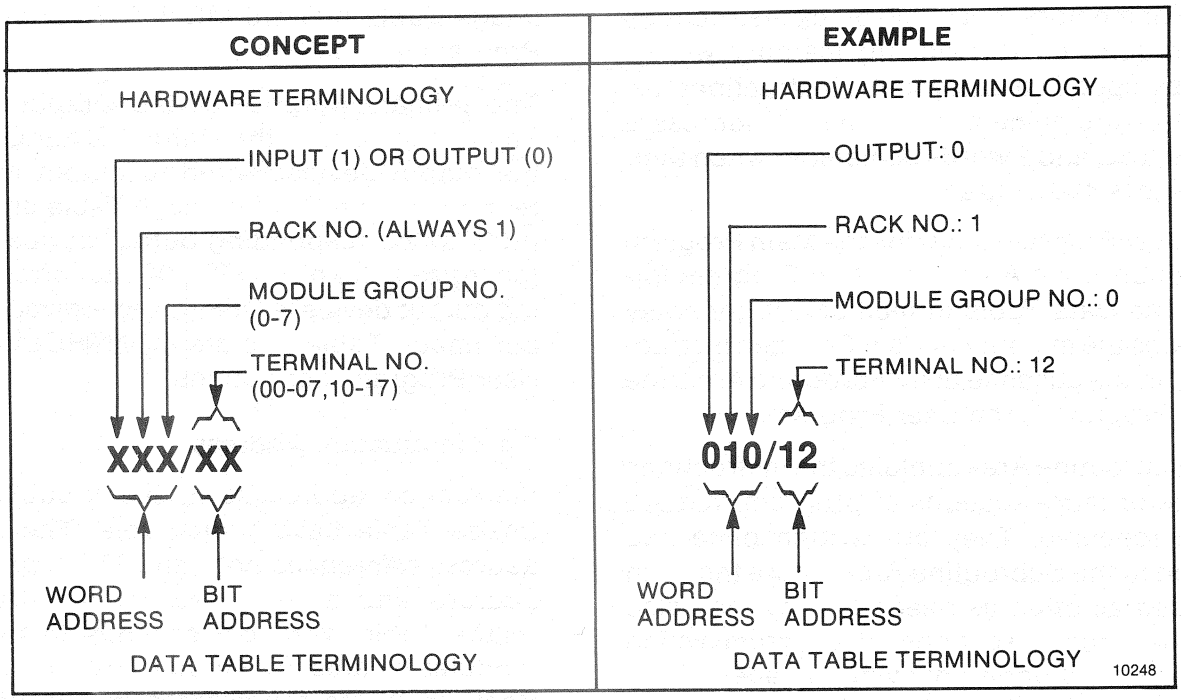


FIGURE 1-8 — Instruction Address Terminology

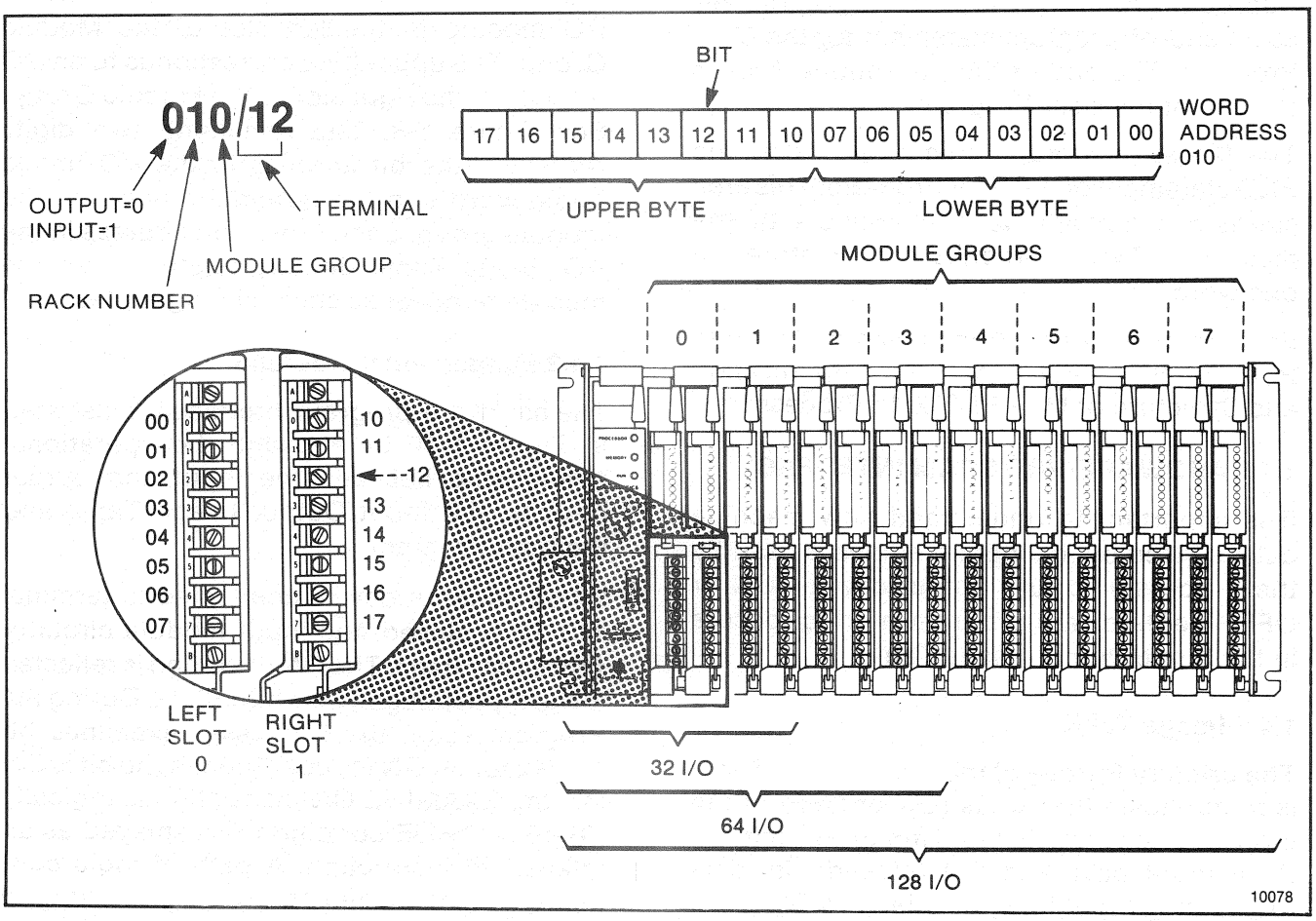


FIGURE 1-9 — Bit Address to Hardware Relationship

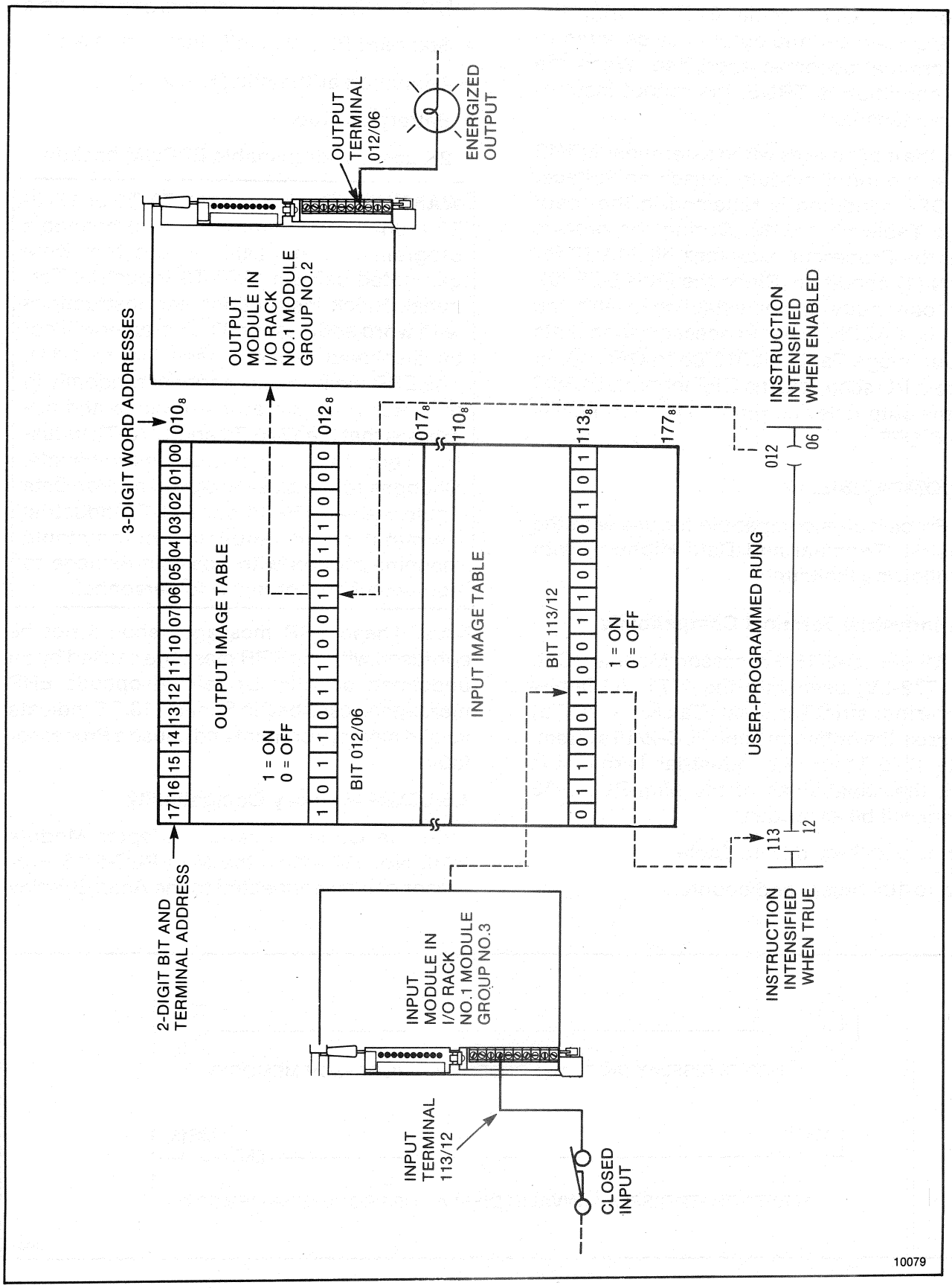


FIGURE 1-10 — Hardware-Program Interface

cessor turns ON terminal 012/06 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 113/12 opens, the input module senses no voltage. The OFF condition is reflected in the Input Image Table bit 113/12. During the proram scan, the Processor examines bit 113/12 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 012/06 to OFF (0). In the next I/O scan, it turns OFF terminal 012/06 and the output device wired to this terminal is turned OFF.

1.5 COMPATIBILITY

The Processor is compatible for use with the Industrial Terminal and Data Highway with the following limitations:

1.5.1 Industrial Terminal Compatibility

The Mini-PLC-2/15 Processor Module (Cat. No. 1772-LV) used with the 1771 I/O family and the Industrial Terminal (Cat. No. 1770-T3) comprise the optimum Mini-PLC-2/15 system. If the 1770-T1 or -T2 Industrial Terminal is used, the capabilities of the Mini-PLC-2/15 system will be as follows:

- Up to a 256 word Data Table
- Up to 104 timers and counters
- I/O forcing in one module group at a time
- Standard PLC-2 Family instruction set
- 4-function arithmetic (+, –, x, ÷)
- Battery backup
- 2K user-programmable EPROM backup

WARNING: Do not use a 1770-T1 or 1770-T2 Industrial Terminal to edit or change a program or Data Table values that were generated using a 1770-T3 Industrial Terminal. Block instructions and instructions with word addresses 400_8 or greater will not be displayed properly (see Figure 1-11). The ERR message may appear randomly in the User Program at instructions and addresses that the 1770-T1 and 1770-T2 Industrial Terminals are not designed to handle. Changes to the User Program and/or Data Table with a 1770-T1 or 1770-T2 Industrial Terminal could result in unpredictable machine motion with possible damage to equipment and/or injury to personnel.

Note: These ERR messages should not be confused with the ERR message caused by an undefined opcode. Undefined opcode ERR messages described in Section 13.7.5 indicate invalid memory content and cause a Processor fault.

1.5.2 Data Highway Compatibility

With the Communication Adapter Module (Cat. No. 1771-KA), the Mini PLC-2/15 Processor can be connected to the Allen-Bradley

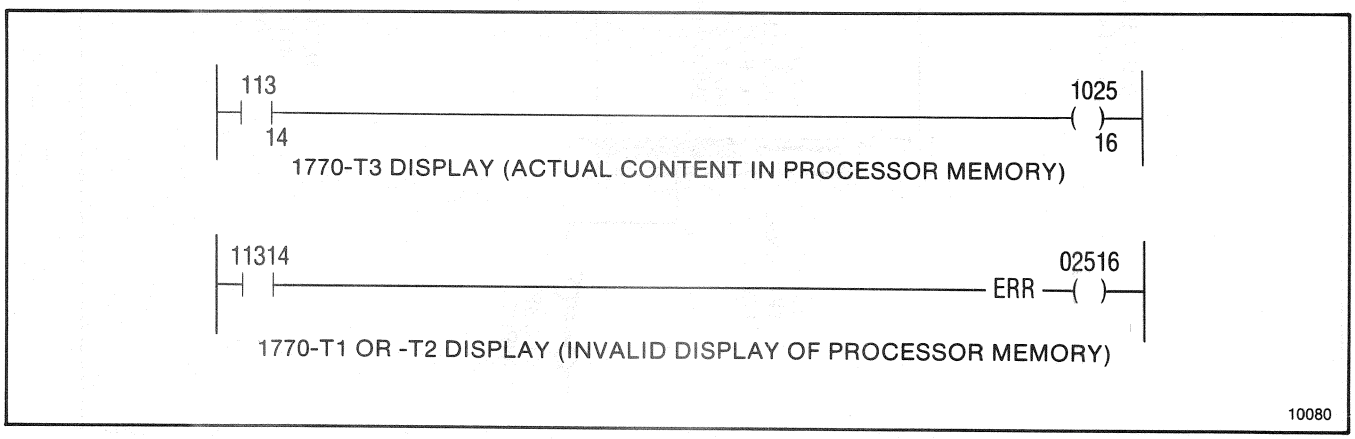


FIGURE 1-11 — ERR Message for Invalid Display of Processor Memory

Data Highway. If the Communication Adapter Module is Series A, Data Table addresses greater than 177_8 (in revisions A, B, C, D) or greater than 377_8 (in revision E) cannot be used in a data highway message or in the data highway communication zone area of the User Program.

WARNING: Do not use the On-Line Programming feature of the Mini PLC-2/15 when the Industrial Terminal (Cat. No. 1770-T3) is connected to the Processor through a Series A Communications Adapter Module (Cat. No. 1771-KA). Unpredictable machine operation could result in possible damage to equipment and/or injury to personnel.

1.5.3 Manual Applicability Using the 1770-T1 or -T2 Industrial Terminal

When the 1770-T1 or -T2 Industrial Terminal is used with the Mini-PLC-2/15, certain functions are not available and the corresponding text of this manual will not be applicable. The functions and corresponding sections of text are listed below.

- The Data Table can be expanded up to 256 words (address 377_8). This should be noted in Figure 1-7 and in Sections 1.3, 12.3.3, 12.3.5, 12.4 and 13.1.
- Keytop Overlay (Cat. No. 1770-KCA) should be used with the 1770-T1 or -T2 Industrial Terminal. This should be noted in Section 2.7.
- The instruction set is the standard set available with the 1772-LN3 Processor. Sections 8, 9, 10, and 11 are not applicable. Block Transfer will operate using multiple GET instructions, not using BLOCK TRANSFER block format instructions as described in Section 15.2. See the PLC-2/20 Programming and Operations Manual, Pub. No. 1771-802.
- The Processor Memory Layout Display is not available. Section 13.1.1 is not applicable.
- Help Directories are not available. Sections 13.3 is not applicable.
- Remote Mode Select is not available. Sections 13.4.6 is not applicable.
- On-Line Programming is not available. Section 13.6 is not applicable.
- Forcing can be applied to one module group at a time and the temporary END statement can be used as described in the PLC-2/20 Programming and Operations Manual, Pub. No. 1771-802. Otherwise, the Troubleshooting Aids described in Section 13.7 are not applicable.
- Data Table Clear and User Program Clear are not available. This should be noted in Section 13.8.
- The address digits of bit instructions described in Section 3 will appear above the instruction symbol on a 1770-T1 or -T2 Industrial Terminal. The programming examples in the manual are 1770-T3 displays which show the word address above the symbol and the bit number beneath it.

1.6 SUPPORT DOCUMENTATION

The following support documents contain additional information regarding Mini-PLC-2/15 Controller components.

- Mini-PLC-2/15 Programmable Controller Assembly and Installation Manual (Publication 1772-803): 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-2/15 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-2/15 Programmable Controller that are used when inputting or debugging the User Program. For information on installation, start-up, troubleshooting, etc, refer to Publication 1772-803, the Mini-PLC-2/15 Programmable Controller Assembly and Installation Manual.

2.1 MODE SELECT SWITCH

The Mini-PLC-2/15 Processor has a four-position keylock Mode Select Switch (Figure 2-1) that places the Processor in one of four operating modes:

- PROGRAM
- TEST
- RUN
- RUN/PROGRAM

The key can be removed from the Processor, thus locking it, in any of the four keyswitch positions.

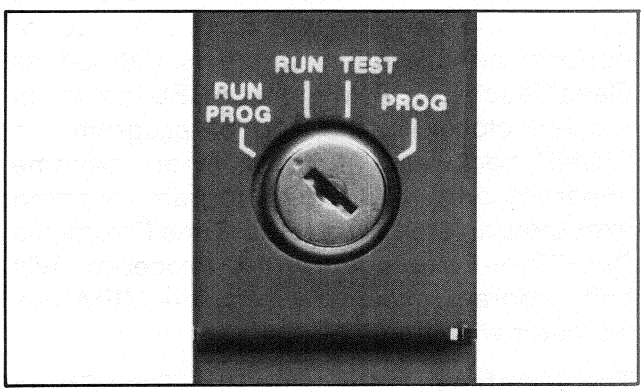


FIGURE 2-1 — Mode Select Keyswitch

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 disabled in this switch position.

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.

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 or Data Table are not permitted in this switch position.

2.1.4 RUN/PROG Position

This switch position places the Processor in the RUN/PROGRAM mode. The User Program will be executed and outputs are controlled by the program. Force functions, the Bit Manipulation function and on-line changes to User Program instructions and/or Data Table values are permitted in this switch position.

Remote Operation is activated when the key switch is in the RUN/PROG position. It allows the Processor to be placed into either the REMOTE PROGRAM or REMOTE TEST mode from the 1770-T3 Industrial Terminal using the appropriate key sequence. When the Processor keylock Mode Select Switch is in the RUN/PROG position, the REMOTE PROGRAM and REMOTE TEST modes are accessible from the Industrial Terminal. Otherwise, the TEST and PROGRAM keylock switch selectable modes operate identically to the REMOTE TEST and REMOTE PROGRAM modes, respectively. All references in this manual to the

PROGRAM and TEST modes also apply to the REMOTE PROGRAM and REMOTE TEST modes.

2.2 USER EPROM SYSTEM

The Mini-PLC-2/15 has the capability to house an ultra-violet light erasable PROM (UV EPROM) backup memory. It gives the option of storing the User Program, messages and the initial Data Table configuration on a non-volatile memory chip. After the ladder diagram program has been fully developed in the Processor's 2K CMOS RAM memory, the EPROM programming sequence can be initiated to duplicate the data onto the EPROM. The EPROM will automatically transfer the program back into RAM memory at power-up if the RAM memory should become invalid.

The EPROM is housed under a hinged access door on the lower left side of the Mini-PLC-2/15 Processor as shown in Figure 2-2. Further hardware information and instructions for initiating the EPROM programming sequence are presented in the ASSEMBLY and INSTALLATION manual, 1772-803, and the EPROM Programming Data Sheet, Publication No. 1770-915.

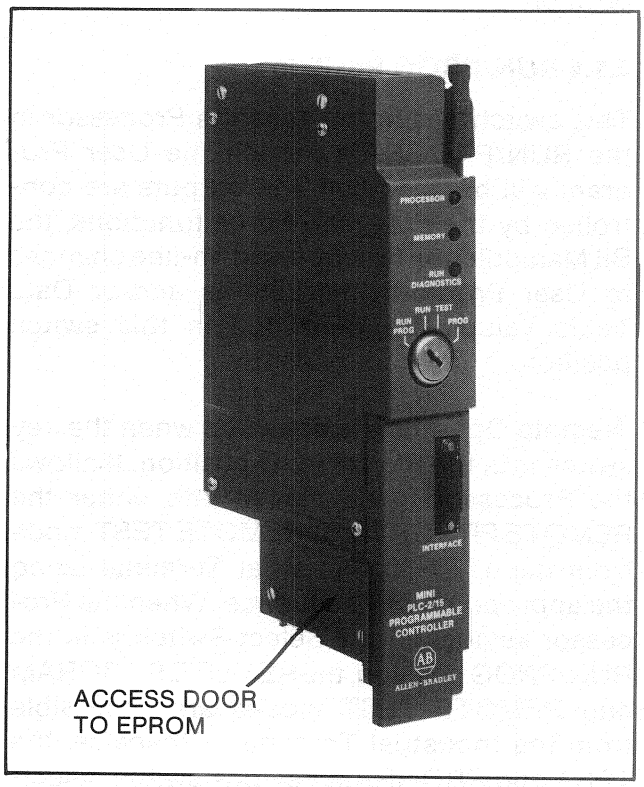


FIGURE 2-2 — Location of EPROM and Access Door

2.2.1 Memory Write Protect

When a "programmed" EPROM is in place in the Mini-PLC-2/15 Processor, Data Table values can be changed between word addresses 010_8 and 177_8 only. Values can be changed only when the Processor is in the PROGRAM mode or the RUN/PROGRAM mode using On-Line Data Change and Bit Manipulation.

The remaining memory words from word address 200_8 to end of memory are protected and cannot be altered by programming. As a result, the Memory Write Protect guards against any programming on-line or any unauthorized change of data on-line at addresses above 177_8, once the desired program has been stored in EPROM.

2.3 RUN-TIME ERRORS

The Processor and Industrial Terminal can diagnose certain errors occurring during the execution of the User Program which result from improper programming techniques. For example, it is possible to program a series of instructions which require the Processor to perform an operation which it cannot do, or perform an operation which is defined as illegal (such as a JUMP to a LABEL that is not located closer to the end of program - a "JUMP backwards"). These errors become apparent only while the program is being executed, so are termed Run-Time Errors. If a Run-Time Error occurs, the Processor will halt program execution and the MEMORY indicator will illuminate.

The first step in diagnosing Run-Time Errors is to connect the Industrial Terminal. It will display the message RUN-TIME ERROR in the initial Mode Select display. If the Industrial Terminal is already connected at the time that a Run-Time Error occurs, the ladder diagram will be replaced by the Mode Select display containing the error message.

Run-Time Errors can be detected by the Industrial Terminal when the Processor is in

either of two modes, PROGRAM or REMOTE PROGRAM. (If the keyswitch is in the RUN/PROG position, the Industrial Terminal will automatically put the Processor into REMOTE PROGRAM mode. If the keyswitch is in the RUN position, or when the Industrial Terminal is connected to the Processor through the Communication Adapter Module (Cat. No. 1771-KA), it must be changed manually to the PROG position).

After returning the Industrial Terminal display to Ladder Diagram mode by pressing keys [1][1] in Mode Selection, the Industrial Terminal will display the instruction that caused the error with a message describing the Run-Time Error. Causes of Run-Time Errors are described for Jump/Subroutine instructions in Section 8, File instructions in Section 9, and Block Transfer instructions in Section 15.

After the Run-Time Error has been corrected by editing the User Program, the Processor can be restarted by switching to the RUN or RUN/PROG position.

2.4 PROCESSOR DIAGNOSTIC INDICATORS

Diagnostic indicators are located on the front panel of the Mini-PLC-2/15 as shown in Figure 2-3. They indicate error conditions.

2.4.1 MEMORY Indicator

This red indicator will illuminate if any of the following errors occur which cause the Processor to halt operation:

- MEMORY PARITY ERROR
- EPROM ERROR
- RUN-TIME ERROR

The Industrial Terminal will display the error category as one of the following once the keyswitch is in the PROG or RUN/PROG position:

- HARDWARE ERROR
- RUN-TIME ERROR

Hardware errors include memory parity and EPROM errors.

MEMORY PARITY ERROR

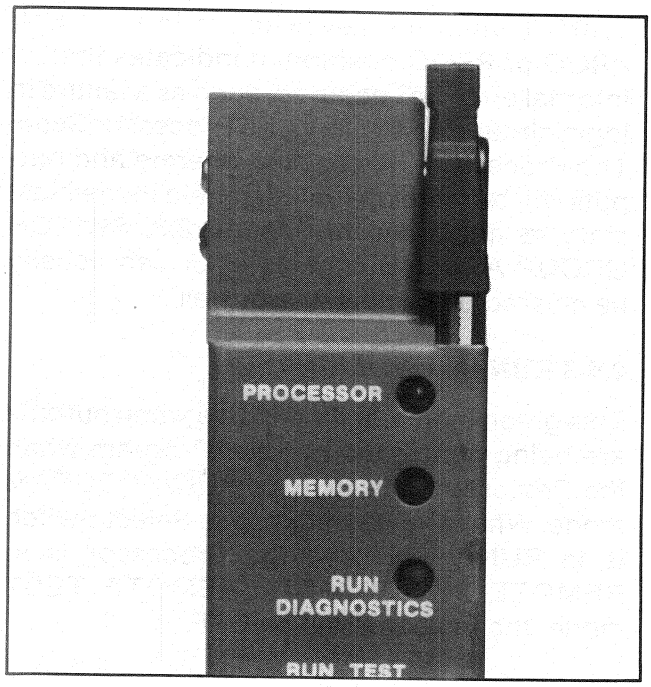


FIGURE 2-3 — Diagnostic Indicators

The MEMORY indicator will illuminate if an error occurs in the parity of data received from memory. In this circumstance, the error usually can be cleared by changing the Mode Select Switch to the PROG position, entering the REMOTE PROGRAM mode from the Industrial Terminal, or cycling line power.

EPROM ERROR

During EPROM programming, the MEMORY indicator will blink ON and OFF until the process is complete. EPROM programming typically takes 5 minutes/2K memory. When the EPROM has been successfully programmed, the MEMORY indicator will turn OFF. If it remains ON after programming, either the EPROM is faulty or an error has occured during EPROM programming. For additional information, see Publication 1770-915, EPROM Programming.

RUN-TIME ERRORS

Run-Time Errors are errors in programming. They are described in Section 2.3.

2.4.2 PROCESSOR Indicator

If this red indicator should illuminate, the Industrial Terminal will display HARDWARE

ERROR when the keyswitch is in the RUN/PROG or PROG position. It indicates that an internal error has occured, such as a failure in logic circuits controlling the Processor Scan. The Processor will cease to operate and outputs will be either disabled or held in their last state as described in Section 2.6, SWITCH GROUP ASSEMBLY. This error can usually be cleared by cycling line power.

2.4.3 RUN Indicator

This green indicator illuminates when outputs are being controlled by User Program when the Processor is in RUN or RUN/PROGRAM mode. When the Keylock Mode Select Switch is in RUN/PROG and the Processor is in REMOTE PROGRAM or REMOTE TEST mode, the indicator will be OFF.

2.5 POWER SUPPLY DIAGNOSTIC INDICATORS

The system Power Supply (Figure 2-4) has two diagnostic indicators on the front panel: The BATTERY LOW and DC ON indicators.

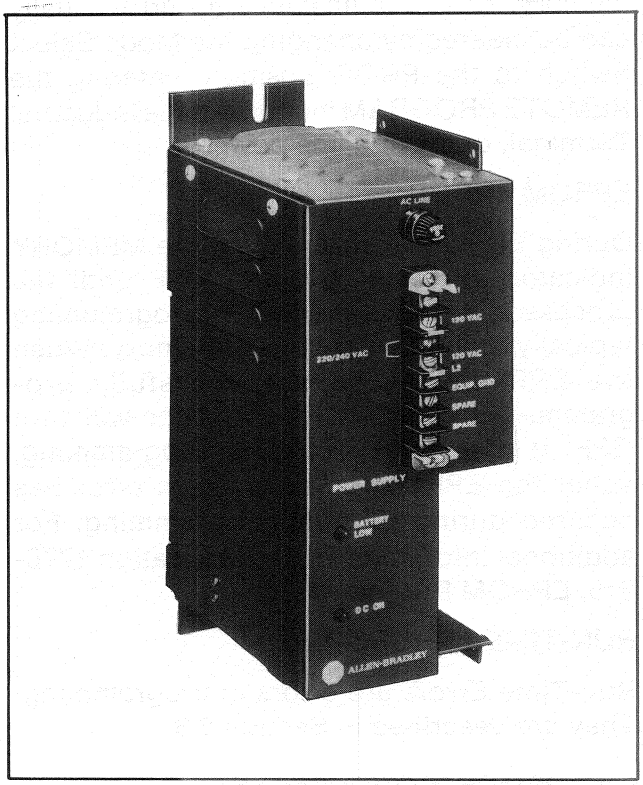


FIGURE 2-4 — System Power Supply (Without Battery Pack)

2.5.1 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 voltage condition is detected and the Mode Select Switch is in any 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.

2.5.2 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.

2.6 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 2-5.) 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 for switch #1:

- 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.

2.7 INDUSTRIAL TERMINAL

The Industrial Terminal (Cat. No. 1770-T3) is used to program the Mini-PLC-2/15.

2.7.1 Hook-Up

Perform the following steps to connect the Industrial Terminal to the Mini-PLC-2/15 Processor: (Figure 2-6)

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)

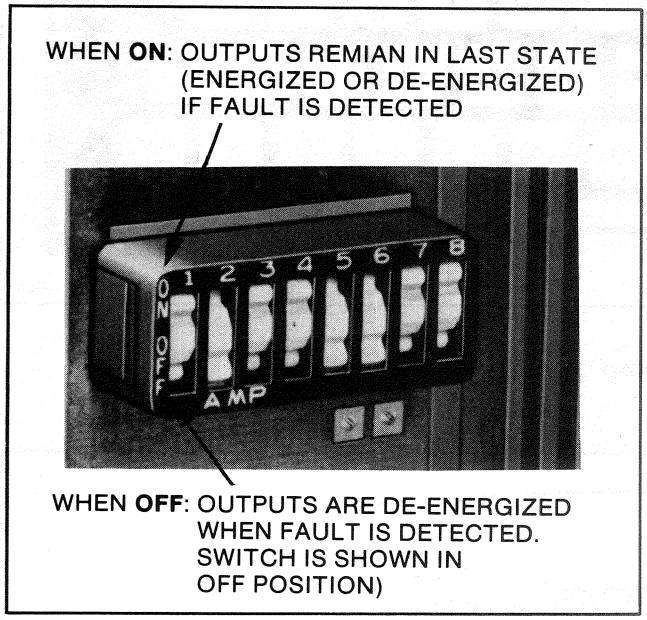


FIGURE 2-5 — Switch Group Assembly

to Channel A on the rear of the Industrial Terminal.

3. Connect the other end of the cable to the socket labeled INTERFACE on the front of the Mini-PLC-2/15 Processor.

4. Insert the PLC-2 Family Keytop Overlay (Cat. No. 1770-KCB), Figure 2-7, 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

2.7.2 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 Select display will appear. Any of the following occurrences can cause 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
- A Run-Time Error.

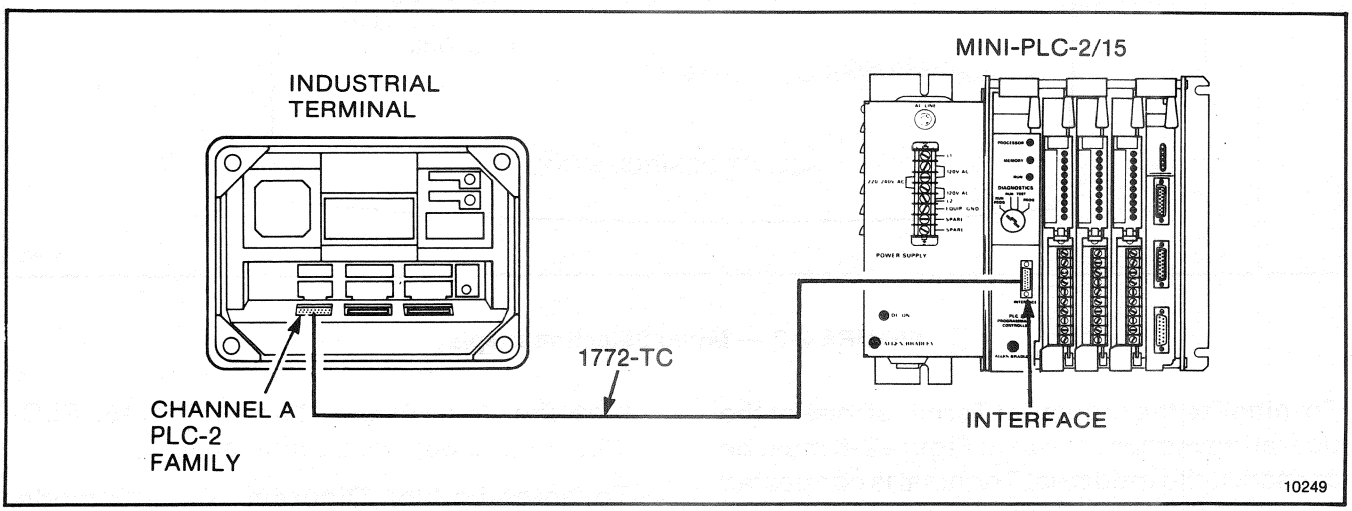


FIGURE 2-6 — Mini-PLC-2/15 Connection Diagram

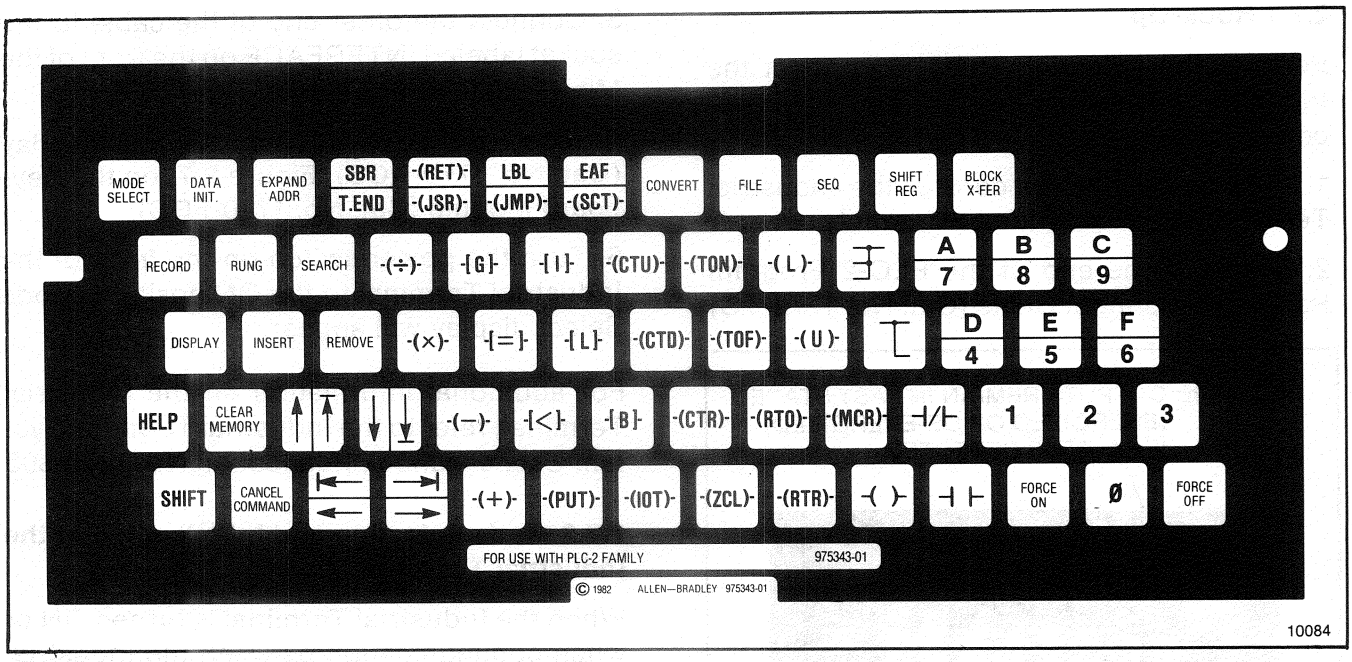


FIGURE 2-7 — PLC-2 Family Keytop Overlay (Cat. No. 1770-KCB)

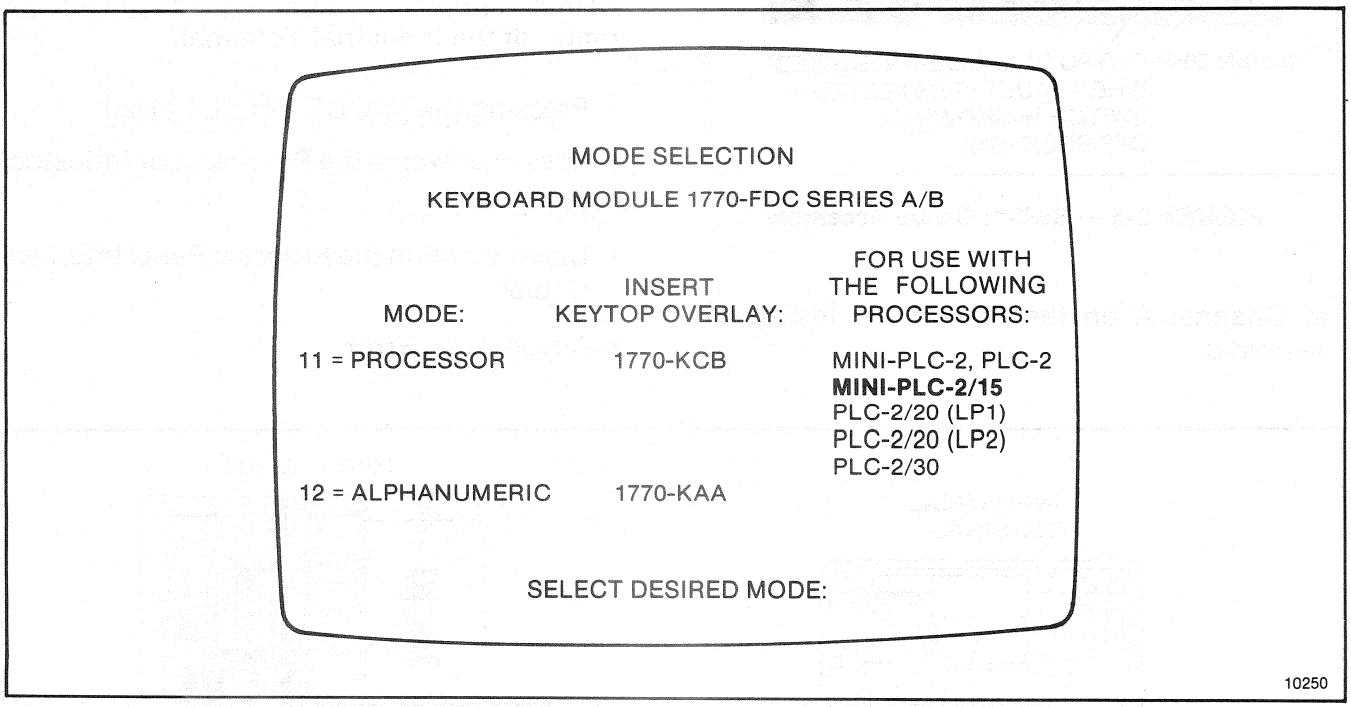


FIGURE 2-8 — Mode Selection Display

To initialize the Industrial Terminal, one of the operating modes shown in Figure 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 Family Keytop Overlay (Cat No. 1770-KCB).

When the Industrial Terminal is to be used as an alphanumeric data terminal, press [1][2] on the Alphanumeric Keytop Overlay (Cat. No. 1770-KAA). Operation of the Industrial Terminal with the Alphanumeric Keytop Overlay is described in Section 14, Peripheral Functions.

2.7.3 Keytop Overlay

The PLC-2 Family Processors are programmed using the PLC-2 Family Keytop Overlay shown in figure 2-7. The overlay is divided functionally into the following three (3) groups.

- Instruction Keys (center area)
- Number Keys (right side)
- Control Keys (left side)

The functions of the keys will be described in detail starting with Section 3 of this manual.

This keytop overlay is common to all the Processors in the PLC-2 Family. Therefore it contains keys for some functions that are not possible with the Mini-PLC-2/15. These keys are:

[EAF / SCT] [Convert] and [SHIFT / REG]

When either of these keys is pressed, the message "FUNCTION NOT AVAILABLE WITH THIS PROCESSOR" will appear on the screen.

The [EAF] key is reserved for future enhancements to the Mini-PLC-2/15 and should not be used.

2.8 MINI PROCESSOR TRANSPORT CABLE

A 1772-LV Processor, whether or not equipped with EPROM backup, can be removed from the I/O rack without loss of memory by using the Mini Processor Transport Cable (Cat. No. 1772-CD). This 2-foot cable electrically connects the battery pack of the system power supply to the Processor module so both can be transported. Using the transport cable, the Processor and battery pack should be removed or installed when system power is ON.

CAUTION: If the Processor is removed from the rack or installed when system power is OFF, the momentary loss of battery backup during the exchange of battery cables could alter memory content. The user Program and messages must be re-entered into memory if memory contents are altered and the EPROM backup has not been programmed.

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.

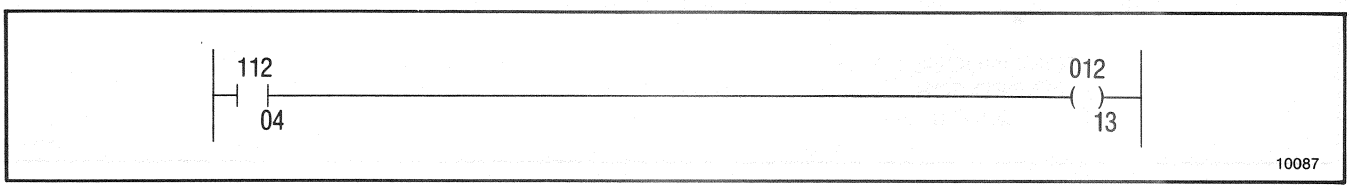


FIGURE 3-1 — EXAMINE ON Instruction

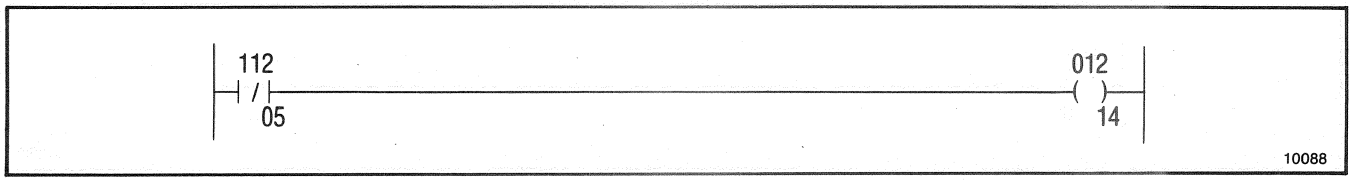


FIGURE 3-2 — EXAMINE OFF Instruction

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. 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 may determine the ON or OFF status of an output device. This instruction can also be used to set a storage bit to ONE 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 this reason care should be taken not to inadvertently enter an unconditional output instruction.

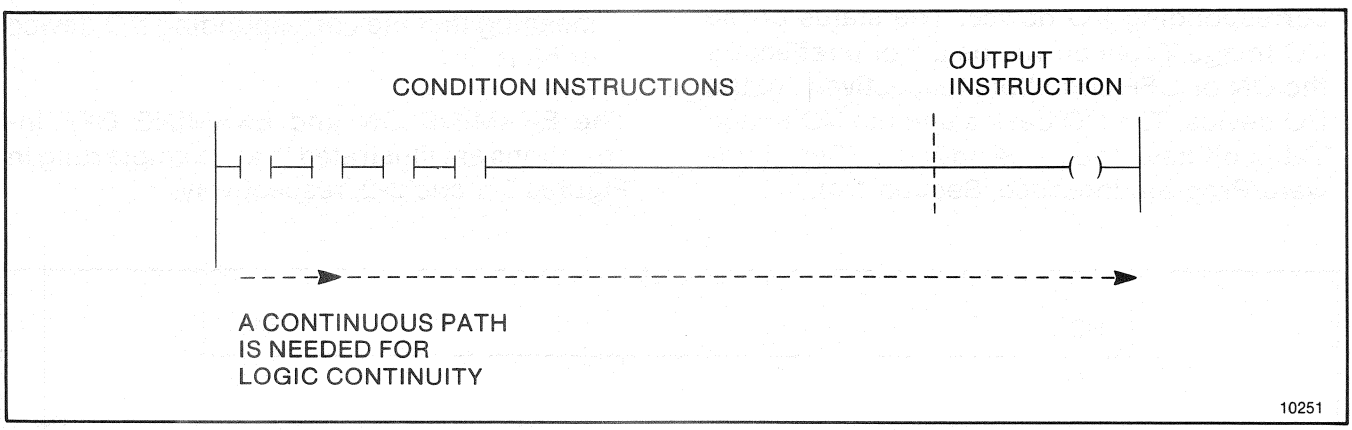


FIGURE 3-3 — Ladder-Diagram Rung

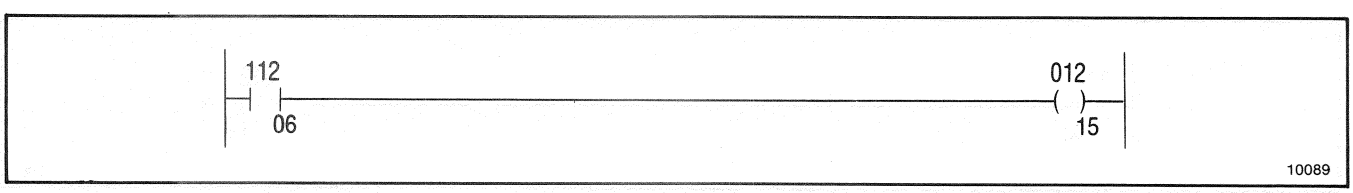


FIGURE 3-4 — OUTPUT ENERGIZE Instruction

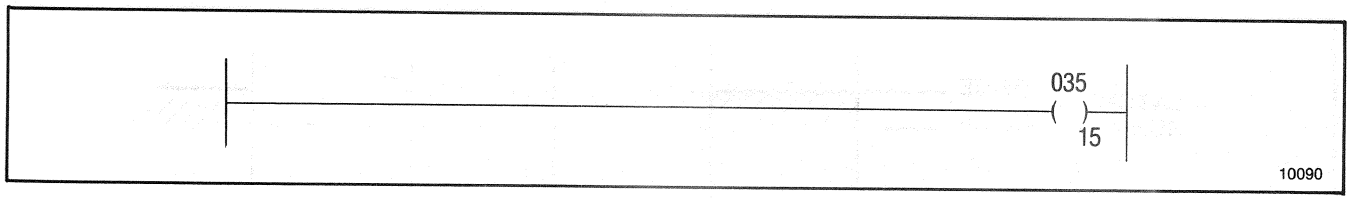


FIGURE 3-5 — Unconditional OUTPUT ENERGIZE Instruction

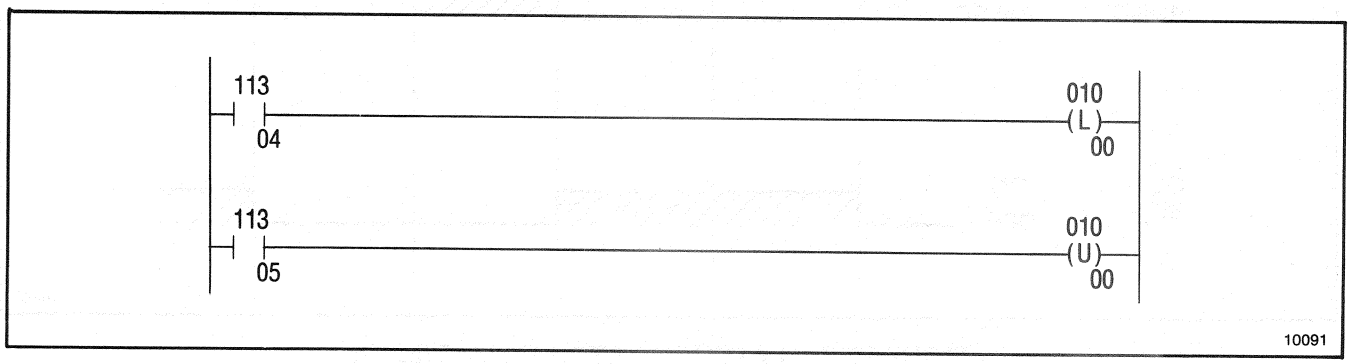


FIGURE 3-6 — LATCH/UNLATCH Instructions

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 de-energize 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 set by an OUTPUT LATCH instruction.

When the Mode Select Switch is changed from the RUN or RUN/PROG position, the last LATCH or UNLATCH instruction continues to control the addressed memory bit, but the output device is disabled by the Processor. When the Mode Select Switch is turned back to RUN or RUN/PROG, 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 or RUN/PROG position, the addressed memory bit and output device, if

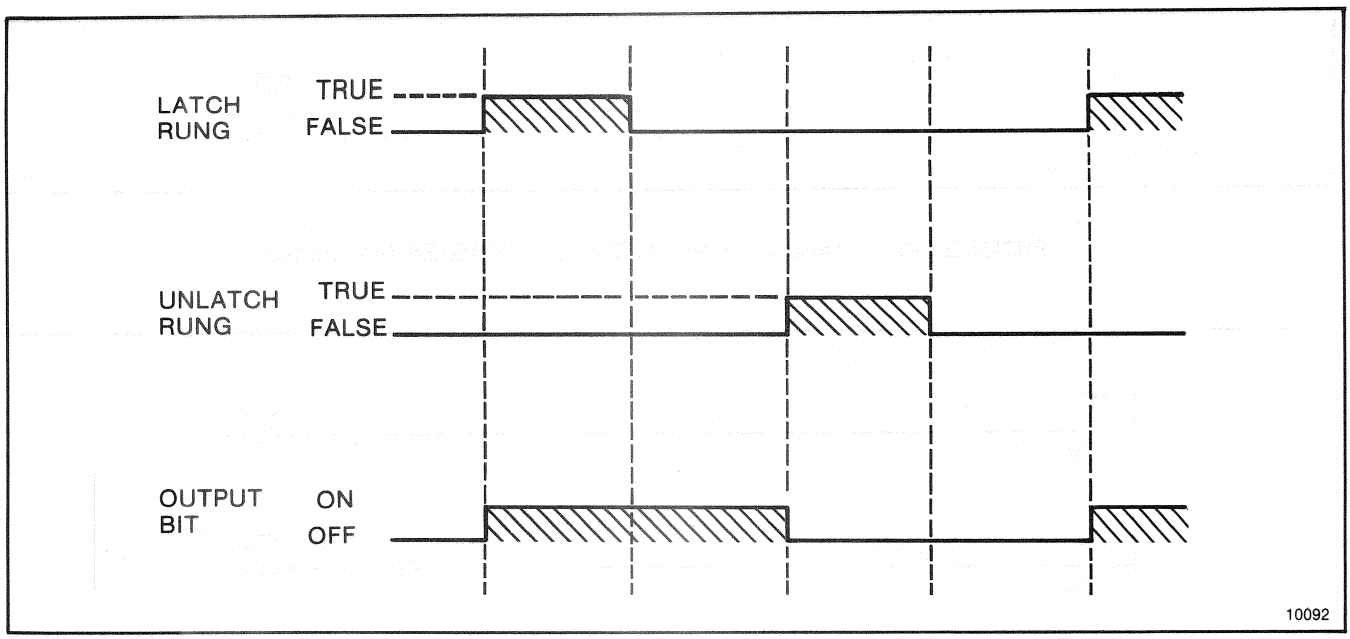


FIGURE 3-7 — LATCH/UNLATCH Timing Diagram

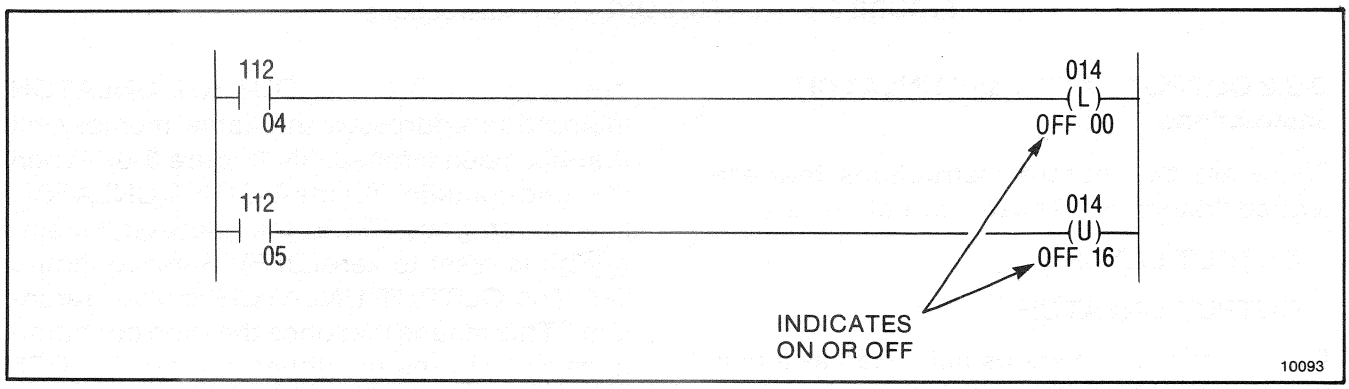


FIGURE 3-8 — LATCH/UNLATCH Indication

latched ON, will 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 will be energized immediately when the Mode Select Switch is turned to the RUN or RUN/PROG 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.

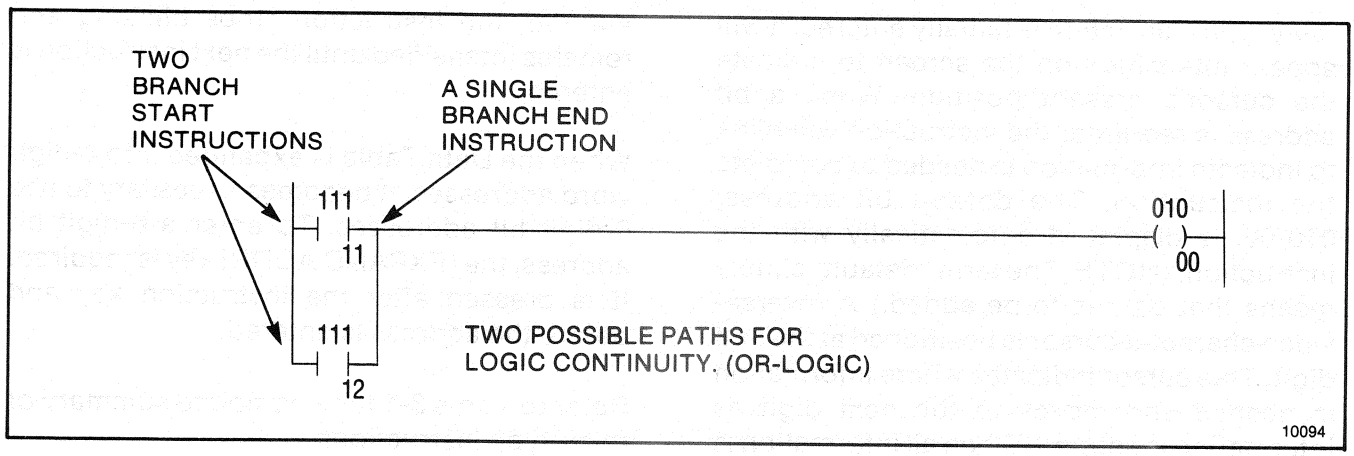


FIGURE 3-9 — Branching Instructions

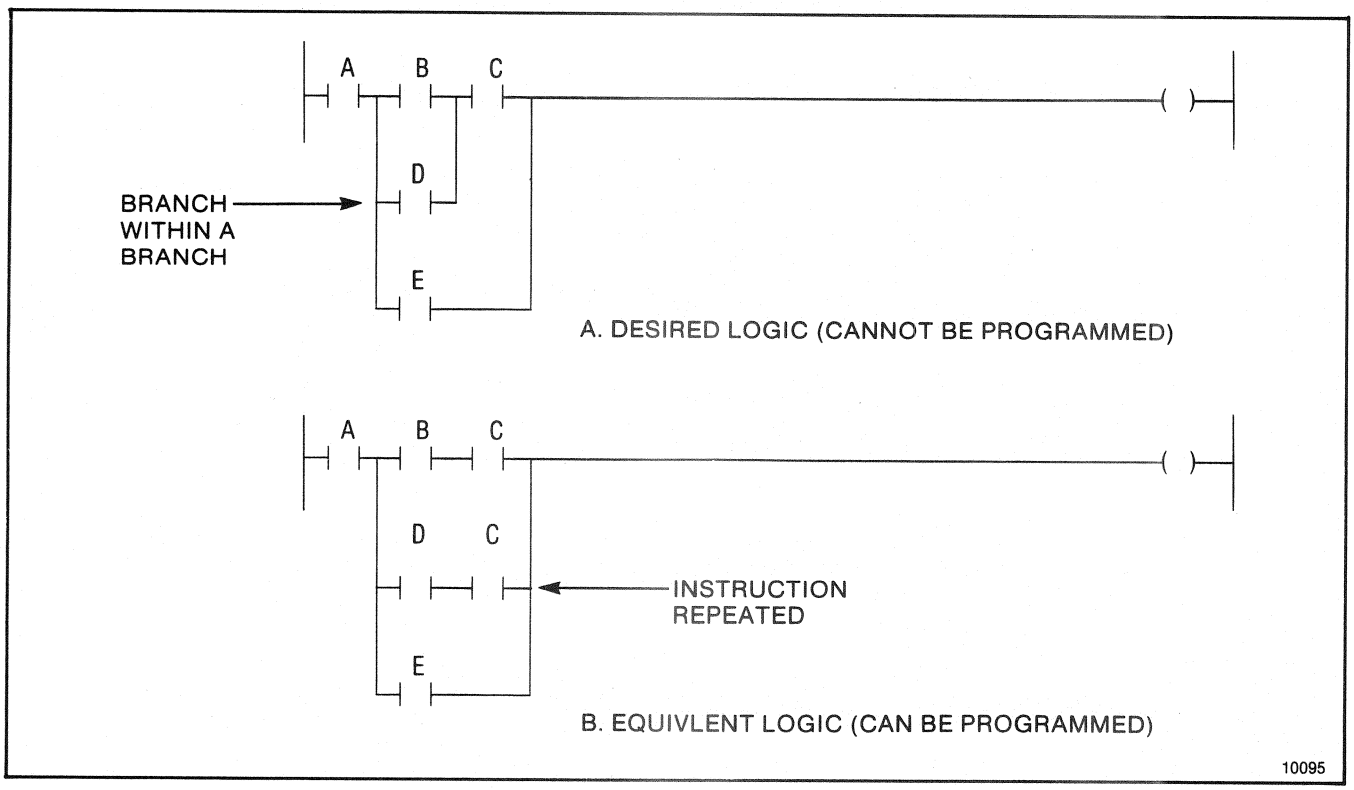


FIGURE 3-10 — Nested Branching vs. Proper Programming

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).

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.

When the Data Table is expanded into 4-digit word addresses, it becomes necessary to use 6-digit bit addresses. To enter a 6-digit bit address, the [EXPAND ADDR] key is required. It is pressed after the instruction key and before the address is entered.

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

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. The word address is displayed above the instruction and the bit number below it. To enter a bit address larger than 5 digits, press the [EXPAND ADDR] key after the instruction key and then enter the bit address. Use a leading zero if necessary.

KEYTOP SYMBOL	INSTRUCTION NAME	1770-T3 DISPLAY	DESCRIPTION
-| |-	EXAMINE ON	XXX -| |- XX	When the addressed memory bit is ON, the instruction is TRUE.
-|/|-	EXAMINE OFF	XXX -|/|- XX	When the addressed memory bit is OFF, the instruction it TRUE.
-( )-	ENERGIZE	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	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	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 de-energized. 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 each 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 488 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 an 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 and are 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 17.

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.

4.1 TIMER INSTRUCTIONS

A timer counts elapsed time-base intervals 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.

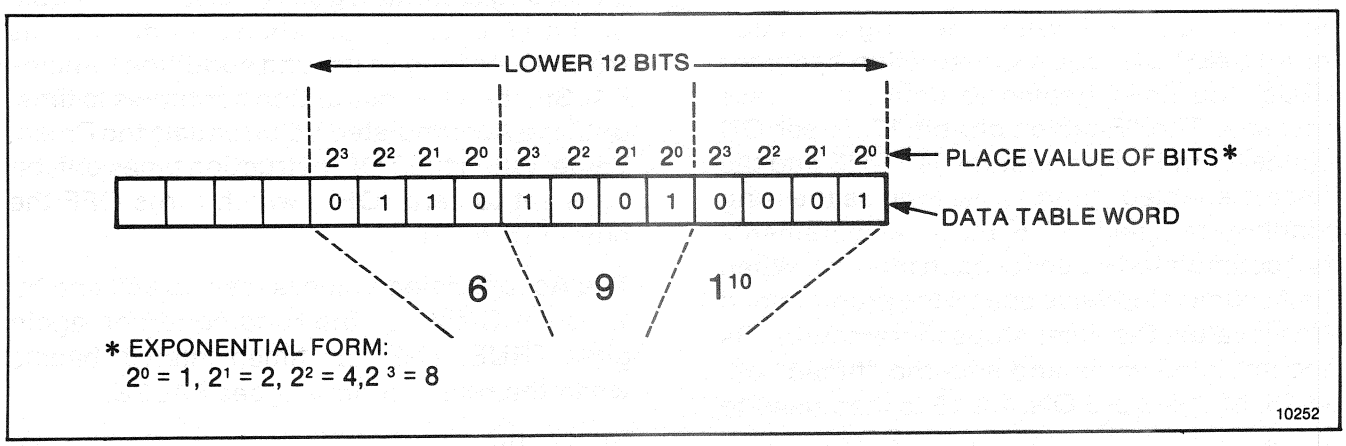


FIGURE 4-1 — BCD Format

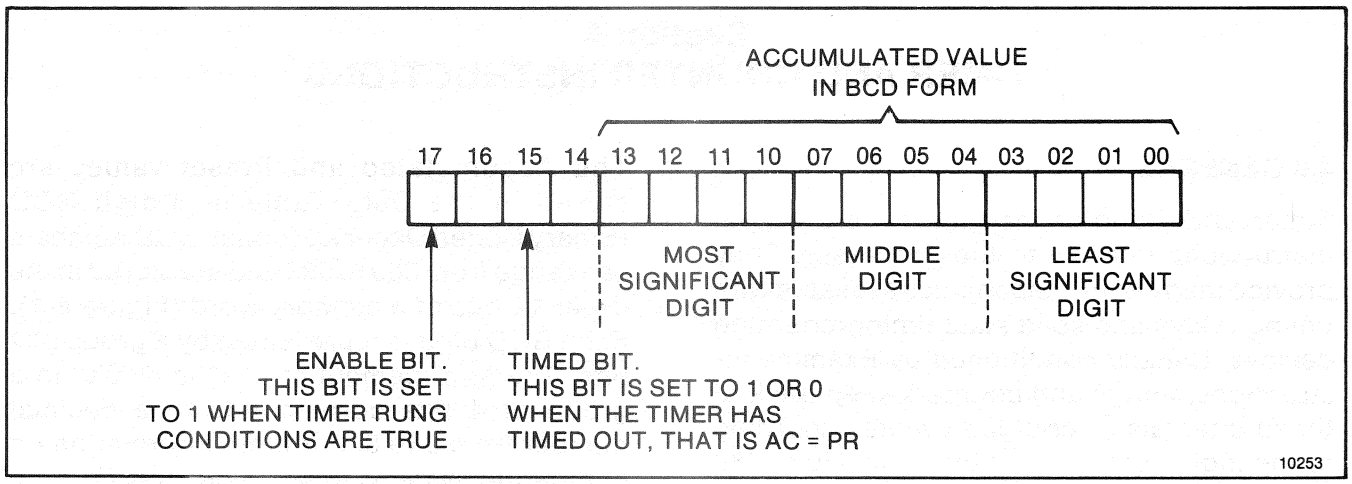


FIGURE 4-2 — Timer Accumulated Value Word

The three types of timers available with the Mini-PLC-2/15 Controller are:

TIMER ON-DELAY -(TON)-

TIMER OFF-DELAY -(TOF)-

RETENTIVE TIMER -(RTO)-

All three 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:

1.0 second
0.1 second
0.01 second (10 milliseconds)

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 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 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

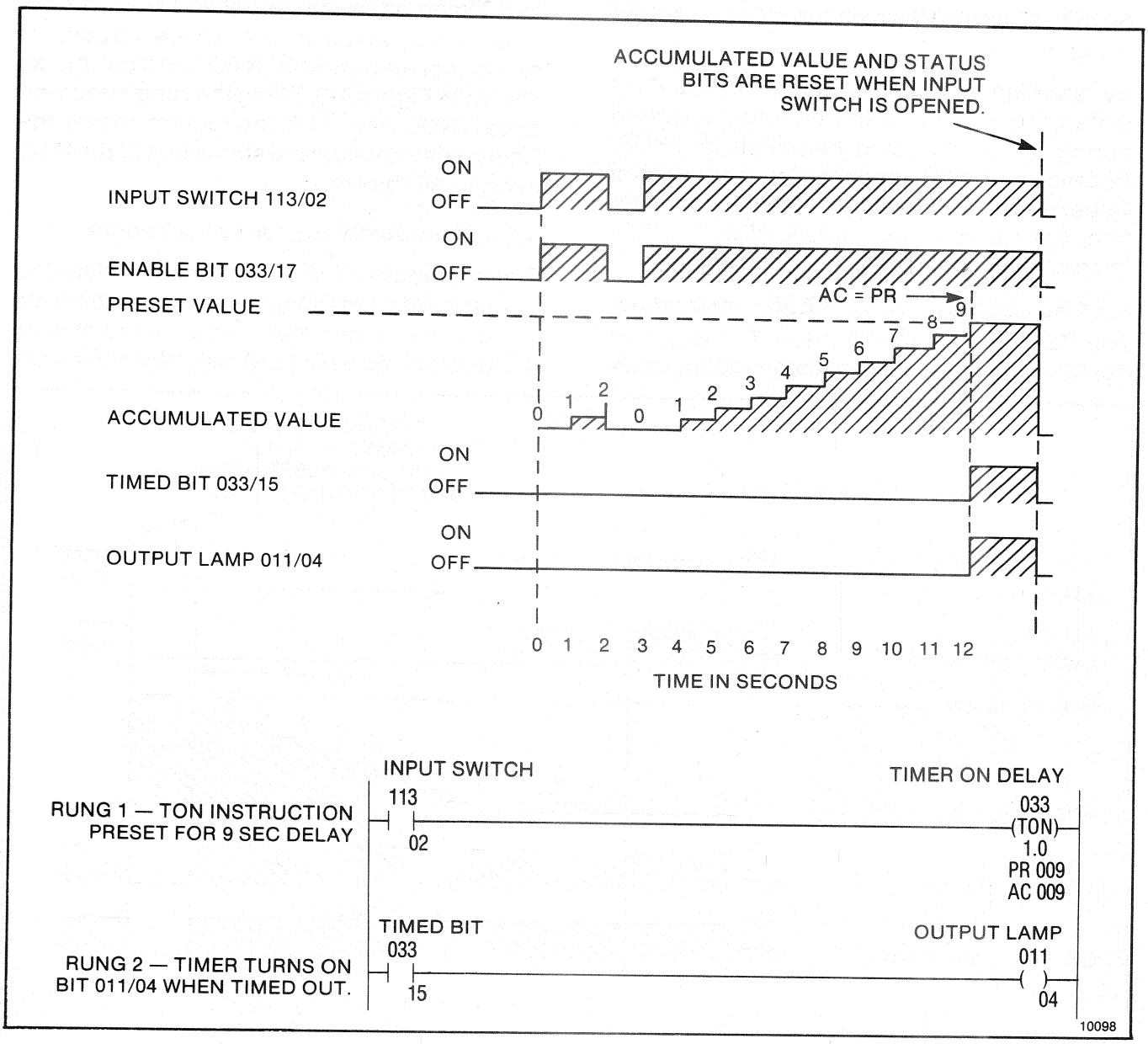


FIGURE 4-3 — TIMER ON-DELAY Timing Diagram & Programming

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
- 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 Enable 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 ms Timers

The accuracy of the 10 ms timer is related to nominal scan time. When scan times are 9 ms or less, the 10 ms timer is accurate to plus or minus one time base (±10 ms). When the scan

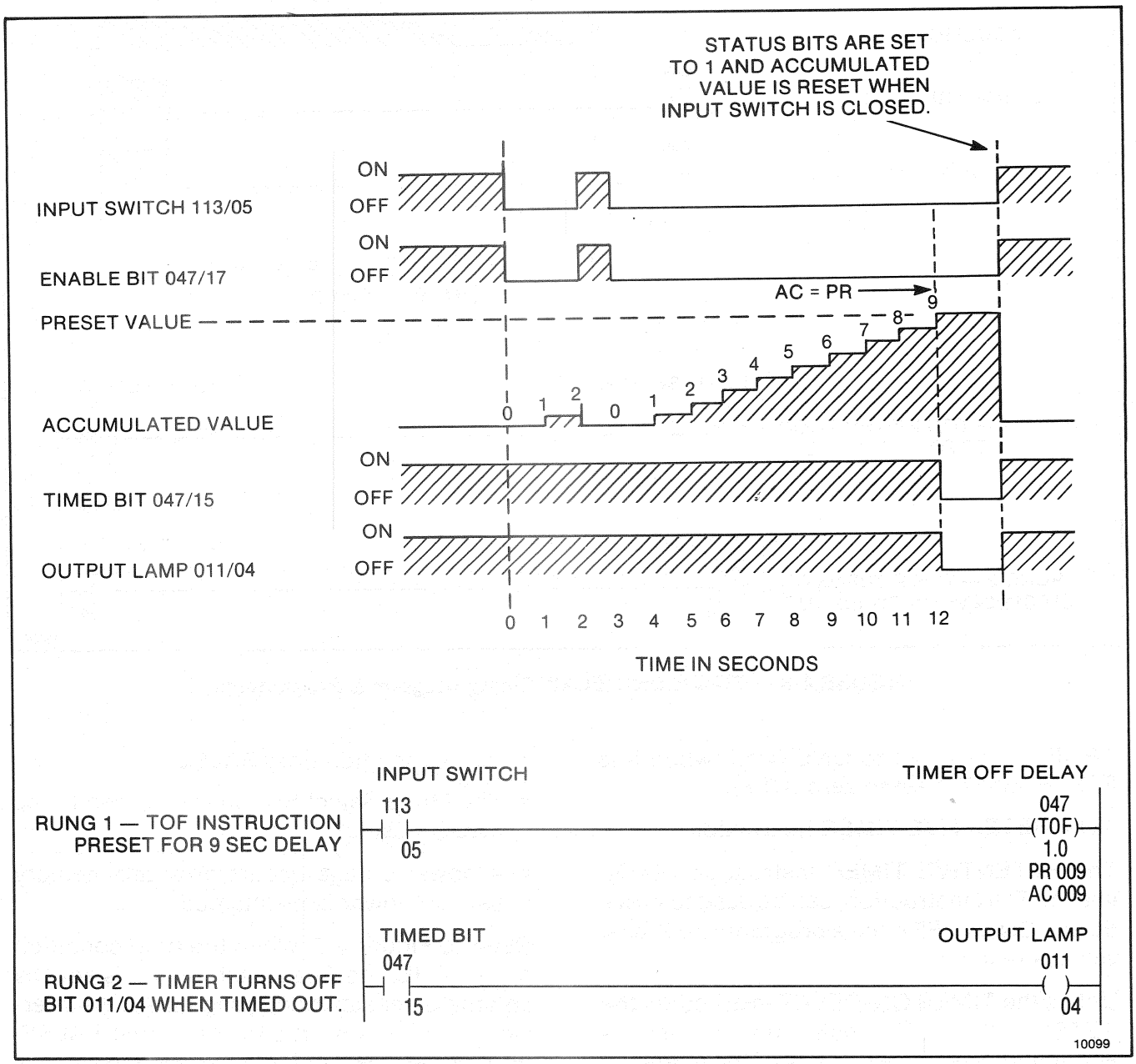


FIGURE 4-4 — TIMER OFF-DELAY Timing Diagram & Programming

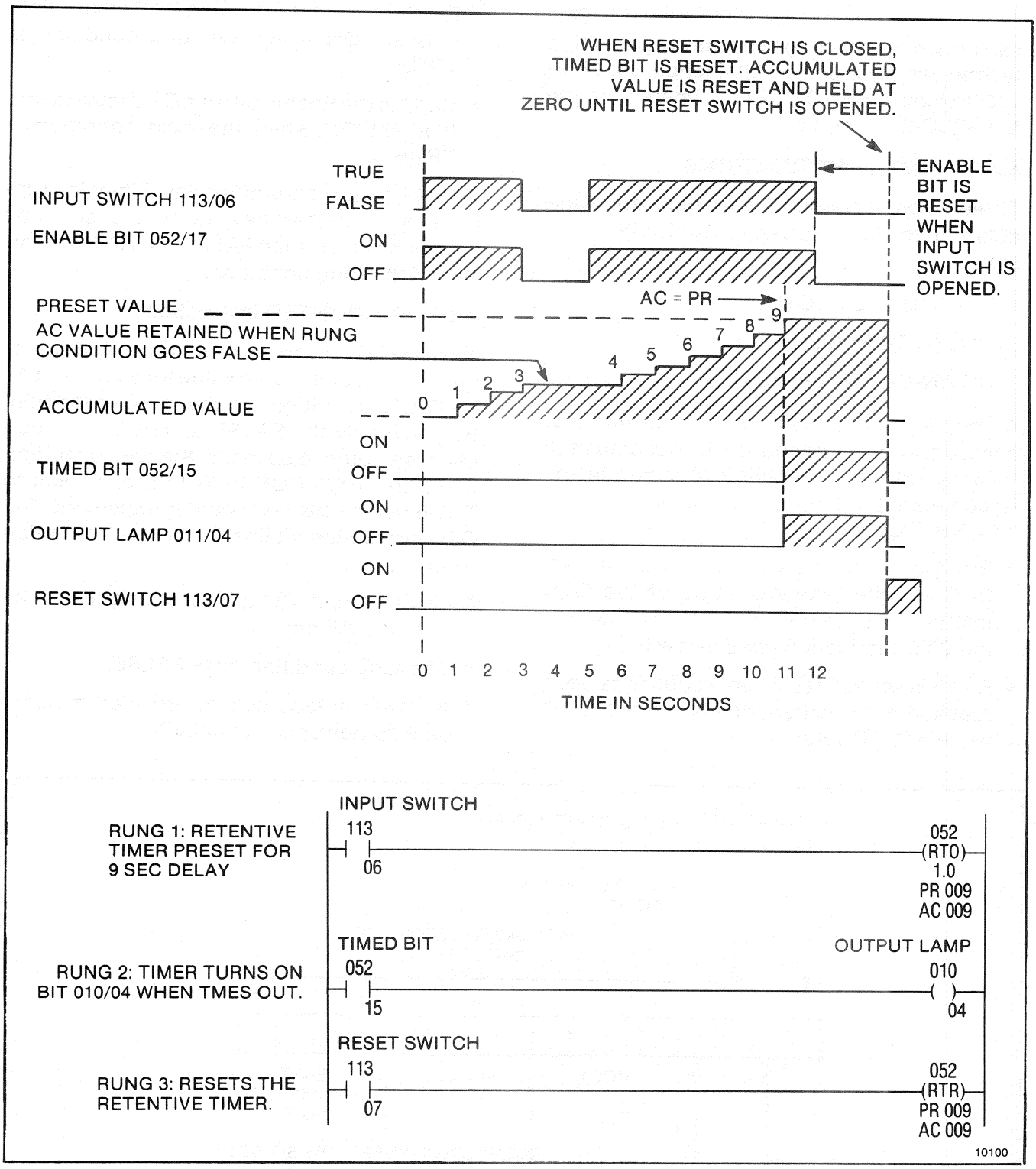


FIGURE 4-5 — RETENTIVE TIMER with RESET Timing Diagram & Programming

time is greater than 9ms, accuracy of ±10ms can be achieved through special programming techniques discribed in Publication No. 1772-702, Programming 0.01 Sec. Timers With the Mini-PLC-2 Controller.

4.2 COUNTER INSTRUCTIONS

Three types of counter instructions are available with the Mini-PLC-2/15 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 ONE 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 ONE 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.
- The rung condition goes FALSE.
- A power outage occurs provided memory backup power is maintained.

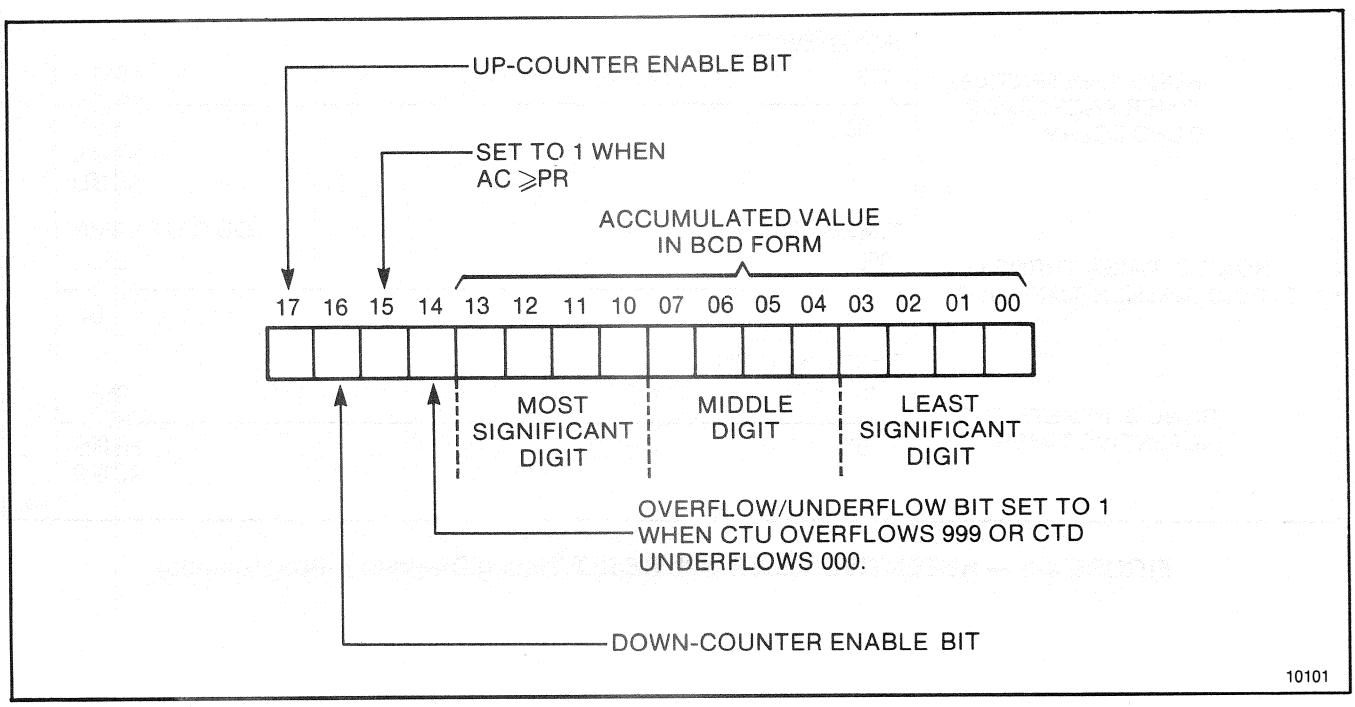


FIGURE 4-6 — Counter Accumulated Value Word

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 instruction 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.

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.

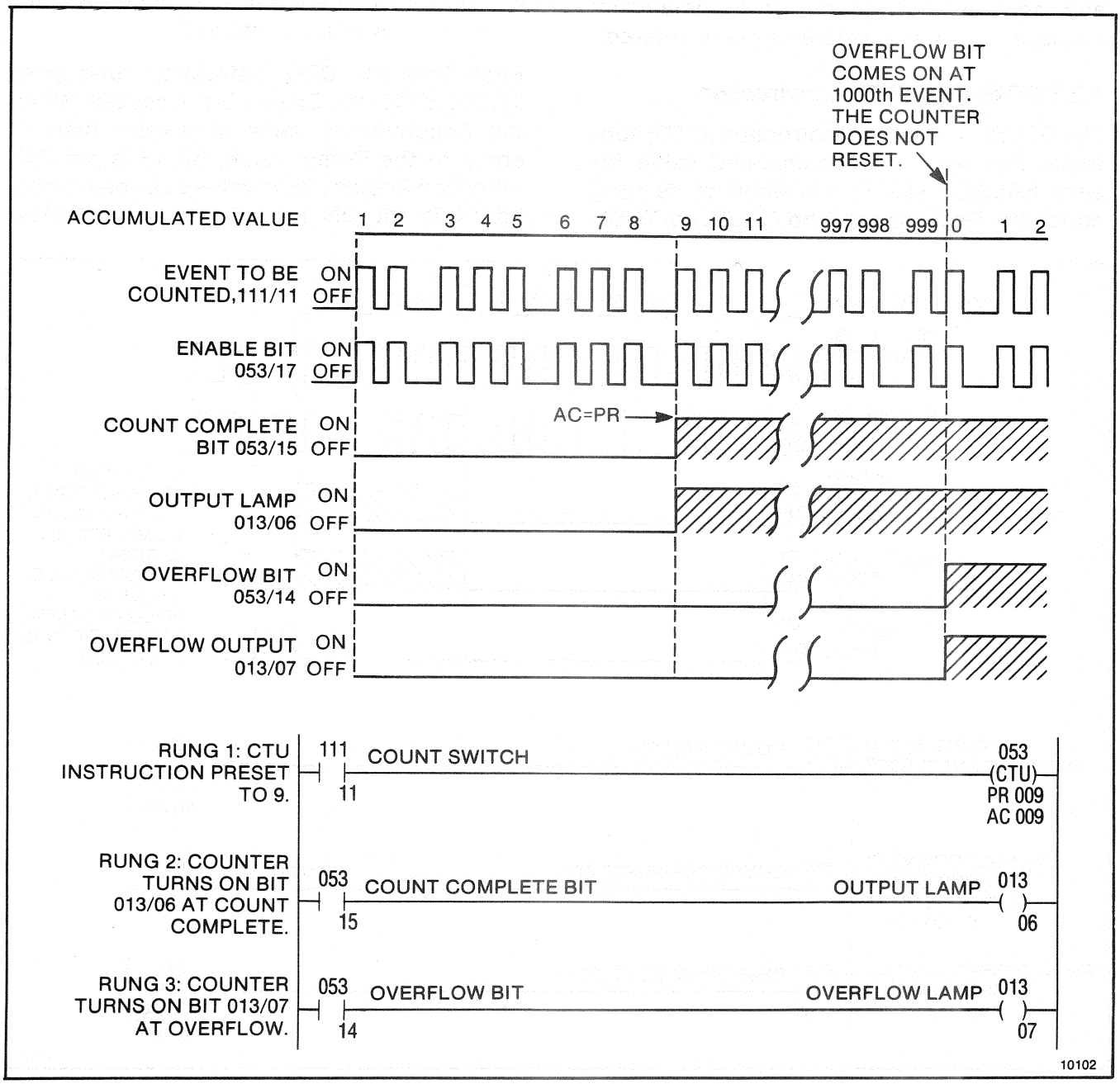


FIGURE 4-7 — UP-COUNTER Diagram & Programming

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.

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

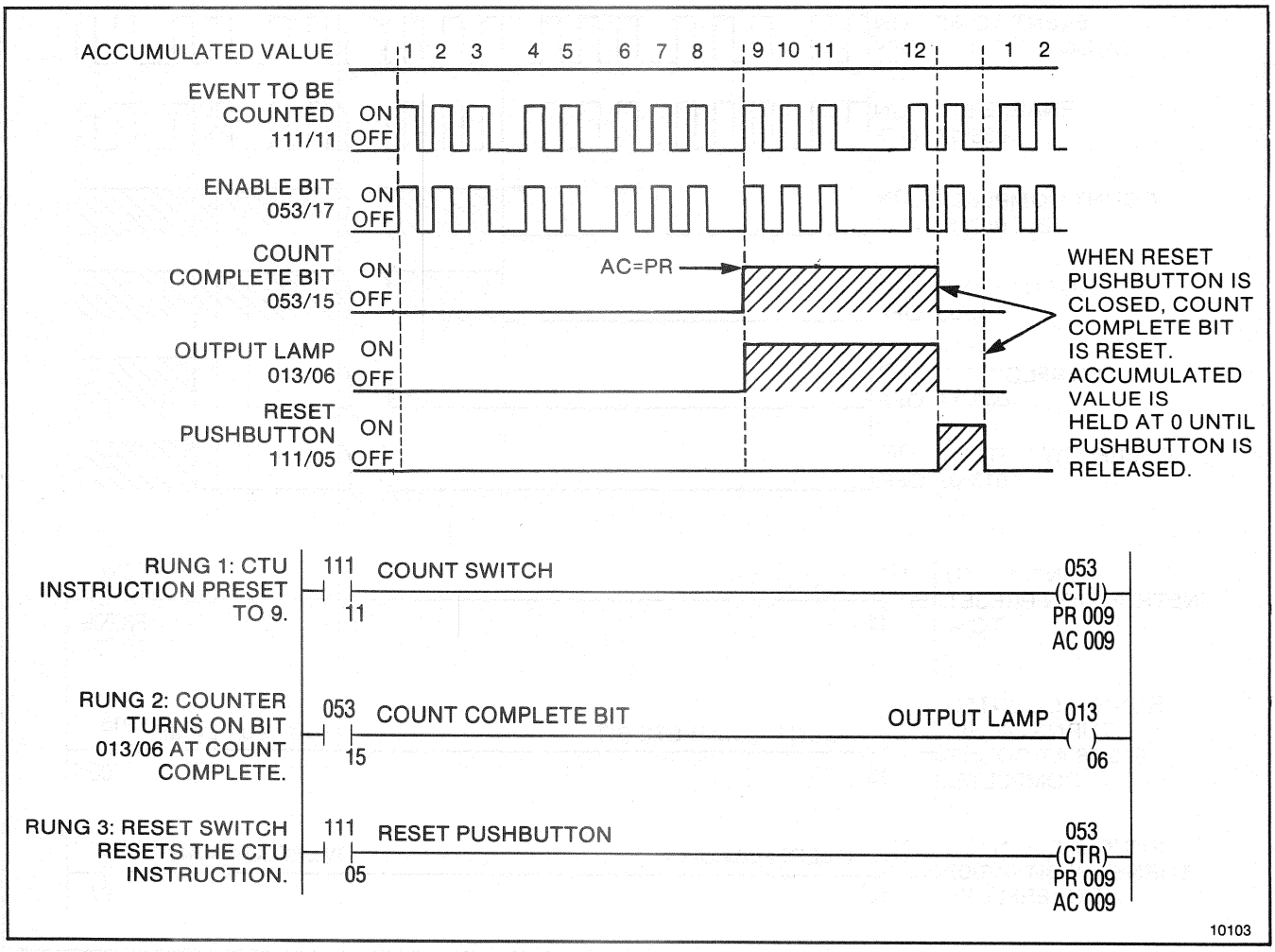


FIGURE 4-8 — UP-COUNTER with RESET Diagram & Programming

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." 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. Their addresses are assigned to the Accumulated areas of the Data Table. Valid addresses for Accumulated values are those which are even-hundreds numbers, such as 0XX, 2XX, 4XX, etc. Valid addresses for Preset values are 100_8 above the Accumulated values. When assigning an address to a Timer or Counter instruction, the 3rd digit from the right in the word address must be an even number.

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 sec., 0.1 sec. or 0.01 sec. is entered as [1][0], [0][1], or [0][0] respectively.

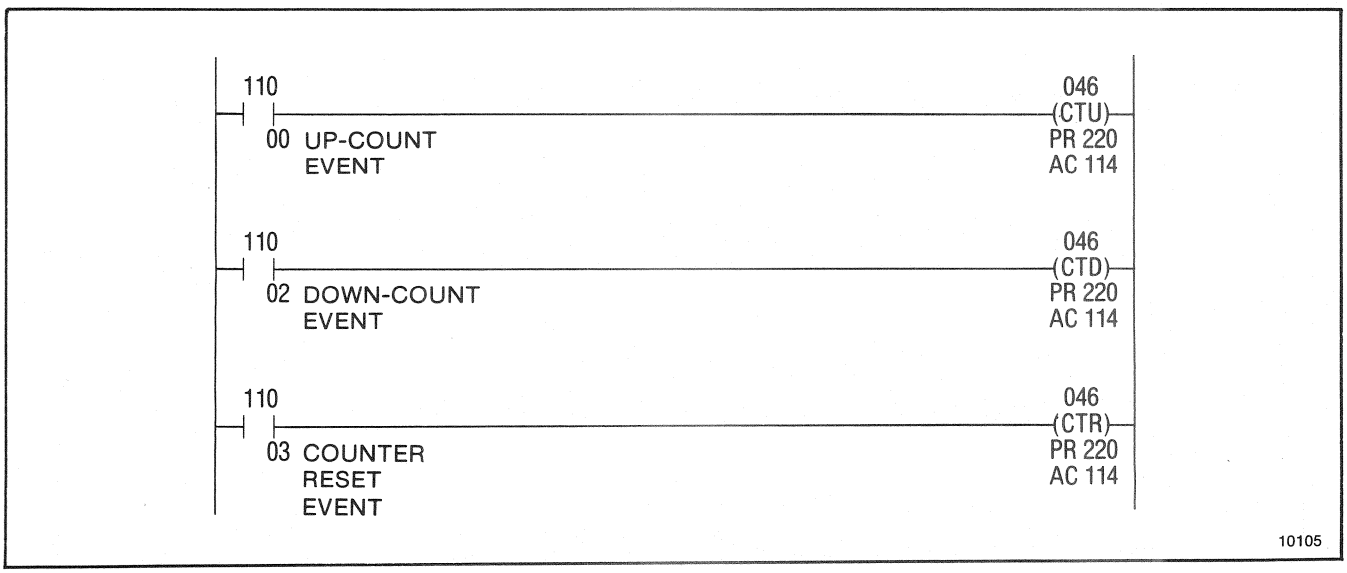


FIGURE 4-9 — UP/DOWN COUNTER Example

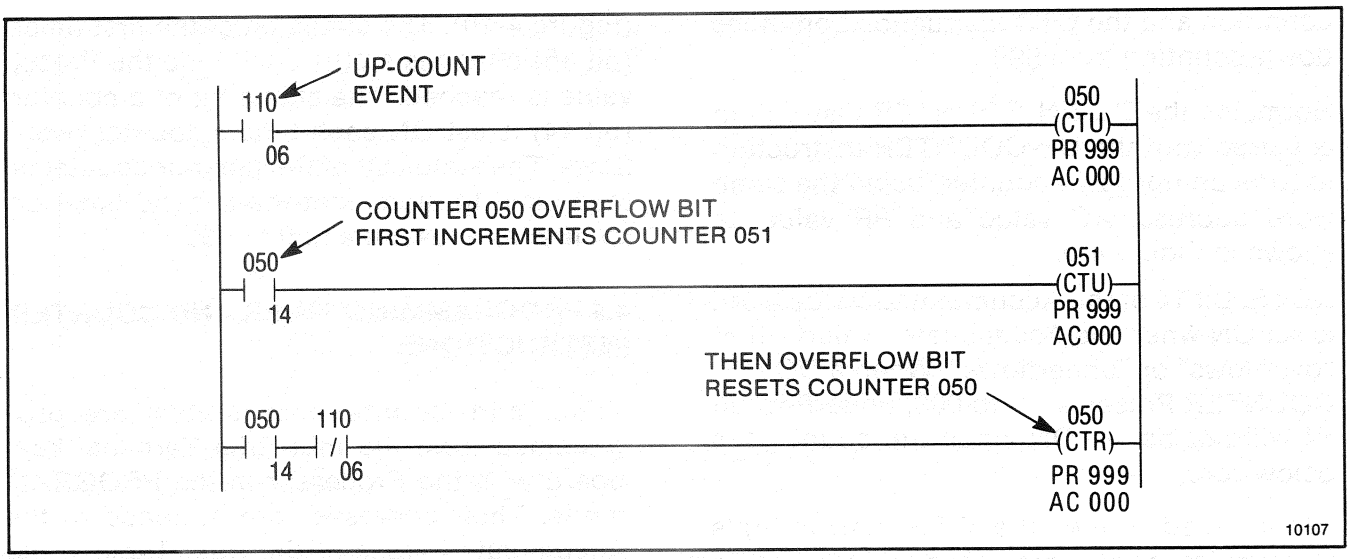


FIGURE 4-10 — Cascading Counters Example

Counter instructions are programmed by entering the word address of the Accumulated value, a Preset value, and if desired, an Accumulated value.

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.

The default word address will be 3 or 4 digits, 030 or 0030. Unlike bit instructions, the [EXPAND ADDR] key is not required. Instead, the Industrial Terminal automatically enters a 3 or 4-digit default word address depending on the Data Table size. When a 4-digit word address is displayed and a 3-digit word address is required, a leading zero must be entered before the word address.

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 areas of the Data Table. To determine which addresses are valid Accumulated areas, the 3rd digit from the right in the word address must be an even number.

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.

The word address displayed will be 3 or 4 digits long depending on the Data Table size. When entering the word address, use a leading zero if necessary.

KEYTOP SYMBOL	INSTRUCTION NAME	1770-T3 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	-(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. The Preset and Accumulated Values are automatically entered by the Industrial Terminal. When the rung is TRUE, the Accumulated Value and status bits are reset to 000.

TABLE 4-2 — Counter Instructions

Note: The Counter word address, XXX, is assigned to the counter Accumulated areas of the Data Table. To determine which addresses are valid Accumulated areas, the 3rd digit from the right in the word address must be an even number.

The word address displayed will be 3 or 4 digits long depending on the Data Table Size. When entering the word address, use a leading zero if necessary.

KEYTOP SYMBOL	INSTRUCTION NAME	1770-T3 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. The 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 lower 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).

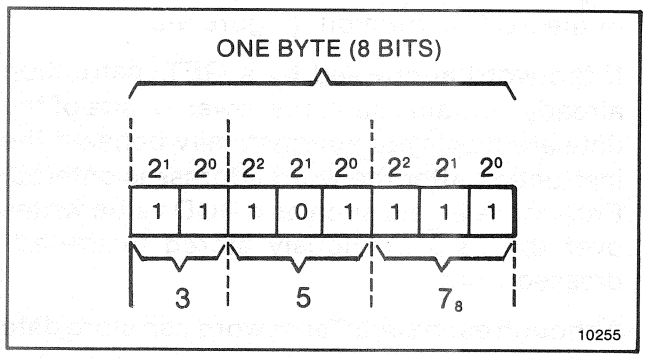


FIGURE 5-2 — Octal Representation

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

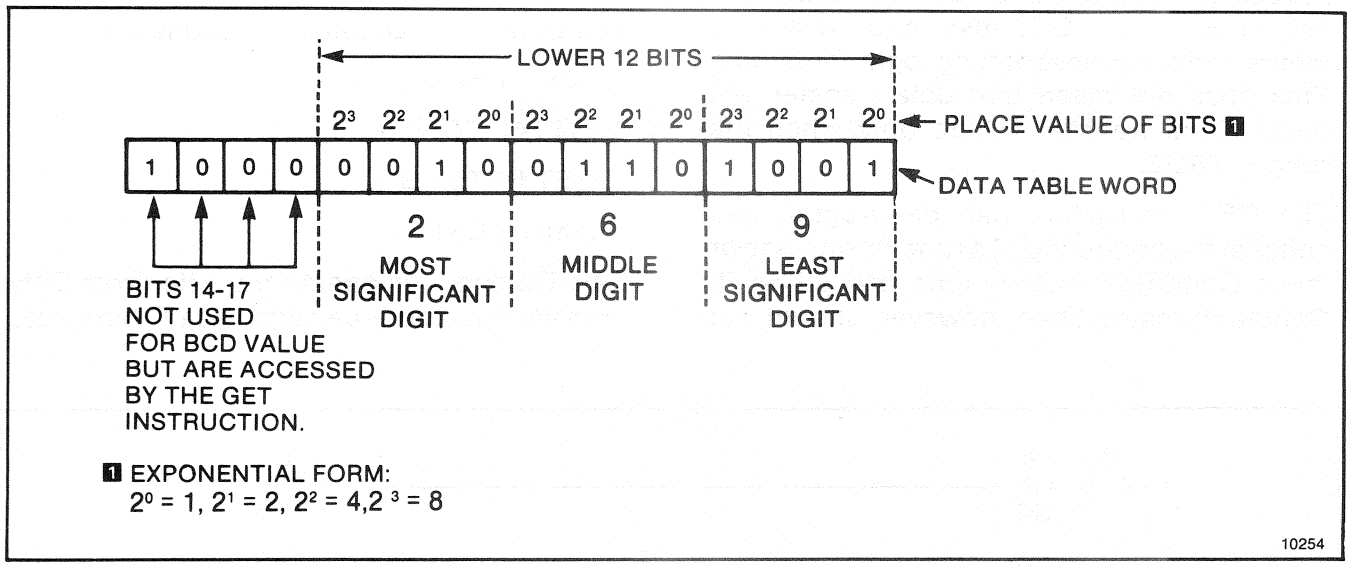


FIGURE 5-1 — BCD Word Format

5.1 DATA TRANSFER INSTRUCTIONS

There are two Data Transfer instructions. They are:

GET -|G|-

PUT -(PUT)-

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, TEST or RUN/PROGRAM 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. 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

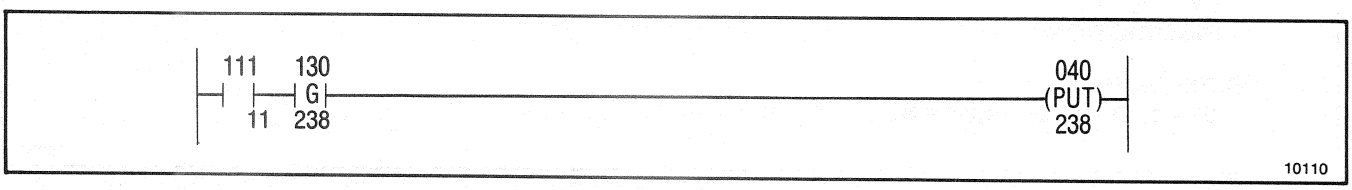


FIGURE 5-3 — GET and PUT Instructions

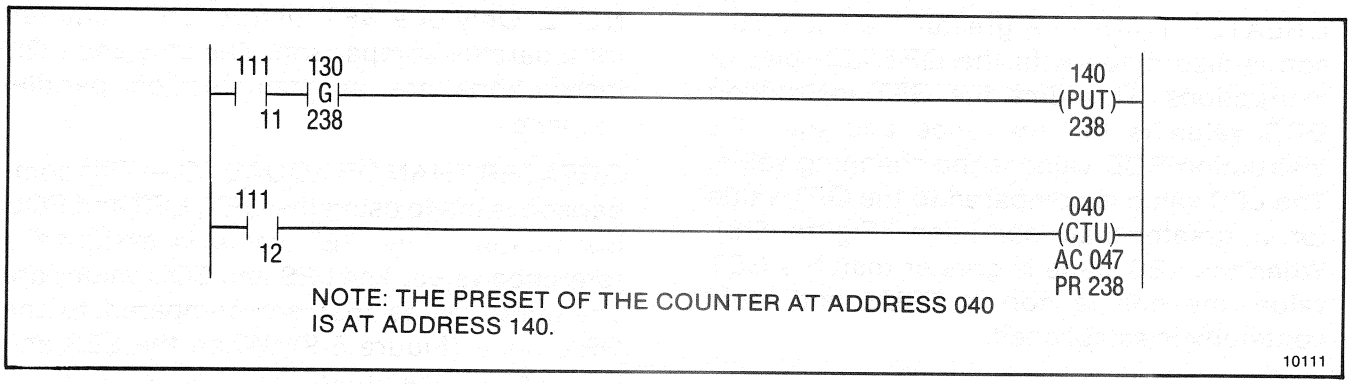


FIGURE 5-4 — Changing a Counter Preset

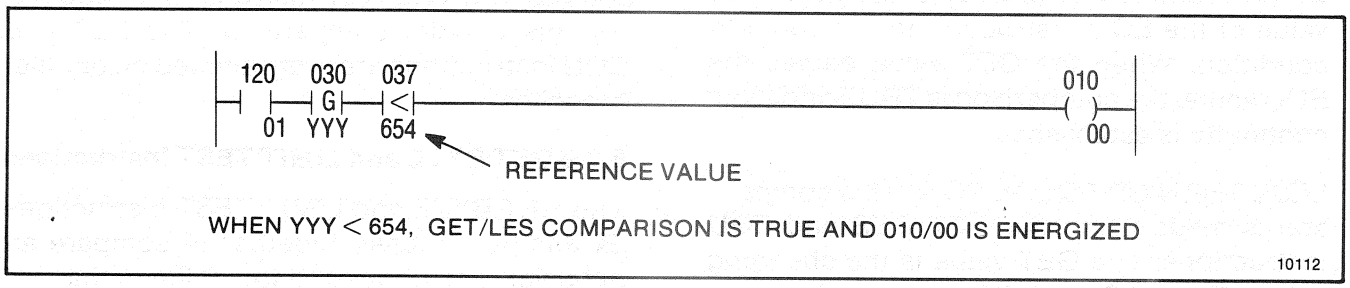


FIGURE 5-5 — LESS THAN Comparison

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. 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 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

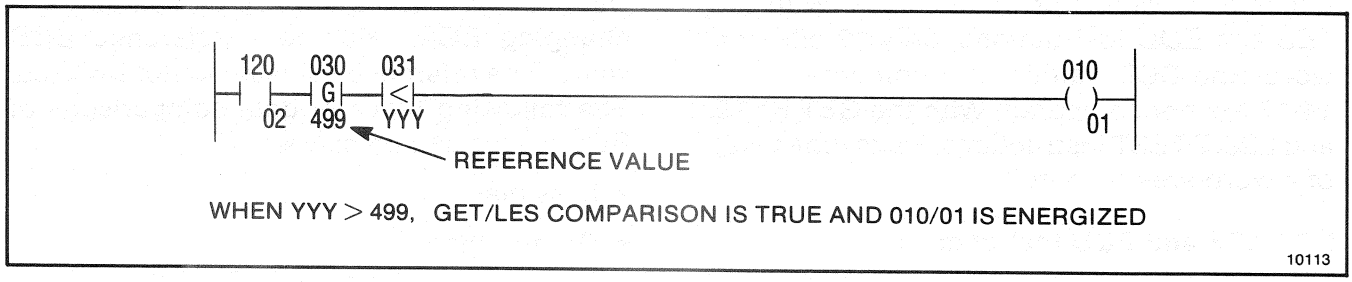


FIGURE 5-6 — GREATER THAN Comparison

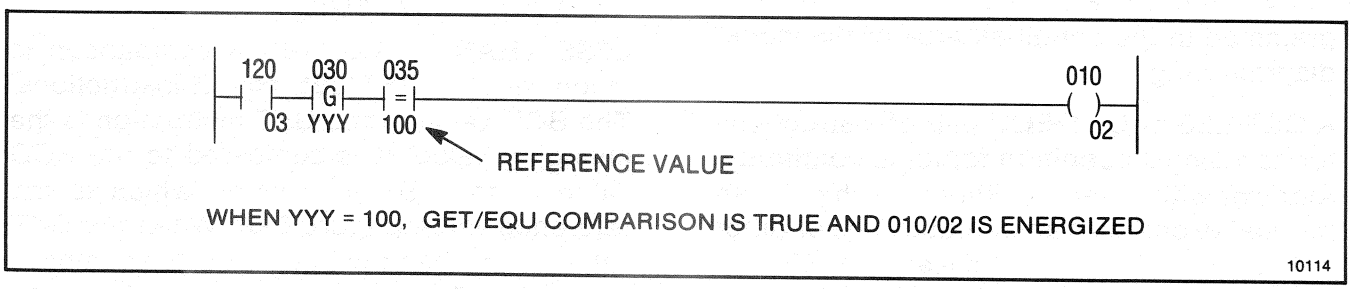


FIGURE 5-7 — EQUAL TO Comparison

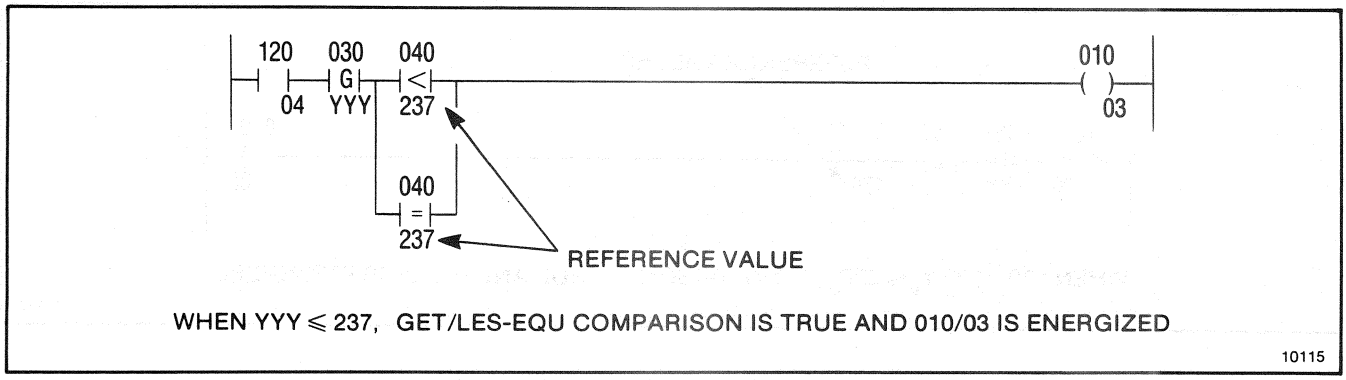


FIGURE 5-8 — LESS THAN or EQUAL TO Comparison

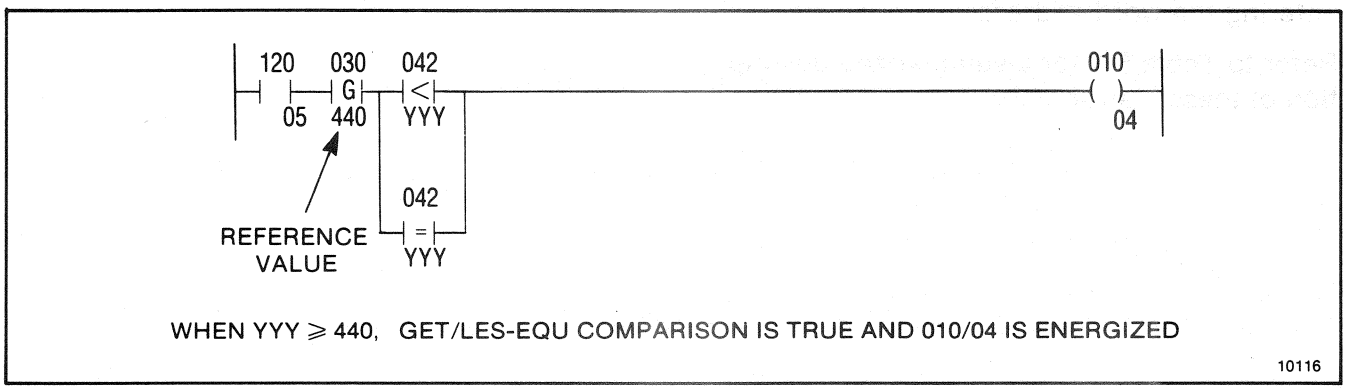


FIGURE 5-9 — GREATER THAN or EQUAL TO Comparison

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.

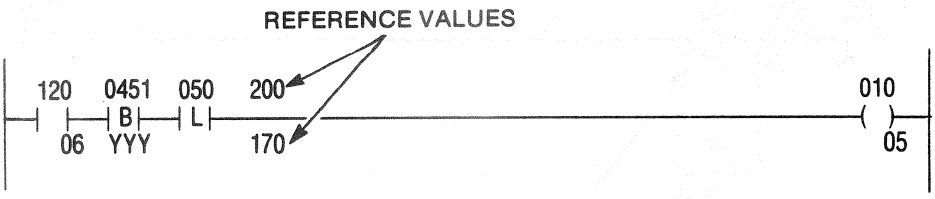
The LIMIT TEST instruction addresses one Data Table word that stores both the upper and lower limits. The upper limit is stored in the upper byte and the lower limit is stored in the lower byte. The upper byte of word 045 would be addressed as 0451. See Figure 5-10.

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.

The default word address can appear as 3 or 4-digits, 010 or 0010, depending on the Data Table size. The Industrial Terminal will automatically display a 4-digit word address provided the Data Table has been expanded to that size. When a 4-digit default address is displayed and a 3-digit word address is re-



WHEN $170_8 \leq YYY_8 \leq 200_8$, COMPARISON IS TRUE AND 010/05 IS ENERGIZED

10117

FIGURE 5-10 — GET BYTE/LIMIT TEST Comparison

quired, a leading zero must be entered before entering the word address.

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

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.

Word address displayed will be either 3 or 4 digits depending upon the Data Table size. When entering the word address, use a leading zero if necessary.

KEYTOP SYMBOL	INSTRUCTION NAME	1770-T3 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 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.
-[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 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.

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

KEYTOP SYMBOL	INSTRUCTION NAME	1770-T3 DISPLAY	DESCRIPTION
-[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-2/15 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)-

DIVIDE -(:)-(:)-

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, TEST or RUN/PROGRAM 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.

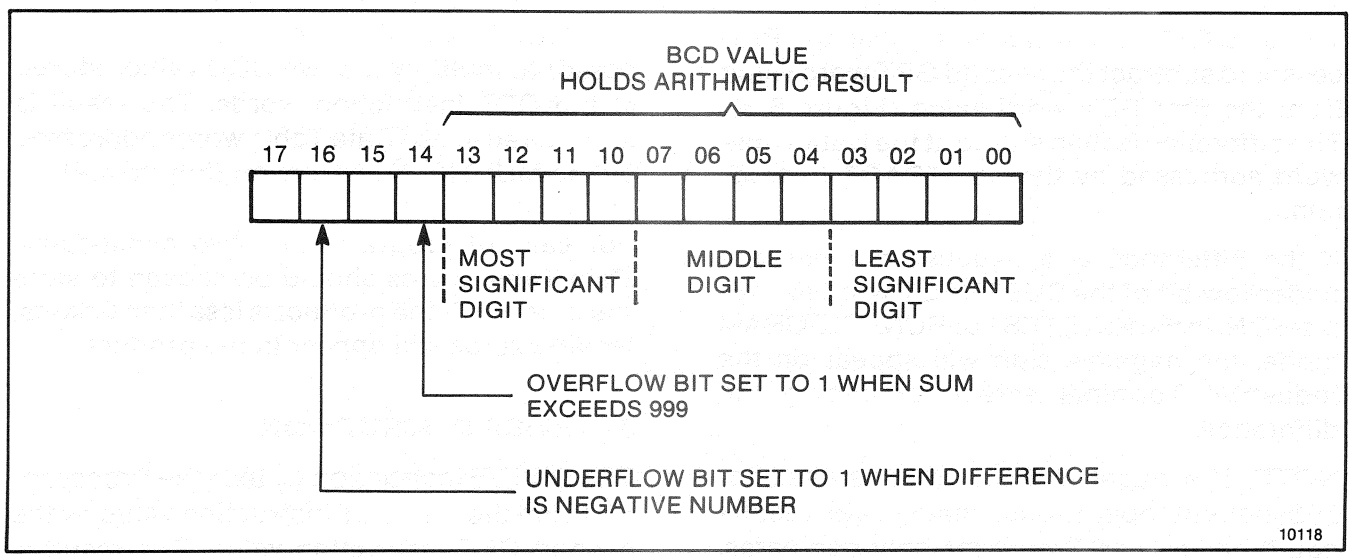


FIGURE 6-1 — Arithmetic Instruction Word

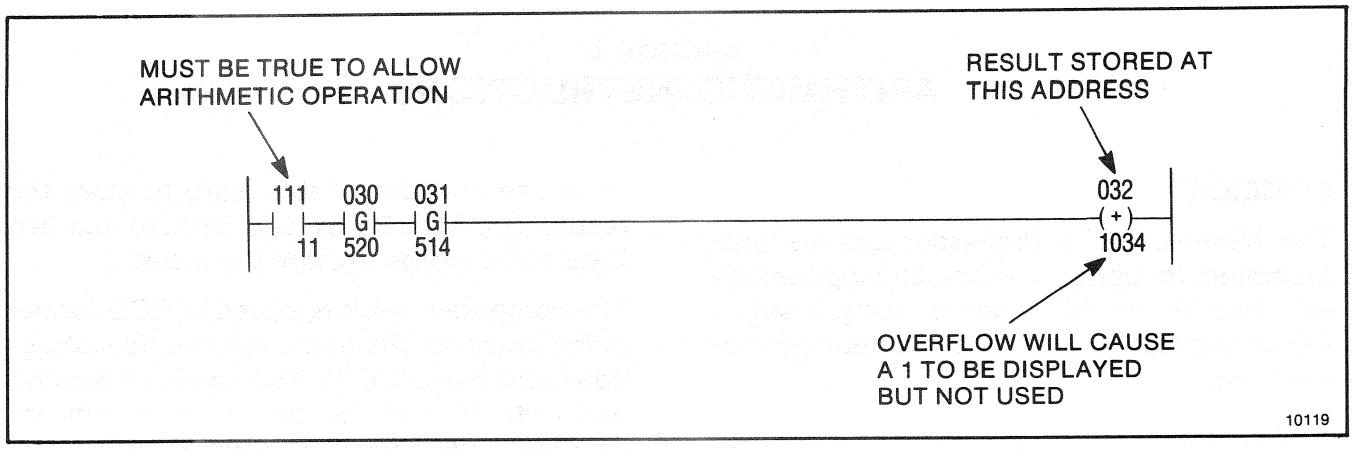


FIGURE 6-2 — ADD Instruction

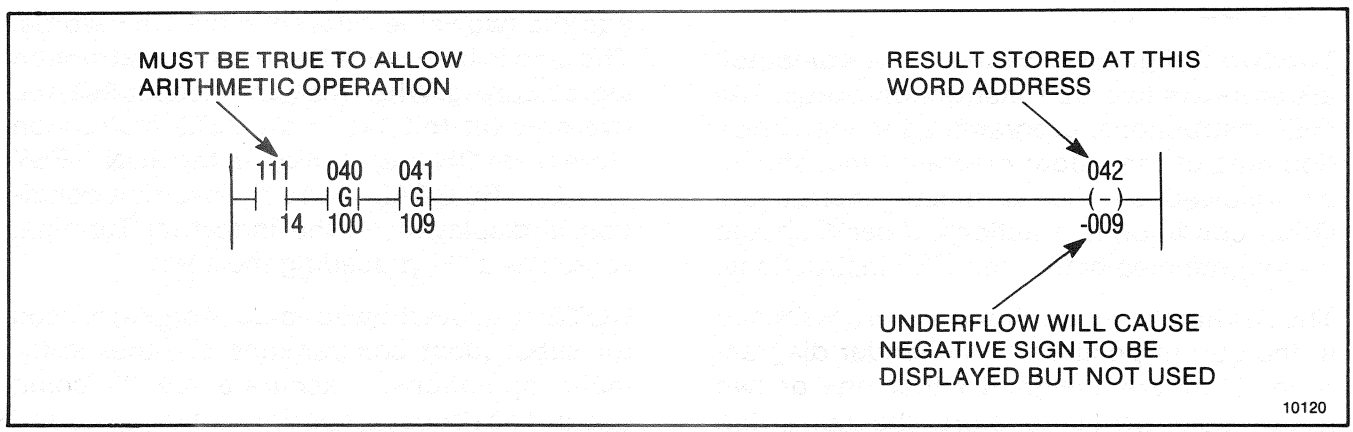


FIGURE 6-3 — SUBTRACT Instruction

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). 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, TEST or RUN/PROGRAM 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

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

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

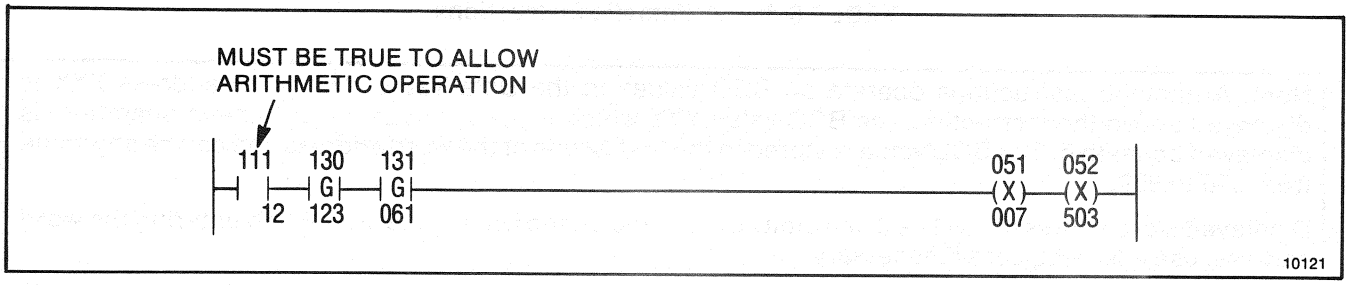


FIGURE 6-4 — MULTIPLY Instruction

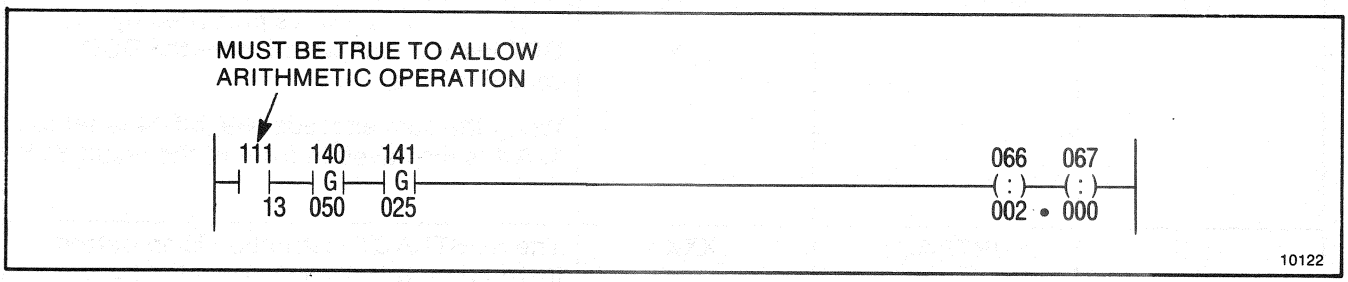


FIGURE 6-5 — DIVIDE Instruction

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 automatically 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 $0 \div 0 = 1.000$.)

6.5 PROGRAMMING ARITHMETIC INSTRUCTIONS

Arithmetic instructions are programmed from the Industrial Terminal keyboard with the Mini-PLC-2/15 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.

The default word address will appear as 3 or 4 digits, 010 or 0010, depending on Data Table size. When the Data Table is expanded to a 4-digit word address, the 4-digit default address will be displayed. To enter a 3-digit word address when 4 digits are displayed, a leading zero must be entered before entering the word address.

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 first 12 bits of the word address and can be any value from 000 to 999.

Displayed word addresses will be 3 or 4 digits depending on the Data Table size. When entering the word address, use a leading zero if necessary.

KEYTOP SYMBOL	INSTRUCTION NAME	1770-T3 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. 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	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	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 RESET 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-2/15 Assembly and Installation Manual (Publication 1772-803). The purpose of the 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 instruction.

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

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.

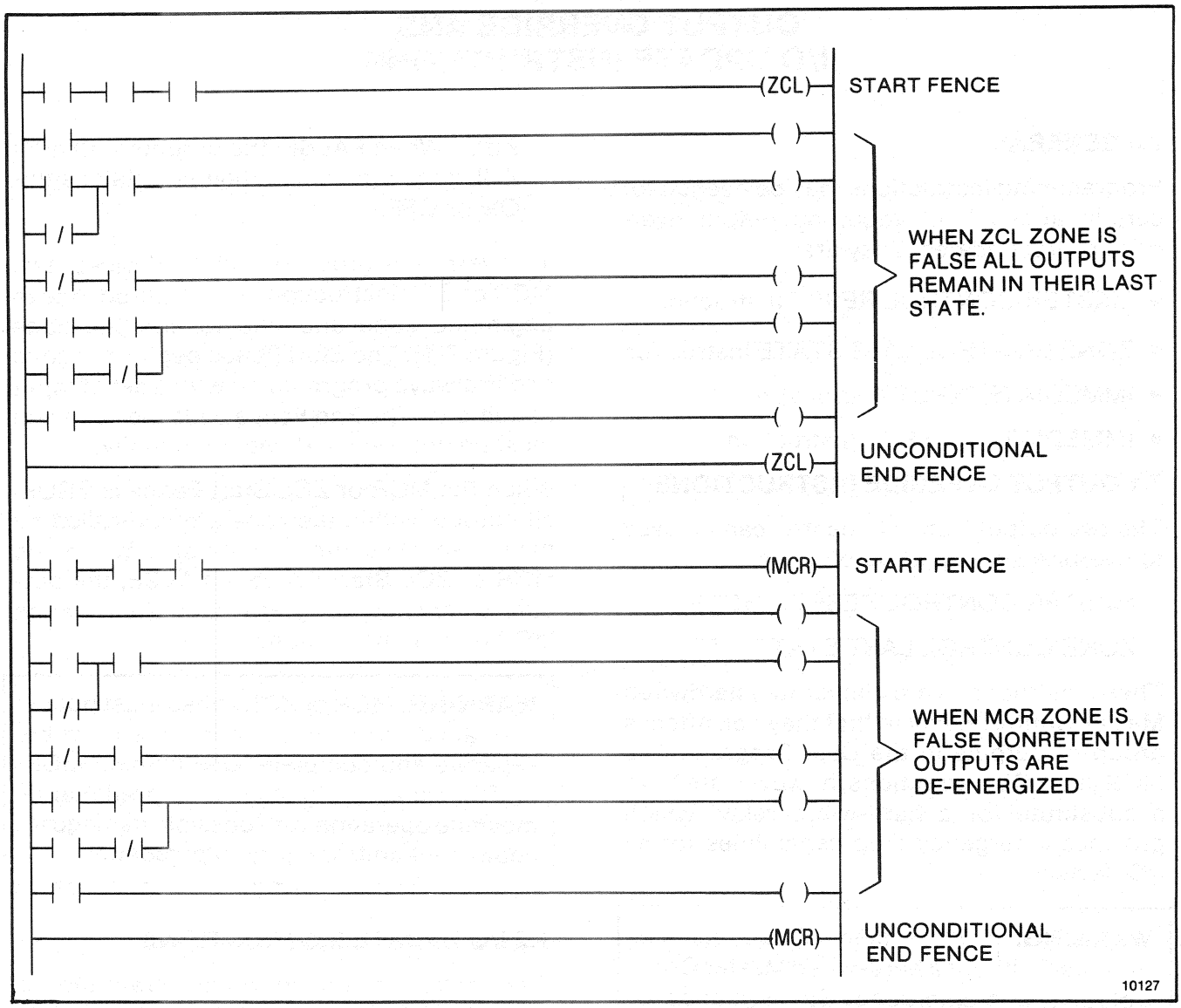


FIGURE 7-1 — MCR and ZCL Zone Programming

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-2/15 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 serially. The time required to complete the I/O scan and 1K of instructions is typically 4 msec plus 18 msec, respectively.

It is clear that typically 40 msec may pass before I/O data is updated in an 2K system. The purpose of IMMEDIATE I/O instructions

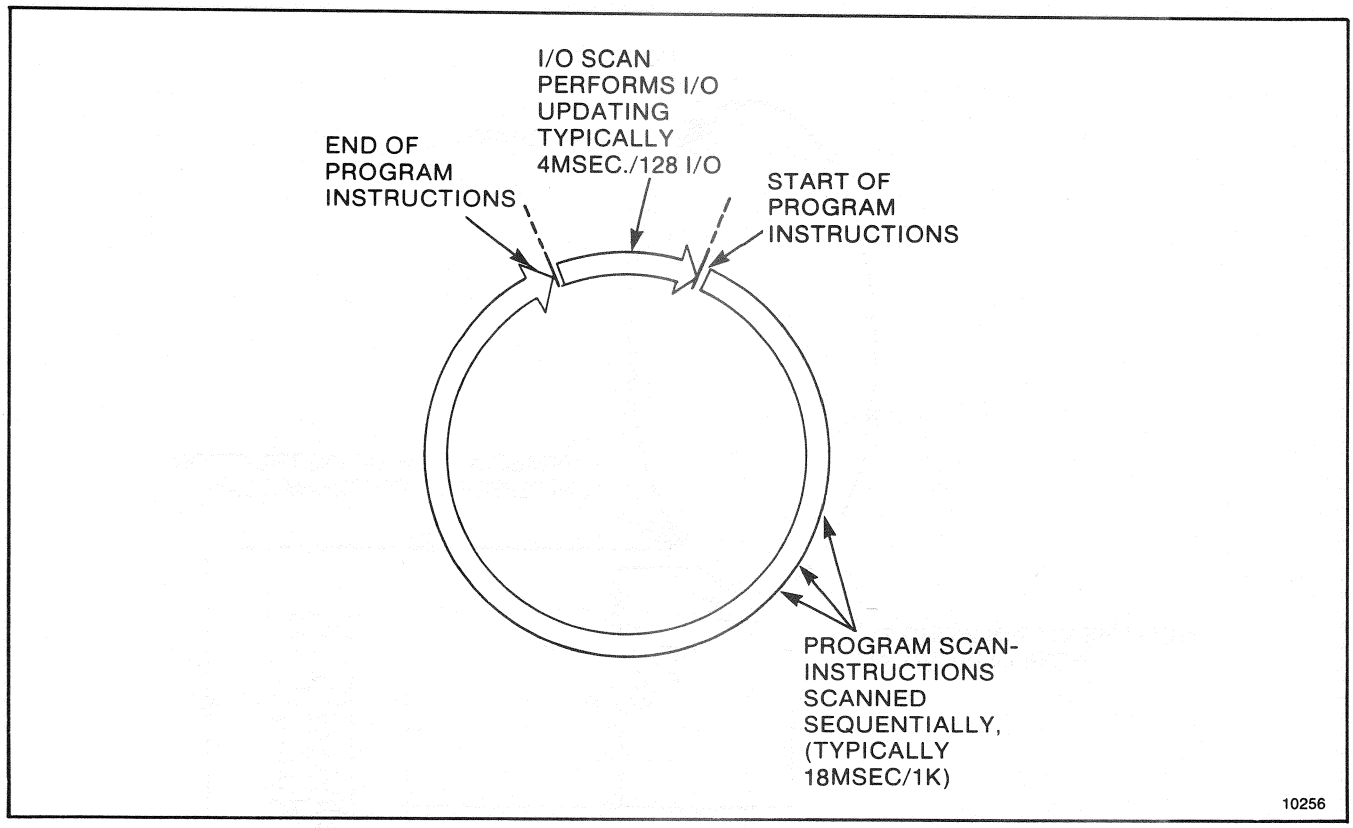


FIGURE 7-2 — Scan Sequence

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.

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.

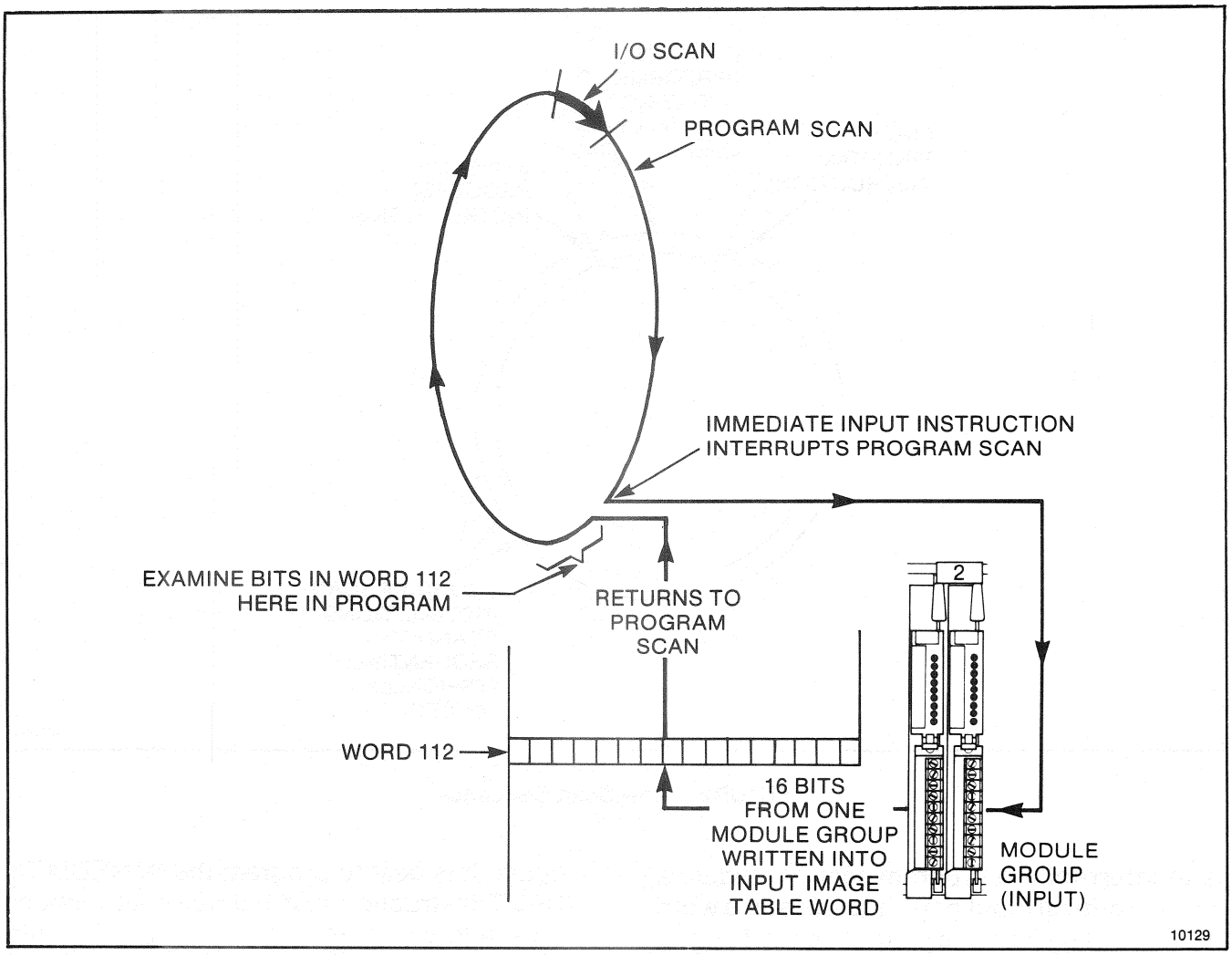


FIGURE 7-3 — IMMEDIATE INPUT Instruction

The IMMEDIATE OUTPUT instruction should be programmed just after the rungs that control 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.

The default word address will automatically appear as 3 or 4 digits, 010 or 0010, depending on the Data Table size. When the Data Table is expanded to a 4-digit word address, the 4-digit default address will be displayed. To enter a 3-digit address when 4 digits are displayed, a leading zero must be entered before entering the word address.

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

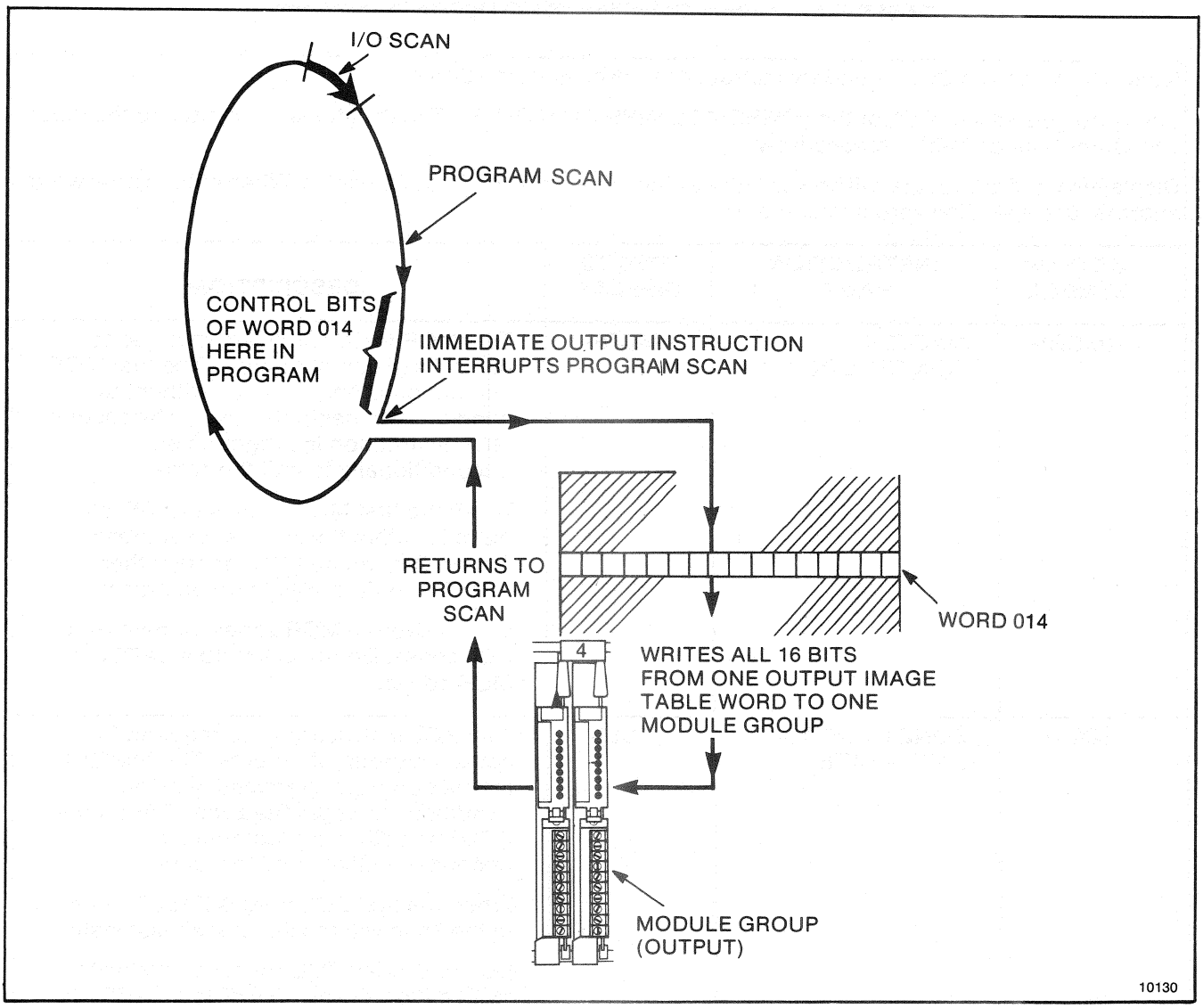


FIGURE 7-4 — IMMEDIATE OUTPUT Instruction

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.

Displayed word addresses will be 3 or 4 digits long, depending on Data Table size. When entering the word address, use a leading zero if necessary.

KEYTOP SYMBOL	INSTRUCTION NAME	1770-T3 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 JUMP to a LABEL in 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. Do not JUMP to a LABEL in ZCL 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.

Section 8
JUMP INSTRUCTIONS AND SUBROUTINE PROGRAMMING

8.0 GENERAL

The Jump instructions and subroutine programming provide programming flexibility and efficiency. Four instructions are used to implement program jumps and subroutines:

JUMP -(JMP)-

LABEL -LBL-

JUMP TO SUBROUTINE -(JSR)-

RETURN -(RET)-

The JUMP and LABEL instructions allow portions of a program to be selectively jumped over in order to reduce scan time. When more than one section of program each controls a separate process or operation, these instructions allow the required section to be executed as needed.

The JUMP TO SUBROUTINE, LABEL and RETURN instructions are used together to access reserved sections of program called subroutines. A subroutine can be called upon repeatedly from selected points in the Main Program. Subroutines can be used to conserve memory in applications where repetitive programming is required, or when sections of User Program do not need to be executed each scan.

This section will describe how Jump instructions and subroutine programming are used and how they direct the path of the program scan through the Main Program and the Subroutine Area.

8.1 SUBROUTINE AREA

The area reserved for subroutines is located in memory between the Main Program and the Message Store area. Its boundary is displayed as SUBROUTINE AREA and serves as the end of program statement for the Main Program. Subroutines are not scanned by the Processor until directed to do so by the JUMP TO SUBROUTINE instruction.

The Subroutine Area is established by placing the cursor on the last instruction in Main Program and pressing the key sequence [SHIFT][SBR]. The boundary marker SUBROUTINE AREA will appear. This instruction uses one (1) word of memory. It can be programmed only once and cannot be removed except by clearing the entire Subroutine Area or the entire memory. The SUBROUTINE AREA instruction cannot be inserted between two rungs but must be programmed as the last instruction in the Main Program. Once the Subroutine Area has been established, rungs can be added by placing the cursor on the SUBROUTINE AREA instruction, pressing [↓], then entering the rungs.

Up to eight (8) subroutines can be programmed in the Subroutine Area. Each begins with a LABEL instruction and ends with a RETURN instruction to the Main Program. The RETURN should be an unconditional rung. Figure 8-1 shows a representative Subroutine Area. Subroutines can not be nested, meaning that a JUMP TO SUBROUTINE instruction can not be programmed in the Subroutine Area.

8.2 LABEL INSTRUCTION

The LABEL instruction -LBL- shown in Figure 8-2 is the target destination for both the JUMP and JUMP TO SUBROUTINE instructions. LABELS are assigned octal identification numbers from 00-07. The LABEL identification number must be the same as that of the JUMP and/or JUMP TO SUBROUTINE instruction with which it is used. A LABEL instruction can be defined only once, meaning that a LABEL with a given identification number can appear only once. However, a LABEL instruction can be the target of JUMP or JUMP TO SUBROUTINE instructions from more than one location.

The LABEL instruction is always logically true. It should be programmed as the first Condition instruction in the rung. If condi-

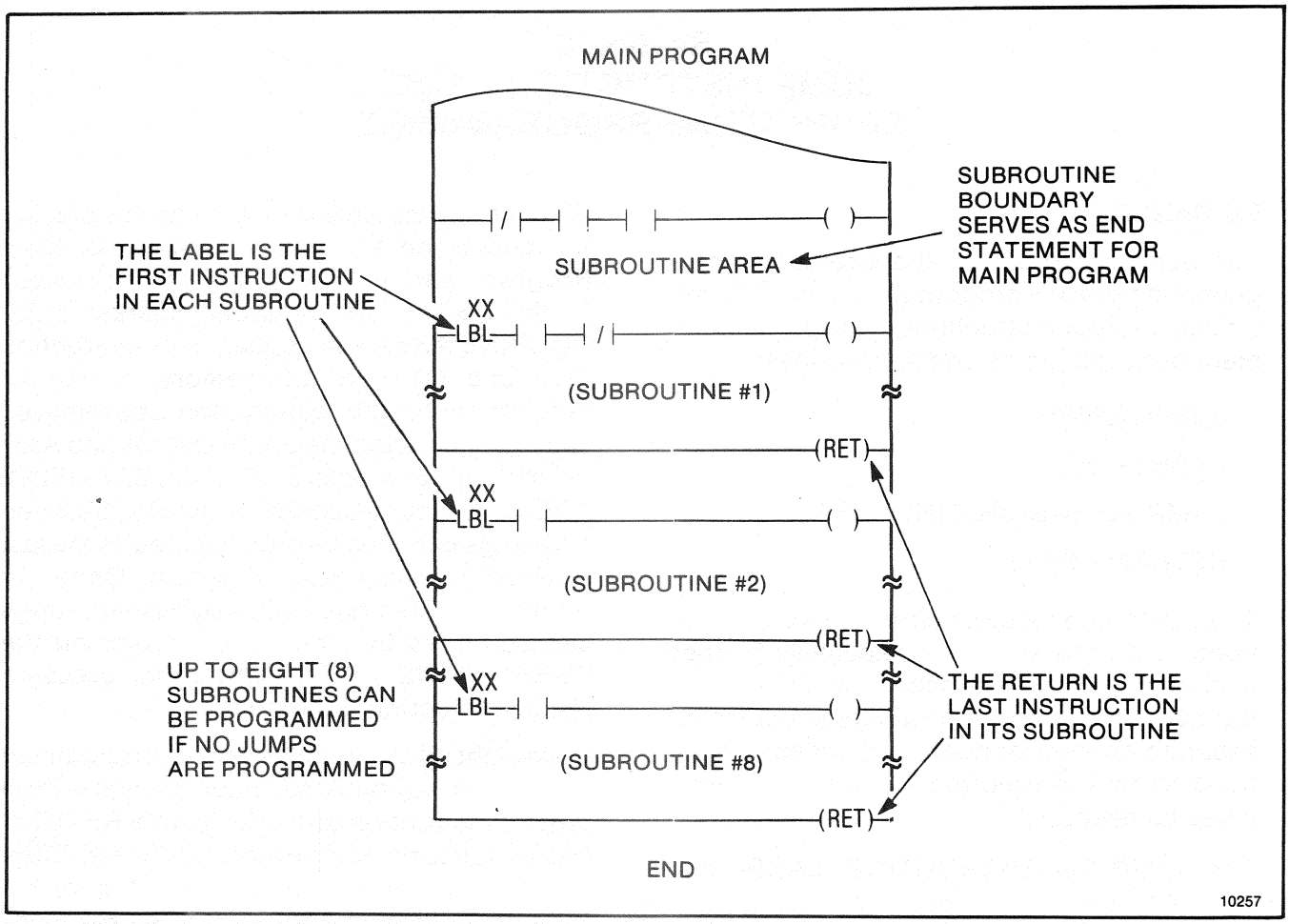


FIGURE 8-1 — Representative Subroutine Area

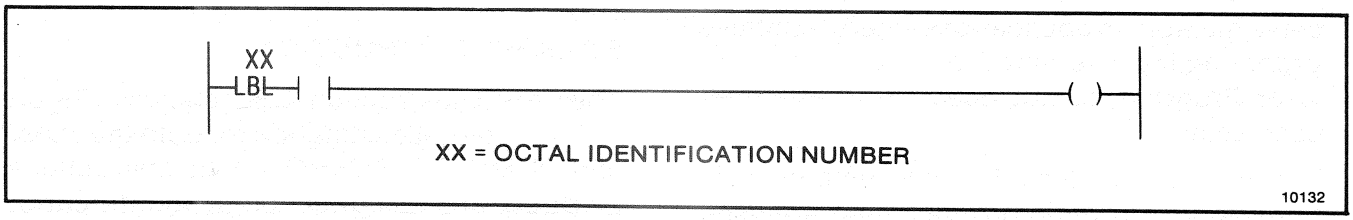


FIGURE 8-2 — LABEL Format

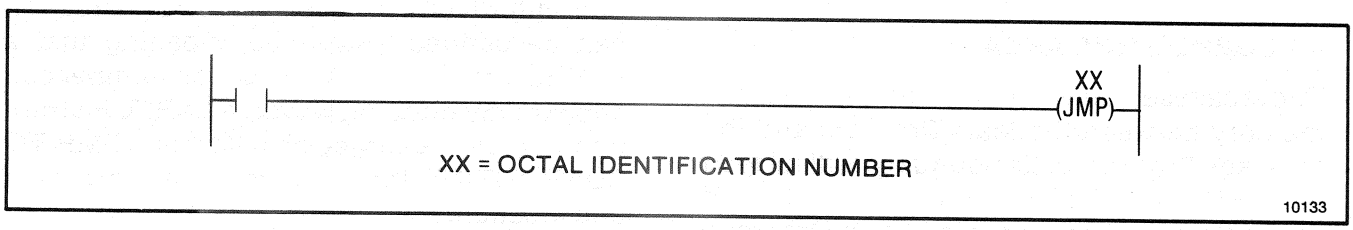


FIGURE 8-3 — JUMP Format

tions precede a LABEL instruction in a rung, they will be ignored by the Processor during a jump operation.

8.2.1 LABEL Instruction Characertistics

- Rung conditioning instruction, always logically TRUE.
- Up to 8 LABELS available, identified by a 2-digit octal identification number, 00-07.
- Each LABEL can be defined and used only once.
- Can be the target of multiple JUMP or JUMP TO SUBROUTINE instructions.
- Uses one word of memory.

WARNING: Do NOT place a LABEL instruction in a ZCL or MCR zone. When jumping over a Start Fence, the Processor will execute the program from the LABEL to the End Fence as if the Start Fence had been TRUE, i.e. outputs controlled by the rungs. The Start Fence may have been FALSE intending that all outputs within the zone be controlled by the Output Override instruction, i.e. OFF for MCR or last state for ZCL instructions. Unpredictable machine operation could occur with possible damage to equipment and/or injury to personnel. (See Section 7.1.)

8.2.2 Causes of Run-Time Errors

Misuse of the LABEL instruction will cause the Processor to fault due to the following Run-Time Errors:

- Multiple placement of the same LABEL number.
- Removing a LABEL instruction but leaving the JUMP or JUMP TO SUBROUTINE instruction(s) to which it was referenced.

8.3 JUMP INSTRUCTION

The JUMP instruction (JMP) shown in Figure 8-3 is an output instruction. It must have the same octal identification number from 00-07 as the LABEL instruction with which it is used. When its rung is TRUE, it instructs the Processor to jump forward in the Main Program to the LABEL instruction having the same identification number (See Figure 8-4). The Main Program is then executed from that point.

A JUMP instruction can be programmed anywhere within the Main Program or Subroutine Area. However, a JUMP instruction may not be programmed to cross the boundary between the Main Program and Subroutine Area and vice versa or to jump backwards in memory.

8.3.1 JUMP Instruction Characteristics

- Output Instruction
- Can jump 1 or more times to the LABEL instruction with the same identification number.
- Can be used in the Main Program or Subroutine Area.
- Uses 1 word of memory.

CAUTION: Allowances should be made for conditions which could be created by the use of the JUMP instruction. Jumped program rungs are not scanned by the Processor so that input conditions are not examined and outputs that are controlled by these rungs remain in their last state. Timers and counters cease to function. Critical rungs should be reprogrammed outside the jumped section of program.

8.3.2 Causes of Run-Time Errors

Misuse of the JUMP instruction will cause the Processor to fault due to the following Run-Time errors:

- Jumping from Main Program into the Subroutine Area or vice versa.
- Jumping backward
- Omitting the LABEL corresponding to the JUMP instruction.
- Jumping over a TEMPORARY END instruction

8.4 JUMP TO SUBROUTINE INSTRUCTION

The JUMP TO SUBROUTINE Instruction (JSR) shown in Figure 8-5 is an output instruction. It must have the same octal identification number from 00-07 as the LABEL instruction with which it is used. When its rung is TRUE it

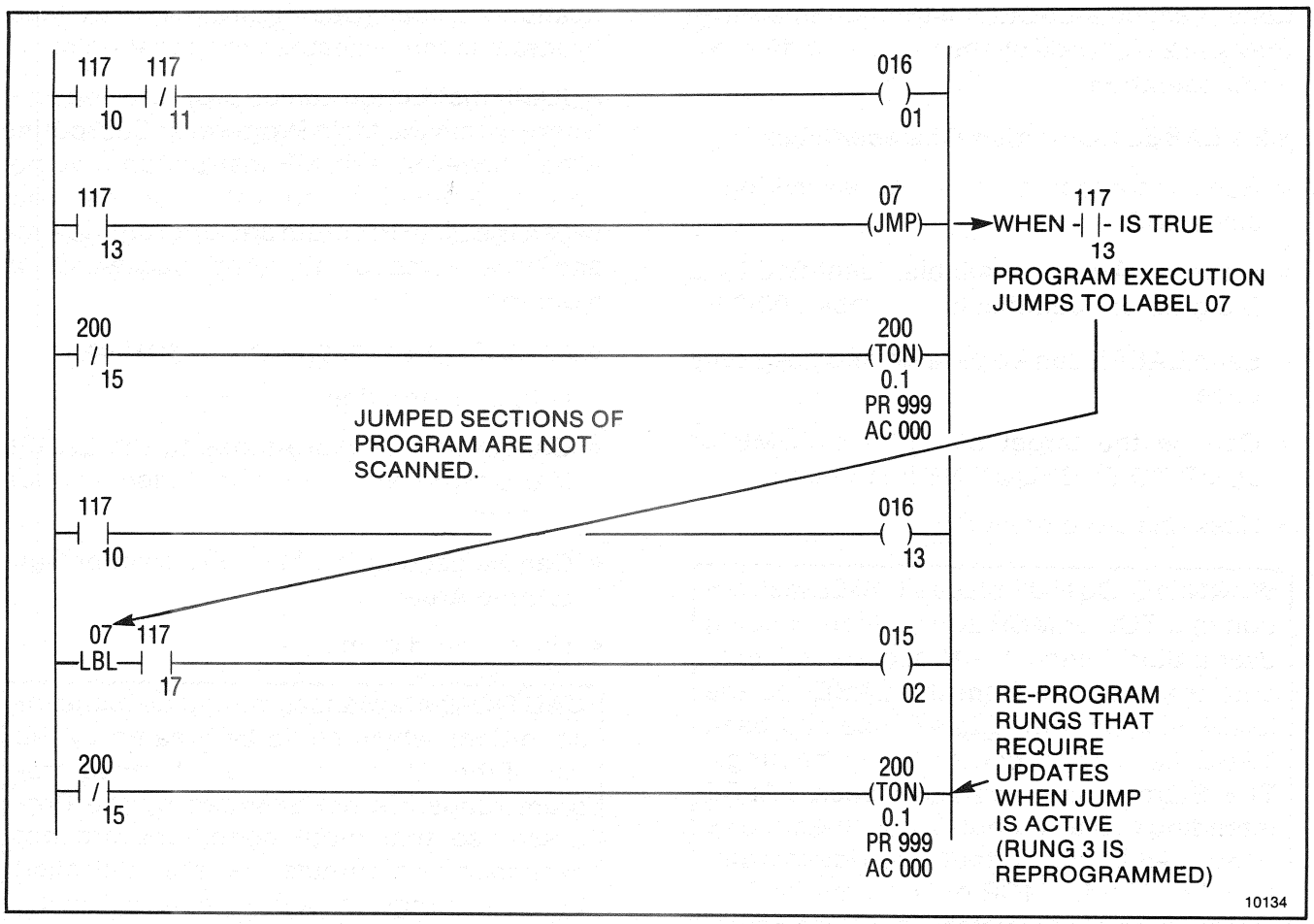


FIGURE 8-4 — JUMP to LABEL Operation

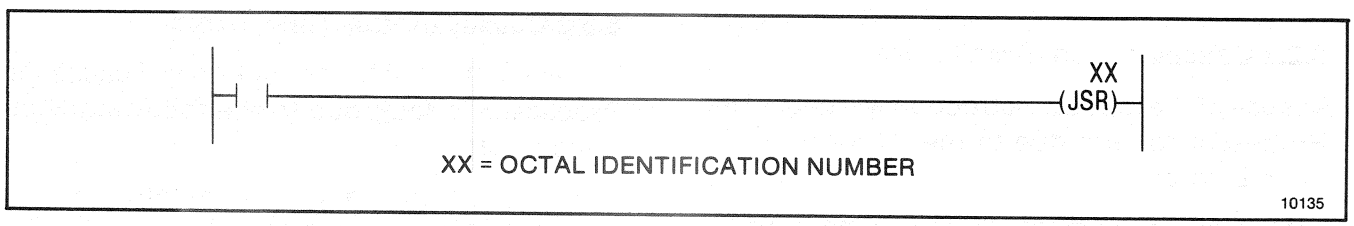


FIGURE 8-5 — JUMP-TO-SUBROUTINE Format

instructs the Processor to jump from the Main Program to the LABEL instruction having the same identification number in the Subroutine Area. See Figure 8-7. Subroutine execution begins at that point.

This instruction must always cause the Processor to cross the boundary from the Main Program to the Subroutine Area in the Mini-PLC-2/15 system.

8.4.1 JUMP TO SUBROUTINE Instruction Characteristics

- Output instruction
- Must always jump from Main Program into Subroutine Area.
- Can jump 1 or more times to the LABEL instruction with the same identification number.
- Uses 1 word of memory.

8.4.2 Causes of Run-time Errors

Misuse of the JUMP TO SUBROUTINE instruction will cause the Processor to fault due to the following Run-Time Errors:

- Using the JSR instruction in the Main Program to jump forward in the Main program.
- Using the JSR instruction in the Subroutine Area.
- Using the JSR instruction without a RETURN instruction.
- Omitting the LABEL corresponding to the JUMP TO SUBROUTINE instruction.
- Jumping over a TEMPORARY END instruction.

8.5 RETURN INSTRUCTION

The RETURN instruction (RET) shown in Figure 8-6 is an output instruction. It is used only in the Subroutine Area to terminate a subroutine and to return the Processor to the Main Program. It returns the Processor to the instruction immediately following the JUMP TO SUBROUTINE instruction that initiated the subroutine. Execution of the Main Program continues from that point. See Figure 8-7.

Note: Every subroutine must have a RETURN instruction. If not, the Processor will scan the subsequent subroutine(s) until a RETURN instruction is found. The RETURN should be programmed unconditionally.

The RETURN instruction does not have a user-assigned identification number because it can be paired with any one of several JUMP TO SUBROUTINE instructions as the result of multiple jumps to the Subroutine Area. This is illustrated in Figure 8-10.

8.5.1 RETURN Instruction Characteristics

- Output instruction
- Every subroutine must have a RETURN instruction.
- Should be used in an unconditional rung.
- Processor returns to the instruction located after the JSR instruction in the Main Program that initiated the subroutine.
- Uses 1 word of memory.
- Does NOT have an identification number.

8.5.2 Causes of Run-time Errors

Misuse of the RETURN instruction will cause the Processor to fault due to following Run-Time Errors:

- Processor finds no RETURN from the Subroutine Area.
- Using a RETURN instruction outside the Subroutine Area.

8.6 MULTIPLE JUMPS TO THE SAME LABEL

There can be many combinations of multiple jumps to the same label in the Main Program or Subroutine Area. Some of these combinations are illustrated in Figures 8-8, 8-9 and 8-10. The RETURN instruction (Figure 8-10) can return the Processor to any one of 3 different points (A, B, or C) in the Main Program depending on the path taken (a, b or c) to the Subroutine Area.

8.7 SUBROUTINE PROGRAMMING CONSIDERATIONS

If an output instruction in the Subroutine Area is left ON after the Processor has completed the program scan, the output will remain ON regardless of its rung condition. No action will be taken to control that output until the Processor is directed back to the Subroutine

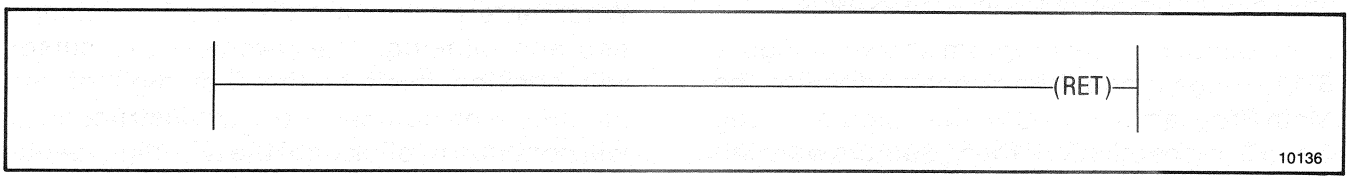


FIGURE 8-6 — RETURN Format

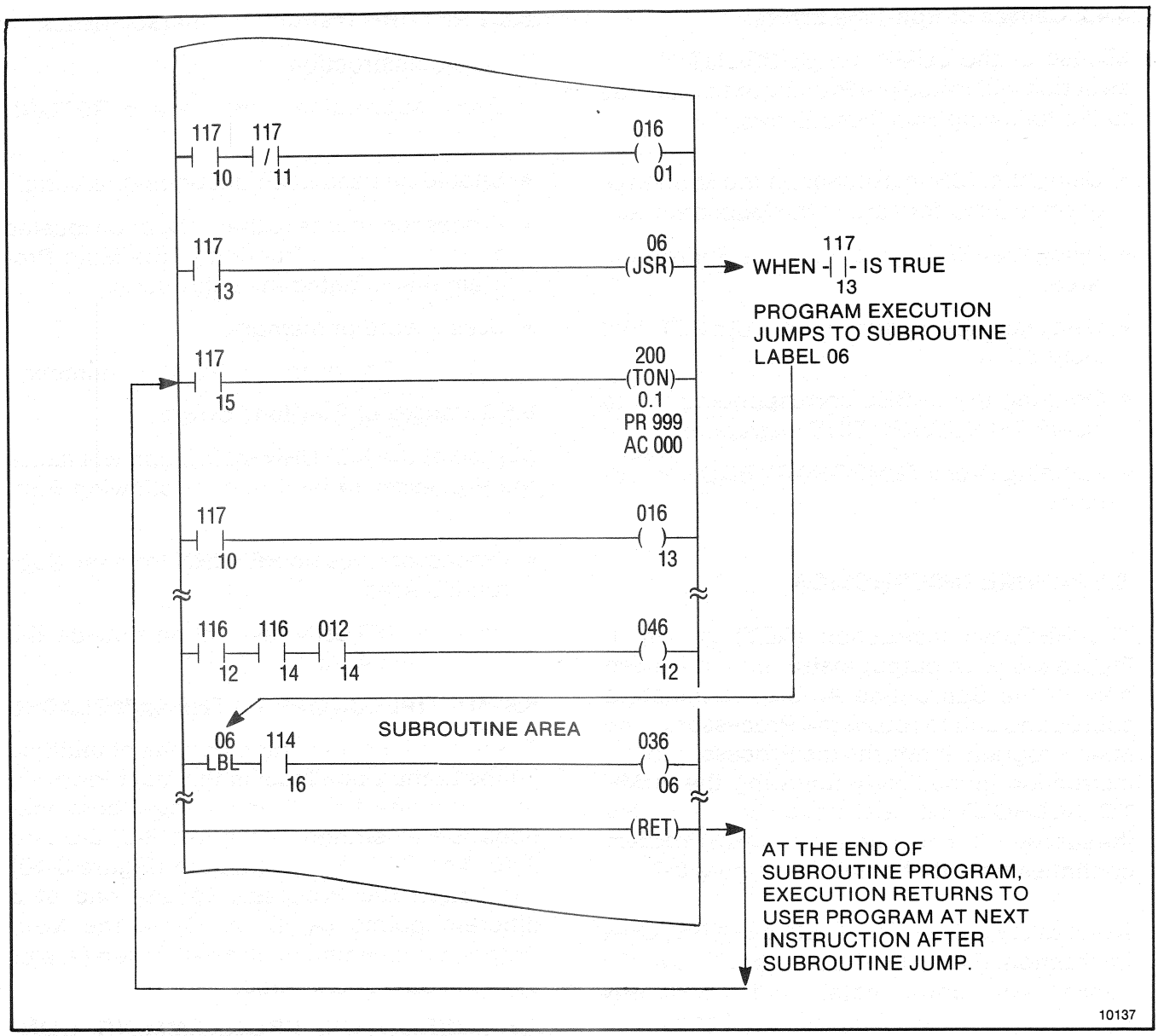


FIGURE 8-7 — JUMP TO SUBROUTINE LABEL Operation

Area, or a rung in the Main Program is executed which also controls that same output. Instructions in a Subroutine Area are acted on only during an actual scan of a subroutine. This also applies to status bits of subroutine timers, counters and block instructions.

For example, in the program shown in Figure 8-11, rungs 3 and 4 have been added to the Main Program to turn OFF the outputs of rung 5 and 6, respectively, if these outputs were left ON after the Processor completed the subroutine scan.

8.8 PROGRAMMING JUMP/SUBROUTINE INSTRUCTIONS

The Jump/Subroutine instructions are programmed from the Industrial Terminal Keyboard with the Processor in PROGRAM mode. When entered they are displayed as intensified and blinking. The reverse-video cursor will position itself at the first digit of the identification number above the instruction. It will continue to blink until the two-digit identification number is entered. The zero is auto-

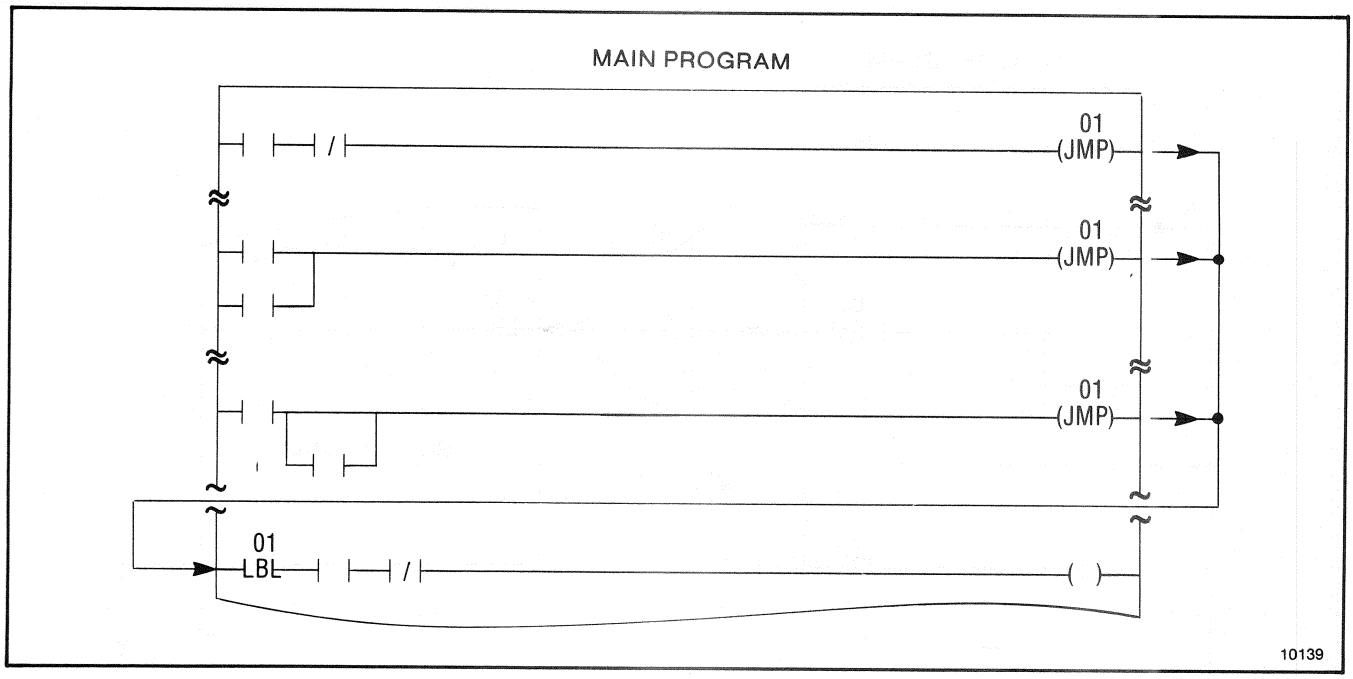


FIGURE 8-8 — Multiple JUMPS to LABEL in User Program

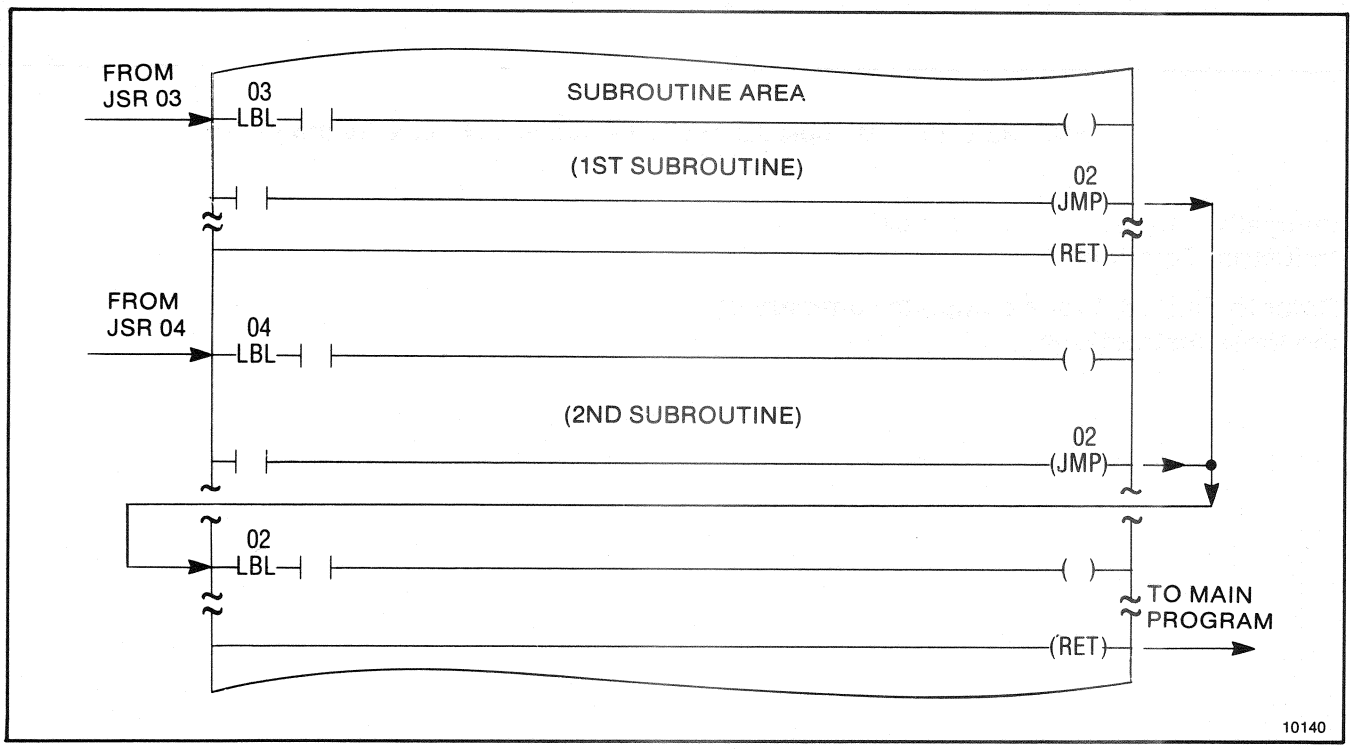


FIGURE 8-9 — Multiple JUMPS to LABEL in Subroutine Area

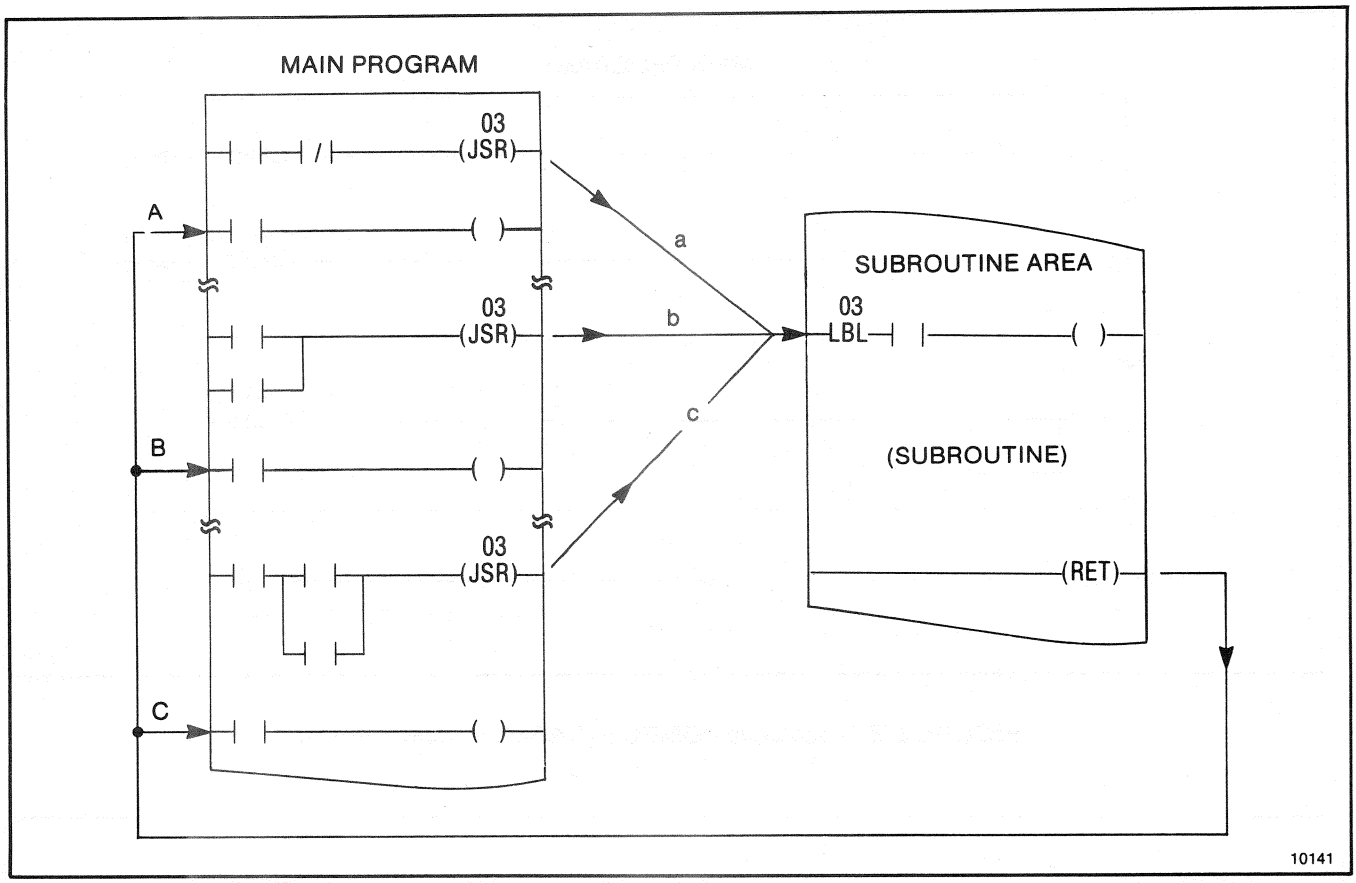


FIGURE 8-10 — Multiple JUMPS to LABEL and Multiple Return Paths

matically entered in the first position by the Industrial Terminal.

Refer to Table 8-1 for a complete summary of the these instructions.

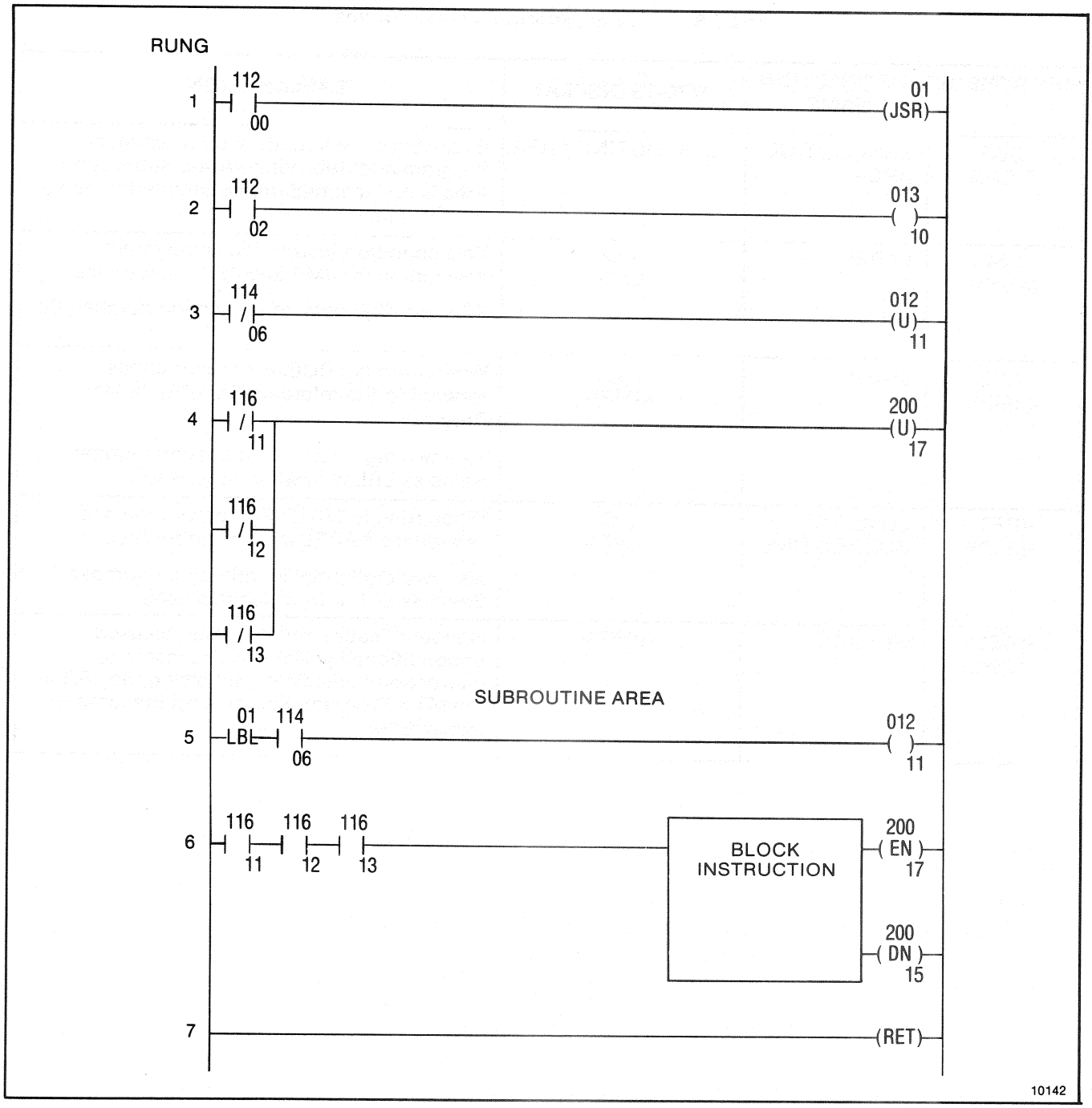


FIGURE 8-11 — Subroutine Programming

TABLE 8-1 — Jump/Subroutine Instructions

KEY SYMBOL	INSTRUCTION NAME	1770-T3 DISPLAY	EXPLANATION
SBR T.END	SUBROUTINE AREA	SUBROUTINE AREA	Establishes the boundary between Main Program and Subroutine Area. Subroutine Area is not scanned unless directed to do so by a JSR instruction.
LBL (JMP)-	LABEL	XX LBL-	This condition instruction is the target destination for JMP and JSR instructions. XX - two digit octal identification number, 00-07.
LBL **-(JMP)-**	JUMP	XX (JMP)-	When rung is TRUE, Processor jumps forward to the referenced LABEL in Main Program. XX - two digit octal identification number. Same as LBL with which it is used.
-(RET)- **-(JSR)-**	JUMP TO SUBROUTINE	XX -(JSR)-	When rung is TRUE, Processor jumps to referenced LABEL in Subroutine Area. XX - two digit octal identification number. Same as LBL with which it is used.
-(RET)- -(JSR)-	RETURN	-(RET)-	No identification number. Can be used unconditionall y. Returns Processor to instruction immediately following the JSR in the Main Program that initiated the jump to subroutine.

Section 9
FILES AND DATA MONITOR MODE

9.0 GENERAL

This section introduces concepts in two major areas:

FILES

DATA MONITOR MODE

Later sections of this manual are written with the assumption that the concepts and terms covered in this section have been thoroughly learned. In particular, do not procede into Sections 10 and 11, File and Sequencer Instructions, until this section is completely understood.

9.1 FILE CONCEPTS

The definition of a file, the pictorial representation of the File instruction and the modes of File instruction operation are covered in this section. The illustrations help to describe these concepts.

9.1.1 File Definition

A file is a group of consecutive Data Table words used to store information. It is defined by a counter and the starting word address of the file. The counter has two functions, (a) it defines the number of words in the file (File Length) with its Preset value and (b) it points to a particular word in the file (Position) with its Accumulated value. The counter address is also referred to as the instruction address. It is the address used by the Processor to search for the instruction. The words in the file must be located one after the other. A file can have from one (1) to a maximum of 999 words in length. The first word in a file is defined as word 1 and is located at position 001.

The structure of a file is presented in Figure 9-1. This figure illustrates a 12-word file starting at word address 600_8. The counter at address 200_8 has an Accumulated value of

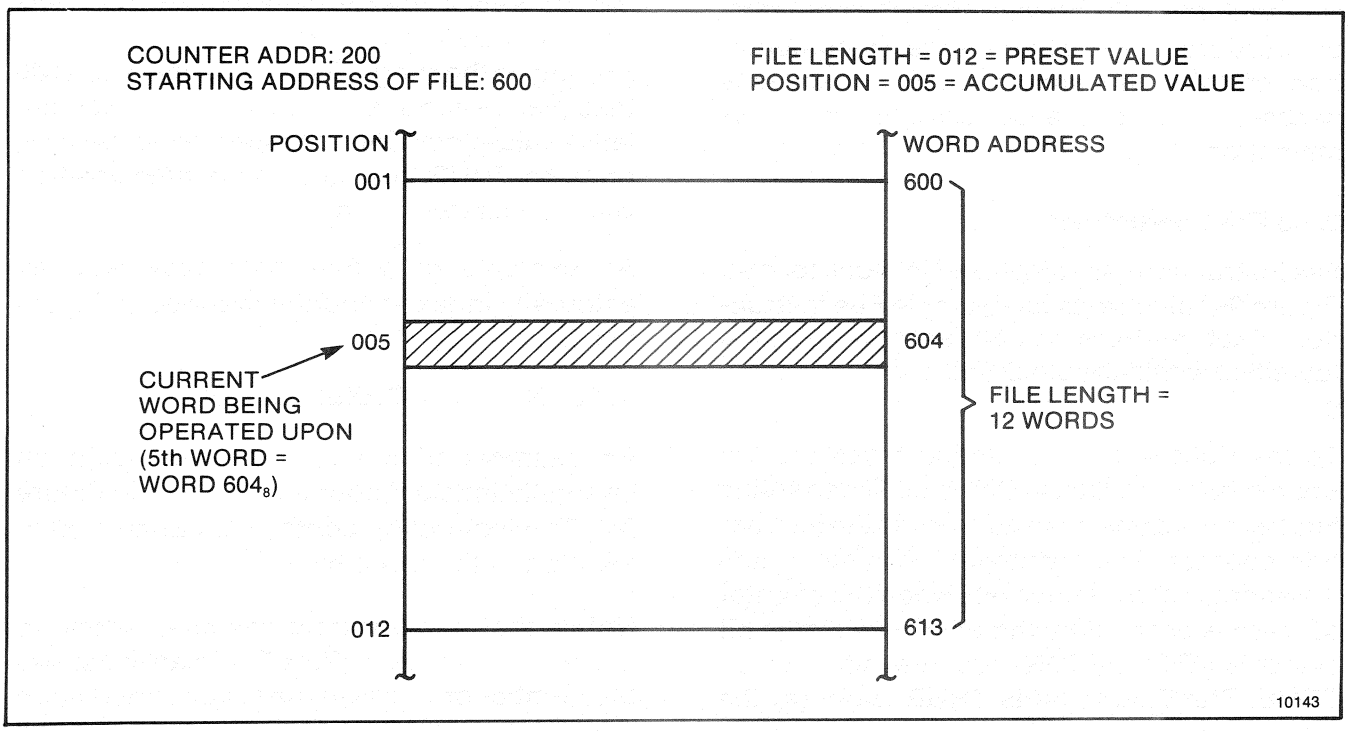


FIGURE 9-1 — File Structure

005. It points to the fifth word in the file, word address 604_8. This word will be either the source or destination of the data that is currently being operated upon by the File instruction.

9.1.2 File Planning

Although files can be located anywhere within the Data Table, they usually should be located after the last timer/counter Preset area. Timer/Counter Accumulated values and Preset values are 100_8 apart. Whereas, files are made up of consecutive words. Therefore, a file which begins in a timer/counter Accumulated area could continue into the Preset area and write over the Preset values found there. Also, a file should not be inadvertently programmed to overlap or be totally contained within another. Care should be taken in assigning file areas to avoid unintentionally altering the contents of one file by the operation of another.

To avoid programming errors of this kind, Data Table Assignment Sheets found at the end of Section 12 should be used. These are bit/word tally sheets on which all Data Table words are logged as they are assigned and/or reserved for I/O devices, timers, counters, bit/word storage and file storage. Documentation of this kind should be included as a necessary part of the Control System Documentation.

9.1.3 File Instructions

File instructions are displayed in block format. Figure 9-2 illustrates an example File instruction. Each address or data entry required is identified within the block and defined beneath it.

To the right of the instruction block are the Enable (EN) and Done (DN) bits. These status bits have the same word address as the instruction counter. The Industrial Terminal enters the address of the status bits when the counter address is entered by the user. The status bit symbols (EN) and (DN) are intensified when TRUE. The Enable bit is TRUE when (a) the rung is TRUE and/or (b) when the instruction is operating. The Done bit is TRUE when the instruction operation is complete (counter Accumulative value equals Preset value).

There are two kinds of File instructions. Each instruction can only be one of the two kinds. The difference between the two is the manner in which the counter Accumulated value is indexed. The two kinds are:

- Externally Indexed - indexed by other instructions in the User Program. The instruction operates on a word.
- Internally Indexed - incremented by the instruction, itself. The instruction operates on a file.

EXTERNALLY INDEXED COUNTER

When the counter is externally indexed, the Accumulated value must be positioned to point to any word in the file by instructions in the User Program. The counter can be indexed randomly by using GET/PUT transfer or sequentially by using another counter. In either case, the PUT or counter instruction is addressed to the Accumulated value of the File instruction counter.

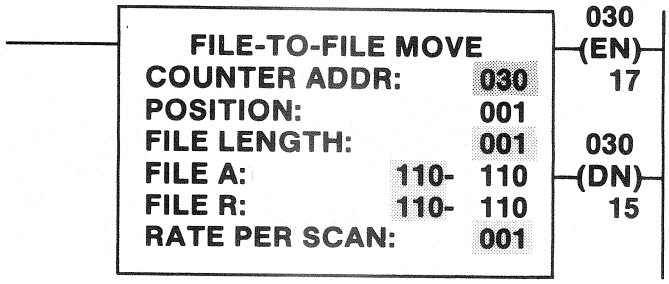
Instructions with externally indexed counters will operate when the rung condition is TRUE. As long as the rung is TRUE the operation will take place each scan. If the counter Accumulative value (position) changes while the rung remains TRUE, the data at the new position will be operated upon.

An example of a File instruction with an externally indexed counter is shown in Figure 9-3.

INTERNALLY INDEXED COUNTER

An example of a File instruction with an internally indexed counter is shown in Figure 9-4. The instruction contains an Enable bit in addition to the Done bit.

Notice that another term has been added to the instruction block: Rate Per Scan. It defines the number of words in the file operated upon during one scan. Its value is user-chosen and determines how the file operation is to take



NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACED BY USER—ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INITIALLY DISPLAYED, 3, OR 4 WILL DEPEND ON THE SIZE OF THE DATA TABLE.

COUNTER ADDRESS : ADDRESS OF THE INSTRUCTION IN THE ACCUMULATED VALUE AREA OF DATA TABLE.

POSITION : CURRENT WORD BEING OPERATED UPON. (ACCUMULATED VALUE OF COUNTER).

FILE LENGTH : NUMBER OF WORDS IN FILE (PRESET VALUE OF THE COUNTER).

FILE A : STARTING ADDRESS OF SOURCE FILE.

FILE R : STARTING ADDRESS OF DESTINATION FILE.

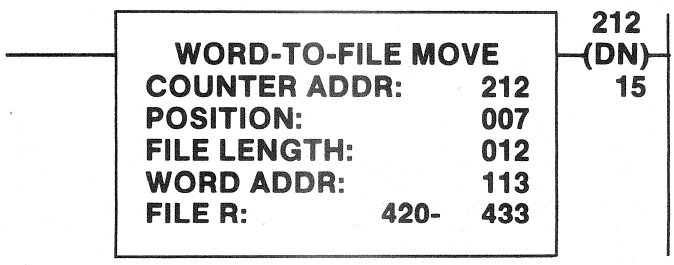
RATE PER SCAN : NUMBER OF DATA WORDS MOVED PER SCAN.

ENABLE BIT -(EN)- : AUTOMATICALLY ENTERED FROM THE COUNTER ADDRESS. SET HIGH WHEN THE RUNG CONDITION IS TRUE AND/OR WHEN THE INSTRUCTION IS OPERATING.

DONE BIT -(DN)- : AUTOMATICALLY ENTERED FROM THE COUNTER ADDRESS. SET HIGH WHEN THE OPERATION IS COMPLETE AND REMAINS HIGH AS LONG AS THE RUNG CONDITION IS TRUE.

10144

FIGURE 9-2 — File Instruction Format



10145

FIGURE 9-3 — Example of an Externally Indexed File Instruction

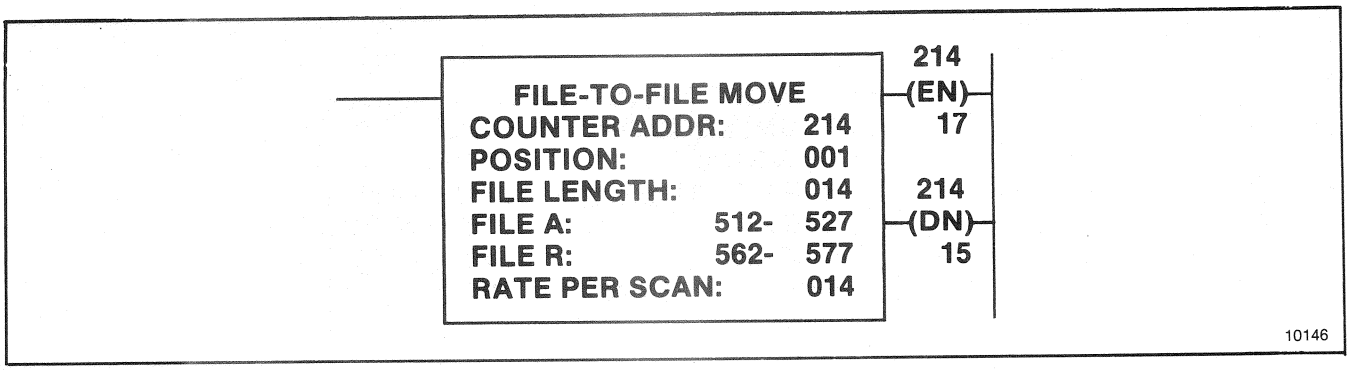


FIGURE 9-4 — Example of an Internally Indexed File Instruction

place. There are three (3) modes of operation based on Rate Per Scan. They are:

- COMPLETE
- DISTRIBUTED COMPLETE
- INCREMENTAL

The relationship between the Rate Per Scan and the modes of operation are summarized in Table 9-1.

Complete Mode

In the Complete mode, the Rate Per Scan is equal to the File Length, and the entire file is operated upon in one scan. On a FALSE-to-TRUE transition of the rung condition, the instruction is enabled and the Accumulated value of the file counter is internally indexed from the first to the last word of the file. As the Accumulated value points to each word, the operation defined by the File instruction is performed. The operation of a File instruction in the COMPLETE mode is shown in Figure 9-5.

The operation of the status bits for the Complete mode is shown in Figure 9-6. The instruction is enabled by the FALSE-to-TRUE transition of the rung condition. The rung would have to go FALSE and then TRUE after the operation is complete in order to repeat the instruction. If the rung remained TRUE on subsequent scans, the instruction would not repeat. After the instruction has operated on the last word, the Done bit is set. The Enable and Done bits are reset to zero and the counter is reset to position 001 when the rung condition goes FALSE. If the rung was enabled for only one scan, the Done bit would come on during that scan and remain set for one additional scan.

Distributed Complete Mode

In cases where it is not necessary that the file operation be completed in one program scan, it may be advantageous to distribute the file operation over several program scans. This is to avoid overextending the scan time of any one program scan. This can be done by selecting the Distributed Complete Mode. Any Rate Per Scan, R, can be chosen where $0 < R < \text{File Length}$.

When the rung containing the File instruction goes TRUE, the number of words equal to the Rate Per Scan is operated upon during one

Table 9-1 — Modes of Instruction Operation

MODE OF OPERATION	R = RATE PER SCAN	NUMBER OF WORDS OPERATED UPON
COMPLETE	R=File Length	Entire file per scan
DISTRIBUTED COMPLETE	0<R<File Length	R words per scan
INCREMENTAL	R = 0	One word per rung transition

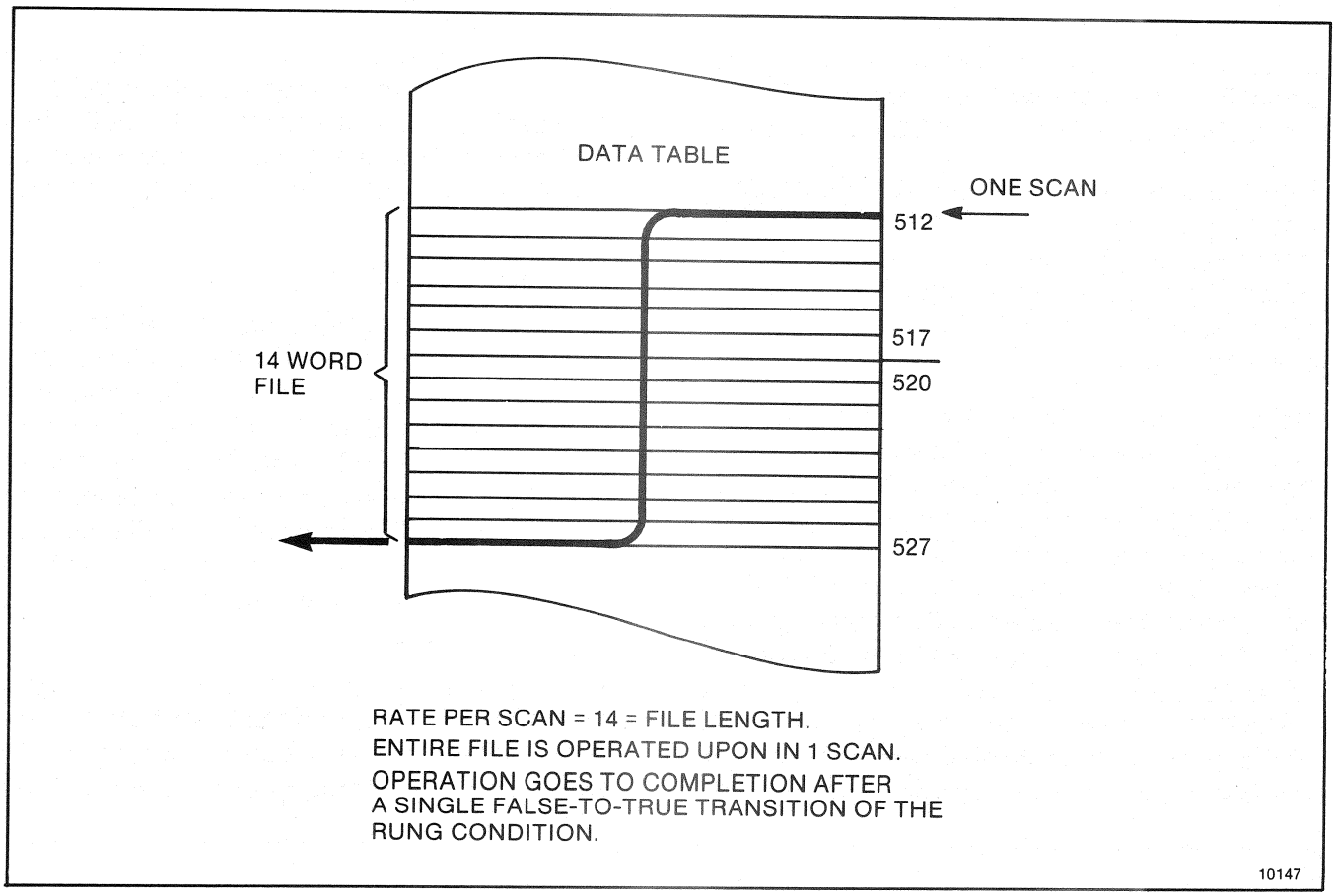


FIGURE 9-5 — Complete Mode Operation

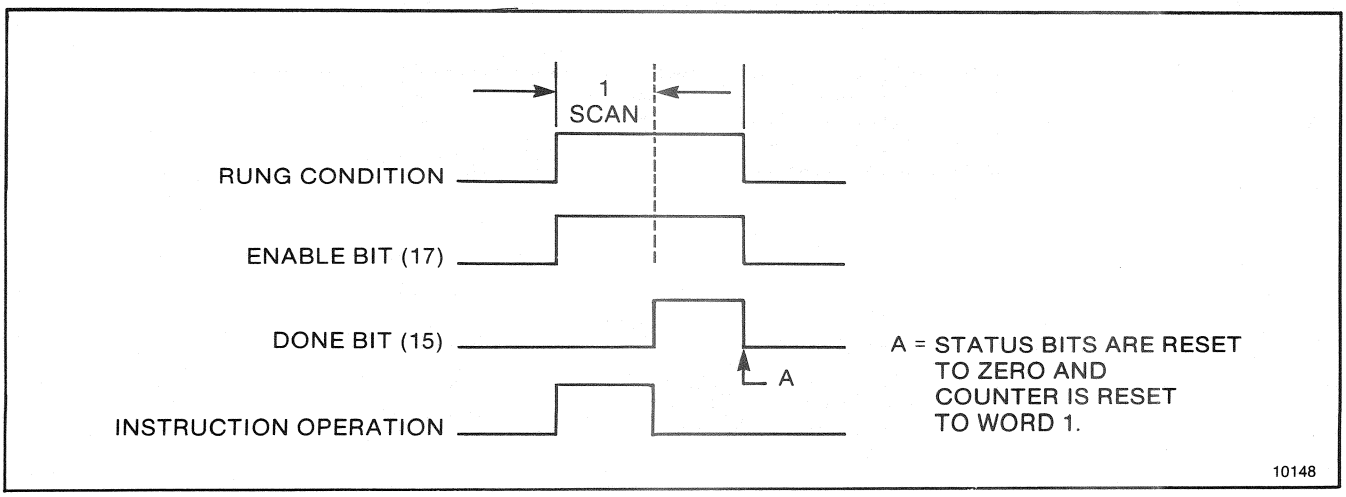


FIGURE 9-6 — Status Bits for Complete Mode

scan. This process is repeated over a number of scans, until the entire file has been operated upon. Figure 9-7 shows the operation of a File instruction in the Distributed Complete mode.

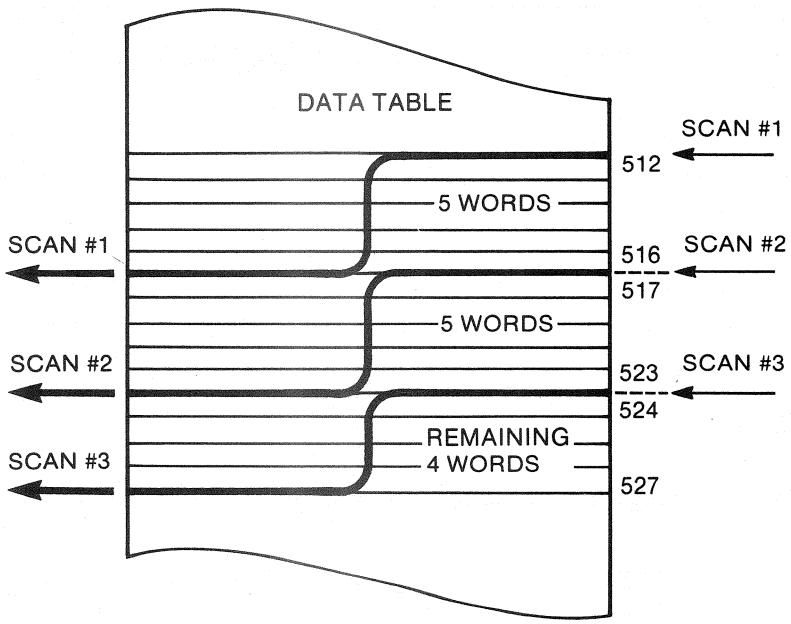
The File Instruction, once enabled, remains enabled for the number of scans necessary to complete the operation. The rung could become repeatedly FALSE and TRUE during this time without interrupting the operation of the instruction.

Note: It is important that the User Program NOT make use of the results of a File instruction operating in the Distributed Complete mode until the Done bit is set. At completion, the rung containing the instruction could be either TRUE or FALSE. If the rung is TRUE at completion, the Enable and Done bits are both ON. They are reset to zero and the counter is reset to position 001 when the rung goes FALSE. However, if the rung is FALSE at completion, the Enable bit is reset to zero after the last group of words is operated upon. At the same time, the Done bit comes ON and stays ON for one scan. The Done bit is reset to zero and the counter is reset to position 001 in the next scan. The operation of the status bits is shown in Figure 9-8 for the two cases: a) TRUE at Completion, and b) FALSE at Completion.

Incremental Mode

The Incremental Mode allows the file to be operated upon one word per rung transition. Each time the rung containing the instruction goes from FALSE to TRUE, the instruction operates on the word pointed to by the counter Accumulated value, and then increments to the next word. The operation of a File instruction in the Incremental mode is shown in Figure 9-9. In this mode, the Rate Per Scan is set equal to zero.

Note: The differences between the Incremental mode (R=0) and the Distributed Complete mode when R=1 can be compared. In both cases the operation defined by the File



RATE PER SCAN = 005
FILE IS OPERATED UPON OVER 3 SCANS.
OPERATION GOES TO COMPLETION AFTER A SINGLE FALSE-TO-TRUE TRANSITON OF THE RUNG CONDITION.

10149

FIGURE 9-7 — Distributed Complete Mode Operation

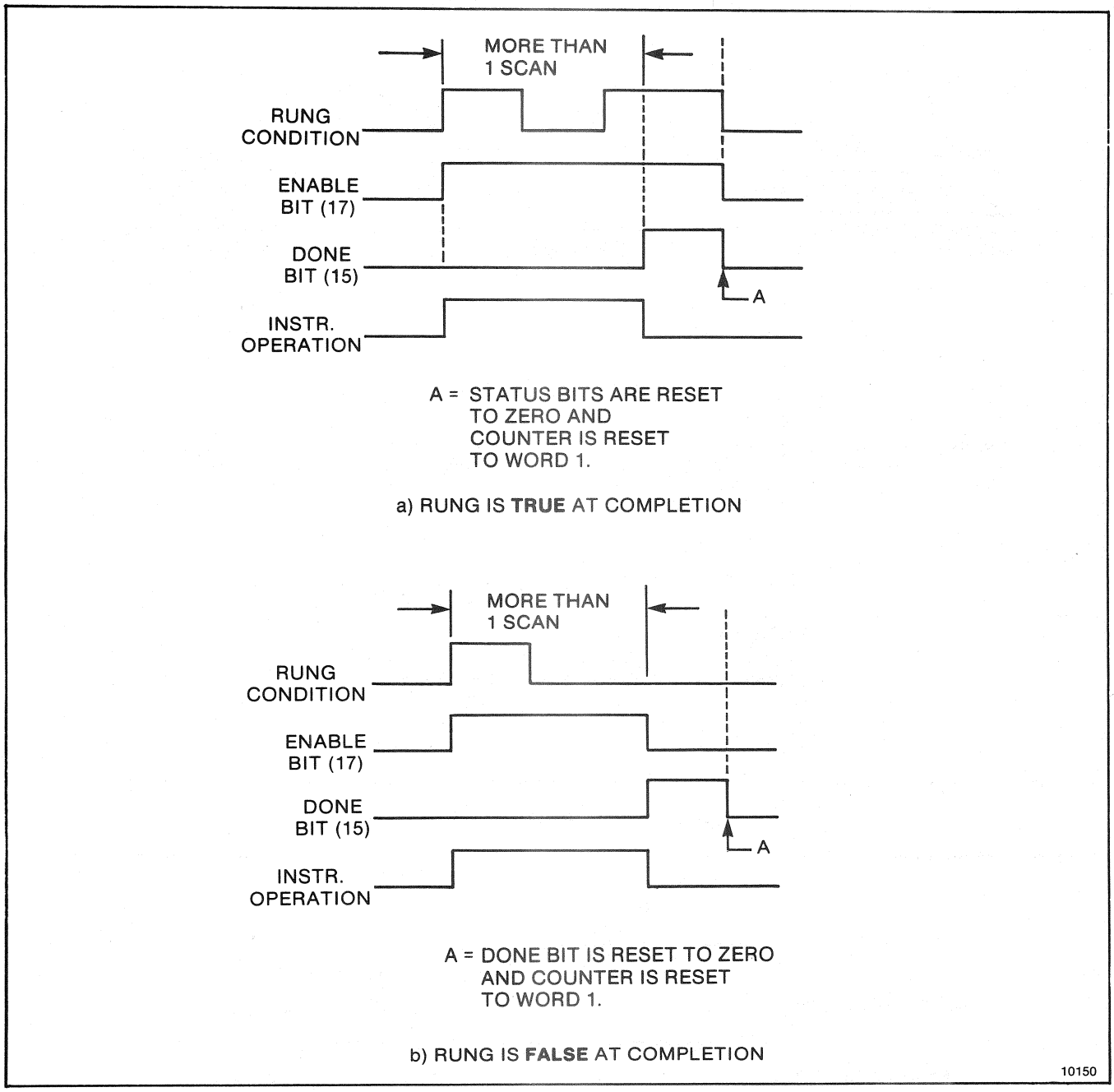


FIGURE 9-8 — Status Bits for Distributed Complete Mode

instruction is performed as the file counter sequentially points to each word in the file. The rung must go from FALSE to TRUE in order to initiate the operation. In the Distributed Complete mode when R=1, the entire file operation requires a single FALSE-TRUE transition. It goes to completion automatically, one word per scan over the number of scans equal to the file length. In the Incremental mode, (R=0), the operation must be enabled by a separate FALSE-TRUE transition for each word in the file.

The operation of the status bits in the Incremental mode is illustrated in Figure 9-10. The Enable bit is ON if the rung is TRUE. After the last word in the file has been operated upon, the Done bit comes ON. When the rung goes FALSE after the last word has been operated upon, the Enable and Done bits are reset to

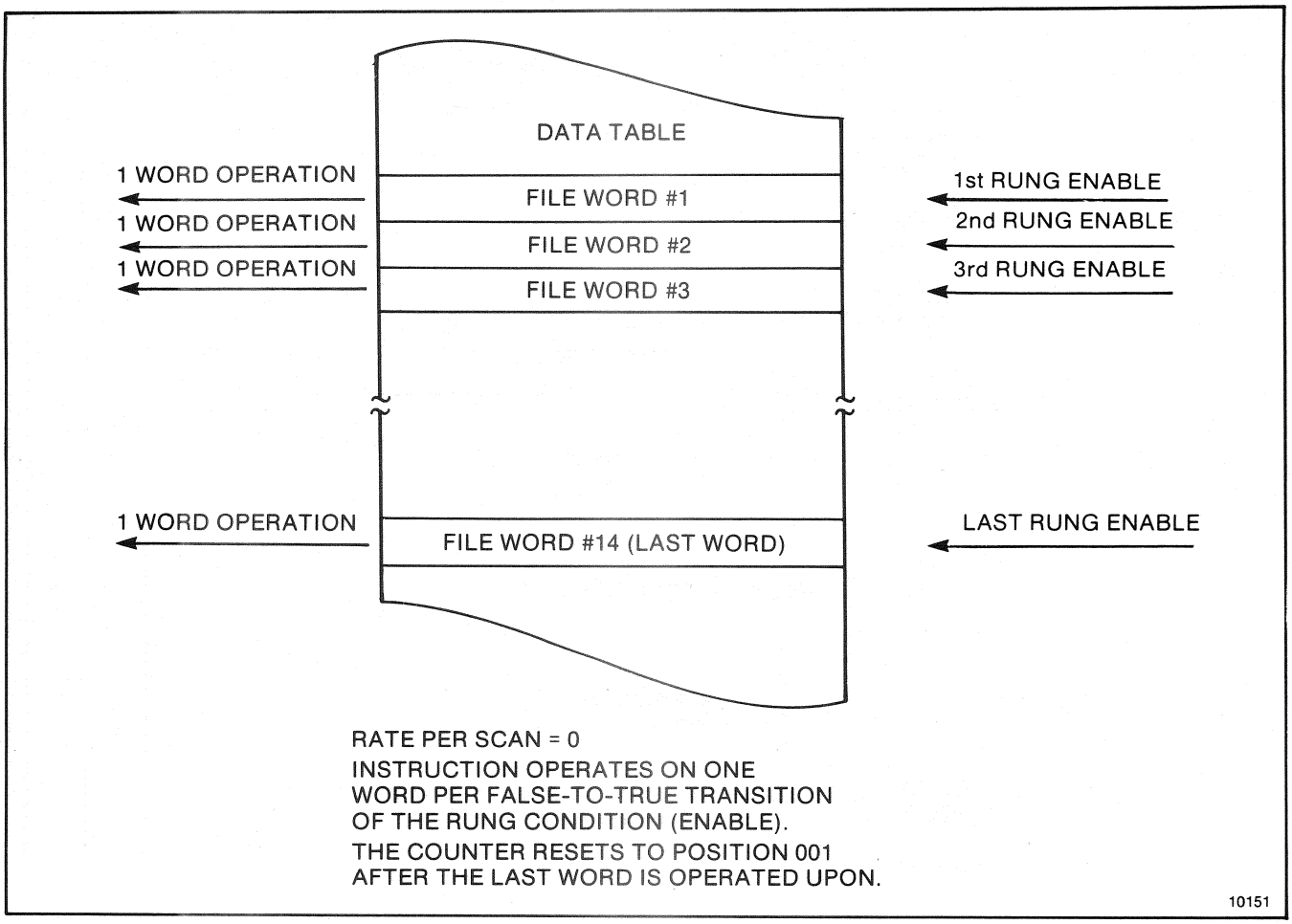


FIGURE 9-9 — Incremental Mode Operation

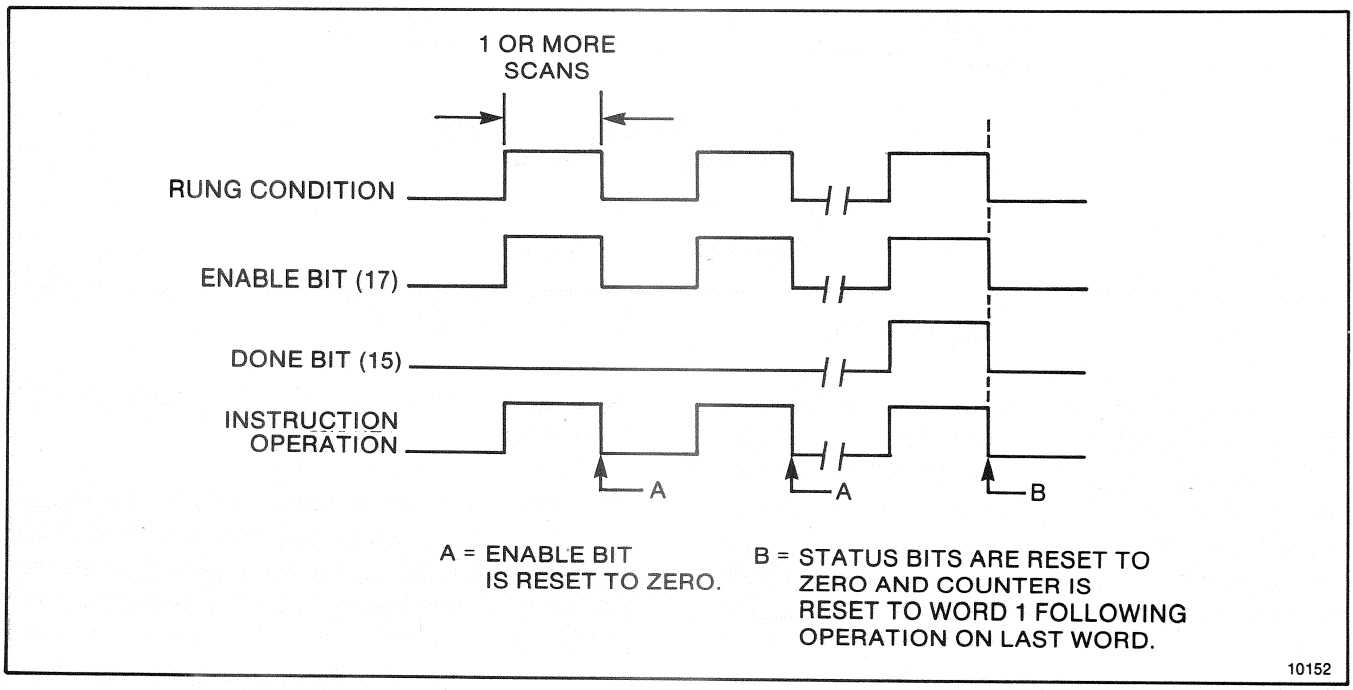


FIGURE 9-10 — Status Bits for Incremental Mode

zero and the counter is reset to position 001. If the rung remains TRUE for more than one scan, the operation does NOT repeat. The operation only occurs in the scan in which the FALSE-TRUE transition occurs.

9.1.4 Programming File Instructions

A File instruction can be entered into the User Program by pressing the key representing the general type of block instruction i.e. [FILE] or [SEQ] followed by a numeric code (such as 0, 1, 2 or 10, 11, 12 etc.) specific to the instruction. A list of numeric codes for each general type of block instruction can be displayed by pressing the general instruction key followed by the [HELP] key. See Section 13.3, Help Directories, for additional information.

After the key sequence is pressed, the instruction block will be displayed. The title will be intensified and blinking until all required information is entered. Default values are presented in the instruction block. A character cursor will indicate where instruction parameters are to be entered.

The programming and operation of the block instructions are covered in detail in the section specifically assigned to each instruction.

9.1.5 File Instruction Run-Time Errors

If the counter Accumulated value exceeds its Preset value, the counter will be indexed outside the file. This will cause a Run-time Error which halts system operation. For instructions with an externally indexed counter, the User Program should insure that the instructions which change the counter Accumulated value do NOT cause the Preset value to be exceeded.

For instructions with an internally incremented counter, the counter address should be used only by that instruction. Preset and Accuulated values should NOT be manipulated inadvertently by User Program, Data Highway communication or by means of programming equipment. Unpredictable machine motion with possible damage to equipment and/or injury to personnel could occur. A Run-time Error will provide protection only when the instruction counter is indexed to outside the file, not when the instruction counter is inadvertently manipulated to an address within the file. Additional information on Run-time Errors can be found in Section 2.3.

9.2 DATA MONITOR MODE

The Data Monitor mode can be used to monitor, load and edit file data. Each File instruction has two corresponding Data Monitor displays. One displays file data in Binary and the other displays file data in Hexadecimal representation. The Binary Data Monitor display shows each word of a file in a 16-character format of 1's and 0's. The Hexadecimal display shows a 4-character format of Hexadecimal digits (0-9 and A-F). Generally, the Binary display is chosen when bit information is pertinent and Hexadecimal display is chosen when word values are desired. Data can be entered and/or displayed in either number system. The Industrial Terminal can automatically convert from one number system to the other when the alternate display is selected. For additional information on number systems, refer to Section 17.

9.2.1 Accessing the Data Monitor Mode

To access the Data Monitor mode, place the cursor on the desired File Instruction in the User Program. All of the user-added information required by the instruction block must be entered before the Data Monitor display mode can be accessed. Then press one of the following key sequences:

- [DISPLAY][0] - for Binary Data Monitor Format
- [DISPLAY][1] - for Hexadecimal Data Monitor Format

These commands are summarized in Table 9-2.

Once pressed, the Industrial Terminal display will change from Ladder Diagram to Data Monitor mode. An example File instruction and its corresponding Data Monitor display in Hexadecimal format are shown in Figure 9-11.

9.2.2 Data Monitor Display

Data Monitor displays, although unique for each File instruction, have commmon charac-

teristics including a Header, a File section and a Command Buffer.

HEADER

The Header is located at the top of the screen and contains information pertinent to its corresponding File instruction such as: Counter Address, File Addresses, Word Addresses, Position and File Length. This information is extracted from the instruction block by the Industrial Terminal.

Some Data Monitor Headers contain word data which can be entered or changed in the same manner as file data. The entry or change of data is described in paragraph 9.2.3 and 9.2.5.

TABLE 9-2 — Accessing the Data Monitor Mode

KEY SEQUENCE	EXPLANATION
[DISPLAY][0]	Binary Data Monitor format
[DISPLAY][1]	Hexadecimal Data Monitor format
Note: The cursor must be positioned on the desired File instruction in the Ladder Diagram.	

FILE SECTION

The File section is located in the center of the screen and displays the data stored in the file. The locations within the file are numbered

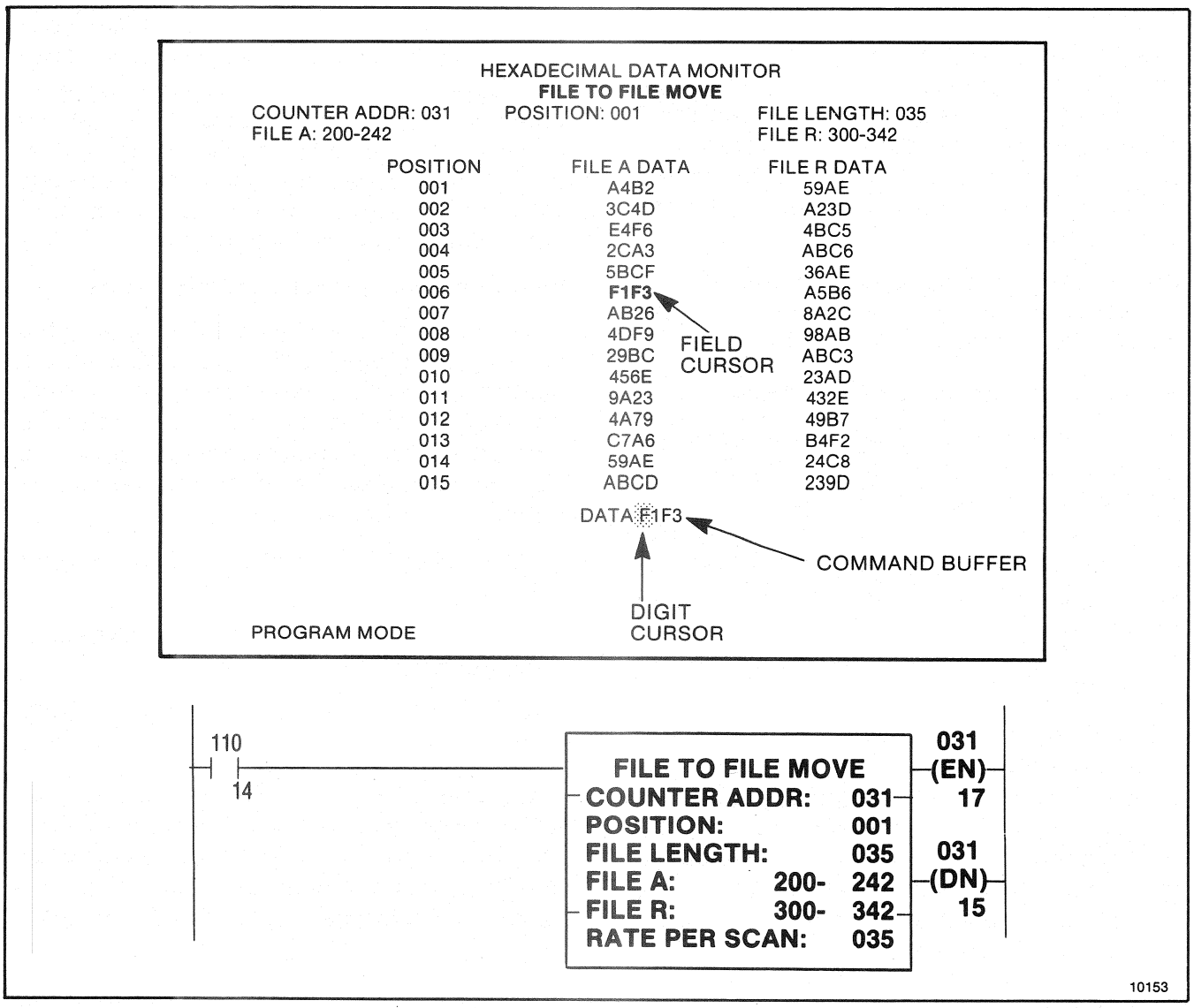


FIGURE 9-11 — Example Hexadecimal Data Monitor Display of a File Instruction

sequentially from the starting word address. For example, Position 001 corresponds to word 1, the starting address of a file. When file data is displayed in Binary representation, data bits, 00-17, are assumed to be numbered from right to left.

Each column in the display represents one file. In Figure 9-11, two files are shown. Each column can contain from 10 to 15 words of data. Files with more words can be displayed by scrolling and/or paging the display. Procedures for scrolling and paging the display will be discussed in paragraph 9.2.4.

The File section contains a Field Cursor which can be positioned on any word in the file. The word pointed to by the Field Cursor is intensified.

COMMAND BUFFER

The Command Buffer is located at the bottom center of the screen and is used to enter or change file data. If the Header contains word data, this information can also be changed using the Command Buffer.

The current word in the Command Buffer is the word that is pointed to and intensified by the Field Cursor. The word data is duplicated in the Command Buffer. A Digit Cursor is active with the Command Buffer and is used to enter or change data.

The Command Buffer is always displayed when the Processor is in PROGRAM mode. When in RUN/PROGRAM mode, the Command Buffer will not be displayed unless the On-Line Data Change feature is being used. On-Line Data Change is discussed in Section 13.5.8.

9.2.3 Cursor Controls

The Field Cursor and Digit Cursor are used together to enter or change file data.

FIELD CURSOR

The Field Cursor initially appears in the top left position of the File section. Within a column, the [▼] and [▲] keys can be used to move the cursor down or up respectively one position number at a time.

The Field Cursor can be moved from one column to another. The [SHIFT][►] and [SHIFT][◄] key sequences are used to move one column right and left, respectively. If the Field Cursor is on the far left or right edge of the screen, it cannot be moved past the edge boundary.

The Field Cursor can be moved into the Header area of a Data Monitor Display which contains user-changeable data by pressing [DISPLAY][0][0][0]. When there, the Field Cursor is controlled by the four cursor control keys [SHIFT][◄], [SHIFT][►], [▼] and [▲]. If the Header contains no changeable data, the Field Cursor cannot be moved there. The Field Cursor can be moved back to the File section by pressing [DISPLAY][X][X][X] where XXX is any position number.

The Field Cursor commands are summarized in Table 9-3.

TABLE 9-3 — Field Cursor Commands and Scrolling

KEY SEQUENCE	EXPLANATION
[▲] or [▼]	Moves the Field Cursor up or down, respectively, one line at a time. If it is at the top or bottom of the File section, the display will scroll one line but the cursor will not move.
[SHIFT][►]	Moves the Field Cursor to the next file column to the right.
[SHIFT][◄]	Moves the Field Cursor to the next file column to the left.
[DISPLAY][0][0][0]	Moves the Field Cursor into the Header, if accessible.
[DISPLAY][X][X][X]	Moves the Field Cursor and file word XXX to the top row of the File section.
[▲], [▼],[SHIFT][◄] or [SHIFT][►]	Positions the Field Cursor in the Header: up, down, left or right, respectively. (If the Header is accessible)

DIGIT CURSOR

The Digit Cursor initially appears in the left-most position in the Command Buffer. It can be moved to the right or left within the Command Buffer by pressing the [►] or [◄] cursor command keys, respectively. It will not respond to a command to move outside the buffer area. Whenever the Command Buffer is displayed, the Digit Cursor will always be displayed in reverse video.

The Digit Cursor commands are summarized in Table 9-4.

TABLE 9-4 — Digit Cursor Commands

KEY SEQUENCE	EXPLANATION
[►] or [◄]	Moves the Character Cursor one position to the right or left, respectively in the Command Buffer.

9.2.4 Data Monitoring Procedures

File Data can be monitored when the Processor is in any mode of operation using the procedures described below.

PAGING

A "page" of information is defined as a full screen of file words in the File section. For example in Figure 9-11, a page is shown which begins at position 001 and ends at position 015. To display longer files, additional full pages can be presented by pressing [SHIFT][▼]. In Figure 9-11, pressing [SHIFT][▼] would change the display to a page beginning at position 016 and ending with position 030. Pressing [SHIFT][▲] would return the display to its previous page (i.e. positions 001-015).

SPECIFIED PAGING

When a word in a particular file position, XXX, is of interest, specified paging will present the page containing that word. The word will be presented on the top row of the File section. The Field Cursor will move to word XXX. Pressing [DISPLAY][X][X][X] will select a page beginning with XXX on the top row.

Paging and Specified Paging commands are summarized in Table 9-5.

TABLE 9-5 — Paging and Specified Paging Commands

KEY SEQUENCE	EXPLANATION
[SHIFT][▼]	Displays the next full page of data.
[SHIFT][▲]	Displays the previous full page of data.
[DISPLAY][X][X][X]	Specified paging presents a page beginning at the desired file word XXX. The Field Cursor moves to word XXX.

SCROLLING

Scrolling the File section, one entry at a time, can be done using the [▲] and [▼] keys when the Field Cursor is on the top or bottom position, respectively. In Figure 9-11, if the [▼] key were pressed once with the Field Cursor on position 015, the display would change to a page ending at position 016 (beginning at 002). The Field Cursor would remain at the bottom of the screen. Pressing the [▼] key four more times would place position 020 on the bottom line, etc. Attempting to scroll in either direction beyond the number of words contained in the file will not be allowed. An INVALID KEY message will appear.

Scrolling procedures are summarized in Table 9-3.

9.2.5 Entering & Changing Data

Data can always be entered or changed when the Processor is in PROGRAM mode. In RUN/PROGRAM mode, data can only be changed using On-Line Data Change. The key sequence [SEARCH][5][1] initializes this feature. See section 13.5.8 for additional information on the On-Line Data Change function.

Data is entered or changed in the Command Buffer. After a character is entered, the Digit Cursor shifts to the right one position and waits for the next entry. The cursor will remain even after the last character is entered. When

the Command Buffer values have been entered or corrected using the numeric, hexadecimal and cursor control keys, press the [INSERT] key. The data in the Command Buffer is then entered into Processor memory in the corresponding file location.

The procedure for entering or changing data is summarized in Table 9-6.

TABLE 9-6 — Data Entry Commands

KEY SEQUENCE	EXPLANATION
[D][D][D][D]1	Data is entered or changed in the Command Buffer.
[INSERT]	Command Buffer data is loaded into Processor memory and placed into the file word located by the Field Cursor.
[CANCEL COMMAND]	Terminates Data Monitor Mode and returns display to Ladder Diagram. If in On-Line Data Change, [CANCEL COMMAND] will terminate On-Line Data Change. A second [CANCEL COMMAND] will terminate Data Monitor mode.

1 Hexadecimal format. The Binary format (1's and 0's) would contain 16 data characters.

Section 10
DATA TRANSFER FILE INSTRUCTIONS

10.0 GENERAL

The Data Transfer File instructions are output instructions and include:

FILE-TO-FILE MOVE

WORD-TO-FILE MOVE

FILE-TO-WORD MOVE

Note: It is assumed that the concepts, terminology and consequences introduced in Section 9, Files and Data Monitor Operation, have been thoroughly understood.

10.1 FILE-TO-FILE MOVE INSTRUCTION

The FILE-TO-FILE MOVE instruction transfers duplicates of data stored in the words of one file to another file location in the Data Table. All 16 bits of each word are transferred.

10.1.1 Instruction Operation

The operation of the FILE-TO-FILE MOVE instruction is illustrated in Figure 10-1. In this example, the data of a 10-word source file, File A beginning at address 510_8, is copied into a 10-word result file, File R beginning at address 574_8. The data in File A remains unaffected by the instruction operation. The counter is internally incremented by the instruction.

The modes of operation of the instruction which are set by the Rate Per Scan, and the operation of the status bits are described in Section 9.1.3 File Instructions (Internally Indexed Counter).

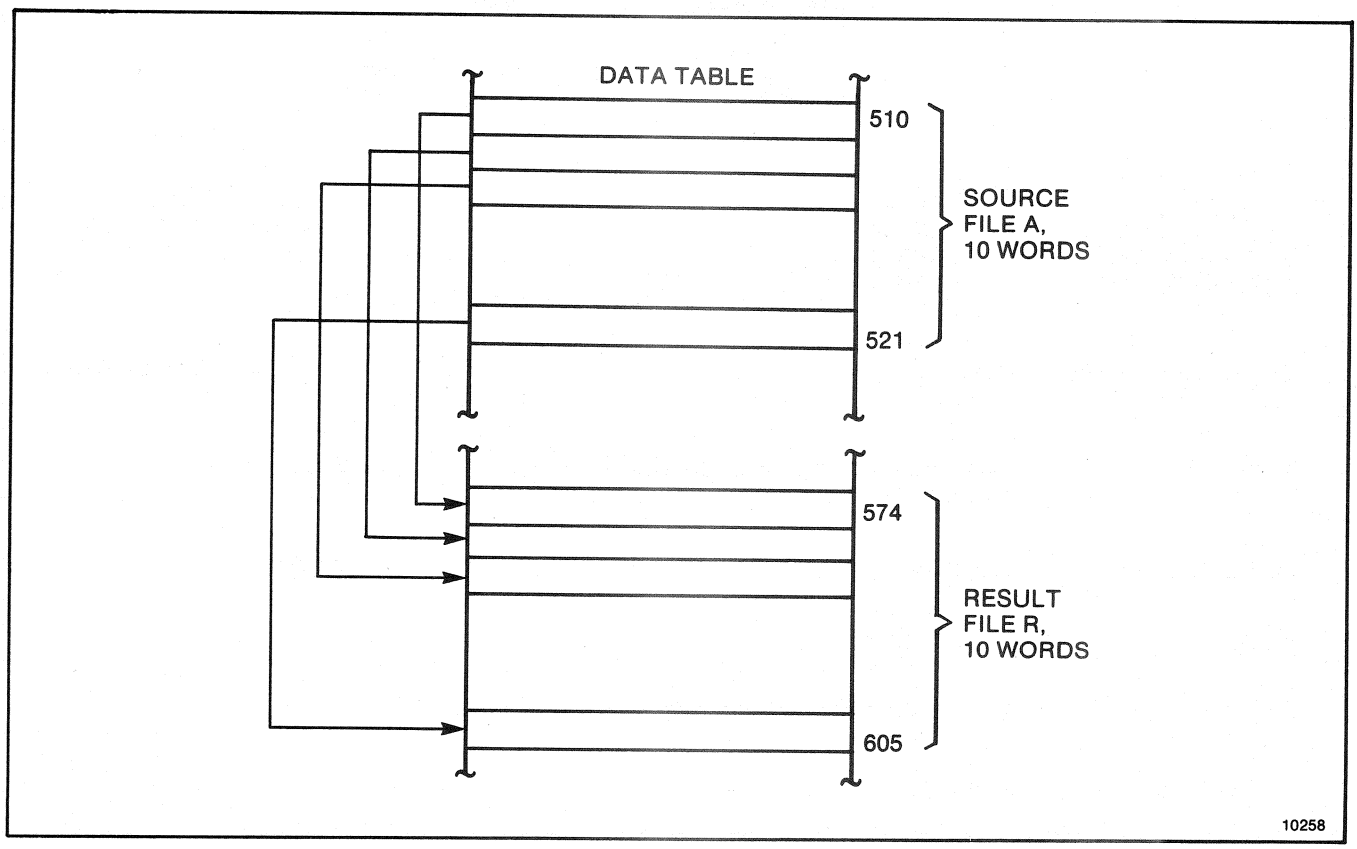


FIGURE 10-1 — FILE-TO-FILE MOVE Operation

10.1.2 FILE-TO-FILE MOVE Instruction Characteristics

- Output instruction
- Can operate in Complete, Distributed Complete or Incremental mode.
- Counter is internally incremented by the instruction.
- Requires 5 words of User Program.
- Key sequence [FILE][1][0]

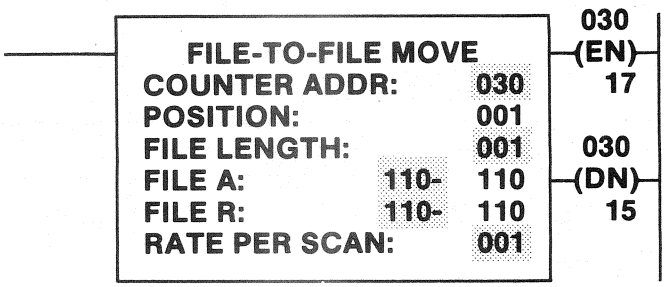
10.1.3 Programming the FILE-TO-FILE MOVE Instruction

To program a FILE-TO-FILE MOVE instruction, press the key sequence [FILE] [1], [0]. Figure 10-2 is representative of the display that will appear.

WARNING: The counter address for the FILE-TO-FILE MOVE instruction should be reserved for that instruction. DO NOT manipulate the counter Accumulated or Preset word. Changes to these values could result in unpredictable machine operation or a Run-Time Error. Damage to equipment and/or injury to personnel could occur.

An example rung containing a FILE-TO-FILE MOVE instruction is shown in Figure 10-3. The following parameters have been entered into the instruction:

Counter Address	214
File Length	010
File A starting address	510
File R starting address	574
Rate Per Scan	010



NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACE BY USER—ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INTIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

COUNTER ADDRESS : ADDRESS OF THE INSTRUCTION IN THE ACCUMULATED VALUE AREA OF DATA TABLE.

POSITION : CURRENT WORD BEING OPERATED UPON. (ACCUMULATED VALUE OF COUNTER).

FILE LENGTH : NUMBER OF WORDS IN FILE (PRESET VALUE OF THE COUNTER).

FILE A : STARTING ADDRESS OF SOURCE FILE.

FILE R : STARTING ADDRESS OF DESTINATION FILE.

RATE PER SCAN : NUMBER OF DATA WORDS MOVED PER SCAN.

10155

FIGURE 10-2 — FILE-TO-FILE MOVE Format

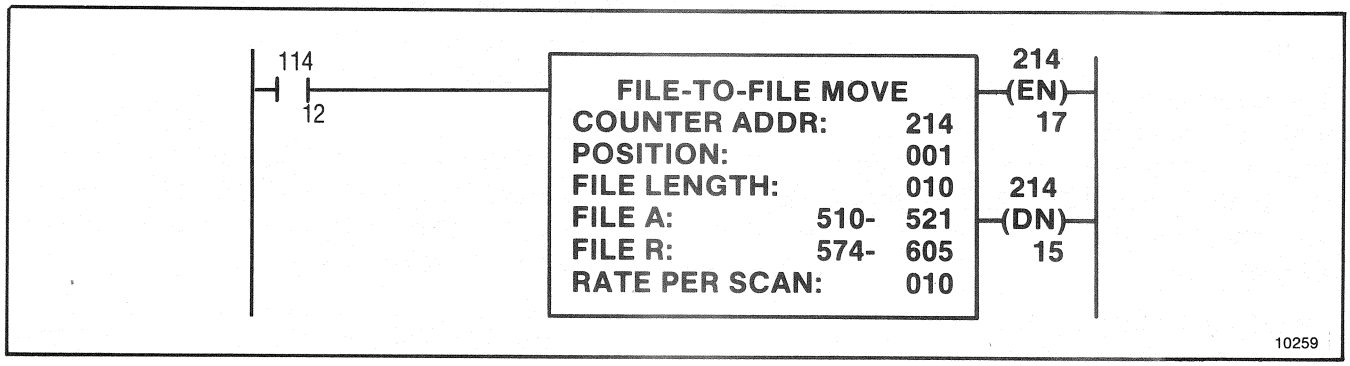


FIGURE 10-3 — FILE-TO-FILE MOVE Example Rung

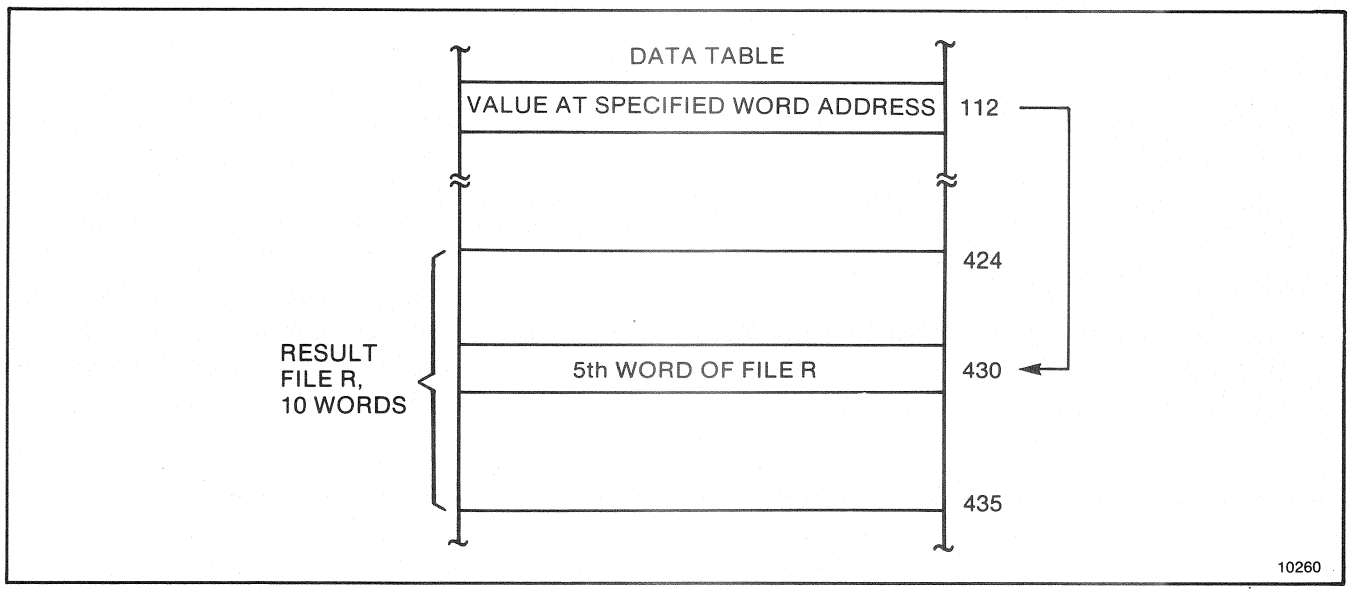


FIGURE 10-4 — WORD-TO-FILE MOVE Operation

When switch 114/12 closes, the data of all 10 words in File A is duplicated into the words of File R. The example instruction is operating in the Complete mode because the Rate Per Scan is equal to the File Length.

The procedure for using the Data Monitor mode to monitor, load or edit file data is presented in Section 9.2. Data Monitor Mode.

10.2 WORD-TO-FILE MOVE AND FILE-TO-WORD MOVE INSTRUCTIONS

The WORD-TO-FILE MOVE instruction duplicates of the data of a specified Data Table word outside the file and transfers it to a word within a file.

The FILE-TO-WORD MOVE instruction duplicates the data of a word within a file and transfers it to a to a specified Data Table word outside the file.

10.2.1 Instruction Operation

The operation of the two instructions is fundamentally the same except that the source and destination of data are reversed. With a WORD-TO-FILE MOVE instruction, the data of a specified word in the Data Table, usually an I/O Image Table word, can be copied into any Position (word) within a file pointed to by the Accumulated value of the instruction counter. The counter is externally indexed by the User Program logic. The operation of a WORD-TO-FILE MOVE instruction is illustrated in Figure 10-4. In this example, the data of the specified word address 112_8 in the I/O

Image Table is copied into the 5th word at address 430_8 in the result file, File R.

The operation of the FILE-TO-WORD MOVE instruction is shown in Figure 10-5. The source of data is the 5th word at address 430_8 in the source file, File A. The data is copied into the specified I/O Image Table word address 112_8.

Details of the instruction and status bit operation are described in Section 9.1.3, File Instructions (Externally Indexed Counter).

10.2.2 WORD-TO-FILE and FILE-TO-WORD MOVE Instruction Characteristics

- Output instruction
- Counter must be externally indexed by User Program
- Data is transferred every scan that the rung is TRUE.
- Requires 4 words of User Program
- Key sequence [FILE][1][1] for WORD-TO-FILE MOVE
  [FILE][1][2] for FILE-TO-WORD MOVE

10.2.3 Programming the WORD-TO-FILE MOVE and FILE-TO-WORD MOVE Instructions

To program a WORD-TO-FILE MOVE or a FILE-TO-WORD MOVE instruction, press key sequence [FILE][1][1] or [FILE][1][2], respectively. The display shown in Figure 10-6 or Figure 10-7, respectively, will appear.

WARNING: The counter address for the WORD-TO-FILE MOVE and FILE-TO-WORD MOVE instructions should be used only for the intended instruction and the corresponding instructions which manipulate the Accumulated value. Do NOT inadvertently manipulate the Preset or Accumulated word. Changes to these values could result in unpredictable machine operation or a Run-Time Error. Damage to equipment and/or injury to personnel could occur.

An example rung containing a WORD-TO-FILE MOVE instruction is shown in Figure 10-8. The following parameters have been entered into the instruction:

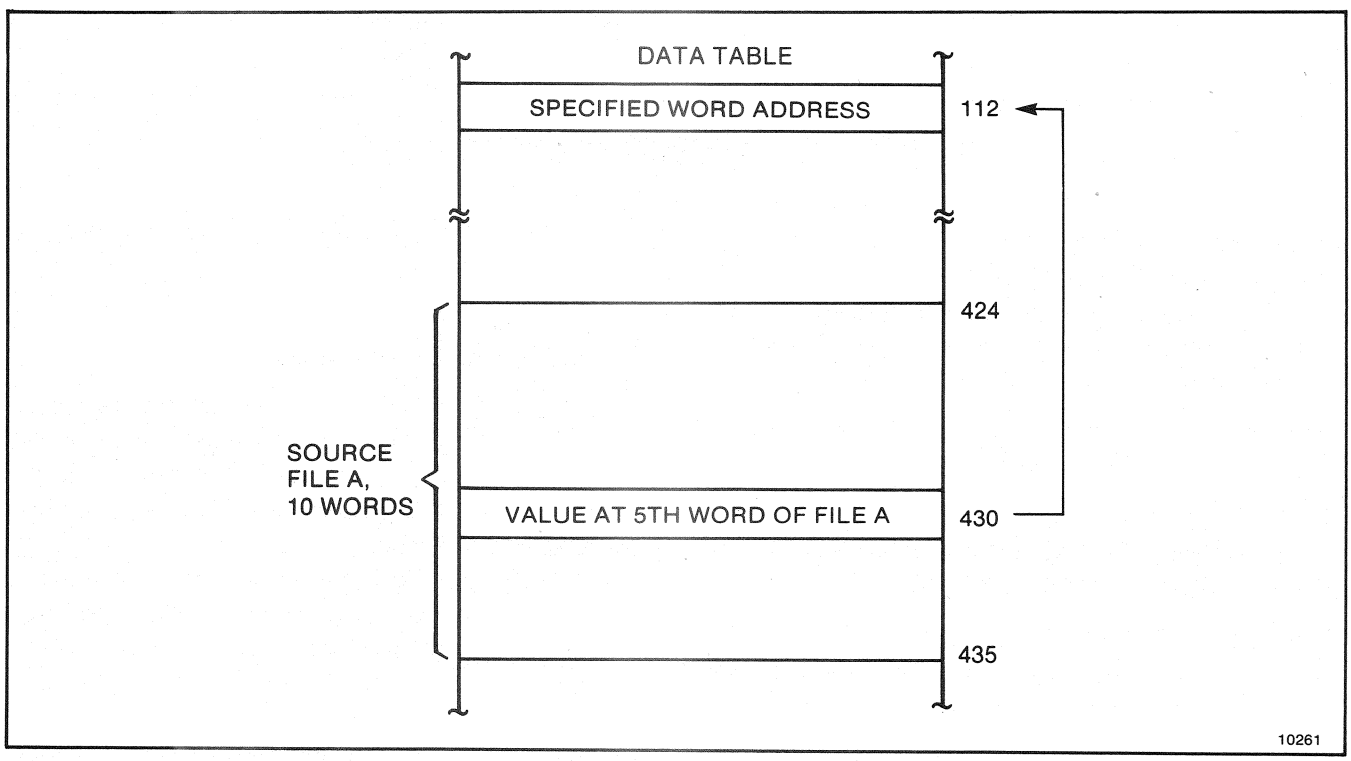
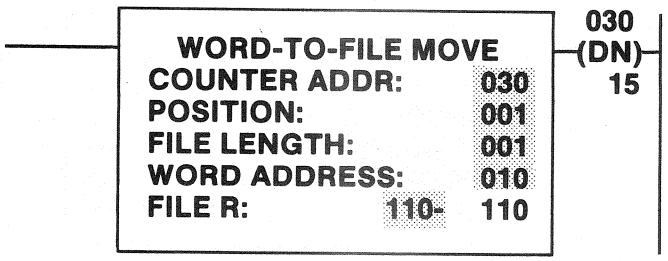


FIGURE 10-5 — FILE-TO-WORD MOVE Operation

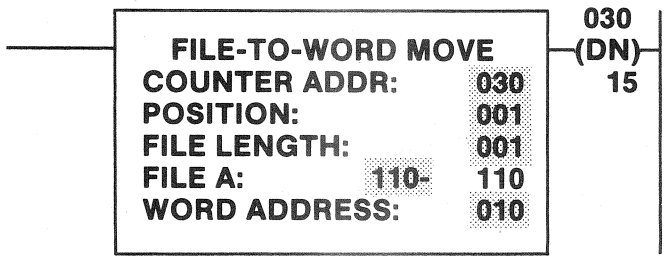


NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACE BY USER—ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INTIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

COUNTER ADDRESS	: ADDRESS OF THE INSTRUCTION IN THE ACCUMULATED VALUE AREA OF THE DATA TABLE.
POSITION	: CURRENT WORD BEING OPERATED UPON. (ACCUMULATED VALUE OF COUNTER).
FILE LENGTH	: NUMBER OF WORDS IN FILE (PRESET VALUE OF THE COUNTER).
WORD ADDRESS	: ADDRESS OF SOURCE WORD OUTSIDE THE FILE
FILE R	: STARTING ADDRESS OF DESTINATION FILE.

10161

FIGURE 10-6 — WORD-TO-FILE MOVE Format



NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACE BY USER—ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INTIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

COUNTER ADDRESS	: ADDRESS OF THE INSTRUCTION IN THE ACCUMULATED VALUE AREA OF THE DATA TABLE.
POSITION	: CURRENT WORD BEING OPERATED UPON. (ACCUMULATED VALUE OF COUNTER).
FILE LENGTH	: NUMBER OF WORDS IN FILE (PRESET VALUE OF THE COUNTER).
FILE A	: STARTING ADDRESS OF SOURCE FILE.
WORD ADDRESS	: ADDRESS OF DESTINATION WORD OUTSIDE THE FILE.

10158

FIGURE 10-7 — FILE-TO-WORD MOVE Format

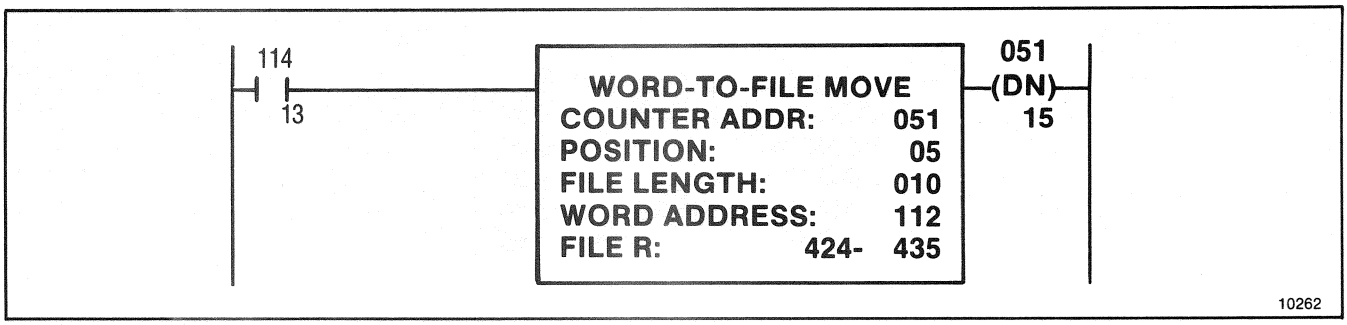


FIGURE 10-8 — WORD-TO-FILE MOVE Example Rung

Counter Address	051
Position (indexed by User Program)	005
File Length	010
Word Address	112
File R starting address	424

When switch 114/13 closes, the 16 bits of data in word 112_8 are copied into the 5th word of File R.

The procedure for using the Data Monitor mode to monitor, load or edit file data is presented in Section 9.2, Data Monitor Mode.

Refer to Table 10-1 for a summary of file instructions.

TABLE 10-1 — File Instructions

KEY SEQUENCE	1770-T3 DISPLAY	INSTRUCTION NOTES
[FILE] [1][0]	FILE-TO-FILE MOVE COUNTER ADDR: 030 POSITION: 001 FILE LENGTH: 001 FILE A: 110-110 FILE R: 110-110 RATE PER SCAN 001 030 (EN) 17 030 (DN) 15	Output Instruction. Modes: Complete, Distributed and Incremental. Counter is internally incremented by the instruction. Requires 5 words of User Program.
[FILE] [1][1]	WORD-TO-FILE MOVE COUNTER ADDR: 030 POSITION: 001 FILE LENGTH: 001 WORD ADDRESS: 010 FILE R: 110-110 030 (DN) 15	Output Instruction. Counter is externally indexed by User Program. Data is transferred every scan that rung is TRUE. Requires 4 words of User Program.
[FILE] [1][2]	FILE-TO-WORD MOVE COUNTER ADDR: 030 POSITION: 001 FILE LENGTH: 001 FILE A: 110-110 WORD ADDRESS: 010 030 (DN) 15	Same as WORD-TO-FILE

NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACED BY USER-ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INITIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

COUNTER ADDRESS	:	ADDRESS OF THE INSTRUCTION IN THE ACCUMULATED VALUE AREA OF DATA TABLE.
POSITION	:	CURRENT WORD BEING OPERATED UPON. (ACCUMULATED VALUE OF COUNTER).
FILE LENGTH	:	NUMBER OF WORDS IN FILE (PRESET VALUE OF THE COUNTER).
FILE A	:	STARTING ADDRESS OF SOURCE FILE
FILE R	:	STARTING ADDRESS OF DESTINATION FILE.
WORD ADDRESS	:	ADDRESS OF SOURCE WORD OR DESTINATION WORD OUTSIDE OF FILE.
RATE PER SCAN	:	NUMBER OF DATA WORDS MOVED PER SCAN.

Note: Access Data Monitor Display as follows:

Enter all instruction parameters. Press key sequence: [DISPLAY][0] for Binary, [DISPLAY][1] for Hex.

Section 11
SEQUENCER INSTRUCTIONS

11.0 GENERAL

Sequencer Instructions are powerful block instructions. They operate on up to 64 bits (4 words) at a time as compared with 1 bit for Examine and Energize instructions and 1 word for Data Transfer and File Transfer instructions. The three Sequencer instructions are:

SEQUENCER OUTPUT

SEQUENCER INPUT

SEQUENCER LOAD

Sequencer instructions can be used to transfer information from the Data Table to output word addresses for the control of sequential machine operation (SEQUENCER OUTPUT); to compare I/O word information with information stored in tables so that machine operating conditions can be examined for control and diagnostic purposes (SEQUENCER INPUT); and to transfer I/O word information into the Data Table (SEQUENCER LOAD). When used in combination and with other block and bit/word instructions, the potential to create powerful concise programs is nearly unlimited.

Note: It is assumed that the concepts, terminology and consequences introduced in Section 9, Files and Data Monitor Operation, have been thoroughly understood.

COMPARISON WITH FILE INSTRUCTIONS

Sequencer instructions are similar to File instructions but have some marked differences. Both are block instructions that contain a counter and a file. The instructions require the entry of more than one word address. Each has a corresponding Data Monitor display for monitoring, loading or editing file data.

File instructions operate on files that are one word or 16 bits wide. In contrast, Sequencer instructions operate on files that are up to four words or 64 bits wide. A sequencer file can be represented graphically by a sequencer table. The length or number of steps (rows) in a sequencer table can be up to 999. The width of a sequencer table can be up to four words (columns) as shown in Figure 11-1.

Note: The Data Table is one word wide by many words long. A sequencer table appears in the Data Table as one continuous file. The length of the file equals the product of the number of words wide (columns) and the number of steps as shown in Figure 11-2. As an example, a 24-step x 4 words wide sequencer table will occupy 96 consecutive words in the Data Table.

Internally indexed File instructions, when enabled, perform the operation and then increment to the next step. In contrast, internally

STEP	WORD 1	WORD 2	WORD 3	WORD 4
001	00110101 11000101	00011101 11001010	10111011 11001011	01011101 01011111
002	01110100 00011101	00010111 00110011	"	01010101 01010101
003	"	"	"	"
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
"	"	"	"	"
024	00010101 10100000	10100010 10101000	01010000 01011111	10111100 00110011

10194

FIGURE 11-1 — Sequencer Table

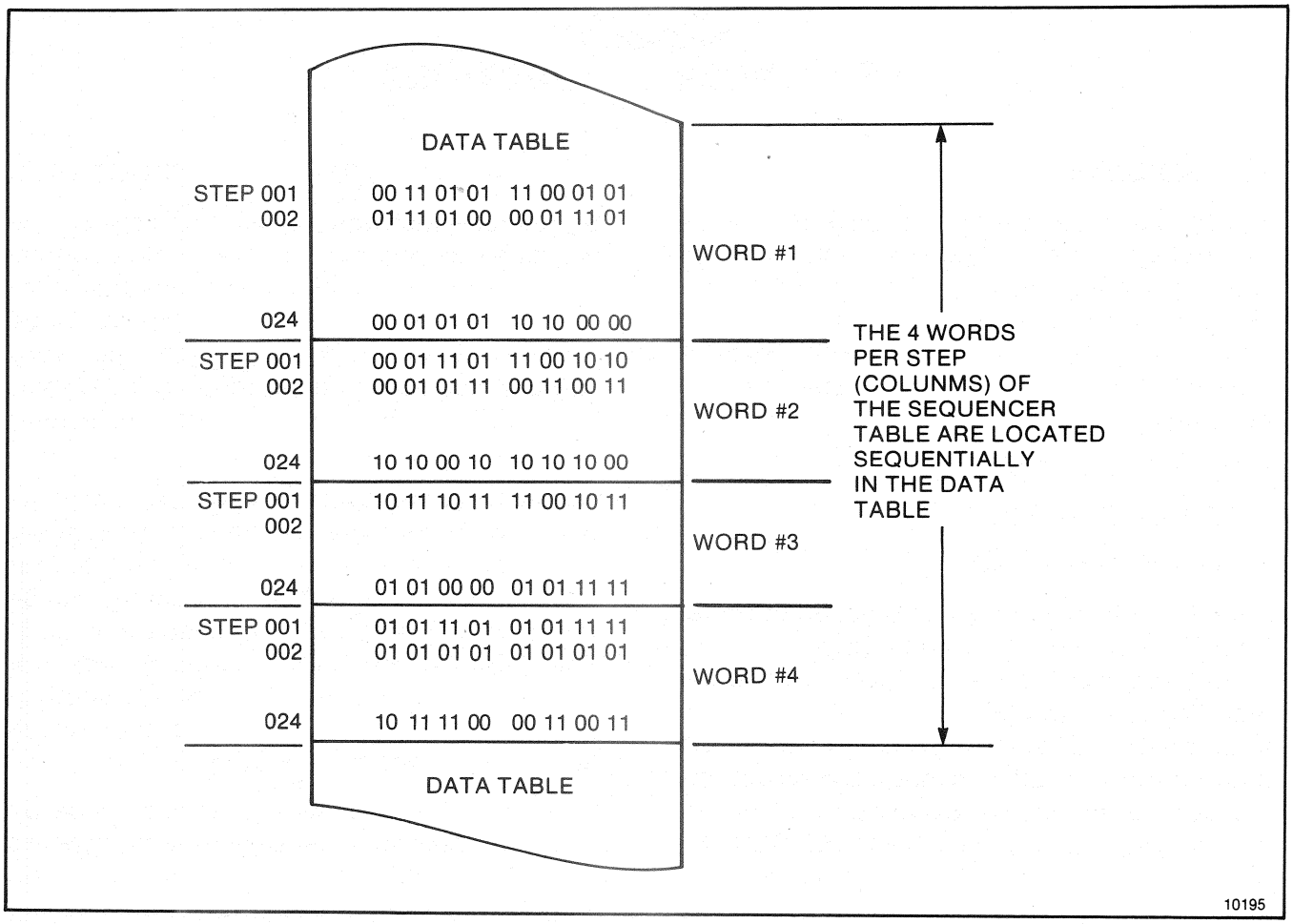


FIGURE 11-2 — Sequencer Table Format in the Data Table

indexed Sequencer instructions, when enabled, increment to the next step and then the operation is performed.

11.1 SEQUENCER OUTPUT INSTRUCTION

The SEQUENCER OUTPUT instruction functions in a manner analogous to a mechanical drum sequencer.

11.1.1 Sequencer Output Analogy

Consider a music box mechanism containing a cylinder with rows of pegs. As the cylinder turns, the pegs produce tones (outputs) as they strike the spring resonators. In this analogy, the pegs on the cylinder wall are analagous to "1"s in bit locations in the sequencer table. As the cylinder turns continuously through many steps, each step presents a new row of peg locations. One or more pegs produce a single tone or a musical chord.

If the cylinder wall containing the pegs could be removed, cut along its axis to separate the first and last rows, and flattened, the resulting rectangular surface would be a representation of a sequencer table. The rows or steps would be numbered from top to bottom. The pegs representing "1"s in bit locations would be numbered across the top as shown in Figure 11-3.

The SEQUENCER OUTPUT instruction can control up to 64 outputs with 999 steps.

11.1.2 Instruction Operation

When the rung containing the SEQUENCER OUTPUT instruction goes from FALSE-to-TRUE, the counter increments to the next step in the sequencer table. The data found there is outputted to the Output Word Address(es) that were specified in the Sequencer instruc-

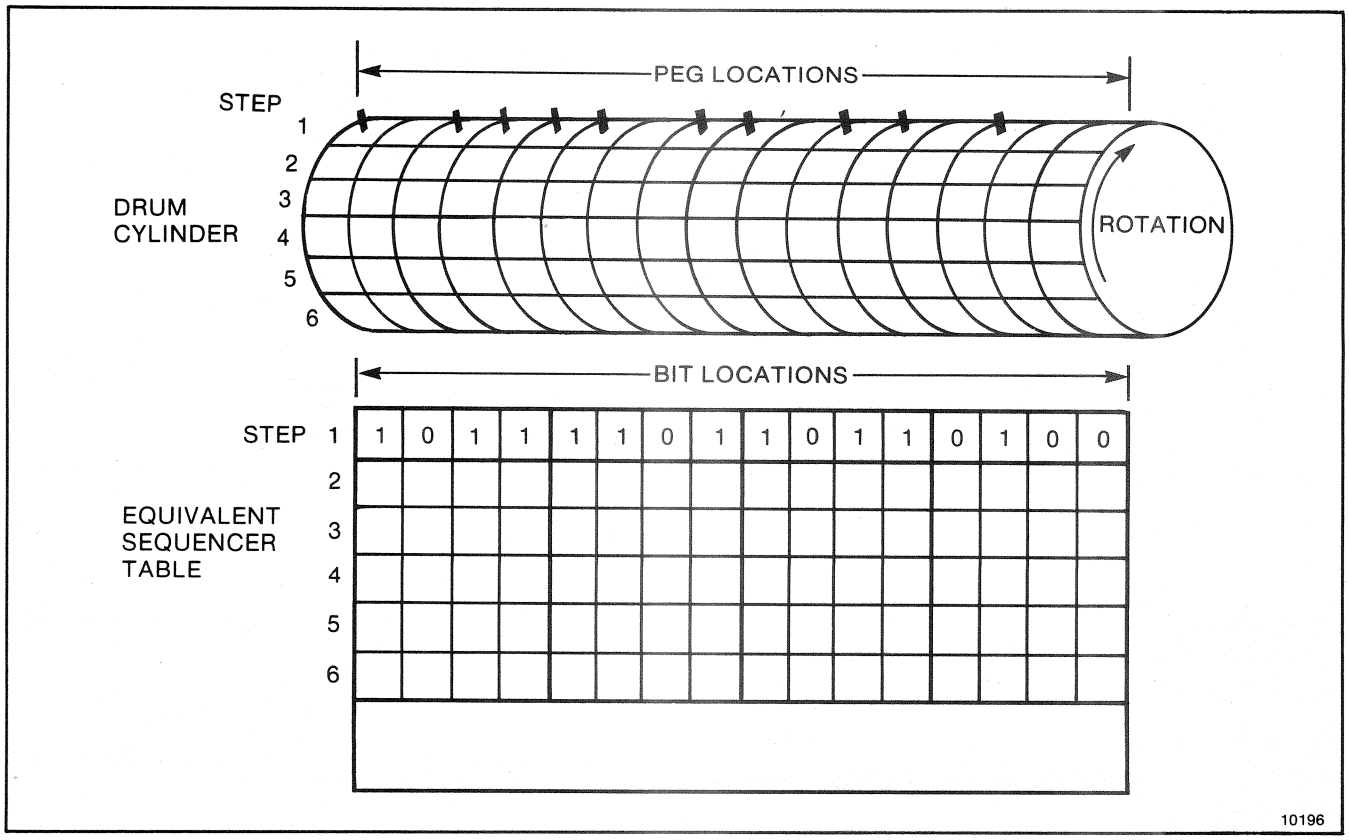


FIGURE 11-3 — Sequencer Output Analogy

tion. The Output Word Addresses need not be consecutive.

The SEQUENCER OUTPUT instructiuon operates whenever the rung is TRUE. This means that once the rung is enabled, the counter is incremented to the next step and the data in that step will be outputted every scan that the rung remains TRUE. The instruction will not advance to the next step until there is a FALSE-to-TRUE transition of the rung condition.

When the last step of the sequencer table is outputted (AC = PR), the Done bit is set. The next FALSE-to-TRUE transition of the rung condition will start the instruction operation again at step 1. The instruction counter should be set to zero for start-up purposes if it is desirable to start at step 1 when the instruction is enabled for the first time.

Note: When the rung is FALSE, data is not transferred by the instruction and outputs remain in their last state unless changed by instructions elsewhere in the User Program.

11.1.3 Masking Output Data

A mask is a means of selectively screening out data. The purpose of the mask in the SEQUENCER OUTPUT instruction is to allow (unused) bits of the Output Word Addresses specified in the instruction to be used independently. For example, the number of output bits required for sequential operation can be less than 16, 32, 48 or 64 corresponding to a 1, 2, 3 or 4-word Sequencer instruction. When fewer outputs are required, masking allows the unused output terminals of a module that is specified in the Sequencer instruction to be used to control output devices which are independent of the sequencer operation.

A zero in a mask bit location prevents the instruction from operating on the data in the corresponding bit location. A 1 in a mask bit location allows the corresponding bit to be

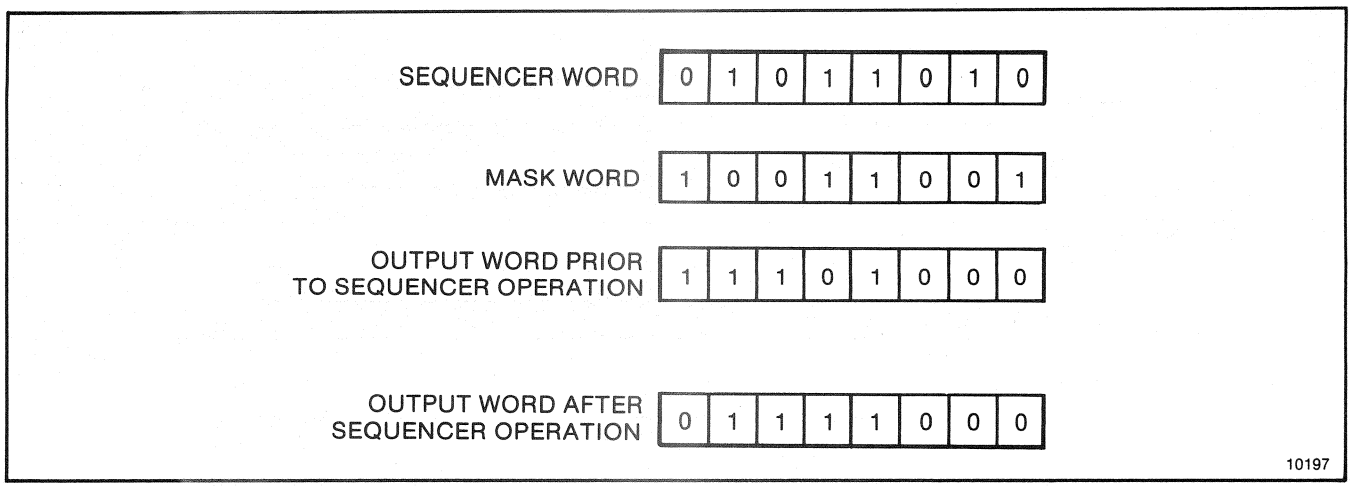


FIGURE 11-4 — Masking Transfer Data

operated upon. When all the output data bits are relevant to the instruction, a mask of all 1's should be used.

A mask can be controlled by other instructions in User Program. If a changing mask is desired for different steps in the Sequencer, a GET-PUT transfer or FILE-TO-FILE MOVE can be used.

A Mask Word must be specified for each Output Word used in the instruction. When the first Mask Word Address is entered, the Industarial Terminal will automatically assign the next 1, 2 or 3 consecutive word address(es) for the required number of mask words.

WARNING: When choosing a Mask Word Address, be sure that the next 1, 2 or 3 consecutive word addresses are not already assigned. Other data written into a mask could cause unpredictable machine operation. Damage to equipment and/or injury to personnel could occur.

The example in Figure 11-4 shows the result of masking the transfer of data bits. Although a mask contains 16 bits, 8 bits are used for the purpose of the illustration.

11.1.4 SEQUENCER OUTPUT Instruction Characteristics

- Output instruction.
- When executed, the counter increments, then data is transferred.
- Same data is transferred each scan that the rung is TRUE.
- Counter is indexed by the instruction.
- Unused bits in the Output Word addresses can be masked.
- Requires 5-8 words of User Program, depending on the number of Output Words used.
- Key sequence [SEQ][0].

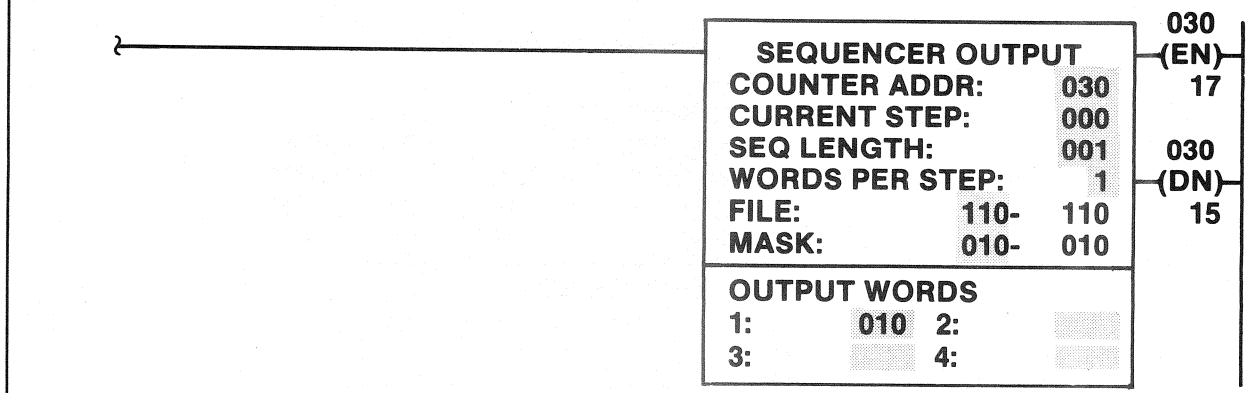
11.1.5 Programming the SEQUENCER OUTPUT Instruction

To program a SEQUENCER OUTPUT Instruction, press the key sequence [SEQ][0]. A display represented by Figure 11-5 will appear.

WARNING: The counter address of the SEQUENCER OUTPUT instruction should be reserved for that instruction. Do NOT manipulate the counter Preset or Accumulated word. Changes to these values could result in unpredictable machine operation or a Run-Time Error. Damage to equipment and/or injury to personnel could occur.

An example rung containing the SEQUENCER OUTPUT instruction is shown in Figure 11-6. The following parameters have been entered into the instruction:

Counter Address	054
Current Step	007
Sequencer Length	009



NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACE BY USER—ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INTIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

COUNTER ADDRESS : ADDRESS OF THE INSTRUCTION IN ACCUMULATED VALUE AREA OF DATA TABLE.

CURRENT STEP : POSITION IN SEQUENCER TABLE (ACCUMULATED VALUE OF COUNTER

SEQ LENGTH : NUMBER OF STEPS (PRESET VALUE OF COUNTER)

WORDS PER STEP : WIDTH OF SEQUENCER TABLE (NUMBER OF COLUMNS)

FILE : STARTING ADDRESS OF SEQUENCER TABLE

MASK : STARTING ADDRESS OF MASK FILE

OUTPUT WORDS : WORDS CONTROLLED BY THE INSTRUCTION.

10198

FIGURE 11-5 — SEQUENCER OUTPUT Format

Words per Step	2
File	600-621
Mask	211-212
Output Words	011, 013

When switch 114/14 closes, the SEQUENCER OUTPUT instruction increments to step 008 and controls the 32 outputs corresponding to the specified Output Words (less those outputs that are masked). The control of the output terminals will be in accordance with the data stored in step 008 of the sequencer table and mask conditions as shown in Figure 11-7.

It will be necessary to use the Data Monitor mode to set the Mask Word bits in order for the instruction to operate. It may also be necessary to load data into the sequencer table. The procedure for using the Data Monitor mode to monitor, load or edit file data is presented in Section 9.2, Data Monitor Mode.

The Binary Data Monitor display for a SEQUENCER OUTPUT instruction with 9

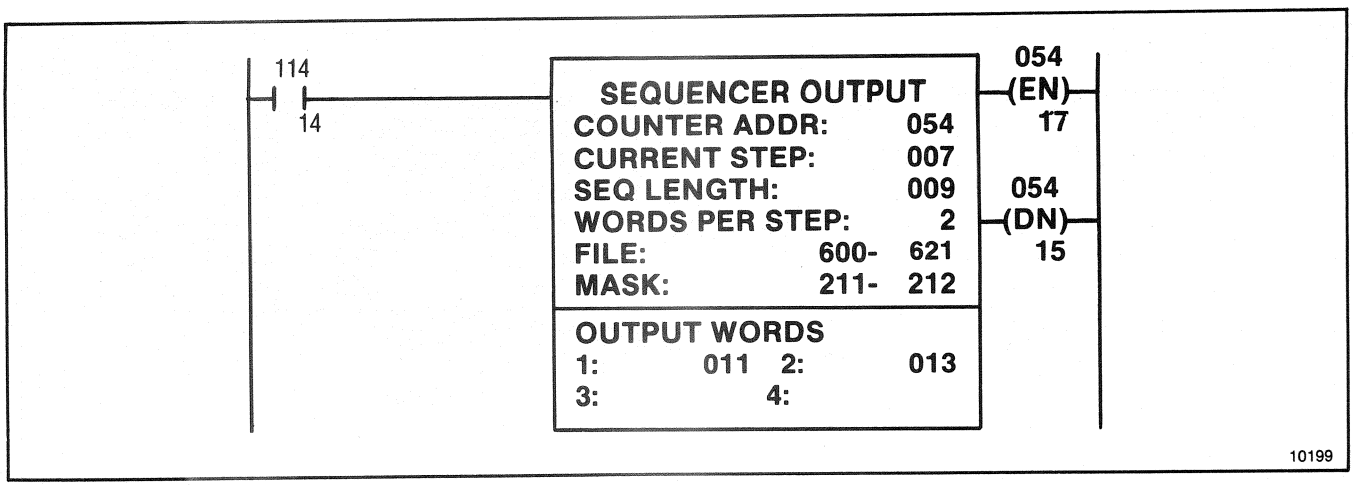


FIGURE 11-6 — SEQUENCER OUTPUT Example Rung

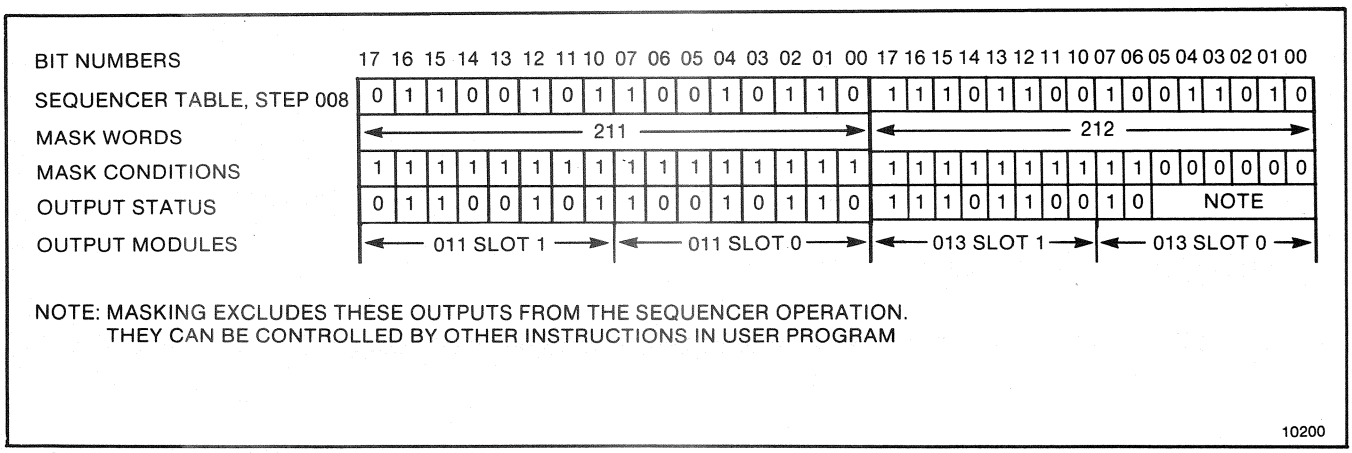


FIGURE 11-7 — Control of Sequencer Outputs

steps and 2 words per step is shown in Figure 11-8. Differences between the Data Monitor display of a Sequencer instruction and the display of a File instruction should be noted. The SEQUENCER OUTPUT instruction will be used as an example. Each column in the sequencer table represents the data for each Output word. This data controls the outputs of the corresponding Output Word Addresses at each step in the sequencer operation. The status of the outputs can be observed in the Header display of Output Address Data. In contrast, each column in a File instruction display represents a complete file. File data is manipulated by the instruction. The results of the instruction operation is contained in a Result file or in a specified Word Address of the instruction.

11.2 SEQUENCER INPUT INSTRUCTON

The SEQUENCER INPUT instruction is a rung conditioning instruction. It compares machine input and other input data with data stored in the Data Table for equality. It can be used alone or in a series and/or parallel combination with other rung condition instructions to determine the status of outputs.

11.2.1 Instruction Operation

The SEQUENCER INPUT instruction compares the status of up to 64 input conditions with the contents stored in a sequencer table, bit-by-bit and step-by-step. When the status of all input bits becomes equal to the status of all bits in the current step of the sequencer table, the instruction becomes logically TRUE.

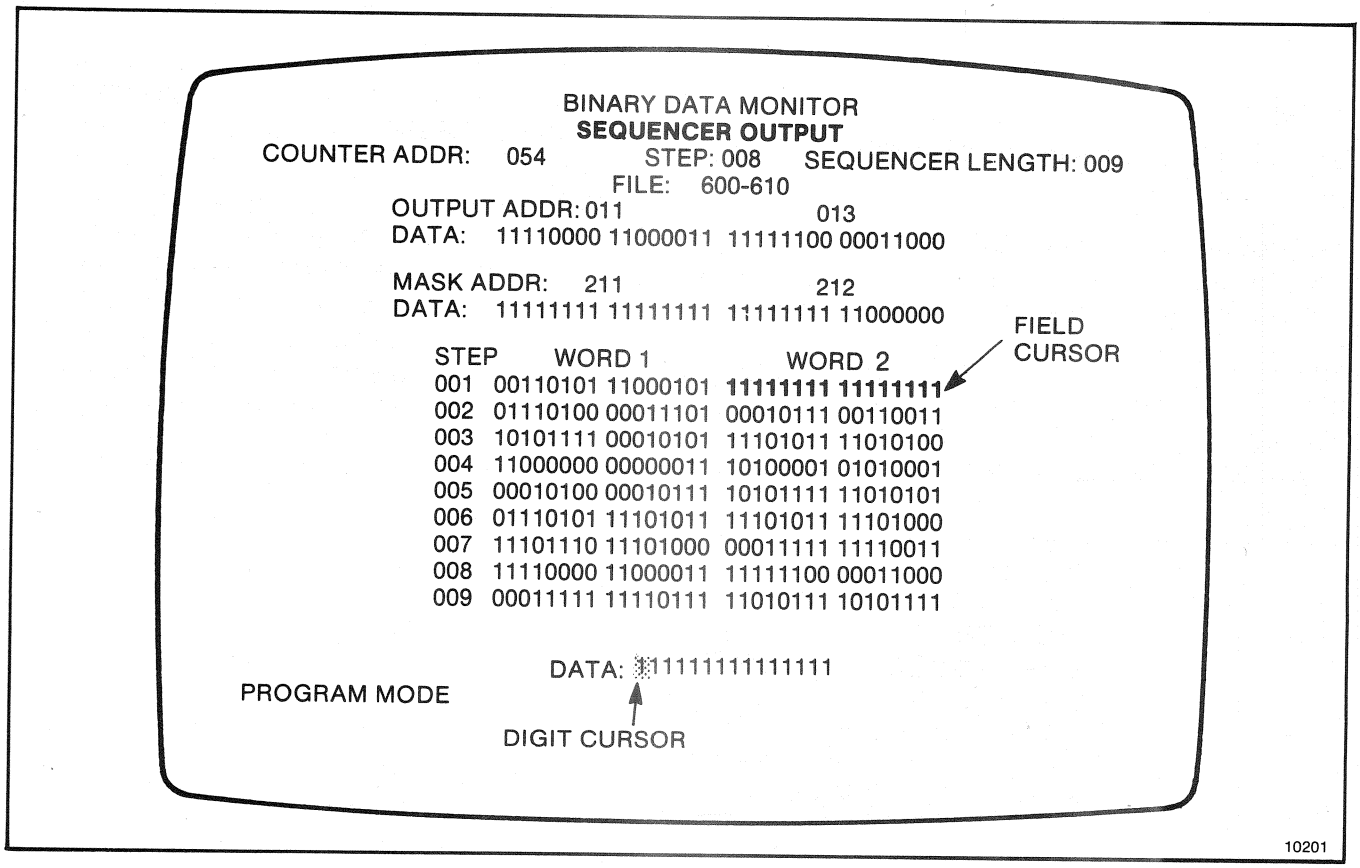


FIGURE 11-8 — Example Binary Data Monitor Display

The SEQUENCER INPUT instruction contains a step counter that points to the step in the sequencer file being operated upon. The counter is NOT controlled by the instruction. Its Accumulated value is indexed by logic from elsewhere in the User Program.

The SEQUENCER INPUT instruction can be programmed in the same rung as a SEQUENCER OUTPUT instruction and its step counter can be indexed by the SEQUENCER OUTPUT instruction. The step counter in both instructions is given the same address. When programmed in this manner, the SEQUENCER INPUT and OUTPUT instructions will track through a controlled sequence of operation. The length of the sequence is equal to the number of steps in the sequencer table.

11.2.2 Masking

Up to four Input Word Addresses can be specified in the SEQUENCER INPUT Instruction. Each Input Word has a corresponding Mask Word. The mask is applied to the data at the Input Word Address when the bit comparisons are made. When the number of input bits used by the instruction is less than 16, 32, 48 or 64 for a 1, 2, 3 or 4-word Sequencer instruction, the bits not used by the instruction should be masked. This allows unused bits (input terminals) of the specified Input Word to be used for inputs which are independent of the sequencer operation. See paragraph 11.1.3 for additional information on Masking.

11.2.3 SEQUENCER INPUT Instruction Characteristics

- Input instruction.
- Compares input data with current step in sequencer table for equality.
- Counter must be externally indexed by other instructions in User Program.
- Unused bits in the Input Word addresses can be masked.

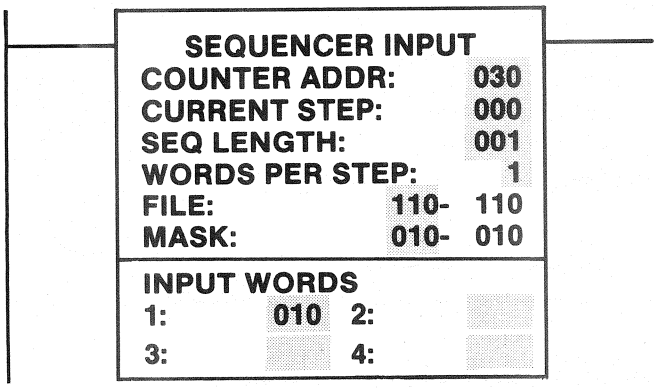
- Requires 5-8 words of User Program depending on number of Input Words used.
- Key sequence [SEQ][1]

11.2.4 Programming the SEQUENCER INPUT Instruction

To program a SEQUENCER INPUT instruction, press the key sequence [SEQ][1]. A display represented by Figure 11-9 will appear.

The format of the instruction block is similar to that of the SEQUENCER OUTPUT Instruction except that it is a rung conditioning instruction. The words specified by the Input Word Addresses can be those of I/O modules and/or storage words.

WARNING: The counter address of the SEQUENCER INPUT Instruction should be used only for the intended instruction and the corresponding instructions which manipulate the Accumulated value. Do NOT inadvertently manipulate the Preset or Accumulated word. Changes to these values could result in unpredictable machine operation or a Run-Time Error. Damage to equipment and/or injury to personnel could occur.



NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACE BY USER—ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INTIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

COUNTER ADDRESS	: ADDRESS OF THE INSTRUCTION IN ACCUMULATED VALUE AREA OF DATA TABLE.
CURRENT STEP	: POSITION IN SEQUENCER TABLE (ACCUMULATED VALUE OF COUNTER
SEQ LENGTH	: NUMBER OF STEPS (PRESET VALUE OF COUNTER)
WORDS PER STEP	: WIDTH OF SEQUENCER TABLE (NUMBER OF COLUMNS)
FILE	: STARTING ADDRESS OF SEQUENCER TABLE
MASK	: STARTING ADDRESS OF MASK FILE
INPUT WORDS	: WORDS MONITORED BY THE INSTRUCTION.

10202

FIGURE 11-9 — SEQUENCER INPUT Format

An example rung containing the SEQUENCER INPUT instruction is shown in Figure 11-10. The following parameters have been entered into the instruction.

Counter Address	0055
Current Step	006
Sequencer Length	014
Words per Step	4
File	0430-0517
Mask	0214-0217
Input Words	0110,0112
	0114,0115

When the status of all 64 inputs corresponding to the specified Input Words (less those inputs that are masked) is equal to the status of all 64 bits of data in step 006 of the sequencer table, the logic of the instruction is TRUE. The storage bit 356/01 is then turned ON.

It will be necessary to use the Data Monitor mode to set the Mask Word bits in order for the instruction to operate. It may also be necessary to load data into the sequencer table. The Data Monitor display for the SEQUENCER INPUT instruction is similar to that of the SEQUENCER OUTPUT instruction shown in Figure 11-8.

The procedure for using the Data Monitor mode to monitor, load or edit file data is presented in Section 9.2, Data Monitor Mode.

11.3 SEQUENCER LOAD INSTRUCTION

The SEQUENCER LOAD instruction is an output instruction. It is used to load data into Data Table locations such as files or sequencer tables.

11.3.1 Instruction Operation

The SEQUENCER LOAD instruction receives data from up to four (4) independent Data Table word addresses specified in the instruction. The Load Word Address(es) can represent input, output and/or storage words. The Load Word Addresses need not be consecutive.

The instruction can load words into a sequencer table or elsewhere in the Data Table, one step at a time, at the location determined by the step counter. The step counter is controlled by the instruction and it increments prior to loading the step.

The SEQUENCER LOAD instruction is enabled by a FALSE-TO-TRUE transition of the rung condition. Data from the Load Words will be loaded into the sequencer table during the scan that the instruction was enabled. If the rung remains TRUE for subsequent scans, the instruction will not repeat. The instruction will advance to the next step only on the next FALSE-to-TRUE transition of the rung condition.

When the last step of the sequencer table is loaded (AC = PR), the Done bit is set. The next FALSE-to-TRUE transition of the rung condition will load the data beginning in step 1. The counter should be set to zero for start-up purposes if it is desirable to start at step 1

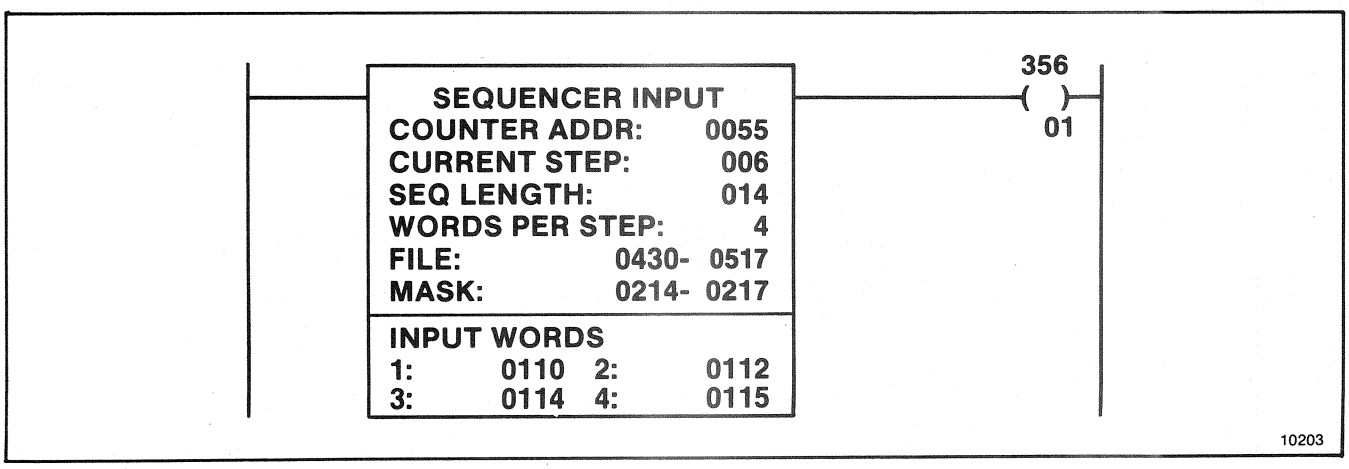


FIGURE 11-10 — SEQUENCER INPUT Example Rung

when the instruction is enabled for the first time.

The instruction does not utilize mask words. If the SEQUENCER LOAD instruction is used with other sequencer instructions, they will perform the required masking.

The SEQUENCER LOAD instruction can be used for machine diagnostics to load a sequencer input or output table or a Data Table file with data representing the desired sequence of machine operation. If/when the actual sequence of operation becomes mismatched with the desired sequence of operation as contained in the SEQUENCER INPUT instruction, a fault signal can be enabled by means of the User Program.

The SEQUENCER LOAD Instruction can also be used to "teach" the Processor the desired sequential operation. The I/O conditions representing the desired operation can be loaded into the sequencer input tables as the machine is manually stepped through the control cycle.

11.3.2 SEQUENCER LOAD Instruction Characteristics

- Output Instruction.
- When executed, the counter increments, then data is loaded.
- Counter is incremented by the instruction.
- Instruction does NOT utilize a mask.
- Requires 4-7 words of User Program depending on the number of Load Words used.
- Key sequence [SEQ][2].

11.3.3 Programming the SEQUENCER LOAD Instruction

To program a SEQUENCER LOAD instruction in the Ladder Diagram mode, press the key sequence [SEQ][2]. A display represented by Figure 11-11 will appear.

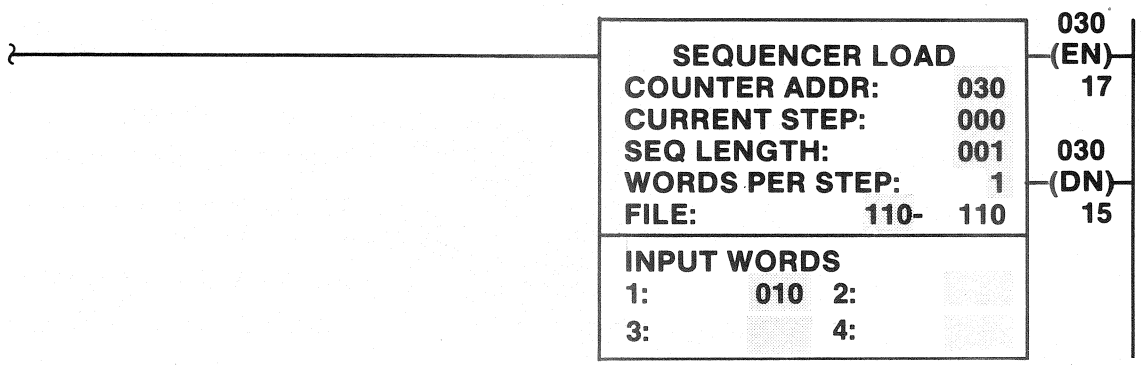
WARNING: The counter address of the SEQUENCER LOAD instruction should be reserved for that instruction. Do NOT manipulate the counter Accumulated or Preset word. Changes to these values could result in unpredictable machine operation or a Run-Time Error. Damage to equipment and/or injury to personnel could occur.

An example rung containing the SEQUENCER LOAD instruction is shown in Figure 11-12. The following parameters have been entered into the instruction:

Counter Address	0056
Current Step	008
Sequencer Length	012
Words Per Step	4
File	0510-0567
Load Words	0111,0113
	0012,0314

When switch 114/16 closes the SEQUENCER LOAD instruction increments to step 009. The data from Load words 0111_8, 0113_8, 0012_8 and 0314_8 are loaded into step 009 of the sequencer table in one scan. Thereafter, no data can be loaded until switch 114/16 opens then closes again.

Refer to Table 11-1 for a summary of sequence instructions.



NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACE BY USER—ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INTIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

COUNTER ADDRESS : ADDRESS OF THE INSTRUCTION IN ACCUMULATED VALUE AREA OF DATA TABLE.

CURRENT STEP : POSITION IN SEQUENCER TABLE (ACCUMULATED VALUE OF COUNTER

SEQ LENGTH : NUMBER OF STEPS (PRESET VALUE OF COUNTER)

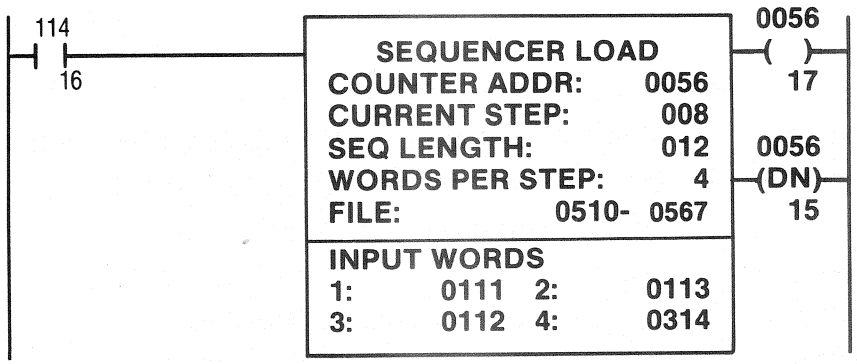
WORDS PER STEP : WIDTH OF SEQUENCER TABLE (NUMBER OF COLUMNS)

FILE : STARTING ADDRESS OF SEQUENCER TABLE

LOAD WORDS : WORDS FETCHED BY THE INSTRUCTION.

10204

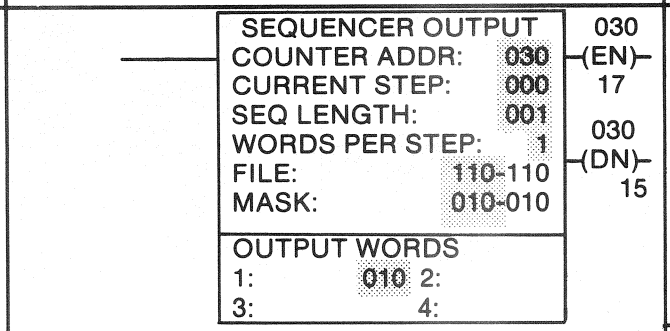
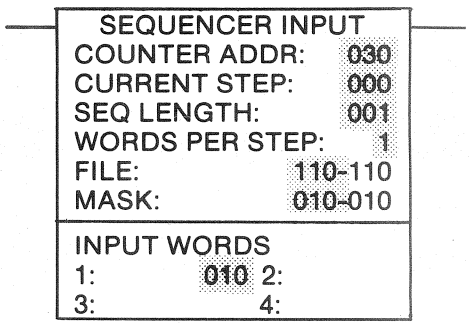
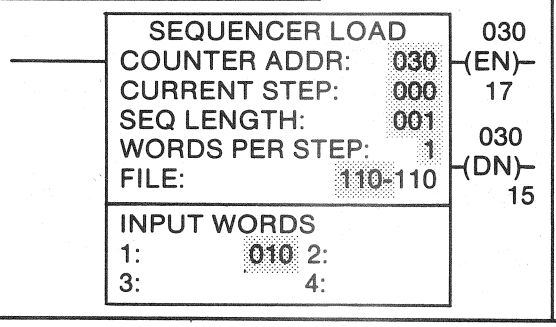
FIGURE 11-11 — SEQUENCER LOAD Format



10205

FIGURE 11-12 — SEQUENCER LOAD Example Rung

TABLE 11-1 — Sequencer Instructions

KEY SEQUENCE	1770-T3 DISPLAY	INSTRUCTION NOTES
[SEQ][0]		Output Instruction. Increments, then transfers data. Same data transferred each scan that the rung is TRUE. Counter is indexed by the instruction. Unused Output bits can be masked. Requires 5-8 words of User Program.
[SEQ][1]		Input Instruction. Compares Input data with current step for equality. Counter must be externally indexed by User Program. Unused Input bits can be masked. Requires 5-8 words of User Program.
[SEQ][2]		Output Instruction. Increments, then loads data. Counter is indexed by the instruction. Does not mask. Requires 4-7 words of User Program.

NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACED BY USER-ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INITIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

COUNTER ADDRESS	:	ADDRESS OF THE INSTRUCTION IN ACCUMULATED VALUE AREA OF DATA TABLE.
CURRENT STEP	:	POSITION IN SEQUENCER TABLE (ACCUMULATED VALUE OF COUNTER)
SEQ LENGTH	:	NUMBER OF STEPS (PRESET VALUE OF THE COUNTER).
WORDS PER STEP	:	WIDTH OF SEQUENCER TABLE (NUMBER OF COLUMNS)
FILE	:	STARTING ADDRESS OF SEQUENCER TABLE
MASK	:	STARTING ADDRESS OF MASK FILE
OUTPUT WORDS	:	WORDS CONTROLLED BY THE INSTRUCTION.
LOAD WORDS	:	WORDS FETCHED BY THE INSTRUCTION.
INPUT WORDS	:	WORDS MONITORED BY THE INSTRUCTION.

Note: Access Data Monitor Display as follows:
Enter all instruction parameters. Press key sequence: [DISPLAY][0] for Binary, [DISPLAY][1] for Hex.

Section 12
WRITING THE USER PROGRAM

12.0 GENERAL

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

12.1 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.

12.2 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 12-1 shows a part moving along a conveyor belt. Each part will trip a series of limit switches and will be sorted according to its height. The desired height 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) is latched

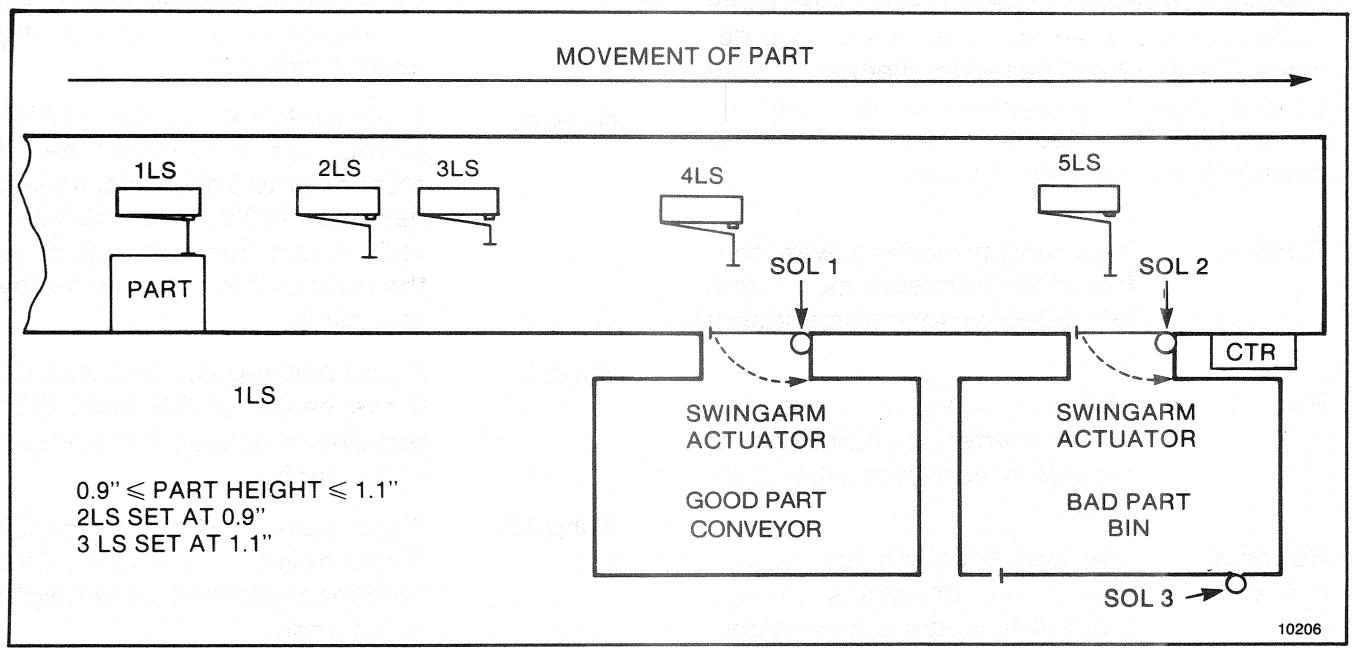


FIGURE 12-1 — Conveyor Belt Example

ON. Although the part will trip 4LS, it will continue along and trip 5LS, which energizes SOL2. The swingarm actuator will direct the part into the bad part bin. Each time a part enters the bad part bin, a counter is incremented. 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 flow 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 12-2). Data Table addresses are assigned to the hardwired devices. (Table 12-1) 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 indicator.

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

Rung 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.

TABLE 12-1 — 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)	114/00
Conveyor RUN Indicator	114/01
Good Part Solenoid (SOL1)	114/02
Bad Part Solenoid (SOL2)	114/03
Bin Dump Solenoid (SOL3)	114/04
JAM Detect Indicator	114/05
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

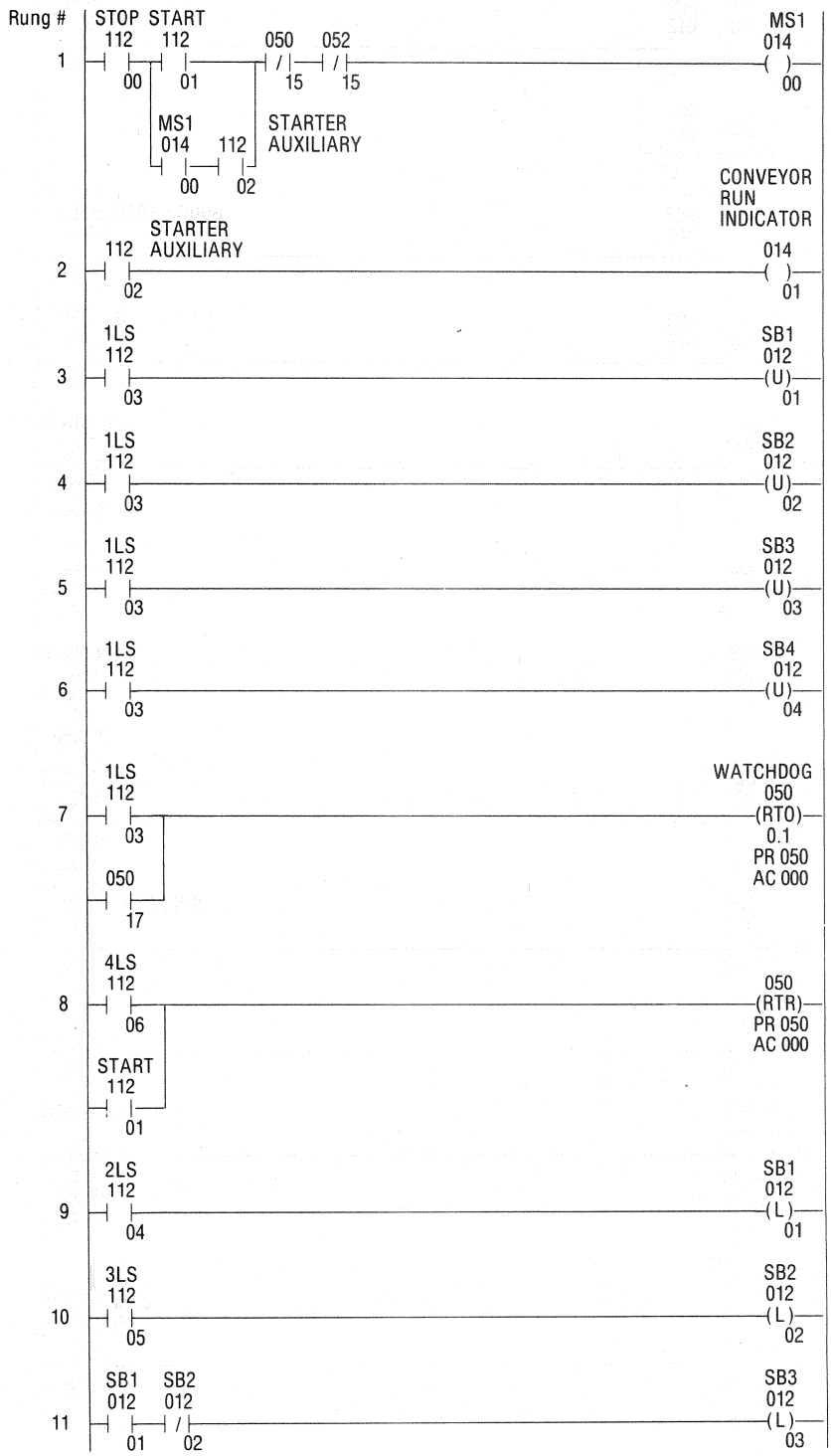


FIGURE 12-2 — Ladder Diagram Progam

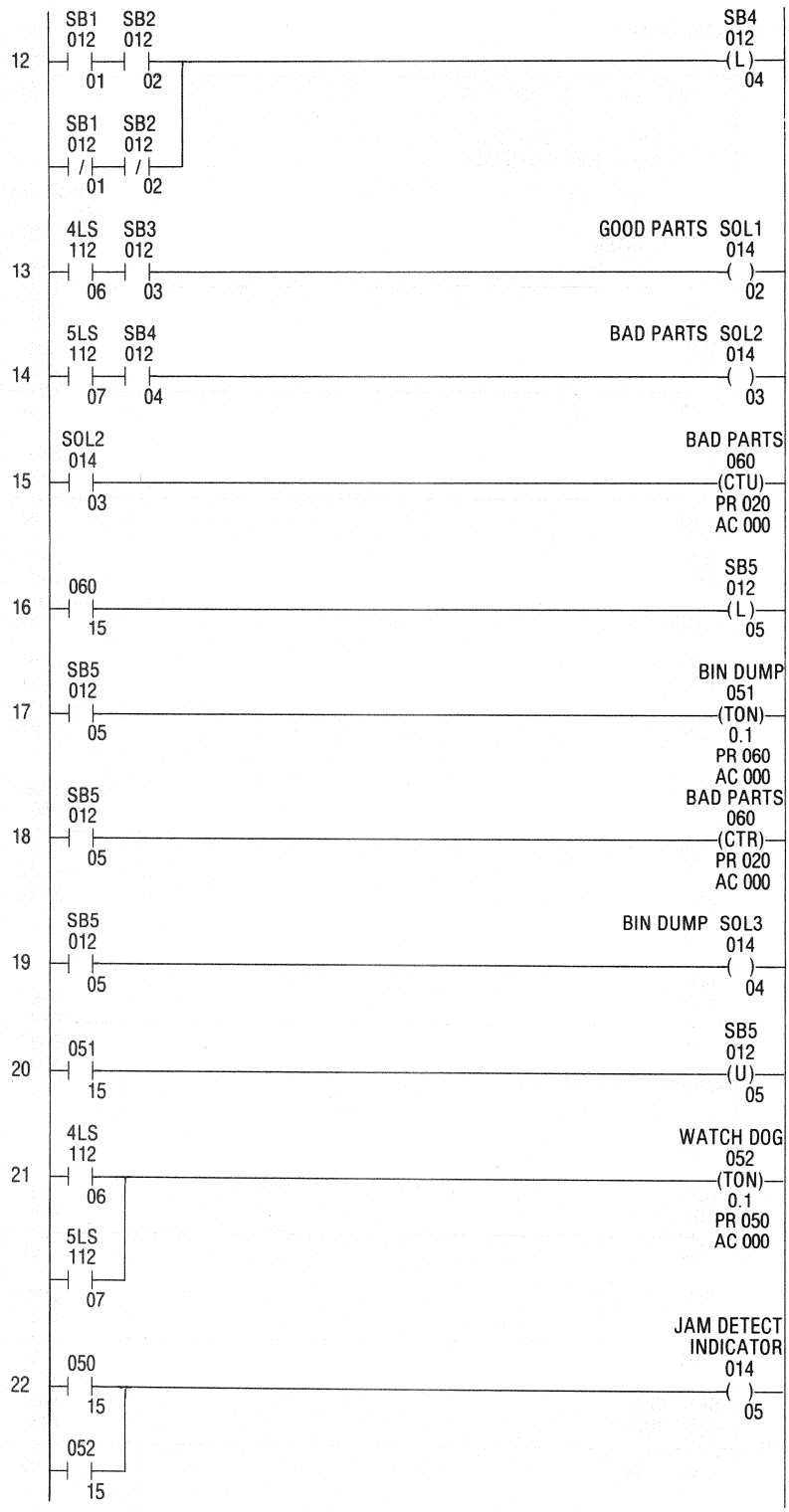


FIGURE 12-2 — Ladder Diagram Program (Cont.)

Rung 12. A part out of tolerance latches SB4.

Rung 13. A good part at 4LS actuates SOL1 with swing arm 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.

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 timer 052_8. If 4LS or 5LS is held closed by a jam, this timer will time out.

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

12.3 DEVELOPING THE DATA TABLE

The factory configured and user-expandable Data Table configurations shown in Section 1, Figures 1-6 and 1-7, respectively, should be used as guides 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.

12.3.1 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 Maps to describe the Data Table as a whole. Publication 5044 and 5045.

2) Data Table Bit and Word Assignment Sheets to log and describe the function of assigned addresses. Publication 5046, 5047 and 5048.

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

DATA TABLE WORD MAP (1024 WORD)

This form can be used to map the addresses of grouped Data Table words and to concisely describe the function of each group. The groups can include I/O Image Tables, BLOCK TRANSFER, TIMER/COUNTER, FILE and SEQUENCER Instructions, Files and Sequencer Tables, etc.

The form has pre-numbered rows representing addresses from $000\text{-}777_8$. Each row has 32 spaces where each space represents one word address. Any group of related word addresses can be designated on the map by labeling or color coding the spaces representing their addresses. For example, Figure 12-3 shows a completed portion of the Data Table Word Map. The timer/counter Accumulated values are labeled in the spaces defined by word address 040_8 through 071_8. Other Data Table areas are similarly labeled.

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. In particular, it can be used to log initial conditions of files such as those used for recipes, and to log assigned storage bits.

The lower two digits of the 3 or 4-digit word addresses are pre-numbered in the left-hand column. The bit numbers, 00-17, complete the 5 or 6-digit bit address. The starting word address can be written once for the entire 64 word column.

For example, Figure 12-4 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

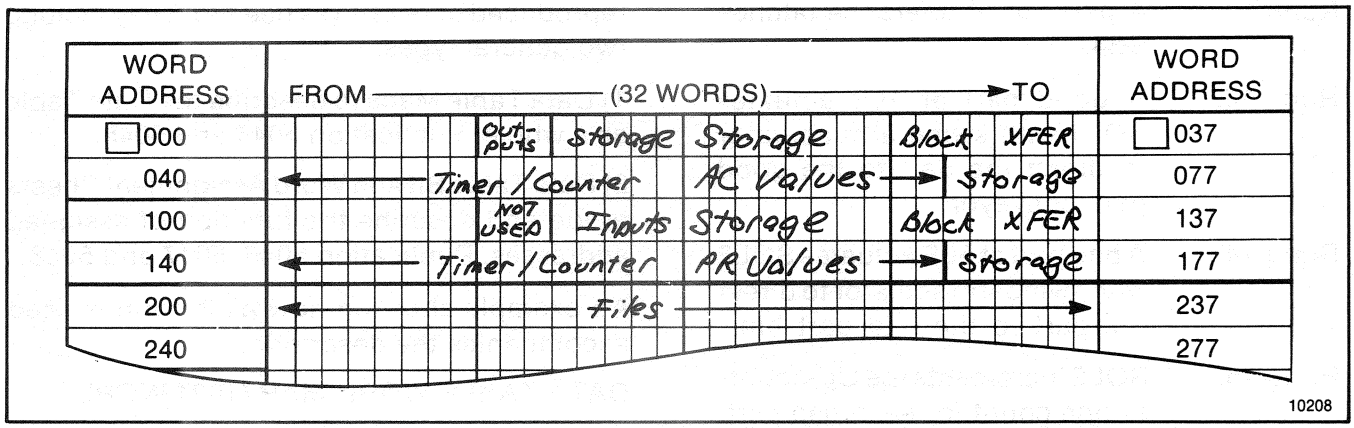


FIGURE 12-3 — Example of a Data Table Word Map

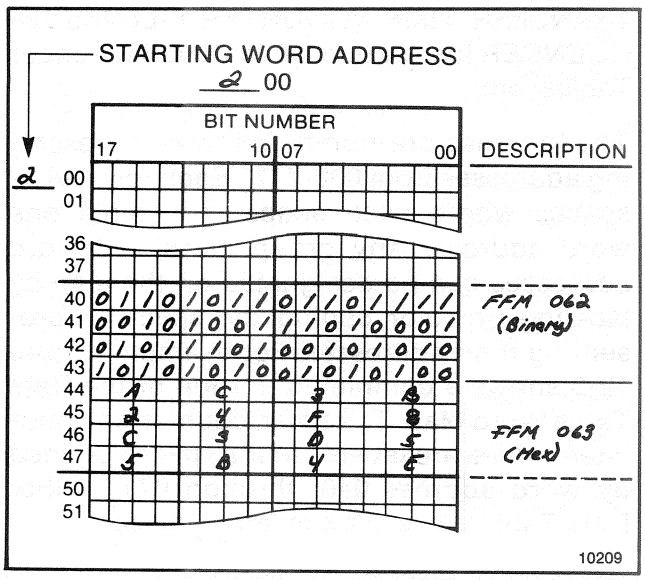


FIGURE 12-4— Example of a Data Table Map

starting word address box at the top of the column. Two 4-word files are illustrated. The data of the FILE-TO-FILE MOVE instruction, FFM062, is entered in binary. The data of the FILE-TO-FILE MOVE instruction, FFM063, is entered in hex.

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 or 4-digit word address is pre-numbered, 0-7.

For example, a portion of the Data Table Word Assignment Sheet is shown in Figure 12-5. It illustrates timer and counter functional descriptions for Accumulated values starting at word address 200_8 and Preset values starting at 300_8. A "20" and "30" 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 12-6. It illustrates logging the input devices associated Module Group 2 and the storage bits of

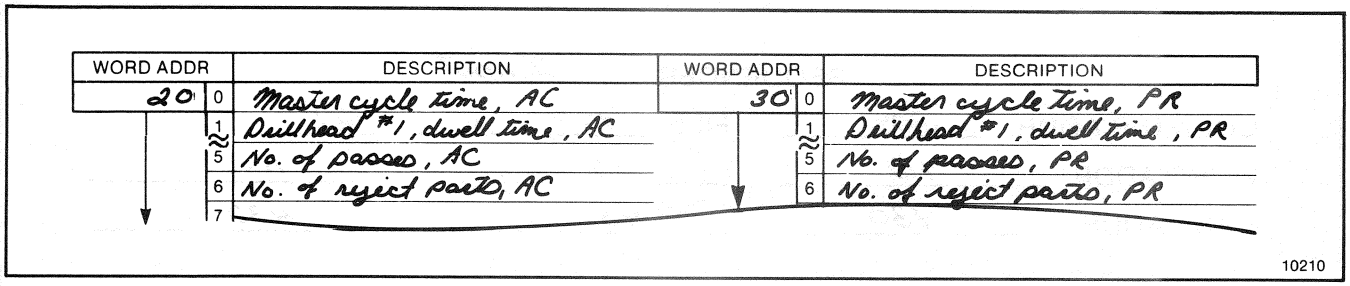


FIGURE 12-5— Example of Data Table Word Assignments

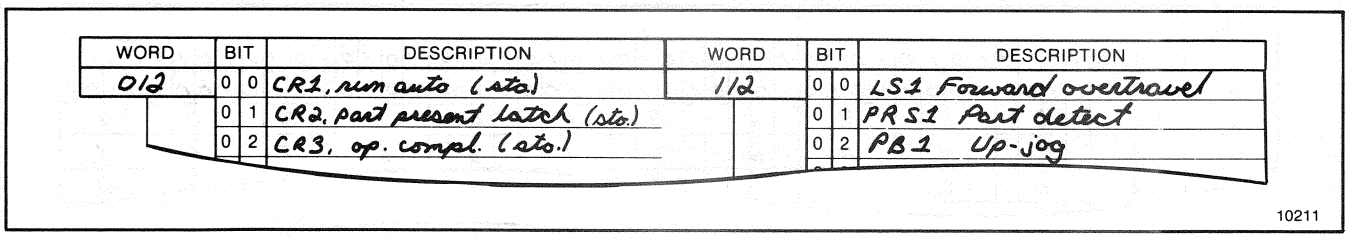


FIGURE 12-6— Example of Data Table Bit Assignments

the corresponding storage word 012_8 (complement of word 112). Word addresses "012" and "112" have been entered into corresponding word address boxes in the left- and right-hand columns, respectively. The 3or 4-digit word address is entered once for all 16 bits.

SEQUENCER TABLE BIT ASSIGNMENTS

This form can be used for any one of the three SEQUENCER instructions to log the data associated with each step.

The information added to the heading of the assignment sheet should be identical to the information displayed in the Data Monitor mode Heading and in the Ladder Diagram mode instruction block of the SEQUENCER instruction. The Mask row is used to log Mask data, if required. The remaining rows are for logging the data of each step. The data can be logged in binary or in hex. The step numbers should be written in the left-hand blank column. The From-To addresses at the bottom of the sheet are the starting and ending file addresses for each column of the Sequencer table.

For example, Figure 12-7 shows a completed portion of a Sequencer Table Bit Assignment Sheet for an 8-step 3-word-wide SEQUENCER INPUT instruction. The mask and steps 1 and 2 have been completed in binary. Steps 3-8 have been completed in hex for the sake of illustration. The starting and ending word addresses of each column of the Sequencer Table, 400_8 to 407_8, 410_8 to 417_8 and 420_8 to 427_8, respectively have been entered at the bottom of each column.

12.3.2 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 place-

SEQUENCER Input

COUNTER ADDR: 204 FILE 400 to 427

WORD ADDR: 112 113 114

MASK ADDR: 430 431 432

	WORD #1				WORD #2				WORD #3			
	17	10	07	00	17	10	07	00	17	10	07	00
MASK	1111 1111				1111 1111				1111 1111			0000
STEP												
1	1010 1010				1010 1010				1010 1010			
2	0010 0100				1010 1000				1000 1000			
3	A	B	F	8	E	A	4	C	A	3	D	9
4	3	C	4	D	2	8	1	2	B	5	F	4
	1	D	C	1	3	4					D	2
							1	F	C	1		
6	C	F	2	6	1 5	2 E	4 C	C H	D 1	4 2	6 7	1 F
7	4	7	D	2	3	9	B	1	A	B	C	6
8	6	3	F	8	2	3	F	E	1	C	2	D
FROM ADDR	400				410				420			
TO ADDR	407				417				427			

10212

FIGURE 12-7— Example of Sequencer Table Bit Assignments

ment 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.

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 1772-803, the Mini-PLC-2/15 Assembly and Installation Manual.

12.3.3 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. This address must be an "even-hundreds" address, meaning that the third digit from the right in the address must be 0, 2, 4, etc. Timer and counter instructions can be assigned Data Table addresses beginning at word address 030_8 through 1677_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 instructions are used, the addresses of these instructions must be entered into the timer/counter areas of the Data Table starting at word 030_8. These in-

structions also use two Data Table word addresses, one at an address 100_8 greater than the other. Each BLOCK TRANSFER instruction used will decrease by one (1) the number of available equivalent timers and/or counters from the maximum quantity of 488. For more information on Block Transfer, see Section 15.2.

12.3.4 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 provided 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. 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 these 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.

12.3.5 File Storage Considerations

File storage should immediately follow the last 128 word section in the Data Table reserved for timer/counter addresses. The Data Table can be expanded to a maximum of 1920 words (through word address 3577_8). Files can include their own unique addresses as well as duplicate pre-assigned addresses. Therefore, files should be carefully entered on the appropriate Data Table Assignment Sheets and/or Data Table Map. The following recommendations should be considered.

- Do not inadvertently overlap files or timer/counter Preset word addresses.
- Leave space between consecutive files for file growth. If addresses between files are used for bit/word storage, the addresses will have to be reassigned elsewhere if the need arises.

12.3.6 Sequencer Table Storage Considerations

Sequencer tables, as with files, should be entered on a Data Table Map and/or Assignment Sheets because they also may have their own unique addresses and/or may duplicate preassigned addresses. Moreover, sequencer tables can be 2, 3, or 4 words wide. This means that the number of steps in a sequencer table must be multiplied by the number of words wide (# columns) in order to obtain the total number of consecutive Data Table words required by a sequencer table. The recommendations for file storage also apply to the storage of sequencer tables.

12.4 SIZING THE DATA TABLE

The location of the last word in the Data Table will determine the size of the Data Table and how it will be configured in the Processor. If, after completing the User Program and logging all address on Data Table Assignment Sheets, the last address is at or less than 377_8, the number of equivalent timers and counters must be computed. Equivalent timers and counters includes internal storage words, i.e. bit/word storage, files, sequencer tables etc. The computation is made using the following formula:

ET = T + C + IS/2

where ET = # equivalent timers and counters

T = # timers
C = # counters
IS = # internal storage words.

If the last Data Table word is at address 400_8 or greater, count the total number of 128-word Data Table sections used (the first 128-word section includes the I/O Image tables). Partially used sections are counted as complete sections.

The final Data Table size must be configured in Processor memory before entering the User Program. This is done by entering the number of 128-word Data Table sections and, if necessary, the number of equivalent timers and counters. The Data Table configuration procedure is described in Section 13.1.

12.5 PROGRAM RECOMMENDATIONS

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

Note: A "Condition" instruction is defined as a non-block input instruction. Special considerations are given for MULTIPLY, DIVIDE and Block 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 12-8a), use a storage bit to make two rungs (figure 12-8b).
- Up to 7 parallel branches can be programmed in a rung provided all inputs are Condition instructions.

Recommendations for Block Instructions

- Up to 8 series Condition instructions can be used with a SEQUENCER INPUT instruction if the output is NOT a Block instruction; up to 4 series Condition instructions when the output IS a Block instruction.

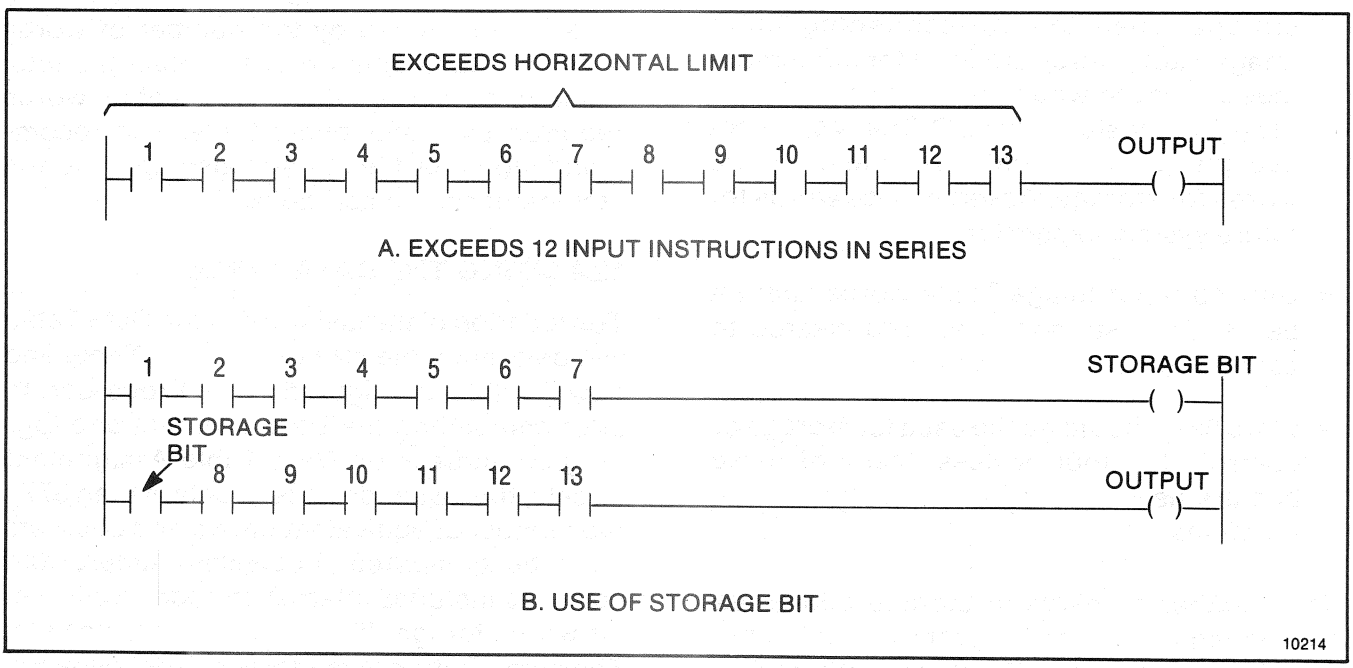


FIGURE 12-8— Storage Bit Example

- Up to 8 series Condition instructions can be used with a SEQUENCER INPUT instruction if the output is NOT a Block instruction; up to 4 series Condition instructions when the output IS a Block instruction.
- Up to 2 branches containing Condition instructions can be used in parallel with a SEQUENCER INPUT instruction.

12.6 CURRENT RECORD

A hard-copy printout of Data Table, User Program and messages should be made after the machine operation is working as desired. This is especially useful if the back-up EPROM memory is used. Before programming the EPROM from Processor memory, set all Data Table values to initial conditions. Record a total memory dump to obtain a hard-copy printout that includes the initial Data Table values. These are the values that will be loaded into Processor memory by EPROM to RAM transfer if a power loss should occur with no battery back up. After obtaining the hard-copy printout, the EPROM can be programmed as described in the Mini-PLC-2/15 Assembly and Installation Manual, Publication 1772-803 or the EPROM Programming Data Sheet, Publication 1770-915.

The hard-copy printout, Data Table Assignment Sheets and memory stored in EPROM constitute the current record of the machine operation. If subsequent changes are required, they should be included in all three records.

ALLEN-BRADLEY
Programmable Controller
DATA TABLE WORD MAP
(1024-WORD)
(Publication 5044 — February, 1982)

PAGE _____ OF _____
ADDRESS _____ TO _____

PROJECT NAME _____ PROCESSOR _____

DESIGNER _____ DATA TABLE SIZE _____

WORD ADDRESS	FROM ——— (32 WORDS) ———→ TO	WORD ADDRESS	REF.
□000		□037	
040		077	
100		137	
140		177	
200		237	
240		277	
300		337	
340		377	
400		437	
440		477	
500		537	
540		577	
600		637	
640		677	
700		737	
740		777	
□000		□037	
040		077	
100		137	
140		177	
200		237	
240		277	
300		337	
340		377	
400		437	
440		477	
500		537	
540		577	
600		637	
640		677	
700		737	
740		777	

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 _____ PROCESSOR _____

DESIGNER _____ 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
Programmable Controller
SEQUENCER TABLE BIT ASSIGNMENTS
(Publication 5048 — February, 1982)

PAGE _____ OF ____

PROJECT NAME _____ PROCESSOR _____

DESIGNER _____ DATA TABLE ADDR _____ TO _____

SEQUENCER _____

COUNTER ADDR: _____ FILE _____ to _____ SEQ LENGTH: _____

WORD ADDR: _____ _____ _____ _____

MASK ADDR: _____ _____ _____ _____

	WORD #1			WORD #2			WORD #3			WORD #4		
	17	10 07	00	17	10 07	00	17	10 07	00	17	10 07	00
MASK												
STEP												

ALLEN-BRADLEY
Cleveland, Ohio 44143

Bulletin 1771 I/O Chassis

CONNECTION DIAGRAM ADDRESSING

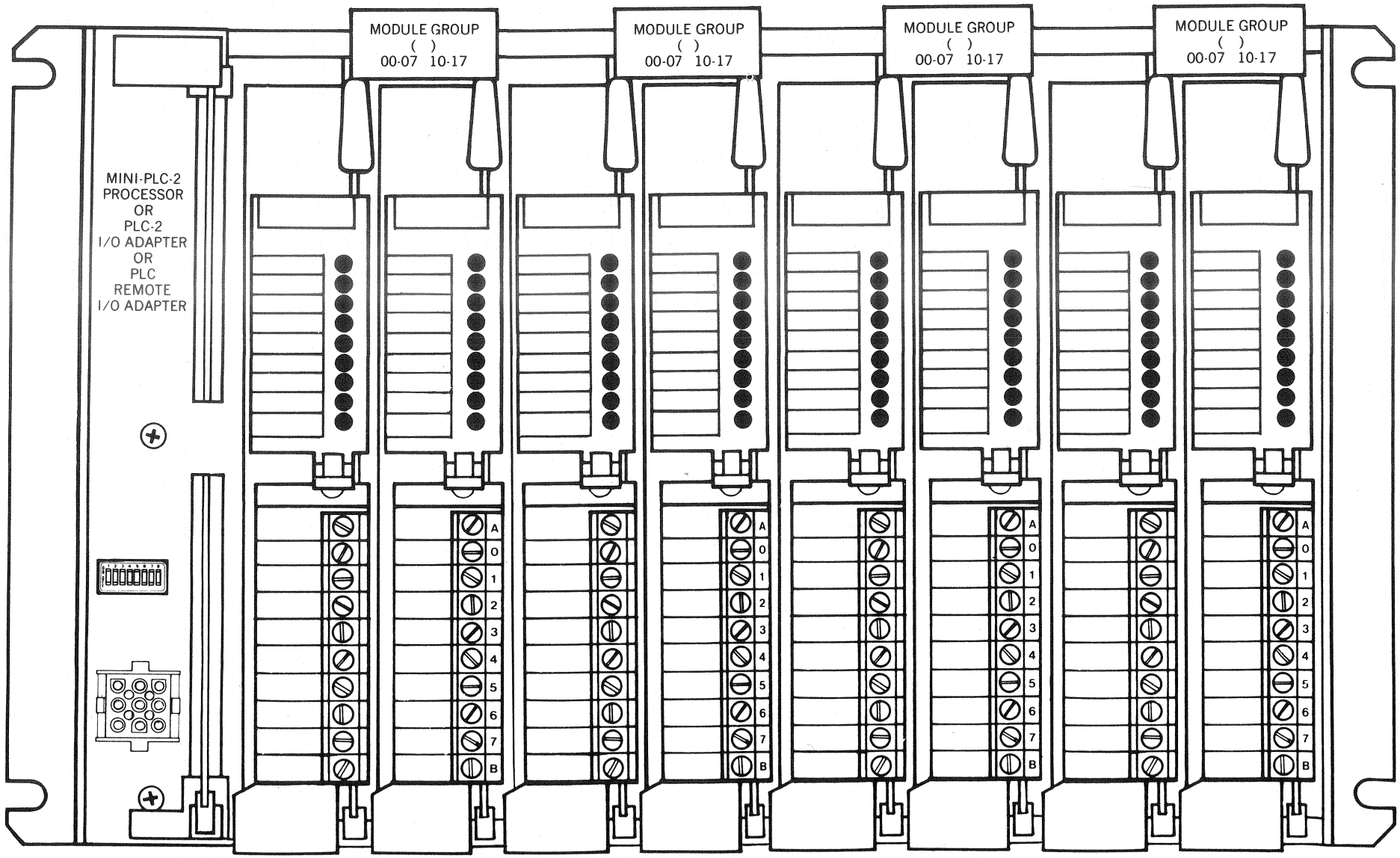
(Publication 5039 — September, 1980)

PAGE _____ OF _____

DATE _____

PROJECT NAME _____

DESIGNER _____



Section 13
OPERATING INSTRUCTIONS

13.0 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 13-1 through 13-6 at the end of this section. They include:

- Data Table Configuration (Table 13-1)
- Addressing
- Help Directories (Table 13-2)
- Search Functions (Table 13-3)
- Editing Functions (Table 13-4)
- On-line Programming
- Troubleshooting Aids (Table 13-5)
- Clear Memory Functions (Table 13-6)

13.1 DATA TABLE CONFIGURATION

After the size of the Data Table has been determined as described in Section 12.3 and 12.4, the SEARCH 50 function is used to configure the Data Table. The following display will appear when the [SEARCH] [5][0] keys are pressed in Program mode.

DATA TABLE CONFIGURATION

Number of 128-word Sections	1
Number of I/O Racks	1
Number of T/C (if applicable)	040
Data Table Size	128

Initially, the Data Table is factory configured to one 128-word section, 1 I/O Rack and 40 timers/counters. The number of 128-word Data Table sections and, if necessary, the number of timers/counters to be entered is prompted by a reverse-video cursor.

When the number of 128-word Data Table Sections is 1 or 2 (Data Table Size is 256 or less), enter this number. The number of equivalent timers/counters must be entered as well. This value can be computed using the formula in Section 12.4. The Processor automatically increases the Data Table size by two words for each equivalent timer or counter added by the user. The Data Table can be reduced in 2-word increments to a minimum of 48 words if zero timers or counters are selected.

When 3 or more 128-word Data Table sections are required, enter this number. The number of timers and counters is included in the number of Data Table sections entered. Therefore, the number of timers and counters need not be entered and the Industrial Terminal will display N/A (Not Applicable). 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.

If the size of the Data Table is increased, the User Program is automatically moved into higher word addresses. However, once the memory is full, further increase will not be permitted and the message "MEMORY FULL" will be displayed.

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

The instructions for configuring the Data Table are summarized in Table 13-1.

13.1.1 Memory Layout Display

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.

The instructions for displaying the Processor Memory layout are summarized in Table 13-1.

13.2 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. It can also be moved to any accessable position within an instruction block using the same keys. 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.

Bit addresses with 6 digits can be entered provided the [EXPAND ADDR] key is pressed. If a 5-digit bit address is displayed and a larger bit address is required, the [EXPAND ADDR] key can be pressed at any time provided the last digit has not been entered. If the last digit was entered, the instruction must be removed and the entire address must be re-entered.

Word addresses, unlike bit addresses, do not require the [EXPAND ADDR] key. Instead, always use leading zeros when necessary.

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

13.3 HELP DIRECTORIES

Help Directories have been developed as an aid in using the Industrial Terminal. They list the several functions or instructions common to a single multipurpose key such as the [SEARCH] or [FILE] key. A Master Help Directory is also available which lists the eight (8) Function and Instruction Directories for the Mini-PLC-2/15 and the key sequence to access them. The Master Help Directory is displayed by pressing the [HELP] key. The Help Directories are sumarized in Table 13-2. The [HELP] key can be pressed any time during a multi-key sequence. The remaining keys in the sequence can be pressed then without having to press [CANCEL COMMAND].

Note: If a particular Function or Instruction Directory or an item in a directory is not available with the Mini-PLC-2/15 Processor, the Industrial Terminal will display a "FUNCTION NOT AVAILABLE WITH THIS PROCESSOR" message if the key sequence is pressed.

13.4 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 Processor mode can be changed from the Industrial Terminal keyboard. The Search instructions are summarized in Table 13-3.

13.4.1 Search For First and Last Rung and User Program Boundaries

Program boundaries including the first or last rung can be located from any point in the User Program by using the [SEARCH][▲] or [SEARCH][▼] keys. The User Program could contain a TEMPORARY END boundary and/or a SUBROUTINE AREA boundary. It always contains an END of program boundary.

The cursor will go directly to the first rung from anywhere in User Program by pressing the [SEARCH][▲] keys.

When the [SEARCH][▼] keys are pressed, the display will go to the next boundary in the direction indicated. By pressing the [SEARCH][▼] keys again, a subsequent boundary will be displayed until the END of program statement is reached.

Boundaries will be displayed at the top of the screen with subsequent program rungs displayed beneath. (No rungs follow the END of program statement.)

13.4.2 Search For First or Last Instruction in a Rung

The first condition instruction of a rung can be accessed from anywhere in the rung by pressing [SEARCH][◄] when in PROGRAM mode. If not in PROGRAM mode, pressing [SEARCH] [◄] will move the cursor off the screen to the left. To bring it back on the screen, press the [►] key.

The output instruction can be accessed from anywhere in the rung by pressing [SEARCH] [►] in any mode.

13.4.3 Search For Incomplete Rung

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. Programming interruptions are further described at the end of paragraph 13.6.2.

13.4.4 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][Key Sequence of Instruction][Key Sequence of Address]. Enter leading zeros before the address if necessary. File and Sequencer instructions can be searched for by using the Counter address.

All addresses (excluding those associated with EXAMINE ON and EXAMINE OFF instructions and those contained within files) 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 Industrial Terminal will begin to search for the address and/or instruction from the cursor's position. It will look past the TEMPORARY END and SUBROUTINE AREA boundaries to the END statement. Then it will continue searching from the beginning of the program to the point where the search began.

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.

If the instruction is found in the Subroutine Area or past the TEMPORARY END instruction, the area in which it is found will be displayed in the lower right hand corner of the screen.

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

13.4.5 Single Rung Display

Upon power-up, a multiple rung display appears on the screen. The user has the option of viewing a single rung by pressing [SEARCH] [DISPLAY]. To return to the multiple rung display, press [SEARCH][DISPLAY] again.

13.4.6 Remote Mode Select

The Industrial Terminal keyboard can be used to change the Processor Mode of operation when the Processor keyswitch is in the RUN/PROG position.

The following key sequences can be used: [SEARCH][5][9][0] for Remote RUN/PROGRAM mode, [SEARCH][5][9][1] for Remote TEST mode, and [SEARCH][5][9][2] for Remote PROGRAM mode.

13.5 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 13-4.

Note: If the Memory Write protect is active, only Data Table values between word address 010_8 and 177_8 can be changed. If a protected User Program must be changed, the ultraviolet light erasable PROM (UV EPROM) must be removed. See the EPROM Programming Data Sheet, Publication No. 1770-915.

13.5.1 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] [Key Sequence of 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 or SUBROUTINE AREA 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] [←] Key sequence of Instruction] [Key sequence of Address]. The new instruction will be inserted before the cursor's present position.

Bit addresses of 6 digits can be entered provided the Data Table is expanded to a 4-digit word address and the [EXPAND ADDR] key is used.

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

13.5.2 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] [Key sequence of Instruction]. If the wrong instruction is pressed, an "INSTRUCTION DOES 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.

13.5.3 Inserting a Rung

A rung can be inserted anywhere within a program by pressing [INSERT][RUNG] and entering the instructions. The cursor must be positioned on the output instruction of the previous rung. The new rung will be inserted after the rung which contains the cursor. If the cursor is on the END or SUBROUTINE AREA statement, the [INSERT][RUNG] keys need not be used. Instructions in a new rung cannot be removed until the rung is complete. The rung can be entered just as in initial program entry.

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

13.5.4 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.

13.5.5 Changing Data of a Word or Block Instruction

The data of any word or block 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]. Once the first digit

has been entered the [►] key can be used. 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].

13.5.6 Changing the Address of a Word or Block Instruction

The address of a word or block instruction with data, excluding Arithmetic and PUT instructions, can be changed without removing and re-entering the instruction. To do this, position the cursor on the instruction and press [INSERT]. The cursor, although not displayed, will position itself on the first data digit. Enter that digit to display the cursor. Then, cursor back to the address digits using the [◄] key and change the address as needed. Use a leading zero if required.

13.5.7 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 the instruction key or key sequence of the desired instruction and the required address(es). This procedure also can be used when changing the address of an instruction that does not contain data.

Bit addresses of 6 digits can be entered provided the Data Table is expanded to a 4-digit word address and the [EXPAND ADDR] key is used.

13.5.8 On-line Data Change

The lower 12 bits of a word or word instruction excluding Arithmetic and PUT instructions, or certain data of a block instruction can be changed while the Processor is in the RUN/PROGRAM mode. This is done by positioning the cursor on the appropriate instruction and pressing [SEARCH][5][1]. The message "ON-LINE DATA CHANGE, ENTER DIGITS FOR: (Required Information)" will be displayed near the bottom of the screen. The new digits will be displayed in a Command Buffer 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.

Note: when the Memory Write Protect is activated by the EPROM back-up memory, On-Line data Change will not be allowed for addresses above 177_8. If attempted, the Industrial Terminal will display the error message MEMORY PROTECT ENABLED.

13.6 On-Line Programming

On-Line Programming allows changes to be made to the User Program during machine operation when the Processor is in the RUN/PROGRAM mode and Memory Write Protect is not active.

WARNING: The task of On-Line Programming should be asigned only to an experienced programmer who understands the nature of Allen-Bradley Programmable Controllers and the machinery being controlled. Proposed on-line changes should be checked and rechecked for accuracy; and all possible sequences of machine operation resulting from the change should be assessed in advance. Be absolutely certain that the change must be done on-line and that the change will solve the problem without introducing additional problems. Notify personnel in the machine area before changing machine operation on-line.

Maintaining accurate Data Table Assignment Sheets and using the Data Initialization key described in this section are important steps in minimizing the chances of error when making On-Line Programming changes.

13.6.1 Data Initialization Key

When programming many kinds of instructions such as the GET, LES, EQU, timers and counters, files and sequencers, two types of information must be entered. They are the instruction address and operating parameters. The data stored at the instruction address is divided into two sections: status bits (bits 14-17) and BCD values (bits 00-13). During program execution these bits are constantly changing to reflect current states and values of program instructions. Therefore, when programming on-line, a decision must be made whether to use the current data or enter new data.

The [DATA INIT] key is used when adding an instruction containing new data. It should NOT be used when adding an instruction that will use the data at a pre-assigned address.

The [DATA INIT] key performs two functions in the On-Line Programming mode:

a) It allows entry of BCD data values (stored at the instruction address).

b) It assures that the status bits are cleared to 0000.

The [DATA INIT] key should be used when programming a data instruction whose address is not currently being used in the program. If the [DATA INIT] key were not used, data at the new address (possibly remaining from previous programming) may interfere with proper machine operation when the new instruction is inserted into the program.

WARNING: When the address of a new instruction duplicates the address of other instructions in the program, the [DATA INIT] key should NOT be used without first assessing the consequences. Pressing the [DATA INIT] key will zero out the status bits stored at the existing instruction address. This may interfere with desired machine operation. Damage to equipment and/or injury to personnel could result.

Note: To determine whether an address has already been used in the program, the Search for Specific Address function (paragraph 13.4.4) can be used to locate user-entered addresses. It will not locate other addresses, such as those within files or sequencer tables. Therefore, the Data Table Assignment Sheets should be checked manually to determine whether an address has been used.

In summary, use the [DATA INIT] key when entering a data instruction with an unused address. Use it when it is necesary to enter new data and clear the status bits of an already used address. Be sure that the consequences are clearly understood.

The [DATA INIT] key should be pressed after the instruction key(s) and before the address is entered.

13.6.2 On-Line Programming Procedures

The changes to User Program that can be made in the On-Line Programming mode include the following:

- Insert an Instruction
- Remove an Insruction
- Insert a Rung
- Remove a Rung
- Change an Instruction or Instruction Address

The On-Line Programming mode is accessible from the Industrial Terminal by pressing the key sequence [SEARCH][5][2]. The Processor keyswitch must be in RUN/PROG position. The heading, "ON-LINE PROGRAMMING" will appear in the top right-hand corner of the screen highlighted in reverse video.

The procedure for programming on-line in RUN/PROGRAM mode is similar to the procedure for editing in PROGRAM mode with the exception that the following three keys have a special purpose in On-Line Programming:

[RECORD]

[CANCEL COMMAND]

[DATA INIT.]

The [RECORD] key is used to enter a change Into User Program. Once pressed, the changed program is active immediately.

The [CANCEL COMMAND] key can be used to abort any On-Line Programming operation prior to pressing the [RECORD] key. It restores the ladder diagram display and program logic to its original state prior to the On-Line Programming operation. It is also used to terminate the On-Line Programming Mode.

The [DATA INIT] key should be used as described in paragraph 13.6.1 to allow entry of data or instruction parameters and to set status bits to their proper initialization states.

TO INSERT AN INSTRUCTION

Instructions can be inserted into User Program using the key sequences described in this section.

The instruction being inserted will be highlighted in reverse video until the [RECORD] key is pressed.

WARNING: When the [RECORD] key is pressed, the instruction is entered into memory immediately. If the rung logic is TRUE, the output instruction will be enabled and will instantly energize the output device. Injury to personnel near the machine could result.

The procedure for inserting an instruction into an existing rung is as follows: (Refer to Editing, paragraph 13.5.1 if necessary.)

- Position the cursor on the preceding instruction.
- Press [INSERT][Key Sequence of Instruction].
- Use the [DATA INIT] key, if appropriate.
- Enter instruction parameters.
- Verify that the instruction is correctly entered.
- Press the [RECORD] key.

The Data Monitor mode of block instructions such as File or Sequencer cannot be entered until after the [RECORD] key is pressed. If file data is required, the rung containing the new instruction should be held FALSE until the data has been entered. This can be done by inserting a BRANCH END instruction at the beginning of the rung and then removing it after the data has been entered. The procedure is as follows for monitoring and/or entering data into block instructions:

- Press [DISPLAY][0] or [DISPLAY][1] for Data Monitor mode.
- Press [SEARCH][5][1] for On-Line Data Change.
- Enter file data if necessary.
- Press [CANCEL COMMAND] to terminate On-Line Data Change.
- Verify file data and/or data words.
- Press [CANCEL COMMAND] to terminate Data Monitor mode.

TO REMOVE AN INSTRUCTION

A condition instruction can be removed using the following procedure. (Refer to Editing, paragraph 13.5.2 if necessary).

- Position the cursor on the instruction to be removed.
- Press [REMOVE][Key Sequence of the Instruction].
- Press [RECORD].

WARNING: When the [RECORD] key is pressed, the instruction will be removed immediately. If the removal of the instruction causes the rung logic to become TRUE, the output instruction will be enabled and will instantly energize the output device. Injury to personnel near the machine could result.

Note: Bit values and the data of word instructions are not cleared. However, the Input Image Table bits are rewritten during the next I/O scan.

TO INSERT A RUNG

A rung can be inserted into an existing program in the following manner (Refer to Editing, paragraph 13.5.3 if necessary)

- Position the cursor on the output instruction of the preceding rung.
- Press [INSERT][RUNG].
- Enter the instructions one at a time using the [RECORD] key to enter each instruction.

The inserted rung becomes active only after the output instruction is entered.

CAUTION: If the rung logic is TRUE, the output instruction and output device will be enabled immediately. Before pressing the [RECORD] key for the output instruction, verify that each instruction has been entered with no errors.

TO REMOVE A RUNG

A completed rung can be removed using the following procedure. (Refer to Editing, paragraph 13.5.4 if necessary)

- Position the cursor on any instruction in the rung.
- Press [REMOVE][RUNG][RECORD].

CAUTION: The rung will be removed immediately. If the rung was used directly or indirectly to control an output, the consequences of removal in terms of machine operation should be assessed beforehand.

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

TO CHANGE AN INSTRUCTION OR ADDRESS

An instruction can be replaced or the address of an instruction can be changed using the following procedure: (Refer to Editing, paragraph 13.5.7 if necessary)

- Place the cursor on the instruction to be changed.
- Press the desired instruction key or key sequence of the instruction.
- Use the [DATA INIT] key if appropriate.
- Enter the instruction address(es) and parameters.
- Verify that the instruction is correctly entered.
- Press the [RECORD] key.

If the substituted instruction is a block instruction requiring the entry of file data, the rung containing the instruction should be held FALSE until the data has been entered using the Data Monitor mode. See TO INSERT AN INSTRUCTION, above.

WARNING: When the [RECORD] key is pressed, the substituted instruction is entered into memory immediately. If the rung is TRUE, the output instruction will be enabled and will instantly energize the output device. Injury to personnel near the machine could result.

Note: Bit values and the data of word instructions are not cleared when an instruction is replaced by another.

Note: If only the data of an instruction is to be changed, use the On-Line Data Change procedure described in paragraph 13.5.8.

TO CORRECT AN ERROR

An error can be corrected during On-Line Programming any time before the [RECORD] key is pressed using either of the following procedures:

a) Using the [◄] and [►] cursor control keys, cursor back to the digit containing the error and correct it.

b) Press the [CANCEL COMMAND] key. It restores the ladder diagram display and program logic to the origial instruction and address(es).

When inserting a rung, an error can be corrected before the output instruction is entered (before the [RECORD] key is pressed) by either of the following procedures:

a) Complete the rung by using an OUTPUT ENERGIZE instruction addressed to an UNUSED storage bit.

b) If the instruction being inserted is in error, press the [CANCEL COMMAND] key to abort.

When either of these are done, proceed as follows:

- Cursor to the instruction containing the error and correct it.
- Enter the desired output instruction.
- Verify that the inserted rung is correct.
- Press the [RECORD] key.

The rung will become active immediately.

PROGRAMMING INTERRUPTIONS

If communication between the Industrial Terminal and Processor is interrupted when programming On-Line in RUN/PROGRAM mode, a rung could be left incomplete (no output instruction). Upon initialization of the Industrial Terminal, if an incomplete rung is thought to exist, proceed as follows:

- Locate the incomplete rung using the key sequence [SHIFT][SEARCH].
- Place the cursor at the end of the rung.
- Complete the rung by changing the blank output to the desired output instruction using the procedure, Replacing an Instruction, paragraph 13.5.7.

13.6.3 General Rules

Once the EPROM back-up memory has been programmed and Memory Write Protect is active, On-Line Programming is NOT allowed. However, Data Table values between words 010_8 and 177_8 can be changed using the On-Line Data Change procedure. In addition, the following rules are always applicable when programming on-line in RUN/PROGRAM mode.

- As in PROGRAM mode, output instructions can be changed but cannot be removed unless the entire rung is removed.
- BLOCK TRANSFER READ and WRITE instructions, JUMP, JUMP TO SUBROUTINE, MCR and ZCL instructions cannot be inserted.
- The LABEL instruction cannot be inserted or removed directly, nor can the rung containing it be removed. However, the LABEL instruction can be changed to another instruction.

> **CAUTION:** When editing out a LABEL instruction, all JUMP and JUMP TO SUBROUTINE instructions with the same label number must be removed. If not, a Run-time error will occur when the Processor executes the JUMP TO SUBROUTINE instruction.

13.7 TROUBLESHOOTING AIDS

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

- Bit Manipulation and Monitor Functions
- FORCE ON and FORCE OFF Functions
- TEMPORARY END Instruction
- ERR Message Display

The Troubleshooting aids are summarized in Table 13-5.

13.7.1 Bit Manipulation and Monitor

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 13.7.2 below.

BIT MANIPULATION

Bit Manipulation can function when the Processor is in PROGRAM mode. When in TEST, or RUN/PROGRAM, 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 13.7.2 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 result.

13.7.2 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/PROGRAM 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.

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

Forcing functions can be applied 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. In either case, 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]

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]

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. In every mode except the PROGRAM mode, the forced status "ON" or "OFF" will be displayed below each forced instruction.

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.

If the Industrial Terminal or Processor is disconnected or loses AC power, or the [MODE SELECT] key is pressed, all Force functions are cleared.

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 near the machine could result.

13.7.3 Forced Address Display

A complete list of bit addresses that are forced ON or 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].

13.7.4 TEMPORARY END Instruction

The TEMPORARY END instruction 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.

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 Main Program to be kept active. Position the cursor on the output instruction. Press [INSERT][T.END]

b) Cursor to the first rung of the Main 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. Press [INSERT][RUNG] and then enter the new rung.

Attempting to use the TEMPORARY END instruction in any of the following ways will either be prevented by the Industrial Terminal or result in a Run-Time error.

- Using more than one TEMPORARY END instruction at a time.
- Using the instruction in the Subroutine Area.
- Inserting or removing the instruction on-line during On-Line Programming.
- Placing the instruction in the path of JUMP or JUMP TO SUBROUTINE instructions.

13.7.5 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 illegal opcode ERR message should not be confused with ERR messages caused when a 1770-T1 or T2 Industrial Terminal is connected to a Processor that was programmed using a 1770-T3 Industrial Terminal. (See Section 1.5.1 Industrial Terminal Compatability.) Those ERR messages do not contain the 4-digit hex value and do not cause a Processor fault.

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 13.5.7 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 13.5.3 and 13.5.4 Inserting and Removing a Rung. The cause of the problem should be identified and corrected in addition to correcting the ERR message.

13.8 CLEARING MEMORY

The option of clearing the Data Table, User Program and Messages is available with various CLEAR MEMORY functions. When Memory Write Protect is active, memory cannot be cleared except between and including address 010_8-177_8 in the Data Table. The Clearing Memory instructions are summarized in Table 13-6.

13.8.1 Data Table Clear

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.

13.8.2 User Program Clear

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 the first boundary: TEMPORARY END instruction, SUBROUTINE AREA or END statement. Neither the Data Table nor Messages are cleared.

13.8.3 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. None of the bits in the Data Table are cleared.

13.8.4 Total Memory Clear

The complete memory can be cleared by positioning 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 13-1 — Data Table Configuration

FUNCTION	MODE	KEY SEQUENCE	DESCRIPTION
Data Table Configuration	Program	[SEARCH] [5][0] [Numbers]	If the number of 128-word sections is 1 or 2, enter this number, and the number of timers/counters. If the number of 128-word sections is 3 or greater, enter only this number. The Industrial Terminal will calculate and display the Data Table size in decimal.
Processor Memory Layout	Any	[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.

TABLE 13-2 — Help Directories

FUNCTION	MODE	KEY SEQUENCE	DESCRIPTION
Help Directory	Any	[HELP]	Displays a list of the keys that are used with the [HELP] key to obtain further directories.
Control Function Directory	Any	[SEARCH] [HELP]	Provides a list of all control functions that use the [SEARCH] key.
Record Function Directory	Any	[RECORD] [HELP]	Provides a list of functions that use the [RECORD] key.
Clear Memory Directory	Program	[CLEAR MEMORY] [HELP]	Provides a list of all functions that use the [CLEAR MEMORY] key.
Data Monitor Directory	Any	[DISPLAY] [HELP]	Provides the choice of Data Monitor display accessed by the [DISPLAY] key.
File Instruction Directory	Any	[FILE][HELP]	Provides a list of all Instructions that use the [FILE] key
Sequencer Instruction Directory	Any	[SEQ][HELP]	Provides a list of all Instructions that use the [SEQ] key
Block Transfer Directory	Any	[BLOCK XFER] [HELP]	Provides a list of all Instructions that use the [BLOCK XFER] key
All Directories	Any	[CANCEL COMMAND]	To terminate

TABLE 13-3 — SEARCH Functions

FUNCTION	MODE	KEY SEQUENCE	DESCRIPTION
Locate first rung of program	Any	[SEARCH][▲]	Positions cursor on the first instruction of the program.
Locate last rung of program area	Any	[SEARCH][▼]	Positions cursor on the TEMPORARY END instruction, SUBROUTINE AREA boundary, or the END statement depending on the cursor's location. Press key sequence again to move to the next boundary.
Locate first instruction of current rung	Program	[SEARCH][◄]	Positions cursor on first instruction of the current rung.
Move cursor off screen	Test, Run or Run/Program	[SEARCH][◄]	Moves cursor off screen to left.
Locate output instruction of current rung	Any	[SEARCH][►]	Positions cursor on the output instruction of the current rung.
Locate rung without an Output Instruction	Any	[SHIFT] [SEARCH]	Locates any rung left incomplete due to an interruption in programming.
Locate specific instruction	Any	[SEARCH] [Instruction keys] [Address keys] 1	Locates instruction searched for. Press [SEARCH] to locate the next occurrence of instruction.
Locate specific word address	Any	[SEARCH][8] [Address keys] 1	Locates this address in the program (excluding -| |- and -|/|- instructions and addresses in files). Press [SEARCH] to locate the next occurrence of this address.
Single rung display	Any	[SEARCH] [DISPLAY]	Displays the first rung of a multiple rung display by itself. Press key sequence again to view multiple rungs.
Remote Mode Select: RUN/PROGRAM	Run/Program	[SEARCH] [5][9][0]	Places the Processor in RUN/PROGRAM mode.
Remote TEST		[SEARCH] [5][9][1]	Places the Processor in Remote TEST mode.
Remote PROGRAM		[SEARCH] [5][9][2]	Places the Processor in Remote PROGRAM mode.

1 Enter leading zeros when bit address exceeds 5 digits or word address exceeds 3 digits.

TABLE 13-4 — Editing Functions [1]

FUNCTION	MODE	KEY SEQUENCE	DESCRIPTION
Inserting a Condition Instruction	Program	[INSERT] [Instruction] [Address][2] or [INSERT][ ] [Instruction] [Address][2]	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	[REMOVE] [Instruction]	Position the cursor on the instruction to be removed and press the key sequence.
Inserting a rung	Program	[INSERT] [RUNG]	Position the cursor on any instruction in the preceding rung and press the key sequence. Enter instructions and complete the rung.
Removing a rung	Program	[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 or block instruction	Program	[INSERT] [Data]	Position the cursor on the word or block instruction whose data is to be changed. Press the key sequence.
Change the address of a word or block instruction	Program	[INSERT] [First Digit] [◄] [Address][2]	Position the cursor on a word or block instruction with data and press [INSERT]. Enter the first digit of the first data value of the instruction. Then use the [◄] and [►] key as needed to cursor up to the word address. Enter the appropriate digits of the word address.
Replace an instruction or Change address of an instruction without data	Program	[Instruction] [Address][2]	Position the cursor on the instruction to be replaced or whose address is to be changed. Press the desired instruction key (or key sequence)and the required (address(es).
On-Line Data Change	Run/Program	[SEARCH] [5][1] [Data]	Position the cursor on the word or block instruction whose data is to be changed. Press the key sequence. Cursor keys can be used.
		[INSERT]	Press [INSERT] to enter the new data into memory.
		[CANCEL COMMAND]	To terminate On-Line Data Change.
All editing functions	Program Run/Program	[CANCEL COMMAND]	Aborts the operation at the current cursor position.

[1] These functions can also be used during On-Line Programming. (Refer to Section 13.6)

[2] When bit address exceeds 5 digits, press the [EXPAND ADDR] key before entering address and enter a leading zero if necessary.

TABLE 13-5 — Troubleshooting Aids

FUNCTION	MODE	KEY SEQUENCE	DESCRIPTION
Bit Monitor	Any	[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 Test Run/Program	[SEARCH] [5][3]	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	Test or Run/Program	[FORCE ON] [INSERT]	Position the cursor on the Image Table bit or bit instruction to be forced ON and press the key sequence. The input bit or output device will be forced ON. [1]
Removing a FORCE ON	Test or Run/Program	[FORCE ON] [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	Test or Run/Program	[FORCE ON] [CLEAR MEMORY]	Position cursor anywhere in program and press key sequence.
FORCE OFF	Test or Run/Program	[FORCE OFF] [INSERT]	Position the cursor on the Image Table bit or bit instruction to be forced OFF and press the key sequence. The input bit or output device will be forced OFF.
Removing a FORCE OFF	Test or Run/Program	[FORCE OFF] [REMOVE]	Position the cursor on the Image Table bit or bit instruction whose force OFF is to be removed and press the key sequence.
Removing all FORCE OFF	Test or Run/Program	[FORCE OFF] [CLEAR MEMORY]	Position the cursor anywhere in program and press key sequence.
Forced Address Display	Any	[SEARCH] [FORCE ON] or [SEARCH] [FORCE OFF]	Displays a list of the bit addresses that are forced ON and forced OFF. The [SHIFT][▼] and [SHIFT][▲] keys can be used to display additional forces.
		[CANCEL COMMAND]	To terminate

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

TABLE 13-5 — Troubleshooting Aids (Cont.)

FUNCTION	MODE	KEY SEQUENCE	DESCRIPTION
Inserting a TEMPORARY END Instruction	Program	[INSERT] [◄][T.END] or [INSERT] [T.END]	Position the cursor on the instruction that will follow the TEMPORARY END instruction. The remaining rungs, although displayed and accessible, are not scanned. Position the cursor on the instruction that will precede the TEMPORARY END instruction. The remaining rungs, although displayed and accessible, are not scanned.
Removing a TEMPORARY END Instruction	Program	[REMOVE] [T.END]	Position cursor on TEMPORARY END instruction and press key sequence.

TABLE 13-6 — Clear Memory Functions

FUNCTION	MODE	KEY SEQUENCE	DESCRIPTION
Data Table Clear	Program	[CLEAR MEMORY] [7][7]	Displays a start address and an end address field.
		[Start Address] [End Address]	Start and end word addresses determine boundaries for Data Table clearing.
		[CLEAR MEMORY]	Clears the Data Table within and including addressed boundaries.
User Program Clear	Program	[CLEAR MEMORY] [8][8]	Position the cursor at the desired location in the program. Clears User Program from the position of the cursor to the first boundary: i.e. TEMPORARY END, SUBROUTINE AREA or END statement. Does not clear Data Table or Messages.
Partial Memory Clear	Program	[CLEAR MEMORY] [9][9]	Clears User Program and messages from position of the cursor. Does not clear Data Table.
Total Memory Clear	Program	[SEARCH][▲] [CLEAR MEMORY] [9][9]	Position the cursor on the first instruction of the program. Clears total memory (Data Table, User Program and Messages).
Note: When Memory Write Protect is active, memory cannot be cleared except for Data Table addresses $010\text{-}117_8$.			

Section 14
PERIPHERAL FUNCTIONS INCLUDING REPORT GENERATION

14.0 GENERAL

There are several functions that can be performed with a Mini-PLC-2/15 and the Industrial Terminal. Some require the use of a peripheral device connected to Channel C of the Industrial Terminal. The functions include:

- Contact Histogram
- Report Generation
- Cassette Recorder Dump and Load
- Data Cartridge Recorder Dump and Load
- Ladder Diagram
- Total Memory Dump

The contact histogram and report generation functions can be monitored by the Industrial Terminal without a peripheral device.

14.1 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 14-1).
- Pressing [RECORD] and a number from 2 to 8 on the Industrial Terminal (Table 14-2).

The settings of the table will override those of Table 14-1, if used.

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 when anyone of the following key sequences is pressed.

[RECORD], [SEARCH][4], [SEARCH][6] [SEARCH][7]

TABLE 14-1 — Switch Group Settings

SWITCH			BAUD
1	2	3	RATE
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 14-2 — 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

14.2 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 14-3:

- 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. Bit addresses larger than 5 digits do not require leading zeros or the EXPAND ADDR key.

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 14-1.

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

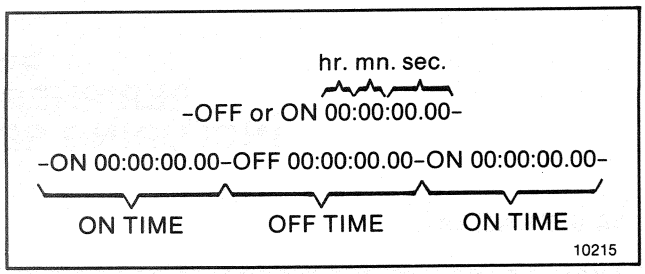


FIGURE 14-1— Contact Histogram Display

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

TABLE 14-3 — Contact Histogram Functions

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

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].

14.3 REPORT GENERATION

The Report Generation function of the Industrial Terminal, performed in the PLC-2 mode, can be used to generate up to 70 messages that contain ASCII and graphic characters and variable Data Table information. The messages are stored in the Processor's memory after the END of program 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 14-2):

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

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 14.3.2).

14.3.1 Report Generation Commands

The report generation function is enterd by pressing [RECORD] [DISPLAY] on the PLC-2 Family Keytop Overlay. A prompt, CHANGE TO ALPHANUMERIC KEYTOP OVERLAY, will be given. Or, set the baud rate. Then enter the message codes from the peripheral device. There are 5 report generation commands used to store, print, report and delete messages

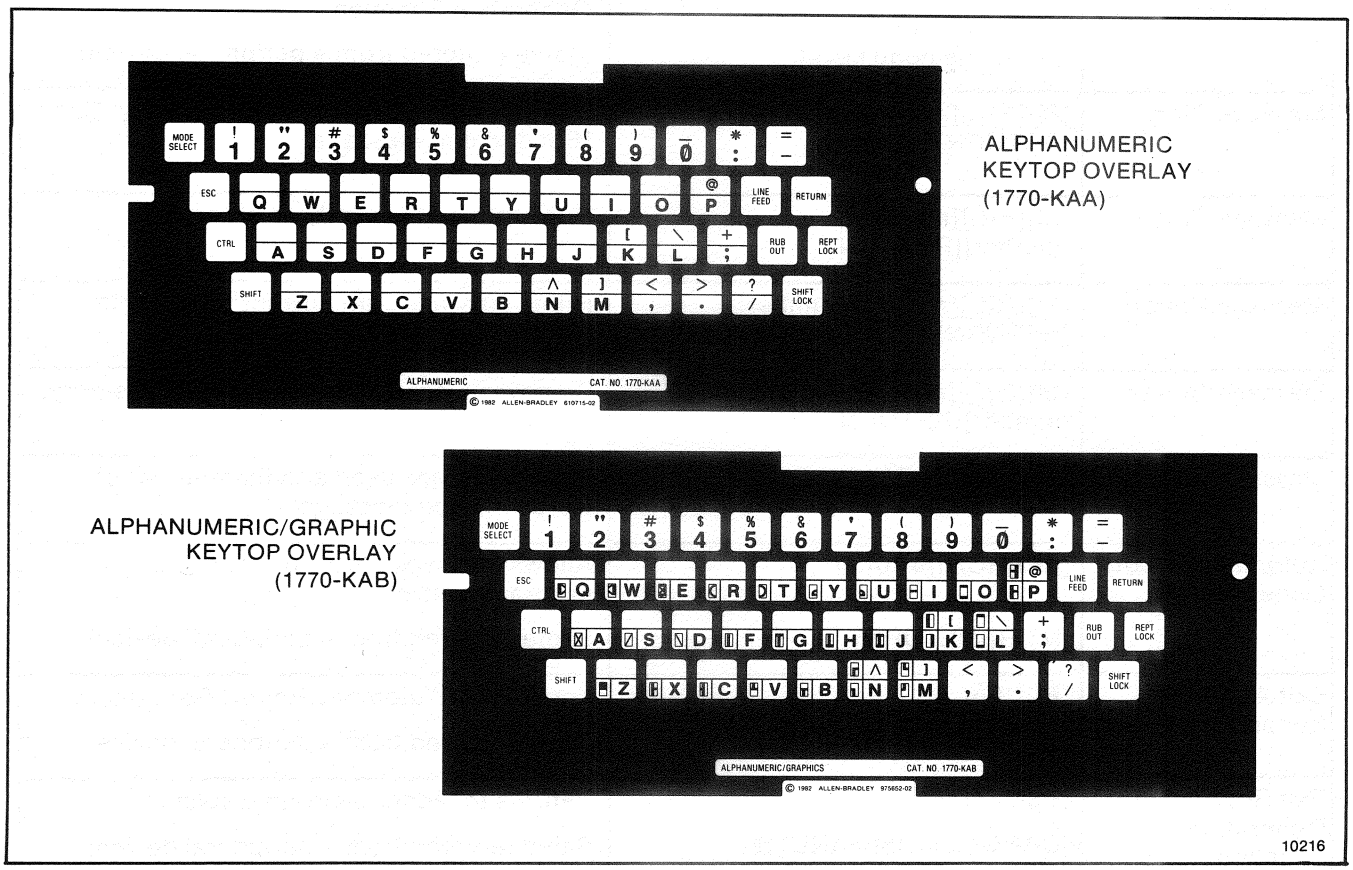


FIGURE 14-2— Alphanumeric Keytop Overlays

and to display an index of existing messages. These are summarized in Table 14-4.

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, 010-017, 110-117, 210-217, 310-317, 410-417, 510-517, 610-617 and 710-717.

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 14-5. The desired delimiter is entered before and after the bit, byte or word address. The

TABLE 14-4 — Report Generation Commands

COMMAND	KEY SEQUENCE	DESCRIPTION
Enter Report Generation Function	[RECORD][DISPLAY] or Set baud rate [message code keys]	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 from a peripheral 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.

TABLE 14-5 — Address Delimiters

DELIMITER FORMAT	EXPLANATION	MESSAGE REPORT FORMAT
XXX	Enter 3-digit word address between delimiters.	Displays BCD value at assigned word address.
XXX1 *XXX0*	Enter 3-digit word address and a "1" for upper byte or a "0" for lower byte between delimiters.	Displays the octal value at assigned byte address.
XXXXX	Enter 5-digit bit address between delimiters.	Displays the ON or OFF status of the assigned bit address.
#XXX#	Enter 3 or 4-digit word address between delimiters.	Displays BCD value at assigned word address.
!XXX!	Enter 3 or 4-digit word address between delimiters.	Displays the 4-digit hex value at address.
&XXX1& &XXX0&	Enter 3 or 4-digit word address and a "1" for upper byte or a "0" for lower byte between delimiters.	Displays the octal value of the byte at assigned address.
^XXXXX^	Enter 5 or 6-digit bit address between delimiters.	Displays the ON or OFF status of the assigned bit address.

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*. The asterisk delimiters, *, should only be used if the Data Table size is less than 512 words (not exceeding address 777_8).

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.

Control Codes and Special Commands

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 14-6 and 14-7). For example, graphic

TABLE 14-6 — 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. Move the cursor one position to the right.

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 14-8). These codes, although not displayed, can be interpreted 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 14-3.

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

TABLE 14-7 — 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]

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

TABLE 14-8 — ASCII Control Codes

CONTROL CODE [1]	DISPLAY [2]	ASCII 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 /	$^{U}_{S}$	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] Invalid key in Report Generation mode.

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.

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 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#.

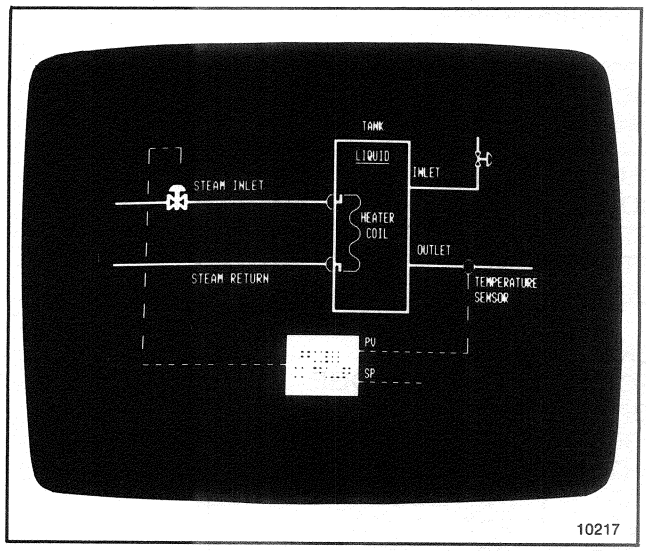


FIGURE 14-3— Example Graphic/Alphanumeric Message

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 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].

14.3.2 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, RUN, or RUN/PROGRAM modes by pressing [SEARCH] [4] [0] or by pressing [M] [R] [RETURN]. It can also be activated automatically upon initialization of the Industrial Terminal by setting parity switches 4 and 5 UP on the Industrial Terminal's main logic board (Figure 14-4).

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. All other messages use a user-defined file of message request bits for control. These two catagories will be discussed separately below.

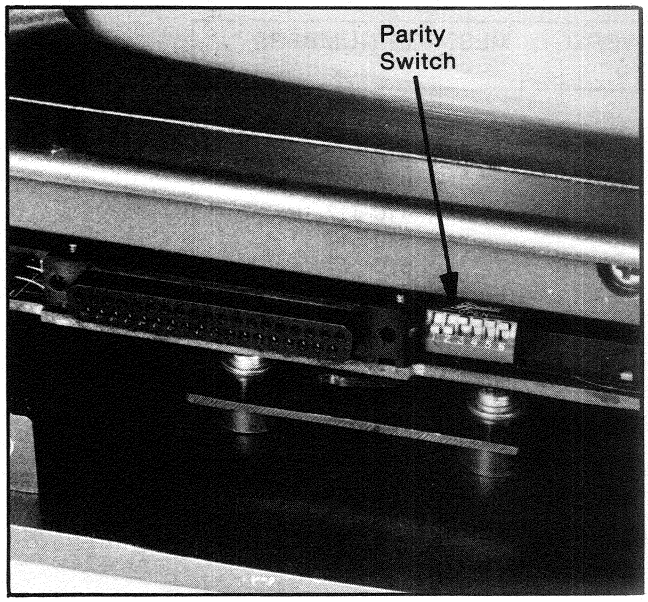


FIGURE 14-4— Parity Switch Location

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.

ADDITIONAL MESSAGES

Bits from eight consecutive user-selected words are used to control the 64 additional messages (1770-T3 Series A Rev B or later).

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 (Y DIGITS REQUIRED): will be printed. (Y is the number of digits, 3 or 4, of a word address for the selected Data Table size. The beginning word address of the message control word file must be entered. The Industrial Terminal will calculate and display the ending address. The message control word file can be located anywhere in the Data Table except Processor Work areas and Input Image Table. If Memory Write Protect is active, the message control word file must be placed in the area of Data Table which can be changed (010_8-177_8). Once the start address is chosen, the Industrial Terminal will also display a table (Table 14-9) which shows the message numbers associated with each message control word.

TABLE 14-9 — Example Message Control Word-Message Number Relationship

CONTROL WORDS [1]	MESSAGE NUMBERS
201	010-017
202	110-117
203	210-217
204	310-317
205	410-417
206	510-517
207	610-617
210	710-717

[1] NOTE: This table assumes user selected message control words begin at 201_8.

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 14-5.

CONTROL WORD NUMBER	CONTROL WORD ADDRESS	MESSAGE NUMBERS
0	201	010-017
1	202	110-117
2	203	210-217
3	204	310-317
4	205	410-417
5	206	510-517
6	207	610-617
7	210	710-717

THE CONTROL WORD ADDRESSES ARE USER SELECTED.

MESSAGE NUMBER 3XX HAS A MESSAGE REQUEST BIT AT ADDRESS 204/XX. MESSAGE REQUEST BIT 204/XX, WHEN ENABLED, WILL ACTIVATE MESSAGE NUMBER 3XX WHERE XX ARE BIT NUMBERS $10\text{-}17_8$.

10218

FIGURE 14-5— Bit Address-Message Number Relationship

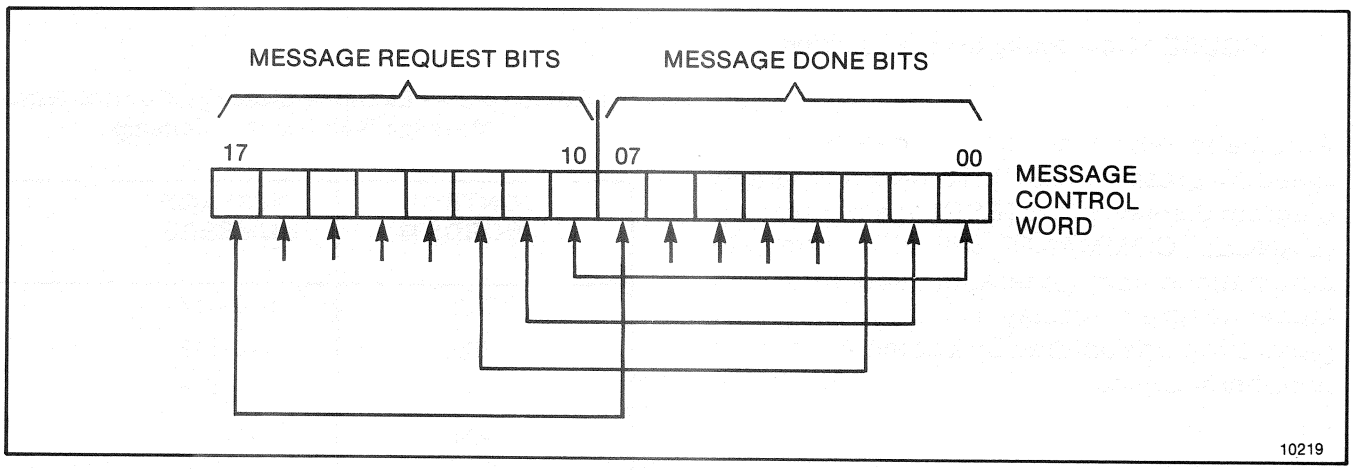


FIGURE 14-6— Message Request Bit-Done Bit Relationship

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 14-6 shows this relationship. For example, if 404/15 is the Request bit for a message, the Done bit is located at 404/05, 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 14-9. 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.

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 14-7 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.

14.4 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.

14.4.1 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 PLC-2 Family 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.

ABCD Program Words
EFGH Data Table Words

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].

14.4.2 Loading Memory from Cassette Tape

Loading the Processor memory from cassette tape can be done only in PROGRAM mode only when the Memory Write Protect is NOT active. 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 13.1 Data Table Configuration. 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 Family 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 com-

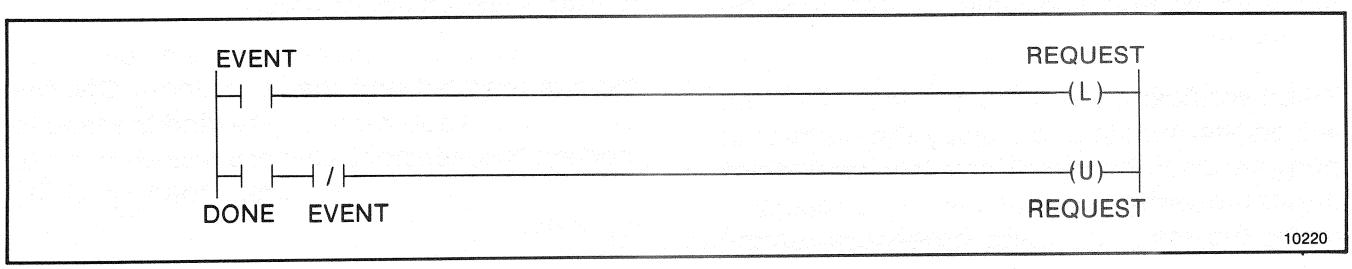


FIGURE 14-7— Example Program to Request a Message

plete, the Processor memory content will be verified automatically with the content on tape unless the cassette function is terminated by pressing [CANCEL COMMAND].

14.4.3 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 14.4.5 unless the cassette function is terminated by pressing [CANCEL COMMAND].

14.4.4 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 Family 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 14.4.5.

14.4.5 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 Family 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 procedures described in Section 13.5.

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

14.5 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 keyboard of the 1770-T1, -T2 or -T3 Industrial Terminal. 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 Key-

board is discussed in the following paragraphs. For operation in the same manner as a Digital Cassette Recorder, refer to paragraph 14.4.

14.5.1 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, this command is initiated by pressing [RECORD] [SHIFT] [B].

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

ABCD Program Words
EFGH 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 1 or more discprepencies are found, the entire recording operation should be repeated.

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

14.5.2 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 13.1 Data Table Configuration. 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 14.4.5, 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].

14.5.3 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 14.5.1, or the Processor Memory can be corrected using the procedure in paragraph 14.4.5.

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

14.6 LADDER DIAGRAM DUMP

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

This command is accessed by pressing the keys [SEARCH][4][4] on the PLC-2 Family 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 Family Overlay.

14.7 TOTAL MEMORY DUMP

This 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.

This command is accessed by pressing the keys [SEARCH][4][5] on the PLC-2 Family Overlay and will print out the complete memory, regardless of cursor position.

The Data Table will be printed in hexadecimal. The bit pattern for each Data Table word will be as shown in Figure 14-8. 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 $ECCB_{16}$ and $024C_{16}$ are stored in word addresses 0020_8 and 0025_8, respectively. For more information on the hexadecimal numbering system, refer to Section 17, 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.

When the printout is complete, this command is automatically terminated. Or, it can be terminated prior to completion by pressing [ESC] on the peripheral printer or [CANCEL COMMAND] on the PLC-2 Family Overlay.

	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
"	"	"	"	"	"	"	"	"
"	"	"	"	"	"	"	"	"

10221

FIGURE 14-8— Data Table Printout in Hexadecimal

Section 15
SPECIAL PROGRAMMING TECHNIQUES

15.0 GENERAL

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

- Scan Counter
- Block Transfer
- One-Shot
- Manual Restart

15.1 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 15-1. 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.

15.2 BLOCK TRANSFER

Block Transfer can be performed with the Mini-PLC-2/15 and any Bulletin 1771 I/O module with Block Transfer capability.

15.2.1 INTRODUCTION

Block Transfer is a combination of an instruction and support rungs 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

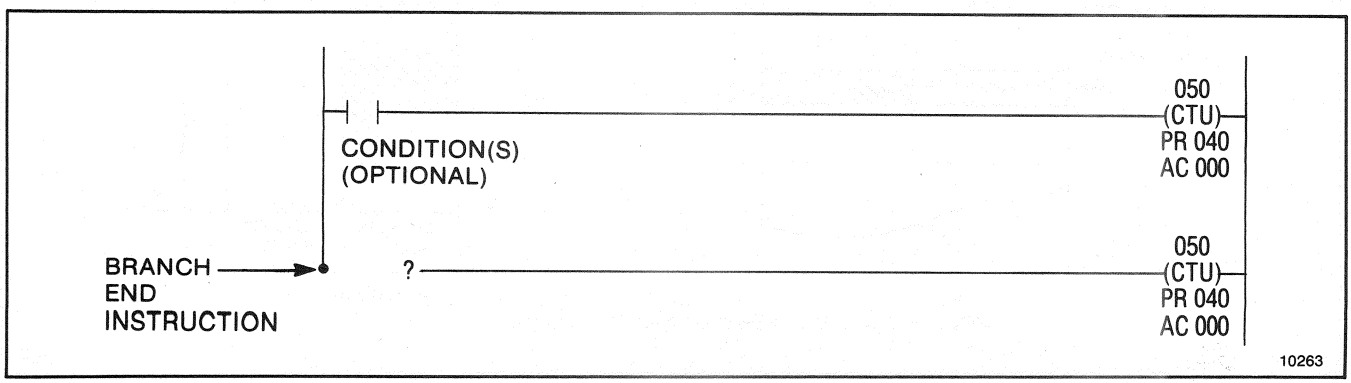


FIGURE 15-1 — Scan Counter

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

15.2.2 Basic Operation

The Processor uses two (2) 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. Dual-slot modules address slot 0. Figure 15-2.

The Block Transfer Read or Write operation is initiated in the program scan and completed in the I/O scan as follows: See Figure 15-3.

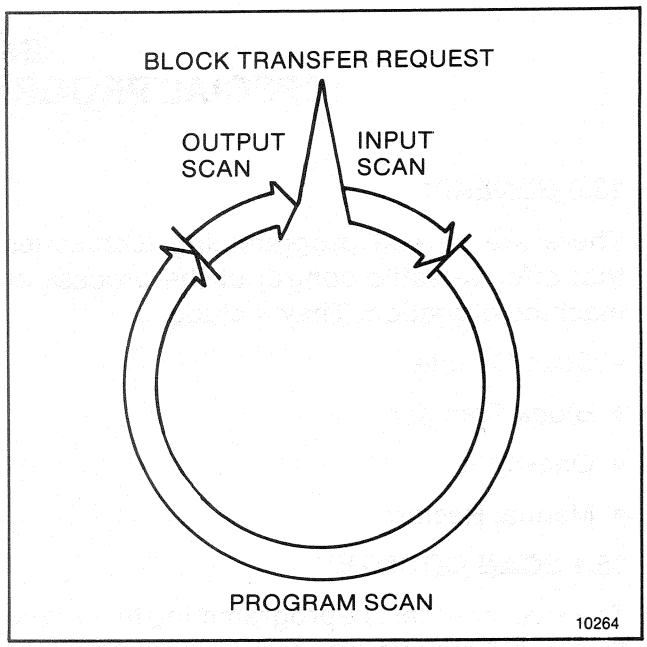


FIGURE 15-3 — Block Transfer Request

PROGRAM SCAN — When the rung goes TRUE, the instruction is enabled. The number of words to be transferred and the Read or Write bit that controls the direction of transfer

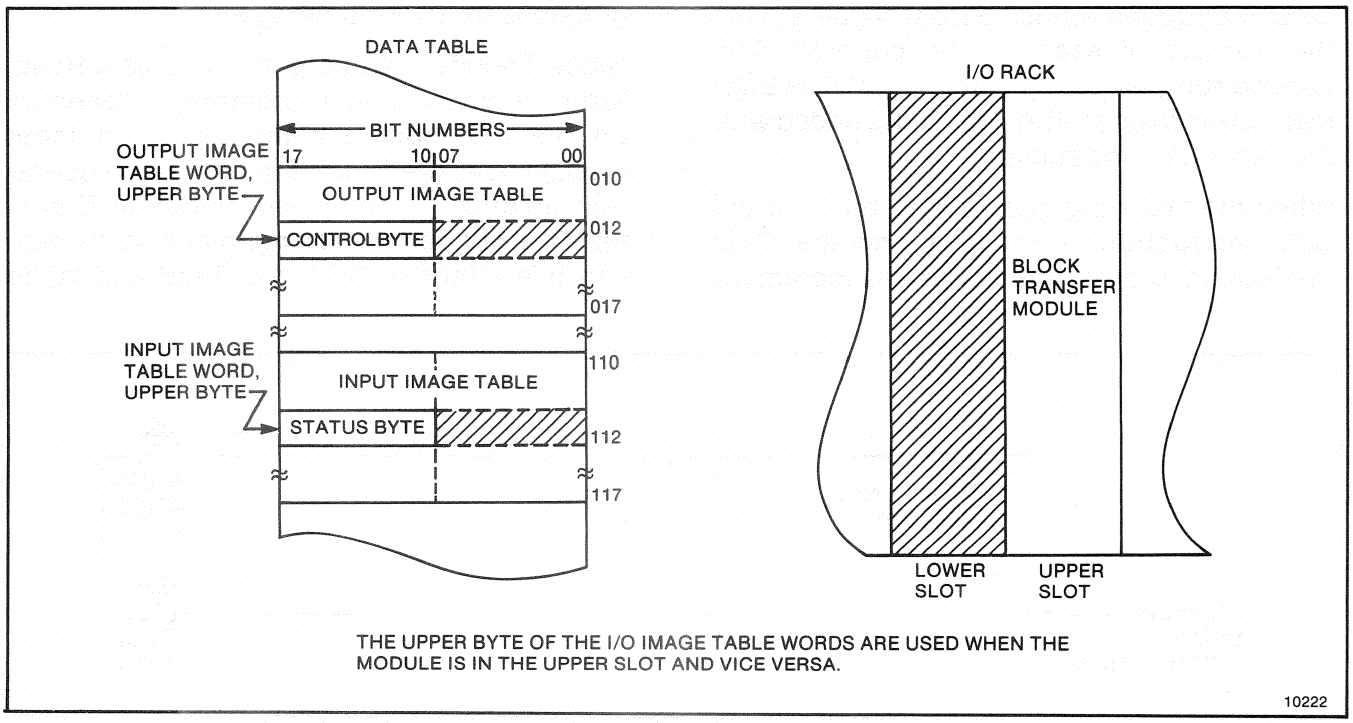


FIGURE 15-2 — Module Position-Image Table Byte Relationship

are set by a bit pattern in the Output Image Table byte.

I/O SCAN — The Processor requests a transfer by sending the Output Image Table byte data to the Block Transfer module during the scan of the Output Image Table. The module signals that it is ready to transfer. The Processor then interrupts the I/O scan and scans the timer/counter Accumulated area of the Data Table, looking for the address of the module that is ready to transfer. The Module Address is stored in BCD at a word address in the same manner as an Accumulated value of a timer is stored. The Module Address was entered by the programmer when entering the block instruction parameters. (The word address at which the Module Address is stored is called the Data Address of the instruction.)

Once the Module Address is found, the Processor locates the address of the file to which (or from which) the data will be transferred. The File Address is stored in BCD at an address 100_8 above the address containing the Module Address. This is done in the same manner that the Processor locates the Preset value of a timer in a word address 100_8 above the Accumulated value address. The analogy between Block Transfer and Timer/Counter data and addresses is shown in Table 15-1.

After locating the File Address in the timer/counter area of the Data Table, the Processor then duplicates and transfers the file data consecutively one word at a time until complete, starting at the selected File Address.

At the completion of the transfer, a Done bit for the Read or Write operation is set in the Input Image Table byte as a signal that a valid transfer has been completed.

15.2.3 BLOCK TRANSFER Instructions

The format of a BLOCK TRANSFER READ and a BLOCK TRANSFER WRITE instruction with default values is shown in Figure 15-4.

DATA ADDRESS AND MODULE ADDRESS

The Data Address is used to store the Module Address of the Block Transfer module. The Data Address must be assigned the first address in the timer/counter Accumulated area of the Data Table starting at word address 030_8. When more than one Block Transfer module is used, consecutive Data Addresses must be assigned ahead of address for Timer and Counter instructions.

The Module Address is stored in BCD by R=Rack, G=Module Group and S=Slot number. When Block Transfer is performed, the Processor searches the timer/counter Accumulated area of the Data Table for a match of the Module Address.

BLOCK LENGTH

The Block Length is the number of words that the module will transfer. It depends on the type of module and the number of channels connected to it. The number of words requested by the instruction (Block Length value) must be a valid number for the module: i.e. from 1 up to the maximum for the module. The Block Length can also be set at the default value of the module, useful when programming Bidirectional Block Tranfers. See paragraph 15.2.10.

The Block Length heading of the instruction will accept any value from 00-63 whether or not valid for a particular module. A value of 00 is entered for the default value and/or for a Block Length of 64.

The Block Length is stored in binary in the byte corresponding to the module's address in the Output Image Table.

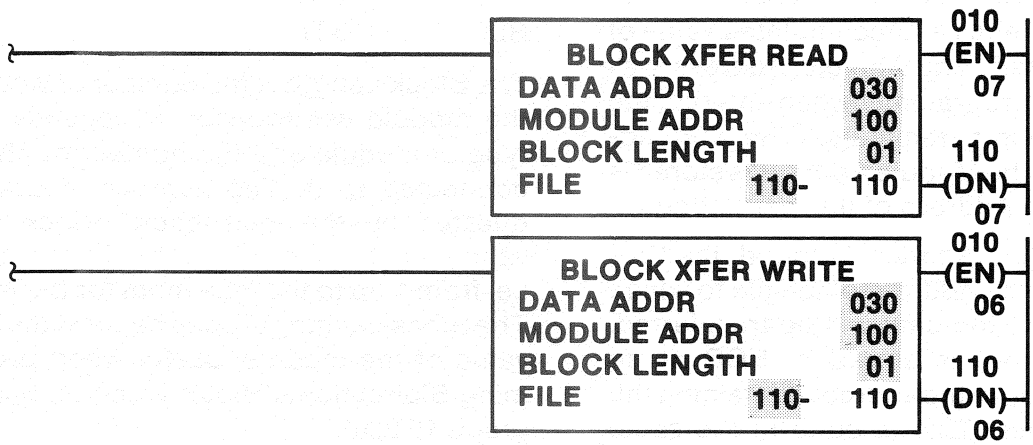
FILE ADDRESS

The File Address is the first word of the file to which (or from which) the transfer will be made. The File Address is stored 100_8 words above the Data Address of the instruction. When the File Address is entered into the instruction block, the Industrial Terminal computes and displays the ending address based on the Block Length.

When reserving an area for a Block Transfer file, an appropriate address must be selected to ensure that Block Transfer data will not write over assigned timer/counter Accumulated or Preset values. The File Address cannot exceed address 3577_8.

TABLE 15-1 — Timer/Counter — Block Transfer Analogy

TIMER/COUNTER		BLOCK TRANSFER
Address of Accumulated Value	=	Data Address of the instruction
Accumulated Value in BCD	=	Module Address in BCD
Address of Preset value	=	100_8 above Data Address
Preset Value in BCD	=	File Address in BCD



NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACE BY USER—ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INTIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

DATA ADDRESS : FIRST POSSIBLE ADDRESS IN ACCUMULATED VALUE AREA OF DATA TABLE.

MODULE ADDRESS : RGS FOR R = RACK, G = MODULE GROUP, S = SLOT NUMBER.

BLOCK LENGTH : NUMBER OF WORDS TO BE TRANSFERRED. (00 CAN BE ENTERED FOR DEFAULT VALUE OR FOR 64 WORDS)

FILE : ADDRESS OF FIRST WORD OF THE FILE, 100_8 ABOVE THE DATA ADDRESS.

ENABLE BIT -(EN)- : AUTOMATICALLY ENTERED FROM THE MODULE ADDRESS. SET ON WHEN RUNG CONTAINING THE INSTRUCTION IS TRUE.

DONE BIT -(DN)- : AUTOMACTICALLY ENTERED FROM THE MODULE ADDRESS. REMAINS ON FOR 1 SCAN FOLLOWING SUCCESSFUL TRANSFER.

10224

FIGURE 15-4— Block Transfer Format

ENABLE AND DONE BITS

The Read and Write bits are the Enable bits for Block Transfer modules. Either one (or both for a Bidirectional transfer) is set ON in the program scan when the rung containing the Block Transfer instruction is TRUE.

The Done bit is set ON in the I/O scan that the words are tranferred, provided that the transfer was initiated and successfully completed. The Done bit remains ON for only one scan.

Block Transfer will be requested in each program scan that the Read and/or Write bits remain ON. The Read and/or Write bits are turned OFF when the rung containing the instruction goes FALSE.

15.2.4 BLOCK TRANSFER READ and WRITE Instruction Characteristics

- Output Instruction
- The Block Length depends on the kind of module.
- The request is made in the program scan.
- The I/O scan is interrupted for the tranfer.
- The entire file is transferred in 1 scan.
- The Done bit remains ON for 1 scan after a valid transfer.
- Uses 2 words of User Program.
- Key sequence [BLOCK XFER][0] for WRITE and [BLOCK XFER][1] for READ.

15.2.5 Causes of Run-Time Errors

Misuse and/or inadvertent changes of instruction data can cause the Processor to fault due to Run-Time Errors when:

- The Module Address is given a non-existant I/O Rack number.
- A Read transfer overruns the file into a Processor Work Area or into User Program by an inadvertent change to the Block Length code.

15.2.6 Programming BLOCK TRANSFER READ and WRITE Instructions

To program a BLOCK TRANSFER READ instruction, press keys [BLOCK XFER][1] and enter the instruction parameters. To program a BLOCK TRANSFER WRITE instruction, press [BLOCK XFER][0] and enter the instruction parameters.

An example rung containing a BLOCK TRANSFER READ instruction and the Data Table areas used by the instruction are shown in Figure 15-5. The following parameters have been entered into the instruction:

Data Address	030
Module Address	121
Block Length	08
File	060

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 for the instruction are 012/17 and 112/17, respectively.

During the Program scan when input switch 113/02 is closed, the instruction 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_8 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 File Address, 060_8, in word address 130_8.

The Processor then transfers the data from the module into the 8-word file beginning at word address 060_8 through 067_8. At the completion of the transfer. Done bit 112/17 is set to 1. The Processor then completes the I/O scan.

15.2.7 Multiple Reads of Different Block Lengths from One Module

Under certain conditions, it may be desirable to transfer part of a file rather than the entire file. For example, a Processor could be programmed to read the first two or three channels of an analog input module periodically but read all channels less frequently. To do this, two or more BLOCK TRANSFER READ instructions would be used: one for each desired Block Length starting at the same first

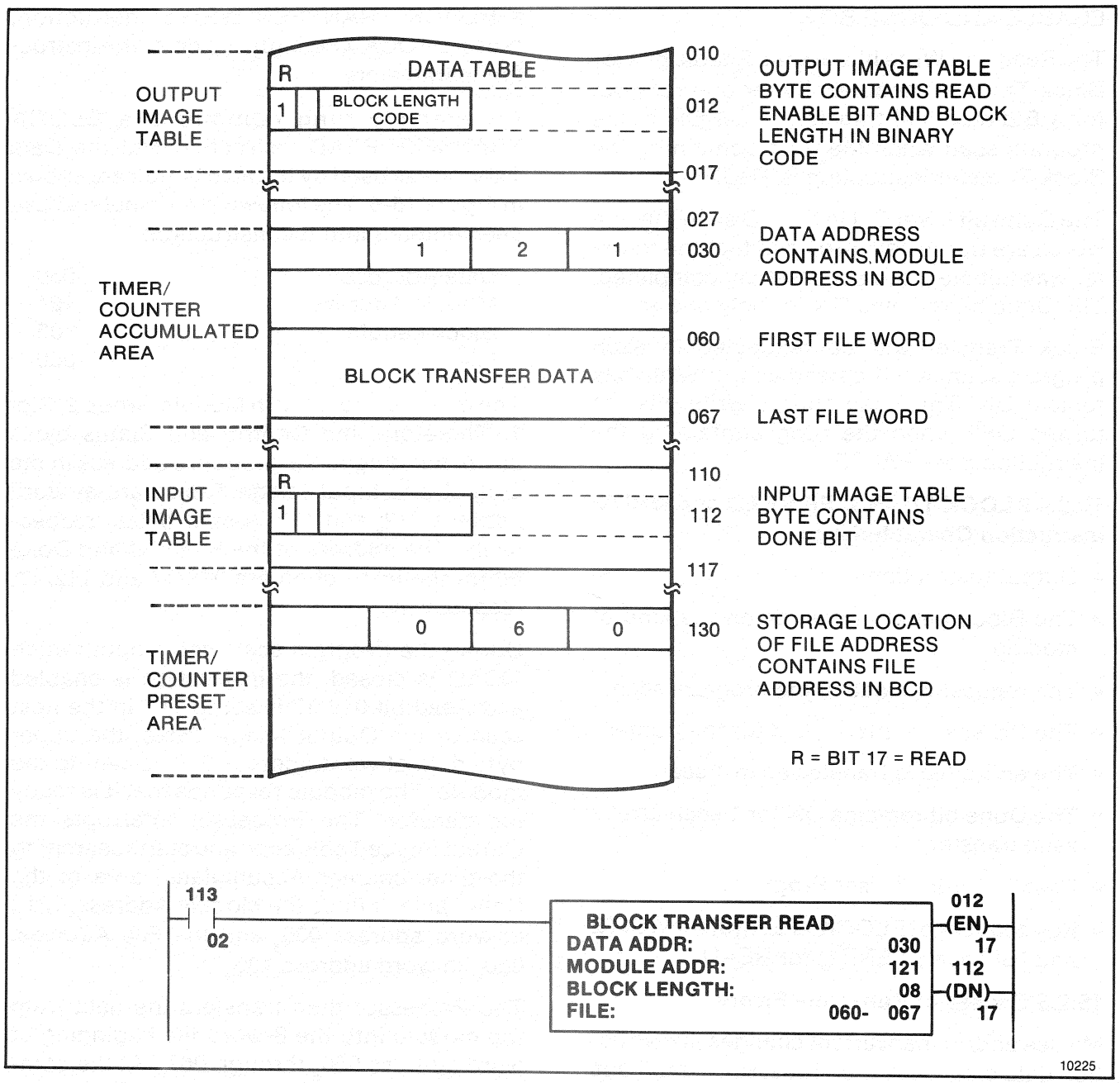


FIGURE 15-5 — Data Table Locations for a BLOCK TRANSFER READ

word. The READ instructions would have the same Module Address, Data Address and File Address. The size of the file would equal the larger Block Length.

When two or more Block Transfer instructions have a common Module Address, careful programming is required to compensate for the following possible situations:

During any program scan, data in the Output Image Table byte can be changed alternately by each successive Block Transfer instruction having a common Module Address. The Enable bit can be turned ON or OFF alternately according to the TRUE or FLASE condition of the rungs containing these instructions. The ON or OFF status of the last rung will govern whether the transfer will occur.

Secondly, the Block Length can be changed alternately in accordance with the Block Lengths of the enabled instructions. The

Block Length of the last enabled Block Transfer instruction having a common Module Address will govern the number of words transferred.

Refer to the User's Manual for additional information on the Block Transfer Module of interest.

The programming example shown in Figure 15-6 illustrates how multiple Reads of different Block Lengths from one module can be programmed. When any one of the input switches is closed, the rung is enabled and the Block Length is established. The last rung enables the Block Transfer instruction regardless of the previous changes in status of the Enable bit. The EXAMINE OFF instructions prevent more than one of the Block Transfer instructions from being energized in the same scan.

Note: The same discussion applies when programming multiple Writes of different Block Lengths to one module.

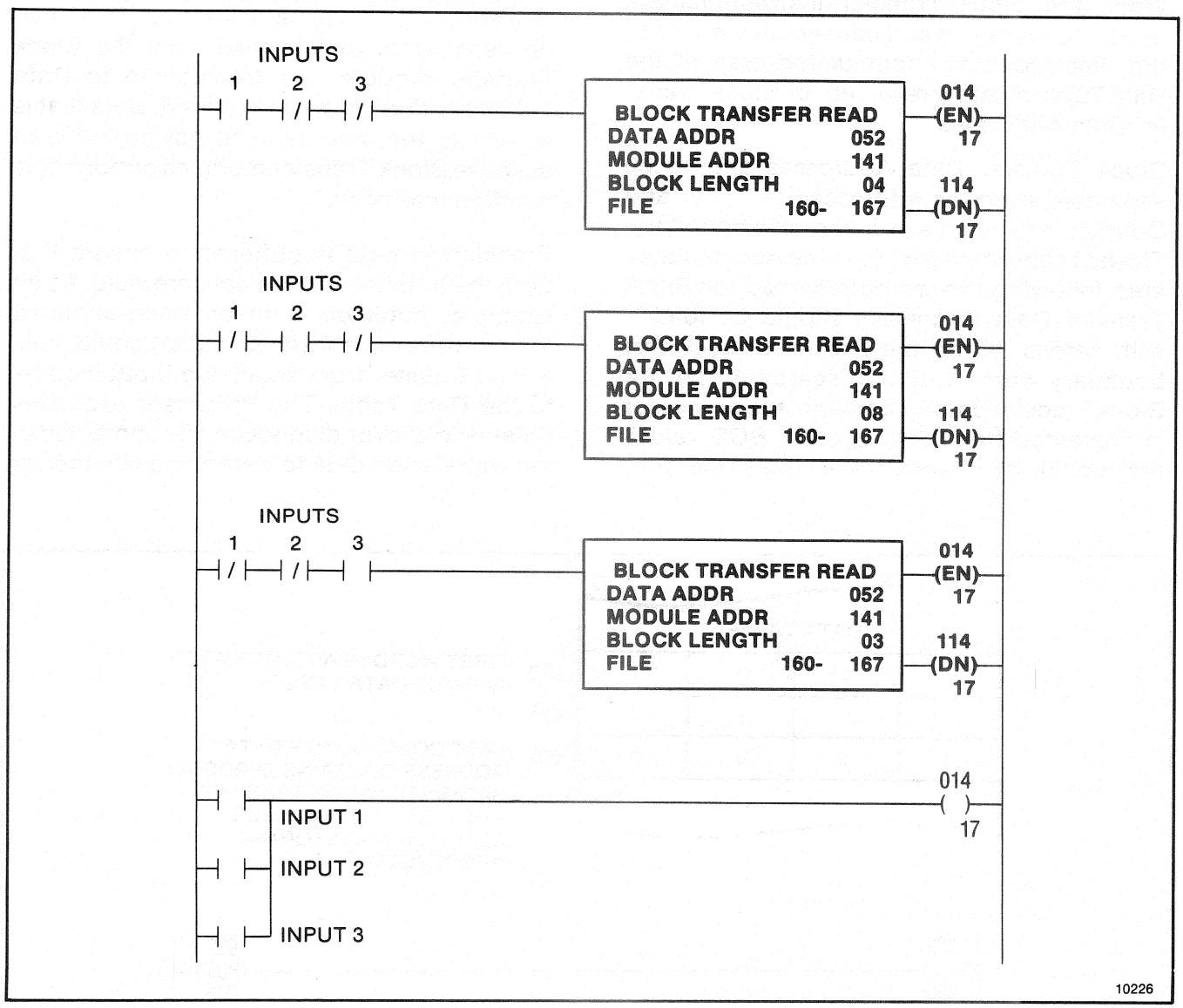


FIGURE 15-6— Programming Multiple Reads from One Module

WARNING: When programming multiple WRITES (or READS) to the same module, programming errors could prevent the desired transfer from taking place or limit the number of words transferred. Invalid data could be sent to an analog output device (or could be operated upon in subsequent scans) resulting in unpredictable machine operation. Damage to equipment and/or injury to personnel could occur.

15.2.8 Defining the Block Transfer Data Address Area

When the Block Transfer instructions are used, the first word and consecutive words of the timer/counter Accumulated area of the Data Table must be reserved for Block Transfer Data Addresses.

Block Transfer Data Addresses should be separated from the addresses of Timer and Counter instructions by inserting a boundary. The last consecutive word in the Accumulated area following the words reserved for Block Transfer Data Addresses should be loaded with zeroes. When the Processor sees this boundary word, it will not search further for Block Transfer data. In addition, the processor is prevented from finding other BCD values that could, by chance, be in the same configuration as the rack, group and slot numbers found in Block Transfer Data Addresses.

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 15-7). The value of the GET instruction can be set to 000 when programmed.

15.2.9 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

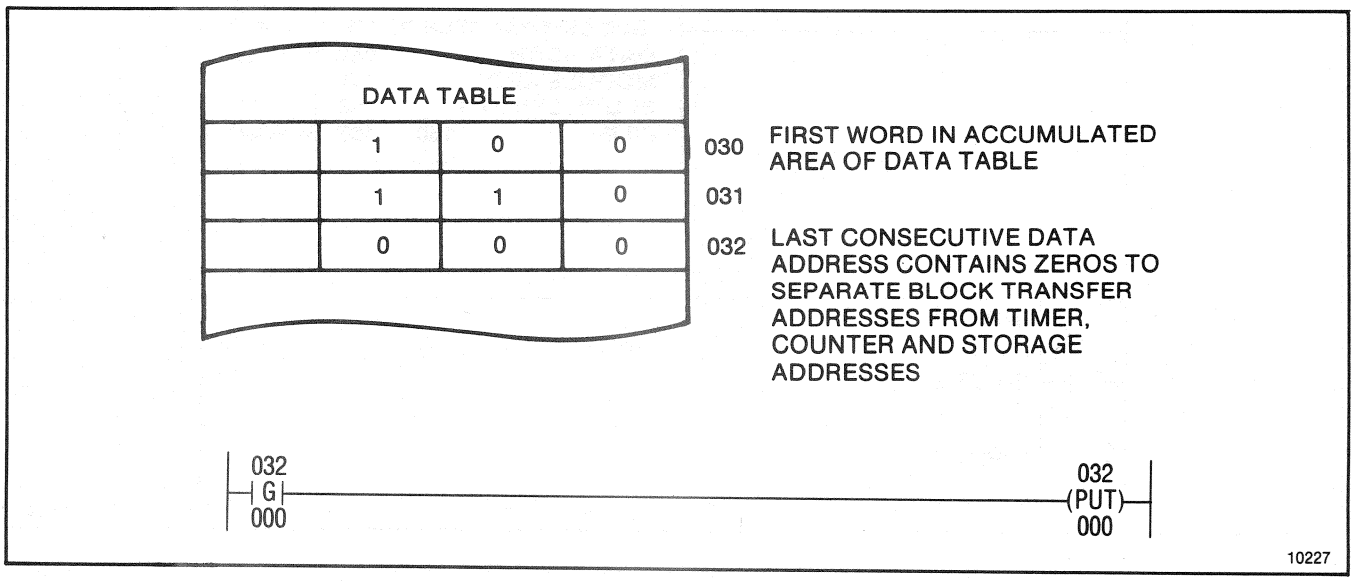


FIGURE 15-7 — Defining the Data Address Area

not the DATA is valid. The Block Transfer Done bit is set 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 file of words transferred, while others set a data-valid or 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 of bufferring data is to store the transferred data in a temporary buffer file. If the data in the buffer is valid, it is immediately transferred to another file 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 file. The technique prevents invalid data from being operated upon by preconditioning the rungs that would transfer data out of a file one word at a time. Diagnostic and/or data-valid bits are examined in these rungs.

Data can be moved from storage word-by-word using GET/PUT transfers. Or, the entire buffer file can be moved at once using a FILE-TO-FILE MOVE instruction. The choice depends on the kinds of diagnostic and/or data-valid bits and the objectives of the User Program. Generally, when one diagnostic bit is contained in each word, a GET/PUT transfer is used. When one is set for the entire file, a FILE-TO-FILE MOVE instruction is used. In either case, the diagnostic bits are examined as conditions for enabling the file move or word transfer.

The example in Figure 15-8 shows the memory map and ladder diagram rungs for buffering 3 words of data that are read from the Block Transfer module. The data is read and buffered in the following sequence:

1. When Rung 3 goes TRUE, bit 014/07 (the Block Transfer Enable 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 enabled 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, Done 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 (the Block Transfer Done bit) 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 6 and 7, the data in words 051_8 and 052_8 will be transferred to words 151_8 and 152_8, respectively, if the diagnostic bits are ON.

15.2.10 Bidirectional Block Transfer

Bidirectional Block Transfer is the sequential performance of both operations. The order of operation is generally determined by the module.

Two rungs of User Program are required, one containing the BLOCK TRANSFER READ instruction, the other containing the BLOCK TRANSFER WRITE instruction. When both instructions are given the same Module Address, the pair are considered as Bidirectional Block Transfer instructions. See Figure 15-4 for Block Transfer format and definitions.

OPERATION

The operation of Bidirectional Block Transfer is similar to that described in paragraph 15.2.2, Basic Operation. Additional considerations for Bidirectional operation will be described using an example READ and WRITE instruction with equal Block Lengths having the following parameters:

READ Instruction Data Address	040
Module Address	130
Block Length	05
File	070-074

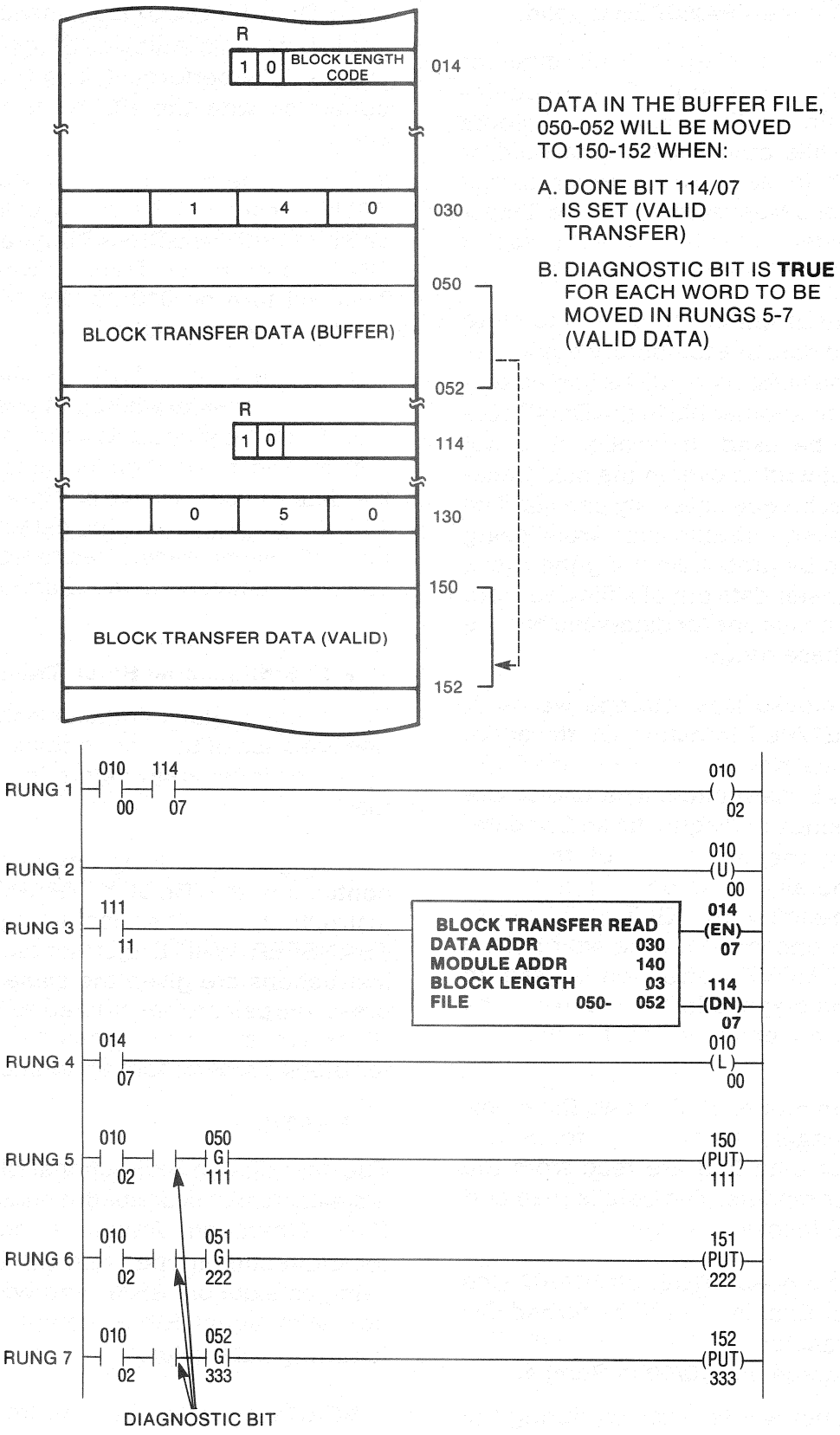


FIGURE 15-8 — Buffering Data

WRITE Instruction Data Address	041
Module Address	130
Block Length	05
File	060-064

The Data Table locations and block instructions for this example are shown in Figure 15-9.

Data Address and Module Address — The Module Address is stored in BCD in the Data Address of the READ and WRITE instructions. In this example, the Module Address is 130_8: Rack 1, Module Group 3, Slot 0.

Two Data Addresses must be used. In this example they are 040_8 and 041_8. Both contain the Module Address. For Bidirectional operation, each Data Address word also contains an Enable bit; bit 16 for a Write operation (in 041_8) and bit 17 for a Read operation (in 040_8). When the Processor searches the Data Addresses in the timer/counter Accumulated area of the Data Table, it finds two consecutive Data Addresses both containing the same Module Address. The Read bit is set high in one Data Address (040_8). the Write bit is set high in the other (041_8). When the Processor finds a match of the Module Address and the Enable bit (Read bit or Write bit) for the desired direction of transfer, it then locates the File Address to which (or from which) the data will be transferred.

File Address — Generally Two File Addresses are required: one to receive data transferred from the module, the other containing data to be transferred to the module. In this example, they are 060_8 and 070_8. The consecutive storage locations containing the File Addresses in BCD are found in the Preset area of the Data Table at Addresses 140_8 and 141_8. They are found 100_8 above the corresponding consecutive Data Addresses in the Accumulated area of the Data Table.

Block Length — The Block Lengths of the READ and WRITE instructions can be set equal or unequal to each other up to any value not exceeding the default (maximum) Block Length of the module. If the default value is used, it instructs the module to control the number of words transferred. Although the default value varies fom one kind of module to another, it can be entered into the instruction block as the number 00 for all Block Transfer modules. See the Module Data Sheet for additional information on the Block Transfer module of interest.

In the example, both Block Lengths are set equal to 05.

PROGRAMMING CONSIDERATIONS

The programming of a Bidirectional Block Transfer module depends on whether the READ and WRITE instruction Block Lengths are equal or unequal.

Equal Block Lengths — When the Block Lengths are set equal or when the default Block Length is specified by the programmer, the following considerations are applicable:

- Both the READ and WRITE instructions could and should be enabled in the same scan (separate but equal input conditions).
- The module decides which operation will be performed first when both instructions are enabled in the same scan.
- The alternate operation will be performed in a subsequent scan.
- Transferred data should not be operated upon until the Done bit is set.

Unequal Block Lengths — Consult the User's Manual for the Block Transfer module of interest for programming guidelines when setting the Block Lengths to unequal values.

WARNING: When the Block Lengths of Bidirectional Block Transfer instructions are set to unequal values, the rung containing the alternate instruction must not be enabled until the Done bit of the first transfer is set. If they are enabled in the same scan, the number of words transferred may not be the number intended, invalid data could be operated upon in subsequent scans or analog output devices could be controlled by invalid data. Unexpected machine operation could occur with possible damage to equipment and/or injury to personnel.

Refer to Table 15-2 for a summary of Block Transfer instructions.

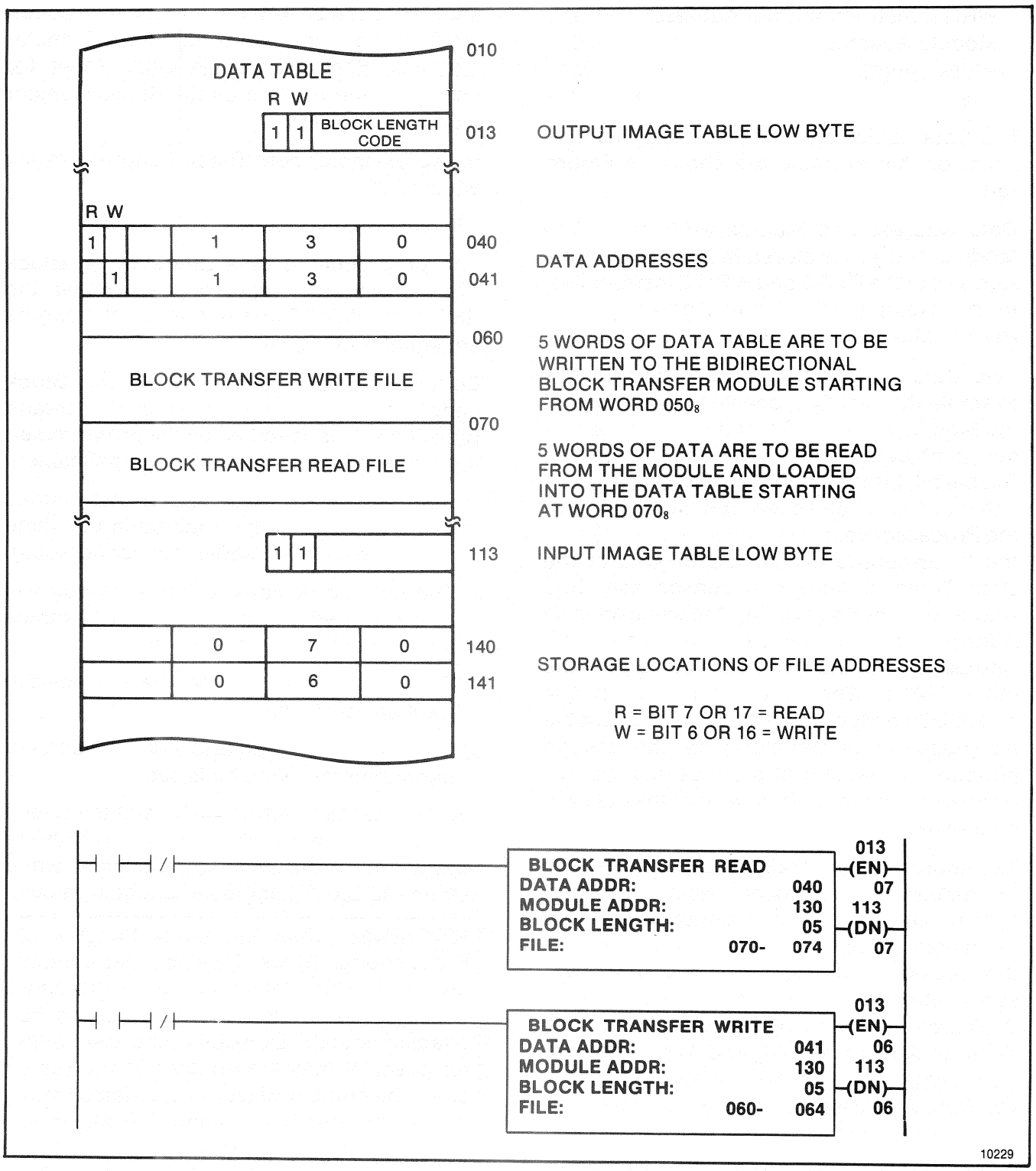
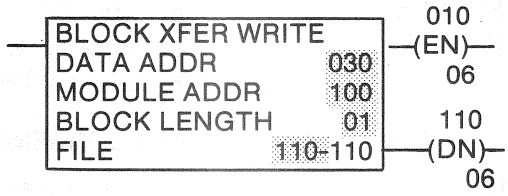
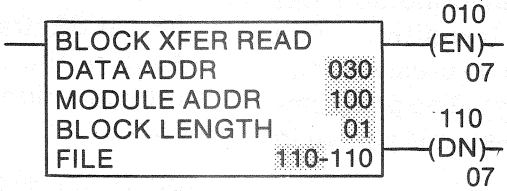


FIGURE 15-9 — Data Table Locations for Bidirectional Block Transfer

TABLE 15-2 — Block Transfer Instructions

KEY SEQUENCE	1770-T3 DISPLAY	INSTRUCTION NOTES
[BLOCK XFER] [0]		Output instruction. Block Length depends on kind of module. Entire file transferred in 1 scan. Done bit remains ON for 1 scan after valid transfer.
[BLOCK XFER] [1]		Data read from I/O module must be bufferred. Uses 2 words of User Program for each instruction
[BLOCK XFER] [0] [BLOCK XFER] [1]	Enter both instruction blocks for Bidirectional block transfer	Set Block Lengths equal or to default value for module. Same Module address used for READ and WRITE instruction. Enable READ and WRITE instructions in same scan. Order of operation determined by the module. Refer to Module User's Manual.

NUMBERS SHOWN ARE DEFAULT VALUES. NUMBERS IN SHADED AREAS MUST BE REPLACED BY USER-ENTERED VALUES. THE NUMBER OF DEFAULT ADDRESS DIGITS INITIALLY DISPLAYED (3 OR 4) WILL DEPEND ON THE SIZE OF THE DATA TABLE.

DATA ADDRESS	:	FIRST POSSIBLE ADDRESS IN ACCUMULATED VALUE AREA OF DATA TABLE.
MODULE ADDRESS	:	RGS FOR R = RACK, G = MODULE GROUP, S = SLOT NUMBER.
BLOCK LENGTH	:	NUMBER OF WORDS TO BE TRANSFERRED. (00 CAN BE ENTERED FOR DEFAULT VALUE)
FILE	:	ADDRESS OF FIRST WORD OF THE FILE, 100_8 ABOVE THE DATA ADDRESS.

Note: Access Data Monitor Display as follows:

Enter all instruction parameters. Press key sequence: [DISPLAY][0] for Binary, [DISPLAY][1] for Hex.

15.3 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

15.3.1 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 15-10.

When bit 112/04 makes a FALSE-to-TRUE transition, the one-shot bit (bit 253/00) 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 253/00 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.

15.3.2 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 15-11.

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 253/00) is set ON and bit 011/14 is unlatched. Bit 253/00 will remain ON for only one scan. The input bit 112/04 must go TRUE and

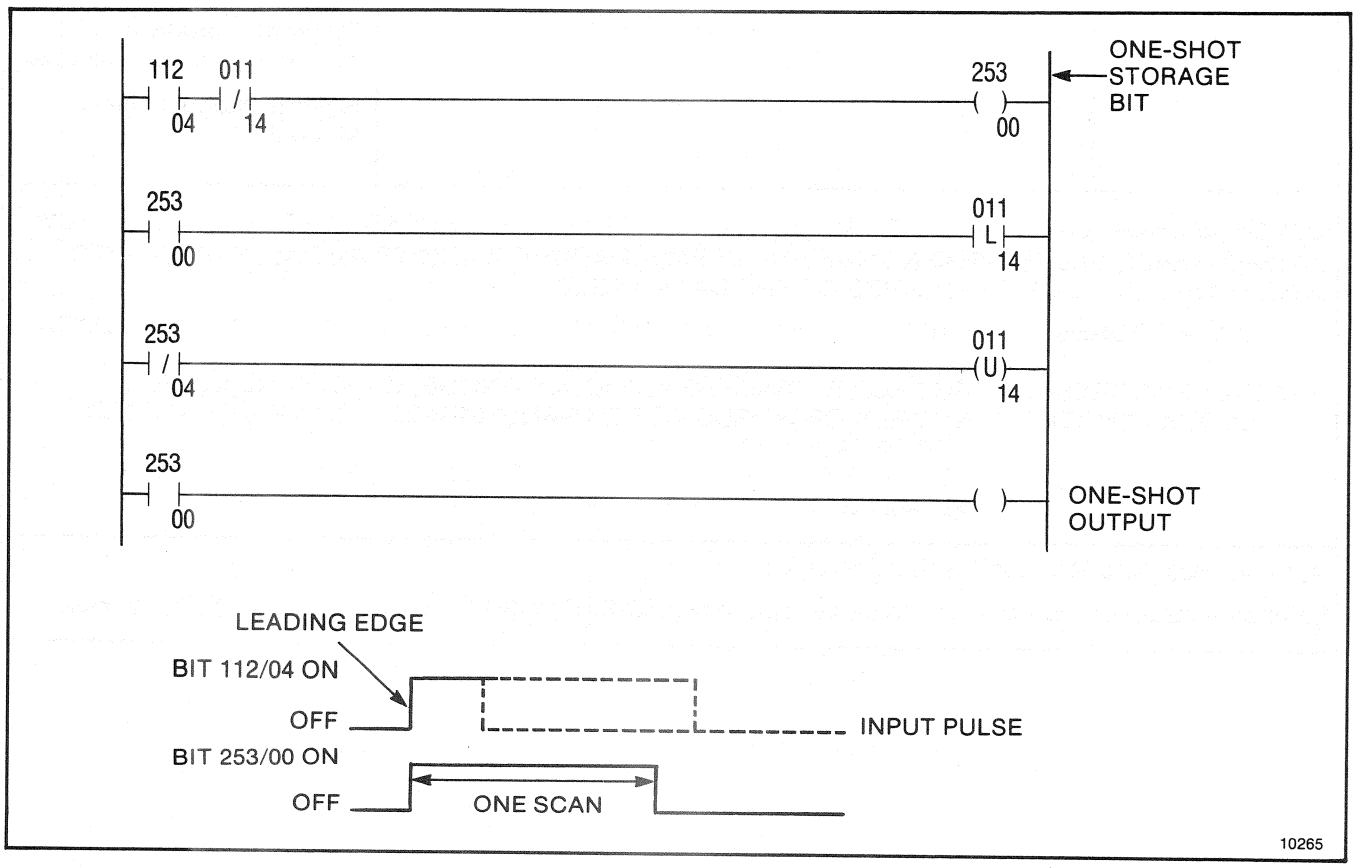


FIGURE 15-10 — Leading Edge One-Shot

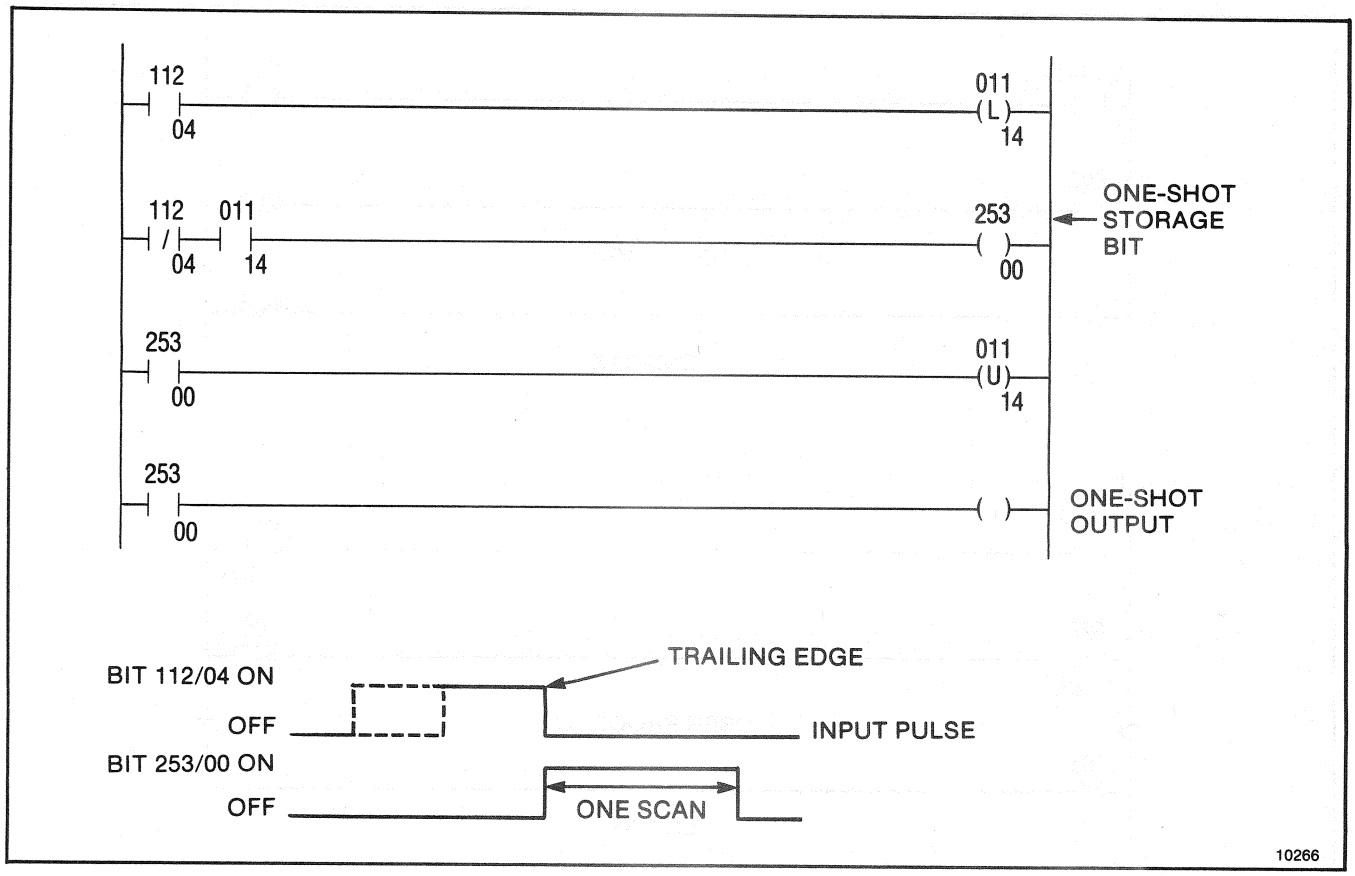


FIGURE 15-11 — Trailing Edge One-Shot

FALSE to set the one-shot bit ON for another scan.

15.4 MANUAL RESTART

Start-up can be controlled manually when using the EPROM back-up memory. Anytime that an EPROM to RAM memory transfer occurs, bit 027/01 in the Data Table will be set to "0" by the Processor. This allows a machine Start switch to be programmed in either of two ways as shown in Figure 15-12 a) and b).

The technique shown in a) can be used when the User Program does not contain any MCR instructions. The technique shown in b) is an alternative if an unused LABEL number is available.

The values in the Data Table at start-up will depend on whether or not the memory was retained by back-up battery. If a battery was used, the Data Table will contain the values which existed when power was removed. If no battery was used, the values programmed into EPROM will be transferred into the Data Table at power-up.

Start-Up conditions can be summarized for manual start-up (using one of the suggested programming techniques) and for automatic start-up (where neither technique is used) as follows:

a) from initial start-up conditions: remove the battery from the system.

b) from in-process conditions at time of power loss: maintain battery back-up.

WARNING: The battery should NOT be replaced during a power loss.

An EPROM to RAM transfer will not take place upon power-up when the battery is operating and the memory content matches EPROM content. A Transfer will take place if any alteration of memory content occurred while the battery was being changed. If a transfer

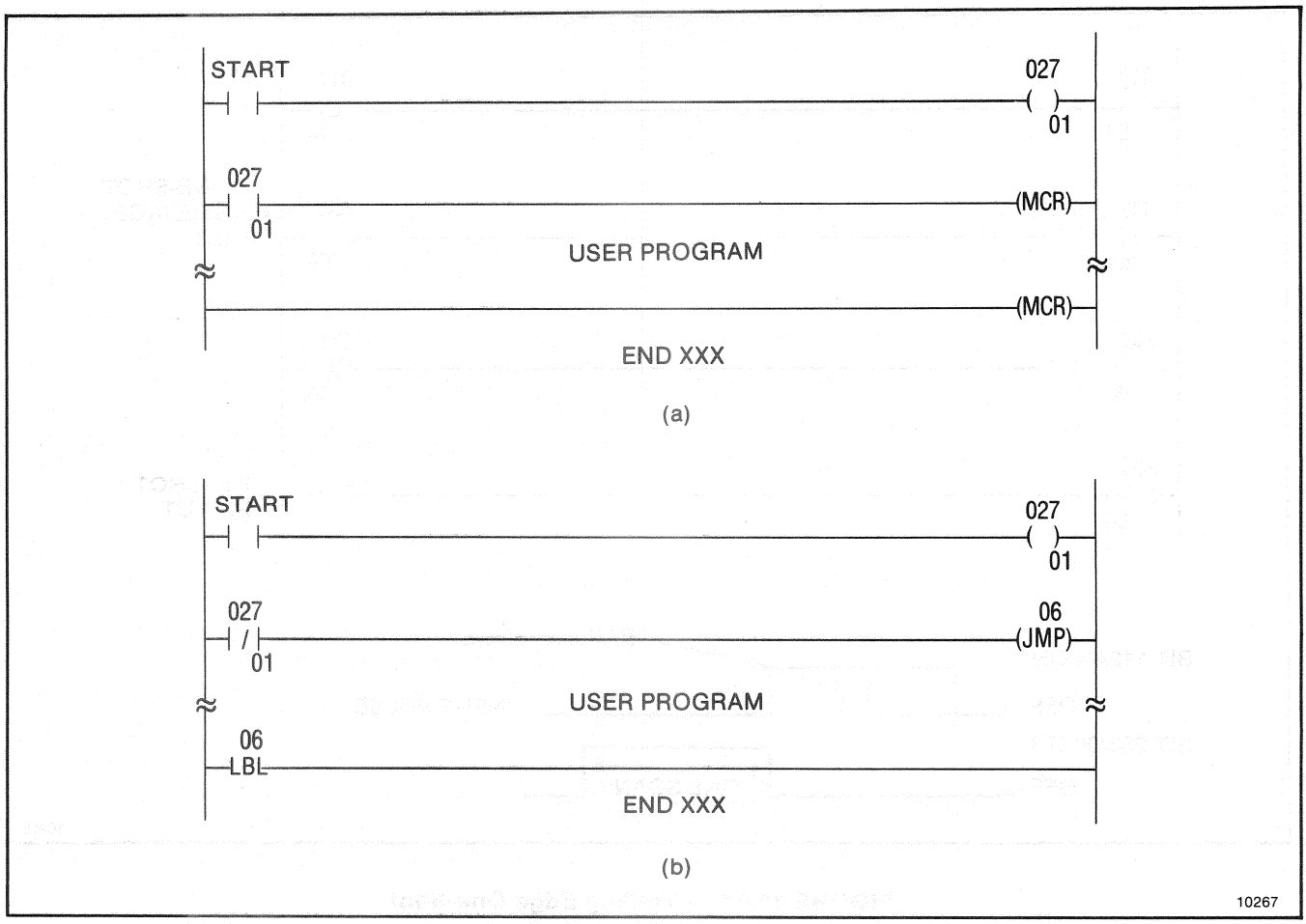


FIGURE 15-12 — Manual Restart Example Programming

occurred (memory was altered), the Data Table will contain the values programmed into the EPROM. If the transfer did not occur (memory was not altered), the Data Table will contain the values which existed at the time the system power was removed. There is no way of determining whether the transfer will occur if the battery is replaced during a power loss.

SECTION 16
SCAN TIME AND INSTRUCTION EXECUTION TIMES

16.0 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.

16.1 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.

The Scan time can increase when the Processor is instructed to execute one or more subroutines that otherwise would not be scanned. The Scan time can decrease when the Processor is instructed to JUMP forward in User Program because the jumped instructions are not scanned.

Scan time can vary from 18-24 msec typically for a 1K program. The scan time is increased by approximately 10% or 3 msec (whichever is greater) when the Industrial Terminal is connected to the Processor. When a Communications Adapter Module or 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 20% or 5 msec (whichever is greater).

16.1.1 Determination of Average Scan Time

Using the scan counter described in Section 15.1, a brief program can be written to display average scan time (See Figure 16-1). For 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 4 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 it is, incorrect input data could be used during program execution. This could cause unpredictable machine operation and result in 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.

16.2 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 130 msec.

The time required for the Processor to execute some instructions can be quite long in comparison with other instructions. Repeated use of instructions with longer execution times could cause the watchdog timer to time out. For this reason, the watchdog timer is reset automatically everytime the Processor executes any one of the following instructions:

FILE-TO-FILE MOVE,FILE-TO-WORD MOVE, WORD-TO-FILE MOVE, SEQUENCER INPUT, SEQUENCER LOAD, SEQUENCER OUTPUT, RETURN, TEMPORARY END, SUBROUTINE AREA

16.3 INSTRUCTION EXECUTION TIMES

Execution times in microseconds for each instruction are presented in Table 16-1. 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.

16.3.1 MULTIPLY and DIVIDE Instructions

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.

16.3.2 Addresses Above 377_8

These instructions require two words of memory and thus take approximately 27 microseconds longer.

16.3.3 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 execution time of each instruction between the Start Fence and End Fence is 30 microseconds per word.

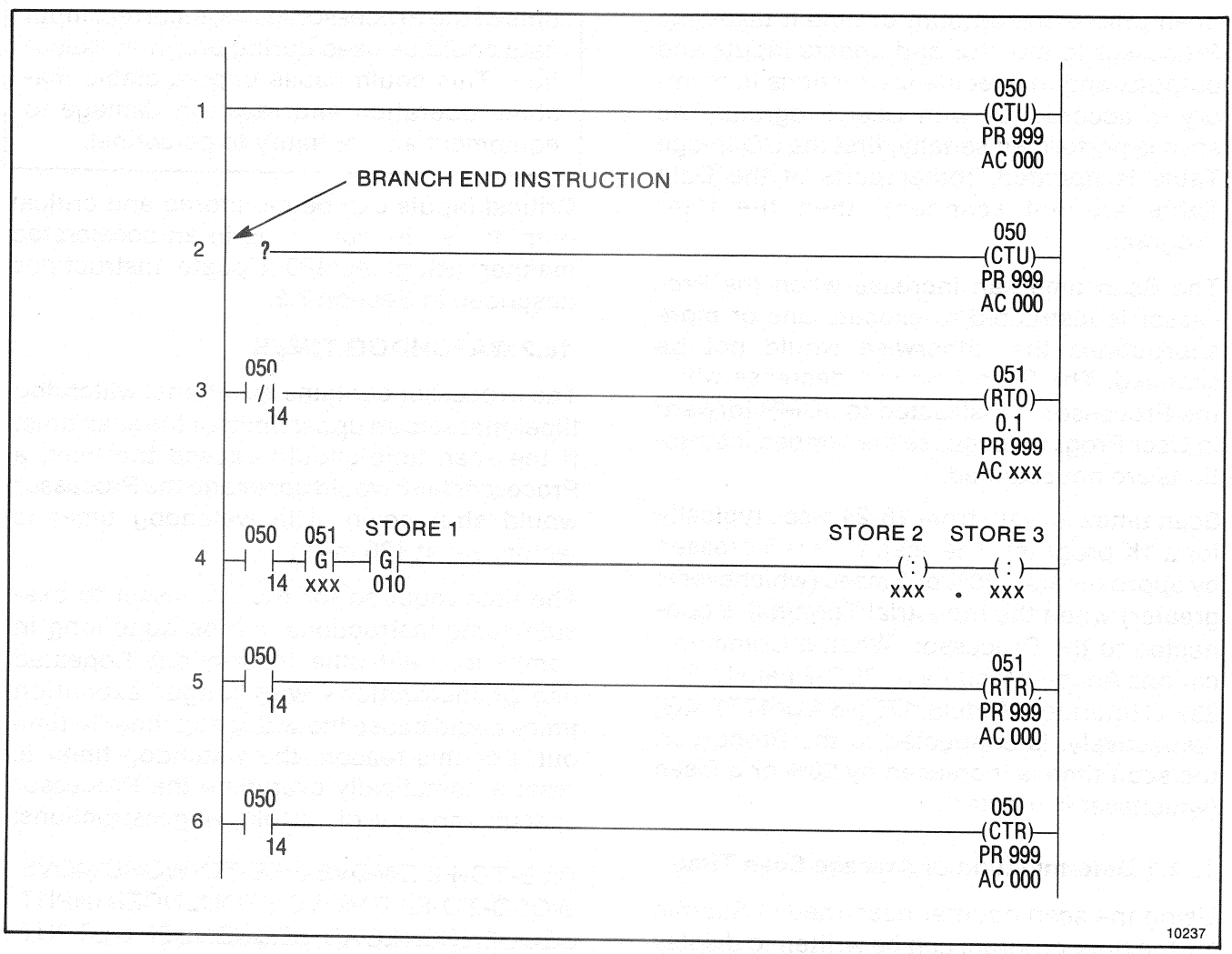


FIGURE 16-1 — Scan Time Program

TABLE 16-1 — Instruction Execution Times

INSTRUCTION NAME	APPROXIMATE EXECUTION TIME PER SCAN (IN MICROSECONDS)	
	INSTRUCTION FALSE	INSTRUCTION TRUE (AVG.)
EXAMINE ON, EXAMINE OFF	6	12
OUTPUT ENERGIZE	23	24
OUTPUT LATCH	13	26
OUTPUT UNLATCH	13	26
GET	—	27
PUT	16	28
EQUAL	6	29
LESS THAN	6	33
GET BYTE	—	11
LIMIT TEST	6	25
COUNTER RESET	16	24
RETENTIVE TIMER RESET	16	24
TIMER ON-DELAY	120	250
RETENTIVE TIMER ON-DELAY	120	250
TIMER OFF-DELAY	145	250
UP COUNTER	150	200
DOWN COUNTER	150	200
ADD	16	52
SUBTRACT	20	92
MULTIPLY	43	350-620 (See 17.3.1)
DIVIDE	40	500-930 (See 17.3.1)
Add to any of the above when its address is 400_8 or greater	27	27 (See 17.3.2)
MASTER CONTROL RESET	21	24
ZONE CONTROL LAST STATE	28 (See 17.3.3)	28
BRANCH START	13	18
BRANCH END	13	18
END, TEMPORARY END	27	27
SUBROUTINE AREA	27	27
IMMEDIATE INPUT	33	140
IMMEDIATE OUTPUT	40	200
LABEL	—	19
RETURN	16	34
JUMP TO SUBROUTINE	16	75
JUMP	16	145
BLOCK TRANSFER READ	144	160 (See 17.3.4)
BLOCK TRANSFER WRITE	144	160 (See 17.3.4)
SEQUENCER LOAD	330	740 + 140 [# Words per step - 1]
SEQUENCER INPUT	250	800 + 180 [# Words per step - 1]
SEQUENCER OUTPUT	400	825 + 180 [# Words per step - 1] (See 17.3.5)
FILE-TO-WORD MOVE	325	570
WORDS-TO-FILE MOVE	325	575
FILE-TO-FILE MOVE		
Incremental Mode	400	1200
Complete or Distributed	400	1200 minimum (See 17.3.6)

16.3.4 BLOCK TRANSFER Instructions

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.

16.3.5 SEQUENCER Instructions

The execution times will vary according to the number of words per step used in the instruction (i.e., the number of Output, Input or Load words from 1 through 4). For example, the execution time for a 3-words per step SEQUENCER OUTPUT instruction, when TRUE, (from Table 16-1) would be:

Execution time = 825 + 180 [# words per step -1]
= 825 + 180 [3 -1]
= 1185 microseconds
= 1.185 msec

16.3.6 FILE-TO-FILE MOVE Instruction

The execution time for the instruction operating in the Complete or Distributed Complete mode depends upon R, the number of words operated upon per scan. When R = 1, the execution time is 1200 microseconds. When R is greater than 1, the execution time is increased by 15 microseconds per word. For maximum efficiency of the instruciton, R should not exceed 20 words per scan. For example, the execution time for a FILE-TO-FILE MOVE instruction, when TRUE, where R = 18 (from Table 17-1) would be:

Execution Time = 1200 + 15 (R)
= 1200 + 15 (18)
= 1.470 microseconds
= 1.47 msec

This would be the execution time (per scan) for an 18 word file operating in the Complete mode or for a larger file (multiple of 18 words) operating in the Distributed Complete mode.

Section 17
NUMBERING SYSTEMS

17.0 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.

17.1 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 17-1), 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.

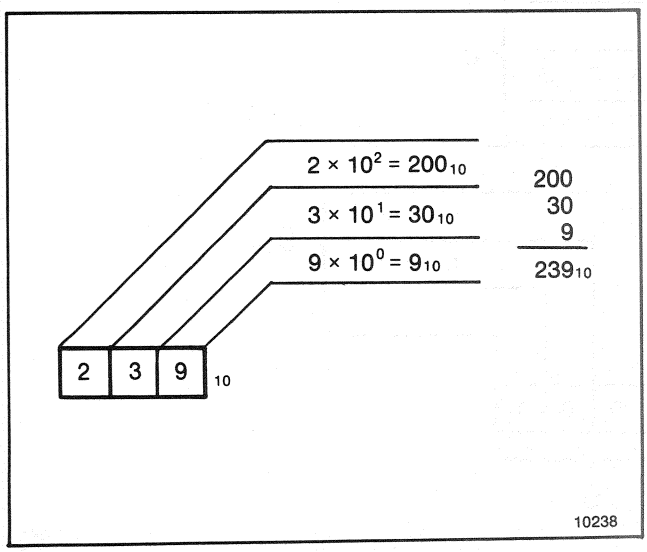


FIGURE 17-1 — Decimal Numbering System

17.2 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 17-2).

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

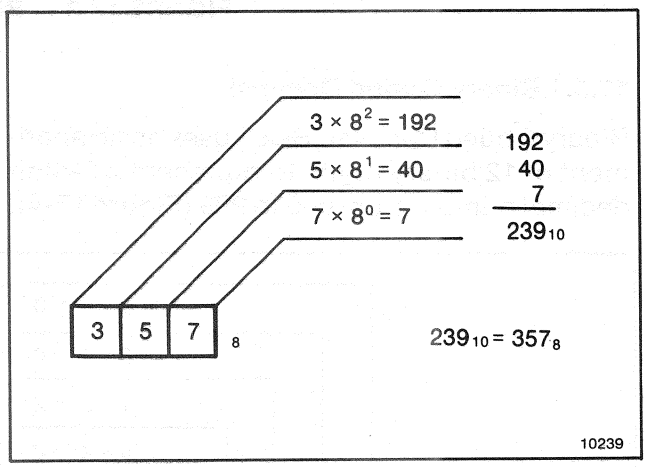


FIGURE 17-2 — Octal Numbering System

17.3 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 17-3). 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.

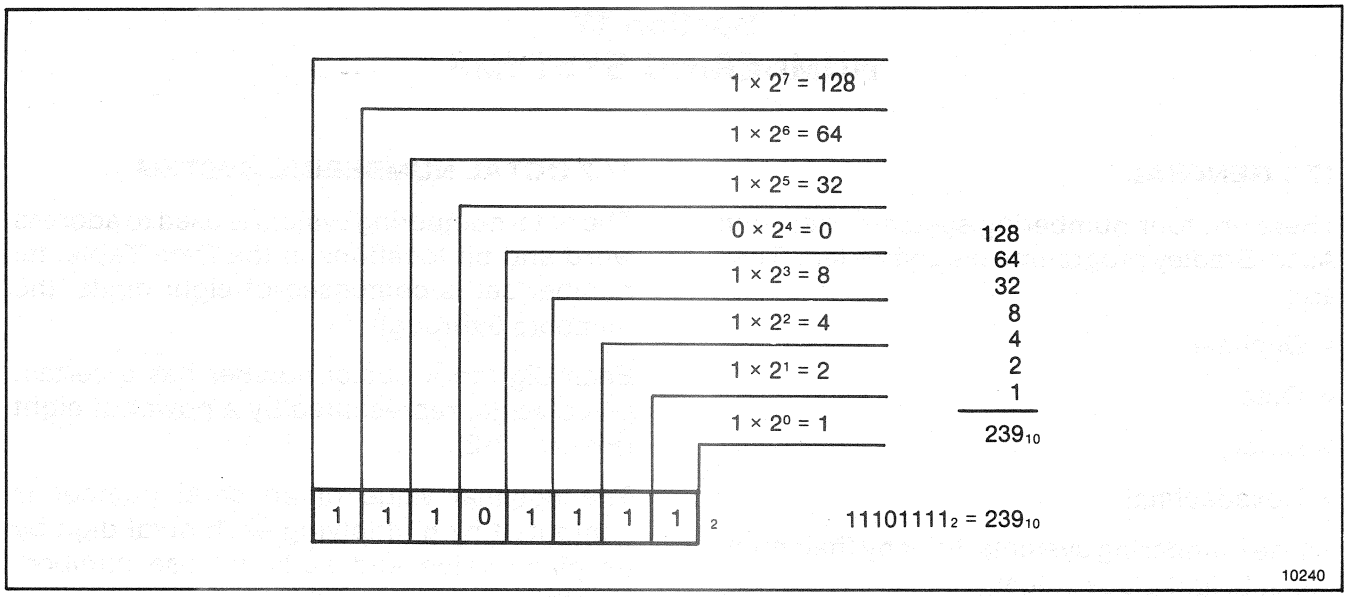


FIGURE 17-3 — Binary Numbering System

17.3.1 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 17-4).

Each group of 4 binary digits is used to represent a decimal number from 0 to 9. The place values for each group of 4 digits are 2^0, 2^1, 2^2 and 2^3 (Table 17-1).

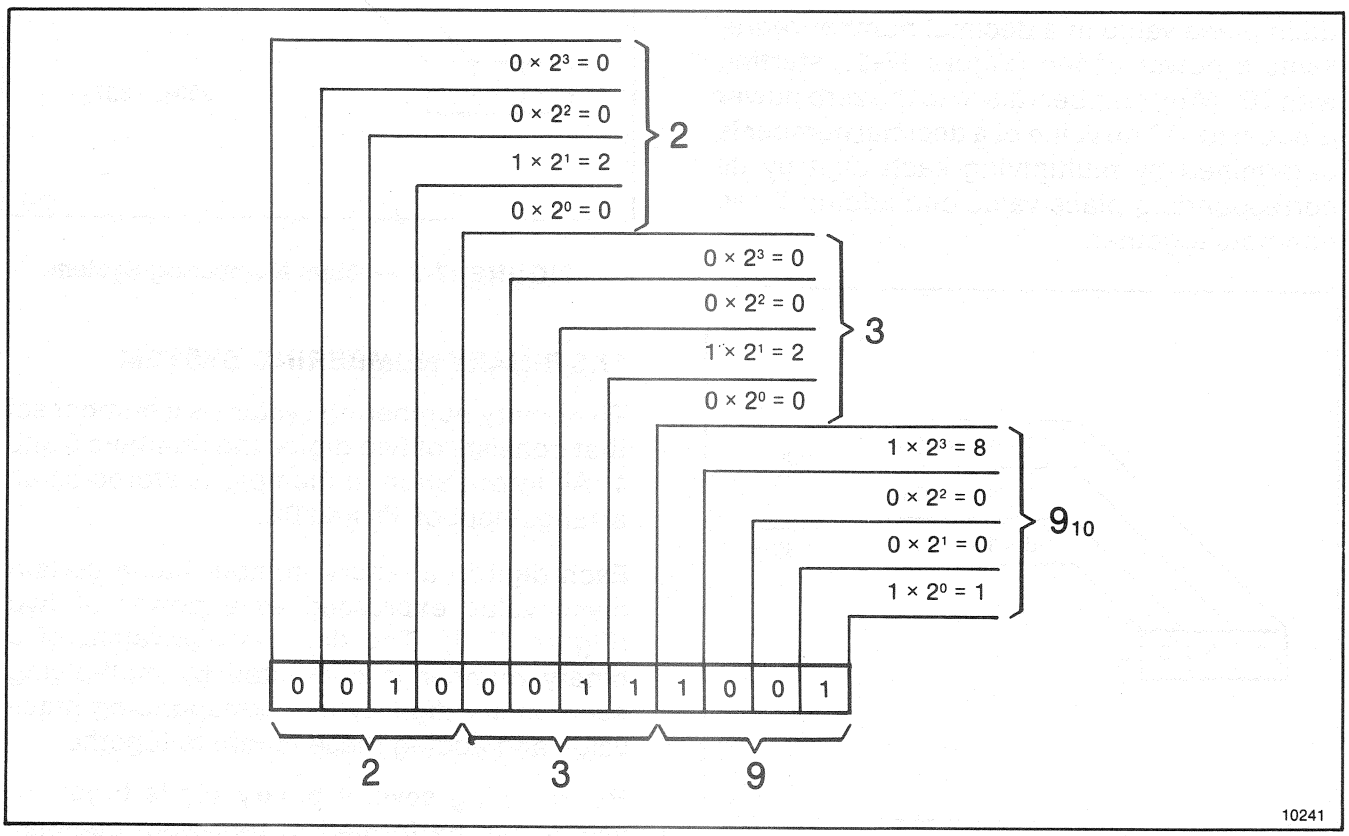


FIGURE 17-4 — Binary Coded Decimal

TABLE 17-1 — 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

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.

17.3.2 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 17-5). The 8 bits are separated into three groups: 2 bits, 3 bits and 3 bits.

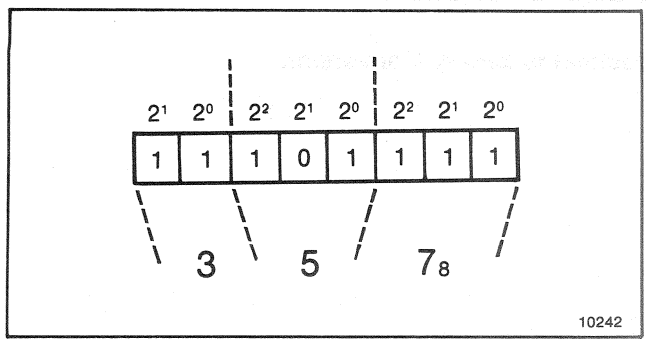


FIGURE 17-5 — Octal Representation

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 17-2).

17.4 HEXADECIMAL NUMBERING SYSTEM

The hexadecimal numbering system has a number set of 16 digits: the numbers 0-9 and the letters A-F (Table 17-3). 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 17-6).

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 17-7).

TABLE 17-2 — 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

TABLE 17-3 — Numbering System Conversion Chart

HEXA-DECIMAL	BINARY	DECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0110	4
5	0101	5
6	0100	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

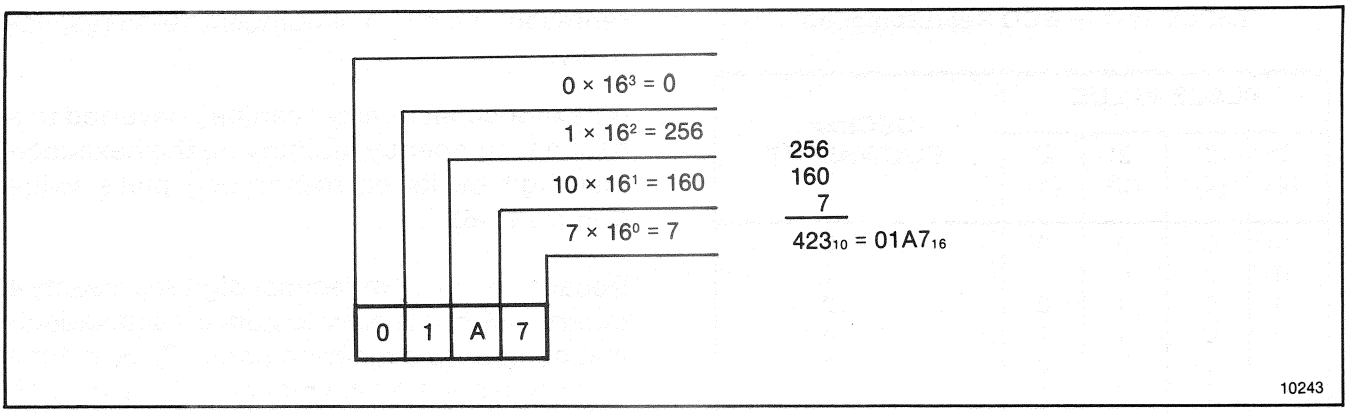


FIGURE 17-6 — Example of Hexadecimal to Decimal Conversion

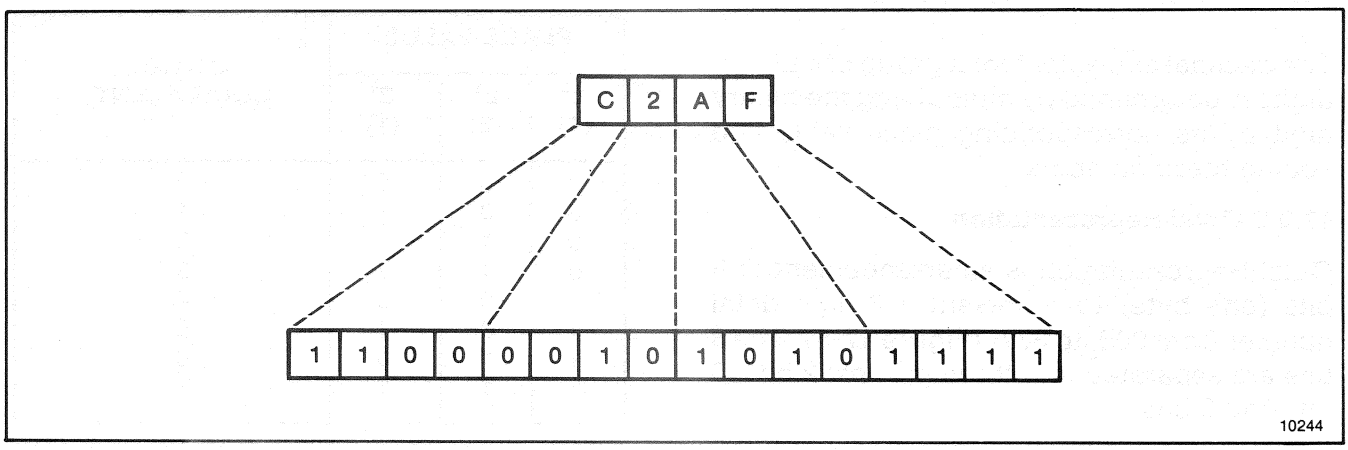


FIGURE 17-7— Example of Hexadecimal to Binary Conversion

INDEX

This is a page number index. The suffix F or T indicates respectively a figure or table on the given page.

Accumulated Value 4-1
ADD -(+)- 6-1
Address Delimiters 14-4
Addressing
  Terminology 1-7
  Method 13-2
ASCII Control Codes 14-7T

Baud Rate Setting 14-1
Bit 1-3
Bit Manipulation and Monitor 13-9, 13-16T
Block Length 15-3, 15-11
BLOCK TRANSFER
  Bidirectional 15-9
  READ and WRITE 15-3
Boundaries to User Program
  Cursor to 13-14T
  END Statement 13-14T
  TEMPORARY END 13-11
  SUBROUTINE AREA 8-1
BRANCHING, START and END 3-4
Buffering Data 15-8
Byte 1-3, 5-1F

Cassette Recording 14-11
Clear Memory Functions 13-12, 13-17T
Command Buffer 9-11
Complete Mode 9-4
Contact Histogram 14-1
Control Bit (see Status/Control Bits)
Counter Address, File instruction 9-1, 9-2
COUNTER, DOWN -(CTD)- 4-8
COUNTER, RESET -(CTR)- 4-7
COUNTER, UP -(CTU)- 4-6
Current Step (Position)
  File Instruction 9-1, 9-4, 9-6
  Sequencer Instruction 11-1, 11-2
Cursor Controls 9-11

Data Address, Block Transfer 15-3, 15-8
Data Cartridge Recording 14-12
Data Initialization 13-6
Data Monitor Mode 9-9
Data Monitor Display
  FILE TO FILE Move 9-10F
  SEQUENCER OUTPUT 11-7F
Data Table
  Factory Configured 1-5F
  Field Expandable 1-6F
  Reduction/Expansion 13-1
  Size Determination 12-9
Data Table Assignment Sheets 12-5
Data Valid Bits 15-8
Delimiter 14-5T
Diagnostic Bit 15-10F
Diagnostic Indicators 2-3, 2-4
Digit Cursor 9-12
Display Key, used for
  Contact Histogram 14-2T
  Data Monitor Mode 9-10T, 9-11T, 9-12T
  Report Generation 14-3
  Single Rung Display 13-3
Distributed Complete Mode 9-4
DIVIDE -(:)- 6-2
Done Bit (See Status/Control Bits)

Editing Functions 13-3, 13-15T
Enable Bit (See Status/Control Bits)
EPROM
  Definition 1-1, 2-2
  Indication 2-3
  Location in Processor 2-2F
  Manual Re-start 15-15
ERR Message for
  Illegal Opcode 13-11
  Incompatable Instructions 1-10
EXAMINE ON and OFF -| |-, -|/|- 3-1
Execution Times 16-2
Expand Address key 3-6, 13-2
Externally Indexed Counter 9-2

Field Cursor 9-11
File Address
  Starting Word Address 9-1
File Instruction
  Definition 9-1
  Data Monitor Display of 9-9
File Instruction Address
  Counter Address of a File instruction 9-1, 9-2

File Length 9-1
File Modes of Operation 9-4
FILE-TO-FILE MOVE 10-1
FILE-TO-WORD MOVE 10-3
Forced Address Display 13-11
Forcing Functions 13-10, 13-16T

GET and PUT -[G]- and -(PUT)- 6-1
GET BYTE -[B]- 5-4
Graphic Display 14-8F

Header 9-10
Help Directories 13-2, 13-13T

Illegal Opcode 13-11
Image Table Byte, Block Transfer 15-2
IMMEDIATE INPUT -[I]- 7-3
IMMEDIATE OUTPUT -(IOT)-
Incremental Mode 9-6
Industrial Terminal
  Compatability when using 1770-T1 or -T2 1-10
  Connection Diagram 2-5F
  Control Codes and Special Commands 14-5
  Initialization 2-5
Insert key, used for
  Editing 13-3
  Forcing Functions 13-10
  On-Line Data Change 13-5
  On-Line Programming 13-7
  TEMPORARY END instruction 13-11
Instruction Address 1-7
Instruction Block
  Example File instruction 9-3F
  FILE TO FILE MOVE 10-2F
  WORD TO FILE MOVE 10-5F
  FILE TO WORD MOVE 10-5F
  SEQUENCER OUTPUT 11-5F
  SEQUENCER INPUT 11-8F
  SEQUENCER LOAD 11-11F
Instruction Tables
  Arithmetic Instructions 6-4T
  Block Transfer Instructions 15-13T
  Counter Instructions 4-12T
  Data Manipulation Instructions 5-7T
  File Instructions 10-7T
  I/O Update Instructions 7-6T
  Jump/Subroutine Instructions 8-10T
  Output Override Instructions 7-6T
  Relay-Type Instructions 3-7T
  Timer Instructions 4-11T
Intelligent I/O Modules 15-1
Internally Indexed Counter 9-2
I/O Image Table 1-7

JUMP -(JMP)- 8-3
JUMP TO SUBROUTINE -(JSR)- 8-3

Keytop Overlay
  PLC-2 Family (Cat. No. 1770-KCB) 2-6F
  Alphanumeric and Alphanumeric/ Graphic 14-3F

LABEL -LBL- 8-1
Last state of Outputs 2-4
LES and EQU [<] and [=] 5-3
LIMIT TEST -[L]-

Manual Restart 15-15
Masking
  Sequencer Output data 11-3
  Sequencer Input data 11-7
MASTER CONTROL RESET -(MCR)- 7-1
Memory Layout Display 13-1
Memory Organization
  Definition 1-3
  Assignment Considerations 12-7
  Configuring the Data Table 13-1
  Developing the Data Table 12-5
  File Planning 9-2
  Sequencer Instructions 11-1
  Subroutine Area 8-1
Memory Write Protect
  Definition 2-2
  Clearing Memory 13-12
  On-Line Data change 13-5
  On-Line Programming 13-9
  Report Generation 14-9
Message Control Word 14-9
Message Generation 14-3
Mode Select Key, used in
  Force Functions 13-10
  Initializing the Industrial Terminal 2-5
Module Address, Block Transfer 15-2, 15-3, 15-11
MULTIPLY -(X)- 6-2

Nested Programming 3-5
Numbering Systems 17-1

On-Line Data change 13-5
On-Line Programming 13-5
One-Shot 15-14
OUTPUT ENERGIZE -( )- 3-2
OUTPUT LATCH -(L)- 3-3
OUTPUT UNLATCH -(U)- 3-3

Parameter 13-6
PC Operation 1-7
Position (Current Step)
  File Instructions 9-1, 9-4, 9-6
  Sequencer Instructions 11-1, 11-2
Preset Value 4-1, 9-1
Print Out
  Automatic Report Generation 14-8
  Contact Histogram 14-1
  Memory Dump 14-14
Processor Modes 2-1
Programming
  see Instruction Tables
  Hardware Program Interface 1-7
  Memory Organization 1-3
  Program Recommendations 12-10
  Sample Program 12-1

Rate per Scan 9-2
Read Bit (see Status/Control Bits)
Record Key, Used for
  Baud Rate Setting 14-11T
  Cassette Recorder Operation 14-11
  Data Cartridge Recorder Operation 14-13
  On-Line Programming 13-6
  Report Generation 14-3
Remote Mode Select 2-1, 13-3
Report Generation
  Direct Access at Power Up 14-8
  Simultaneous Message Requests 14-11
RETURN -(RET)- 8-5
Run-Time Errors
  Definition 2-2
  Indication 2-3
  Block Transfer Instructions 15-5
  File Instructions 9-9
  JUMP Instruction 8-3
  JUMP TO SUBROUTINE Instruction 8-5
  LABEL Instruction 8-3
  RETURN Instruction 8-5
  Sequencer Instructions 11-4, 11-8, 11-10

Scan Counter 15-1
Scan Sequence 7-2
Scan Time
  Determination of 16-1
  Typical 7-2
Search Functions 13-2, 13-12T, 13-14T
Sequencer Length 11-5F
Sequencer Table 11-1
SEQUENCER INPUT 11-6
SEQUENCER LOAD 11-9
SEQUENCER OUTPUT 11-2
Single Rung Display 13-3
Status/Control Bit, Operation of:
  Arithmetic Instructions 6-1F
  Automatic Report Generation 14-9
  Block Transfer Instructions 15-2, 15-5, 15-6F, 15-11, 15-12F
  Buffering Data 15-8, 15-10F
  Counter Instructions 4-6
  File Instructions 9-2
  Complete mode 9-4, 9-5F
  Distributed Complete mode 9-6, 9-7F
  Incremental mode 9-7, 9-8F
  SEQUENCER OUTPUT Instruction 11-3
  SEQUENCER LOAD Instruction 11-9
  Timer Instructions 4-2
Storage Considerations
  Bit/Word 12-9
  File and Sequencer 12-9
  Block Transfer 15-8
Subroutine Area
  Definition 8-1
  Location in memory 1-4F
SUBTRACT -(-)- 6-2
Switch Group Assembly 2-4

TEMPORARY END 13-11
Timer Accuracy 4-4
TIMER, OFF-DELAY -(TOF)- 4-2
TIMER, ON-DELAY -(TON)- 4-2
TIMER, RETENTIVE -(RTO)- 4-3
TIMER, RETENTIVE RESET -(RTR)- 4-4
Troubleshooting Aid Functions 13-16T

Valid Transfer of data 15-8

Watchdog Timer 16-1
Word Structure
Definition 1-3
Accumulated Value Word 4-2F, 4-6F
BCD Format 4-1F, 5-1F
Binary Data Monitor Display 11-7F
Data Table Mapping 15-6F
Message Control Word 14-10F
WORD-TO-FILE MOVE 10-3
Words-Per-Step 11-1, 11-5F
Write Bit (see Status/Control Bits)

ZONE CONTROL LAST STATE -(ZCL)- 7-1



ALLEN-BRADLEY

Publication 1772-804 — July, 1982

Systems Division — PC Business
747 Alpha Drive, Cleveland, Ohio 44143

P/N 955091-21A