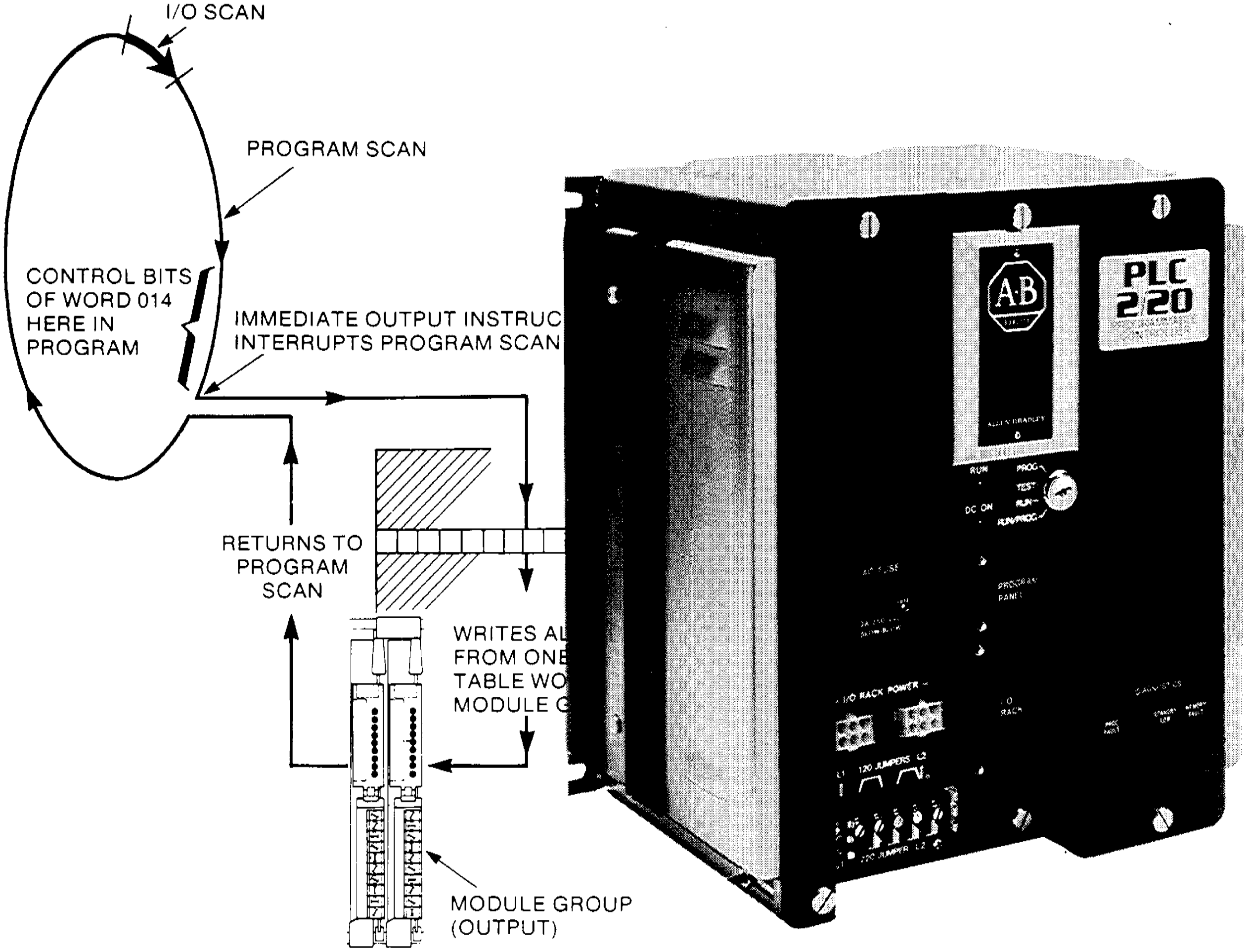
ALLEN-BRADLEY

1772-6.8.1

PLC-2/20 Programmable Controller

Programming and Operations Manual



Price: $25.00 per copy

Important User Information

Because of the variety of uses for the solid state equipment described herein, and because of the differences between it and electromechanical equipment, you must satisfy yourself as to its acceptability for each of your applications. In no event will Allen-Bradley Company be responsible or liable for indirect or consequential damages that may result from installation or use of this equipment.

The illustrations, charts, and layout examples shown in this manual are intended solely to help you understand the text, not to guarantee operation. Because of the many variables and requirements associated with any particular installation, Allen-Bradley Company will not assume responsibility for actual use based upon illustrations of 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 any part of this manual, without written permission of Allen-Bradley Company, is prohibited.

© 1984 Allen-Bradley Company

PLC is a registered trademark of Allen-Bradley Company

Table of Contents

Chapter 1

Introduction

Overview

The bulletin 1772-PLC-2/20 programmable controller is a rugged, solid state programmable controller that consists of the PLC-2/20 Processor (cat. no. 1772-LP1, or -LP2) and an I/O structure (I/O racks containing I/O modules).

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

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

The processor has a central read/write CORE or RAM memory that stores user program instructions, numeric values, I/O device status and messages. 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, rung by rung in ladder diagram format from the keyboard of an Industrial Terminal (cat. no. 1770-T1, -T2 or -T3). The ladder diagram symbols closely resemble the relay symbols used in hardwired relay control systems.

During program operation, the processor continuously monitors the status of input devices and, based on user program instructions, either energizes or de-energizes output devices. Because the memory is programmable the user program can be readily changed to suit application needs.

In addition to on/off control, the controller can perform functions with 3- digit numeric values such as:

- Timing/counting operations
- Arithmetic (+, -, x, _)
- Data transfers
- Data comparisons
- Block transfer

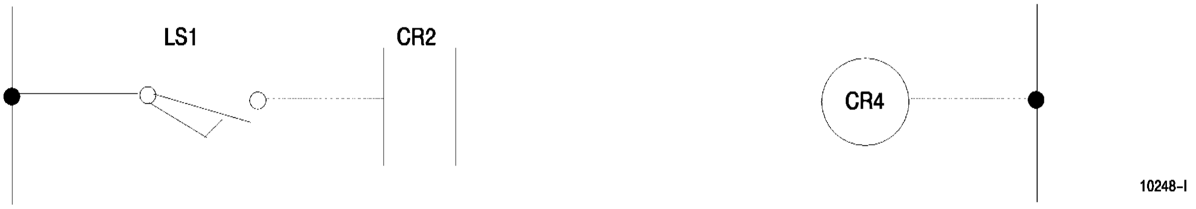
When the industrial terminal (cat. no. 1770-T1, -T2 or -T3) and certain peripheral devices are used with the PLC-2/20 programmable controller, application data can be displayed using a variety of peripheral functions:

- Report generation
- Contact histogram (the on/off history of a bit in memory)
- Hardcopy printout of the user program or the complete memory
- Recording/loading/verifying the user program using magnetic tape

Ladder Diagram Logic

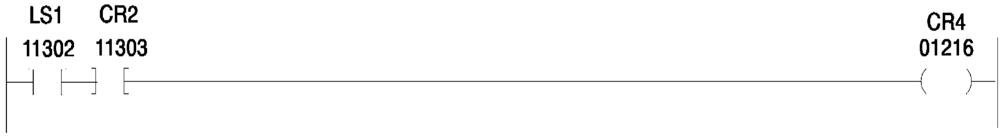
PC ladder diagram logic closely resembles hardwired relay logic. Hardwired relay control systems require electrical continuity to turn output devices on and off. For example, the relay diagram in Figure 1.1 shows that limit switch LS1 and relay contact CR2 must be closed to energize relay coil CR4.

Figure 1.1
Relay Diagram



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 1.2 shows the two input devices and the output device are assigned bit addresses in the data table. 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
Ladder Diagram Rung



Memory Structure

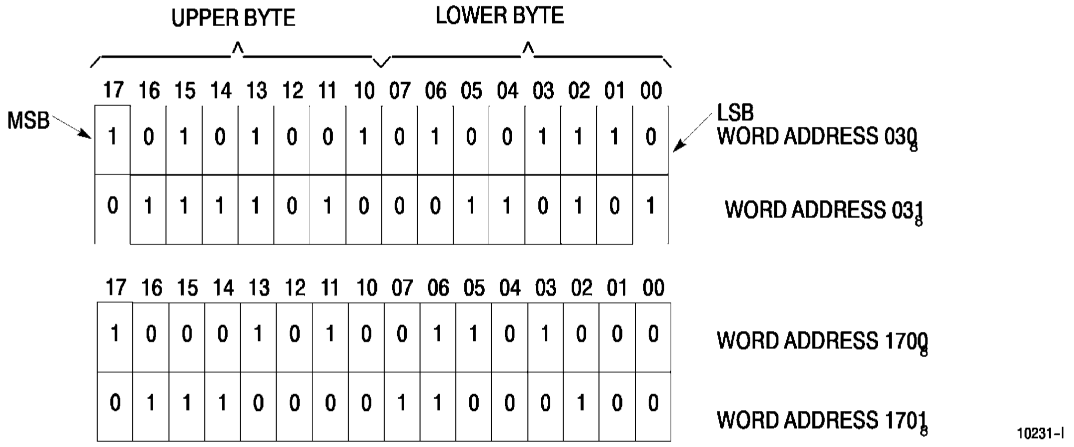
The memory of the processor is made up of an arrangement of storage points called bits (BInary digiTs). A bit is the smallest unit of memory and can store information as a “1” or a “0” (Figure 1.3). When “1” occupies a bit, that bit is on; when a “0” occupies a bit, that bit is off.

A group of 8 bits forms a single byte. Two bytes, or 16 bits, make up one memory word. All memory words are identified by their word address: a 3, 4 or 5-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) at the left and the least significant bit (LSB) at the right.

A specific bit in memory can be identified by combining the word address and bit number to form the bit address, such as 03012 or 170116.

Figure 1.3
Memory Word Structure



17	16	15	14	13	12	11	10	07	06	05	04	03	02	01	00	
1	0	1	0	1	0	0	1	0	1	0	0	1	1	1	0	WORD ADDRESS 030$_8$
0	1	1	1	1	0	1	0	0	0	1	1	0	1	0	1	WORD ADDRESS 031$_8$

17	16	15	14	13	12	11	10	07	06	05	04	03	02	01	00	
1	0	0	0	1	0	1	0	0	1	1	0	1	0	0	0	WORD ADDRESS 1700$_8$
0	1	1	1	0	0	0	0	1	1	0	0	0	1	0	0	WORD ADDRESS 1701$_8$

Memory Organization

Both PLC-2/20 processors (cat. no. 1772-LP1 and -LP2) have a memory capacity of 8192 (8K) memory words. These memory words are organized by their word address and are divided into three major areas (Figure 1.4 and Figure 1.5):

- Data table
- User program
- Message storage (if used)

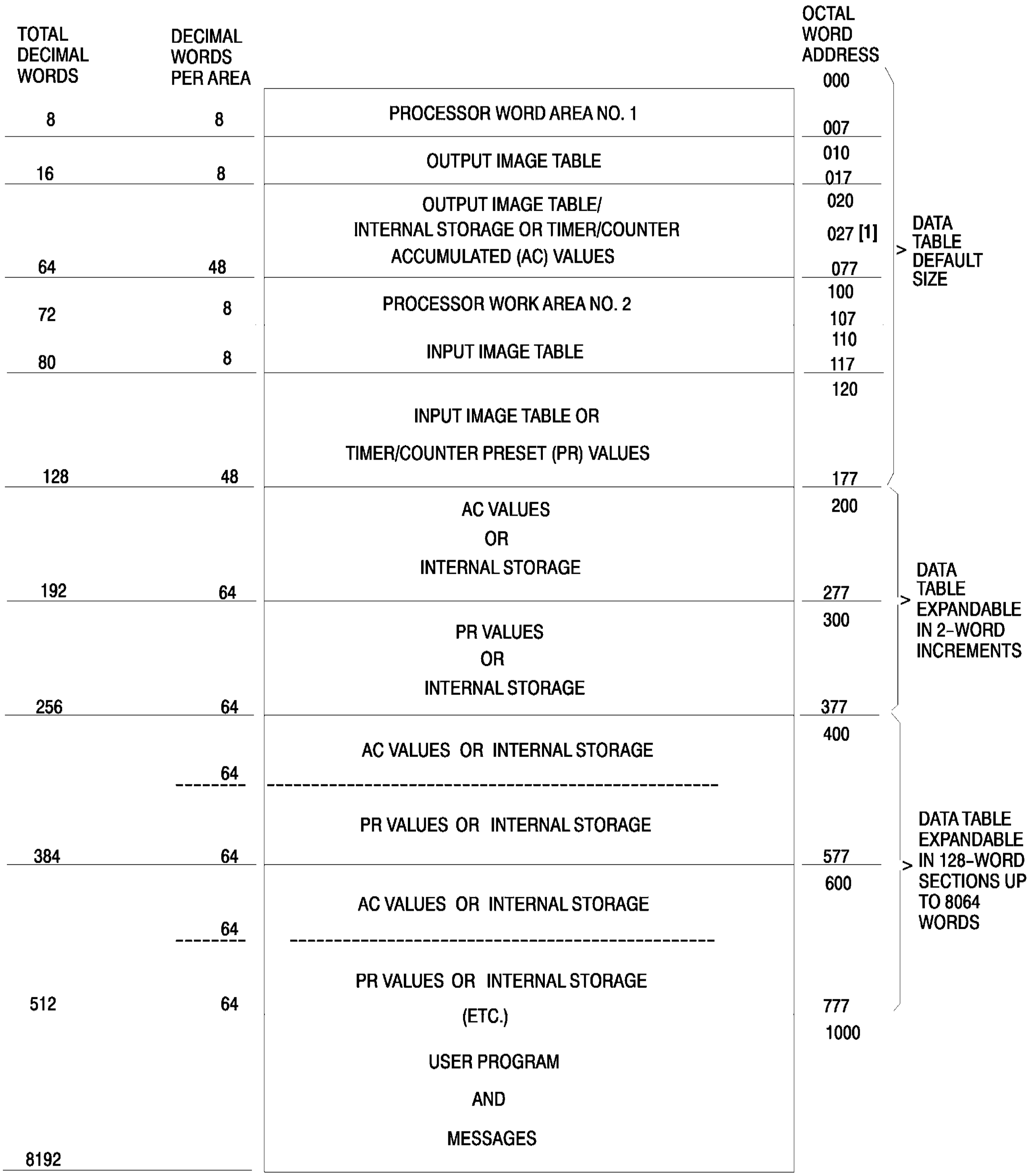
Figure 1.4
Memory Organization PLC-2/20 (cat. no. 1772-LP1)

TOTAL DECIMAL WORDS	DECIMAL WORDS PER AREA		OCTAL WORD ADDRESS	
8	8	PROCESSOR WORD AREA NO. 1	000 007	DATA TABLE DEFAULT SIZE
16	8	OUTPUT IMAGE TABLE	010 017	
40	24	OUTPUT IMAGE TABLE/ INTERNAL STORAGE OR TIMER/COUNTER ACCUMULATED (AC) VALUES	020 027 [1] 047	
64	24	AC VALUES OR INTERNAL STORAGE	050 077	
72	8	PROCESSOR WORK AREA NO. 2	100 107	
80	8	INPUT IMAGE TABLE	110 117	
104	24	INPUT IMAGE TABLE OR TIMER/COUNTER PRESET (PR) VALUES	120 147	
128	24	PR VALUES OR INTERNAL STORAGE	150 177	
192	64	AC VALUES OR INTERNAL STORAGE	200 277	DATA TABLE EXPANDABLE IN 2-WORD INCREMENTS
256	64	PR VALUES OR INTERNAL STORAGE	300 377	
8192		USER PROGRAM AND MESSAGES	400	

[1] BITS IN THIS WORD ARE RESERVED FOR BATTERY LOW CONDITION, MESSAGE GENERATION AND DATA HIGHWAY.

10232–I

Figure 1.5
Memory Organization PLC-2/20 (cat. no. 1772-LP2)



[1] BITS IN THIS WORD ARE RESERVED FOR BATTERY LOW CONDITION, MESSAGE GENERATION AND DATA HIGHWAY.

10233-I

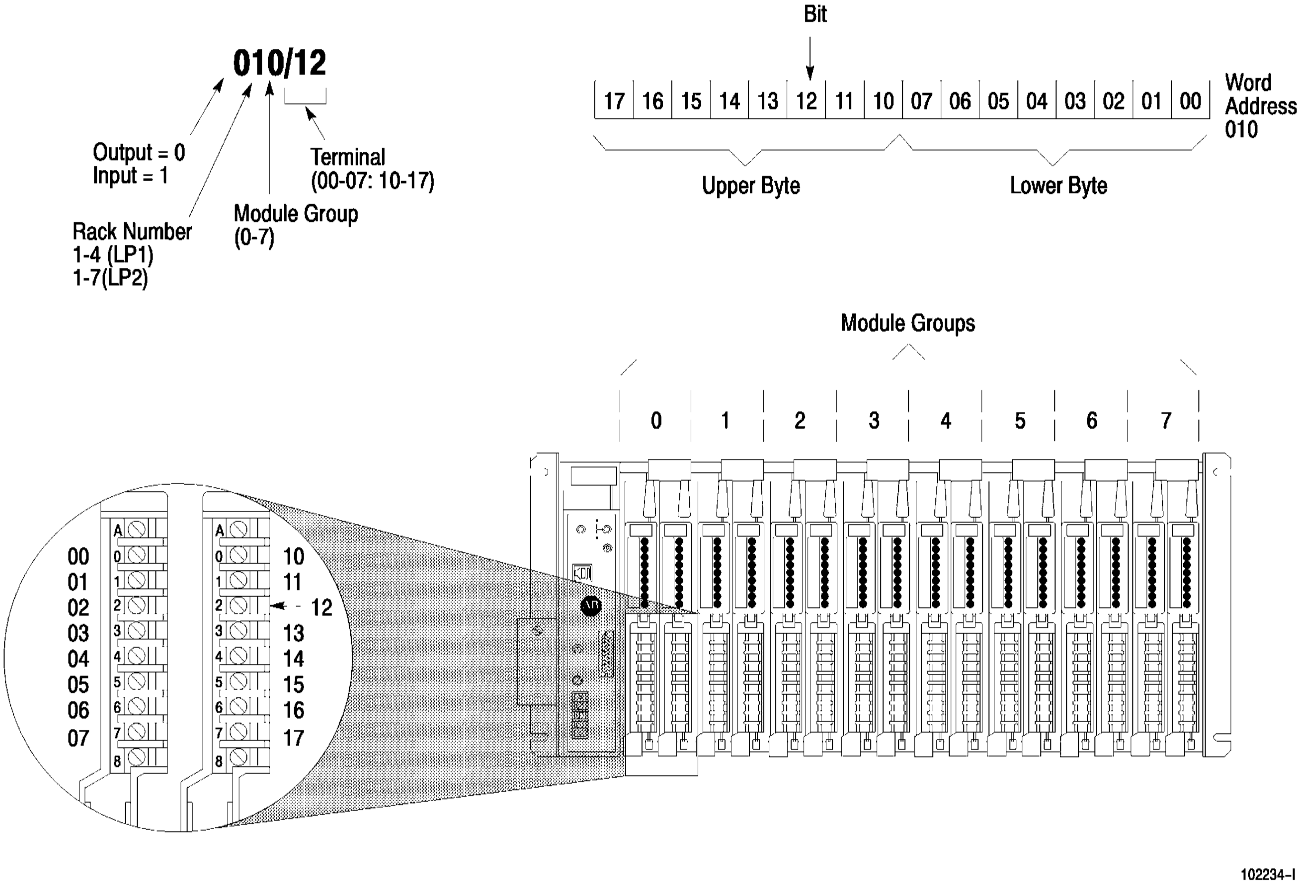
Data Table

The data table area has a default size of 128 words. This area stores the information needed in the execution of the user program such as input and output device status, 3-digit numeric values and the status of internal storage points.

Of all the data table areas, only the input/output image table has a direct relationship to hardware. An I/O image table word corresponds to two I/O modules located in a group of an I/O rack (Figure 1.6). The lower byte of an I/O image table word corresponds to an I/O module in the left slot of the module group and the upper byte corresponds to an I/O module in the right slot of the module group. Each 5 digit bit address in the I/O image table then directly relates to an I/O module in the following way:

- The first digit indicates either input (1) or output (0)
- The second digit identifies the I/O rack number, determined by setting switches on the switch group assembly (section titled Switch Group Assembly, chapter 2)
- The third digit identifies the module group within the I/O rack
- The last two digits identify the I/O module terminal

Figure 1.6
Bit Address to Hardware Relationship



User Program and Messages

The user program follows the Data Table in memory and stores all the user program instructions that make up the ladder diagram program. Instructions with addresses up to 3778 are stored in one memory word. Instructions with addresses of 4008 or above require two memory words.

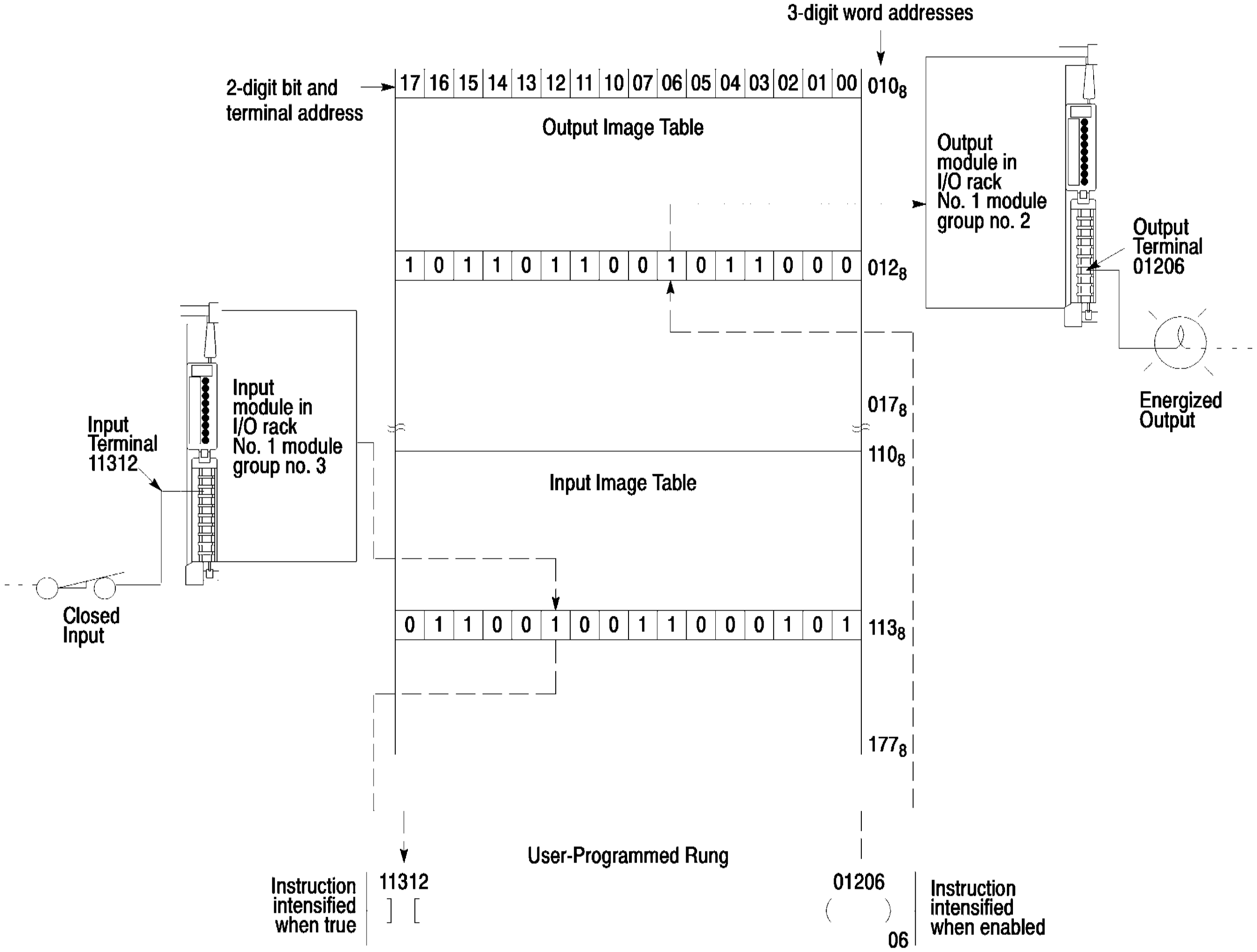
The message storage area begins after the end statement and stores the alphanumeric characters of the messages. Two characters can be stored in one word.

For a more detailed complete description of memory, refer to memory Organization and Structure of PLC-2/20 Processors (publication 1772-909).

Hardware/Program Interface

The way the processor uses input device data to turn output devices on and off is shown in Figure 1.7.

Figure 1.7
Hardware/Program Interface



17	16	15	14	13	12	11	10	07	06	05	04	03	02	01	00	
1	0	1	1	0	1	1	0	0	1	0	1	1	0	0	0	012_8
0	1	1	0	0	1	0	0	1	1	0	0	0	1	0	1	113_8

When the input device connected to terminal 11312 is closed, the input module circuitry senses a voltage. The on condition is reflected in input image table bit 11312. During the program scan, the processor examines bit 11312 for an on (1) condition. If the bit is on (1), a path of logic continuity is established and causes the rung to be true. The processor sets output image table bit 01206 on (1) and sends a logic one to turn on terminal 01206 during the next I/O scan. The output device wired to this terminal becomes energized.

When the input device wired to terminal 11312 opens, the input module senses no voltage. The off condition is reflected in input image table bit 11312. During the program scan, the processor examines bit 11312 for an on (1) condition. Since the bit is off (0), logic continuity is not established and the rung is false. the processor sets output image table bit 01206 OFF (0) and sends a logic zero to turn off terminal 01206 during the next I/O scan. The output device wired to this terminal is then de-energized.

PLC-2/20 Programmable Controller Capabilities

Although both PLC-2/20 controllers can have up to 8K (8192) words of memory and have a data table default size of 128 words, their expanded data table areas can differ in size. This different size affects the number of I/O racks and timers/counters that can be used.

PLC-2/20 Processor (cat. no. 1772-LP1)

The data table for the 1772-LP1 processor can be expanded to 256 words in 2-word increments (figure 1-4). When the industrial terminal (cat. no. 1770-T1, -T2 or -T3) and the 1771 I/O components are used, the processor can control up to 512 I/O points, or four 128 I/O racks (Table 1.A).

Table 1.A
PLC-2/20 Processor (Cat. No. 1772-LP1) Capabilities

#I/O Racks	Industrial Terminal (Cat. No. 1770-T1, -T2 or -T3)	
	Maximum I/O Points	Maximum Timers/ Counters
1	128	111
2	256	104
3	384	96
4	512	88

PLC-2/20 Processor (cat. no. 1772-LP2)

The data table for the 1772-LP2 processor can be expanded to 8064 words only when the 1770-T3 industrial terminal is used (Figure 1.5). An 8064 word data table, however, is impractical since there would be little memory available for the user program. The user can expand the data table to 256 words (word address 377_8) in 2-word increments.

From word address 400_8 on the data table must be expanded in 128-word sections. When the 1770-T3 industrial terminal and 1771 I/O components are used, the processor can control up to 896 I/O points, or seven I/O racks (Table 1.B). If the 1770-T1 or -T2 industrial terminal is used, the 1772-LP2 processor will have the same capabilities as the 1772-LP1 processor.

Table 1.B
PLC-2/20 Processor (Cat. No. 1772-LP2) Capabilities

	Max. I/O Points	
#I/O Racks	1770-T3 Industrial Terminal	1770-T1 or-T2 Industrial Terminal
1	128	128
2	256	256
3	384	384
4	512	512
5	640	
6	768	
7	896	

Compatibility

WARNING: Use only Allen-Bradley authorized programming devices to program Allen-Bradley programmable controllers. Using unauthorized programming devices may result in unexpectedoperation, possibly causing equipment damage and/or injury to personnel. The Allen-Bradley Company will not be responsible or liable for any damages, whether direct, indirect, or consequential, arising out of the use of such unauthorized programming devices.

The PLC Program Panel (cat. no. 1774-TA) with the PLC-2 Program Panel Adapter (cat. no. 1772-T4), or the PLC-2 Program Panel (cat. no. 1772-T1) can be used to program the PLC-2/20 processors; however, they will limit the capabilities of the processors in the following ways:

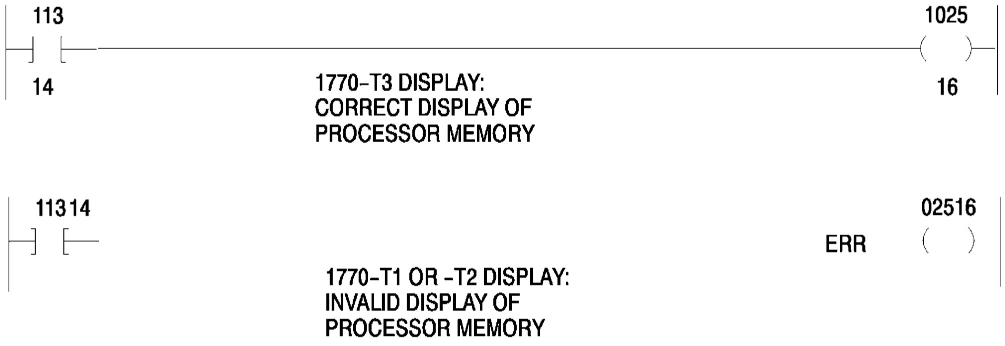
- Control of 256 I/O points
- 4K (4096) words of memory
- Only words, not bits, can be addressed in the expanded data table area.

For this reason, only the industrial terminals and 1771 I/O components will be discussed in this manual. For information on PLC-2/20 processor use with program panels and 1771 I/O components, refer to the PLC-2 user's manual (publication 1772-800).

any differences between the PLC-2/20 processors (cat. no. 1772-LP1 and -LP2) will be discussed in this manual.

WARNING: Do not use a 1770-T1 or 1‘770-T2 industrial terminal to edit or change a program or data table values in PLC-2/20 memory 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 1.8). 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 T2 terminal could result in unpredictable machine motion with possible damage to equipment and/or injury to personnel.

Figure 1.8
Correct and Invalid Displays of Processor Memory



Chapter 1
Introduction

The PLC-2/20 processor (cat. no. 1771-LP1 or -LP2) can be connected to the Allen-Bradley data highway with the Communication Adapter Module (cat. no. 1771-KA). Certain data table addresses cannot be used in the data highway communication zone or in data highway messages, depending on the series and revision level of the module.

- With a series A/rev.D or earlier module, data table addresses up to 177_8 can be used.
- With a series A/rev.E module, data table addresses up to 377_8 can be used.
- With a series B or later module, all data table addresses can be used.

Related Publications

PLC-2/20, -2/30 Programmable Controller Assembly and Installation Manual (publication 1772-807) contains necessary information on installation, assembly, maintenance and troubleshooting.

Chapter 2

Hardware Considerations

General

This section will describe those hardware features of the PLC-2/20 controller that are used when inputting or debugging the user program. For more complete information, refer to the PLC-2/20, -2/30 Programmable Controller Assembly and Installation Manual (publication 1772-807).

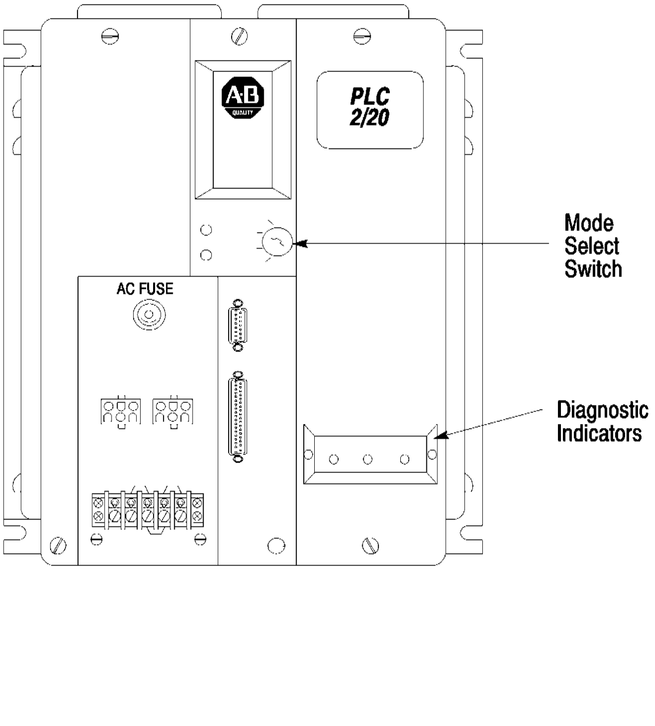
Mode Select Switch

The processor has a four-position keylock mode select switch (NO TAG) that places the processor in one of four operating modes:

- PROGRAM
- TEST
- RUN
- RUN/PROGRAM

The key can be removed from the processor in any of the four keyswitch positions.

Figure 2.1
PLC-2/20 Processor



PROG Position

This switch position places the processor in the program mode. User program instructions can only be entered in this mode. They can be entered from either the industrial terminal, the Digital Cassette Recorder (cat. no. 1770-SA), or the Data Cartridge Recorder (cat. no. 1770-SB). All outputs are disabled in this switch position.

TEST Position

This switch position places the processor in the test mode. The user program is tested under simulated operating conditions without actually energizing the user's output devices. All outputs are disabled in this switch position.

RUN Position

This switch position places the processor in the run mode. The user program will be executed and outputs are controlled by the program. Changes to the user program are not permitted in this switch position.

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 and changes to data table values are permitted in this switch position.

Processor Diagnostic Indicators

One the front of the processor are diagnostic indicators that indicate fault conditions (NO TAG). The user should become familiar with these indicators.

MEMORY FAULT Indicator

This red indicator illuminates when a parity error occurs in the data retrieved from memory. Changing the mode select switch to the PROG position or cycling line power may reset this fault condition. Reloading the program may also reset the fault.

STANDBY LOW Indicator

When the batteries for RAM memory backup are low, this red indicator flashes on and off. when this occurs, the batteries will continue to support memory for about one week.

The battery low bit, bit 02700, will cycle on and off when a battery voltage low condition is detected and the mode select switch is not in the PROG position. Programming techniques can be used to examine this bit and to control some type of alerting device when a battery low condition exists.

When CORE memory is used in a PLC-2/20 processor, battery backup is not needed since the memory is non-volatile. Thus there is no STANDBY LOW indicator on the CORE memory module.

PROCESSOR FAULT Indicator

This red indicator illuminates when the logic circuits controlling the processor scan fail.

RUN Indicator

This green indicator illuminates when the processor is in the run or run/program mode. It also indicates that outputs are being controlled by the user program.

DC ON Indicator

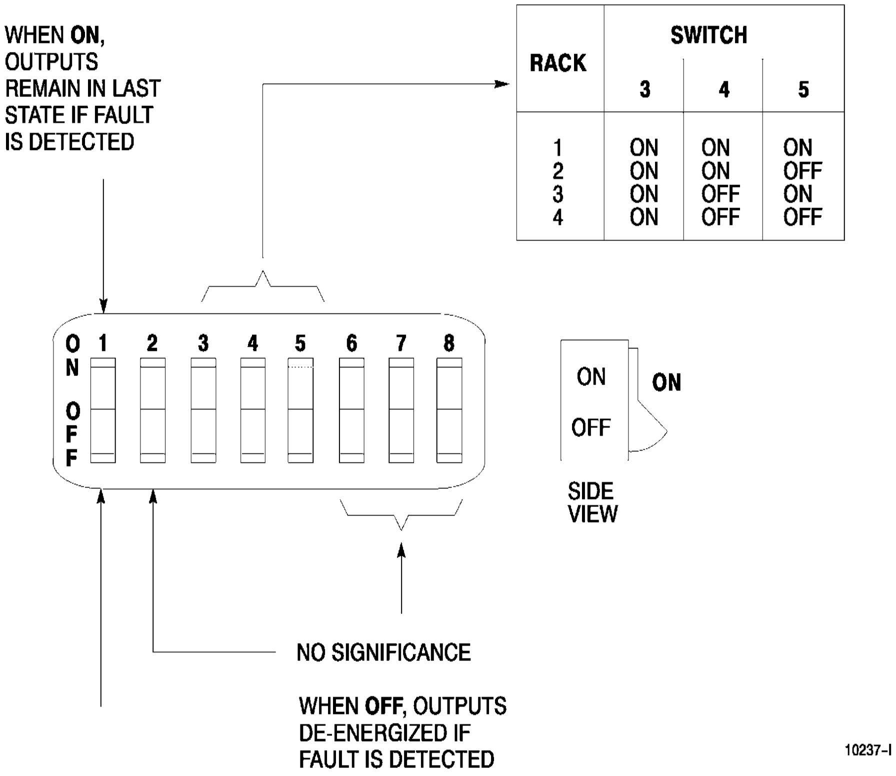
The red DC ON indicator illuminates when the AC input voltage and DC output voltages from the power supply are within the proper tolerances.

Switch Group Assembly

Located on the backplane of each I/O rack is a switch group assembly. It is used to control output behavior when a fault occurs and to identify the rack number for local systems. The switch positions for each rack must be set by the user.

Up to four 128 I/O racks can be used with the PLC-2/20 processor (cat. no. 1772-LP1). The corresponding switch settings are shown in Figure 2.2.

Figure 2.2
1771 Switch Group Settings (4 Racks)

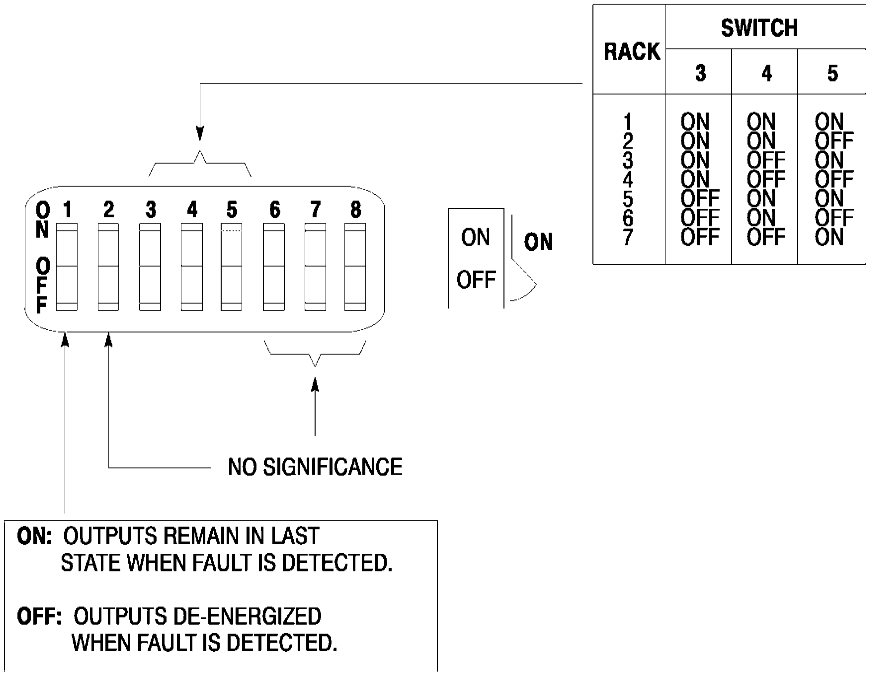


RACK	SWITCH		
	3	4	5
1	ON	ON	ON
2	ON	ON	OFF
3	ON	OFF	ON
4	ON	OFF	OFF

WHEN ON, OUTPUTS REMAIN IN LAST STATE IF FAULT IS DETECTED
O N
O F F
1
2
3
4
5
6
7
8
ON
OFF
ON
SIDE VIEW
NO SIGNIFICANCE
WHEN OFF, OUTPUTS DE-ENERGIZED IF FAULT IS DETECTED
10237-I

Up to seven 128 I/O racks can be used with the PLC-2/20 processor (cat. no. 1772-LP2). The corresponding switch settings are shown in Figure 2.3.

Figure 2.3
1771 Switch Group Settings (7 Racks)



RACK	SWITCH		
	3	4	5
1	ON	ON	ON
2	ON	ON	OFF
3	ON	OFF	ON
4	ON	OFF	OFF
5	OFF	ON	ON
6	OFF	ON	OFF
7	OFF	OFF	ON

ON: OUTPUTS REMAIN IN LAST STATE WHEN FAULT IS DETECTED.

OFF: OUTPUTS DE-ENERGIZED WHEN FAULT IS DETECTED.

WARNING: For local and remote systems, switch no. 1 of the 1771-I/O rack should be set 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. Damage to equipment and/or personal injury could result.

In addition, for remote systems, the switches on the 1772-SD or 1772-SD2 Remote I/O Scanner/Distribution Panel and the 1772-AS Remote I/O Adapter Module Assembly must be set. Refer to Publications 1772-910, 1772-929 and 1771-938 respectively for information on their switch settings.

For remote systems, the last state switch (switch 1) on the 1771 I/O rack must be set. In addition, the switches on the 172-SD or 1772-SD2 Remote I/O Scanner/Distribution Panel and the 1771-AS Remote I/O Adapter Module Assembly must be set. Refer to Publications 1772-910, 1772-929 and 1771-938 respectively for information on their switch settings.

Industrial Terminal

The industrial terminal system (cat. no 1770-T1, -T2 or -T3) is used to program the PLC-2/20 processor.

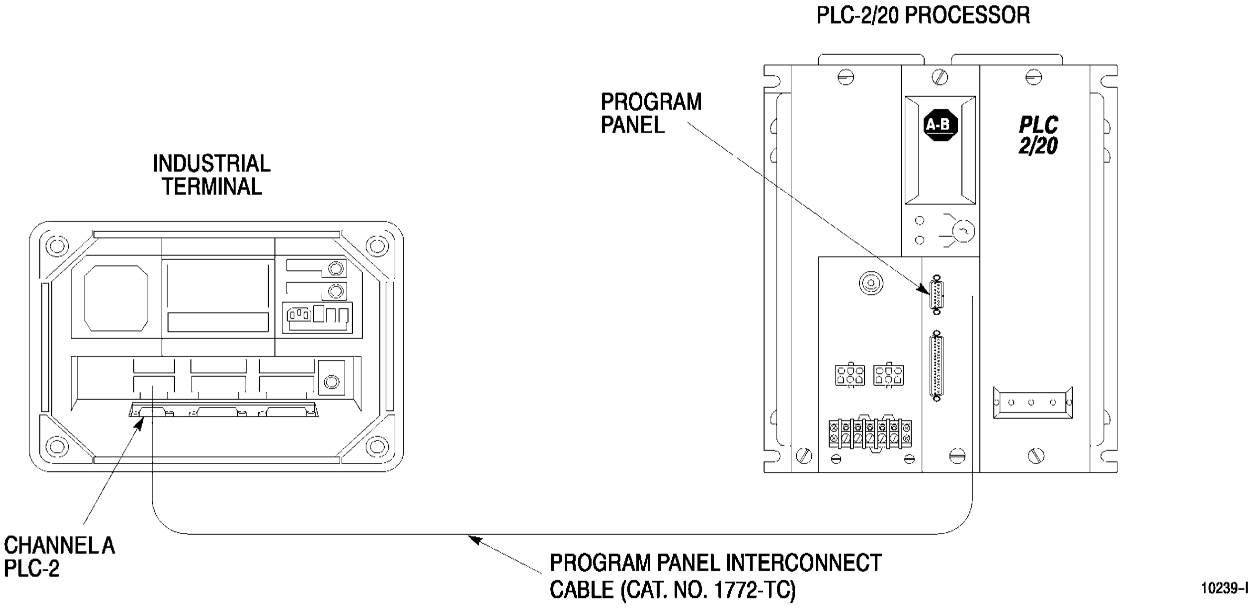
Perform the following steps to connect the industrial terminal to the processor:

1. Plug the AC power cord of the industrial terminal into a grounded AC outlet.

2. Connect one end of the PLC-2 Program Panel Interconnect Cable (cat. No. 1772-TC) to channel A at the back of the industrial terminal (Figure 2.4).

3. Connect the other end of the cable to the socket labeled PROGRAM PANEL on the front of the processor (Figure 2.4).

Figure 2.4
PLC-2/20 Processor/Industrial Terminal Connection Diagram



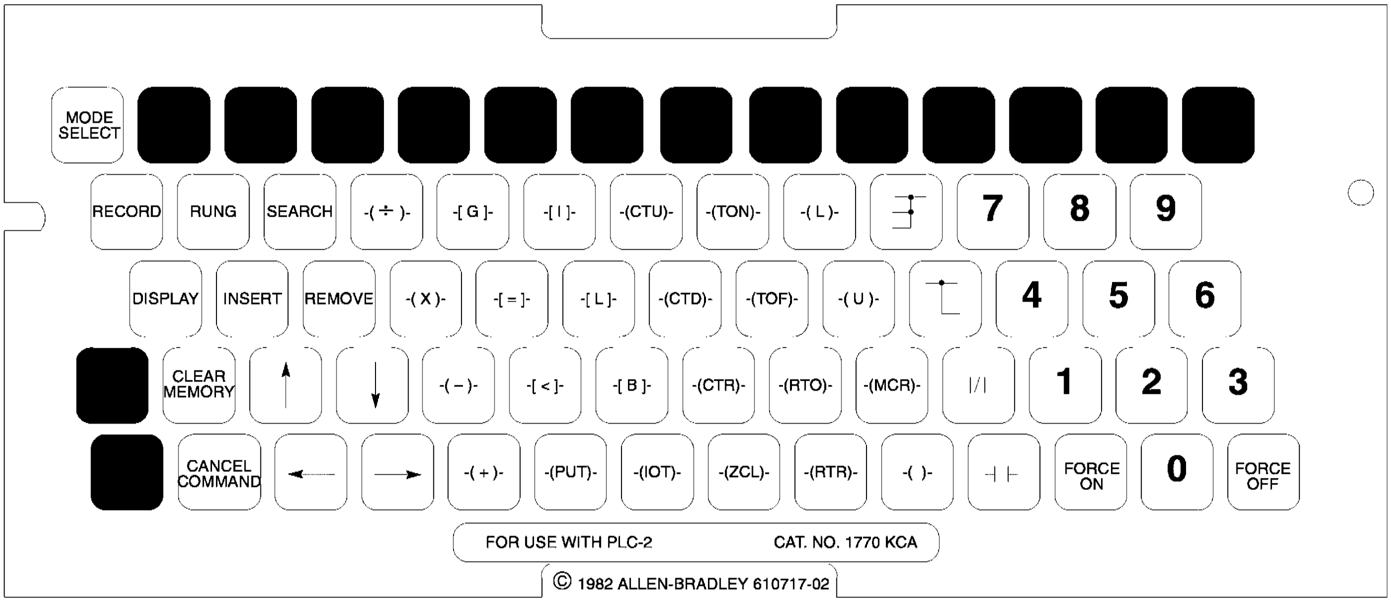
4. Install the appropriate keytop overlay on the appropriate keyboard (figures 2-5 and 2-6).

5. Turn power switch on the front of the industrial terminal to the one position. mode select display will appear.

6. Press [1][1] to select PLC-2 mode and begin programming.

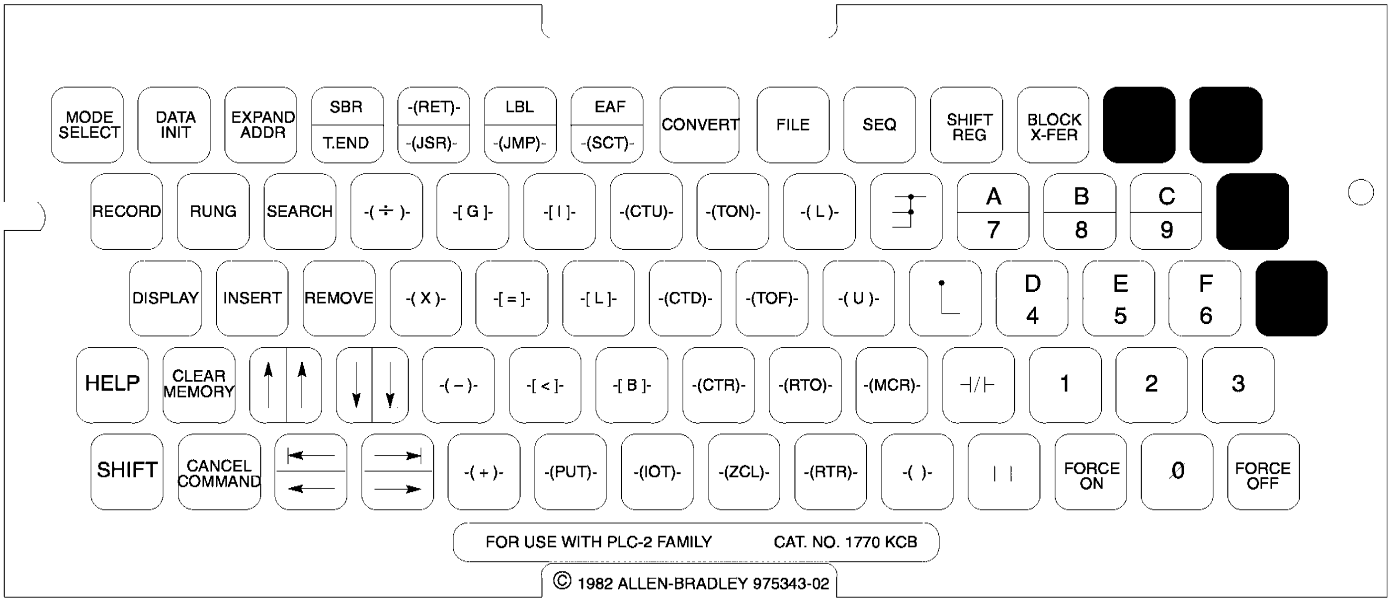
For setup information on the industrial terminals, refer to the Industrial Terminal System user's Manual, (publication 1770-805).

Figure 2.5
PLC-2 Keytop Overlay for Industrial Terminal (cat. no. 1770-T1 or T2)



10240–I

Figure 2.6
PLC-2 Family Keytop Overlay for Industrial Terminal (cat. no. 1770-T3)



102241–I

Compatibility

The following shows compatibility of existing processor hardware

Processor Control Module/Interface

Table 2.A shows the compatibility and capabilities of various processor control modules and processor interface modules.

Table 2.A
Processor Control Module/Interface Module Combinations

Control	Processor Interface Module		
Module	1772-LC	1772-LF	1772-LH
1772-LB	1772-LR	X	X
1772-LE	X	1772-PL1	1772-LP1
1772-LG	X	X	1772-LP3
1772-LI	X	[1]	1772-LP2

[1] This combination gives 1772-LP2 processor capability but can only handle 4 I/O racks

Processor/Memory Module

Table 2.B shows the compatibility of several processor and memory module combinations.

Table 2.B
Processor/Memory Module Combinations

	Memory Module			
Processor	1772-MG	1772-ME[1]	1772-M8[2]	1772-M16[2]
1772-LR	2K	.5 to 4K	X	X
1772-LP1	2K	.5 to 8K	8K	X
1772-LP2	2K	.5 to 8K	8K	X
1772-LP3	2K	.5 to 8K	8K	16K

[1] Maximum of 4K addressable with PLC-2 program panel.
[2] 1772-M8 and -M16 memory modules require 1772-Pi series C power supply.

Memory Segment/Memory Module

Table 2.C shows the compatibility of memory segment and memory module combinations.

Table 2.C
Memory Segment/Memory Module Combinations

Memory	Memory Segment		
Module	1772-MR	1772-MS	1772-MT
1772-MG	.5 to 2K	X	X
1772-ME[1]	.5 to 2K	2 to 8K	.5 to 2K

[1] Maximum of 4K addressable with PLC-2 program panel.

Power Supply Compatibility

The cat. no. 1772-P1 and cat. no 1772-P2 power supplies have been produced as series A, B and C. The following summarizes the functional characteristics of each series.

Series A - The original production level. It will operate only PLC-2 family processors without 1771 I/O, remote I/O or core memory.

Series B - This series provides power sufficient for use with all PLC-2 family processors and local or remote I/O. It will not support core memory.

Series C - Series B power increased sufficiently to permit use of core memory with any PLC-2 except 1772-LR. However, when core memory is used, the processor supply cannot be used to power any I/O.

Chapter 3

Relay-type Instructions

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

Examine Instructions

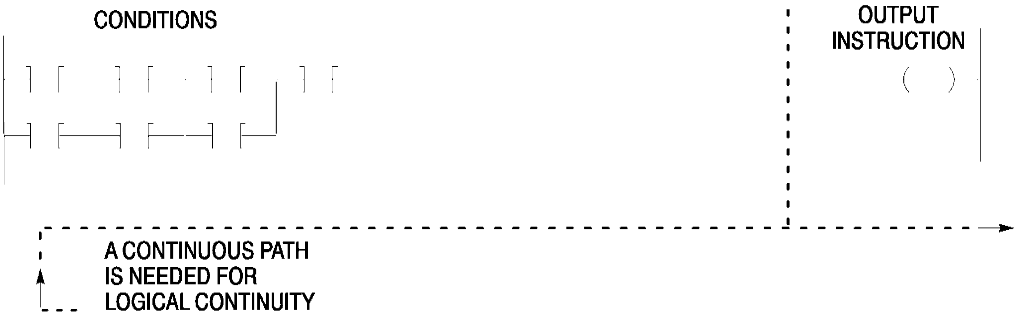
There are two examine instructions:

- Examine On
- Examine Off

They command the processor to check the on/off status of a specific bit address in memory. A one or zero stored at the bit address may represent the actual on or off status of a single input or output device.

Examine instructions are programmed in the condition area of the ladder diagram rung (Figure 3.1). As condition instructions, their on or off states determine the true or false condition of the rung. Any bit in the data table, excluding the processor work areas, can be addressed by an examine instruction. A single bit can be examined several times within the same rung or program.

Figure 3.1
Areas of the Ladder-Diagram Rung



Examine On Instruction

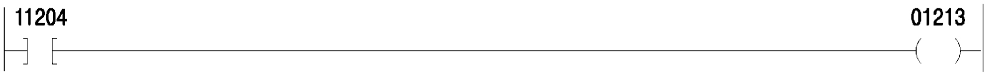
The Examine On instruction tells the processor to check the status of the addressed memory bit for an on (one) condition. When addressing the I/O image table, this instruction can examine a single input or output bit for an on voltage state.

The Examine On instruction is either true or false:

- True – The addressed memory bit is one, meaning that the corresponding I/o device or bit is on.
- False – The addressed memory bit is zero, meaning that the corresponding I/O device or bit is off.

When using the Examine On instruction to address an input device, the conventional normally–open or normally–closed distinctions are not made. The Examine On instruction only checks for an on or energized status of a device or bit (Figure 3.2).

Figure 3.2
Examine On Instruction



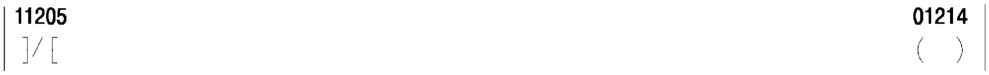
Examine Off Instruction

The Examine Off instruction is the logical opposite of the Examine On instruction. It tells the processor to check the status of the addressed memory bit for an off condition. When addressing the I/O image table, this instruction can examine a single input or output bit for an off voltage state (Figure 3.3).

The Examine Off instruction is either true or false:

- True – The addressed memory bit is zero, meaning that the corresponding I/O device or bit is off.
- False – The addressed memory bit is one, meaning that the corresponding I/O device or bit is on.

Figure 3.3
Examine Off Instruction



Output Instructions

The output instructions set an addressed memory bit to one (on) or reset it to zero (off). An output image table bit, as one or zero, can cause an output device to be turned on or off.

Output instructions are programmed at the end of the ladder–diagram rungs (Figure 3.1). Only one output instruction can be programmed on each rung. An instruction in this position of the rung is executed only if the rung conditions preceding the instruction are logically true.

These output instructions are:

- Output Energize
- Output Latch
- Output Unlatch

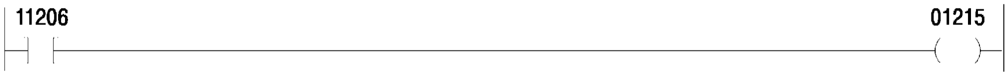
These instructions are used to set memory bits on or off in any area of the data table, excluding the processor work areas and the input image table.

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.

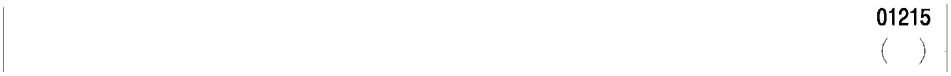
The Output Energize instruction tells the processor to turn the addressed memory bit off when rung conditions go false (Figure 3.4).

Figure 3.4
Output Energize Instruction



CAUTION: 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 desirable in output device programming.

Figure 3.5
Unconditional Output Energize Instruction



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 is somewhat similar to the Output Energize instruction. The Output Latch instruction tells the processor to set an addressed memory bit on when rung conditions are true. Unlike the Output Energize instruction, the Output Latch instruction is “retentive.” This means that once conditions go false, the latched bit remains on until changed by an Output Unlatch instruction. If power is lost and battery backup for RAM memory is maintained, all latched bits will remain on.

The Output Unlatch instruction 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 conditions for the Output Latch instruction go 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 conditions go false, the unlatched bit remains off until changed by an Output Latch instruction.

Figure 3.6
Latch/Unlatch Instructions

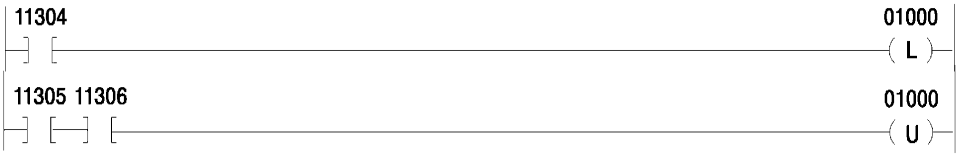
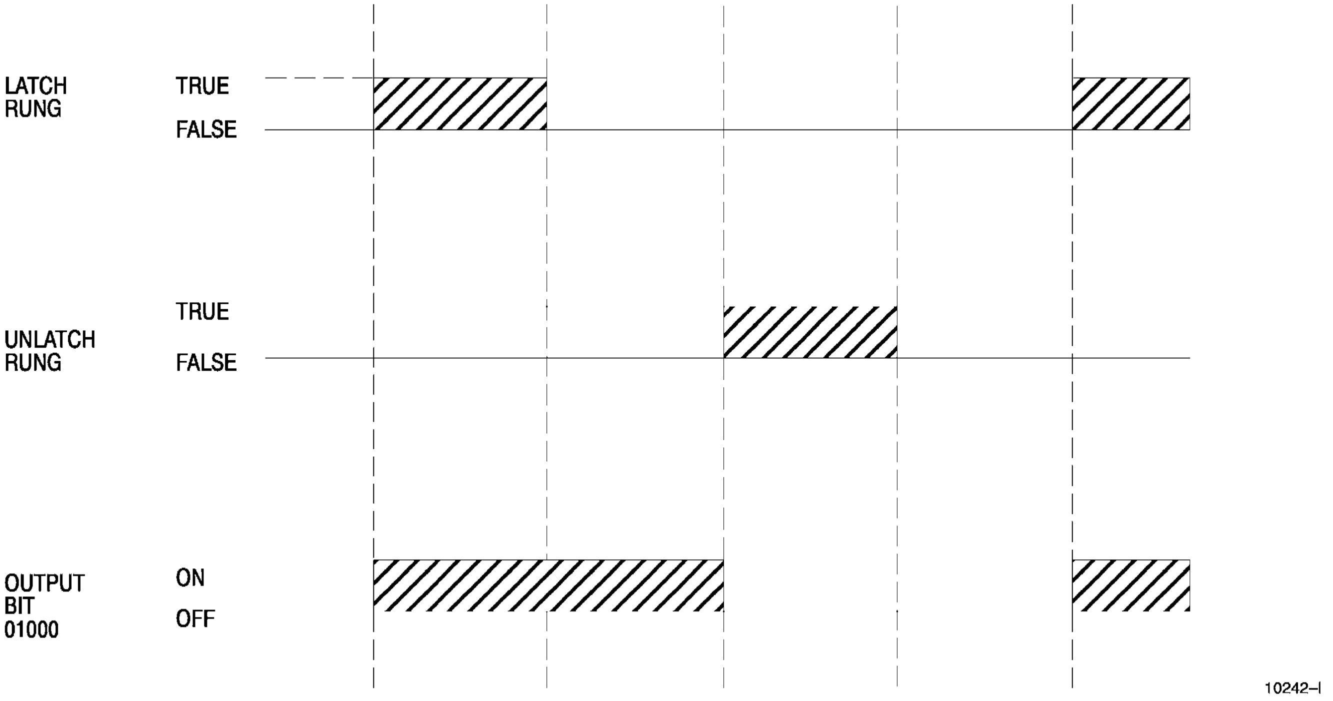


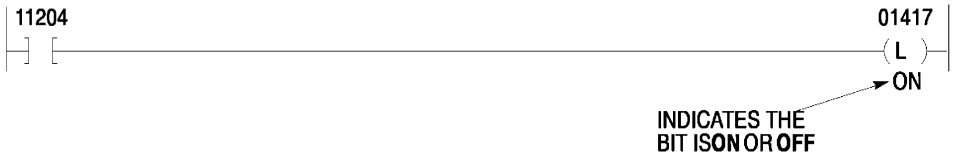
Figure 3.7
Latch/Unlatch Timing Diagram



When the mode select switch is changed from the RUN or RUN/PROG position, the last true Latch or Unlatch instruction continues to control the addressed memory bit, but disables the output device. When the mode select switch is turned back to RUN or RUN/PROG, an output device connected to the terminal corresponding to the latched bit will be energized.

The bit controlled by the Output Latch instruction, when entered, is automatically set off. It can be initially preset on by entering the number [1] immediately after the bit address. The on or off condition will be displayed below the Output Latch instruction when the processor is in the prog 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 latched on, will immediately be energized, regardless of rung conditions.

Figure 3.8
Latch Indication



WARNING: Do not preset a bit controlled by a Latch or Unlatch instruction on if it controls potentially hazardous machine motion. If it is preset on, the output device controlled by the memory bit is energized immediately when the mode select switch is turned to the RUN or RUN/PROG position. Hazardous machine operation could damage equipment and/or personal injury could result.

CAUTION: 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.

Branch Instructions

The branch instructions allow more than one combination of input conditions to energize an output device (Figure 3.9).

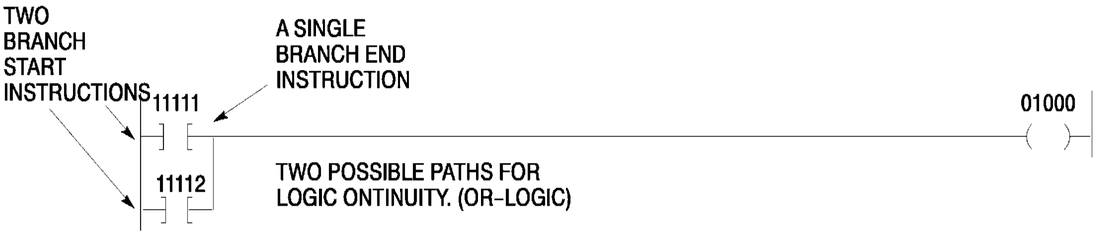
There are two branch instructions:

- Branch Start
- Branch End

Branch start – This instruction begins each parallel logic branch of a rung. The Branch Start is programmed immediately before the first instruction of each parallel logic path.

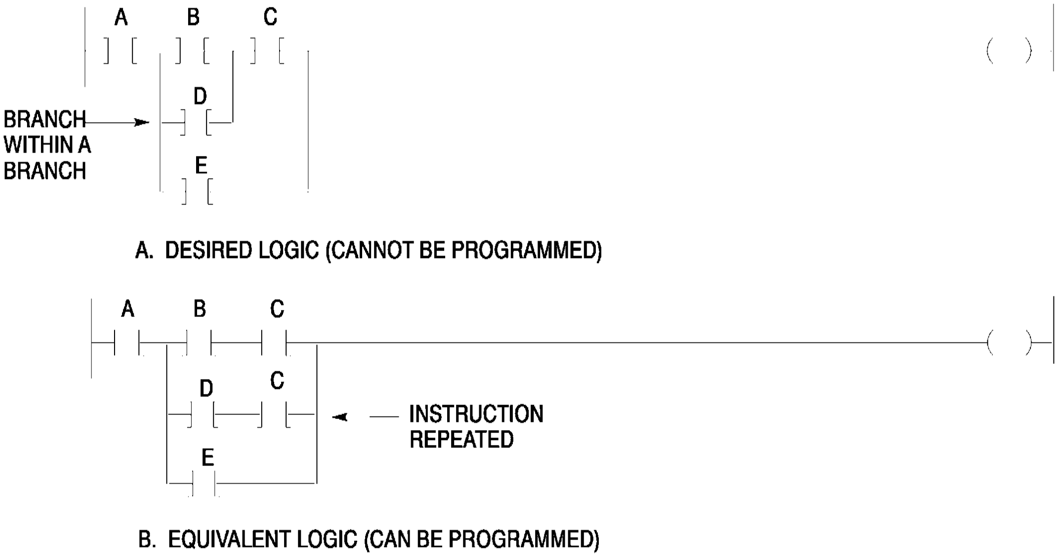
Branch End – This instruction completes a set parallel branches. The Branch End is entered after the last instruction of the last branch to end a set of parallel branches.

Figure 3.9
Branching Instructions


TWO BRANCH START INSTRUCTIONS
A SINGLE BRANCH END INSTRUCTION
11111
11112
01000
TWO POSSIBLE PATHS FOR LOGIC ONTINUITY. (OR-LOGIC)

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).

Figure 3.10
Nested Programming vs. Proper Programming


A
B
C
D
E
BRANCH WITHIN A BRANCH
A. DESIRED LOGIC (CANNOT BE PROGRAMMED)
A
B
C
D
C
E
INSTRUCTION REPEATED
B. EQUIVALENT LOGIC (CAN BE PROGRAMMED)

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, 01000, is displayed with a reverse–video character cursor positioned at the first digit. This cursor

indicates where information is needed and moves to the next digits as information is entered. When all information is entered, the instruction stops blinking and remains intensified until the next instruction is entered.

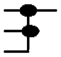
Rely–type instructions are entered and displayed in slightly different formats depending on the industrial terminal and processor being used. Refer to Table 3.A at the end of this section for a description of these instructions.

The industrial terminal (cat. no. 1770–T1 or –T2) displays the complete 5–digit bit address above the instruction for both PLC–2/20 processors. On the other hand, the industrial terminal (cat. no. 1770–T3) displays the word address above the instruction and the bit number below it.

With the 1772–LP2 processor and the 1770–T3 industrial terminal, 6 or 7–digit bit addresses can be entered provided the data table has been expanded to a 4 or 5–digit word address. To enter a 6 or 7–digit address, the EXPAND ADDR key is required. It is pressed after the instruction is entered and before the address is entered. The EXPAND ADDR key will display either a 6 or 7–digit default bit address, 001000 or 0001000, depending on the data table size. When a 7–digit bit address is displayed and a 6–digit address is required, a leading zero must be entered before the bit address.

Table 3.A
Relay–type Instruction

Keytop Symbol	Instruction Name	1770-T1 or -T2 Display	1770-T3 Display	Description
–| |–	Examine On	XXXXX -] [-		XXXXX-bit address. User can assign any bit address in the data table, excluding processor work areas. When instruction is true, the addressed memory bit is on.
			XXX -| |- XX	Same as above. 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 and enter the bit address using a leading zero if necessary.
-|/|-	Examine Off	XXXXX -]/[-		XXXXX-bit address. Can assign any bit address in the data table excluding processor work areas. When instruction is true, the addressed memory bit is off.
			XXX -|/|- XX	Same as above. 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 and enter the bit address using a leading zero if necessary.
-( )-	Output Energize	XXXXX -( )-		XXXXX-bit address. Can assign any bit address in the data table excluding processor work areas and the input image table. When rung is true, the addressed memory bit is set on. If the bit controls an output device, that output device will be on.
			XXX -( )- XX	Same as above. 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 and enter the bit address using a leading zero if necessary.

Keytop Symbol	Instruction Name	1770-T1 or -T2 Display	1770-T3 Display	Description
-(L)-	Output Latch	XXXXX -(L)- On or Off		XXXXX-bit address. Can assign any bit address in the data table excluding processor work areas and the input image table. When rung is true, the addressed bit is latched on and remains latched on until it is unlatched. The bit controlled by the Output Latch instruction is initially off when entered and is indicated below the instruction. It can be preset on by pressing [1] after entering the bit address. Status will be indicated below the instruction in the prog mode.
			XXX -(L)- On XX or Off	Same as above. 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 and enter the bit address using a leading zero if necessary.
-(U)-	Output Unlatch	XXXXX -(U)-		XXXXX-bit address. Address the same bit that was latched on. When the rung is true, the addressed bit is unlatched. If the bit controls an output device, that device is also unlatched.
			XXX -(U)- On XX or Off	Same as above. 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 and enter the bit address using a leading zero if necessary. The bit controlled by the Output Latch instruction is initially off. It can be preset on by pressing [1] after entering the bit address. Status will be indicated below the instruction in the prog mode.
	Branch Start	┼	┌	Begins a parallel logic path. Entered at the beginning of each parallel path. Always ended with a Branch End instruction.
	Branch End	┼	┘	Ends a group of parallel logic paths.

Chapter 4

Timer and Counter Instructions

General

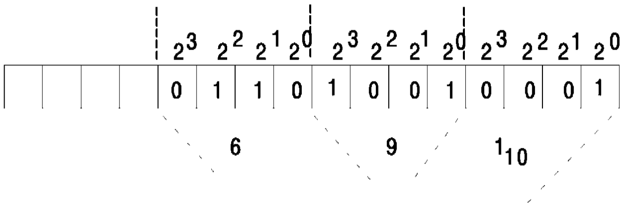
Timer and counter instructions are output instructions internal to the processor. They provide many of the capabilities available with timing relays and solid state timing/counting devices. Usually conditioned by examine instructions, timers and counters keep track of timed intervals or counted events according to the logic continuity of the rung.

Each timer or counter instruction has two 3–digit values associated with it, and thus requires two words of data table memory. These 3–digit values are:

- Accumulated (AC) Value – Stored in the accumulated value area of the data table. 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 the preset value area of the data table, always 100_8 words greater than its corresponding AC value. This value is entered into memory by the user. 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 an output device on or off.

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" "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 chapter 13.

Figure 4.1
BCD Format



10243-I

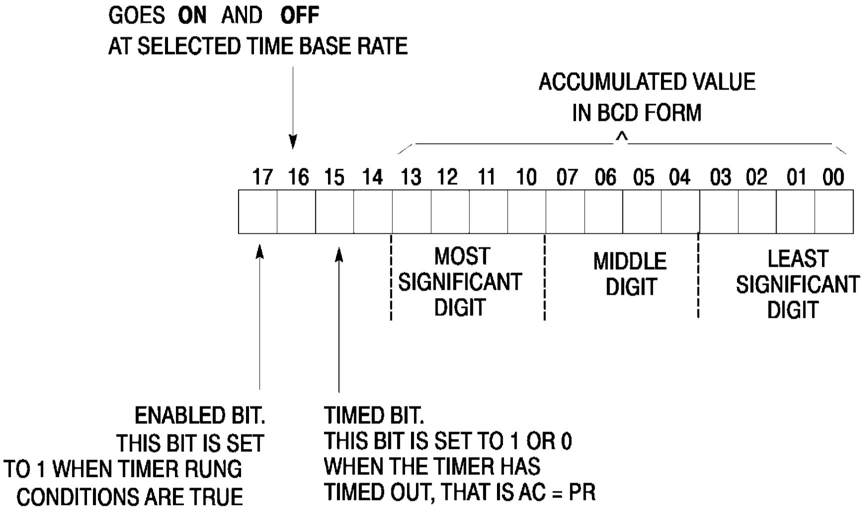
The remaining 4 bits in a word (bits 14–17) are not used to form a BCD number. In the accumulated value word, they are used as status bits. In the preset value word, they are not used and are available for internal storage. For 10 ms timers, however, bits 14–17 of the preset value word are used for internal timing functions. Thus, they cannot be used for storage.

See section titled Storage Assignments and Recommendations in chapter 8 for additional restrictions on the use of storage.

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 (Figure 4.2).

Figure 4.2
Timer Accumulated Value Word



10244-I

The three types of timers available with the PLC–2/20 controller are:

- Timer On–Delay
- Timer Off–Delay
- Retentive Timer

All three timers differ in the way they set and reset status bits, respond to rung logic continuity and reset the accumulated value. With each timer, the programmer must select one of the following time bases:

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

Bit 16 of the timer accumulated value word reflects the time base. It will go on and off at the selected time base rate acting as a "pulse train" (Figure 4.2). with 10 ms timers, however, bit 16 of the accumulated value word is used internally by the timer. Thus, bit 16 of 10 ms timers must never be used in the program.

Timer On–Delay Instruction

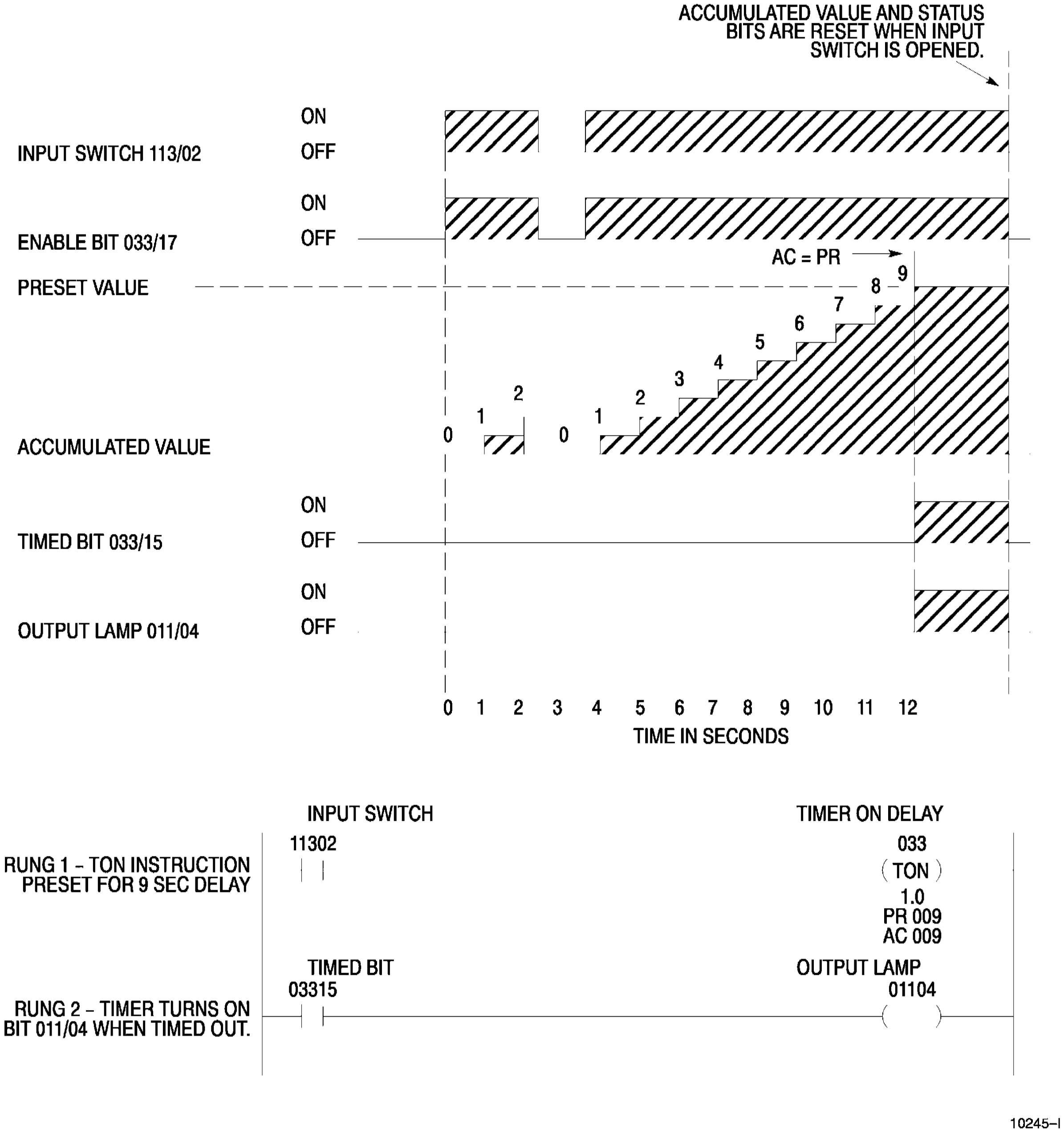
The Timer On–Delay instruction (TON) can be used to turn a device on or off once an interval is timed out (Figure 4.3).

When rung conditions for a Timer On–Delay instruction become true, the timer begins to count time–base intervals (Figure 4.3). As long as conditions remain true, it increments its accumulated value word for each counted 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 may then be used to control an output device.

Bit 17 of the accumulated value word is termed the "enabled" bit. It is set on whenever the rung conditions are true and the timer is enabled.

Whenever the rung conditions for the TON instruction go false, the accumulated value is reset to 000 and bits 15 and 17 of that word are reset to zero. The accumulated value and status bits are also reset when the mode select switch is turned to the PROGRAM position or when there is a loss of power to the system.

Figure 4.3
Timer-On Delay Timing Diagram for a Preset Value of 9 Seconds



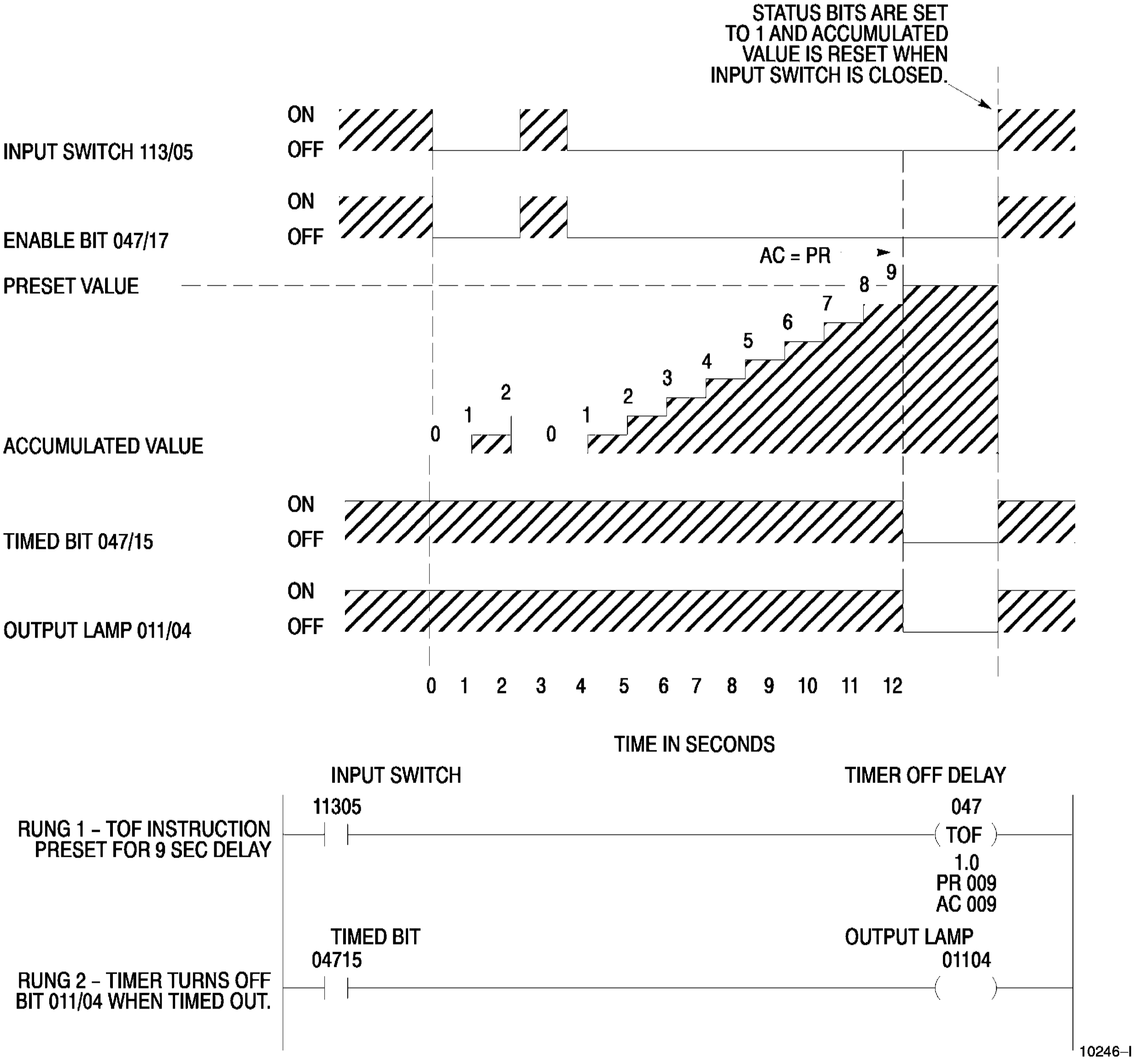
Timer Off-Delay Instruction

The Timer Off–Delay instruction (TOF) can be used to turn a device on or off after a timed interval (Figure 4.4). Like the other timer instructions, the TOF instruction counts time–base intervals and stores this count in its accumulated value. The TOF instruction, however, varies from the other instructions in significant ways.

The Timer Off–Delay instruction begins to time an interval as soon as its rung conditions o false (Figure 4.4). As long as its rung conditions remain false, the TOF continues to time, until the accumulated value equals the preset value. When the TOF times out, bit 15 is set to zero (off). As the rung conditions go true, bit 15 is set on and the accumulated value is reset to 000.

Bit 17, the enabled bit, is controlled by the logic continuity of the rung. When the rung is true, bit 17 is set to one (on); when it is false, bit 17 is set to zero (off).

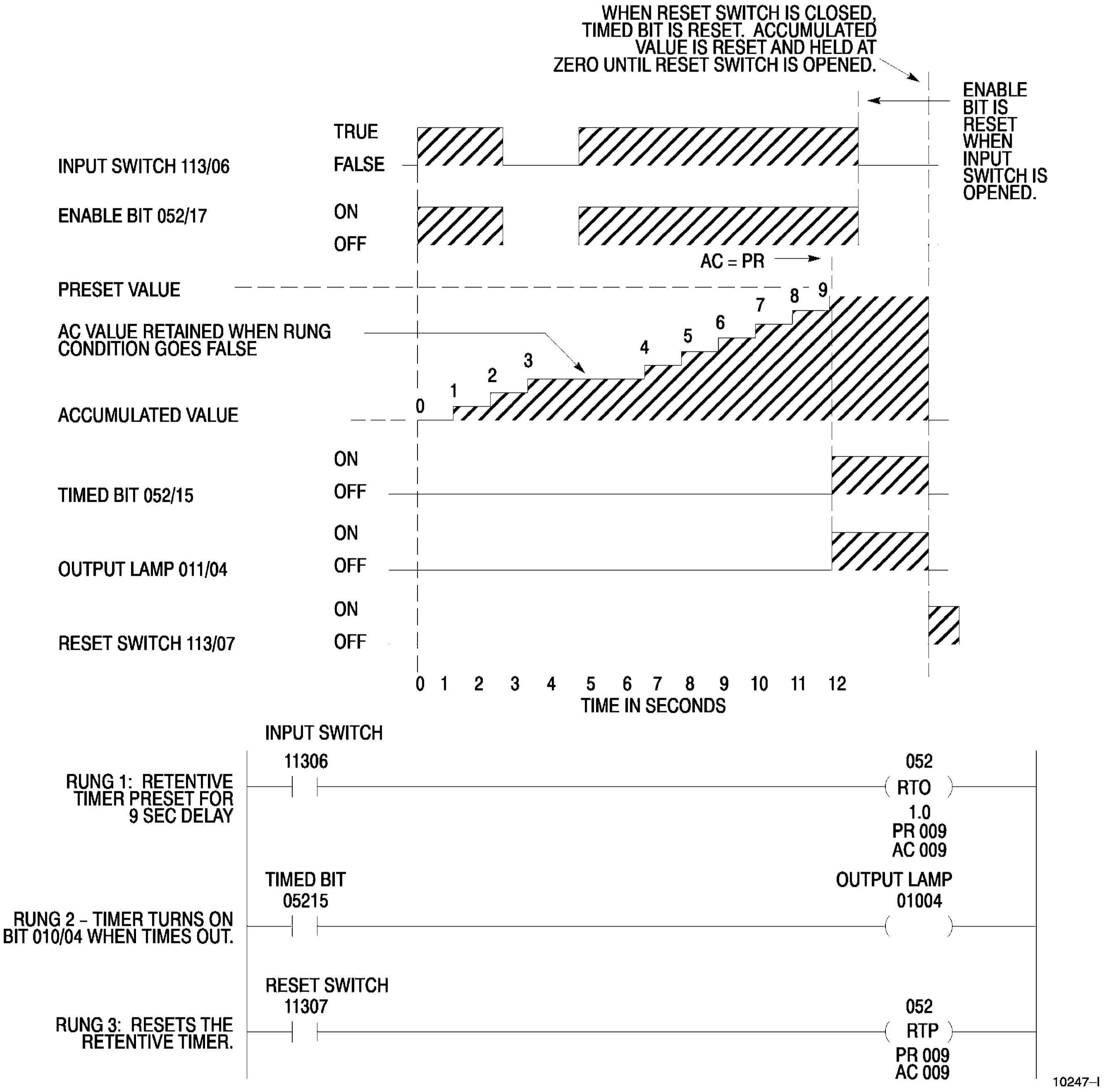
Figure 4.4
Timer–Off Delay Timing Diagram for a Preset Value of 9 Seconds



Retentive timer Instruction

The Retentive Timer instruction (RTO), much like the TON instruction, is typically used to turn a device on or off once a programmed preset value is reached (Figure 4.5).

Figure 4.5
Retentive Timer with Retentive Timer Reset Timing Diagram


WHEN RESET SWITCH IS CLOSED, TIMED BIT IS RESET. ACCUMULATED VALUE IS RESET AND HELD AT ZERO UNTIL RESET SWITCH IS OPENED.
ENABLE BIT IS RESET WHEN INPUT SWITCH IS OPENED.
TRUE
FALSE
INPUT SWITCH 113/06
ENABLE BIT 052/17
ON
OFF
AC = PR
PRESET VALUE
AC VALUE RETAINED WHEN RUNG CONDITION GOES FALSE
0
1
2
3
4
5
6
7
8
9
ACCUMULATED VALUE
ON
OFF
TIMED BIT 052/15
ON
OFF
OUTPUT LAMP 011/04
ON
OFF
RESET SWITCH 113/07
0 1 2 3 4 5 6 7 8 9 10 11 12
TIME IN SECONDS
INPUT SWITCH
11306
RUNG 1: RETENTIVE TIMER PRESET FOR 9 SEC DELAY
052
RTO
1.0
PR 009
AC 009
TIMED BIT
05215
OUTPUT LAMP
01004
RUNG 2 – TIMER TURNS ON BIT 010/04 WHEN TIMES OUT.
RESET SWITCH
11307
RUNG 3: RESETS THE RETENTIVE TIMER.
052
RTP
PR 009
AC 009
10247–I

Unlike the Timer On–Delay instruction, the Retentive Timer instruction retains its accumulated value when any of the following conditions occur:

- Rung conditions go false
- The mode select switch is changed to the PROGRAM position
- A power outage occurs provided memory backup power is maintained for RAM memory

When rung conditions go true, the enabled bit (bit 17) is set on and the timer starts counting time base intervals. 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 (Figure 4.5).

By retaining its accumulated value, the RTO instruction measures the cumulative period during which rung conditions are true. Because this timer retains its accumulated value, it must be reset by a separate instruction, the Retentive Timer Reset (RTR) instruction.

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 (Figure 4.5). When rung conditions go true, the RTR instruction resets the AC value and status bits of the RTO instruction to zero.

Timer Accuracy for 10 ms Timers

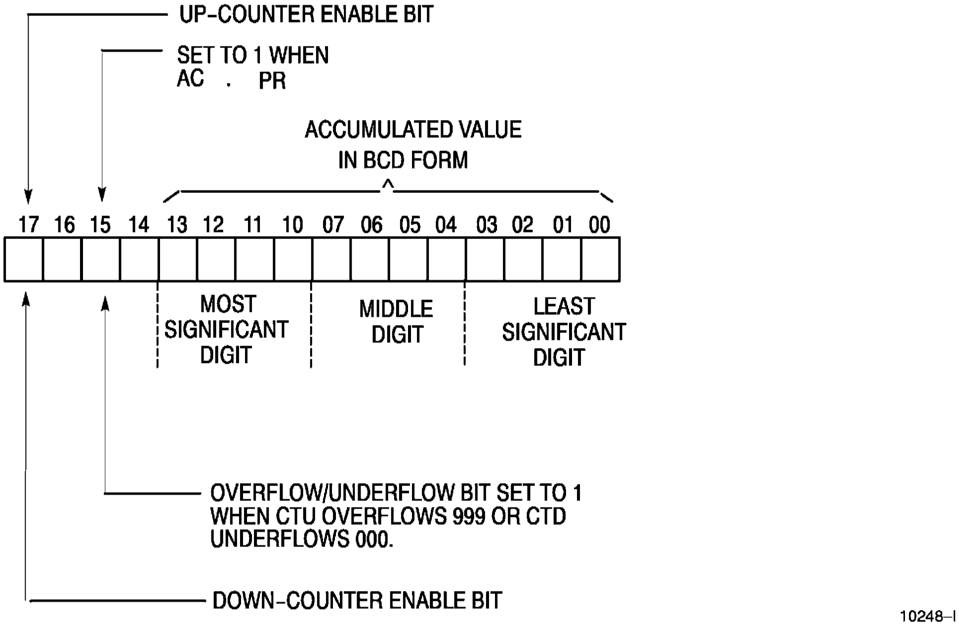
The accuracy of a 10 ms timer is related to nominal scan time. When scan times are 9 ms or less, the 10 ms timer is accurate to plus or minus one time base (±10 ms). When scan time is greater than 9 ms, accuracy to ±10 ms can be achieved through special programming techniques described in publication 1772– 702, Programming 0.01 Second Timers with the Mini–PLC–2 Controller.

Counter Instructions

Two types of counters are available with the PLC–2/20 controller: an Up–Counter (CTU) and a 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 (Figure 4.6).

Figure 4.6
Counter Accumulated Value Word



- Bit 14 is the overflow/underflow bit. it is set to one when the AC value of the CTU exceeds 999 or the AC value of the CTD 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 enabled bit for a CTD instruction. it is set on when rung conditions are true.

- Bit 17 is the enabled bit for a CTU instruction. It is set on when rung conditions are true.

Counter instructions differ from timer instructions in that they have no time base. They count one event each false–to–true transition of the rung.

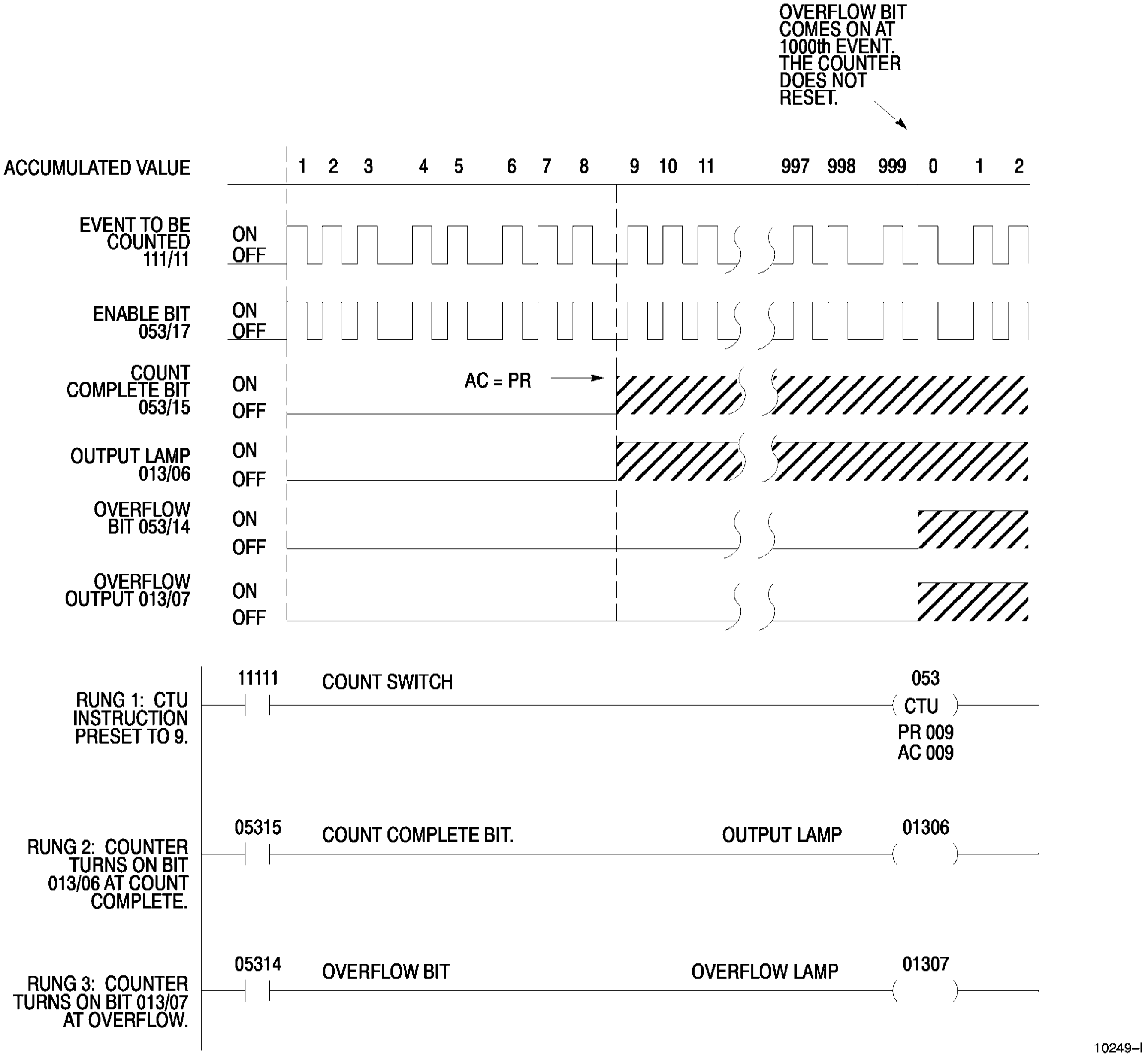
Up-Counter Instruction

The Up–Counter (CTU) instruction increments its accumulated value for each false–to–true transition of rung conditions. Because only the false–to–true transition causes a count to be made, rung conditions must go from true to false and back to true before the next count is registered. The CTU instruction retains its accumulated value when:

- The mode select switch is changed to the PROGRAM position
- The rung conditions go false
- A power outage occurs provided memory backup power is maintained.

As shown in Figure 4.7, each time the CTU rung goes true, bit 17, the enabled bit, is set on. When the accumulated value reaches the preset value, bit 15 is set on. Unlike a timer, the CTU instruction continues to increment its accumulated value after the preset value has been reached. If the accumulated value goes above 999, bit 14 is set on to indicate an overflow condition and the CTU continues up–counting from 000. Bit 14 can be examined to cascade counters for counts greater than 999 (section title Cascading Timers or Counters).

Figure 4.7
Up-Counter Diagram and Programming for Preset = 9

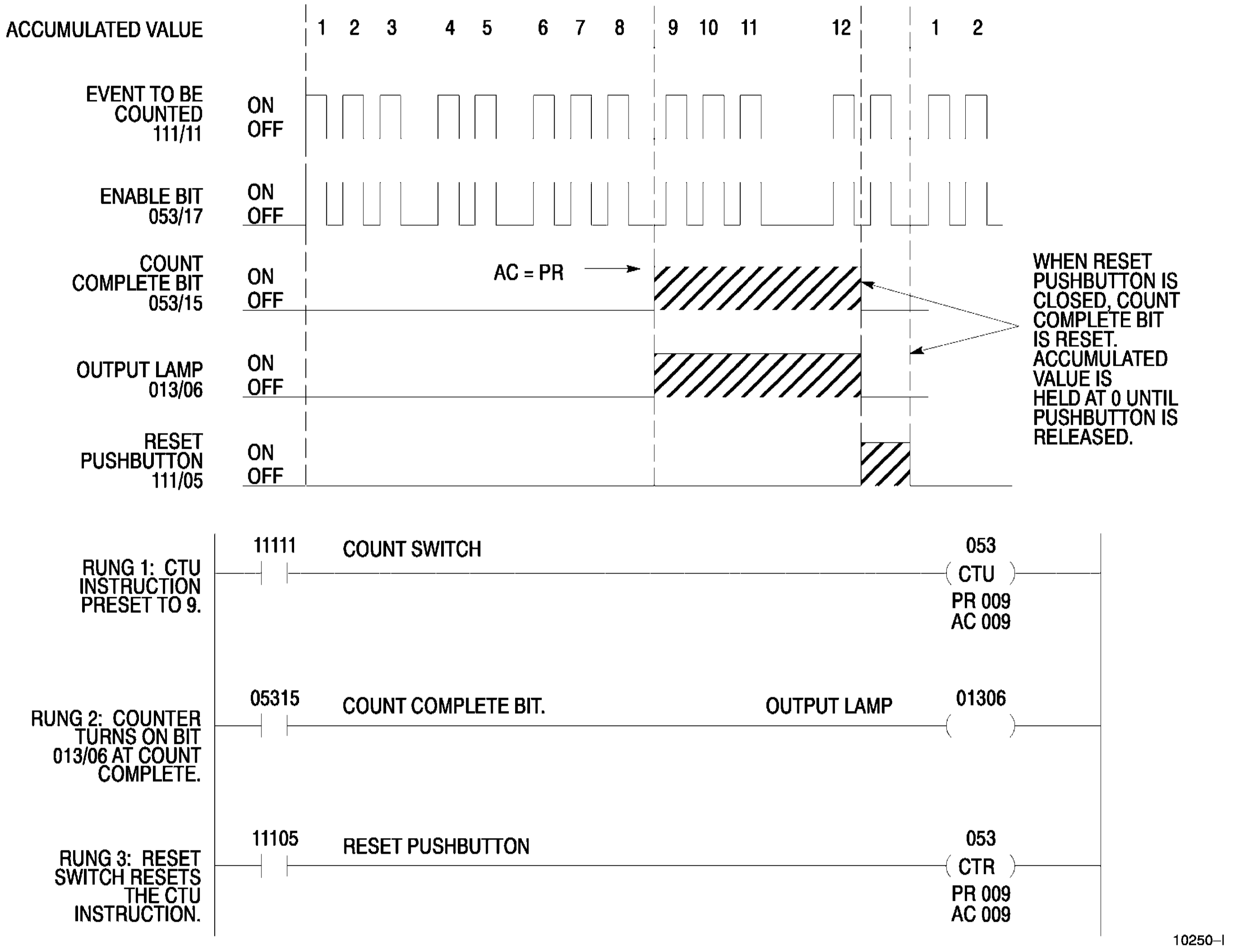


Counter Reset Instruction

The Counter Reset (CTR) instruction is an output instruction that resets the CTU accumulated value and status bits to zero (Figure 4.8).

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.

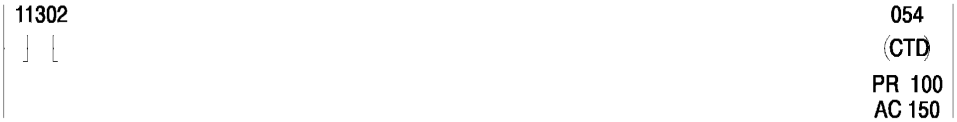
Figure 4.8
Counter with Reset Diagram for Rest = 9 and Programming


ACCUMULATED VALUE
1 2 3 4 5 6 7 8 9 10 11 12 1 2
EVENT TO BE COUNTED 111/11
ON
OFF
ENABLE BIT 053/17
ON
OFF
COUNT COMPLETE BIT 053/15
ON
OFF
AC = PR
OUTPUT LAMP 013/06
ON
OFF
RESET PUSHBUTTON 111/05
ON
OFF
WHEN RESET PUSHBUTTON IS CLOSED, COUNT COMPLETE BIT IS RESET. ACCUMULATED VALUE IS HELD AT 0 UNTIL PUSHBUTTON IS RELEASED.
RUNG 1: CTU INSTRUCTION PRESET TO 9.
11111
COUNT SWITCH
053
CTU
PR 009
AC 009
RUNG 2: COUNTER TURNS ON BIT 013/06 AT COUNT COMPLETE.
05315
COUNT COMPLETE BIT.
OUTPUT LAMP
01306
RUNG 3: RESET SWITCH RESETS THE CTU INSTRUCTION.
11105
RESET PUSHBUTTON
053
CTR
PR 009
AC 009
10250–I

Down–Counter Instruction

The Down–Counter (CTD) instruction subtracts one from its accumulated value for each false–to–true transition of its rung conditions (Figure 4.9). Because only the false–to–true transition causes a count to be made, rung conditions must go from true to false and back to true before the next count is registered.

Figure 4.9
Down Counter Instruction



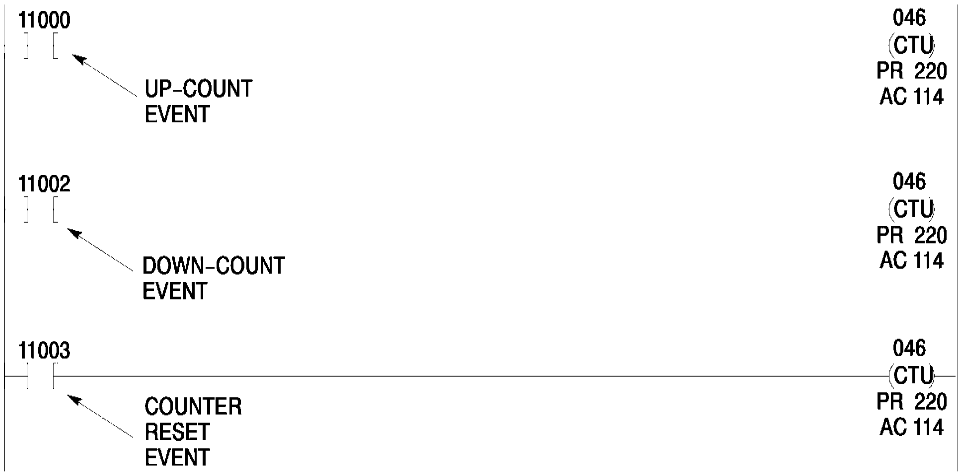
The CTD accumulated value is retained when:

- The mode select switch is changed to the PROGRAM position
- The rung conditions go false
- A power outage occurs provided memory backup power is maintained for RAM memory.

Each time the CTD rung goes true, bit 16, the enabled bit, is set on. When the accumulated value is greater than or equal to the preset value, bit 15 is set on. When the accumulated value goes below 000, bit 14 is set on to indicate an underflow condition and the CTD 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 (Figure 4.10).

Figure 4.10
Up/Down Counter Example



NOTE: Bit 14 of the accumulated value word is set on when the accumulated value either "overflows" or "underflows." Because of this, the programmer may need to monitor bit 14 in some applications. When a Down–Counter preset is set to 000, underflow bit 14 is not set on when the count goes below zero.

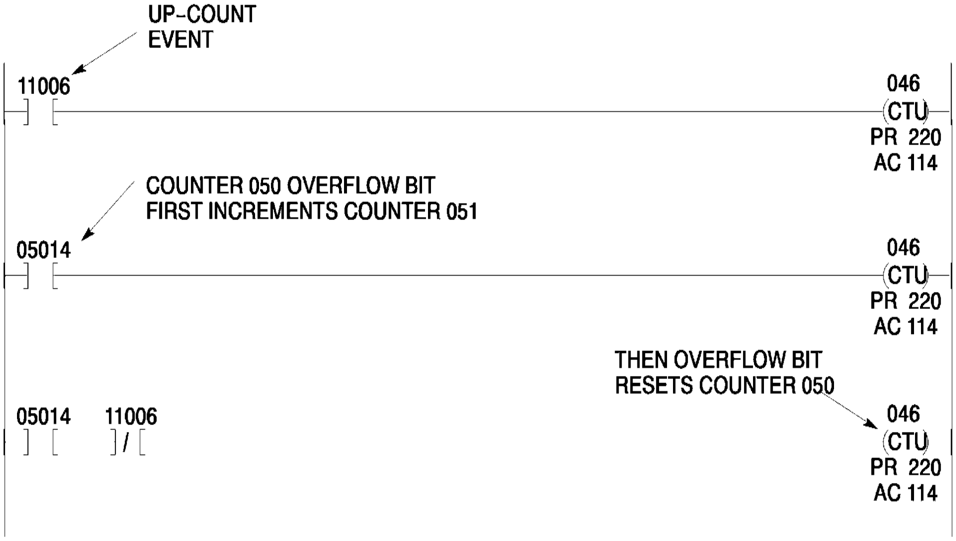
When used alone, the CTD accumulated value may need to be "reset in the program to its original value (usually a value other than 000). For this reason, a GET/PUT transfer, rather than a CTR instruction, is usually used to load a value in the CTD accumulated value word.

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 4.11). 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.

Figure 4.11
Cascading Counters Example



Programming Timer and Counter Instructions

Timer and counter instructions are entered into memory with the processor in the program mode.

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

Counter instruction 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 default word address displayed will depend on the data table configuration (Figure 4.12). Refer to Table 4.A for a complete summary of the instructions.

Figure 4.12
Timer/Counter Default Word Address

#I/O RACKS	T/C ADDRESS
1	020
2	030
3	040
4	050
5	060
6	070
7	200

With the industrial terminal (cat. no. 1770–T1 or –T2) and either processor, the default word address will always be 3 digits.

When the 1770–T3 industrial terminal and 1772–LP2 processor are used, the default word address can be 4 or 5 digits provided the data table is sized accordingly. Unlike bit instructions, the EXPAND ADDR key is not required. Instead, the industrial terminal automatically enters a 4 or 5–digit default word address depending on the data table size. When a 4 or 5–digit word address is displayed and a 3 or 4–digit word address is required, the programmer must enter leading zeros before the word address.

Table 4.A
Timer and Counter Instructions

Keytop Symbol	Instruction Name	1770-T1 or T2 Display	1770-T3 Display	Description
-(TON)-	Timer on Delay	XXX -(TON)- TB PR YYY AC ZZZ		XXX-word address of AC value. TB-time base in seconds; [1][0],0.1 sec = [0][1], 0.01 sec - [0][0]. YYY - preset value from 000 to 999. ZZZ - accumulated value from 000 to 999. Not entered by the user. When rung is true, timer begins to increment the accumulated value at a rate specified by the time base. When rung is false, timer resets the accumulated value word to zero.
			Same	Same as above. Word address displayed will be 3,4 or 5 digits long, depending on the data table size. When entering the word address, use leading zeros if necessary.
-(TOF)-	Timer Off Delay	XXX -(TOF)- TB PR YYY AC ZZZ		XXX-word address of AC value. TB-time base in seconds; 1.0 sec-[1][0],0.1 sec = [0][1],0.01 sec = [0][0]. YYY - preset value from 000 to 999. ZZZ - accumulated value from 000 to 999. Not entered by the user. When rung is true, timer resets the accumulated value word to zero. When rung is false, timer begins to increment the accumulated value.
			Same	Same as above. Word address displayed will be 3,4 or 5 digits long, depending on the data table size. When entering the word address, use leading zeros if necessary.
-(RTO)-	Retentive Timer	XXX -(RTO)- TB PR YYY AC ZZZ		XXX-word address of AC Value. TB - time base in seconds; 1.0 sec = [1][0], 0.1 sec = [0][1],0.01 sec = [0][0] YYY - preset value from 000 to 999. ZZZ - accumulated value from 000 to 999. When rung is true, timer begins to increment the accumulated value. When rung is false, accumulated value is retained. it is reset only with the RTR instruction.

Keytop Symbol	Instruction Name	1770-T1 or T2 Display	1770-T3 Display	Description
			Same	Same as above. Word address displayed will be [A3,4 or 5 digits long, depending on the data table size. When entering the word address, use leading zeros if necessary.
-(RTR)-	Retentive Timer Reset	XXX -(RTR)- TB PR YYY AC ZZZ		XXX-word address of the retentive timer it is resetting. YYY-preset value automatically entered by industrial terminal. ZZZ-accumulated value automatically entered by industrial terminal. When rung is true, the RTO accumulated value word is reset to zero.
			Same	Same as above. Word address displayed will be 3,4 or 5 digits long, depending on the data table size. When entering the word address, use leading zeros if necessary.
-(CTU)-	Up Counter	XXX -(CTU)- PR YYY AC ZZZ		XXX-word address of AC value. YYY-preset value from 000 to 999. ZZZ-accumulated value from 000 to 999. Each time the rung goes true, the accumulated value is incremented one count. The counter will continue counting after the preset value is reached. Accumulated value is reset only by the CTR instruction.
			Same	Same as above. Word address displayed will be 3,4 or 5 digits long, depending on the data table size. When entering the word address, use leading zeros if necessary.
-(CTR)-	Counter Reset	XXX -(CTR)- TB PR YYY AC ZZZ		XXX-word address of the CTU it is resetting. YYY-preset value automatically entered by industrial terminal. ZZZ-accumulated value automatically entered by industrial terminal. When rung is true, the CTU accumulated value word is reset to zero.
			Same	Same as above. Word address displayed will be 3,4 or 5 digits long, depending on the data table size. When entering the word address, use leading zeros if necessary.

Keytop Symbol	Instruction Name	1770-T1 or T2 Display	1770-T3 Display	Description
-(CTD)-	Down Counter	XXX -(CTD)- PR YYY AC ZZZ		XXX-word address of AC value. YYY-preset value from 000 to 999. ZZZ-accumulated value from 000 to 999. Each time the rung goes true, the accumulated value is decreased one count.
			Same	Same as above. Word address displayed will be 3, 4 or 5 digits long, depending on the data table size. When entering the word address, use leading zeros if necessary.

Data Manipulation Instructions

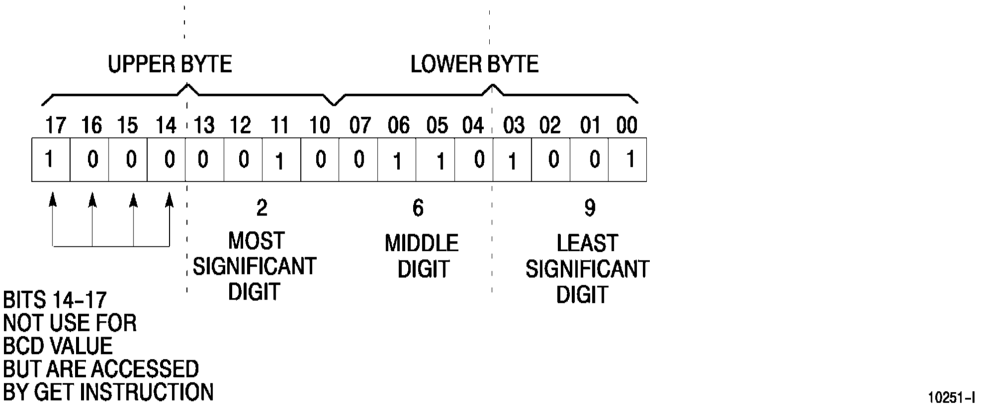
General

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, Put, Les and Equ instructions store the data as 3-digit numeric values in BCD format using the first 12 bits of a data table word (Figure 5.1). This 3-digit value can be decimal number ranging from 000 to 999.

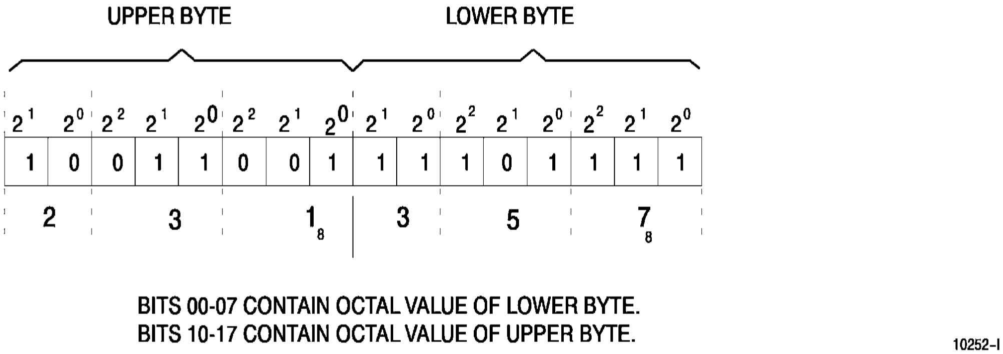
Figure 5.1
BCD Word Format



The Get Byte and Limit Test instructions store the data as 3-digit values in binary coded 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 0008 to 3778. Note that two 3-digit values can be stored in a word: one in the upper byte (bits 10-17) and one in the lower byte (bits 00-07).

A data manipulation instruction can address any word in the data table, excluding processor work areas.

Figure 5.2
Binary Coded Octal Format



2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
1	0	0	1	1	0	0	1	1	1	1	0	1	1	1	1
2		3			1_8			3		5			7_8		

BITS 00-07 CONTAIN OCTAL VALUE OF LOWER BYTE.
BITS 10-17 CONTAIN OCTAL VALUE OF UPPER BYTE.
10252-I

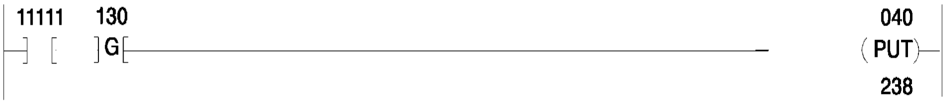
Data Transfer Instructions

The Get and Put instructions are used together to transfer the contents of one word to another word in the data table.

Get Instruction

Get instructions are programmed in the condition area of the ladder diagram rung. They tell the processor to make a duplicate of all 16 bits in the addressed memory word. Although only 12 bits are used to store the BCD value, all 16 bits are duplicated and transferred to the word address of the Put instruction for a data transfer operation (Figure 5.3).

Figure 5.3
GET and PUT Instruction



The Get instruction is not a "condition" that determines rung logic continuity. When the processor is in the run, test or run/prog 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 conditions preceding it. Condition instructions, however, should not be programmed after a Get instruction. When one or more conditions precede the Get instruction, the conditions 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.

If the word addressed by a Get instruction already contains a value, this value is displayed automatically after the word address is entered. Entry of a new BCD value writes over the BCD value previously stored in the addressed word.

Although each data table word stores one BCD value, the word address can be assigned Get instructions many times in the same program. This allows the program to perform several different functions with the same data.

Put Instruction

The processor transfers the duplicate of the 16-bit word from the Get instruction to the word address of the Put instruction (Figure 5.3). Although bits 14-17 are not displayed, they are transferred with the BCD value when the rung goes true. The Put instruction is programmed at the end of the ladder diagram rung and should be preceded immediately by a Get instruction. The Put instruction can be considered "retentive." This means that when the rung goes false, the data at the Put instruction does not change.

Data Comparison Instructions

Data comparison operations differ from data transfer operations in that data table values are not transferred. Instead, these values at different word locations are compared.

Data comparison instructions operate with either BCD values or binary coded octal values. With the Les and Equ instructions, only 12 bits of a word (the BCD values) are compared. Bits 14-17, although duplicated, are not compared. With the Get Byte and Limit Test instructions, 8 bits in a word are compared.

Less and Equ Instructions

The Les (less than) and Equ (equal to) instructions 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.

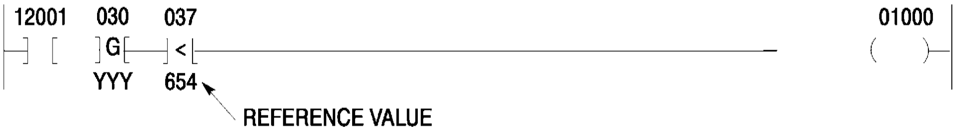
A Get/Less or Get/Equ pair of instructions forms a single condition for logic continuity. Alone or with other conditions, each pair can be used to energize an output device or other output instruction. In all cases, the Get instruction must be programmed before the Les or Equ instruction. If other conditions are also programmed, they must 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 BCD value need not be fixed, however in the following examples it is shown as a fixed value. The Get, Les or Equ instruction can be assigned a fixed or changing value, depending on the type of comparison being made. 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 Less instruction, the fixed value (Figure 5.4). When the Get value is less than the Les value, the comparison is true and logic continuity is established.

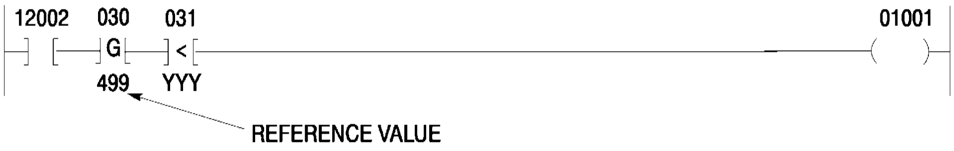
Figure 5.4
Less Than Comparison



WHEN YYY < 654, GET/LES COMPARISON IS TURE AND 010/00 IS ENERGIZED

Greater Than - A greater than comparison is also made with the Get/Les pair of instructions. This time the Get instruction BCD value is fixed 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 5.5). When the Les value is greater than the Get value, the comparison is true and logic continuity is established.

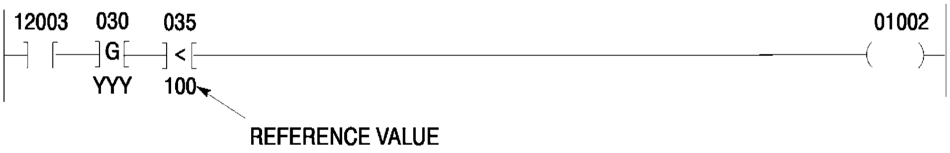
Figure 5.5
Greater Than Comparison



WHEN YYY > 499, GET/LES COMPARISON IS TRUE AND 010/00 OS ENERGIZED

Equal To - An equal to comparison is made with the Get and Equ instructions (Figure 5.6). The Get value is the changing variable and is compared to the fixed 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.

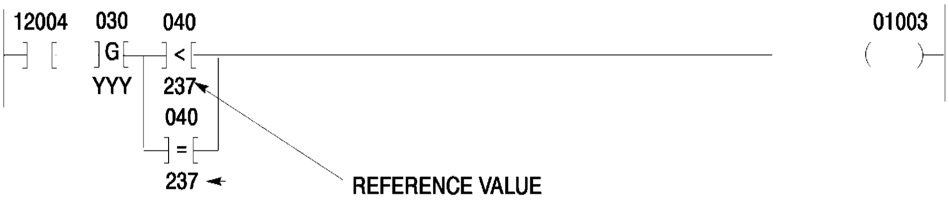
Figure 5.6
Equal To Comparison



WHEN YYY = 100, GET/EQO COMPARISON IS TRUE AND 010/02 IS ENERGIZED

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 fixed value (Figure 5.7). 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.

Figure 5.7
Less Than or Equal To Comparison

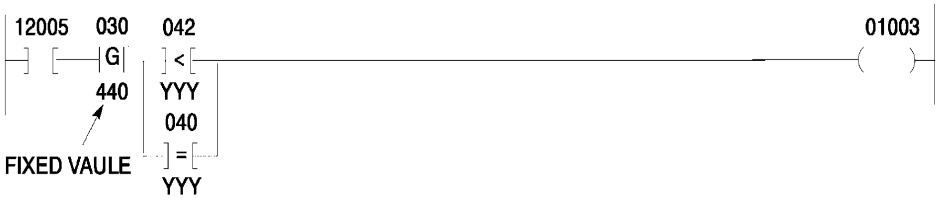


WHEN YYY ≤ GET/LES-EQU COMPARISON IS TRUE AND 010/03 IS ENERGIZED

NOTE: Only one Get instruction is required for a parallel comparison. The Les and Equ instructions are programmed on parallel branches. Their word addresses and BCD values need not be the same.

Greater Than or Equal To - This comparison is made using the Get, Les and Equ instructions. The Get value is assigned a fixed value. the Les and Equ values are changing values that are compared to the Get value (Figure 5.8). When the Les and Equ values are greater than or equal to the Get value, the comparison is true and logic continuity is established.

Figure 5.8
Greater Than or Equal To Comparison



WHEN YYY≥ GET/LES-EQU COMPARISON IS TRUE AND 010/03 IS ENERGIZED

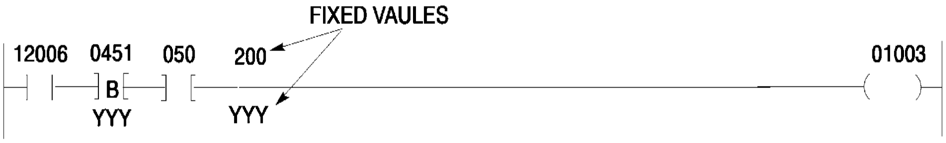
NOTE: Only one Get instruction is required for this parallel comparison. the Les and Equ instructions are programmed on parallel branches and need not have the same word addresses and BCD values.

Get Byte and Limit Test Instructions

The Get Byte and Limit Test instructions are used together to compare an octal value to upper and lower limits that are also octal values. These values can range from 0008 to 3778.

The Get Byte and Limit Test instructions are programmed in the condition area of the ladder diagram rung. Together they form a single condition for logic continuity. condition instructions can be programmed before the Get Byte instruction or after the Limit Test instruction but not between them (Figure 5.9).

Figure 5.9
Get Byte/Limit Test Comparison



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

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 processor makes a duplicate of the upper of 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.

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 as intensified and blinking, and will continue to blink until all information is entered.

The default word address, 010, can appear as 3, 4 or 5 digits, depending on the processor, industrial terminal and data table size. Refer to table 5-1 for a summary of the data manipulation instructions.

The industrial terminal (cat. no. 1770-T1 or-T2) can only display a 3-digit word address no matter what processor is used. The 1770-T3 industrial terminal, however, can display a 4 or 5-digit word address with the 1772-LP2 processor provided its data table has been expanded to that size. When a 4 or 5-digit default address is displayed and a 3 or 4-digit word address is required, the programmer must enter leading zeros before entering the word address.

**Table 5.A
Data Manipulation Instructions**

Keytop Symbol	Instruction Name	1770-T1 or -T2 Display	1770-T3 Display	Description
-[G]-	Get	XXX -[G]- YYY		XXX-word address
				YYY-BCD value from 000 to 999 stored in the lower 12 bits of the word address. Get instruction is used with another data manipulation or arithmetic instruction. When rung is true, all 16 bits of Get instruction are duplicated and the following operation performed.
			Same	Same as above. Word address displayed will be 3, 4, or 5 digits depending on the data table size. When entering the word address use leading zeroes if necessary.
-(PUT)-	Put	XXX -(PUT)- YYY		XXX-word address YYY-BCD value from 000 to 999 stored in the lower 12 bits of the word address. Not entered by user. Should be preceded by a Get instruction. Value at Get instruction is transferred to Put word address when rung is true.
			Same	Same as above. Word address displayed will be 3, 4, or 5 digits depending on the data table size. When entering the word address use leading zeroes if necessary.
-[<]-	Less Than	XXX -[<]- YYY		XXX-word address YYY-BCD value from 000 to 999 stored in the lower 12 bits of the word address. Should be preceded by a Get instruction. BCD values at Get and Less Than words are compared.
			Same	Same as above. Word address displayed will be 3, 4, or 5 digits depending on the data table. size. When entering the word address use leading zeroes if necessary.
-[=]-	Equal To	XXX -[=]- YYY		XXX-word address YYY-BCD value from 000 to 999 stored in the lower 12 bits of the word address. Should be preceded by a Get instruction. BCD values at Get and Less Than words are compared.

Keytop Symbol	Instruction Name	1770-T1 or -T2 Display	1770-T3 Display	Description
			Same	Same as above. Word address displayed will be 3, 4, or 5 digits depending on the data table size. When entering the word address use leading zeroes if necessary.
-[B]-	Get Byte	XXX D -[B]- YYY		XXX-word address YYY-Octal value from 000_8 to 377_8 stored in the upper or lower byte of the word address. D - designates the upper or lower byte of the word, 1 = upper byte, 0 = lower byte. Should be followed by a Limit Test instruction. A duplicate of the designated byte is made and compared to the upper and lower limits of the Limit Test instruction.
			Same	Same as above. Word address displayed will be 3, 4, or 5 digits depending on the data table. size. When entering the word address use leading zeroes if necessary.
-[L]-	Limit Test	XXXAAA -[L]- BBB		XXX-word address AAA-The upper limit for the comparison. Must be ≥BBB. An octal value from 000_8 to 377_8. BBB-The lower limit for the comparison. An octal value from 000_8 to 377_8. Should be preceded by a Get Byte Instruction. Compares the value at the Get Byte instruction to the values at the Limit Test instruction.
			Same	Same as above. Word address displayed will be 3,4 or 5 digits depending on the data table size. When entering the word address use leading zeros if necessary.

Chapter 6

Arithmetic Instructions

General

The PLC–2/20 processor can be programmed to perform arithmetic operations with two BCD values using a set of arithmetic instructions. These output instructions are:

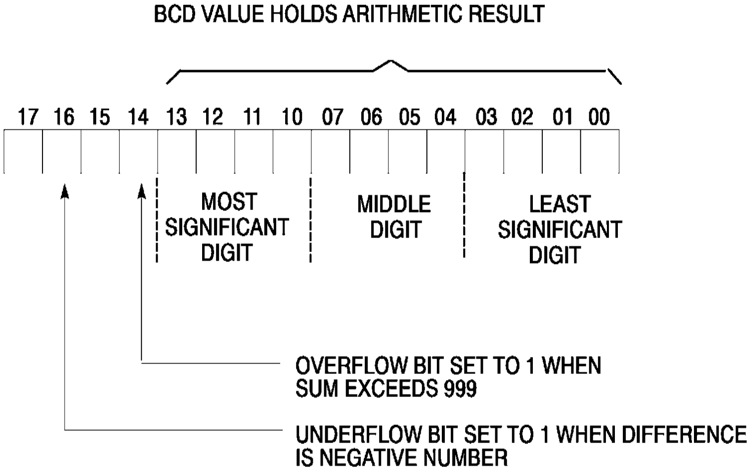
- Add
- Subtract
- Multiply
- Divide

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 (fNO TAG). Two of the remaining bits (bits 14 and 16) are used to indicate overflow and underflow conditions.

Figure 6.1
Arithmetic Instruction Word


BCD VALUE HOLDS ARITHMETIC RESULT
17 16 15 14 13 12 11 10 07 06 05 04 03 02 01 00
MOST SIGNIFICANT DIGIT
MIDDLE DIGIT
LEAST SIGNIFICANT DIGIT
OVERFLOW BIT SET TO 1 WHEN SUM EXCEEDS 999
UNDERFLOW BIT SET TO 1 WHEN DIFFERENCE IS NEGATIVE NUMBER

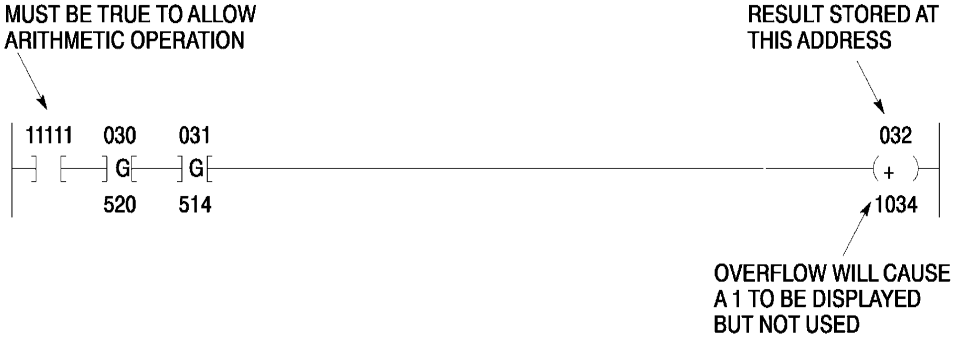
10253-I

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/prog mode, the overflow condition is displayed on the industrial terminal screen as a “1”.

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

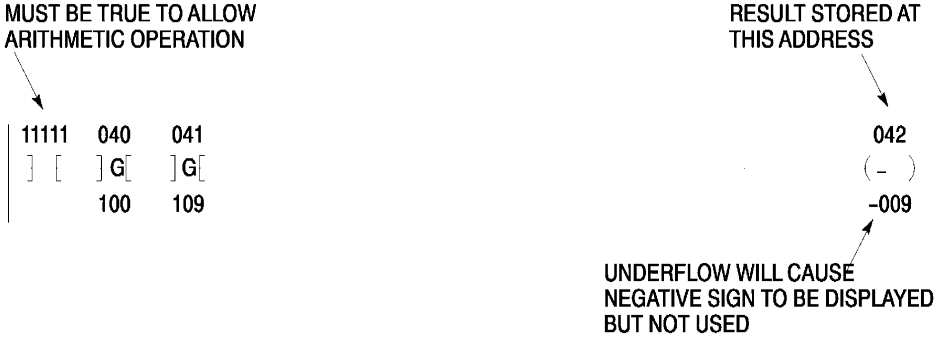
Figure 6.2
Add Instruction



Subtract Instruction

The Subtract instruction tells the processor to subtract the second Get word value from the first Get word value (fFigure 6.3). The difference is then stored at the data table word addressed by the Subtract instruction.

Figure 6.3
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/prog mode, the negative sign will appear on the industrial terminal screen.

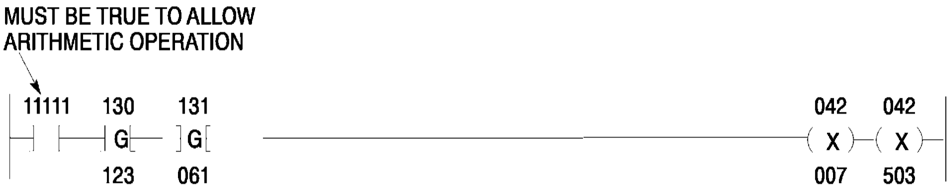
NOTE: If a negative BCD value is used for subsequent operations, inaccurate results will occur. The processor only compares, transfers and computes the BCD value and not the underflow bit.

Multiply Instruction

The Multiply 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, the programmer should choose two consecutive data table words to store the product. If the product is less than 6 digits, leading zeros will appear in the product where there is not value.

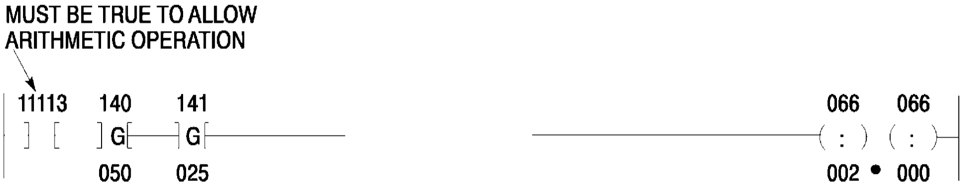
Figure 6.4
Multiply Instruction



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 two consecutive data table word locations are chosen to store the quotient for ease of programming.

Figure 6.5
Divide Instruction



The quotient is not rounded off and is always 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 following results are displayed when dividing by 0:

$\frac{0}{0} = 001.00$ $\frac{1 \text{ to } 999}{0} = 999.9990$

This differs from the Mini–PLC–2 and the Mini–PLC–1/15 where 0 ÷ 0 = 999.999

Programming Arithmetic Instructions

Arithmetic instructions are entered into memory with the PLC–2/20 processor in the program mode. When entered, these instructions will be intensified and blinking. They will continue to blink until the word address is entered. Refer to Table 6.A for a summary of these instructions.

When the industrial terminal (cat. no. 177–T1 and –T2) is used with either processor, the 3–digit default word address, 010, will appear above the arithmetic instruction.

When the 1770–T3 industrial terminal and 1772–LP2 processor are used, a 4 or 5– digit default word address, 0010 or 00010, can be displayed above the instruction provided the data table is expanded accordingly. when the data table is expanded to a 5–digit word address, the 5–digit default address will be displayed. To enter a 3 or 4–digit word address when 4 or 5 digits are displayed, the programmer must enter leading zeros before entering the word address.

Table 6.A
Arithmetic Instructions

Keytop Symbol	Instruction Name	1770-T1 or T2 Display	1770-T3 Display	Description
-(+)-	Add	XXX -(+)- YYY		XXX-word address YYY-Result of addition operation. A BCD value from 000 to 999 stored in the first 12 bits of the word address. Not entered by user. An output instruction always preceded by two Get instructions. The two Get instructions store the two BCD values to be added. When the sum exceeds 999, bit 14 is set to 1 and is displayed in front of YYY.
			Same	Same as above. Word address displayed will be 3, 4 or 5 digits depending on the data table size. When entering the word address use leading zeros if necessary.
-(-)-	Subtract	XXX -(-)- YYY		XXX-word address YYY-Result of subtraction operation. A BCD value from 000 to 999 stored in the first 12 bits of the word address. Not entered by user. An output instruction always preceded by two Get instructions. The two Get instructions store the two BCD values to be subtracted. When the difference is negative, bit 16 is set to 1 and a minus sign is displayed in front of YYY.
			Same	Same as above. Word address displayed will be 3, 4 or 5 digits depending on the data table size. When entering the word address use leading zeros if necessary.
-(X)-	Multiply	XXX ZZZ -(X)-(X)- AAA BB		XXX, ZZZ - two word addresses to store the product AAA BBB - 6-digit product. AAA = most significant digits, BBB = least significant digits. Not entered by user. An output instruction always preceded by two Get instructions. The two Get instructions store the two BCD values to be multiplied.
			Same	Same as above. Word address displayed will be 3,4 or 5 digits depending on the data table size. When entering the word address use leading zeros if necessary.

Keytop Symbol	Instruction Name	1770-T1 or T2 Display	1770-T3 Display	Description
-(÷)-	Divide	XXX ZZZ -(÷)-(÷)- AAA, BBB		XXX, ZZZ – two word addresses to store the quotient. AAA,BBB-6-digit quotient. AAA=whole number, BBB=decimal number. Not entered by user. An output instruction always preceded by two Get instructions. The two Get instructions store the two BCD values to be divided. The decimal point is automatically entered between AAA and BBB by the industrial terminal.
			Same	Same as above. Word address displayed will be 3,4 or 5 digits depending on the data table size. When entering the word address use leading zeros if necessary.

Chapter 7

Other Program Instructions

General

There are four other program instructions the user may need 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

Output Overrides

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

- Master Control Reset
- Zone Control Last State

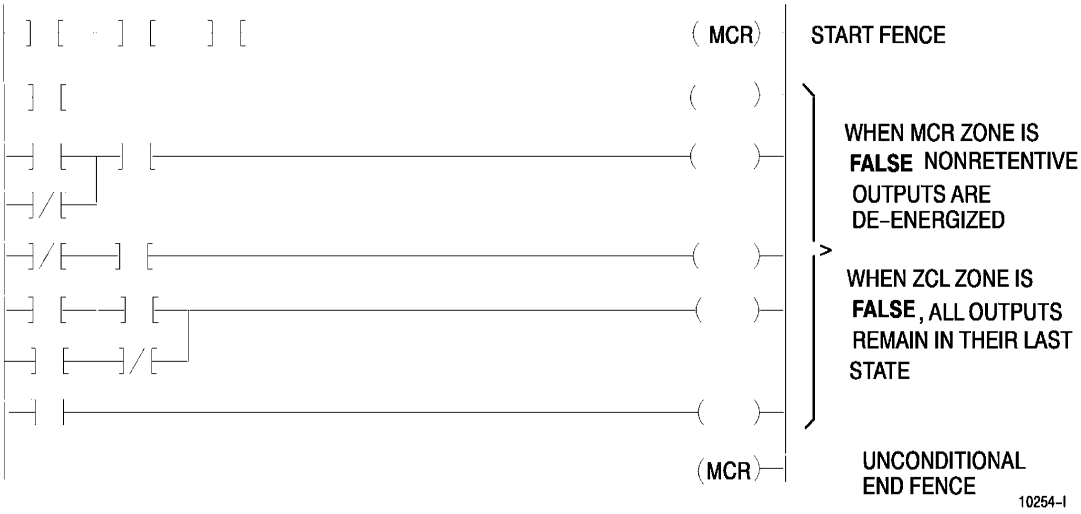
These instructions are similar to a hardwired master control relay in that they can affect a group of outputs in the user program. The MCR and ZCL instructions, however, are not a substitute for a hard–wired relay, which provides emergency stop capabilities for all I/O devices.

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 shut down. Emergency stop switches can be monitored but should not be controlled by the user program. These devices should be wired as described in the PLC–2/20, 2/30 Assembly and Installation Manual (publication 1772–807).

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 is always programmed with a set of input conditions. The end fence is 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.

Figure 7.1
MCR/ZCL Zone Programming (MCR shown)


(MCR)
START FENCE
WHEN MCR ZONE IS FALSE NONRETENTIVE OUTPUTS ARE DE-ENERGIZED
WHEN ZCL ZONE IS FALSE, ALL OUTPUTS REMAIN IN THEIR LAST STATE
(MCR)
UNCONDITIONAL END FENCE
10254-I

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.

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

I/O Updates

Two instructions used to update I/O data during the execution of the user program are:

- Immediate Input
- Immediate Output

These instructions are used to transfer critical I/O data ahead of the normal scan sequence. This speeds up the response of output devices to the program and the update of input data for program use.

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.

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.

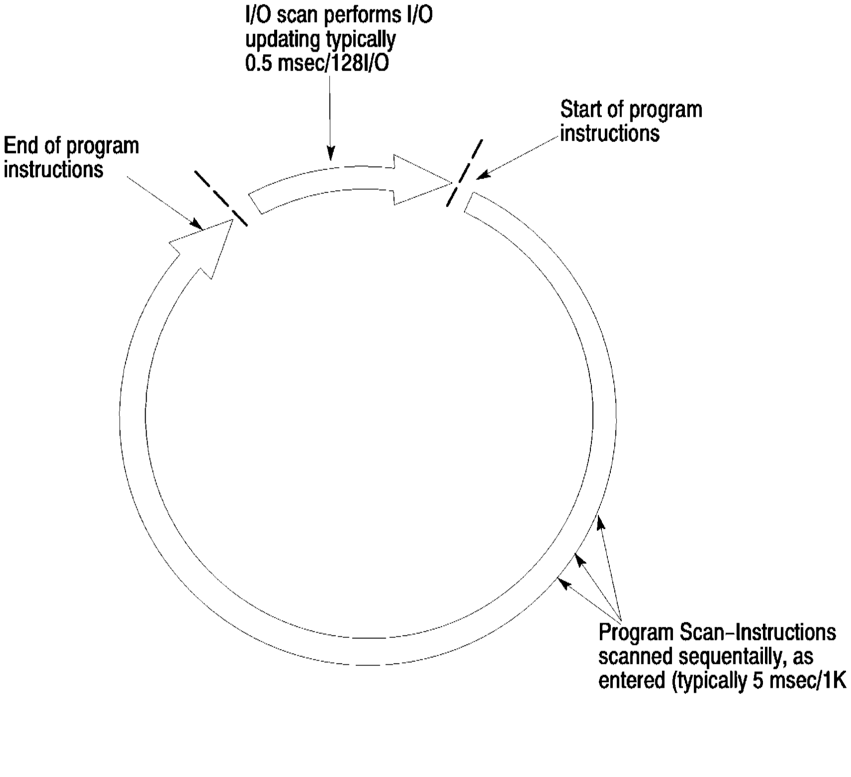
Scan Sequence

The PLC–2/20 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 the input modules is transferred to the input image table. Data from the output image table is transferred to the output modules.

Figure 7.2
Scan Sequence


I/O scan performs I/O updating typically 0.5 msec/128I/O
Start of program instructions
End of program instructions
Program Scan-Instructions scanned sequentailly, as entered (typically 5 msec/1K
10255-I

After completing the I/O scan, the processor begins the program scan. Here, all user program instructions are scanned and executed in the order they were entered.

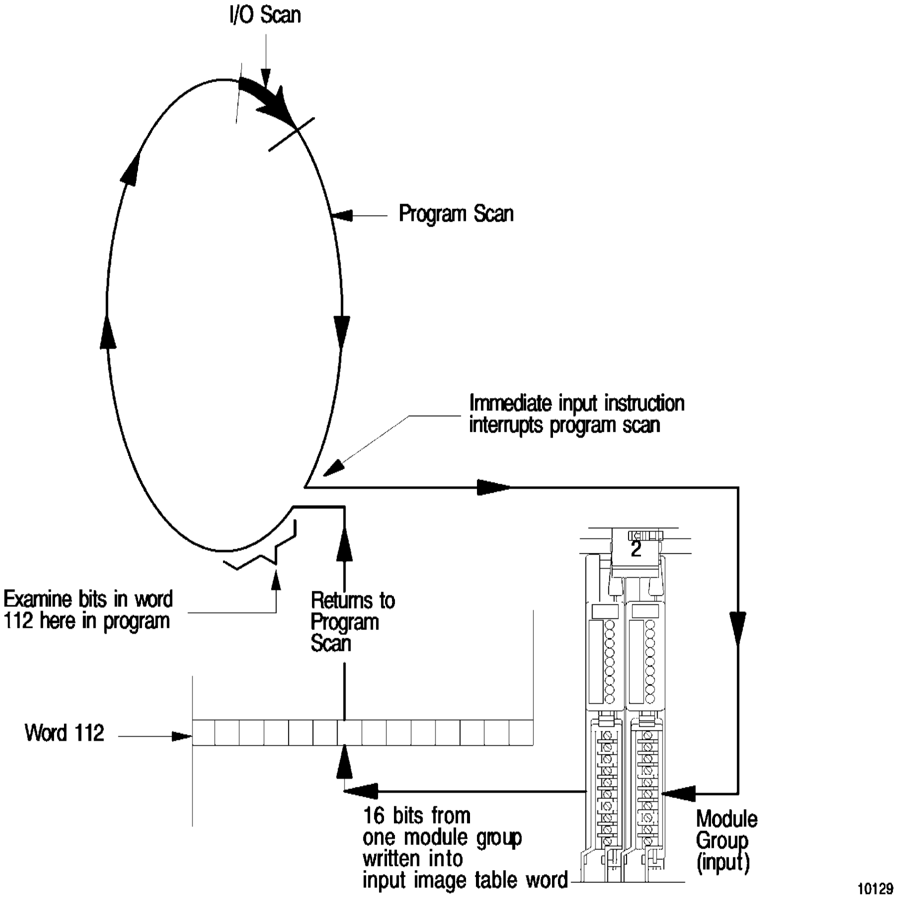
In a local system, the I/O scan and program scan are synchronously performed, one after the other. The time required to complete both scans is typically 5 ms/lK instructions plus 0.5 ms/128 I/O rack. For scan information with remote systems, refer to section 12.

It is clear that 40–50 ms may pass before I/O data is updated in a 8K system. The purpose of the Immediate I/O instructions is to interrupt the program scan to update a word of critical I/O data in advance of the normal update sequence.

Immediate Input Instruction

The Immediate Input instruction 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.

Figure 7.3
Immediate Input Instruction


I/O Scan
Program Scan
Immediate input instruction interrupts program scan
Examine bits in word 112 here in program
Returns to Program Scan
Word 112
2
16 bits from one module group written into input image table word
Module Group (input)
10129

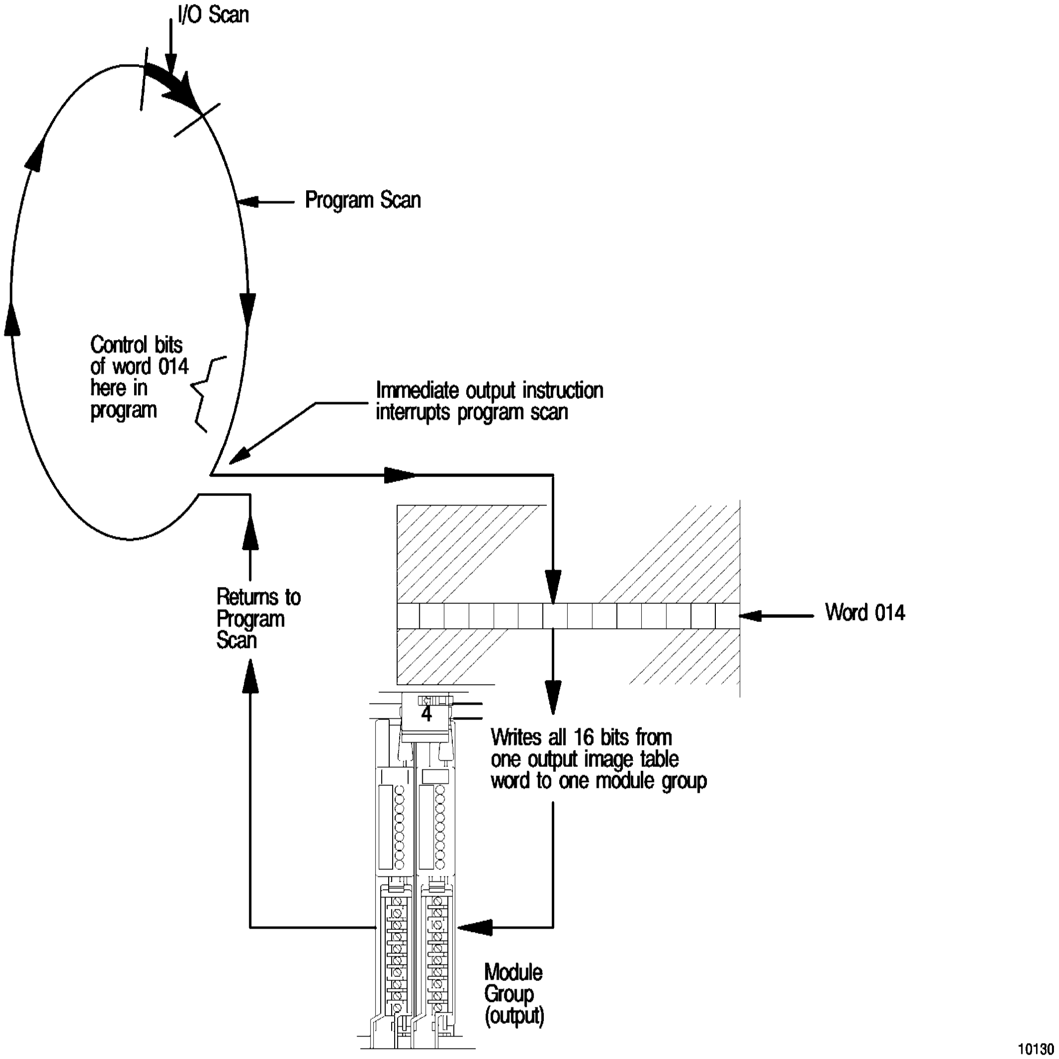
The Immediate Input instruction is programmed in the condition area of the ladder–diagram rung. The Immediate Input instruction can be considered as always true; it is always executed, whether or not other rung conditions allow logic continuity.

Program the Immediate Input instruction only when necessary. This depends on both the response time of the specific input devices and modules and on the position of the rungs examining these inputs in the program. It is best to program the Immediate Input instruction just before inputs in the module group are examined.

Immediate Output Instruction

The Immediate Output instruction updates one module group with data from one output image table word ahead of the normal scan sequence (Figure 7.4).

Figure 7.4
Immediate Output Instruction


I/O Scan
Program Scan
Control bits of word 014 here in program
Immediate output instruction interrupts program scan
Returns to Program Scan
Word 014
4
Writes all 16 bits from one output image table word to one module group
Module Group (output)
10130

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 module group.

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 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.

Programming Immediate I/O Instructions

The Immediate I/O instructions are programmed with the processor in the program mode. When entered from the industrial terminal, they will be displayed as intensified and blinking with the reverse–video cursor positioned on the first digit of the default word address. The number of digits in the default address can range from 3 to 5 depending on the processor and industrial terminal used. Refer to Table 7.A for a summarized description of these instructions.

When the industrial terminal (cat. no 1770–T2 or –T2) and either PLC–2/20 processor are used, the default word address is always 3 digits, 010 or 110.

A 4 or 5–digit word address is possible only when the 1772–LP2 processor and the 1770–T3 industrial terminal are used. The 4 or 5–digit default word address with leading zeros will automatically appear above the instruction provided the data table has been expanded accordingly. To enter a 3 or 4–digit address when 4 or 5 digits are displayed, the programmer must enter leading zeros before entering the word address.

Table 7.A
Other Program Instructions

Keytop Symbol	Instruction Name	1770-T1 ot -T2 Display	1770-T3 Display	Description
-(MCR)-	Master Control Reset	-(MCR)-	Same	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 MCR is false, all outputs except those forced on or latched on, will be de-energi<ed. Do not overlap MCR zones.
-(ZCL)-	Zone Control Last State	-(ZCL)-	Same	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 ZCL is false, outputs will remain ion their last state. Do not overlap ZCL zones.
-[I]	Immediate Input	XXX -[I]-		XXX - Input image table word address. Processor interrupts program scan to update input image table with data from corresponding module group. Updated before the normal I/O scan. Executed each program scan.
			Same	Same as above. Word address displayed will be 3, 4 0r 5 digits depending on the data table size. When entering the word address use leading zeroes if necessary.
-(IOT)-	Immediate Output	XXX -(IOT)-		XXX - Output image table word address. WHen rung is true, processor interrupts program scan to update module group with data from corresponding output image table word address. Updated before the normal I/O scan. Eecuted each program scan when programmed unconditionally. .
			Same	Same as above. Word address displayed will be 3, 4 or 5 digits depending on the data table size. When entering the word address use leading zeroes if necessary.

Chapter 8

Writing the User Program

General

A thorough understanding of the programming instructions and processor operation as described in chapter 1 section titled, 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.

Developing the Program

The first step in developing the user program is 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 schematic of hardware wiring or a process diagram exists, it can be used as an aid in developing a ladder diagram program.

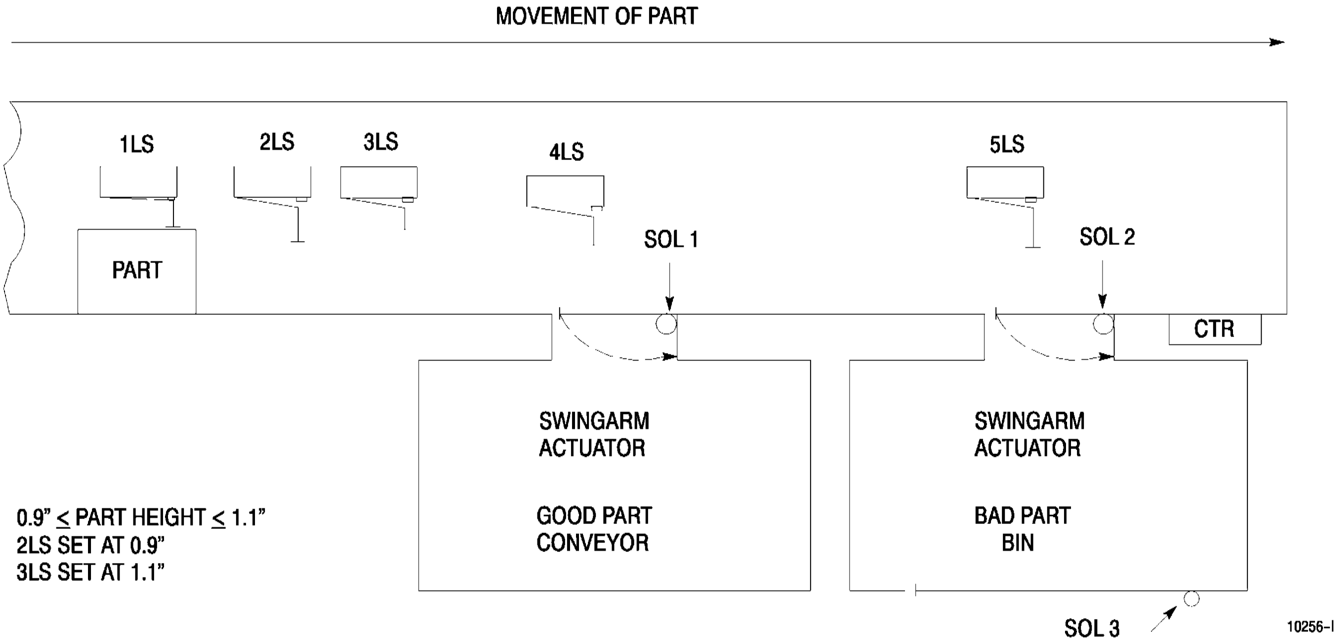
Sample Program

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

The application is one of separating good parts from bad parts. Figure 8.1 shows a part moving along a conveyor belt. Each part will trip a series of limit switches and will be sorted according to its height. The desired height is 1.0" ±10.1".

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.

Figure 8.1
Conveyor Belt Example

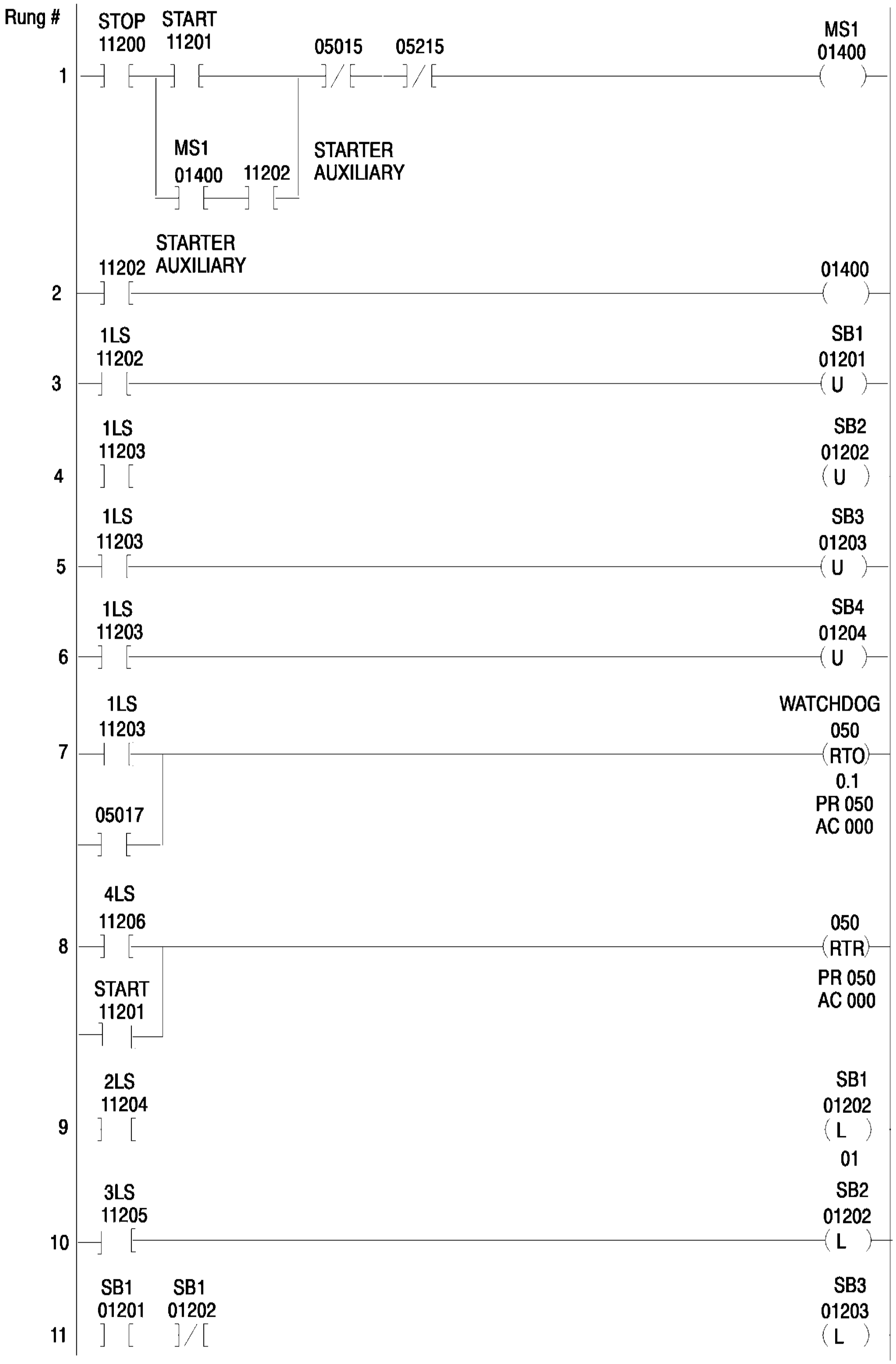


If the part trips both or neither 2LS and 3LS, the part is too large or too small. When either condition occurs, a storage bit (4SB) is latched on. Although the part will trip 4LS, it will continue along and trip 5LS, which energizes SOL2. The swingarm actuator will direct the part into the bad part 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, lLS 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 allows remote observation of the conveyor operation. Additional documentation (not shown) would include a power distribution schematic showing a hardwired master control relay and emergency stop switches.

The logic can be written as a PC ladder diagram program (Figure 8.2). Data table addresses are assigned to the hardwired devices (Table 8.A). The ladder diagram should be developed by analyzing the logic required to operate the machine. A rung by rung description of the logic follows.

Figure 8.2

Ladder Diagram Program



10257-I

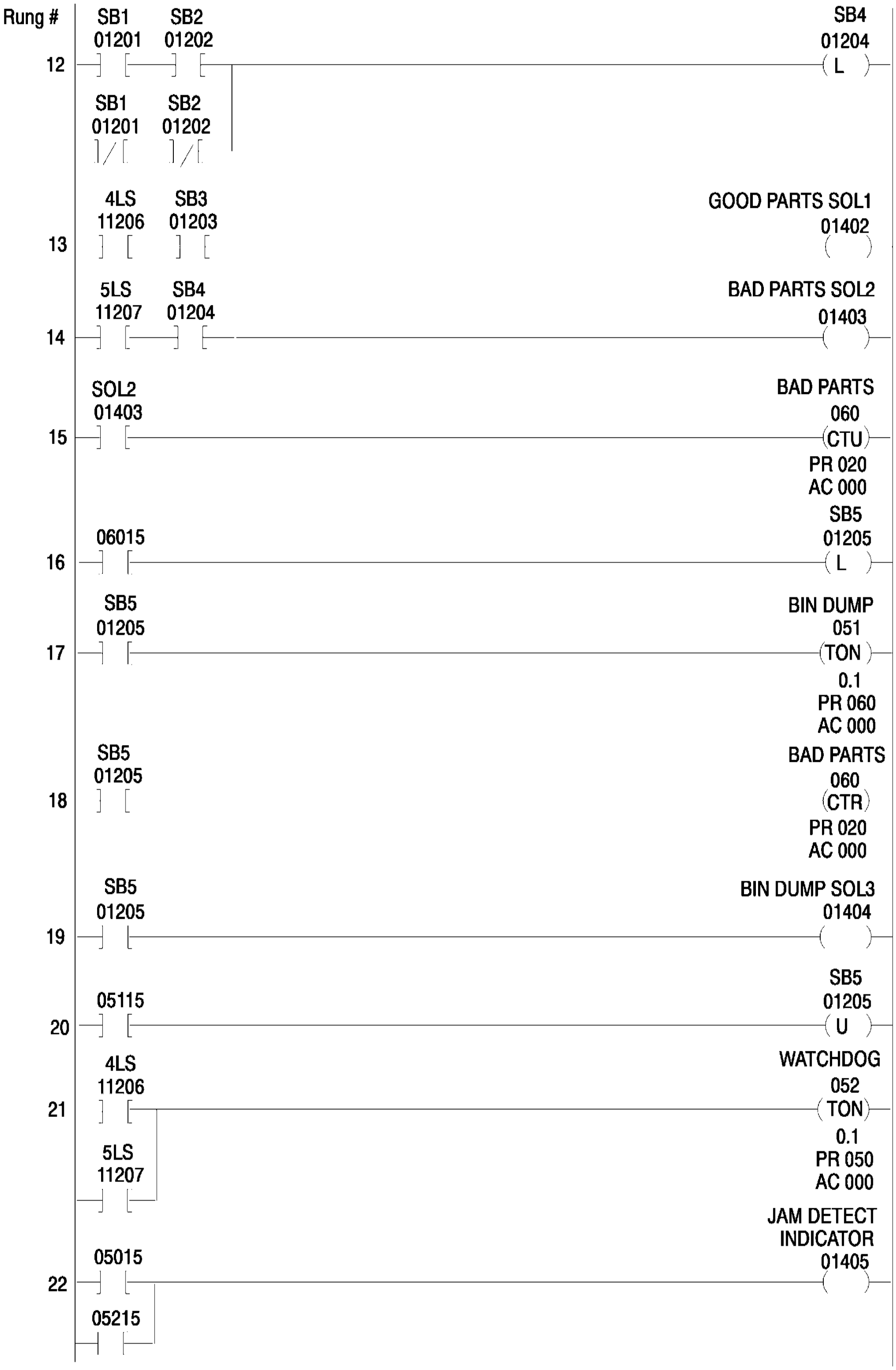

Rung #
12
SB1 01201
SB2 01202
SB4 01204 L
SB1 01201
SB2 01202
13
4LS 11206
SB3 01203
GOOD PARTS SOL1 01402
14
5LS 11207
SB4 01204
BAD PARTS SOL2 01403
15
SOL2 01403
BAD PARTS 060 CTU PR 020 AC 000
16
06015
SB5 01205 L
17
SB5 01205
BIN DUMP 051 TON 0.1 PR 060 AC 000
18
SB5 01205
BAD PARTS 060 CTR PR 020 AC 000
19
SB5 01205
BIN DUMP SOL3 01404
20
05115
SB5 01205 U
21
4LS 11206
5LS 11207
WATCHDOG 052 TON 0.1 PR 050 AC 000
22
05015
05215
JAM DETECT INDICATOR 01405

Table 8.A
Data Table Addresses for Hardwired Devices

Input Device	Address
STOP Pushbutton	112/00
START Pushbutton	112/01
Motor Starter Auxiliary	112/02
Limit Switch (1LS)	112/03
Limit Switch (2LS)	112/04
Limit Switch (3LS)	112/05
Limit Switch (4LS)	112/06
Limit Switch (5LS)	112/07
Output Device	
Motor Starter (MS1)	014/00
Conveyor RUN Indicator	114/01
Good Part Solenoid (SOL1)	014/02
Bad art Solenoid (SOL2)	014/03
Bin Dump Solenoid (SOL3)	014/04
JAM Detect Indicator	014/05
Internal Functions	
Storage Bit 1 (SB1)	012/01
Storage Bit 2 (SB2)	012/02
Storage Bit 2 (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

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 ≥0.9 inch. SB1 remains unlatched if the height <0.9 inch.

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

Rung 11 – A part within tolerance latches SB3.

Rung 12 – A part out of tolerance latches SB4.

Rung 13 – A good part of 4LS actuates SOL1 with swing arm actuator to direct the part to the good part conveyor.

Rung 14 – A bad part of 5LS actuates SOL2 with swing arm 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 reset 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 0518 unlatches SB5 which in turn resets the timer.

Rung 21 – 4LS or 5LS enables the watchdog timer 0528. If 4LS or 5LS is held closed by a jam, this timer will time out.

Rung 22 – Timed–out bits are monitored to provide a JAM indicator to the operator.

I/O Assignments

Once the rough sketch of the application is complete, the programmer can assign data table bit addresses 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, module group and terminal number. Because the bit address is hardware–related, the programmer cannot arbitrarily assign bit addresses to I/O devices. Some intelligent I/O modules use word addresses rather than 5–digit bit addresses. Refer to their user's manuals for addressing and wiring information.

The installer and programmer of the PLC–2/20 controller should work closely together to determine the best placement of the I/O modules within the I/O chassis. To simplify installation and troubleshooting procedures, it may be desirable to group like modules together. It is also helpful to document I/O assignments on a form such as publication 5039 and their image table assignments on a form like publication 5047. Both forms are presented at the end of this chapter. Recommendations for I/O wiring and module placement can be found in the PLC–2/20, –2/30 Programmable Controller Assembly and Installation Manual (publication 1772–807).

Timer/Counter Assignments

Timers and counters and block transfer areas must also be assigned data table word addresses. It is best to map out these word addresses used for timers and counters and block transfer on data table assignment sheets like publication 5046 presented at the end of this chapter. Later, when sizing the data table, this list will be useful.

The first available timer/counter address depends on the number of I/O racks used. the PLC–2/20 processor (cat. no. 1772–LP1) can have up to 4 I/O racks. The corresponding addresses for the first timer/counter locations are shown in Table 8.B. The PLC–2/20 processor (cat. no. 1772–LP2) can have up to 7 I/O racks. The corresponding addresses for the first timer/counter locations are shown in Table 8.C.

Table 8.B
Timer/Counter Address for 1772-LP1 Processor

I/O Racks	First Timer/Counter Word Address
1	020
2	030
3	040
4	050

Table 8.C
Timer/Counter Address for 1772-LP2 Processor

I/O Racks	First Timer/Counter Word Address
1	020
2	030
3	040
4	050
5	060
6	070
7	200

Storage Assignments and Recommendations

Data table addresses for bit and word storage should be chosen carefully to optimize memory use. 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. These bits for a 10ms timer, however, cannot be used for storage. They are used for internal timing functions.
- Unused data table words in timer/counter areas can be used for bit/word storage. To conserve memory, use both the accumulated and preset value words for storage.
- Output image table words can be used for storage when the corresponding input image table words are used for wired input modules. When there is a vacant module group or slot in the I/O chassis, however, do not use these output image table words for storage. This will allow room for future system expansion.
- Unused input image table words cannot be used for storage. They are cleared to zero during each I/O scan.
- Word 027_8 cannot be used for storage. Many of the bits are used by the processor for control functions.
- Do not use output image table words for bit storage when the I/O rack has a module with block transfer capability at the corresponding rack address.

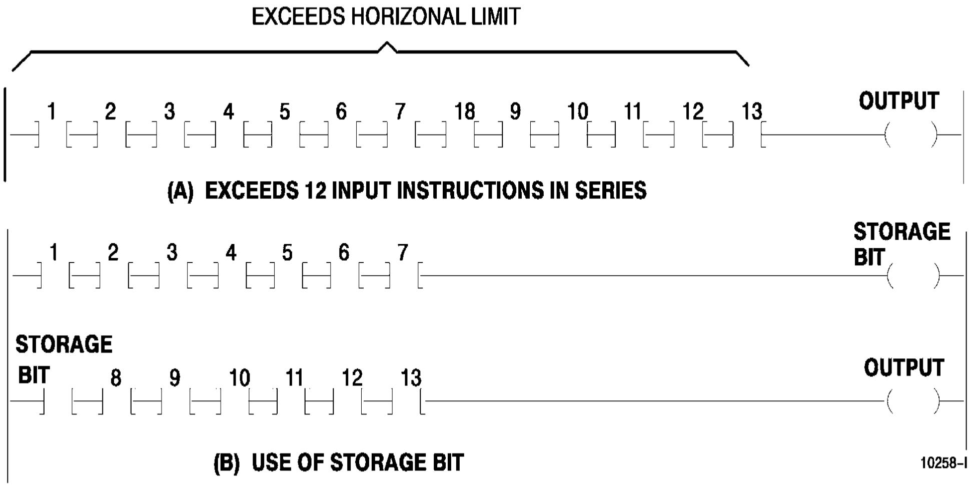
Program Recommendations

The following recommendations for constructing a ladder diagram rung should be considered:

- Any Examine instruction can be repeated as many times as needed in a ladder diagram program or rung.
- Up to 12 input instructions in series can be programmed in a rung provided the output instruction is not a Multiply or Divide instruction. Only one output instruction can be programmed in a rung.
- Limit the number of parallel branches in a rung to 7 since this is the vertical limit of the industrial terminal display.
- Program only one rung to energize an output device to simplify troubleshooting and maximize safety.
- When more than 12 input instructions in series are required to energize an output device (Figure 8.3a), use a storage bit to make two rungs (Figure 8.3b).

- Assign address for Get instructions associated with block transfer to the lowest address above the image tables. Assign T/C addresses and bit storage after the zeros word associated with block transfer (see chapter 11 section titled Block Transfer).

Figure 8.3
Storage Bit Example



Sizing the Data Table

Initially, the data table is set to 128 words for 2 I/O racks and 40 timers/counters. Before the user program is entered into memory, the number of I/O racks and "equivalent" timers/counters should be determined. With this information, the data table can then be set to the correct configuration.

The "equivalent" timers/counters include storage bits and words as well as block transfer areas. To determine the number of "equivalent" timers/counters, use the following formula:

ET = T + C + IS/2 where T = timers
C = counters
IS = internal storage words

Because timers and counters require 2 data table words and storage words require 1 data table word, IS must be divided by 2 to equate it to timers and counters. If IS/2 does not divide evenly, round up to the next whole number.

The data table is configured by pressing SEARCH 50 on the industrial terminal and entering the appropriate information. One of two displays will appear on the screen, depending on the processor and industrial terminal being used.

The 1772–LP1 processor can have a data table up to 256 words long. It can be expanded from 128 to 256 words in two–word increments. When sizing the data table with the SEARCH 50 function, the number of timers/counters allowed will depend on the number of I/O racks selected. See Table 8.D.

The 1772–LP2 processor can have a data table up to 8064 words long. The first two 128–word sections (to 256 words) are expanded in two–word increments. After 256 words, the data table is expanded in 128–word sections. When sizing the data table with the SEARCH 50 function, the user must specify the number of 128–word data table sections.

To determine the number of 128–word sections, first determine the total number of "equivalent" timers/counters using the formula ET = T + C + IS/2. Determine the number of timers/counters that are permitted with the required I/O racks in 256 words (2 128–word sections). See Table 8.D. If more data table is required for timers/counters, increase the number of 128–word data table sections to accommodate the remaining timers/counters. (Each additional 128– word section can accommodate 64 "equivalent" timers/counters.)

For example, suppose 475 timers/counters and six I/O racks are required. In the first two 128–word data table sections, 72 timers/counters can be accommodated (Table 8.D). the remaining 403 timers/counters can fit in seven 128–word data table sections. Thus, nine 128–word sections are required for 475 timers/counters.

Refer to chapter 9 section titled Search Functions for a description of the SEARCH 50 function.

Table 8.D
Data Table Configuration

#I/O Racks	#128 Word Data Table Sections	Maximum # Timers/ Counters
1	1	47
	2	111
2	1	40
	2	104
3	1	32
	2	96
4	1	24
	2	88
5	1	16
	2	80
6	1	8
	2	72
7	1	0
	2	64

ALLEN-BRADLEY
A ROCKWELL INTERNATIONAL COMPANY

Bulletin 1771 I/O Chassis

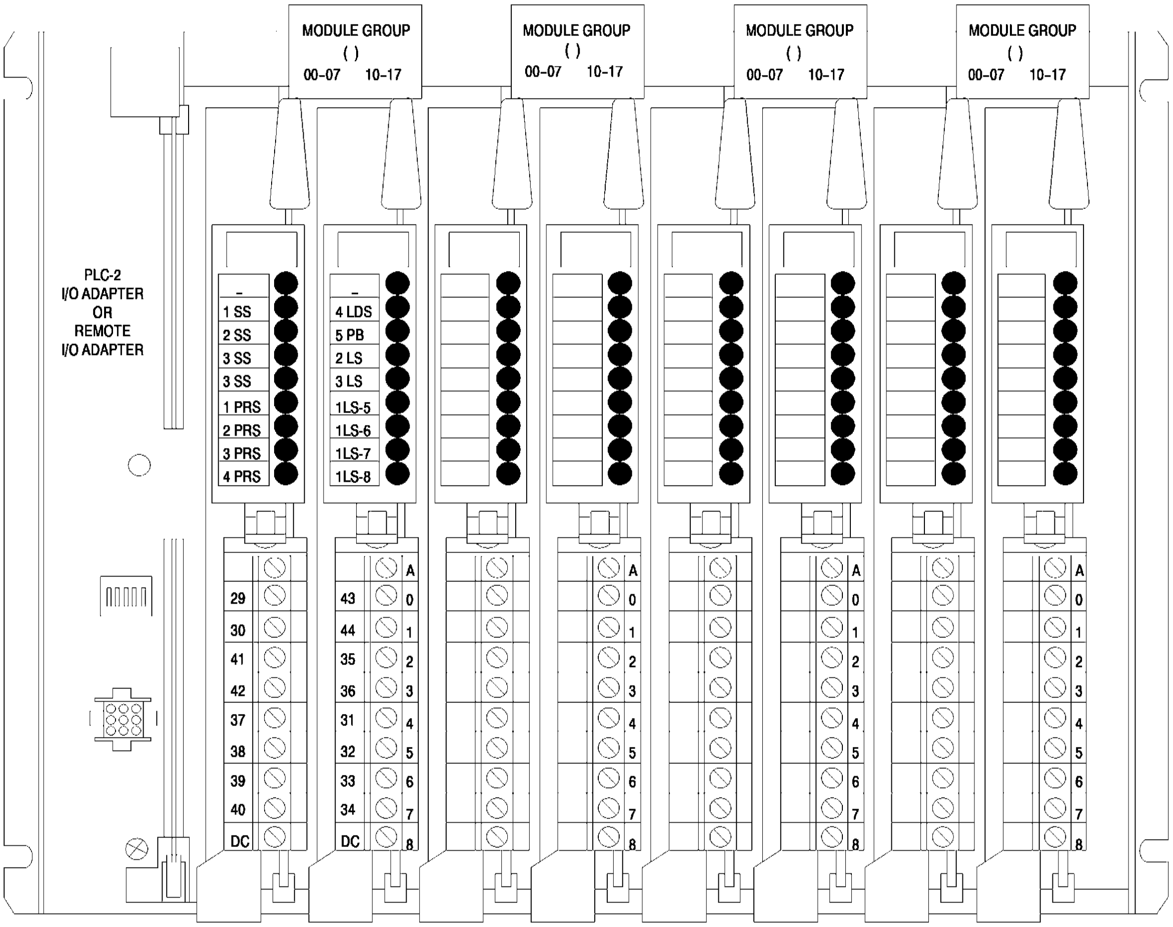
CONNECTION DIAGRAM ADDRESSING

(Publication 5039 – September, 1980)

PAGE ____________ OF ____________

DATE ______________________

PROJECT NAME ______________________________________ DESIGNER



10295-I

ALLEN-BRADLEY
Programmable Controller

DATA TABLE WORD ASSIGNMENTS
(64 - WORD)
(Publications 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
(Publications 5047 - 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 ____

Chapter 9

Operating Instructions

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 in Table 9.A through Table 9.F at the end of this section and include:

- Data Table Configuration (Table 9.A)
- Addressing
- Editing (Table 9.B)
- Directories (Table 9.C)
- Searching (Table 9.D)
- Forcing (Table 9.E)
- Clearing Memory (Table 9.F)

Data Table Configuration

The SEARCH 50 function is used to configure the data table. One of two displays will appear on the industrial terminal screen depending on the industrial terminal and processor being used. The layout and size of memory areas can also be displayed by the SEARCH 54 function. See Table 9.A for a summary of these functions.

1772-LP1 Processor

With this processor and any industrial terminal, the following display will appear when SEARCH 50 is pressed.

```
Number of I/O racks            2
Number of timers/counters    040
Data table size              128
```

Initially, the data table is set to 128 words for 2 I/O racks and 40 timers/counters. The user is prompted by a reverse-video cursor to enter the number of I/O racks (1 to 4) and the number of timers/counters (0 to 111). Once these values are entered, the industrial terminal computes and enters the data table size, and the cursor returns to the first line.

Any time the I/O size is increased or the timer/counter area is decreased, the processor searches for instructions in those areas to be deleted. If an instruction exists in the proposed deleted area, the change will not be permitted and the following message will be displayed: “Instruction Exists In Deleted Area.” To display the rung that is preventing the change in size, press [SEARCH]. At that time, the user can determine whether to keep or delete the instruction.

1772-LP2 Processor

With this processor and the 1770-T1 or -T2 industrial terminal, the SEARCH 50 display will be the same as that of the 1772-LP1 processor.

When the 1770-T3 industrial terminal is used, the following display appears when SEARCH 50 is pressed.

```
Number of 128-word D.T. Sections      1
Number of I/O racks                   2
Number of T/C (if applicable)       040
Data table size                     128
```

Initially, the data table is set to one 128-word section for 2 I/O racks and 40 timers/counters. The user is prompted by a reverse-video cursor to enter the number of 128-word data table sections, the number of I/O racks (1 to 7) and the number of timers/ counters, if applicable.

Any time the number of 128-word data table sections is 1 or 2, the number of timers/ counters must be specified as determined by the formula in section title Sizing the Data table in chapter 8. When there are 3 or more 128-word data table sections, the number of timers/ counters is included in the number of data table sections. Thus, the number of timers/counters is Not Applicable and the industrial terminal will display N/A. When the appropriate information is entered, the industrial terminal will compute and enter the data table size and the cursor will return to the first line.

Any time the I/O size is to be increased or the timer/counter area reduced, the processor searches in the areas to be deleted. If an instruction exists in an area to be deleted, the change will not be allowed and the following message will be displayed: “Data Table Size Change Not Allowed - Instruction Reference To Deleted Area Exists.” To display the rung that is preventing the change, press [SEARCH]. At that time, the user can determine whether to keep or delete the instruction.

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 size of each area is indicated in decimal words.

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, respectively.

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

Any time a digit being entered is not within the proper limits, the message "Digit Out Of Range" will be displayed. The cursor will remain in the same position until a valid digit is entered.

When entering bit instructions with a 1770-T3 industrial terminal and a 1772-LP2 processor, bit addresses larger than 5 digits can be entered provided the EXPAND ADDR key is pressed. If a 5-digit bit address is being entered and a larger bit address is required, the user can press the EXPAND ADDR key at any time provided the last digit has not been entered. If the last digit was entered, the instruction must be re-entered.

Entering word addresses with a 1770-T3 industrial terminal and a 1772-LP2 processor does not require the EXPAND ADDR key. Instead, use leading zeroes when necessary.

Editing

Changes to an existing program can be made through a variety of editing functions. Instructions and rungs can be added or deleted; addresses, data and bits can be changed; temporary end statements can be added; the hex value of illegal opcodes can be displayed. These functions are summarized in Table 9.B.

Inserting an Instruction

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

One way is to press [INSERT][Instruction][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 previous output instruction. If the cursor is on the End statement, however, the instruction will be inserted before the End statement.

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

With the 1772-LP2 processor, bit addresses larger than 5 digits can be entered provided the data table is expanded to a 4 or 5-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.

Removing an Instruction

Only non-output type 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 [REMOVE][Instruction]. Bits and values of word instructions are not cleared. If the wrong instruction is pressed, an "Instruction Does Not Match" message will be displayed.

Inserting a Rung

A rung can be inserted anywhere within a program by pressing [INSERT][RUNG] and entering the instructions. The cursor can be positioned anywhere on the previous rung. The new rung will be inserted after the rung which contains the cursor. Instructions in the new rung cannot be removed until the rung is complete.

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

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 the rung and press [REMOVE][RUNG]. Only bit addresses in the output position are cleared to zero. all other word and bit addresses are not cleared when a rung is removed.

Changing Data of a Word Instruction

The data of any word instruction, except the Arithmetic and Put instructions, can be changed in the program mode without removing and re-entering the instruction. This is done by positioning the cursor on the appropriate word instruction and pressing [INSERT][Data]. When the last digit of the data is entered, the function is terminated. The function can also be terminated before the last digit is entered by pressing [CANCEL COMMAND]. This will enter the value into memory as displayed on the screen.

On-line Data Change

The data of a word instruction, excluding Arithmetic and Put instruction, can be changed while the processor is in the run/program mode.

> **WARNING:** The user must thoroughly understand the program and the consequences of changing machine or operation/process parameters while the processor is operating in the run/program mode. Those who are not familiar with the machine or operation/process are advised to seek the help of someone who is before initiating changes. unexpected machine operation could result in damage to equipment and/or personal injury.

On-line data change is done by positioning the cursor on the appropriate instruction and pressing [SEARCH][5][1]. This key sequence will display the message “On-Line Data Change, Entering Digits” near the bottom of the screen. The new digits will be displayed next to this message as they are entered. Once the new data is displayed, press [INSERT] to enter the data into the program and memory.

to terminate this function, press [CANCEL COMMAND].

Changing the Address of an Instruction

Addresses of bit instructions and some word instructions (Arithmetic, immediate I/O and Put) can be changed in prog mode by re-entering the instruction. To do this in a completed rung, position the cursor on the instruction to be edited and enter the same instruction. The instruction will be displayed with a default address and default values if it has data. The desired address and data can then be entered. If the rung is incomplete, the instruction to be edited must have another instruction after it to change its address.

The address of a word instruction with data, excluding Arithmetic and Put instructions, can also be changed without re-entering the instruction. To do this, position the cursor on the instruction and press [INSERT]. At this point, the first digit pressed will be entered in the first value of the instruction and the reverse-video cursor will move to the next digit of the value.

After entering the first digit, use the [→] key to cursor to the word address and enter the correct address. When using a 1770-T3 industrial terminal and a 1772-LP2 processor, enter leading zeroes before the address if required.

Changing Instructions

An instruction that is already programmed can be changed to another instruction in the prog mode. In a completed rung, this is done by positioning the cursor on the instruction to be changed and entering the desired instruction. The new instruction will be displayed with a default address, and default values if it has data. The address and data can then be entered. If the rung is incomplete, the instruction to be edited must have another instruction programmed after it to change it.

Bit Manipulation

The bit manipulation function is quite useful for start-up, monitoring and troubleshooting a system. With the 1770-T3 industrial terminal, the user can view the 16 bits of a data table word in program mode. In program mode, the user can selectively change the status of any I/O bit.

By pressing [SEARCH][5][3] and the desired word address, the status of all 16 bits of that word will be displayed as well as the force conditions, if any. While the word address is being entered, the [→] and [←]keys can be used to change address digits. Once the address is entered, the user can cursor over to any bit with the [←] and [→] keys and change its status by entering a "1" for on or a "0" for off. The user can also force an input bit or output device on or off in test or run/prog mode as described in table 9-5.

CAUTION: The user must thoroughly understand the program and the consequences of changing machine or operation/process parameters while the processor is operating in the run/program mode. Those who are not familiar with the machine or operation/process are advised to seek the help of someone who is before initiating changes. Unexpected machine operation could result in damage to equipment and/or personal injury.

In the SEARCH 53 function, the user can also display the status of the 16 bits in the next highest or next lowest word address by pressing [v] or [↑] respectively.

To terminate this function, press [CANCEL COMMAND].

Temporary End Statement

Inserting temporary end statements can be helpful during start-up and troubleshooting. The user can insert a temporary end statement at any location in the program. The way this is done depends on the industrial terminal being used.

Only one temporary end statement can be inserted at a time and requires one word of memory. Program instructions beyond the temporary end statement are not scanned or executed.

With a 1770-T1 or -T2 industrial terminal, a temporary end statement is inserted after the cursor's position by pressing [INSERT][-][8]. The temporary end statement will look just like the normal end statement. All program instructions beyond the temporary end statement will not be visible or accessible to the user.

To remove a temporary end statement with a 1770-T1 or -T2 industrial terminal, position the cursor on the end statement and press [REMOVE][-]/[-]. If it is temporary, it will be removed and the subsequent program instructions will be displayed.

With a 1770-T3 industrial terminal, a temporary end statement is inserted after the cursor's position by pressing [INSERT][T.END]. "Temporary End" will be displayed and the subsequent program instructions will be visible and accessible to the user. To remove the temporary end statement, position the cursor on the temporary end statement and press [REMOVE][T.END].

Hex Display of Illegal Opcodes

All instructions in the PLC-2/20 instruction set are defined by a specific binary code called an opcode. If the processor cannot recognize an opcode in the user program, a processor or memory fault can occur.

The 1770-T3 industrial terminal can be connected to the processor to display the rung where the illegal opcode occurred. When the 1770-FD series B keyboard is used, the industrial terminal will display the illegal opcode as ERR in the rung with a 4-digit hexadecimal value above it (See section 13 for information on the hexadecimal numbering system). The hex value simply represents the information at the location in the user program. When the 1770-FD series A keyboard is sued, ERR will be displayed without the hex value.

Any illegal opcode should be corrected by removing and replacing the entire rung. A hard copy printout of the ladder diagram program will be helpful to determine the desired instructions in the rung.

Directories

There are four directories, summarized in Table 9.C, the user can access with a 1770-T3 industrial terminal that display a list of functions and corresponding key sequences.

The help directory, accessed by pressing [HELP], gives a master list of directories and the key sequence to access them. From the help directory, the user can access 3 other directories with the PLC-2/20 processor:

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

The other directories listed in the help directory cannot be accessed. If one of the corresponding keys is pressed, the industrial terminal will issue a "Function Not Available With This Processor" message.

Search Functions

The industrial terminal allows the user to search the program for a specific instruction or address, the first or last rung, the first or last instruction of a rung, and incomplete rungs using the SEARCH key as part of the key sequence. In addition, the industrial terminal allows a single rung or multiple rungs to be displayed. All these functions are summarized in Table 9.D.

First Rung

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

Last Rung

The last rung of the program (end statement) can be located from any position in the program by pressing [SEARCH][v]. The cursor will be positioned on the temporary end statement or the end statement.

First Instruction of a Rung

With the processor in the program mode, the first instruction of the rung containing the cursor can be located by pressing [SEARCH][←]. If the processor is in any other mode, the cursor will move off the screen to the left.

Last Instruction of a Rung

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

Specific Instruction

Any instruction in the program can be located by pressing [SEARCH][Instruction][Address]. With a 1772-LP2 processor, the user must enter leading zeroes before the address if necessary.

Once the key sequence is pressed, this information and an "Executing Search" message will be displayed near the bottom of the screen. The industrial terminal will begin to search for the instruction from the cursor's position to the temporary end statement or the end statement.

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

A 1770-T3 industrial terminal will look past the temporary end statement for the instruction to the end statement, and will continue searching from the beginning of the program to the point where it began the search.

If found, the rung containing the first occurrence of the instruction will be displayed as well as the rungs after it. If the SEARCH key is pressed again, the next occurrence of the instruction will be displayed. When the instruction cannot be located or all instructions have been found, a "Not Found" message will be displayed.

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

Specific Address

All addresses excluding those associated with Examine On and Examine Off instructions, can be located by pressing [SEARCH][8][Address]. The address entered is the word address. for the Output Energize, Latch and Unlatch instructions, the industrial terminal will locate all 16 bit addresses.

Once the key sequence is pressed, this information and an "Executing Search" message will be displayed near the bottom of the screen.

The 1770-T1 or -T2 industrial terminal begins to search for the address from the cursor position to the temporary end statement or end statement. It will not search past the temporary end statement, but continues searching from the beginning of the program to the point where it began the search.

The 1770-T3 industrial terminal searches for the address from the cursor position to the end statement. It looks past the temporary end statement and continues searching from the beginning of the program to the point where it began the search.

If found, the rung containing the first occurrence of the address will be displayed as well as the rungs after it. By pressing [SEARCH] again, the next occurrence of the address will be displayed. When the address cannot be found or all addresses have been found, a "Not Found" message will be displayed.

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

Incomplete Rungs

Incomplete rungs can be entered in the user program only by inserting them while editing. If there are ladder diagram rungs without output instructions, a processor fault will occur when the processor is placed in the run or run/prog mode. The 1770-T3 industrial terminal can be connected to the 1772-LP1 or -LP2 processor to locate and correct only those incomplete rungs that were initiated by the insert rung command.

After placing the processor in the prog mode and selecting PLC-2 mode on the industrial terminal, press [SHIFT][SEARCH]. The industrial terminal will display the first incomplete rung in the program at the top of the screen with the cursor in the output position. In addition, a message will be displayed near the bottom of the screen indicating the number of incomplete rungs.

After entering the correct output instruction, the next incomplete rung can be found by pressing [SHIFT][SEARCH].

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.

Force Functions

The Force On and Force Off instructions are quite useful during start-up and troubleshooting. They allow the user to selectively force an input bit or an output device on or off. To do this, the processor must be in the tester run/program mode.

NOTE: When in test mode, the processor will hold outputs off regardless of attempts to force them on even though the output bit instructions are intensified.

> **CAUTION:** A fast means of shutting off power to output devices must be available if forced output devices control machine motion. An alert and competent person should be stationed at the emergency stop switch during these procedures.

The Force instructions determine the on/off status of input image table bits and output devices by overriding the I/O scan. An input bit will be forced on or off regardless of the actual state of the input device. Likewise, an output device will be either forced on or off regardless of the rung logic. Table 9.E summarizes the force functions for the PLC-2/20 processor.

Forcing Capabilities

Both the Force On and Force Off instructions can address one input image table word and one output image table word at one time. This means up to 4 image table words can be addressed by Force instructions as follows:

- From 1-16 input bits in one input image table word Forced On
- Form 1-16 input bits in one input image table word Forced Off
- From 1-16 output devices of one output image table word Forced On
- From 1-16 output devices of one output image table word Forced Off

Any time an I/O word is forced and a new I/O word is selected for forcing, the force conditions in the previous I/O word are terminated. For example, if any bits in word 110 are forced on and bit 11201 is forced on, all Force On conditions for word 110 will be removed.

In all modes, a “Forced I/O” message will be displayed near the bottom of the screen when bit instructions are forced on or off. In every mode except the program mode, “on” or “off” will be displayed below each forced instruction.

If the industrial terminal is disconnected, the MODE SELECT key is pressed, or line power is lost, all Force instructions are cleared.

Forced Address Display

With the 1770-T3 industrial terminal, the user has the option of viewing a complete list of bit addresses that are forced on and off in the program.

To display a list of the bit addresses forced on and off, either press [SEARCH][FORCE ON] or [SEARCH][FORCE OFF].

Clearing Memory

The user has the option of clearing the data table, user program and messages with various clear memory functions. A summarized description of clear memory functions can be found in Table 9.F.

Data Table Clear

With the 1770-T3 industrial terminal, the user can clear part or all of the data table by pressing [CLEAR MEMORY][7][7] and entering a start and end word address. The user must then press [CLEAR MEMORY] to execute this instruction. The data table will be cleared between and including these two word addresses, excluding the processor work areas.

User Program Clean

With the 1770-T3 industrial terminal, the user can clear part of all of the user program 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.

Total Memory Clear

With any industrial terminal, the complete memory can be cleared by pressing [SEARCH][[↑]] to position the cursor on the first instruction of the program and then pressing [CLEAR MEMORY][9][9]. This resets all the data table bits to zero, excluding the processor work areas.

Table 9.A
Data Table Configuration

Function	Mode [1]	Industrial Terminal	Key Sequence	Description
Data Table Configuration	P	Any	[SEARCH] [5][0] [Numbers] [CANCEL COMMAND]	User enters the number of I/O racks and number of timers/counters to be allocated in the data table. Industrial terminal enters the data table size. To terminate
		1770-T3 (1772-LP2 Processor only)	Same	User enters the number of 128-word data table sections and the number of I/O racks. If the number of 128-word sections is 1 or 2, user enters the number of timer/counters. If the number o 128-word sections is 3 or grater, user does not enter number of timers/counters. Industrial terminal enters the data table size.
Memory Layout Display	Any	1770-T3	[SEARCH] [5][4] [CANCEL COMMAND]	Displays the number of words in the data table area, user program area, message area and unused memory. To terminate

[1] PROGRAM = P, RUN = R, TEST = T, RUN/PROGRAM = R/P

Table 9.B
Editing Functions

Function	Mode [1]	Industrial Terminal	Key Sequence	Description
Inserting a Condition Instruction	P	Any	[INSERT] [Instruction] [Address]	Position the cursor on the instruction that will precede the instruction to be inserted. Then press key sequence.
			[INSERT][←] [Instruction] [Address]	Position the cursor on the instruction that will follow the instruction to be inserted. Then press key sequence.
		1770-T3 (1772-LP2 Processor only)	Same; use [EXPAND ADDR] key	Same as above. When bit address exceeds 5 digits, press the [EXPAND ADDR] key before entering address and enter a leading zero if necessary. For word addresses, enter leading zeroes when necessary.
Removing a Condition Instruction	P	Any	[REMOVE] [Instruction]	Position the cursor on the instruction to be removed and press the key sequence.
Inserting a rung	P	Any	[INSERT] [Instruction]	Position the cursor on any instruction in the preceding rung and press the key sequence. Enter the appropriate instructions to complete the rung.
Removing a rung	P	Any	[REMOVE] [RUNG]	Position the cursor anywhere on the rung to be removed and press the key sequence. **NOTE:** Only bit addresses in the output position are cleared to zero when the rung is removed.
Change data of a word instruction	P	Any	[INSERT] [Data] [CANCEL COMMAND]	Position the cursor on the word instruction whose data is to be changed. Press the key sequence. To terminate.
Change data of a word instruction ON-LINE	R/P	Any	[SEARCH] [5][1] [Data] [INSERT] [CANCEL COMMAND]	Position the cursor on the word instruction whose data is to be changed. Press the key sequence. Press [INSERT] to enter the new data into memory To terminate
Change the address of a word instruction	P	Any	[INSERT] [First Digit] [←] [Address] [CANCEL COMMAND]	Position the cursor on a word instruction with data and press [INSERT]. Enter the first digit of the first data of the instruction. Then use the [←] key as often as needed to cursor up to the word address. Enter the appropriate digits of the word address. To terminate.

Function	Mode [1]	Industrial Terminal	Key Sequence	Description
Bit Manipulation	Any	1770-T3	[SEARCH] [5][3] [Address] [↑] or [↓] [CANCEL COMMAND]	Displays the on/off status of all 16 bits at specified word address and corresponding force conditions if they exist. The [↑] key will display the status of 16 new bits at the next lowest word address. The [↓] ,key displays the status of 16 bits of the next highest word address. To terminate
	R or R/P	1770-T3	[←] or [→] [1] or [0]	Move cursor to the bit to be changed. Enter a "1" to set bit on or a "0" to set bit off.
	T or R/P	1770-T3	See Table 9-5	Force the input bit or output device on or off.
Inserting a Temporary End Statement	P	1770-T1 or T-2	[INSERT] [-|/|-][8]	Position the cursor on the instruction that will precede the Temporary End statement. Press the key sequence and End will be displayed on the screen. The remaining rungs will not be displayed or scanned.
		1770-T3	[INSERT] [T.END]	Position the cursor on the instruction that will precede the Temporary End statement and press the key sequence. Temporary End will be displayed on the screen. The remaining rungs, although displayed and accessible, are not scanned.
Removing a Temporary End Statement	P	1770-T1 or -T2	[REMOVE] [-|/|-]	Position cursor on End statement and press key sequence. If temporary, it can be removed. If it cannot be removed, it is a permanent End statement.
		1770-T3	[REMOVE] [T.END]	Position cursor on Temporary End statement and press key sequence.

[1] PROGRAM = P, RUN = R, TEST = T, RUN/PROGRAM = R/P

Table 9.C
Help Directories

Function	Mode [1]	Industrial Terminal	Key Sequence	Description
Help Directory	Any	1770-T3	[HELP] [CANCEL COMMAND]	Displays a list of the keys that are used with the HELP key to obtain further directories. To terminate
Control Function Directory	Any	1770-T3	[SEARCH] [HELP] [CANCEL COMMAND]	Provides a list of all control functions that use the SEARCH key. To terminate
Record Function Directory	Any	1770-T3	[RECORD] [HELP] [CANCEL COMMAND]	Provides a list of functions that use the RECORD key. To terminate
Clear Memory Directory	Any	1770-T3	[CLEAR MEMORY] [HELP] [CANCEL COMMAND]	Provides a list of all functions that use the CLEAR MEMORY key. To terminate.

[1] PROGRAM = P, RUN = R, TEST = T, RUN/PROGRAM = R/P

Table 9.D
SEARCH Functions

Function	Mode [1]	Industrial Terminal	Key Sequence	Description
Locate first rung of program	Any	1770-T1 or -T2	[SEARCH][↑]	Positions cursor on the first instruction of the program.
		1770-T3	Same	Positions cursor on the Temporary End statement or the first instruction of the program depending on the cursor's location.
Locate last rung of program	Any	Any	[SEARCH][↓]	Positions cursor on the Temporary End statement or the End statement.
Locate first instruction of current rung	P	Any	[SEARCH][←]	Positions cursor on first instruction of the current rung.
Locate last instruction of current ring	Any	Any	[SEARCH][→]	Positions cursor on the output instruction of the current rung.
Locate specific instruction	Any	Any	[SEARCH] [Instruction] [Address]	Locates instruction searched for. Press [SEARCH] to locate the next occurrence of instruction.
		1770-T3 (1772-LP2 Processor only)	Same	Same as above. User must enter leading zeroes when bit address exceeds 5 digits or word address exceeds 3 digits.
Locate specific word address	Any	Any	[SEARCH][8] [Address]	Locates this address in the program. Press [SEARCH] to locate the next occurrence of this address.
		1770-T3 (1772-LP2 Processor only)	Same	Same as above. User must enter leading zeroes when word address exceeds 3 digits.
Incomplete rungs	P	1770-T3	[SHIFT] [SEARCH]	Locates the first incomplete rung in the program. After entering the correct output instruction, the next incomplete rung, if any, can be located by pressing [SHIFT][SEARCH]again.
Single rung display	Any	Any	[SEARCH] [DISPLAY]	Displays the first rung of a multiple rung display by itself. Press key sequence again to view multiple rungs.

[1] PROGRAM = P, RUN = R, TEST = T, RUN/PROGRAM = R/P

Table 9.E
Force Functions

Function	Mode [1]	Industrial Terminal	Key Sequence	Description
Force On instruction	T or R/P	Any	[FORCE ON] [INSERT]	Position the cursor on the bit instruction to be forced on and press the key sequence. The input bit or output will be forced on. **[2]**
Removing a Force On instruction	T or R/P	Any	[FORCE ON] [REMOVE]	Position the cursor on the bit instruction whose force on is to be removed and press the key sequence.
Removing all Force On instruction	T or R/P	Any	[FORCE ONE] [CLEAR MEMORY]	Position cursor anywhere in program and press key sequence.
Force Off instruction	T or R/P	Any	[FORCE OFF] [INSERT]	Position cursor on bit instruction to be forced off and press key sequence. The input bit or output device will be forced off.
Removing a Force Off instruction	T or R/P	Any	[FORCE OFF] [REMOVE]	Position cursor on bit instruction whose force off is to be removed and press key sequence.
Removing all Force Off instructions	T or R/P	Any	[FORCE OFF] [CLEAR MEMORY]	Position cursor anywhere in program and press key sequence.
Forced Address Display	Any	1770-T3	[SEARCH] [FORCE ON] or [SEARCH] [FORCE OFF]	Displays a list of the bit addresses that are forced on and forced off.
			[CANCEL COMMAND]	To terminate

[1] PROGRAM = P, RUN = R, TEST = T, RUN/PROGRAM = R/P
[2] When TEST mode the Processor will hold outputs off regardless of attempts to force them on.

Table 9.F
Clear Memory Functions

Function	Mode [1]	Industrial Terminal	Key Sequence	Description
Data Table Clear	P	1770-T3	[CLEAR MEMORY] [7][7] [Start Address] [End Address] [CLEAR MEMORY]	Displays a start address and an end address field. User enters start and end word addresses to determine boundaries for data table clearing. [⟵] and [⟶] keys are used to cursor along the addresses to correct them. Clears the data table within addressed boundaries.
User Program Clear	P	1770-T3	[CLEAR MEMORY] [8][8]	Clears user program from the position of the cursor to the End statement or Temporary End statement. Does not clear data table or messages. Position the cursor at the desired location in the program and press the key sequence.
Partial Memory Clear	P	Any	[CLEAR MEMORY] [9][9]	Clears user program and messages from position of the cursor. Does not clear data table.
Total Memory Clear	P	Any	[SEARCH][↑] [CLEAR MEMORY] [9][9]	Position the cursor on the first instruction of the program by pressing [SEARCH][↑]. Then press [CLEAR MEMORY][9][9] to clear total memory (data table, user program and messages).

[1] PROGRAM = P, RUN = R, TEST = T, RUN/PROGRAM = R/P

Chapter 10

Peripheral Functions

General

There are several functions that can be performed with a peripheral device connected to channel C of the industrial terminal and the industrial terminal connected to the processor. They include:

- Contact histogram
- Report generation
- Recording programs on magnetic tape
- Ladder diagram dump
- Total memory dump

The contact histogram and report generation functions can operate without a peripheral device connected, however only monitoring on the industrial terminal will be possible.

Peripheral Device Usage

When a peripheral device other than the digital cassette recorder is used, the baud rate for channel C must be set to match the baud rate of the peripheral device. The baud rate is the number of bits per second sent to/from channel C. The baud rate for channel C can be set in one of two ways:

- Setting switches 1,2 and 3 of the switch group assembly on the industrial terminal's main logic board (Table 10.A)

Table 10.A
Switch Groupt 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

- Pressing [RECORD] and a number from 2 to 8 (Table 10.B) on the industrial terminal.

Table 10.B
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

In addition to setting the baud rate, channel C must be on when a peripheral device is used.

With a 1770-T2 or -T2 industrial terminal, channel C is always on. Thus, when a peripheral device is connected to channel C, the peripheral device must be turned on or completely disconnected from channel C.

With a 1770-T3 industrial terminal, in the PLC-2 mode, channel C is on until turned off. For use, with a peripheral device, channel C can be turned on by pressing [RECORD][9]. To turn off channel C, press [RECORD][9] again.

To determine whether channel C is on or off, one of four key sequences can be entered to display the on or off status and baud rate at the bottom right of the screen. When either [RECORD], [SEARCH][4], [SEARCH][6], or [SEARCH][7] is pressed, the following message will be displayed:

Channel C is on/off at XXXX baud.

Contact Histogram

The contact histogram function displays the on/off history of a specific memory bit. It can be monitored on the industrial terminal and can also be sent out to a peripheral printer connected to channel C.

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 accurate to within 00.01 seconds (10 ms resolution).

There are two operating modes for the contact histogram, shown in Table 10.C:

- Continuous, accessed by pressing [SEARCH][6]
- Paged, accessed by pressing [SEARCH][7]

Table 10.C
Contact Histogram Functions

Function	Mode [1]	Industrial Terminal	Key Sequence	Description
Contact Histogram Continuous	R or R/P	Any	[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.
			[CANCEL COMMAND]	To terminate.
Contact Histogram Paged	R or R/P	Any	[SEARCH][7] [Bit Address] [DISPLAY]	Displays 11 lines of the on/off history of the addressed bit in hours, minutes and seconds.
			[DISPLAY]	Displays the next 11 lines of contact histogram. Can obtain a hard copy printout of contact histogram by connecting a peripheral device to Channel C and selecting proper baud rate.
			[CANCEL COMMAND]	To terminate.

[1] PROGRAM = P, RUN = R, TEST = T, RUN/PROGRAM = R/P

After pressing [SEARCH][6] or [SEARCH][7], the user is asked to enter the bit address to be monitored. With a 1770-T3 industrial terminal and a 1‘772-LP2 processor, bit addresses larger than 5 digits do not require leading zeros or the EXPAND ADDR key.

After pressing [DISPLAY], the histogram will display the data 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 in the following format.

-ON or OFF 00:00:00.00-

If the bit is changing states faster than can be printed or displayed, a buffer is maintained to store these changes. If the buffer becomes full, all monitoring stops and a “Buffer Full” message will be displayed, monitoring resumes and a “Buffer Reset” message will be displayed.

The industrial terminal screen can display up to 11 lines of data double-spaced 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 display and stop. All subsequent changes are stored in the buffer until [DISPLAY] is pressed. The data stored in the buffer will then be displayed.

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

Report Generation

The report generation function of the industrial terminal is performed in the PLC-2 mode. It is used to generate messages that contain ASCII and graphic characters as well as data table information. The messages are stored in the processor’s memory after the End statement. Messages can be generated manually by a key sequence each time a message is desired. They can also be generated automatically by programming specific data table bits in the ladder diagram program and entering the automatic report generation command. Each time a message request bit goes from 0 to 1, the corresponding message will be displayed.

The number of messages that can be stored depends on the industrial terminal and keyboard module being used. Up to 6 messages of any length can be stored in the processor’s memory when one of the following industrial terminals is used:

- 1770-T1 industrial terminal
- 1770-T2 industrial terminal
- 1770-T3 industrial terminal with a 1770-FDC keyboard module (series A/rev.A).

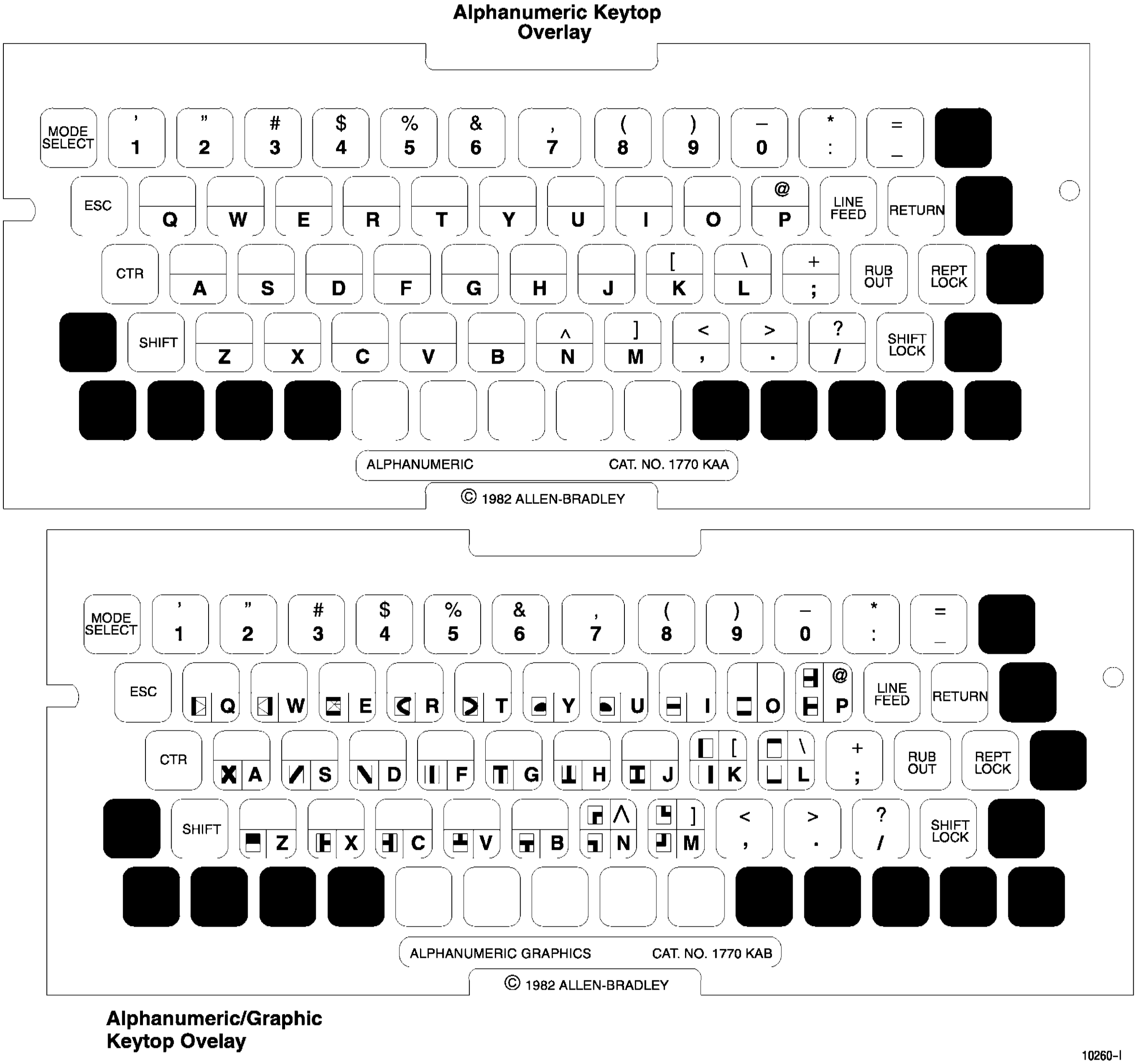
When using a series A/rev. B or later 1770-FDC keyboard module, an additional 64 messages, totalling 70,can be stored in the processor’s memory.

The way the messages are numbered and the bits used to control message generation are described below.

Messages can be entered into memory from either the industrial terminal or a peripheral device connected to channel C. If the industrial terminal is used, one of two keytop overlays can be used, depending on whether graphic characters are desired (Figure 10.1):

- Alphanumeric Keytop Overlay (cat. no. 1770-KAA)
- Alphanumeric/Graphics Keytop Overlay (cat. no. 1770-KAB)

Figure 10.1
Alphanumeric Overlays



Alphanumeric Keytop Overlay

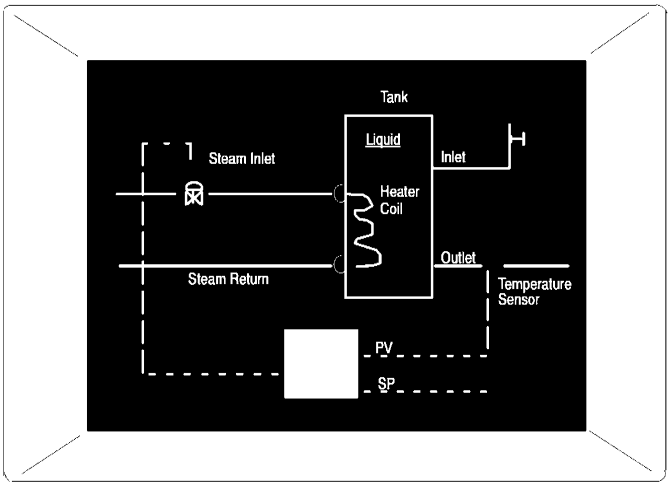
Alphanumeric/Graphic Keytop Ovelay

The industrial terminal screen size is an 80 x 24 format: 80 columns across by 24 lines down. A typical message using graphic and alphanumeric characters is shown in Figure 10.2.

Message Numbering and Message Control Words

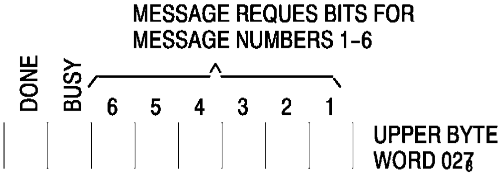
Six messages, numbered 1-6, are available with all industrial terminals. For automatic report generation, bits 02710-02715 are used to request messages 1-6 respectively. Bits 02716 and 12717 are the busy and done bits, respectively, for the six message request bits (Figure 10.3).

Figure 10.2
Typical Graphics/Alphanumeric Message



10261-I

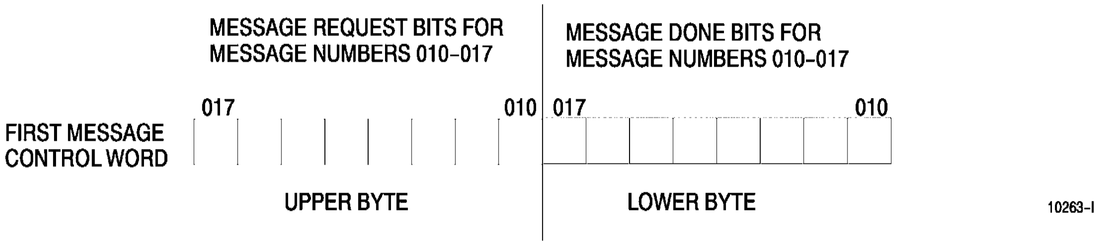
Figure 10.3
Message Generation (6 Message)



10262-I

64 additional messages are available with a series A/rev B 1770-FDC keyboard module. They are numbered octally from 010-017, 110-117, 210-217, 310-317, 410- 417, 510-517, 610-617 and 710-717. These messages can be requested for automatic report generation by defining eight message control words. Each message control word is used for 8 messages. The first message control word is used for message numbers 010-017. The bits in the upper byte of the message control word are used to request messages 010-017 (Figure 10.4). The corresponding bits in the lower byte are used as done bits for message numbers 010-017. This arrangement is the same for all message control words; that is, the second message control word is used for message numbers 110-117, the third for message numbers 210-217, and so on.

Figure 10.4
Message Generation (64 Additional Messages)



The 8 message control words are defined by establishing a two-word message called message 0. First access report generation by pressing [RECORD][DISPLAY]. Then, changing to an alphanumeric overlay and pressing [M][S][,][0][RETURN], the following prompt will be displayed:

Message Control Word (Y Digits Required):

The "Y" will be displayed as 3, 4 or 5 depending on the data table size. After this prompt, the user enters a 3, 4 or 5 digit word address using leading zeros if necessary. The industrial terminal will compute and enter the end word address. Any area of the data table, except processor work areas and input image table, can be used for message control words.

CAUTION: Do not use message control words for any purpose other than report generation. If output image table words are used, unexpected machine motion could occur. Damage to equipment and/or personal injury could result.

Report Generation Commands

There are 5 manual report generation commands used to store, print, record and delete messages and to display an index of messages. After pressing [RECORD][DISPLAY] on the PLC-2 overlay, these commands can be accessed.

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][,][Number][RETURN] on the alphanumeric overlay or the peripheral device. The number represents the message number to be entered and stored in memory. It can be a number from 1-6, 010-017, 110-117, 210-217, 310-317, 410-417, 510- 517, 610-617 and 710-717.

After pressing the key sequence, the user is prompted to enter the desired message by the "Ready For Input" message.

When entering messages, there are several keys and industrial terminal control codes that are used to move through the display and perform a variety of functions (table 10-4 and 10-5). For example, graphics capability can be accessed by the control code, CTRL P5G. In addition, standard ASCII control codes can be used with the industrial terminal (Table 10.F). These codes, although not displayed, can be interpreted and acted on by a peripheral device connected to channel C.

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

The control code, [CTRL][P][Column #][;][Line #][A], should be used for cursor positioning when possible to conserve memory. Note that columns and line numbers begin at zero instead of one. For example, [CTRL][P][3][9][;][1][0][A] uses 3-1/2 words of memory, storing [CTRL][P] in one byte and each remaining character is one byte. Cursoring to this location using [LINE FEED] and space keys would use many more words of memory.

Table 10.D
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] if 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.

Table 10.E
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 ON.
[CTRL][P][4][P]	Turns Channel C OFF.
[CTRL][I]	Horizontal tab that moves the cursor to the next preset 8th position.
[CTRL][K]	Clears the screen from the 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]

Table 10.F
ASCII Control Codes

Control Code [1]	Display	ASCII Mnemonic	Name
CTRL 0	N_U	NUL	Null
CTRL A	S_H	SOH	Start of Header
CTRL B	S_X	STX	Start of Text
CTRL C	E_X	ETX	End of Text
CTRL D	E_T	EOT	End of Transmission
CTRL E	E_Q	ENQ0	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 Text Buffer
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

Control Code [1]	Display	ASCII Mnemonic	Name
CTRL-	G_S	FS	Group Separator
CTRL.	R_S	RS	Record Separator
CTRL/	U_S	US	Unit Separator
DELETE	D_T	DEL	Delete
ESCAPE	E_C	ESC	Escape

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

By using delimiters, the user can enter messages that will report the data table value of a word or the on/off status of a data table bit (Table 10.G). A delimiter is entered before and after the bit or word address. As many addresses as needed can be entered sharing the same delimiter, such as *XXX*XXX*XXX*. When using the asterisk (*) delimiters with a 1770-T3 industrial terminal, the user is limited to entering 3-digit word addresses and 5-digit bit addresses.

Table 10.G
Address Delimiters

Delimiter Format	Explanation	Message Report Format
XXX	Enter 3-digit word address between delimiters.	Displays BCD value at assigned word address.
XXXX	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 3,4, or 5-digit word address between delimiters.	Displays the BCD value at assigned word address.
!XXX!	Enter 3,4 or 5-digit word address between delimiters.	Displays the 4-digit hex value at assigned word address.
&XXXX&	Enter 3,4, or 5-digit word address and a "1" for upper byte or a "0" for lower byte between delimiters.	Displays the octal value at the assigned byte address.
XXXXX	Enter 5,6 or 7-digit bit address between delimiters.	Displays the on or off status of the assigned bit address.

To terminate the message store command,press [ESC] on the alphanumeric overlay or the peripheral device. To return to the ladder diagram display, press [ESC] again. The user can also press [CANCEL COMMAND] on the PLC-2 overlay to terminate message store and to return to the ladder diagram display.

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][,][Number][RETURN] on the alphanumeric overlay or the peripheral device. The number in this key sequence refers to the message number (1-6, 010-1‘7, 110-117, 210-217, 310-317, 410-417, 510-517, 610-617 and 710-717).

The message print command is self-terminating. Once [ESC] or [CANCEL COMMAND] is pressed, the ladder diagram display will resume.

Message Report

Accessible in any mode, the message report command is used to print a message with the data table value or bit status that corresponds to the address between the delimiters. This command is accessed by pressing [M][R][,][Number] [RETURN]. The Number in the key sequence refers to the message number (1-6, 010-017, 110-117, 210-217, 310-317, 410-417, 510-517, 610-617 and 710-717).

The message report command is self-terminating. When [ESC] or [CANCEL COMMAND] is pressed the ladder diagram display will resume.

Message Delete

Accessible only in the program mode, the message delete command is used to clear messages from memory. This command is accessed by pressing [M][D][,] [Number] [RETURN] on the alphanumeric overlay or the peripheral device. The number in this key sequence refers to the message number (1-6, 010-017, 110- 117, 210-217, 310-317, 410-417, 510-517, 610-617 and 710-717).

The message delete command cannot be terminated before completion. It will self terminate after the message has been cleared from memory. With a 1770-FCC keyboard module ser. A/rev.C or later, or a 1770-FDC keyboard module, a "Message Deleted" prompt will be displayed once a message has been deleted. To return to the ladder diagram display, press [ESC] or [CANCEL COMMAND].

Message Index

Accessible in any mode, the message index command displays a list of message numbers and the number of memory words used for each message. In addition, the number of unused memory words will be displayed at the end of this list.

The message index command is accessed by pressing [M][I][RETURN] on the alphanumeric overlay or the peripheral device. This command cannot be terminated before completion. It will self-terminate after the complete list is displayed. To return to the ladder diagram display, press [ESC] or [CANCEL COMMAND].

Automatic Report Generation

Messages stored in memory can be generated automatically provided the user has programmed ladder diagram rungs to energize the message request bits.

Automatic report generation can be accessed in the test, run or run/program modes. It is accessed by pressing [SEARCH][4][0] or by pressing [RECORD] [DISPLAY] and then [M][R][RETURN] on the alphanumeric overlay or peripheral device. With a 1770-T3 industrial terminal, automatic report generation can also be activated upon power-up by setting parity switches 4 and

5 UP on the industrial terminal's main logic board. Refer to the Industrial Terminal System User's Manual (publication 1770-805) for switch location and settings.

Once automatic report generation is activated, the message request bits are scanned for a zero to one transition. Each time one of the message request bits goes true, the corresponding message will be reported automatically.

For messages 1-6, bit 02716 (busy bit) is set on when one or more of these messages is requested and will remain on until all messages have been printed. Once all messages are generated, bit 02716 is set off and bit 02717 (done bit) is set on. Bit 02717 will stay on for 300 ms and will then be set off.

Automatic report generation for the additional 64 messages works a bit differently. Each of the 64 messages has its own done bit. The done bits are located one byte lower than the message request bits in the message control word (Figure 10.4). After one of the 64 messages has been reported, the done bit is set until the user program resets the request bit.

Automatic report generation can be terminated by pressing [ESC] on the alphanumeric overlay or peripheral device. To return to the ladder diagram display, press [ESC] again. The user can also press [CANCEL COMMAND] on the PLC-2 overlay to terminate automatic report generation and to return to the ladder diagram display.

Example Programming

Using latch and unlatch instructions, report generation can easily be programmed to handle multiple or simultaneous message requests. Simultaneous requests are handled by priority: lower message numbers are handled first. Figure 10.5 shows a sample program that can be used to generate each message. When the event occurs, the request bit is latched on. When the event is over and the message is reported, the done bit comes on and the request bit is unlatched.

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 against the tape.

Dumping 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, the command should be accessed only in the program mode because the data table is constantly changing in other modes.

To dump the complete memory onto the cassette tape, position the cursor anywhere on the first rung and rewind the tape to the beginning. The cassette dump command is then activated by pressing [RECORD][0] on the PLC-2 overlay, and by pressing [RECORD ON TAPE] on the cassette recorder.

As memory is being recorded, the industrial terminal will count and display the number of data table words and program words that were recorded on tape. With a 1770-FCC keyboard module prior to ser.A/rev. C, this information is displayed as follows:

```
XXXX P
XXXX D
```

With a 1770-FCC keyboard module ser.A/rev. C or later or a 1770-FDC keyboard module, this information is displayed as follows:

```
XXXX Program Words
XXXX Data Table Words
```

The cassette dump command is self-terminating. At that time the user has the option of automatically verifying memory against the tape or terminating the cassette function by pressing [CANCEL COMMAND].

Loading from Cassette Tape

Accessible in the program mode only, the cassette load command is used to load the contents of a tape into processor memory. This command is accessed by pressing [RECORD][0] on the PLC-2 overlay and by pressing [READ FROM TAPE] or [PLAY] on the cassette recorder. To load the complete memory, rewind the tape to the beginning of the program.

As memory is being loaded, the number of data table words and program words will be counted and displayed. When loading is complete, the user has the option of automatically verifying the tape against the processor memory or terminating the cassette function by pressing [CANCEL COMMAND].

Automatic Verification

This command can be accessed immediately after dumping or loading memory to/from the cassette tape to verify processor memory against the tape. This is the only way the data table can be verified. To verify the data table, the processor must be in the program mode.

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. The user has the option of displaying and locating any program errors or terminating the cassette function by pressing [CANCEL COMMAND].

Program Verification

Accessible in any mode, this command is used to verify the user program and messages in memory with the version of the cassette tape. Although the data table size and configuration are checked, the data table is not verified.

This command is accessed by pressing [RECORD][1] on the PLC-2 overlay and by pressing [READ FROM TAPE] or [PLAY] on the cassette recorder. To verify the complete program, rewind the tape to the beginning of the program.

When verification is complete, the command will self-terminate and display the number of program errors. The user has the option of displaying and locating any program errors or terminating the cassette function by pressing [CANCEL COMMAND].

Displaying and Locating Errors

After automatic or program verification, the user can press [SEARCH][9] on the PLC-2 overlay to display the number of program and data table errors and whether the data table was verified. Up to 19 errors can be displayed (counted).

The user can search for and locate each program error by pressing [SEARCH] and a number from [0][1] to [1][9]. Each time an error is searched for, the rung containing the error will be displayed with the cursor positioned on the erroneous instruction.

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 the processor's memory.

The data cartridge recorder can be operated just like a Digital Cassette Recorder (cat. no. 1770-SA) using both the industrial terminal and the recorder's control panel. The data cartridge recorder can also be operated using just the 1770-T3 industrial terminal. Whichever method is selected, the baud rate switch on the data cartridge recorder must be set to 1200.

It should be noted that when a data cartridge tape is installed with power applied, the data cartridge recorder automatically fast-forwards and rewinds the tape to correct tape tension. This process should not be confused with the dump, load and verify operations.

Operating the data cartridge recorder using just the 1770-T3 industrial terminal is discussed in the following sections. For operating like a digital cassette recorder, refer to section titled Digital Cassette Recorder.

Dumping to Data Cartridge

The processor's data table, user program and messages can be dumped onto a data cartridge and automatically verified by a single command from the 1770-T3 industrial terminal. The processor should be in program mode to ensure the data table is not changing.

With the cursor positioned on the first rung of the program, the cartridge dump command is initiated by pressing [RECORD][SHIFT][B].

As memory is being recorded, the industrial terminal will count the number of program words and data table words and display them as follows:

```
XXXX Program Words
XXXX Data Table Words
```

After memory has been recorded, the tape automatically rewinds to begin verifying the processor's memory against the tape. During verification, the number of program and data table words are counted and displayed.

Once verification is complete, a message will be displayed indicating the number of program errors. The user has the option of displaying and locating errors. The procedure for this is described in section titled Displaying and Locating Errors.

The cartridge dump command can be aborted at any time by pressing [CANCEL COMMAND].

Loading from Data Cartridge Tape

The processor's memory can be loaded from a data cartridge and automatically verified by pressing [RECORD][SHIFT][A] on the 1770-T3 industrial terminal. The processor must be in the program mode.

As memory is loaded, the number of program and data table words will be counted and displayed. When loading is complete, the tape automatically rewinds to begin verifying the tape against the processor's memory. While the tape and memory are being verified, the number of program and data table words will be counted and displayed. When verification is complete, a message indicating the number of program errors will be displayed. The user can search for and locate these errors as described in section titled Displaying and Locating Errors.

The cartridge load command can be aborted at any time by pressing [CANCEL COMMAND].

Data Cartridge Verification

This command is used to verify the processor's user program and messages with the version on the data cartridge tape. Although the data table size and configuration are checked, the data table is not verified.

With the processor in any mode, the data cartridge tape can be verified by pressing [RECORD][SHIFT][C] on the 1770-T3 industrial terminal. As the tape is being verified, the number of program words and data table words are counted and displayed. Then, when verification is complete, the number of program errors will be displayed. Program errors can be located and displayed as described in section titled Displaying and Locating Errors.

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

Ladder Diagram Dump

Accessible in any mode, the ladder diagram dump command is used to print out a hard copy of the user program using a peripheral printer that is connected to channel C.

This command is accessed by pressing [SEARCH][4][4] on the PLC-2 overlay. The printout will begin from the current rung, allowing all or part of the program to be printed.

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

Total Memory Dump

The total memory dump command can be activated only with the 1770-T3 industrial terminal. 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 [SEARCH][4][5] on the PLC-2 overlay and will print out the complete memory, regardless of rung position.

The data table will be printed first in the format shown in Figure 10.6. The bit pattern for each data table word will be printed (in hexadecimal). The hexadecimal numbering system (Table 10.H) is a convenient way of representing 16 binary digits using 4 hexadecimal digits. There are 16 hexadecimal digits that range from 0-9 and A-F. Each hexadecimal digit displayed in the printout represents a 4-bit binary pattern. For example, the hexadecimal number $2A4F_{16}$ is equivalent to the following 16-bit pattern:

2	A	4	F
0010	1010	0100	1111

Figure 10.5
Example Program

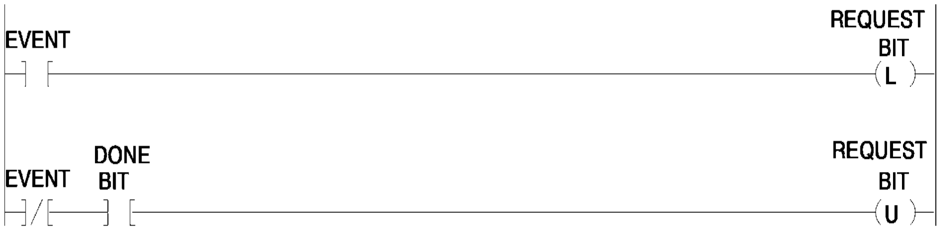


Table 10.H
Hexadecimal Numbering System

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

For more information on the hexadecimal numbering system, refer to chapter 13 of this manual.

The data table printout will be followed by the user program in ladder diagram format and the messages identified by number.

When the printout is complete, this command is automatically terminated. the user can terminate the command prior to completion by pressing [ESC] on the peripheral printer of [CANCEL COMMAND] on the PLC-2 overlay.

Figure 10.6
Data Table Printout

DATA TABLE

WORD ADDR				DATA				
00010	26C1	A4FF	952B	F073	D572	43CE	FFFF	300F
00020	ECCB	9A00	4621	002F	5101	024C	3128	AC0B
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Chapter 11

Special Programming Techniques

General

There are several programming techniques that offer versatile control of the application program. They include:

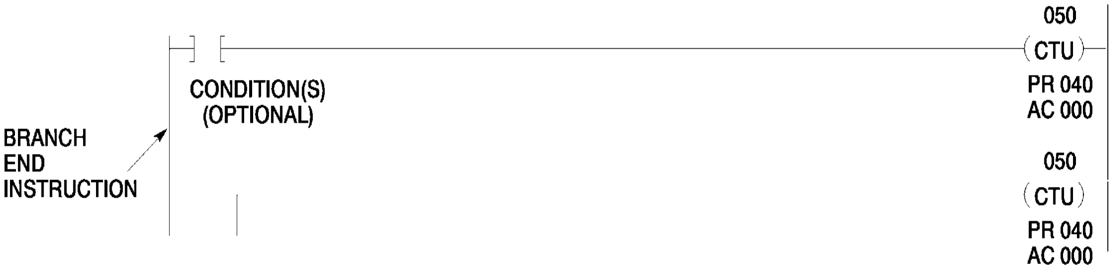
- Scan Counter
- Block Transfer
- Remote Fault Zone Programming
- One–Shot

Scan Counter

The scan counter is a programming technique that can be used where timing may be an important consideration in the application program. In a local system, it is used to count program and/or I/O scans. In a remote system, it is used to count program scans only.

The scan counter is programmed using two rungs (Figure 11.1). The first rung contains optional condition instructions and a CTU instruction, whose preset value equals the number of scans to be counted. The second rung is started with a Branch End instruction to open the rung and is ended with the same CTU instruction.

Figure 11.1
Scan Counter



When the first rung goes true, bit 17 of the CTU instruction is set to one and the 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 enable bit (bit 17) to zero so the CTU can increment its count during the next scan. The CTU will continue to increment past the preset value unless it is reset by a CTR instruction.

Block Transfer

Block transfer can be performed with the PLC–2/20 processor (cat. no. 1772–LP1 or LP2), the 1771 I/O structure and any 1771 I/O module with block transfer capability.

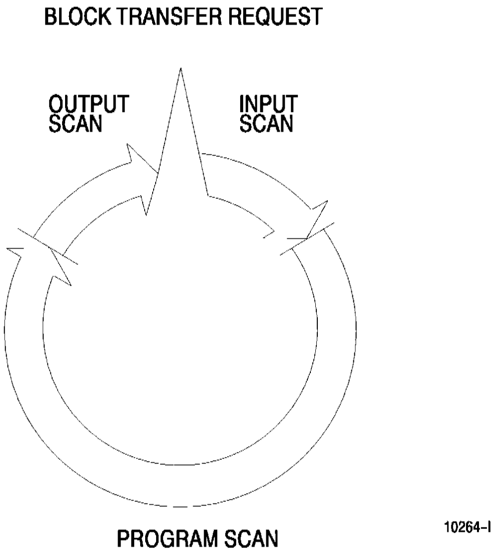
Block transfer is a programming technique used to transfer many words of data in one scan using intelligent 1771 I/O modules such as the analog, thermocouple or encoder/counter modules. It replaces single transfer programming where one word of data is transferred per scan. Thus, block transfer reduces the amount of ladder diagram programming.

Block transfer can be performed as a read,write or bidirectional operation, depending on the I/O module being used. During a read operation, data is read into the processor's memory from the I/O module. During a write operation, data is written out to the I/O module from the processor's memory. Bidirectional block transfer modules can perform both the read and write operations.

The number of words transferred in one scan can range from 1 to 64, depending on the I/O module being used. The time required to perform block transfer depends on the number of words being transferred and whether the block transfer module is located in a local or remote I/O rack.

When block transfer is requested in the ladder diagram the output part of the next I/O scan is interrupted (Figure 11.2). If the I/O module is ready for block transfer, block transfer will be performed. When the transfer of data is complete, the output image table byte of the module's location is duplicated in the corresponding input image table byte. The processor then continues its normal scan.

Figure 11.2
Block Transfer Request



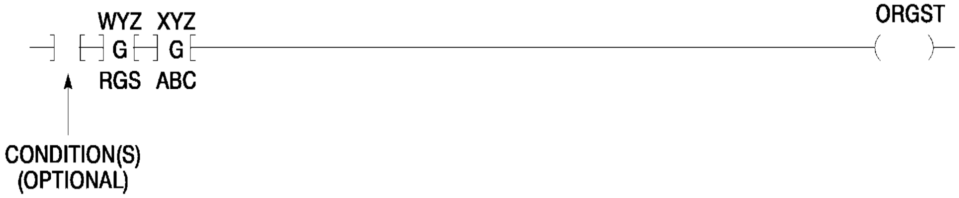
The I/O module may not be ready for block transfer 100% of the time. For a short time, the I/O module must perform internal checks on its own operation. If block transfer is requested when the I/O module is not ready, the read or write bit in the corresponding input image table byte is zeroed out. The processor will then continue with its normal I/O scan.

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

Block Transfer Rungs

The block transfer rung must be programmed in a certain format (Figure 11.3). It consists of conditional instructions that are optional, two Get instructions and an Output Energize instruction.

Figure 11.3
Block Transfer Rung



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

10264-I

First Get Instruction

The first Get instruction (Figure 11.3) is used to identify the location of the block transfer module. It must be assigned the first timer/counter accumulated address in the data table. This address will vary depending on the data table configuration (Table 11.A). When more than one block transfer module is used, consecutive timer/counter addresses must be assigned.

Table 11.A
Timer/Counter Addresses

# I/O Racks	First Timer/Counter Accumulated Address
1	020
2	030
3	040
4	050
5	060
6	070
7	200

As its "data", the first Get instruction stores the location of the block transfer module by rack number, module group number and slot number. when block transfer is performed, the processor searches the timer/counter area for a match of the module's rack, group and slot number. The slot number entered can either be a "0" to indicate the left slot of a module group or a "1" to indicate the right slot of a module group. For single slot modules, the slot number can be "0" or "1" depending on the module's location. For double slot modules, the slot number must always be entered as a "0".

NOTE: Double slot modules, must occupy a complete module group. Overlapping of module groups is not permitted.

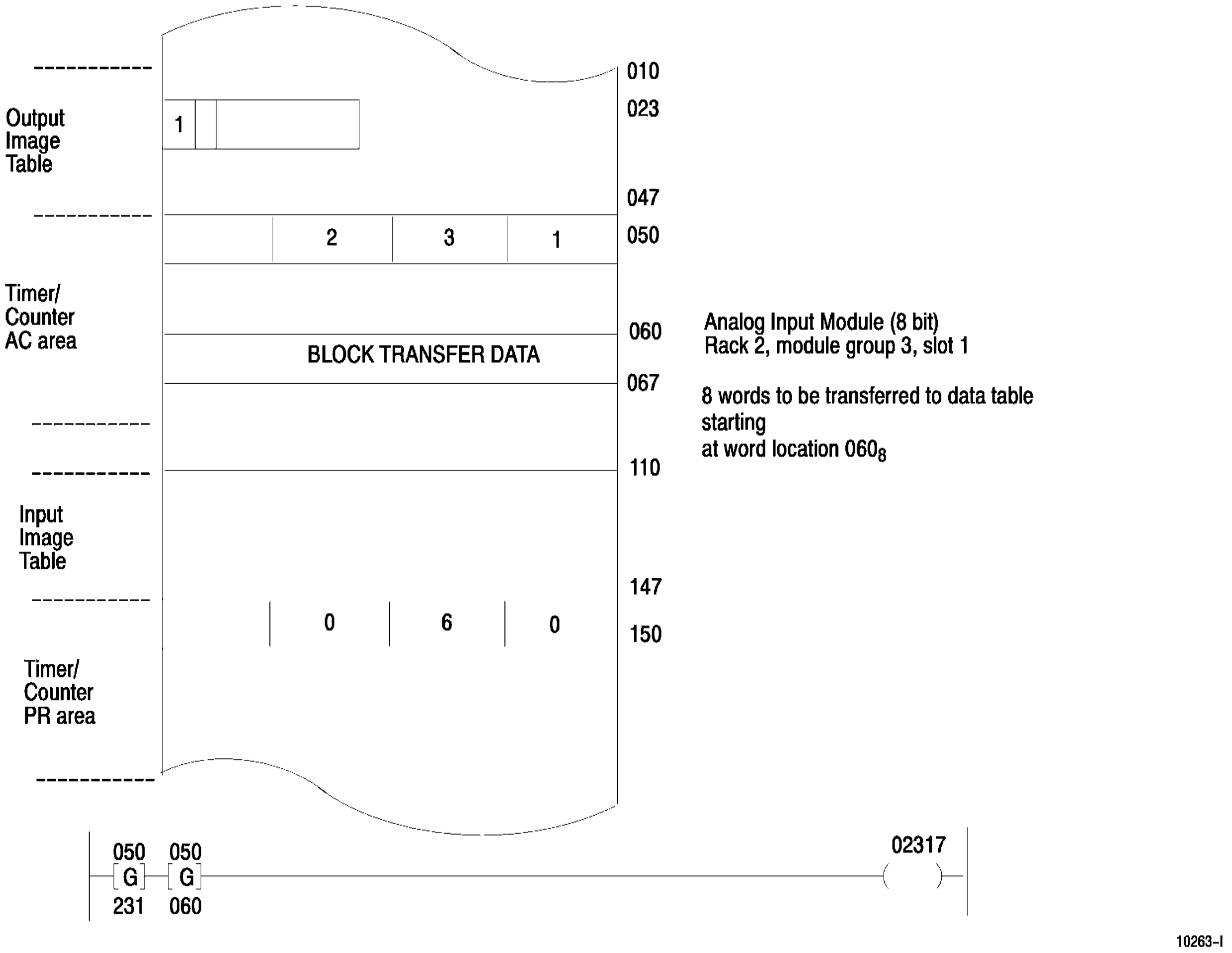
Second Get Instruction

The second Get instruction (Figure 11.3) must be assigned the preset timer/counter address in the data table, 1008 words above the first Get address.

As its "data", the second Get instruction stores a data table address that designates the beginning area reserved for block transfer data. During a read operation, data is loaded into consecutive word locations starting with the designated address. During a write operation, data is sent to the module from consecutive word locations starting with the designated word address.

When reserving an area for block transfer data, the user must select an appropriate address to ensure all block transfer data will be located in one 1008 word accumulated area or preset area (Figure 11.4). If an address is selected so the block transfer data cannot be located in a single 1008 word area, the processor will jump 1008 words to the next PR or AC area to continue the block transfer area. If timers or counters are stored in these next areas, the block transfer data will write over these values which may cause undesirable operation. For example, if 0738 in Figure 11.4 were selected as the beginning address for block transfer data, data would be read into words 0738–0778 and 2008–2028.

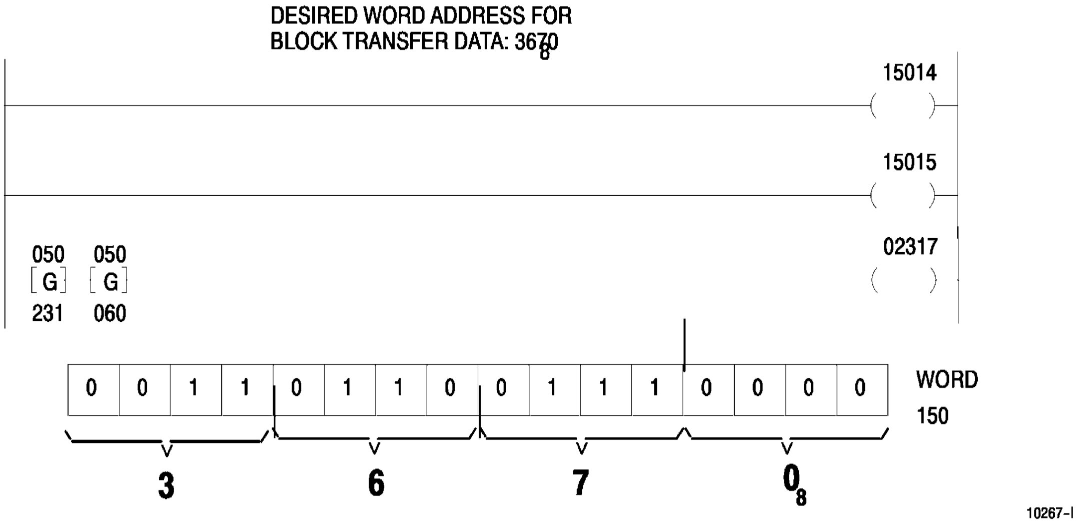
Figure 11.4
Selecting Block transfer Data Area



With a 1772–LP2 processor, the data table address for block transfer data can be larger than 7778, however the Get instruction cannot accept more than 3 digits in its data field. To use a word address up to 77778 for block transfer data, the user must set a bit pattern using bits 14–16 of the Get instruction's word address to correspond to the fourth digit (Figure 11.5). These bits can be set on using unconditional Output Energize or Output Latch instructions. These rungs should be programmed immediately before the block transfer rung.

Figure 11.5
4-digit Word Address for Block Transfer Data

DESIRED	BIT		
NUMBER	16	15	14
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



Output Energize Instruction

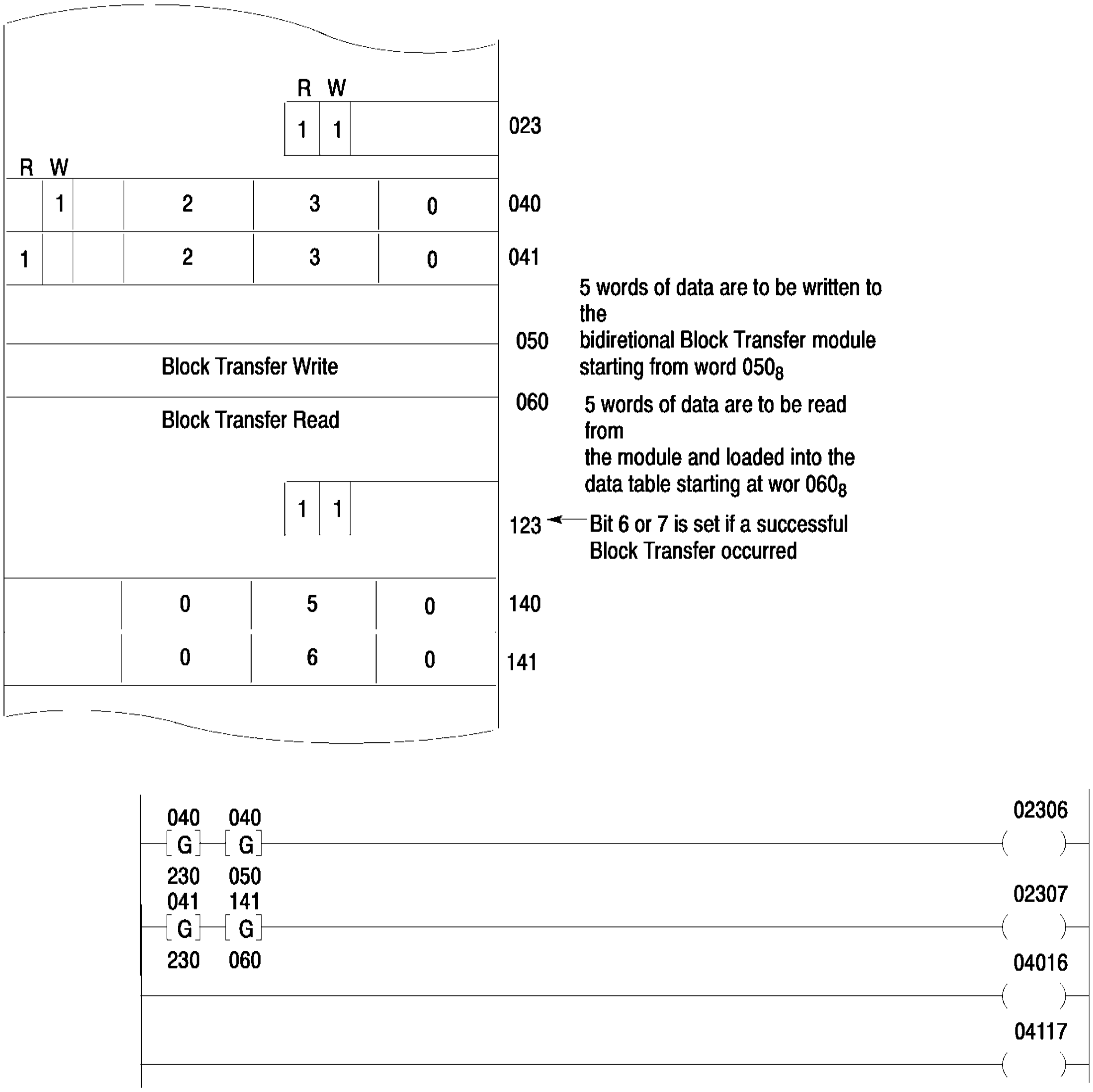
The Output Energize instruction (Figure 11.3) is used to initiate block transfer. It is given an address that indicates the module location and the type of block transfer operation. The first digit of the address is always "0" for output byte, even though an input or output block transfer module can be used. The next three digits identify the module location by rack, group and slot. The last digit is either a "6" for a write operation or a "7" for a read operation.

When block transfer is successfully completed, the output image table byte is mirrored in the corresponding input image table byte.

Bidirectional Block Transfer

A bidirectional operation requires one rung for a read operation and one rung for a write operation. consecutive timer/counter addresses should be selected for the Get instructions (Figure 11.6). For example, the first Get instruction of both rungs should be assigned consecutive word addresses such as 0408 and 0418. Both will have the same "data" to identify the module location.

Figure 11.6
Bidirectional Block Transfer Rungs



5 words of data are to be written to the
bidiretional Block Transfer module starting from word 050_8

5 words of data are to be read from
the module and loaded into the data table starting at wor 060_8

Bit 6 or 7 is set if a successful Block Transfer occurred

10268-I

The second Get instruction of both rungs will be assigned addresses 1008 words above the first Get instructions. As “data”, they will store different word locations for block transfer data such as 0508 and 0608.

The Output energize instruction is addressed for a read operation is one rung and a write operation the other rung.

To let the processor know whether a read or write operation is to be performed, bit 16 or 17 of the first Get instructions must be set on (Figure 11.6). This can be done by programming an Output Energize or Output Latch instruction unconditionally. For example, bit 16 of word 0408 is set on to indicate a write operation. Bit 17 of word 0418 is set on to indicate a read operation.

For bidirectional block transfer, use the default size set by the module for the number of words to transfer. Refer to the appropriate documentation for the module for programming techniques.

Support Rungs

There are additional techniques that can be used to support the block transfer operation:

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

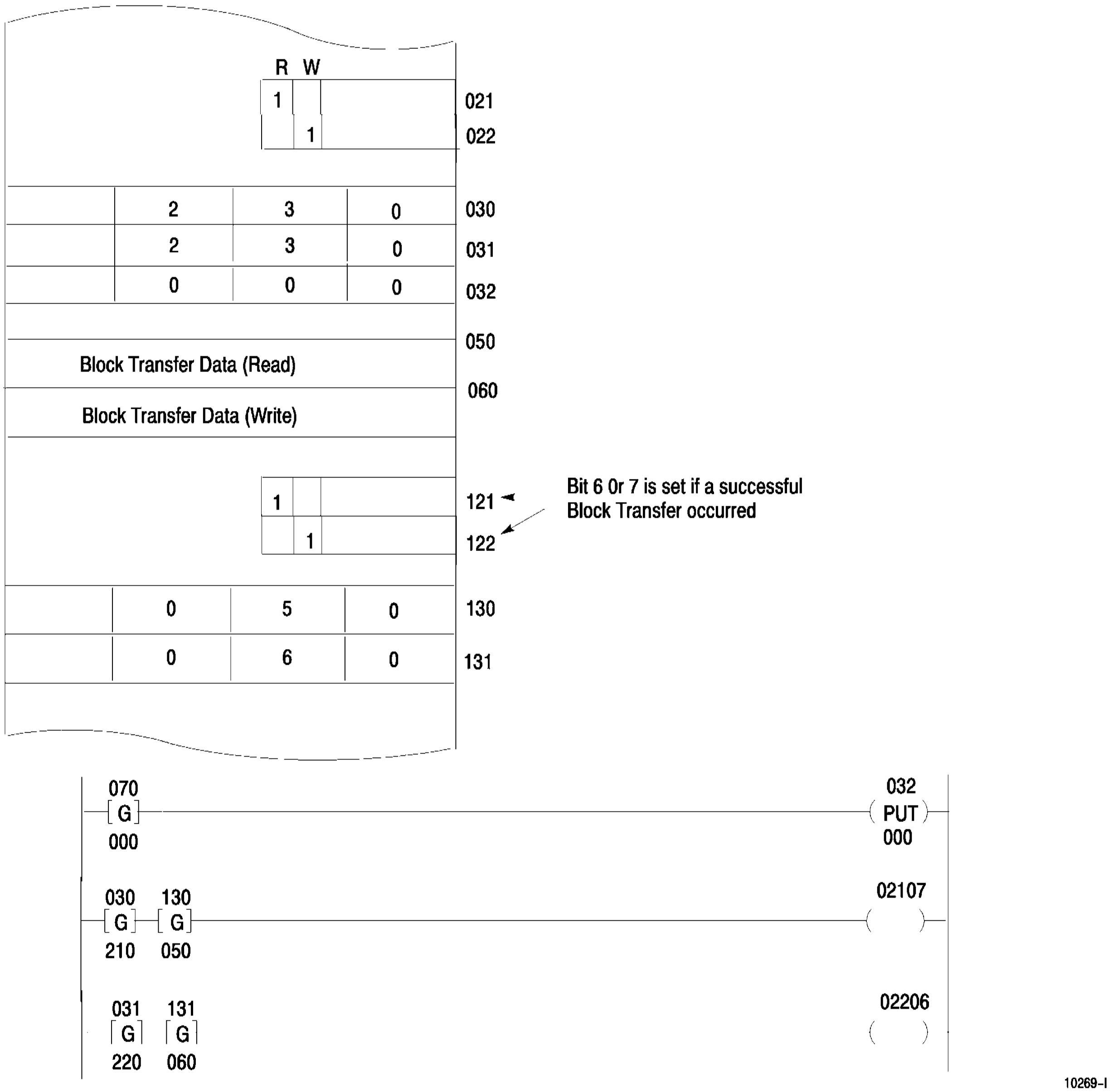
Of these support rungs, buffering data must be programmed to assure the block transfer data is valid. Other techniques, such as an Immediate Output instruction or a scan monitor, can also be programmed. The IOT instruction is used to request block transfer more than once per scan by assigning it the word address corresponding to the module’s location. A scan monitor is used to monitor the number of scans that have occurred between each block transfer operation. For programming information on the scan monitor, refer to the respective user’s manual for the block transfer module.

Also, if the number of words to be transferred or the location of block transfer data must change at different times, special programming techniques must be used. Refer to the appropriate documentation for the module for these programming techniques.

Loading Zeros

One rung that can be programmed is a Get/Put transfer (Figure 11.7). It is used to load zeros into the timer/counter accumulated word, immediately following the last Get address that identifies the module location. The Get/Put transfer is programmed by selecting an unused storage word for the Get instruction and entering zeros for its BCD value. The Put instruction is assigned the timer/counter accumulated address immediately following the last Get address that identifies the module location. When block transfer is requested, the processor starts at the first timer/counter address and searches the timer/counter area until it finds the module's rack, group, and slot number or timer/counter address with all zeros. By loading zeros into this consecutive timer/counter word, the processor will not search the remaining timer/counter area. In addition, the processor will not find another BCD value that may, by chance, be the same number as the rack, group and slot number.

Figure 11.7
Loading Zeros into Timer/Counter Accumulated Word



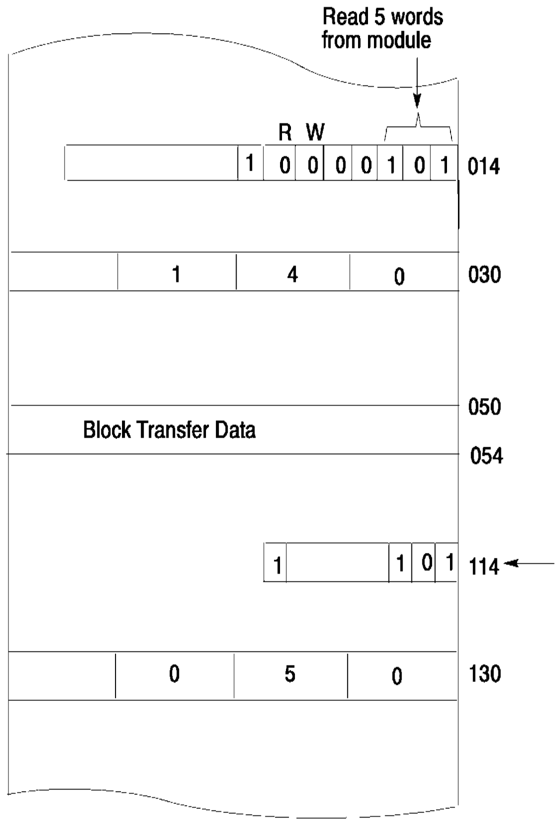
Setting The Number of Words To Transfer

Each block transfer module is set to a default value that specifies the maximum number of words to be transferred. The user can either select a different number of words to transfer or use the module's default value. Because each default value differs, consult the appropriate documentation for the module.

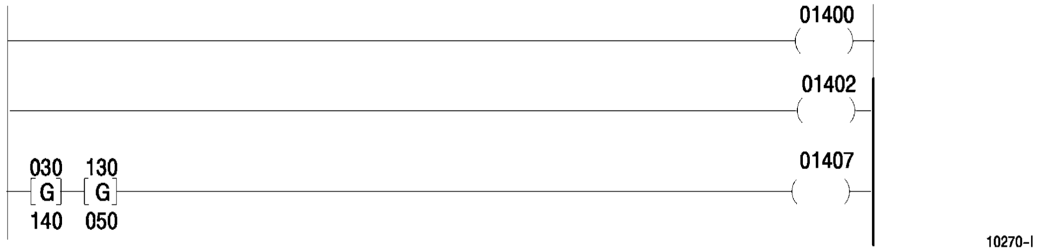
The number of words to transfer is stored in the upper or lower output image table byte that corresponds to the modules location (Figure 11.8). The user must set the appropriate bits on to specify a binary value that equals the number of words to transfer. These bits can be set on by programming unconditional Output Energize or Output Latch instructions (Figure 11.8).

Figure 11.8
Setting the Number of Words to Transfer

# WORDS TO TRANSFER	BIT PATTERN							
	7	6	5	4	3	2	1	0
Default			0	0	0	0	0	0
1			0	0	0	0	0	1
2			0	0	0	0	1	0
3			0	0	0	0	1	1
4			0	0	0	1	0	0
5			0	0	0	1	0	1
62			1	1	1	1	1	0
63			1	1	1	1	1	1
64			0	0	0	0	0	0



Bit 6 or 7 is set if a successful transfer occurred



When a successful transfer is complete, the output image table byte is mirrored in the corresponding input image table byte.

Buffering Data

Data that is read from the block transfer module and transferred to data table locations must be buffered to assure the data is valid. Data that is written to the module need not be buffered, since all block transfer modules buffer the data they receive.

Data read into the processor's memory is buffered in two stages. First, the data is examined as a group to ensure block transfer was successfully completed. Then, a diagnostic bit is examined as assure the bits in each word transferred are valid. If the data is valid, a Get/Put transfer is performed with this data. The diagnostic bit for each block transfer module differs. Refer to the respective documentation for the block transfer module to determine the correct diagnostic bit.

The example in Figure 11.9 shows the memory map and ladder diagram rungs for buffering 3 words of data that are read from the block transfer module. The data is read and buffered in the following sequence:

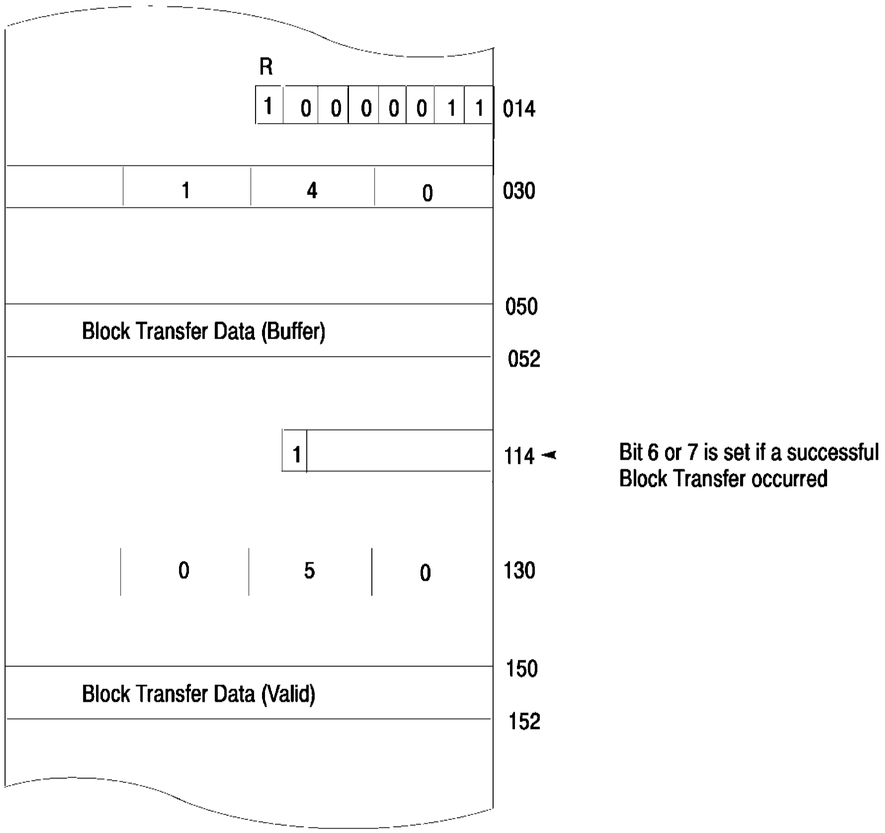
1. When rung 3 goes true, bit 01407 will be turned on and block transfer will be requested. This latches on storage bit 02000 in rung 4.

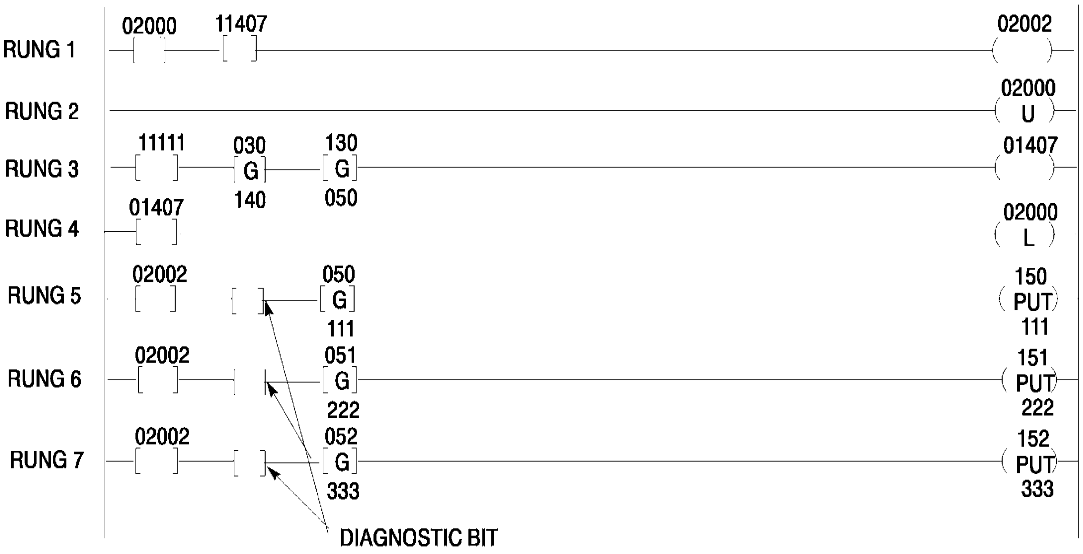
2. During the I/O scan, block transfer will be attempted, interrupting the output scan. If the module is ready, block transfer will be performed. Data from the module will be loaded into words 0508–0528. When the transfer is complete, the output image table byte associated with the module's location is duplicated in the corresponding input image table byte. This indicates block transfer was successfully performed (the number of words requested equals the number of words sent). The processor then continues with the I/O scan and program scan.

3. During the program scan, bit 02000 in rung 1 is still latched on. Bit 11407, the block transfer complete bit, is on indicating that a successful transfer has occurred. Because rung 1 is true, bit 2002 is turned on. In rung 2, bit 02000 is then unlatched.

4. In rung 5, a diagnostic bit is examined to assure the data read into memory during the last block transfer is valid. With bit 02002 still on and the diagnostic bit on, the data will be transferred from 0508 to 150. In rungs 5 and 6, the data in words 0518 and 0528 will be transferred to words 1518 and 1528 if the diagnostic bits are on. The data transferred to words 1508–1528 can then be used for other program logic.

Figure 11.9
Buffering Data

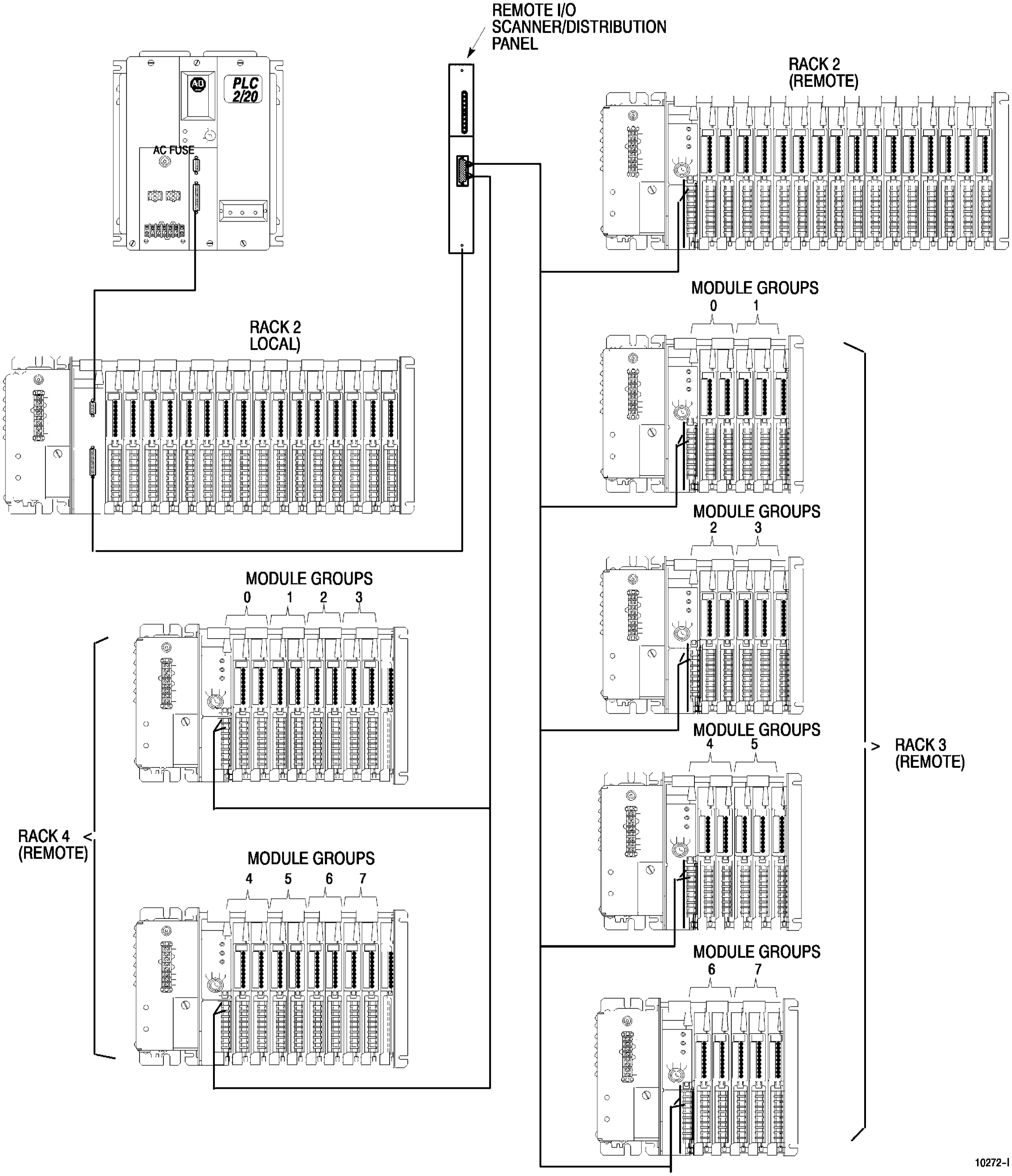

R
1 0 0 0 0 0 1 1
014
1 4 0
030
050
Block Transfer Data (Buffer)
052
1
114
Bit 6 or 7 is set if a successful Block Transfer occurred
0 5 0
130
150
Block Transfer Data (Valid)
152


RUNG 1
02000
11407
02002
RUNG 2
02000
U
RUNG 3
11111
030
G
140
130
G
050
01407
RUNG 4
01407
02000
L
RUNG 5
02002
050
G
111
150
PUT
111
RUNG 6
02002
051
G
222
151
PUT
222
RUNG 7
02002
052
G
333
152
PUT
333
DIAGNOSTIC BIT
1

Remote Fault Zone Programming

The remote fault zone programming technique is used to disable parts of or the entire user program when a fault occurs in a remote I/O rack. Remote I/O racks are controlled by the processor via the Remote I/O Scanner/Distribution Panel (cat. no. 1772–SD) or (cat. no. 1772–SD2) and can be located up to 10,00 feet from the S/D panel. Up to 2 local I/O racks can be used in a system with remote I/O racks (Figure 11.10). Refer to publications 1772–910 and 1772–929 for information on switch settings.

Figure 11.10
Remote I/O Configuration Example



Unlike local I/O racks, each remote I/O rack can have up to 128 I/O points using one of the following arrangements:

- One 128–I/O chassis
- Two 64–I/O chassis
- One 64–I/O chassis and two 32–I/O chassis
- Four 32–I/O chassis

The 1772–LP1 processor can control up to four I/O racks. The 1772–LP2 processor can control up to seven I/O racks. For information on wiring, switch settings and use of the remote I/O scanner/distribution panel, refer to the Remote I/O Scanner/Distribution Panel (cat. no. 1772–SD), publication 1772–910 or (cat. no. 1772–SD2), publication 1772–929.

Fault zones can be programmed around certain parts of the program or the entire program using fault status bits and MCR or ACL zones. The fault status bits used for remote fault zone programming are located in data table words 1258 and 1268 (Figure 11.11).

CAUTION: Input modules cannot be located in rack 2, module groups 5 and 6 if words 125 and 126 are used for fault status bits.

A group of four fault status bits corresponds to a single I/O rack (figure 11– 11). For example, bits 1225078–125048 correspond to rack 1; bits 125038–125008 correspond to rack 2. Although bits 126138–126108 are not used as fault status bits, they cannot be used for storage.

Each fault status bit within a group of four corresponds to two consecutive module groups or 32 I/O points (Figure 11.11). When a fault occurs in a remote rack, one or more of the four status bits are set on depending on the configuration of the I/O rack.

In Figure 11.10, rack 1 is a local 128–I/O rack. Rack 2 consists of a 128–I/O chassis. Rack 3 consists of four 32 I/O chassis and rack 4 consists of two 64–I/O chassis.

If a fault occurs in rack 2, bits 125038–125008 will be set on. If a fault occurs in the first I/O chassis of rack 3, bit 12517 will be set on. Similarly, if a fault occurs in the first I/O chassis of rack 4, bits 125138 and 125128 will be set on.

By selecting either dependent or independent fault zone programming, the user can disable certain parts of the program or the entire program when a fault occurs in a remote I/O rack. Alternate parts of the program can also be enabled when a fault occurs.

Figure 11.11
Fault Status Bits

I/O Rack	Module Groups	Fault Status BIt
1	0-1	12507
	2-3	12506
	4-5	12505
	6-7	12504
2	0-1	12503
	2-3	12502
	4-5	12501
	6-7	12500
3	0-1	12517
	2-3	12516
	4-5	12515
	6-7	12514
4	0-1	12513
	2-3	12512
	4-5	12511
	6-7	12510
5	0-1	12607
	2-3	12606
	4-5	12605
	6-7	12604
6	0-1	12603
	2-3	12602
	4-5	12601
	6-7	12600
7	0-1	12617
	2-3	12616
	4-5	12615
	6-7	12614

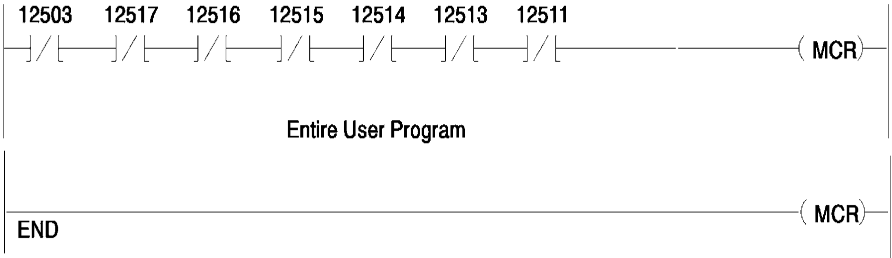
Dependent Programming

Dependent fault zone programming is used to disable the entire program when a fault occurs in one remote I/O chassis. The entire user program is zoned off using an MCR or ZCL zone (Figure 11.12). The appropriate fault status bits for the remote I/O chassis are programmed as Examine Off conditions for the zone. When a fault occurs in a remote I/O chassis, the corresponding fault status bit is set on, causing the MCR or ZCL zone to go false. All outputs will then be controlled by the MCR or ZCL zone, including outputs of local I/O racks.

In addition to programming a dependent fault zone, the user must ensure that the fault control switch on the S/D or SD2 panel is set off for dependent mode. Refer to publication 1772–910 or 1772–929, respectively, for the switch locations and settings of the S/D panel.

NOTE: If a fault occurs in a local rack, all racks will behave according to their last state switch whether dependent or independent mode has been selected.

Figure 11.12
Dependent Fault Zone Programming



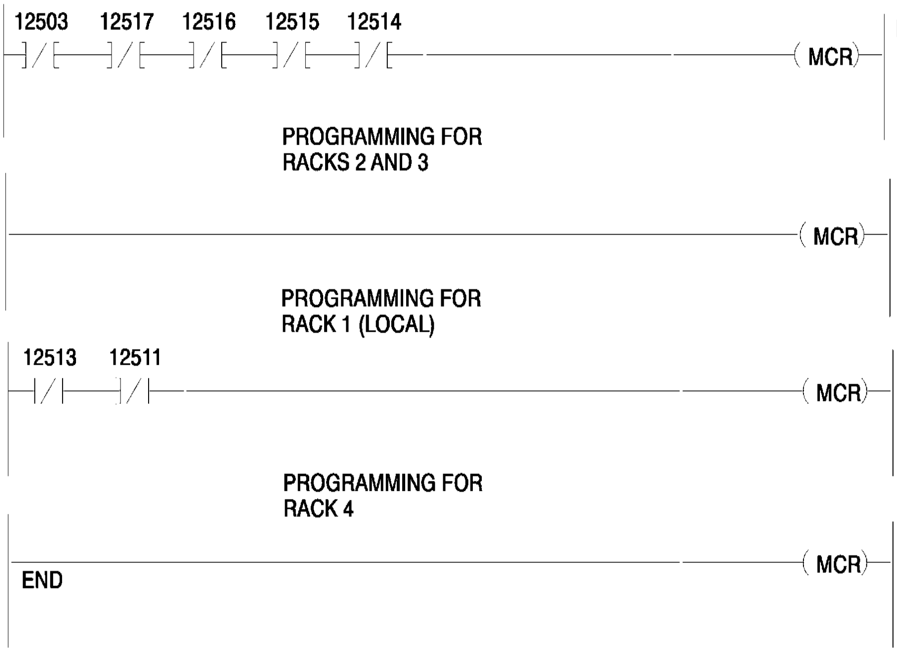
Dependent programming to I/O configuration in Figure 11.10.

When a fault status bit is set ON, the MCR r ZCL zone is FALSE and all outputs are controlled by the zone.

Independent Programming

Independent fault zone programming is used to zone off independent sections of user program. The programming for each I/O chassis can be contained in separate MCR or ZCL zones or more than one I/O chassis can be contained in a single zone (Figure 11.13). The user may also wish to enable alternate parts of a program when a fault occurs in a remote I/O chassis (Figure 11.14).

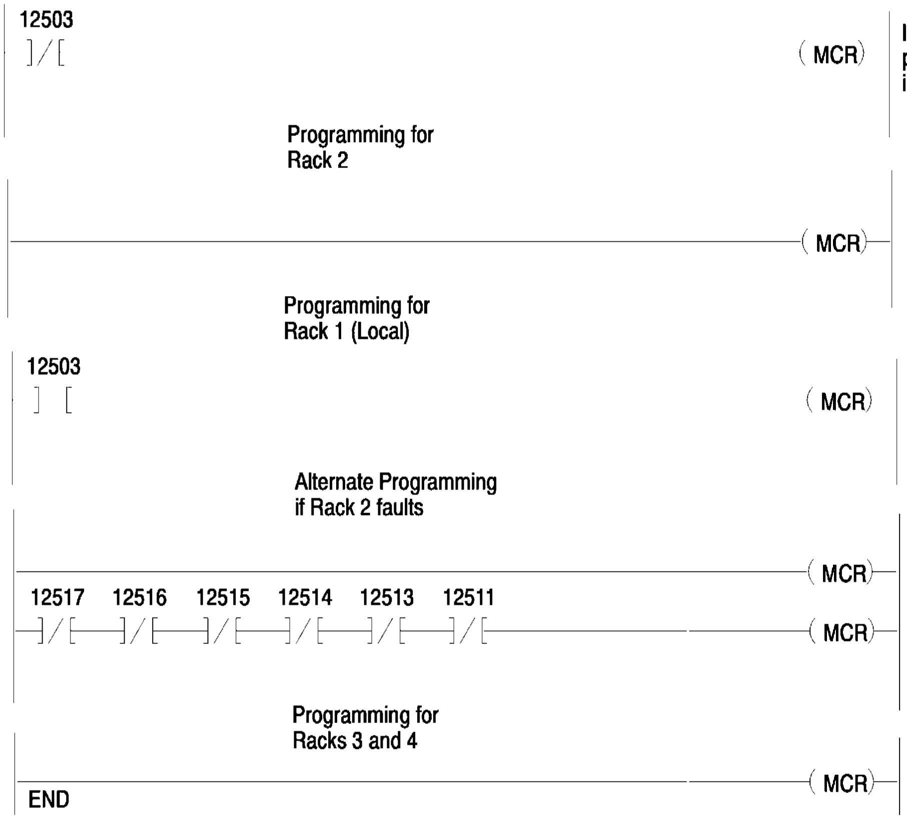
Figure 11.13
Independent Fault Zone Programming



Independent programming for I/O configuration in Figure 11.10.

When a fault status bit is set ON, the MCR r ZCL zone is FALSE and controls all outputs in the zone.

Figure 11.14
Independent Fault Zone Programming



Independent
programming for I/O configuration
in Figure 11.10.

When a fault status bit is set ON,
the MCR r ZCL zone is FALSE
and controls all outputs in the
zone. The Alternate program
is enabled when fault status
bit 12503 is set ON.

Independent fault zones are programmed using the appropriate fault status bits as Examine Off conditions for the MCR or ZCL zones (Figure 11.13 and Figure 11.14). When a fault occurs in a remote I/O chassis, the corresponding fault status bit is set on. The MCR or ZCL zone conditioned by that fault status bit will go false, enabling the zone. All outputs within the zone will be controlled by the zone.

In addition to programming independent fault zones, the user must ensure that the fault control switch on the SD or SD2 panel is set on for independent mode. Refer to publication 1772–910 or 1772–929, respectively, for the switch locations and settings of the SD panel.

NOTE: If a fault occurs in a local rack, all racks will behave according to their last state switch whether dependent or independent mode has been selected.

One-Shot

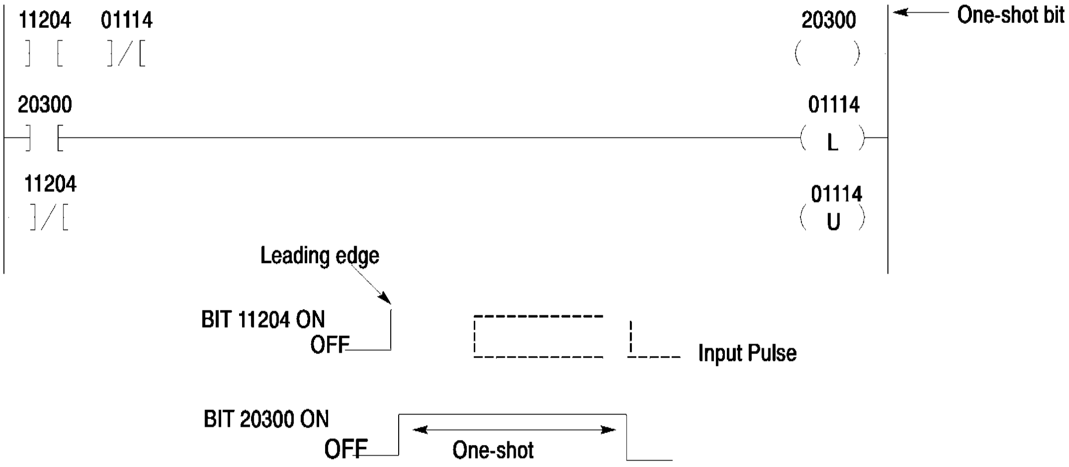
The one–shot programming technique is used for certain applications to set a bit on for one scan only. There are two types of one–shots that can be programmed:

- Leading Edge
- Trailing Edge

Leading Edge One-Shot

A leading edge one–shot is used to set a bit on for one scan when it input condition has made a false–to–true transition. The false–to–true transition represents the leading edge of the input pulse. The programming for a leading edge one–shot is shown in Figure 11.15.

Figure 11.15
Leading Edge One Shot



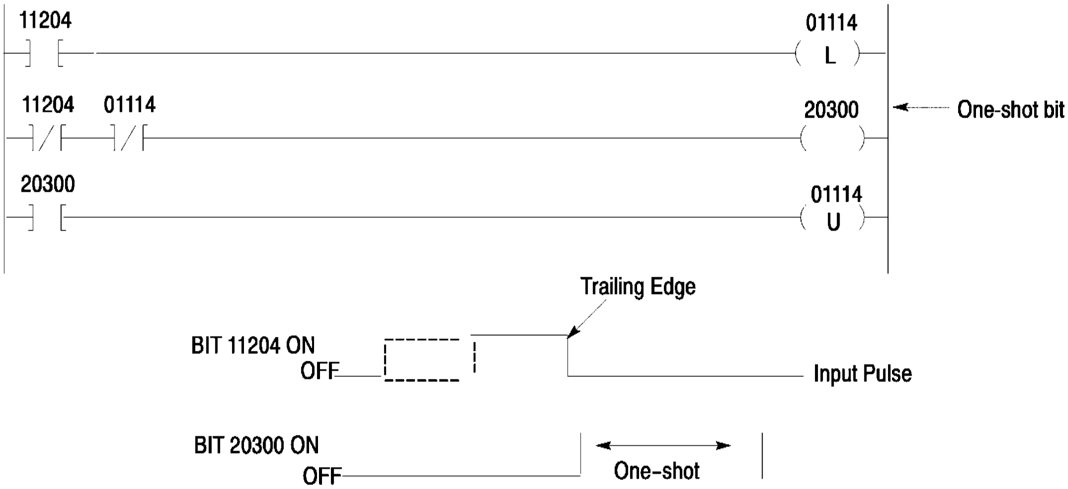
When bit 11204 makes a false–to–true transition, the one–shot bit (bit 20300) is set on for one scan. The length of time bit 11204 remains on does not affect the one–shot bit due to the next two rungs. Bit 01114 will be latched on when bit 11204 is on or bit 01114 will be unlatched when bit 11204 is off. During the next scan, either set of conditions will prevent bit 20300 from being set on. The one–shot bit is set on for one scan only when bit 11204 makes another false–to–true transition.

Trailing Edge One-Shot

A trailing edge one–shot is used to set a bit on for one scan when its input condition has made a true–to–false transition. The true–to–false transition represents the trailing edge of the input pulse. Programming for a trailing edge one–shot is shown in Figure 11.16.

When bit 11204 goes true, bit 01114 is latched on. As soon as bit 11204 makes a true–to–false transition, the one–shot bit (bit 20300) is set on and bit 01114 is unlatched. Bit 20300 will remain on for only one scan. The input bit 11204 must go true and false to set the one–shot bit on for another scan.

Figure 11.16
Trailing Edge One-Shot



Chapter 12

Scan Time and Execution Time

General

Scan time and execution times may be important considerations for developing certain application programs. Scan time depends on many factors and will vary for each program. The following sections will examine some of these factors.

San Time

Scan time is the total time it takes to perform the program scan and the I/O scan. The PLC-2/20 processor performs both scan serially. First it updates the I/O image tables; then it executes the user program.

Program Scan Time

Program scan time is the time needed to execute the instructions in the user program. This time depends on the type of instructions, the data operated on and whether the instructions are true or false. Nominal scan time for the PLC-2/20 processor is 5 ms per 1K of memory. This time will increase by approximately 4% or 1 ms, whichever is greater, when the industrial terminal is connected to the processor. If a data highway interface module is connected to the processor, the program scan time will increase by approximately 8% or 1 ms, whichever is greater.

I/O Scan Time

The time needed to monitor and update the I/O racks depends on the number of I/O racks, whether they are local or remote, and whether block transfer occurs.

For local I/O racks, nominal I/O scan time is 0.5 ms per 128 I/O. If block transfer occurs during the I/O scan, this time will increase for each block transfer by approximately 100 ~s + (80 ~s x the number of words transferred).

Remote Systems

In remote systems, the Remote I/O Scanner Distribution Panel (cat. no. 1772-SD, -SD2) performs the remote I/O scan asynchronously to the processor scan. The 1772-SD panel gathers I/O data from the Remote I/O Adapter Module (cat. no. 1771-AS) and stores the data in its buffer. The processor, during its I/O scan, updates local I/O racks first and then gets the remote I/O data from the 1772- SD panel buffer. The processor takes approximately 0.5 ms per remote I/O rack to get the information from the buffer. (Remember that a remote I/O rack can be 128 I/O in up to four I/O chassis.) Thus, scan time in a remote system is the sum of the program scan time, the local I/O scan time and the time the processor takes to access the 1772-SD panel buffer.

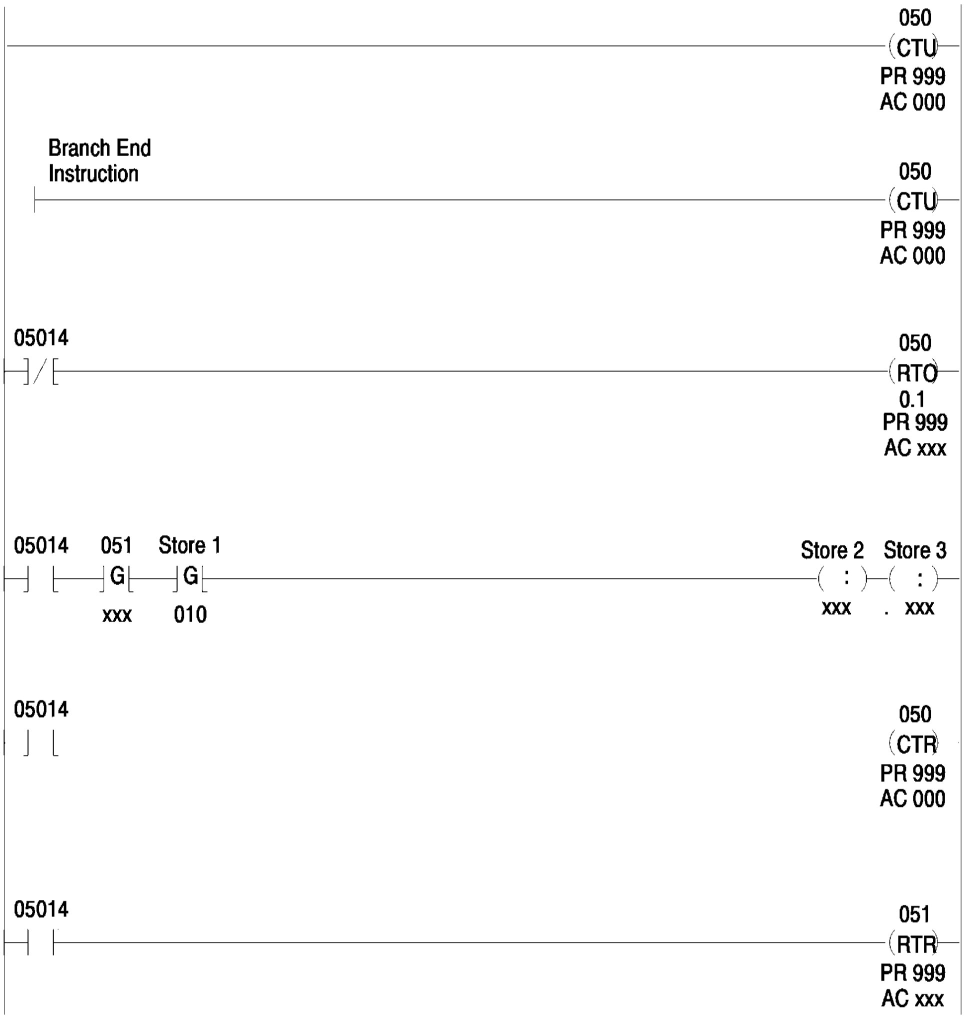
The time it takes the 1772-SD panel to scan the adapter modules does not affect the total scan time. It simply affects how current the data is in the buffer. The remote I/O scan time depends partly on the baud rate of the 1772-SD panel, which is set by it on-board dip switches. Assuming there are not block transfer modules in remote I/O racks, the nominal remote I/O scan time per adapter module is 8.5 ms for 57.6K baud and 7 ms for 115.2K baud.

Calculating Total Scan Time

Total scan time of an existing program can be determined by using the six-rung program in Figure 12.1.

The first two rungs are used to count 1000 scans using an Up-Counter instruction. Rung 3 is used to time the first 1000 scans. When the counter overflows to 1000, bit 05014 comes on and the timer stops. Rung 4 gets the value of the timer after 1000 scans and divides the value by 010 to display the result in milliseconds. Rungs 5 and 6 reset the counter and timer.

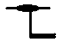
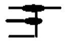
Figure 12.1
Program for Determining Average Scan Time



Instruction Execution Time

The program scan time for programs that are not yet entered into memory can be estimated by calculating the execution time for all the instructions. The sum of all these execution times would be equal to an average program scan time. Average execution times for PLC-2/20 instructions are listed in Table 12.A.

Table 12.A
Average Execution Times for PLC-2/20 Instructions [1] [2]

Instruction Name	Symbol	Average Execution TIme in Microseconds
Examine On	-[ ]-	3.0
Examine Off	-[/]-	3.0
Output Energize	-( )-	4.0
Output Latch	-(L)-	4.0
Output Unlatch	-(U)-	4.0
Timer On-Delay	-(TON)-	15.0
Timer Off-Delay	-(TOF)-	15.0
Retentive Timer On-Delay	-(RTO)-	15.0
Retentive Timer Reset	-(RTR)-	4.8
Up Counter	-(CTU)-	15.0
Down Counter	-(CTD)-	15.0
Counter Reset	-(CTR)-	4.8
Get	-[G]-	4.4
Put	-(PUT)-	4.8
Less Than	-[<]-	5.6
Equal to	-[=]-	5.6
Get Byte	-[B]-	3.0
Limit Test	-[L]-	4.8
Add	-(+)-	5.6
Subtract	-(-)-	5.6
Multiply	-(x)-	40
Divide	-(:)-(:)-	90
Branch Start		3.3
Branch End		2.2
Zone Control Last State **[3]**	-(ZCL)-	6.4
Master Control Reset	-(MCR)-	2.2
Immediate Input	-(I)-	12
Immediate Output	-(IOT)-	12
End	[END]	1
TEMPORARY END	[T.END]	1

[1] Times given are for instructions which are true. When instructions are false, execution time is <1 microsecond.
[2] When instructions addresses are 400_8 or greater, execution time increase by 3.4 microseconds for the instruction.
[3] When the ZCL instruction is false (i.e., outputs within the zone are held in their last state) execution time increases by 1. 2 microseconds for each word within the zone.

Watchdog Timer

The processor has an internal "watchdog" timer to monitor the logic circuits. The watchdog timer is set to a certain time depending on the processor interface module and is reset each I/O scan. If the total scan time exceeds the watchdog timer setting, the watchdog timer will time out and cause a processor fault. The 1772-LF processor interface module has the watchdog timer setting of 57 ms. The 1772-LH processor interface module has the watchdog timer setting of 115 ms. Thus, if the scan time exceeds either 57 ms using the 1772-LF module or 115 ms using the 1772-LH module, the watchdog timer will time out and cause a processor fault. The watchdog timer will be reset when the processor fault is corrected.

WARNING: The lower limit of the input device cycle time should not be less than the scan time of the processor. If so, incorrect data could be used during program execution. Critical inputs can be monitored and critical output can be controlled in an accelerated manner using the Immediate I/O instructions described in section title I/O Updates in chapter 7.

Chapter 13

Numbering System

General

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

- Decimal
- Octal
- Binary
- Hexadecimal

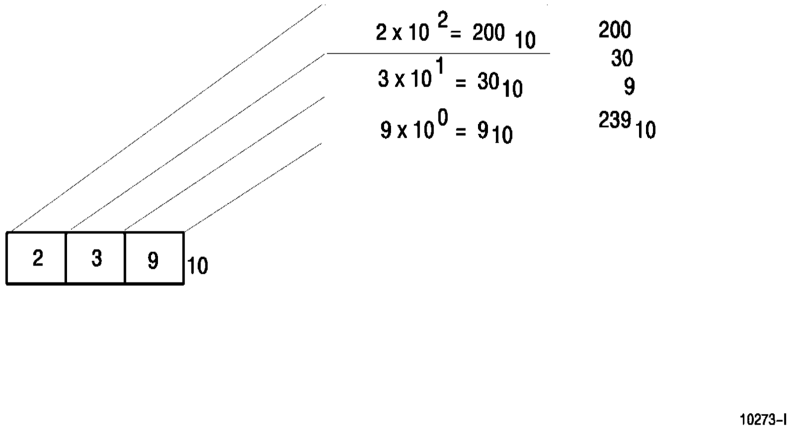
These numbering systems differ by their number sets and place values.

Decimal Numbering System

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

Each place value in a decimal number represents a power of ten (Figure 13.1), starting with 100. The value of a decimal number is determined by multiplying each digit by its corresponding place value and adding these numbers together.

Figure 13.1
Decimal Numbering System



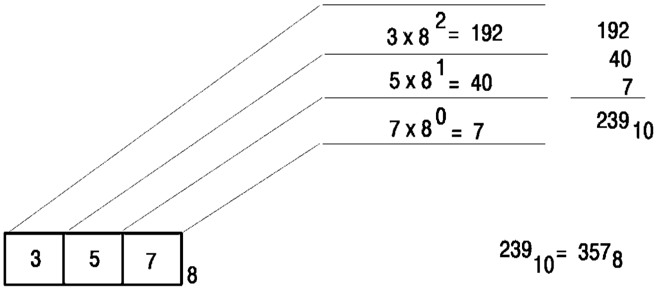
Octal Numbering System

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.

Just like all numbering systems, each digit in an octal number has a certain place value, represented by a power of eight (Figure 13.2).

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

Figure 13.2
Octal Numbering System



10274-I

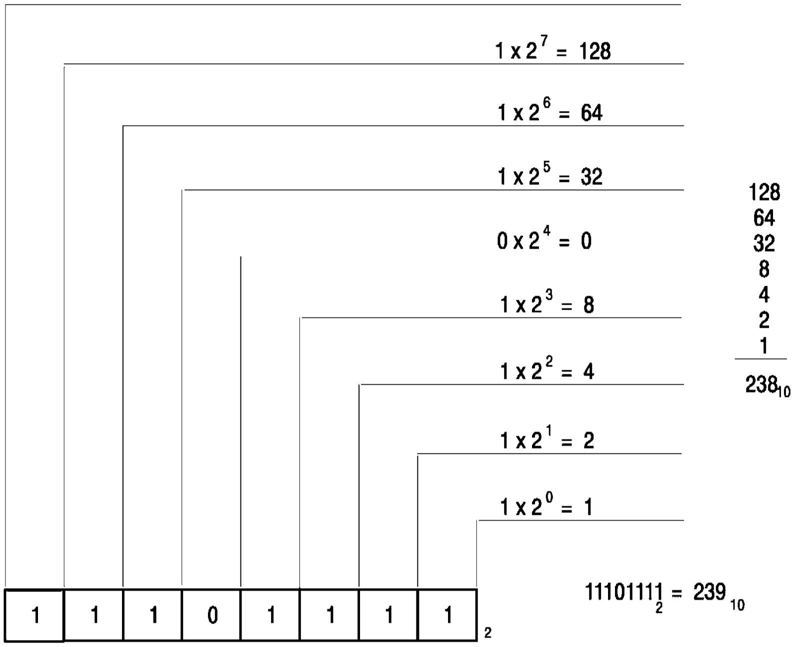
Binary Numbering System

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

Each digit in a binary number has a certain place value expressed as a power of two (Figure 13.3). The decimal equivalent of a binary number is computed by multiplying each binary digit by its corresponding place value and adding these numbers together.

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

Figure 13.3
Binary Numbering System

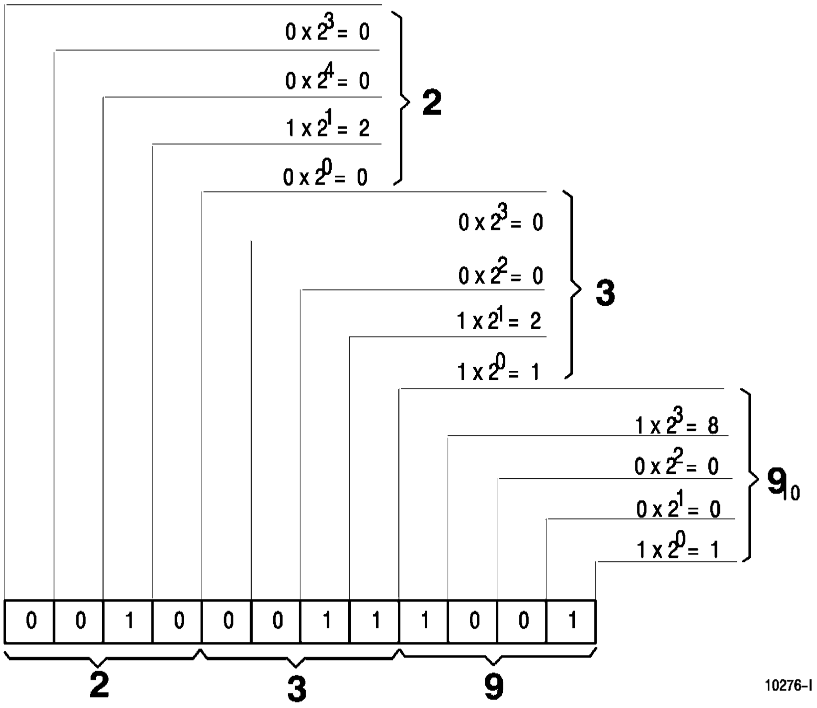


10275-I

Binary Coded Decimal

Binary coded decimal (BCD) uses an arrangement of 12 binary digits to represent a 3-digit decimal number from 000 to 999 (Figure 13.4). Each group of 4 binary digits is used to represent a decimal number from 0 to 9.The place values for each group of 4 digits are 2^0, 2^1, 2^2 and 2^3 (Table 13.A).

Figure 13.4
Binary Coded Decimal



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.

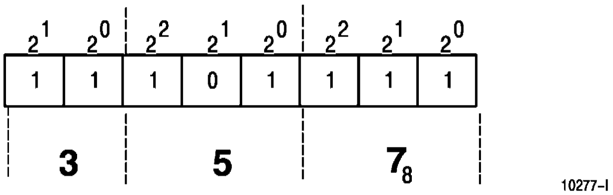
Table 13.A
BCD Representation

Binary Position				Decimal
2^3 (8)	2^2 (4)	2^1 (2)	2^0 (1)	Equivalent
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

Binary Coded Octal

Binary coded octal (BCO) uses an arrangement of a 8 bits (one byte) to represent a 3-digit octal number from 008 to 3778 (Figure 13.5). The 8 bits are broken down into three groups: 2 bits, 3 bits and 3 bits.

Figure 13.5
Binary Coded Octal



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

Table 13.B
Binary Coded Octal Representation

Binary Position			Octal
2^2 (4)	2^1 (2)	2^0 (1)	Equivalent
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

Hexadecimal Numbering System

The hexadecimal numbering system has a number set of 16 digits: the numbers 0-9 and the letter A-F (Figure 13.6). The letters A-F represent the decimal numbers 10-15 respectively.

Figure 13.6
Hexadecimal Numbering System

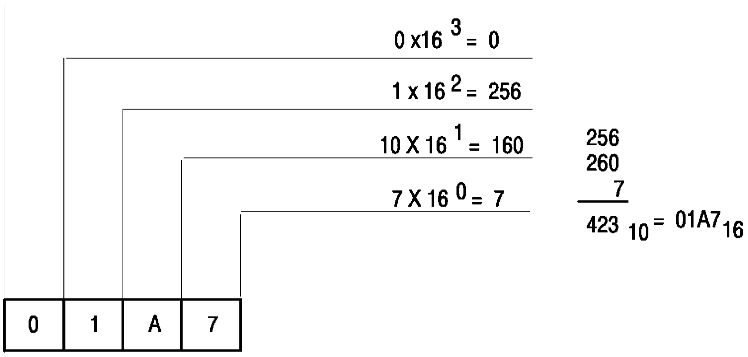
Hexadecomal Digits	Binary Equivalent	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

A hexadecimal number can be converted to a decimal number by multiplying the hexadecimal digit by its corresponding place value (Figure 13.7).

Because each hexadecimal digit represents 4 binary digits, it is easy to convert a hexadecimal number to a binary number. This is done by writing out the 4-bit binary pattern for each hexadecimal digit (Figure 13.8).

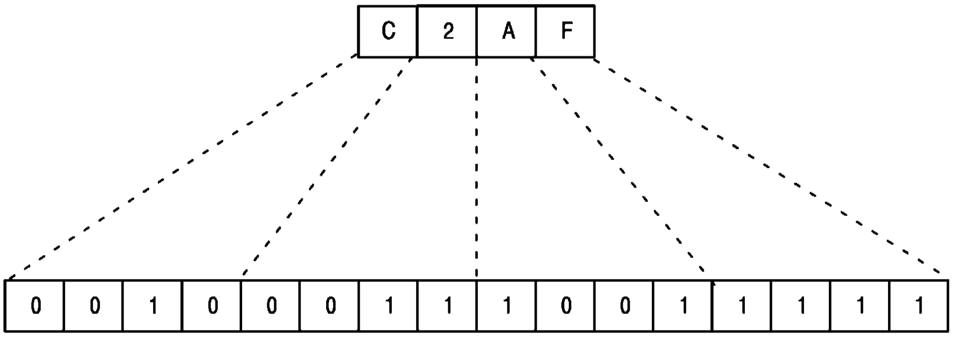
NOTE: Any number to the zero power is always equal to one.

Figure 13.7
Hexadecimal to Decimal Conversion



10278-I

Figure 13.8
Hexadecimal to Binary Conversion



10279-I

Index

A-B QUALITY ALLEN-BRADLEY

Publication 1772-6.8.1 — June, 1984
Supersedes Publication 1772-802 — June, 1984

PLC is registered trademark of Allen-Bradley Company

Systems Division — PC Business
747 Alpha Drive, Highland Heights, Ohio 44143

P/N 955094-66 PRINTED IN U.S.A.