SECTION 4. PROGRAMMING APPROACH

4-1. GENERAL DESCRIPTION

Programming the PC-1100/1200 programmable controller system is simply a way to specify a sequence of operations that the application is to perform. The PC-1100/1200 controller, through programming, digitally duplicates the operation of a relay control system. The PC-1100/1200 programs consist of a set of programmable functions which specify the relay circuits and devices to be duplicated and the operational sequences to be performed.

In an application, these programmed functions together produce control programs, which are called reference ladder diagrams. These reference ladder diagrams are similar to the ladder diagrams used in relay control systems. A description of each programmable function is given in Section 5.

The reference ladder diagram clearly shows the types of contacts used in each of the program's control circuits and how these user-selected contacts are configured to cause an assigned device to operate in the desired manner. A properly constructed reference diagram simplifies the entry of circuits into the controller and makes the function of these circuits easier to understand.

4-2. REFERENCE LADDER DIAGRAMS

The Westinghouse programmable controllers use reference ladder diagrams to document an installed control program. The diagrams differ from conventional relay control system ladder diagrams in four ways:

1. Circuits are redrawn, when necessary, to ensure compatibility with the scanning process used in the programmable controllers.

2. Circuits are redrawn, when necessary, to make contact connections easier to recognize.

3. Circuits are redrawn, when necessary, to fit within the allowed contact area.

4. Contacts, coils and special functions are assigned appropriate labels.

In PC-1100/1200 reference ladder diagrams, input devices (e.g., pushbuttons, limit switches, pressure switches, etc.) are shown as relay contacts. Figure 4-1 illustrates this basic difference from conventional relay circuits. As shown, PB6 is replaced by IN0012 (input No. 12) and LS14 is replaced by IN0025 (input No. 25).

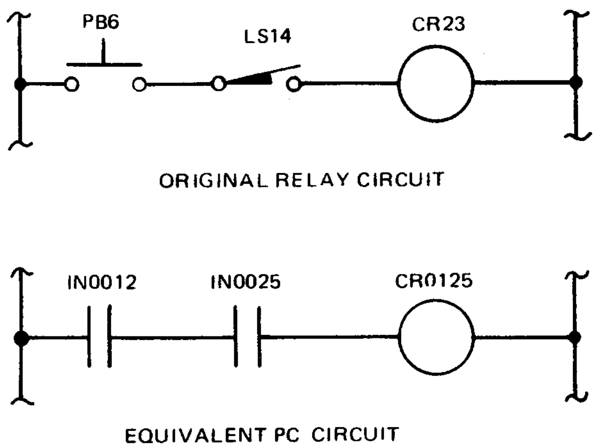


Figure 4-1. Input Devices Shown as Contacts

There is another major difference between the two types of diagrams. Each element in the reference ladder diagram (contact, coil, special function) is shown with a label (e.g., IN0025 or CR0125) which identifies the element in the controller. A label consists of a type designation and a reference number. (Labels are discussed in detail in Paragraph 4-6.)

As shown in Figure 4-1, the IN part of the label identifies the type of element. In this example, the In contacts are controlled by input circuits. As previously described, real-world input devices (e.g., pushbuttons and switches) are connected to converter circuits on input modules. The input circuits convert real-world signal levels to controller signal levels. Also illustrated in Figure 4-1 are reference numbers such as 0012. These indicate which input circuit address controls each of the input contacts.

In the reference diagram, output devices (e.g., motor starters, solenoids, lights, etc.) are shown as coils. (See Figure 4-2.) Conventional control relay coils are also shown as coils. Therefore, there are really two different kinds of coils shown in the reference diagram. One kind of coil, called an output coil, controls internal contacts and is also associated with an output converter circuit that controls a real-world output device. The second kind of coil, called a logic coil, only controls internal contacts. It cannot directly control a real-world output device.

Another difference between the reference diagram and the relay ladder diagram is that many special functions (e.g., timers, counters, comparisons, addition, etc.) are shown as simple blocks. Each special function has at least one (and up to four) contact circuit which controls its operation. For example, Figure 4-3 shows a timer. The timer accumulates when both the timing circuit and the enable circuit are conducting.

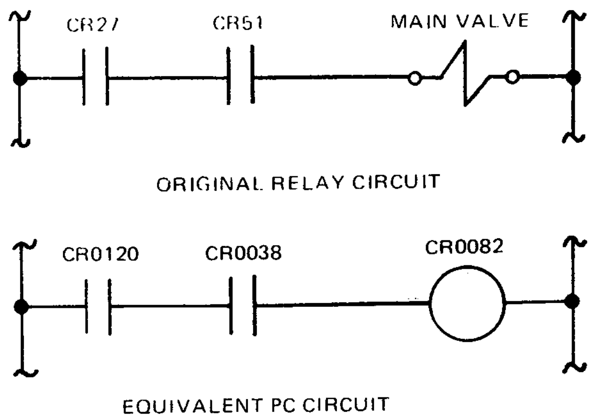


Figure 4-2. Outputs Shown as Coils

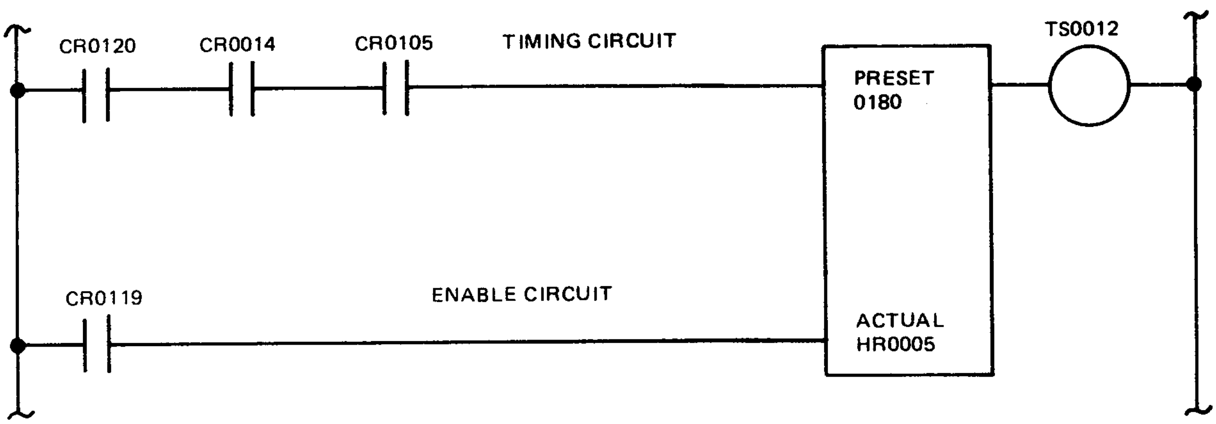


Figure 4-3. Special Function Block with Multiple Contact Circuits

In general, special function labels are similar to the labels used with coils. The first two characters form the type designation, which indicates the type of special function. For example, TS (Timer Seconds) indicates a timer with a resolution of one second (00000 through 65535). The reference number indicates which contacts are controlled by the special function coil. As with CR coils, both output coils and logic coils may be defined.

Special functions may respond to additional information supplied during the programming process. For example, Figure 4-3 shows a timer with a preset value of 180 seconds. This timer runs up to 180 seconds; it then energizes the coil. This operating data need not be a constant, but can be changed. It may be periodically stored in some memory location by another function or by some real-world input. For example, a thumbwheel could be used to set the preset value for a timer.

The example also illustrates that special functions can use numerical data, in addition to having discrete inputs and outputs (contact circuits and coils). They accept the numerical data from registers and supply similar numerical data to registers. When numerical data is required by a special function, the storage location of that data (input register, output register, or holding register) must be specified. For example, in Figure 4-3, the actual value of the timer is stored in Holding Register HR0005.

A register may be associated with more than one special function. For example, a given function may store numerical data in a register which is also used to supply numerical data to another special function.

Registers can perform a variety of functions, depending upon the application. For example, the value in an output register could operate a BCD display, in addition to providing numerical input to a special function.

In addition to inputs and outputs, the holding register is used as an internal storage location for 16 bits of binary or BCD information. The holding register must be considered in the scheme used for designating the elements of the reference ladder diagram.

4-3. PROGRAMMING CONSIDERATIONS

To determine which outputs must be activated, the controller repeatedly scans the circuits programmed into it.

First, the controller examines all inputs and stores their states in a portion of its memory. Then, the controller scans the programmed circuits, starting with the first circuit programmed at the top of the reference diagram. It does this to determine which circuits are conducting. The condition of each circuit (conducting or non-conducting) depends upon the states of any associated inputs and upon the states of the contacts that are controlled by other programmed coils. As the controller sequentially scans the programmed circuits, the coil states are updated one by one.

When a coil is updated during a scan, subsequent references to the updated coil's contacts reflect the updated status. In Figure 4-4, for example, coil CR0062 is controlled by the status established for IN0018 and CR0053 during the current scan, and by the status established for CR0101 during the previous scan (since CR0101 has not yet been scanned when CR0062 is scanned.)

As the circuits are scanned, the controller stores the newly determined coil states in a portion of memory. These memory states are used to update the states of any associated real-world outputs at the end of the scan.

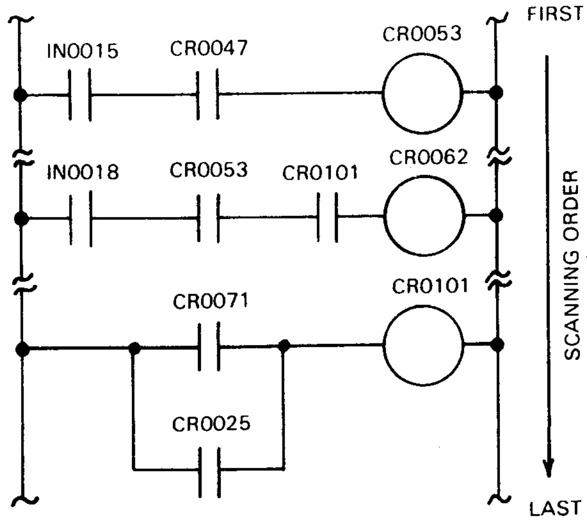


Figure 4-4. Controller Scanning

To produce the desired system operation, circuits should be drawn and programmed in the order in which they should conduct. This order should be evaluated in light of the following discussion concerning conduction within ladder circuits.

4-4. Conduction Within Ladder Circuits

When the controller is scanning the programmed circuits, a conduction path through contacts is considered to occur only from left to right. A conduction path for branches may also occur vertically (either up or down), as well as from left to right.

For example, consider the circuit shown in Figure 4-5. There are only three possible paths to energize coil G: A to B to C, A to D to E, and F to E. The path F to D to B to C does not energize coil G because it would require conduction from right to left through contact D.

4-5. Construction Guidelines

There are five rules which should be observed when drawing the circuits in the reference diagram:

Rule 1: Always draw contacts as part of horizontal branches, not as vertical branches.

Rule 2: Arrange those branches that do not contain contacts so that they run vertically not horizontally.

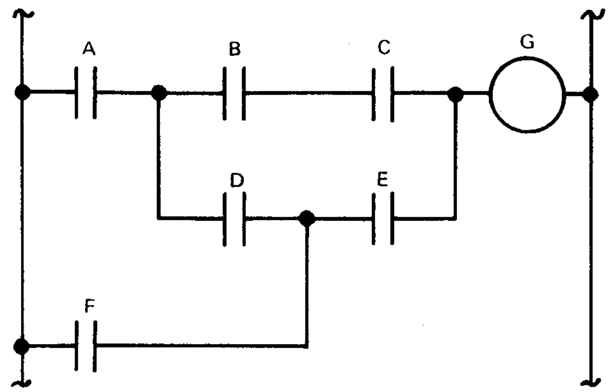


Figure 4-5. Circuit Conduction

Rule 3: Contacts connected to the right of a vertical branch should be connected to the topmost available junctions in the branch.

Rule 4: For coils that do not have attached special function blocks (e.g., CR, BS, BC, etc.), the contact area can accommodate a maximum of 10 series contacts in any path among a maximum of 7 parallel rows.

Rule 5: For coils that have attached special function blocks (e.g., TT, TS, UC, etc.), the contact area can accommodate a maximum of 8 series contacts in any path among a maximum of 7 parallel rows.

RULE 1

To illustrate Rule 1, consider Figure 4-6. In the top circuit, contact C is drawn vertically, making it difficult to correctly identify the attributes associated with each contact. In addition, it is difficult to determine the direction of conduction through contact C.

In the bottom circuit, the attributes associated with each contact are easier to identify. The direction of conduction through contact C is from left to right, as discussed previously. Note that in this circuit, path D to C to B does not energize coil F. Figure 4-7 shows the same circuit with path D to C to B included to energize coil F.

RULE 2

To illustrate Rule 2, consider Figure 4-8. In the circuit at the top of the figure, the branch connection from contacts B and C to contacts H and I is drawn horizontally, making it difficult to identify the attributes associated with contacts B, D, and H. It also makes it difficult to identify path E to F to G to H to C as a valid path to energize coil J. In the circuit at the bottom of the Figure, the attributes are easier to identify. It is also easier to identify all valid parallel paths.

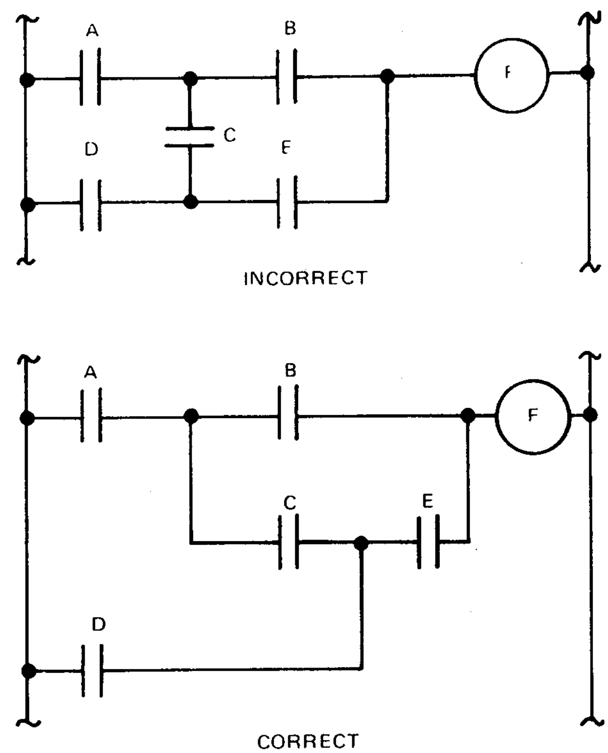


Figure 4-6. Rule 1: Contacts Must Occur in Horizontal Branches

RULE 3

To illustrate Rule 3, consider Figure 4-9. In the circuit on the top, contacts D and E are drawn at the bottom of the branch, making it difficult to correctly identify the attributes associated with contacts B, C and D.

In the bottom circuit contacts D and E are drawn at the top of the branch, thus making it easier to identify the UP attributes associated with contacts B and C. Contact D has no UP attribute, as the correctly drawn circuit shows. Note that this rule also implies that coils should appear in the topmost row of any circuit.

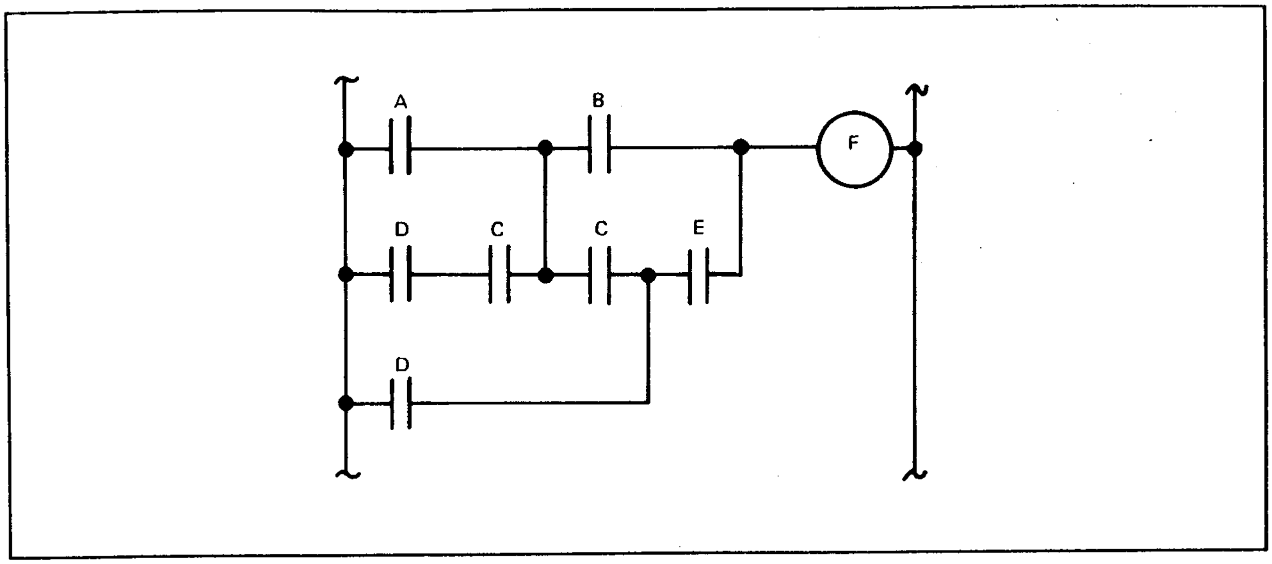


Figure 4-7. The Addition of Two Contacts Adds Path D to C to B

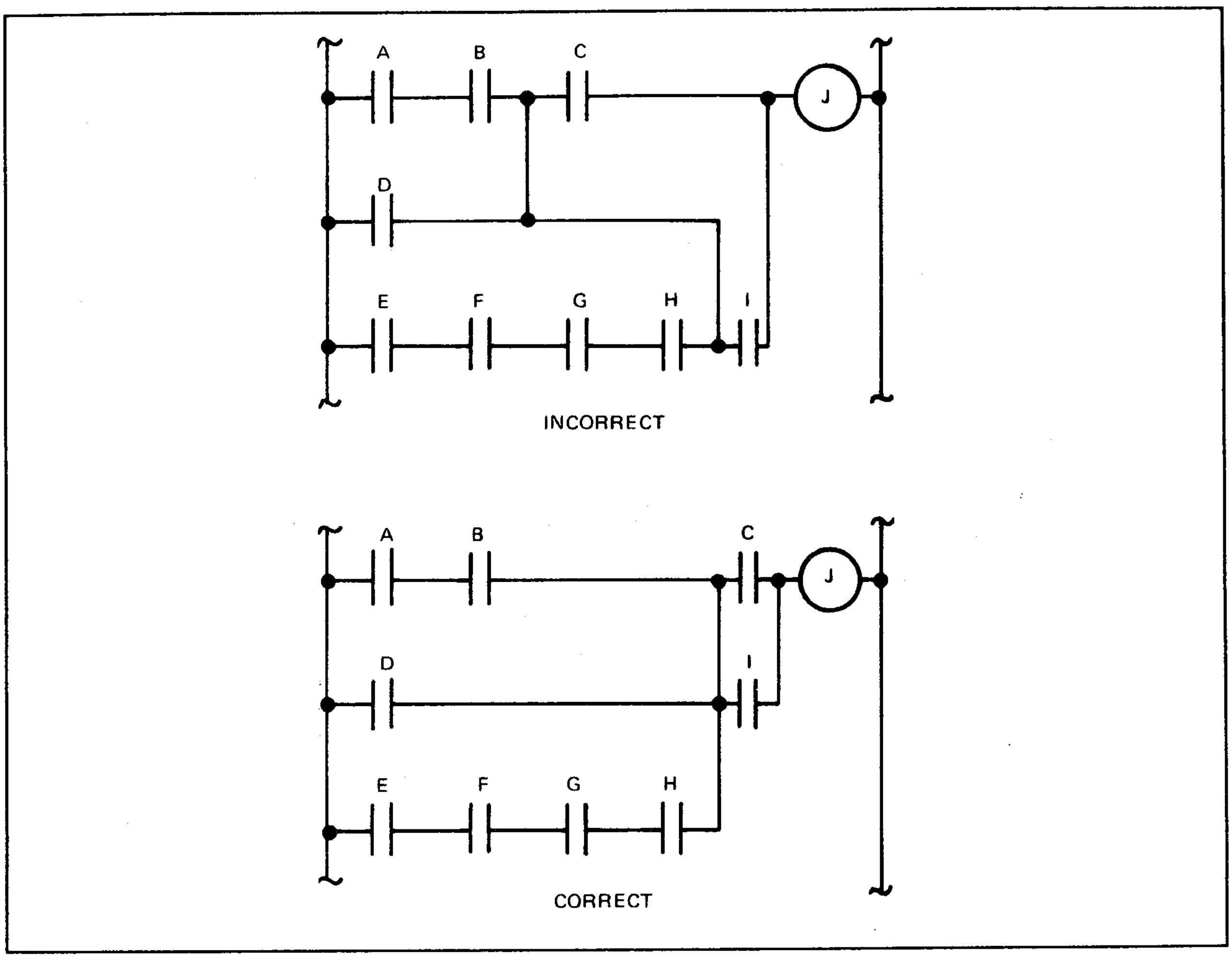


Figure 4-8. Rule 2: Branches Should Run Vertically

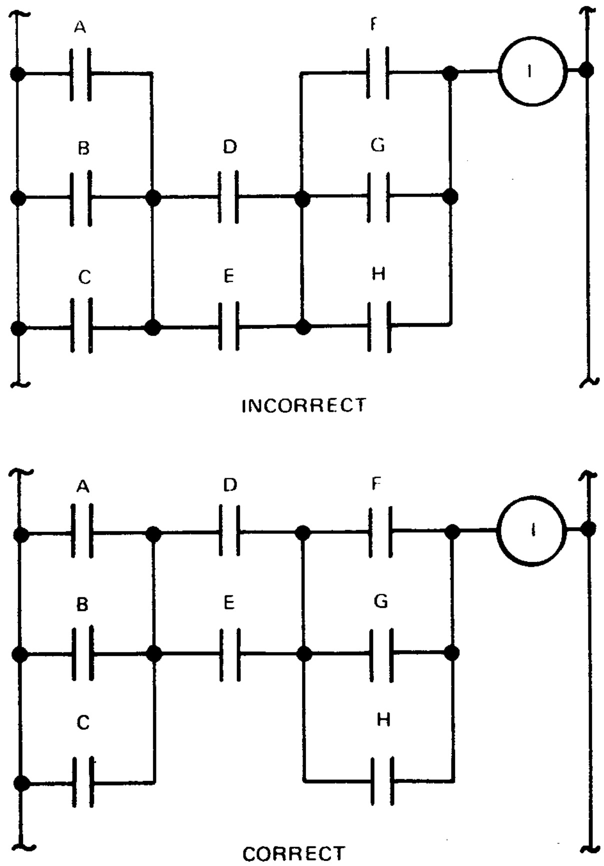


Figure 4-9. Rule 3: Contacts Should Be Connected to the Topmost Available Junctions

RULES 4 AND 5

Rules 4 and 5 simply state the limits on the size of the contact area available for each function. Figures 4-10 and 4-11 illustrate this concept. It is important to note that the limit on the number of series contacts is for any path. Series contacts need not appear in the same row. (See Figure 4-12.)

The purpose of Rules 1, 2, and 3 is to make programming easier. If violated, they will not inhibit the proper operation of the controller. Rules 4 and 5, however, are necessary for proper operation of the controller. Any lines that violate Rules 4 and 5 are illegal and cannot be entered. For example, Figure 4-13 illustrates an otherwise legal path which is not acceptable because of the number of contacts.

Figure 4-10. Rule 4: 10 x 7 Contact Area for Coils

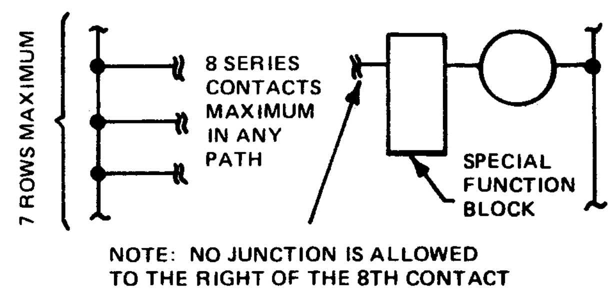


Figure 4-11. Rule 5: 8 x 7 Contact Area for Special Functions

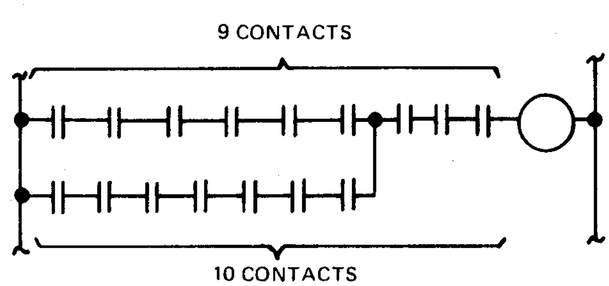


Figure 4-12. Example of Two Legal Paths

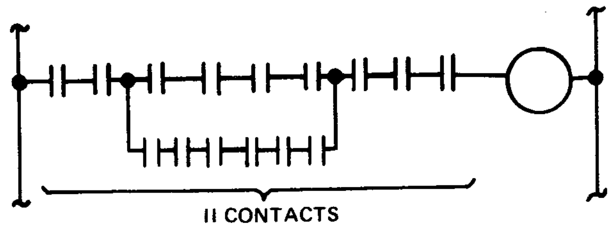


Figure 4-13. Example of an Illegal Path

4-6. Labeling Scheme Guidelines

A correctly documented reference diagram has all contact, coil and special functions designated by type and numbered according to the numbering scheme used for the controller. Registers must also be considered in the numbering scheme. Each label consists of a type designation and a reference number. For example:

LL XXXX

- XXXX — Reference number
- LL — Type designation

Type Designations

The type designation indicates whether an input circuit (IN), a coil (CR) or a register bit (BP) will control the contact. The reference number indicates which particular input circuit, coil, or register bit is to control the contact.

The type designations used in PC-1100/1200 controllers are two-letter abbreviations for the function performed. Some of the basic type designations include:

- IN = Input (discrete inputs)

  Inputs associated with real-world devices (e.g., limit switches, pushbuttons and relay contacts).

- IG = Input Group

  A group of 16 discrete inputs handled as a 16-bit register.

- CR = Control Relay (discrete outputs)

  Outputs associated with real-world devices (e.g., motor starters, solenoids and lights).

- OG = Output Group

  A group of 16 discrete outputs handled as a 16-bit register.

- IR = Input Register

  Inputs capable of accepting numerical data from the real-world in 16-bit binary or BCD form.

- OR = Output Register

  Outputs capable of presenting numerical data to the real-world in 16-bit binary or BCD form.

- HR = Holding Register

  Internal registers used for storing internal data.

Reference Numbers

In the PC-1100 and -1200 programmable controllers, reference numbers are assigned in accordance with the limits indicated in Table 4-1. (See Section 5 for more information.)

Holding registers should be assigned consecutive reference numbers starting with HR0001. This is important because the Highest Register Reference Used (HRRU) sets aside a block of memory specifically for holding register use. If, for example, HR0025 were specified, 25 words of memory would be set aside as holding registers whether or not 25 holding registers are actually used. (See Paragraph 4.8.)

4-7. Contacts

Only three types of contacts are used when programming a PC-1100/1200 programmable controller:

- Input contacts
- Control relay contacts
- Bit pick contacts

TABLE 4-1. REFERENCE NUMBER ASSIGNMENT

Contact/Coil Labels [1]	PC-1100 (all models)	PC-1200 1020/1040	PC-1200 1041/1042/1043	PC-1250 (all models)
Discrete Inputs INXXXX	0001 to 0064	0001 to 0064	0001 to 0128	0001 to 0256
Input Groups IGXXXX	0001 to 0004	0001 to 0004	0001 to 0008	0001 to 0016
Discrete Outputs CRXXXX	0001 to 0064	0001 to 0064	0001 to 0128	0001 to 0256
Output Groups OGXXXX	0001 to 0004	0001 to 0004	0001 to 0008	0001 to 0016
Logic Coils CRXXXX [2]	0065 to 0256	0065 to 1024	0129 to 1024	0257 to 1024
Logic Groups OGXXXX	0005 to 0008	0005 to 0032	0009 to 0064	0017 to 0128
Input Registers IRXXXX	0001 to 0008	0001 to 0032	0001 to 0064	0001 to 0128
Output Registers ORXXXX	0001 to 0008	0001 to 0032	0001 to 0064	0001 to 0128
Holding Registers HRXXXX [3]	PC-1100 with 0.5K RAM: 0001 to 0511 PC-1100 with 1.5K RAM: 0001 to 1535 PC-1100 with 2.5/3.5K RAM: 0001 to 1792	0001 to 1792	0001 to 1792	0001 to 1792

1 XXXX = reference number

2 Logic coil CR0128 is assigned the function of Battery Status for the PC-1100. For the PC-1200 and -1250, CR1024 serves this function. This function is described in detail in Section 5.

3 The maximum number of holding registers available for the PC-1100 is determined by memory size minus ladder diagram words.

Input (IN) contacts are used to represent contacts operated by pushbuttons, limit switches and other pilot devices. Control relay (CR) contacts are used to represent contacts operated by the coils of control relays. CR contacts are also used by many special functions (such as timers and counters). Bit Pick (BP) contacts are used to indicate the state of individual bits of register data (e.g., the presence of a bit in shift register(s). (Not all PC-1100 controllers support the BP function.) Figure 4-14 shows an example of each type of contact.

Note that the solution time of IN and CR contacts is much faster than for BP contacts, since they are processed in a special line solver gate array, rather than through the CPU.

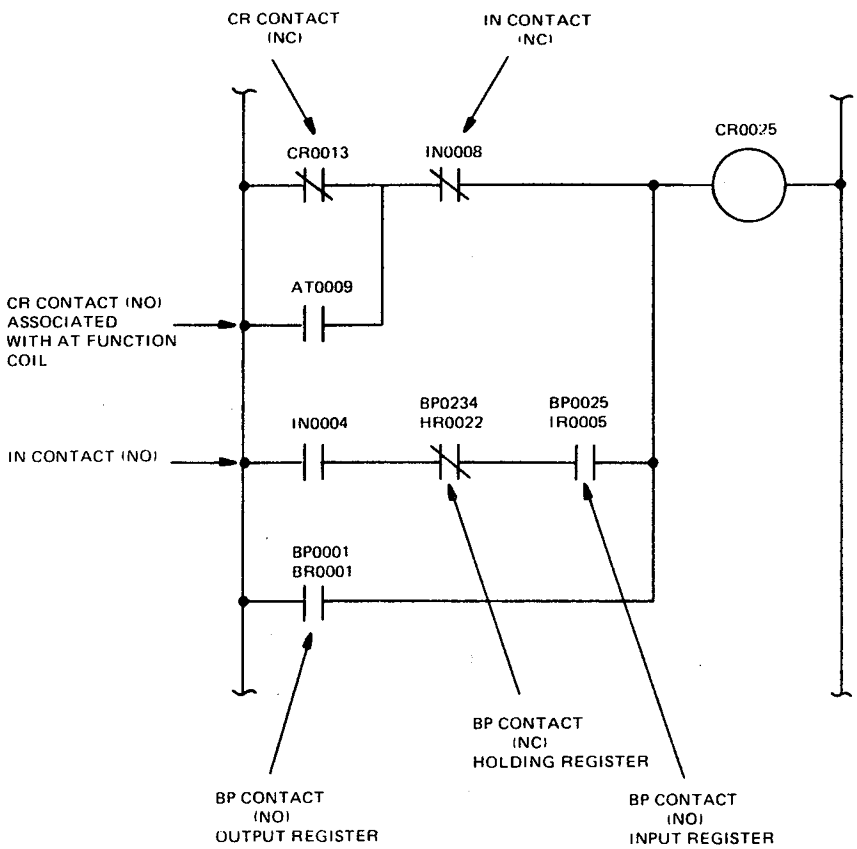


Figure 4-14. Contact Types

Contact Selection

The process of entering each contact in a line involves selecting a normally open (NO) or normally closed (NC) contact, specifying the label of the contact, and assigning the appropriate connections to other elements. Contact operating data is provided in Table 4-2.

TABLE 4-2. CONTACT OPERATING DATA

Contact Type	Normally Open Operation	Normally Closed Operation
IN	Open when the associated circuit is OFF; closed when the associated input circuit is ON.	Closed when the associated input circuit is OFF; open when the associated input circuit is ON.
CR	Open when the associated coil is de-energized; closed when the associated coil is energized.	Closed when the associated coil is de-energized; open when the associated coil is energized.
BP	Open when the associated bit is a logic 0; closed when the associated bit is a logic 1.	Closed when the associated bit is a logic 0; open when the associated bit is a logic 1.

Normally Open (NO) Contacts

NO IN contacts are open when the associated input circuit is OFF (activating voltage not applied). They are closed when the associated input circuit is ON (activating voltage applied).

NO CR contacts are open when the associated coil is de-energized. They are closed when the associated coil is energized.

NO BP contacts are open when the associated bit is a logic 0. They are closed when the associated bit is a logic 1.

Normally Closed (NC) Contacts

NC IN contacts are closed when the associated input circuit is OFF. They are open when the associated input circuit is ON.

NC CR contacts are closed when the associated coil is de-energized. They are open when the associated coil is energized.

NC BP contacts are closed when the associated bit is a logic 0. They are open when the associated bit is a logic 1.

Contact Reference Numbers

Table 4-3 lists the reference numbers used in the PC-1100 and -1200. The following descriptions explain the use of reference numbers.

TABLE 4-3. CONTACT REFERENCE NUMBERS

Function	Controller (Memory Size)	Available Reference Numbers
Input Contact (IN)	PC-1100 (all)	0001 through 0064
	PC-1200 (2K)	0001 through 0064
	PC-1200 (4/8/16K)	0001 through 0128
	PC-1250 (all)	0001 through 0256
Control Relay Contact (CR)	PC-1100 (all)[1]	0001 through 0064 from output coils; 0065 through 0256 from logic coils;
	PC-1200 (2K)	0001 through 0064 from output coils; 0129 through 1024 from logic coils;
	PC-1200 (4/8/16K)	0001 through 0128 from output coils; 0129 through 1024 from logic coils;
	PC-1200 (4/8/16K)	0001 through 0256 from output coils; 0257 through 1024 from logic coils;
Bit Pick [2] Contact (BP)	PC-1100 (0.5K)	Input registers 0001 through 0008; Output registers 0001 through 0008; Holding registers up to 0511
	PC-1100 (1.5K)	Input registers 0001 through 0008; Output registers 0001 through 0008; Holding registers up to 1535
	PC-1100 (2.5/3.5K)	Input registers 0001 through 0008; Output registers 0001 through 0008; Holding register up to 1792
	PC-1200 (2K)	Input registers 0001 through 0032; Output registers 0001 through 0032; Holding registers up to 1792
	PC-1200 (4/8/16K)	Input registers 0001 through 0064; Output registers 0001 through 0064; Holding registers up to 1792
	PC-1250 (8/16K)	Input registers 0001 through 0128; Output registers 0001 through 0128; Holding registers up to 1792

[1] Early versions of the PC-1100 support only 64 logic coils plus the 64 output coils, for a total of 128 coils. The number of logic coils supported can be checked using the program loader "PC Status" screen.

[2] Bits 1 through 16 may be selected within a register. If a bit beyond number 16 is selected, it will be picked from the following register(s). For example, if bit 17 of HR0008 is specified, bit 1 of HR0009 will actually be picked. This feature allows chained 16-bit registers to be used as single registers containing more than 16 bits. Not all PC-1100 controllers support the BP function.

[3] For the PC-1100, the number of holding registers depends on program size.

Input Contacts: The reference number for an input contact designates a particular rack position. input module, input circuit and input device to operate the contact. For example, any contact labeled IN0001 is controlled by the first circuit on the module placed in the rack position assigned discrete reference numbers 0001 through 0008.

Control Relay Contacts: The reference number for a CR contact designates the coil that operates the contact. CR contacts can be operated by the coils of special functions, as well as by control relay coils. For example, a contact entered as CR0027 could be controlled by the counter labeled DC0027.

Even though contacts controlled by special function coils are entered with a CR type designation, they will be displayed with the special function type designation. For example, the contacts controlled by coil TT0101 will be entered as CR0101 contacts. Once coil TT0101 is programmed, the type designation TT0101 will appear when the program loader recalls a CR0101 contact from memory.

Note that the type designations TT, TS, UC, DC, etc., are provided to help interpret circuit operations and are not necessary for the operation of the controller.

Bit Pick Contacts: The reference number for a BP contact designates the bit that operates the BP contact. The label of the register containing that bit must also be specified. For example, BP0012/HR0003 would define a BP contact controlled by the twelfth bit of HR0003.

4-8. Memory and Register Use

After the PC-1100/1200 programmable controller memory has been cleared, the program loader can be used to display the number of words available (in the "PC Status" screen). This number is always two less than the anticipated number (for example, 510 is shown instead of 512, etc.). This occurs because two words of the total memory are never available for use, and therefore are not included in the "words available" tabulation.

One of these words is used to designate the End-of-Program (EOP). The other indicates the first holding register (HR0001). (See Figure 4-15.) The program, consisting of ladder diagrams, uses memory from the bottom, pushing up the location of the EOP word after checking to ensure that enough memory is available for each ladder diagram before it is input.

Registers are assigned memory from the top starting at HR0001, as illustrated in Figure 4-15. The PC-1100/1200 programmable controller always assumes that a reference has been made to HR0001, even if no such reference has in fact been made in the program. The controller keeps track of the highest register reference used (HRRU) and uses this value (in relation to HR0001) to set aside a block of user memory for holding register use. If, for example, HR0200 was the only holding register reference made in the program, 200 words of user memory would still be set aside even though no other holding registers were actually used. Thus, it is important to reference holding registers in numerical order starting with HR0001.

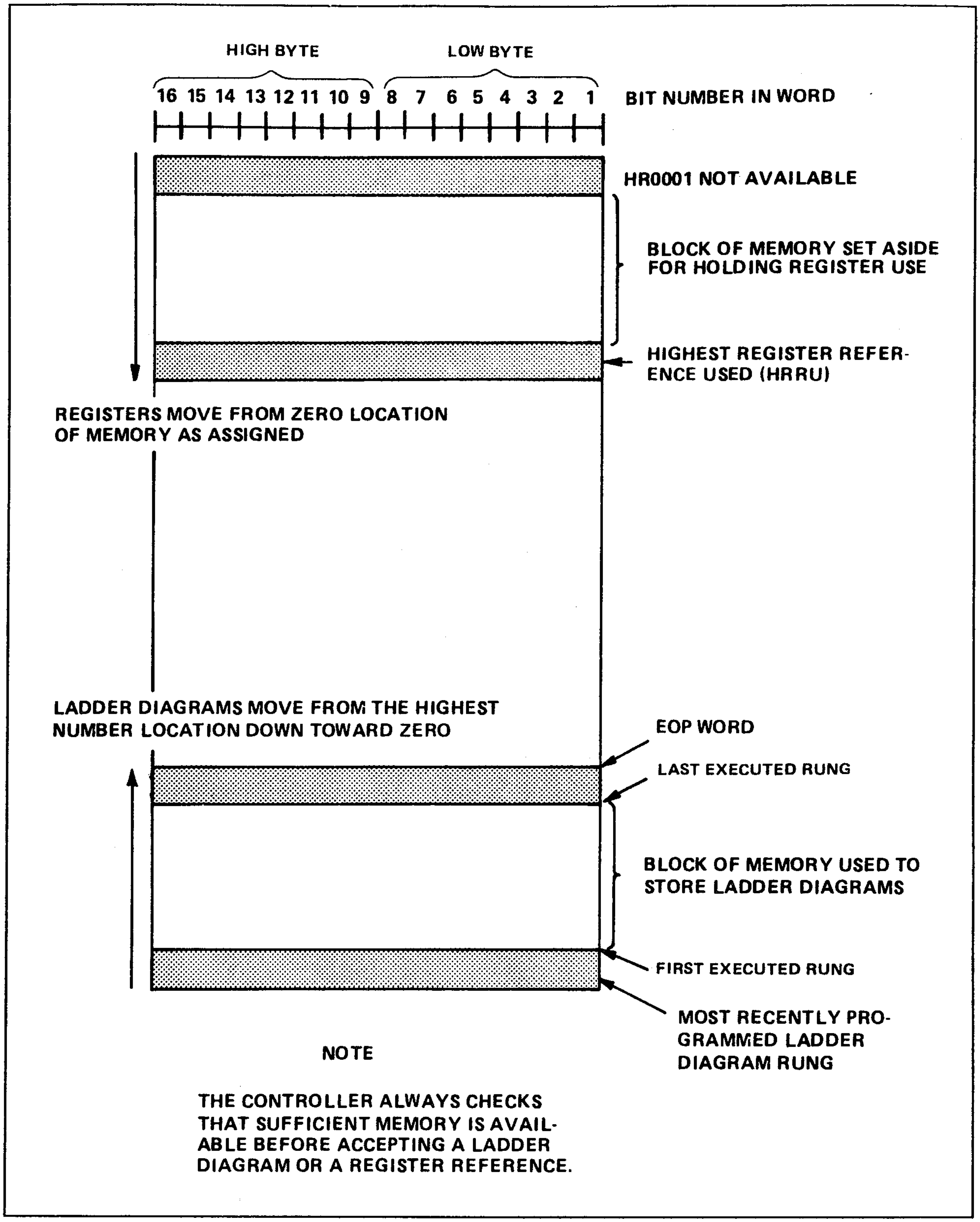


Figure 4-15. Program/Register Memory Assignment

An exception to this general rule applies when using the assignable keys feature of the NLPL-789 Mini Loader. In this case, HR0001 through HR0007 should be reserved, starting program references at HR0008 or higher.

Noting the significance of the HRRU, the amount of memory required for a given program can be established by determining the ladder memory used by the coils, contacts, and special functions, and then adding the HRRU. For example, a simple program might consist of the following elements:

10 IN contacts	@ 1 word	=	10 words
3 CR contacts	@ 1 word	=	3 words
2 Timers	@ 3 words	=	6 words
1 Down Counter	@ 3 words	=	3 words
3 CR coils	@ 1 word	=	3 words
	Subtotal	=	25 words
	plus HRRU	=	6 words
	Subtotal	=	31 words
	plus EOP	=	1 word
	Total	=	32 words

The ladder memory required by each programmable function can be found in Section 5. Keep in mind that the memory usage listed in Section 5 always refers to the *ladder memory* used to program the special function and its coil. It does not include the input structure or holding register use.

4-9. Scan Times

The time required to scan the PC-1100/1200 reference ladder diagram depends on the number and type of contacts, coils, and programmable functions used. Table 4-4 provides scan time information for the PC-1100 and -1200.

4-10. COIL UTILIZATION

Output coils, either control relay (CR) or special function, can be programmed in a variety of ways. It is important to note that a truly effective program not only performs its intended function, but also is designed for maximum efficiency and minimum memory consumption. The programmer determines the manner in which techniques are applied to optimize both time and storage space. With practice, the programmer becomes adept at structuring programs for effectiveness. Some techniques used to achieve this objective are discussed below:

- Dummy Coils
- Rapid clocking pulse
- Transitional functions
- Programming order
- Multiple programs

TABLE 4-4. SCAN TIMES[1]

Function	PC-1100		PC-1200	
	OFF	ON	OFF	ON
Coils and Contacts				
CR Coil	86.0	86.0	34.0	34.0
SK Skip	140.0	310.0	50.0	58.0
MR Master Relay	360.0	140.0	58.0	50.0
BS/BC-HR Bit Set/Clear	140.0	230.0	50.0	70.0
BS/BC-OR Bit Set/Clear	140.0	250.0	50.0	72.0
BF-HR Bit Follow	230.0	230.0	69.0	69.0
BF-OR Bit Follow	250.0	250.0	71.0	71.0
CR or IN Contact	7.0	7.0	0.7	0.7
BP-HR Contact	229.0	229.0	45.0	45.0
BP-IR/OR Contact	238.0	238.0	48.0	48.0
Timers and Counters				
TT or TS Timer	340.0	340.0	84.0	100.0
UC Up Count	340.0	430.0	84.0	95.0
DC Down Count	310.0	410.0	84.0	95.0
Math				
AD Add	240.0	590.0	60.0	110.0
SB Subtract	240.0	430.0	60.0	110.0
MP Multiply	240.0	1160.0	60.0	113.0
DV Divide	240.0	1160.0	60.0	125.0
SQ [19] Square Root	440.0	7160.0	63.0	175.0
Comparisons/Conversions				
EQ-GE	150.0	250.0	59.0	79.0
BD Binary to BCD	260.0	670.0	64.0	137.0
DB BCD to Binary	260.0	990.0	64.0	232.0
Moves				
MV Move Register	200.0	310.0	63.0	90.0
TR [84] Table to Register	1070.0	1100.0	61.0	146.0
IM [95] Indirect Move	400.0	760.0	66.0	126.0
MB [63] Move Byte	340.0	1300.0	64.0	108.0
BT [60] Block Transfer	700.0	780.0	67.0	133.0
Per register add		5.2		0.7

[1] All times given in microseconds.

TABLE 4-4. SCAN TIMES, CONT'D[1]

Function	PC-1100		PC-1200	
	OFF	ON	OFF	ON
Stack Functions				
FI [81] First In	999.0	999.0	87.0	145.0
LO [86] Last Out	999.0	1090.0	83.0	140.0
FO [85] First Out	999.0	999.0	83.0	160.0
Per register moved		15.0		8.0
Table Operations - Bit				
NR-NL N-Bit Shift	670.0	770.0	60.0	135.0
Per Register - 1 Bit		14.4		4.4
Per Register - 8 Bits		117.6		36.3
Per Register - 16 Bits		235.6		72.8
BO [61] Bit Operate	790.0	870.0	129.0	129.0
SM [56] Search Matrix	670.0	1080.0	65.0	150.0
Per register to bit		41.0		15.2
CM [57] Compare Matrix	590.0	660.0	65.0	150.0
Per register add		22.0		10.4
XM [88] XOR Matrix	999.0	1280.0	68.0	190.0
Per register add		41.0		35.6
OM [89] OR Matrix	1010.0	1290.0	68.0	190.0
Per register add		41.0		34.7
AM [90] AND Matrix	1030.0	1310.0	68.0	190.0
Per register add		38.0		35.1
Miscellaneous				
I/O Bus Cycle [2]	3510.0	3510.0	9870.0	9870.0
UI Update Immediate	170.0	480.0	55.0	204.0
CG [15] Cont. Group	300.0	320.0	62.0	147.0
LS [26] Lock Scan	340.0	1300.0	60.0	85.0

[1] All times given in microseconds.

[2] The stated I/O bus cycle time is for all I/O. By using the US function to update I/O by quarters, I/O bus cycle time can be significantly reduced.

4-11. Dummy Coils

A dummy coil is one that is never energized and whose contacts never change state. It might appear that the easiest method of developing a "dummy" coil is simply not to program the coil. However, there are two conditions that make this impractical. The first is that the coil could be forced ON and, with no logic to cause it to change state, remain ON when the force is deleted. The second drawback is that if someone else uses the program, an undocumented dummy coil could be programmed to serve an entirely different function, creating confusion and potential danger.

The best method for programming a dummy coil is illustrated in Figure 4-16. As shown, forcing coil CR0127 causes its contacts to change state. When the force is removed, the coil returns to the OFF state. In addition to consistently functioning properly, another benefit of using this circuit is that the coil is documented on any printout or drawing, guarding against erroneous future use.

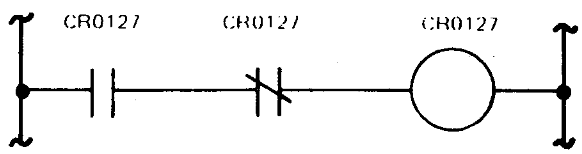


Figure 4-16. Dummy Coil Circuit

4-12. Rapid Clocking

If a rapid clock signal is required, a simple rapid clock can be programmed, as illustrated in Figure 4-17. In this circuit, the scanning technique used in the controller causes the controller to see CR0126 as being OFF and the NC contacts of CR0126 as being closed (because CR0126 is OFF). It then energizes coil CR0126. On the next scan, the controller sees coil CR0126 as being ON and CR0126 NC contacts as open (because coil CR0126 was energized during the last scan). Coil CR0126 will then be turned OFF. The cycle then repeats itself. Coil CR0126 is ON every other scan.

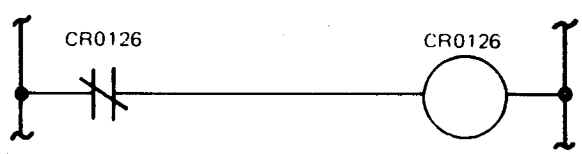


Figure 4-17. Rapid Clocking Circuit

4-13. Transitional Functions

Many of the special functions are transitional in operation. This means that a change in state of a set of contacts (non-conducting to conducting) is required to perform the function. Following are some examples of functions that are transitional in operation:

- Counters
- Shift Registers
- Add
- Subtract
- Conversions

For these transitional functions, a truth table is included in the function description in Section 5. The truth tables, which describe the result of each possible value of the function's enabling circuit, use the following format:

- 0 represents a non-conducting circuit
- ↑ represents the transition (non-conducting to conducting)
- 1 represents a conducting circuit

The processor determines that the change in state has occured by checking the history bit of the associated coil. For transitional functions, if this bit is OFF when the contact matrix solves true, then the function is performed (and the history bit is set to ON). If the history bit is ON when the contact matrix solves true, the function is not performed (and the history bit remains ON). When the matrix solves false, the history bit is cleared (set to OFF).

Note

> While the transitional functions use the history bit as described above, other functions may use the history bit in other ways.

In the PC-1100, only transitional functions use the history bit. In the PC-1200, the following continuous functions use the history bit: LC, RW, SK, TS, and TT. Because the history bit is associated with the coil, *do not duplicate the coil number* of any function which uses the history bit.

In the case of counters and shift registers, transitional operation is a necessary and desired part of the function. However, in the case of the arithmetic and conversion functions, a transitional operation may or may not be desirable. If the controller does not have a Continuous Group Select (CG) function, scanning can be made to occur on a continuous (every scan) basis by using the format illustrated in Figure 4-18. (For additional information, refer to the CG function description).

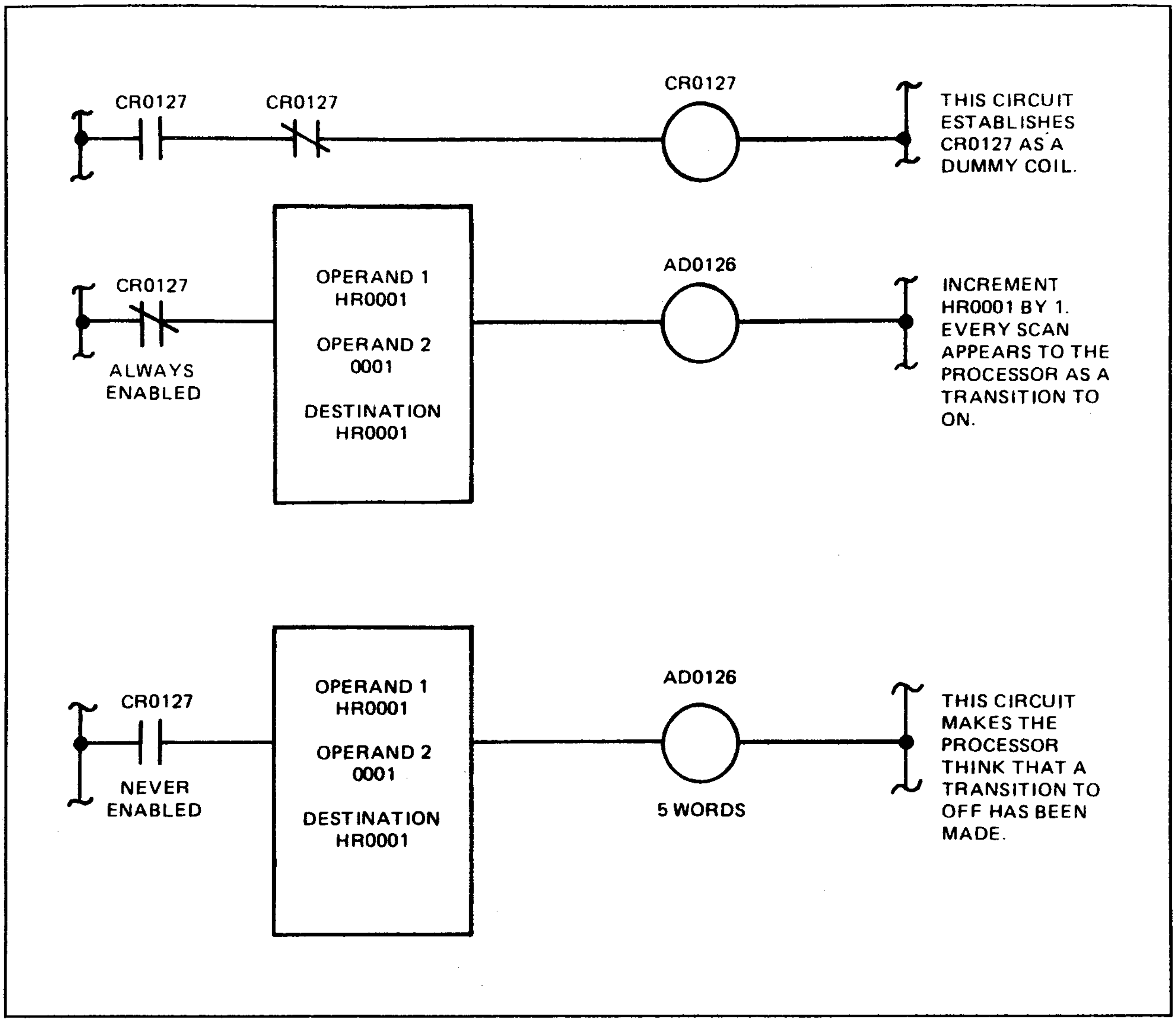


Figure 4-18. Every Scan Operation

This form requires only that the second coil be that of a transitional function. In the example shown in Figure 4-18, UC0126 could be used. The assignment of operand and designation is unimportant, since the function is not used. (See Paragraph 4-8 for a discussion of HRRU and register manipulation.)

4-14. Program Order

In general, the exact placement of a coil within the program is not critical. However, when determining program order, there may be instances that do require careful consideration of the location of a coil. It is important to avoid the following situations since improper program order can result in many hours of problem analysis.

Example 1

The circuit shown in Figure 4-19 illustrates one of these situations. Since coil CR0059 is currently OFF, coil CR0025 energizes through the NC contacts of CR0025. Coil CR0011 does not energize because the NO contact of CR0059 is open. Since coil CR0025 is now energized, coil CR0059 is energized. Scanning the circuit a second time finds the NC contacts of CR0059 now open, turning coil CR0025 OFF. CR0011 does not energize because this time the NO contacts of CR0025 are open, and coil CR0059 turns OFF. As a result, CR0025 and CR0059 alternate ON and OFF, but CR0011 is never ON.

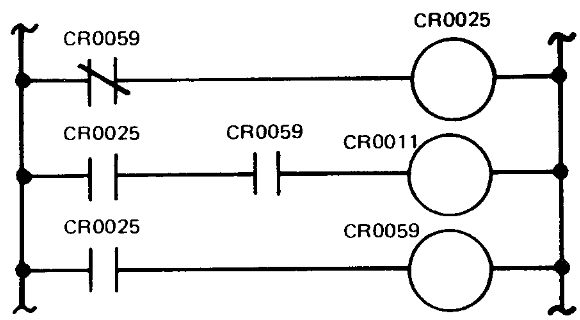


Figure 4-19. Improper Program Order

The results of the program order in Figure 4-19 are shown below:

Coils	Scan 1	Scan 2	Scan 3	Scan 4
CR0025	ON	OFF	ON	OFF
CR0011	OFF	OFF	OFF	OFF
CR0059	ON	OFF	ON	OFF

Changing this programming situation requires changing the program order. Figure 4-20 illustrates what is required to correct the programming problem discussed above.

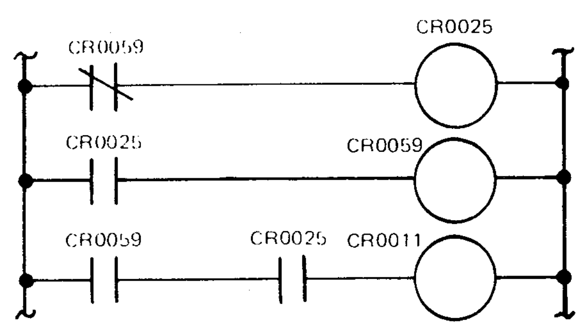


Figure 4-20. Correct Program Order

The results of the program order in Figure 4-20 are shown below.

Coils	Scan 1	Scan 2	Scan 3	Scan 4
CR0025	ON	OFF	ON	OFF
CR0011	ON	OFF	ON	OFF
CR0059	ON	OFF	ON	OFF

Example 2

Another situation that requires attention to program order occurs with the use of output groups, as with a Table-to-Register (TR) function. In Figure 4-21, the TR function is identified by the coil reference number 0001. Its destination is the output group OG0001, which consists of the discrete outputs 1 through 16. Placing the coil reference number in the same output group as the destination defeats the purpose. This could cause the user program to malfunction since invalid information may be transferred to the destination register.

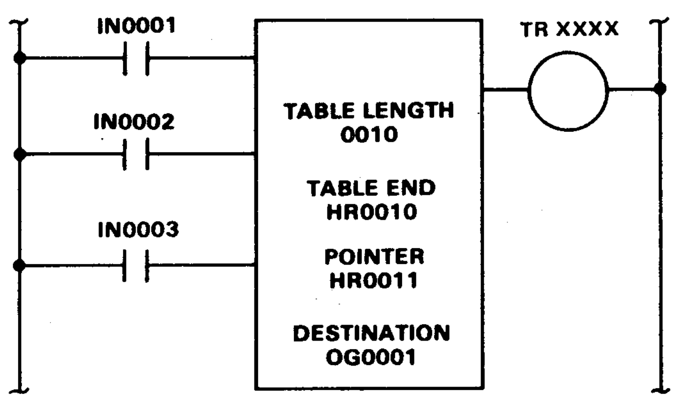
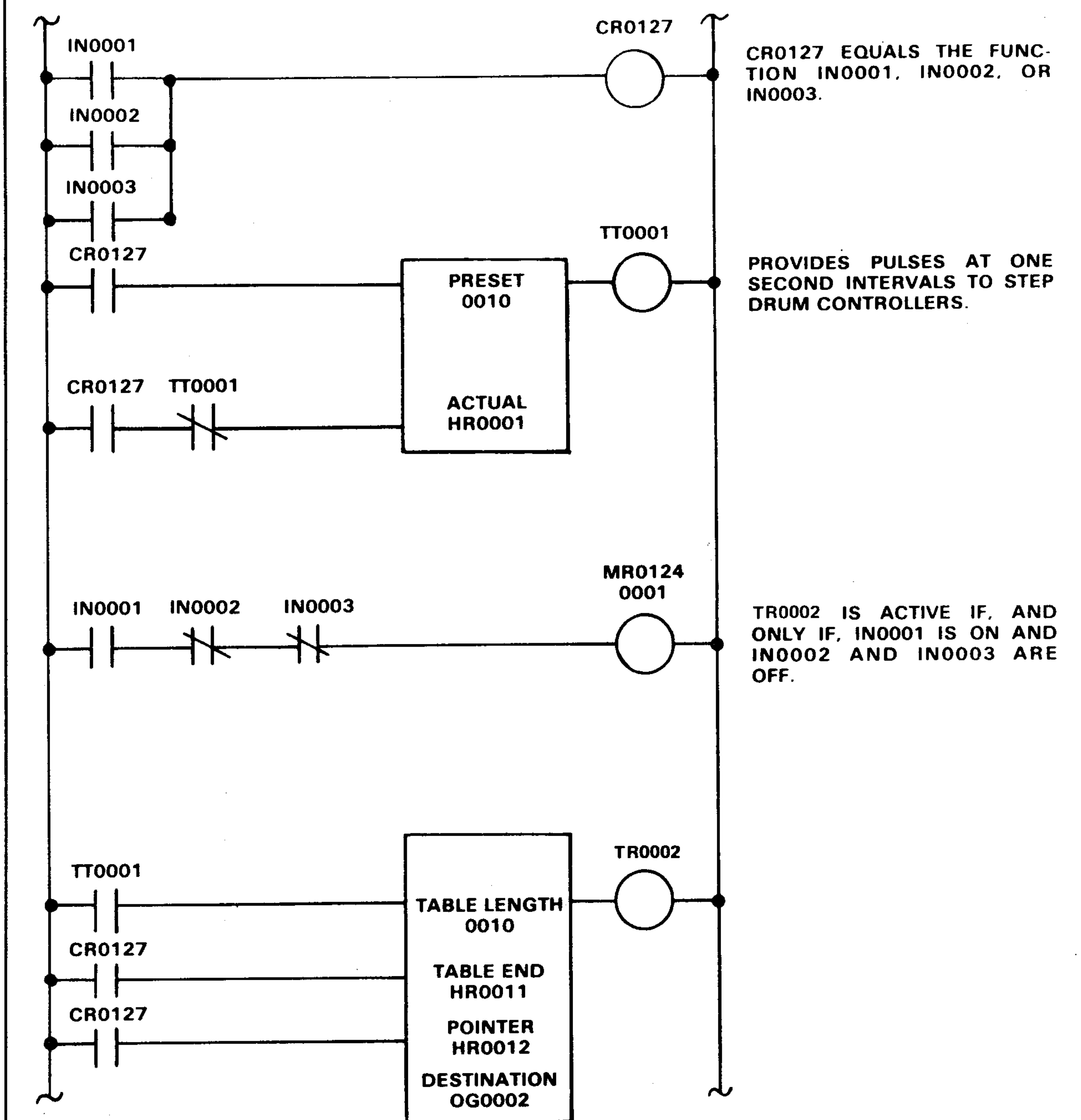


Figure 4-21. Table-to-Register Pitfalls

4-15. MULTIPLE PROGRAMS

A PC-1100/1200 programmable controller is capable of performing a variety of sequences or operations on command. This capability is desirable in many instances. In the example shown in Figure 4-22, three separate Table-to-Register functions are used to operate the same set of outputs (CR0017 through CR0032) through three separate sequences. This example has several features that are of interest:

- Every Master Control Relay (MR) is controlled by mutually exclusive circuits (i.e., the logic is such that one and only one program can be selected at any given time).
- If no program is selected, all outputs (CR0017 through CR0032) are turned OFF.
- The functions are combined, when possible, into a single coil such as rung 1 of the program.



(Cont'd.)

Figure 4-22. Multiple Programs

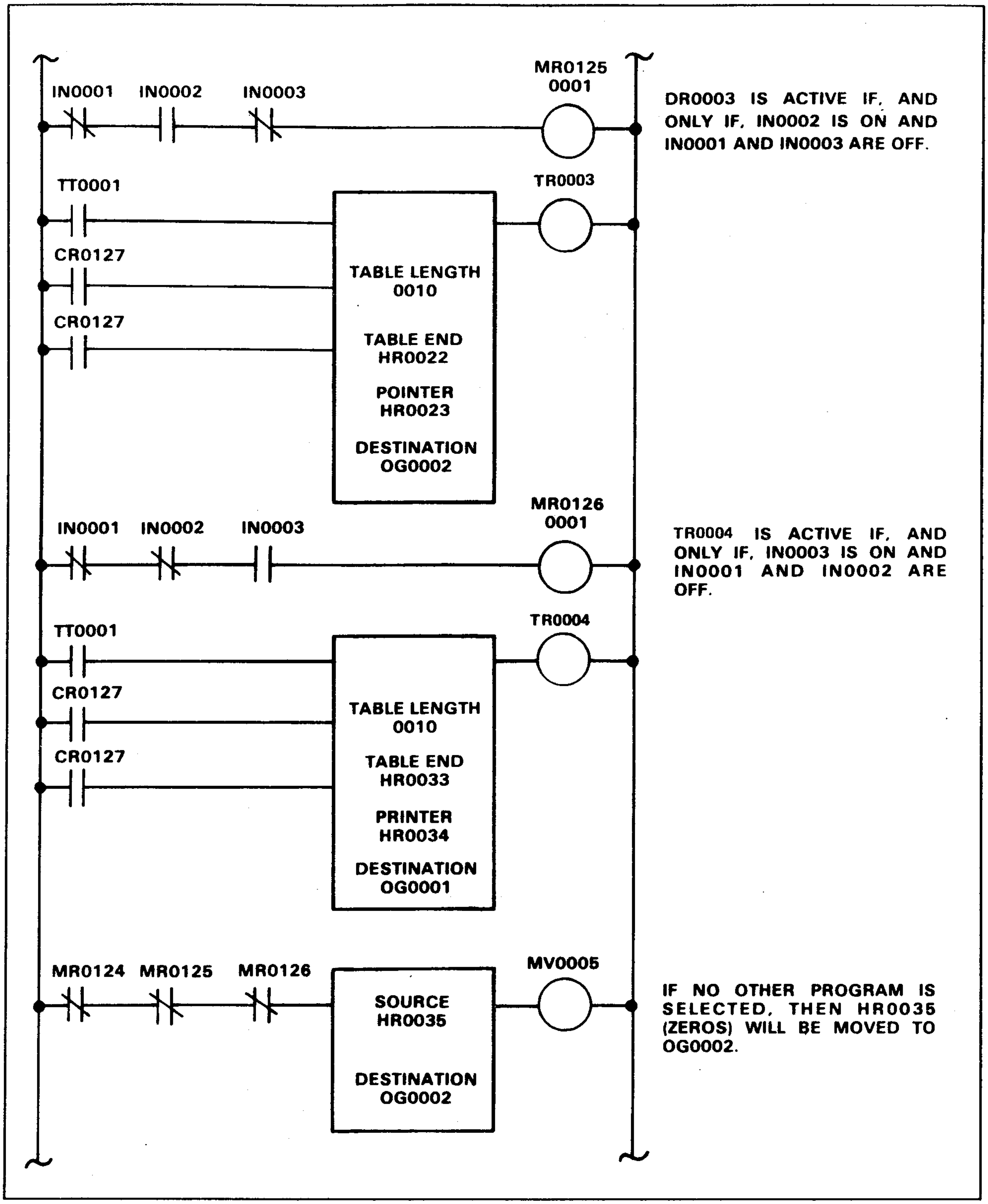


Figure 4-22. Multiple Programs (Cont'd.)

It should be noted that this program could easily be expanded with each Table-to-Register function assuming different lengths and each being stepped by an independent timing chain.