SECTION 3. PROGRAMMING
PC-700/900 Programmable Controllers

3-1. GENERAL DESCRIPTION

Programming the **Numa-Logic** PC-700 or PC-900 Programmable Controller system is simply a way to specify a sequence of operations the system is to perform. The PC-700 and PC-900 processors, through programming, digitally duplicate the operation of a relay control system. PC-700 and PC-900 processor programs consist of a set of programmable functions which specify the relay circuits and devices to be duplicated and the operation sequences to be performed.

In PC-700 and PC-900 processor applications, the user's program uses these programmed functions to 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 Sections 4 and 5.

The reference ladder diagram clearly shows the types of contacts used in each 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 processor and makes the function of these circuits easier to understand.

3-2. REFERENCE LADDER DIAGRAM

The **Numa-Logic** Programmable Controllers use reference ladder diagrams to document an installed control program. **Numa-Logic** reference ladder 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 **Numa-Logic** 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 **Numa-Logic** reference ladder diagrams, input devices (e.g., pushbuttons, limit-switches, pressure switches, etc.) are shown as relay contacts. Figure 3-1 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).

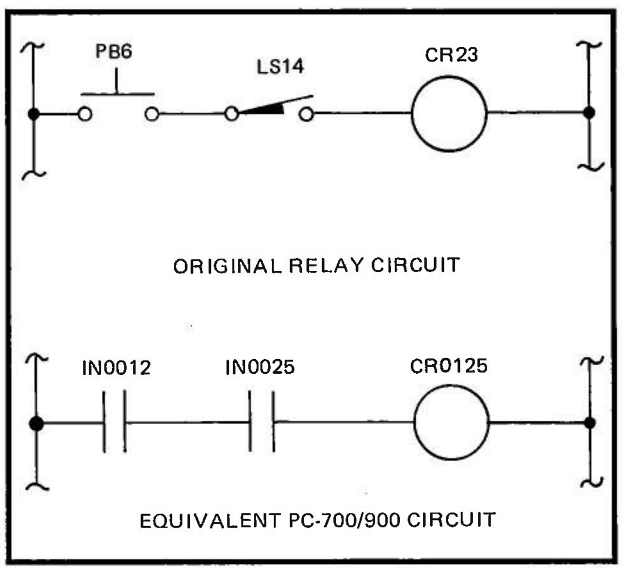

Figure 3-1. Ladder Diagram Input Devices

Also different from a relay ladder diagram is that 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 3-7.

As shown in Figure 3-1, 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 in this manual, real-world input devices (e.g., pushbuttons and switches) are connected to converter circuits on input modules. These input circuits convert real-world signal levels to processor signal levels. Also illustrated in Figure 3-1 are reference numbers indicating 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 3-2.) 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 contacts and is also associated with a 700 series output converter circuit and a real-world output device. The second kind of coil, called a logic coil, only controls contacts; it cannot directly control a real-world output device.

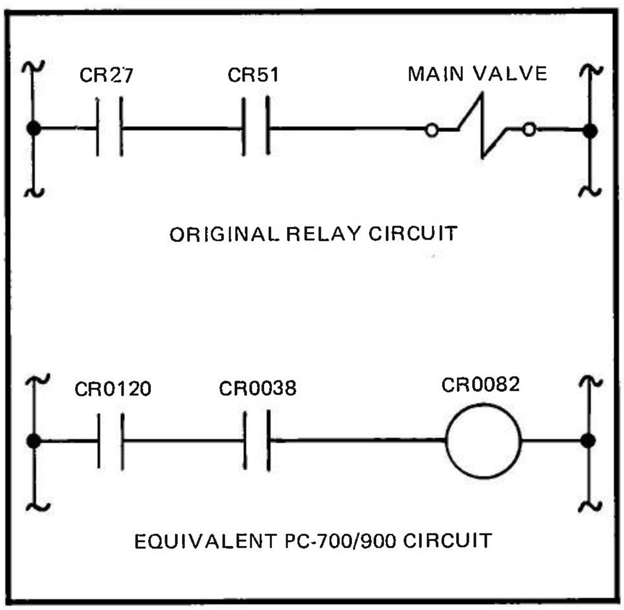

Figure 3-2. Outputs Shown as Coils

Another difference between the reference diagram and the original ladder diagram is that many special functions (e.g., timers, counters, shift registers, addition, drum control, etc.) are shown as simple blocks. Each special function has one to three contact circuits which control its operation (with the exception of the Loop Controller, which has four contact circuits). For example, Figure 3-3 shows a timer. The timer accumulates time when both the timing circuit and the enable circuit are conducting.

In general, special function labels are similar to the labels used with coils. The type portion of the label indicates what the special function does. For example, TS (Timer in Seconds) indicates a timer with a resolution of one second (preset at 1 — 9999 for a constant or 1 — 65535 for a register). 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 3-3 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; it may be stored in some memory location by another function or by some real-world input. Thumbwheels, for example, could be used to set the preset value for a timer.

Another issue that the timer example illustrates is that special functions use numerical data. Special functions accept numerical data from registers and supply similar numerical data to registers, in addition to having discrete inputs and outputs (contact circuits and coils). When numerical data is required by a special function, its storage location (input register, output register, or holding register) must be specified.

For example, in Figure 3-3, the actual value of the timer is stored in Holding Register 5 (HR0005). Registers may be associated with more than one special function. One special function may store numerical data in a register which is also used to supply numerical data to another special function. Thus, an output register (a register whose value may, for example, operate a seven segment display) can also provide numerical input data to another special function. Registers can perform a variety of functions, depending upon the application.

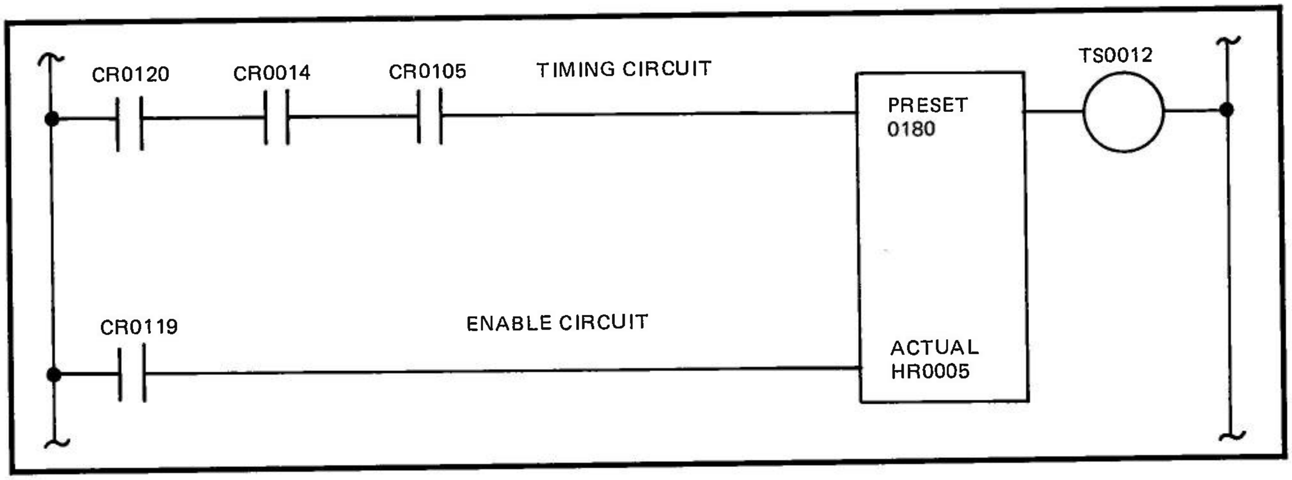

Figure 3-3. Special Function Block with Multiple Contact Circuits

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

3-3. CONSIDERATIONS

3-4. PROCESSOR SCANNING

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

First, the processor examines all inputs to the system and stores their states in a portion of its memory.

Then, the processor scans the programmed circuits, starting with the first circuit programmed (top of the reference diagram), 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 processor 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 3-4, for example, Coil CR0062 is controlled by the status established for IN0018 and CR0053 during the current scan, and by the status established for CR0317 during the previous scan (since CR0317 has not yet been scanned when CR0062 is being scanned).

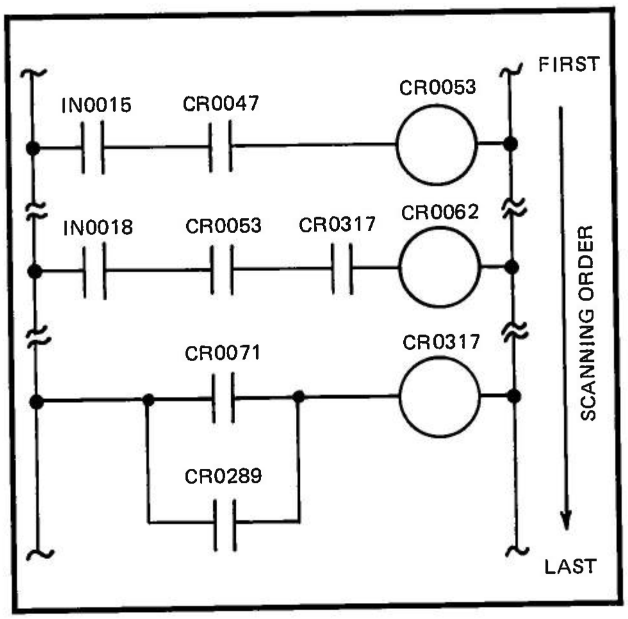

Figure 3-4. Processor Scanning

As the circuits are scanned, the processor 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. Circuits should be drawn and programmed in the order in which they should conduct to produce the desired system operation. The order in which circuits are drawn and programmed should be evaluated in light of the following discussion concerning conduction within ladder circuits.

3-5. CONDUCTION WITHIN LADDER CIRCUITS

When the processor is scanning the programmed circuits, conduction through contacts is considered to occur only from left to right. Conduction in branches may also occur vertically (either up or down), as well as from left to right.

For example, consider the circuit shown in Figure 3-5. 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.

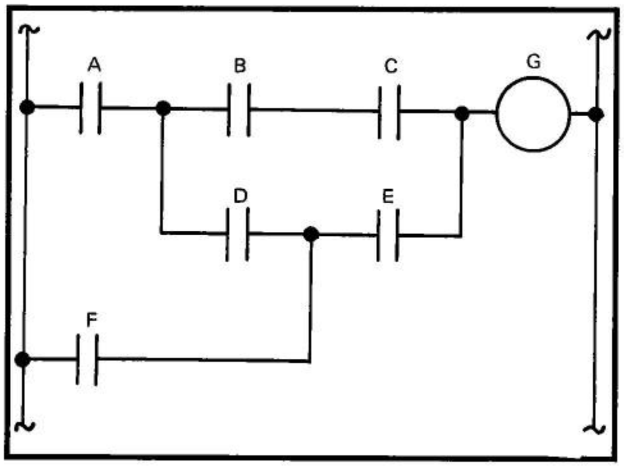

Figure 3-5. Circuit Conduction

3-6. 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. If there are two junctions on a horizontal line, there must be a contact between them.

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

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 seven 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 eight series contacts in any path among a maximum of seven parallel rows.

Note

Refer to the "CRT Programming Manual" for more information on program entry.

RULE 1

To illustrate Rule 1, consider Figure 3-6. 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 and the direction of conduction through Contact C is from left to right, as discussed previously. Note that in this circuit, the Path D to C to B does not energize Coil F. Figure 3-7 shows the same circuit with the Path D to C to B included to energize Coil F.

RULE 2

To illustrate Rule 2, consider Figure 3-8. 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.

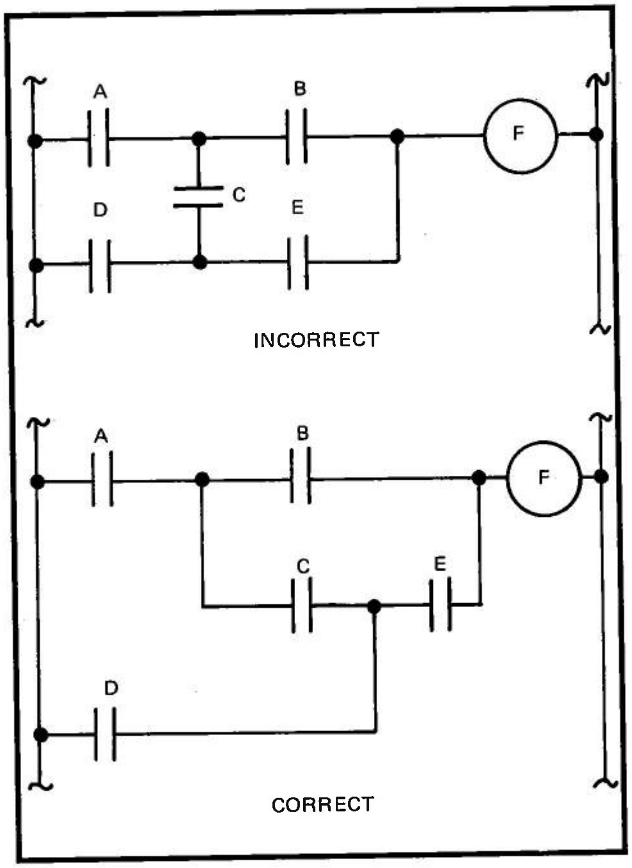

Figure 3-6. Rule 1: Contacts Must Occur in Horizontal Branches.

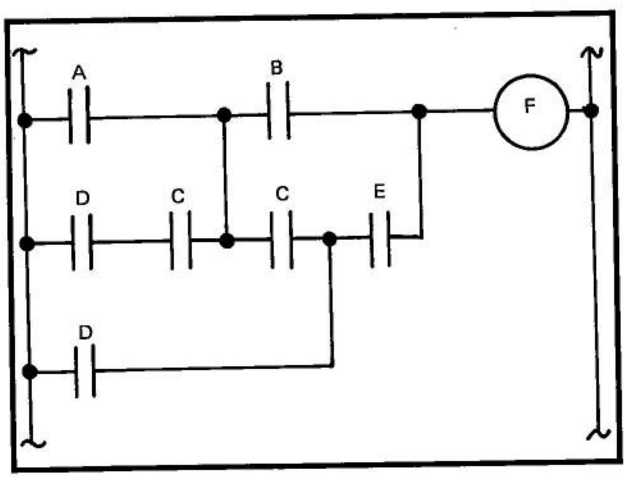

Figure 3-7. The Addition of Two Contacts Adds Path D to C to B

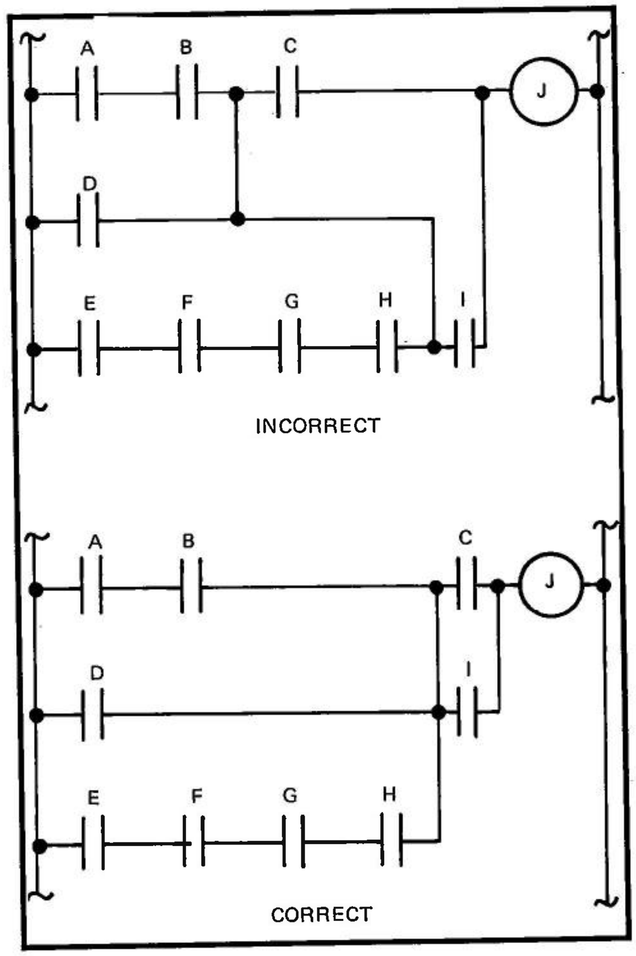

Figure 3-8. Rule 2: Branches Should Run Vertically

RULE 3

To illustrate Rule 3, consider Figure 3-9. 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, 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.

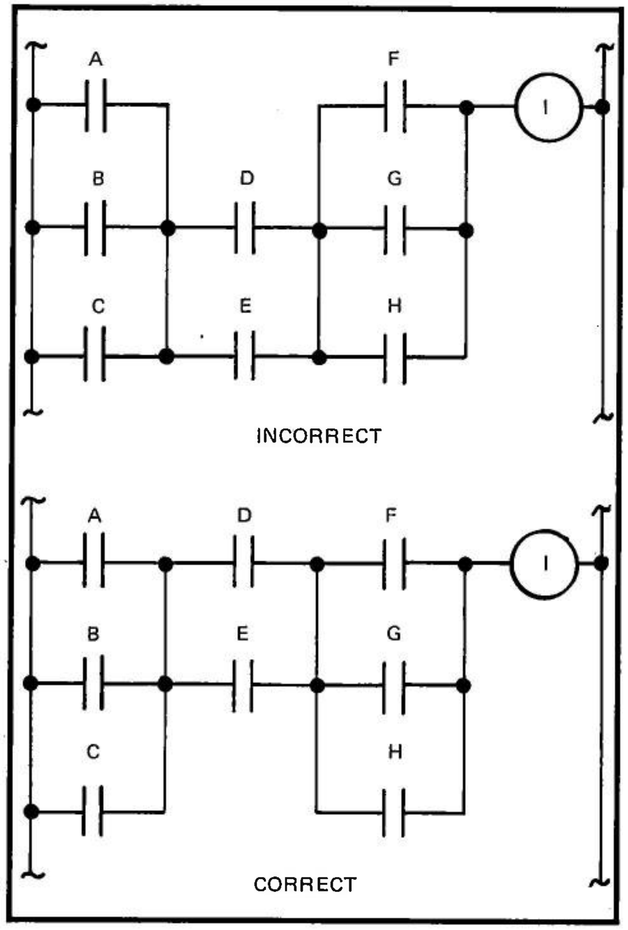

Figure 3-9. Rule 3: Contacts Should Be Connected to the Topmost Available Junctions

RULES 4 AND 5

Rules 4 and 5 simply state what limits are placed on the size of the contact area available for each different function. Figures 3-10 and 3-12 illustrate this concept. Also, 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 3-11.)

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 3-13 illustrates a legal path, but because of the number of contacts, it would not be acceptable. In other words, any line that complies with Rules 4 and 5 is a valid line, even though the line may be drawn such that Rules 1, 2, and 3 are violated.

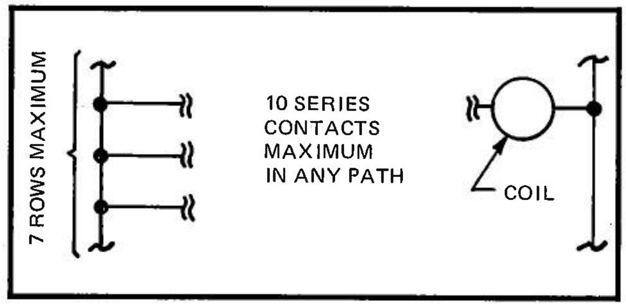

Figure 3-10. Rule 4: 10 x 7 Contact Area for Coils

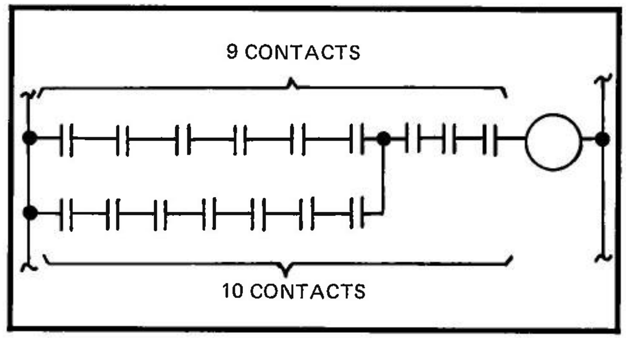

Figure 3-11. Example of Two Legal Paths

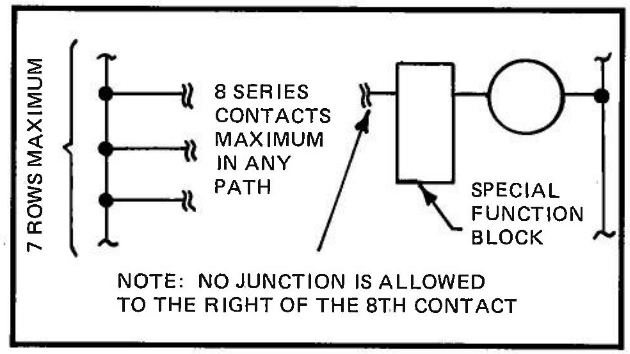

Figure 3-12. Rule 5: 8 x 7 Contact Area for Special Functions

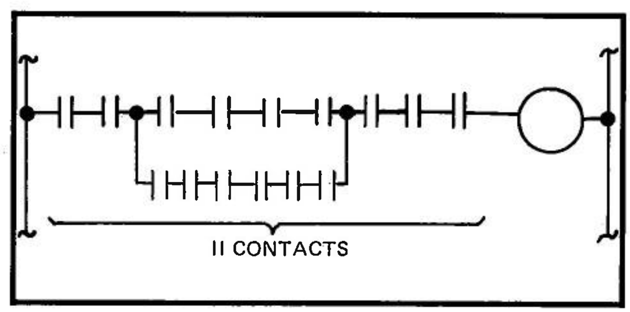

Figure 3-13. Example of an Illegal Path

3-7. LABELING SCHEME GUIDELINES

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

LL NNNN

- **LL** — Type Designation
- **NNNN** — Reference Number

Type Designation

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.

Type designations used in **Numa-Logic Programmable Controllers** are two-letter abbreviations for the function performed. (See Section 4.) Some of the basic type designations of a label include:

IN = Input (discrete inputs)
IG = Input Group
CR = Control Relay (discrete outputs)
OG = Output Group
IR = Input Register
OR = Output Register
HR = Holding Register

Other type designations are also used. These are addressed (where the topic is discussed) elsewhere in this manual.

Discrete Input Circuit: An input associated with a real-world device (e.g., limit-switch, pushbutton, or relay contact).

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

Discrete Output Circuit: An output associated with a real-world device (e.g., motor starter, solenoid, or light).

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

Input Register: An input capable of accepting numerical data from the real world in 16-bit binary or BCD form.

Output Register: An output capable of presenting numerical data to the real world in 16-bit binary or BCD form.

Reference Number

In **Numa-Logic** Programmable Controllers, reference numbers are assigned in accordance with the limits indicated in Table 3-1. (See Section 4 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 is specified, 25 words of memory are set aside as holding registers whether or not 25 holding registers are actually used.

3-8. CONTACTS

Only three types of contacts are used when programming a **Numa-Logic** Programmable Controller:

- Input contacts
- Control Relay contacts
- Bit Pick contacts

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 and of most special functions (e.g., timers or 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 a shift register).

TABLE 3-1. REFERENCE NUMBER ASSIGNMENT

	**PC-700 Standard Function Software 2.X				**PCX-700, PC-700 (See Note)	***PC-900A	PC-900B
Memory Size	256	512	1024	2048 +	All	All	All
Discrete Inputs IN XXXX	1 through 64	1 through 128	1 through 192	1 through 256	1 through 256	1 through 128	1 through 128
Input Groups IG XXXX	1 through 4	1 through 8	1 through 12	1 through 16	1 through 16	1 through 8	1 through 8
Discrete Outputs LL XXXX	1 through 64	1 through 128	1 through 192	1 through 256	1 through 256	1 through 128	1 through 128
Output Groups OG XXXX	1 through 4	1 through 8	1 through 12	1 through 16	1 through 16	1 through 8	1 through 16
Logic Coils LL XXXX	65 through 512	129 through 512	193 through 512	257 through 512	257 through 512	——	129 through 256
Input Registers IR XXXX	1 through 8	1 through 16	1 through 24	1 through 32	1 through 32	1 through 8	1 through 16
Output Registers OR XXXX	1 through 8	1 through 16	1 through 24	1 through 32	1 through 32	1 through 8	1 through 16
Holding Registers HR XXXX	1 through 256*	1 through 512*	1 through 1024*	1 through 1792*	1 through 1792*	1 through 1792*	1 through 1792*

*The maximum number of holding registers available is memory size minus ladder diagram words with an upper limit of 1792 total holding registers.

**PCX-700 Extended Function Software 1.X, 3.X, 5.X, and PC-700 Advanced II Function Software 6.X

***No longer offered.

Notes

1. The PC-700 with Version 6.2 software has 256 words that are unavailable for user memory.

2. The PC-700 with Version 6.2 software and the PC-900 with Version X.7 support Westnet II™ Data Highway operation.

3-9. 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. This includes the type and reference number as discussed earlier in this section, and assigning the appropriate connections to other elements. Contact operating data is provided in Table 3-2.

Normally-Open (NO) Contacts

- NO IN contacts are open when the associated input circuit is OFF (activating voltage not applied), and closed when the associated input circuit is ON (activating voltage applied).
- NO CR contacts are open when the associated coil is de-energized, and closed when the associated coil is energized.
- NO BP contacts are open when the associated bit is a logic 0, and 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, and open when the associated input circuit is ON.
- NC CR contacts are closed when the associated coil is de-energized, and open when the associated coil is energized.
- NC BP contacts are closed when the associated bit is a logic 0 and open when the associated bit is a logic 1.

3-10. DESCRIPTION

Table 3-3 lists the reference numbers used in **Numa-Logic** Programmable Controllers. Following are descriptions explaining the use of reference numbers.

Input Contacts: The reference number for an input contact designates a particular rack, 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 top module of the rack whose switches are set to position 1/1/1.

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, this does not mean that all contacts controlled by coils should be labeled with a CR type designation on the reference diagram. On the contrary, contacts controlled by special function coils should be shown with the same type designation as the coil. (All contacts controlled by Coil TT0101 should be labeled TT0101 on the reference diagram. When a TT0101 contact is programmed, it is entered as a CR0101 contact. When the program loader recalls a TT0101 contact from memory, the type designation TT will appear in the type display, provided that the coil TT0101 has been previously programmed.) Note that the type designations (e.g., TT, TS, UC, DC, etc.) are provided to help interpret circuit operations and are not necessary for the operation of the processor.

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 defines a BP contact controlled by the twelfth bit of HR0003.

TABLE 3-2. 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.

TABLE 3-3. CONTACT REFERENCE NUMBERS

Function	Processor	Memory Size	Available Reference Number
Input (IN) contact	PCX-700 Software *1.X *3.X *5.X PC-700 Software 6.X	All	1 through 256
	PC-700 Software *2.X	256 512 1024 2048 through 4096	1 through 64 1 through 128 1 through 192 1 through 256
	PC-900A/B	All	1 through 128
Control Relay (CR) contact	PCX-700 Software *1.X *3.X *5.X PC-700 Software 6.X	All	1 through 256 from output coils 257 through 512 from logic coils
	PC-700 Software *2.X	256	1 through 64 from output coils 65 through 512 from logic coils
		512	1 through 128 from output coils 129 through 512 from logic coils
		1024	1 through 192 from output coils 193 through 512 from logic coils
		2048 through 4096	1 through 256 from output coils 257 through 512 from logic coils
	*PC-900A	All	1 through 127 from output coils CR0128 is a battery status coil
	PC-900B	All	1 through 127 from output coils CR0128 is a battery status coil 129 through 256 from logic coils

TABLE 3-3. CONTACT REFERENCE NUMBERS (Cont'd)

Function	Processor	Memory Size	Available Reference Number
Bit Pick (BP) contact	PCX-700 Software *1.X *3.X *5.X PC-700 Software 6.X	All	Input Registers 1 through 32 Output Registers 1 through 32 Holding Registers up to 1792 depending on program and memory size Bits 1 through 16**
	PC-700 Software *2.X	256	Input Registers 1 through 8 Output Registers 1 through 8 Holding Registers up to 256 depending on program size Bits 1 through 16**
		512	Input Registers 1 through 16 Output Registers 1 through 16 Holding Registers up to 512 depending on program size Bits 1 through 16**
		1024	Input Registers 1 through 24 Output Registers 1 through 24 Holding Registers up to 1024 depending on program size Bits 1 through 16**
		2048 through 4096	Input Registers 1 through 32 Output Registers 1 through 32 Holding Registers up to 1792 depending on program size Bits 1 through 16**
	*PC-900A	All	Input Registers 1 through 8 Output Registers 1 through 8 Holding Registers up to 1792 depending on program and memory size Bits 1 through 16**
	PC-900B	All	Input Registers 1 through 16 Output Registers 1 through 16 Holding Registers up to 1792 depending on program and memory size Bits 1 through 16**

*No longer available.

**It is possible to pick bits beyond No. 16. 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.

3-11. HISTORY BITS

The Executive Random Access Memory (RAM) contains one bit for each output or internal coil that can be set, cleared, or tested by the Executive program. These bits are called History Bits because the Executive program usually uses them to determine what actions have taken place during previous scans. There are five general types of uses:

1. Some special functions do not use History Bits because they execute every scan and are dependent only on the associated contact matrix. The following functions do not set or clear the History Bit associated with the reference number:

   BO, CR, EQ, GE, LC, MR, RP, RW, SK, SP, UI, and US

2. The most common use of the History Bit associated with the coil reference number is to cause the associated function to operate only when an OFF to ON transition is seen on the input to a special function block. In this case, when the contact matrix solves TRUE, the History Bit associated with the coil reference number is tested. If the History Bit is OFF, the special function is performed and the History Bit is set. If the History Bit is ON, the special function is not executed. Any scan that the contact matrix solves FALSE causes the History Bit to be cleared.

   The special functions that use the History Bit in this way are:

   AD, AM, BD, BT, CM, CT, DB, DV, FO, LO, MP, OM, OT, SB, SQ, and XM

3. Timers use the History Bit associated with the coil reference number in the reverse manner and will accumulate time only if the History Bit is set. This guarantees that the timer will give at least one scan delay after the time line is energized. For this reason, clearing the History Bit associated with a timer will prevent it from timing that scan. The History Bit follows the top or time input for the TS and TT functions.

4. Some special functions use the top entry line in the special function block to cause a transitional operation of the function. The History Bit associated with the coil reference number is set when the top line is conducting, and cleared when the top line is not conducting with the following functions:

   DC, FI, NL, NR, RT, SL, SR, and UC

5. The remaining special functions use the History Bit associated with the coil reference number as described below:

AS, AT	These are multi-scan functions. The History Bit is set when the enable line is ON and the operation is complete. The History Bit is cleared when the enable line is not conducting.
CG	This function does not use the History Bit associated with its coil reference number; however, it does clear the History Bit(s) associated with the operand of the function.
CS	This function is in the PC-700 only and clears the History Bit associated with the coil reference number when enabled. The coil status is not changed with this function.
DR, TR	The History Bit associated with the coil reference number for the TR function follows the step (top) line if the enable (bottom) line is conducting. If the enable line is not conducting, the History Bit is set. This means that to step the pointer register by using the step line, the function must see the enable or move data line conducting with the enable line conducting on the next scan.

If the same contact matrix is used for both the enable and step line, the pointer will not step.

Note

The second or middle line of this function resets the printer to zero, does not affect the History Bit.

MV The History Bit is not used in the PC-900. In the PC-700, the History Bit follows the enable line but does not affect the operation of the function.

SM, TL, TO These functions set the History Bit when both the step and reset lines are conducting. The History Bit is cleared when either the step or reset line is not conducting.

Notes

1. The PC-700 does not support the RP, RW, and SP functions.

2. The PC-900 does not support the AS, CS, OT, TL, TO, SQ, and US functions.