SECTION 7
PROGRAMMING THE MICRO 84

The basic element used in programming the MICRO 84 Programmable Controller is the contact. A contact can be either normally open or normally closed. Associated with each contact is a logic element (indicating whether the contact is normally open or normally closed) and a reference number. The reference number ties the logic element to a specific connection on the I/O input module. The input used to control the status of a contact is called a discrete input (i.e., it turns the contact either to the non-normal condition or returns it to the normal condition).

Another type of input that can be used is numeric data. This information is stored in an input register. Each register also has an associated reference number.

A coil is another logic element and its state (energized or de-energized) is determined by the various inputs that control it. A coil is an output used to control a specific piece of user equipment or as input to another network. Each coil is associated with a specific reference number.

7.1 NETWORKS

In the MICRO 84, each program element occupies one or two "nodes". The nodes are arranged so that seven elements can be entered horizontally (a rung on the ladder). Up to four rungs can be combined to form a network. A network then, is the ladder diagram program. The number of logic elements that can be contained in a MICRO 84 depends upon the complexity of the networks and the memory size of the controller. Figure 7-1 illustrates a typical multi-node network.

7.2 SOLVING NETWORK LOGIC

The MICRO 84 function is to "solve" the ladder logic networks. To do this, it "scans" the various elements of a network, checking the state of each input element (are contacts open or closed, etc.) and setting the status of the output coils accordingly. In performing its scan of a network, the MICRO 84 checks the status of the leftmost COLUMN of elements first and then proceeds COLUMN by COLUMN to the right until the network is "solved". An element in Row 3, Column 4 is scanned (and solved) before an element in Row 1, Column 6. The output of the first element can be used as input to the second element. Figure 7-2 illustrates the sequence the controller uses in solving a network.

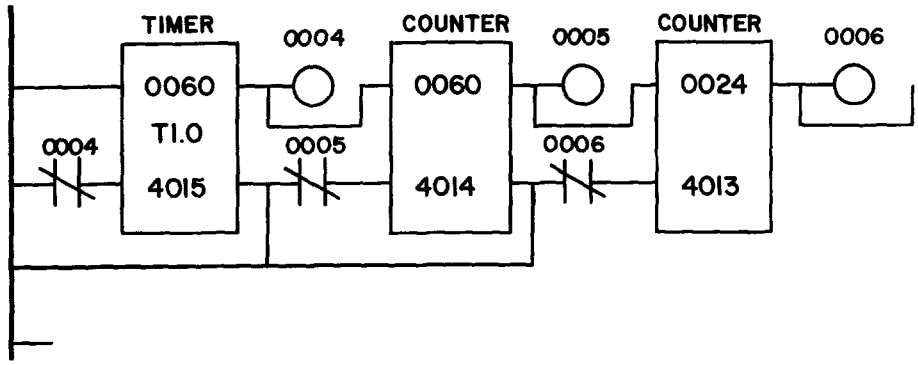


Figure 7-1. Typical Multi-Node Network

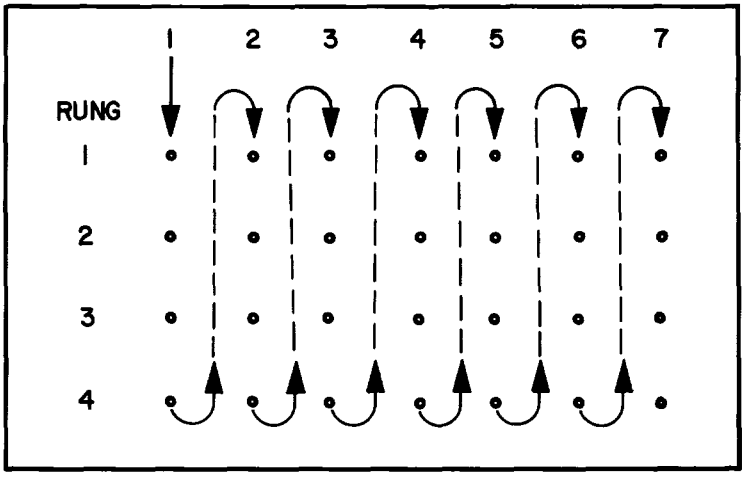


Figure 7-2. Network Solving Sequence

Because the MICRO 84 solves a network in a particular fixed sequence, it is possible to arrange the logic elements in such a way that the status of a coil can be used as discrete input to a contact in a column further to the right than the coil (see Figure 7-3). The status of a coil also can be used as discrete input to another network.

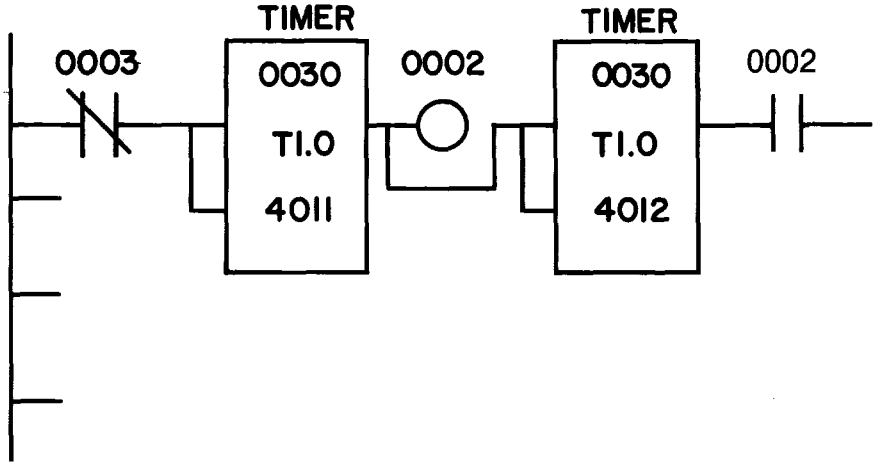


Figure 7-3. Coil Used as Discrete Input

7.3 NETWORK POWER FLOW

Within a network, power flow is from left to right, bottom to top, or top to bottom. When displayed on the LCD programming panel, power flow is indicated for all contacts by illuminating those contacts that are passing power from left to right.

7.4 LADDER LOGIC PROGRAMMING PAD

A Ladder Logic Programming Pad has been designed and is available for use in planning and documenting ladder logic programs. Figure 7-4 shows the format of the pad.

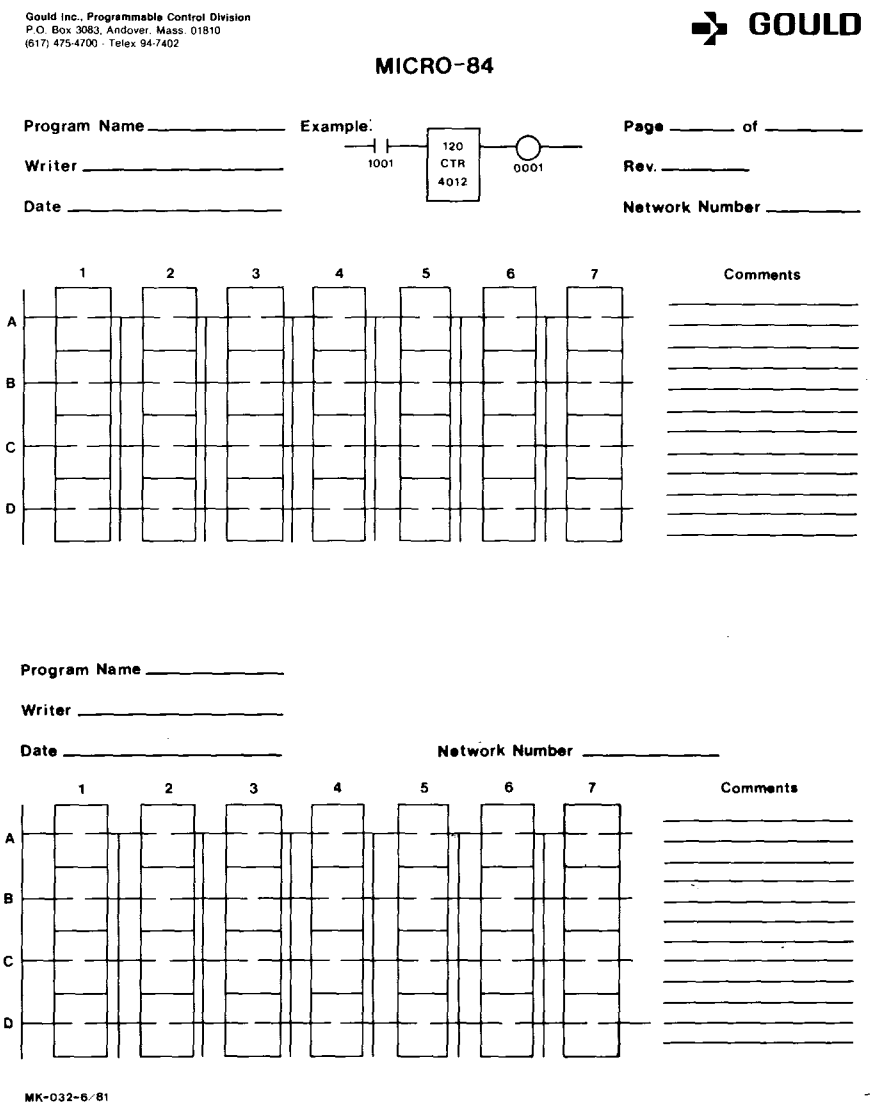


Figure 7-4. Ladder Logic Programming Pad

7.5 REFERENCE NUMBERS

In the programming of the MICRO 84 Programmable Controller, four digit reference numbers are used in building the program logic. These reference numbers are divided into three broad categories: discrete, register, and sequencer. Discrete references are used for items that can be either ON or OFF, such as contacts, pushbuttons, motor starters, relay coils, limit switches, solenoid valves, etc. Register references are used to store numeric values such as counters, timers, analog values, etc.; all registers can store numbers that are three decimal numbers long (0-999). These numbers are stored in binary-coded decimal (BCD) format. Sequencers operate like stepping drum rotary switches with a single reference contact at each position of the stepping switch. Sequencer operations are controlled by numeric values placed in specific registers (see "Sequencers" (Drum Programmers) later in this section).

Any specific reference number can be used as many times as required throughout the logic except for a coil. A coil can be used as an output only once but can be used as many times as desired as an internal contact. Reference numbers are defined as follows:

Type	Reference Number M84-001	Reference Number M84-002
Coils and discrete outputs (lights, solenoids) or internal contact	0001-0032	0001-0064
Input registers (numeric data)	3001-3004	3001-3004
Sequencers	21XX-28XX XX-Step (01-16)	Same
Usable as sequencer registers	4001-4008	4001-4008
Discrete inputs (contacts, pushbuttons)	1001-1032	1001-1064
Internal registers (numeric data)	4001-4020	4001-4032
Output registers	4010, 4012, 4014, 4016	4010, 4012, 4014, 4016, 4018, 4020, 4022, 4024

7.6 PROGRAMMING FUNCTIONS

The MICRO 84 Programmable Controller provides the capability to program or simulate the operation of relay contacts, timers, sequencers, counters, and arithmetic functions. All programming is done on the basic format of up to seven elements in each horizontal row or rung. A network can be a single rung, two rungs, or up to four rungs. Relay type elements occupy a single node position within a network and non-relay type elements (counters, timers and arithmetic operations) occupy two vertically oriented node positions within a network.

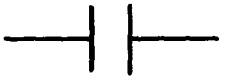
Data is entered directly into the memory of the controller (it is not stored in the programming panel). If AC line power is interrupted prior to completion of the programming, whatever data has been entered is retained. No additional operations are required; whatever data the user enters is the data stored for use by the controller.

Network elements can be added, deleted, or changed at any time using the programming panel. The logic changes do not interrupt the controller's operation. In addition, the effect of any coil (input or output) on system operation can be tested by simulating inputs and outputs using the MICRO 84's disable feature.

7.6.1 Basic Programming Elements

The basic programming elements are the contacts, coils, and connectors shown below:

7.6.1.1 Normally Open Contact



The normally open contact is the basic logic element point used in ladder logic diagrams. The logic element is controlled by a discrete input (reference number 10XX) or by a coil (reference number 00XX).

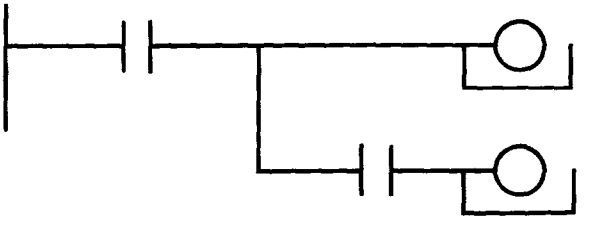
7.6.1.2 Normally Closed Contact

The normally closed contact is controlled the same way as a normally open contact.

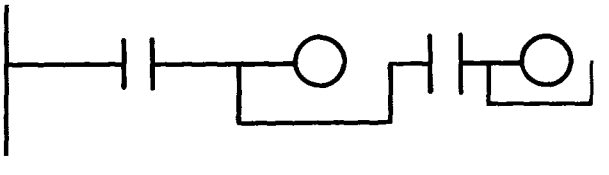
7.6.1.3 Coil

The coil (reference number 00XX) is the discrete output from ladder logic. A coil is used either to provide the control signal for an external discrete device (solenoid, motor starter, indicator lamp, etc.) and/or as the input for an internal contact. A coil can be programmed only once as a discrete output. Once programmed, it can be referred to as many times as desired as an internal contact.

More than one coil (unlatched or latched) can appear on a single rung. Each coil on a rung is treated as if it were connected to the right rail of the network. Therefore,



is programmed in the MICRO 84 as:



7.6.1.4 Latched Coil

The latched coil retains its state when power is removed (or fails). Normally, coils are reset to OFF upon power-up. A latched coil will not be altered and thus will retain the ON or OFF condition it had when power was removed.

7.6.1.5 Horizontal Connector

The horizontal connection is a short between two nodes on a ladder logic rung. The shifted horizontal connector is a horizontal open.

7.6.1.6 Vertical Connector

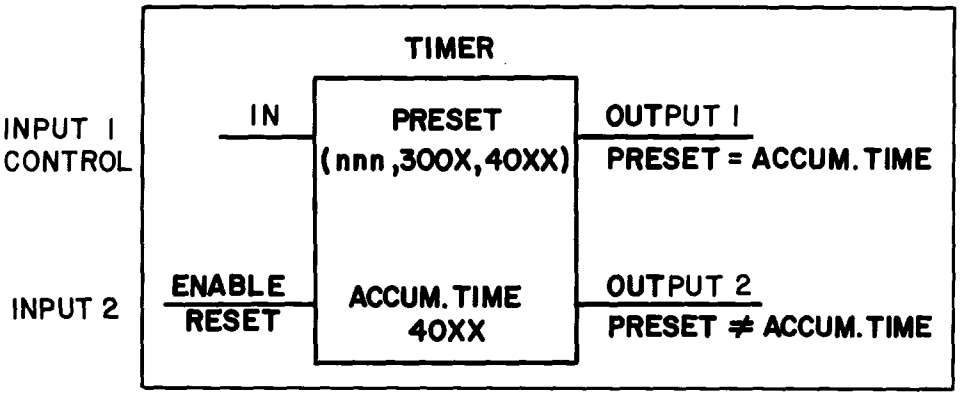
The vertical connector is a short between two rungs in a ladder logic network. The shifted vertical connector is a vertical open.

7.6.2 Timers

Two timer rates are provided in the MICRO 84: one second and one tenth of a second. These internal timers can be used to create as many logic program timers as are necessary.

The timer content is not lost due to a power failure.

The one second timer (T1.0) is used to measure times between 1 second and 999 seconds. The one tenth of a second timer (T0.1) is used to measure times between 0.1 second and 99.9 seconds. The format of the timer element is shown below:



where:
Input 1
The timer is activated when power is connected to Input 1. A normally open contact is usually in this input line.

Input 2
The timer is enabled (able to be activated) when power is connected to Input 2 and is reset when power is removed. A normally closed contact is usually in this input line.

Preset
The upper element of the timer (preset) is used for setting the time period in seconds, or in tenths of seconds, that is to elapse before Output 1 is activated. The preset can be a fixed value (up to 999) or a register. When the preset is a register (reference number 300X or 40XX), the content of the register (a value up to 999) is used as the preset value. This register may be used by all timers having the same preset value.

Accumulated Time
The lower element of the timer must be a register (reference number 40XX) in which the actual (accumulated) time is stored. This "holding" register must be unique for this timer.

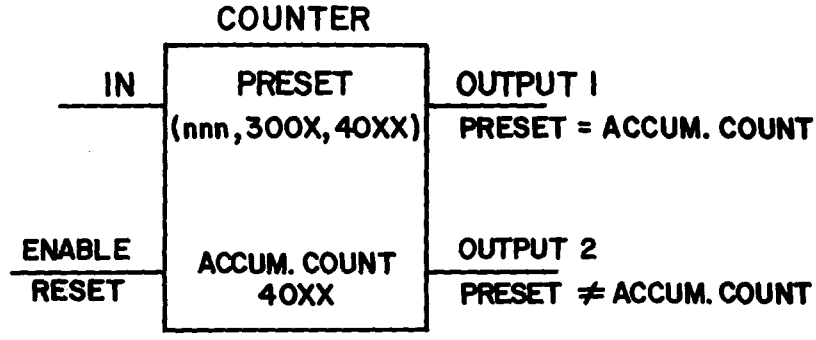
Output 1
Output 1 is a normally open circuit that becomes a closed circuit when the accumulated time equals the preset time. Remains closed until the timer is reset or until the equality no longer exists.

Output 2
Output 2 is a normally closed circuit that becomes an open circuit when the accumulated time equals the preset time. Remains open until the timer is reset or until the equality no longer exists.

7.6.3 Counters

The MICRO 84 contains circuits that simulate event counters. The counter increases its value by one whenever the input signal goes from OFF (de-energized) to ON (energized). Only on this positive transition (OFF to ON) is the count increased. Even if the input signal remains ON for several hours, it represents only one transition from OFF to ON and is thus only one count.

The format of the counter element is shown below:



The counter functions the same as the timer except that the preset value and the value being accumulated in the lower register are in terms of the number of times an event has occurred rather than the amount of time that has elapsed.

7.6.4 Cascading Timers and Counters

Timers and counters can be interconnected or cascaded to satisfy any required logic. As many timers and counters as necessary can be placed within a network limited only by the 4 x 7 logic element format.

Two timers (or counters) can be placed in series (interconnected) by using output 1 of one timer as input 1 to a second timer. These two timers allow twice as much time to be measured; they are cumulative in effect.

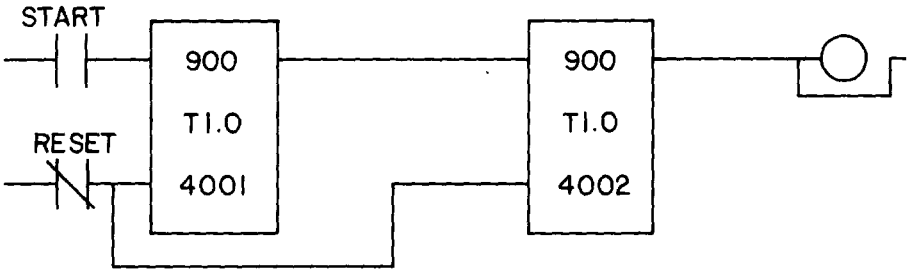
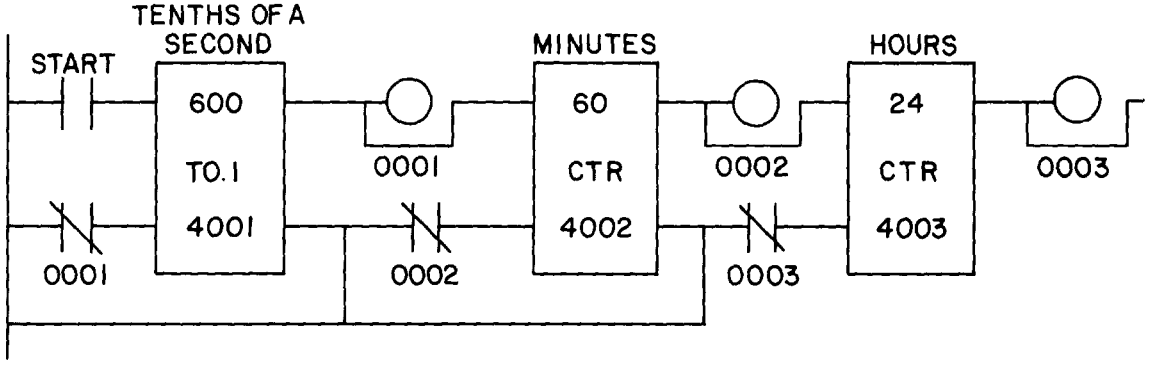


Figure 7-5. Interconnecting Timers

Timers and counters can also be cascaded to, in effect, multiply their range. Figure 7-6 illustrates a timer/counter network that produces a time-of-day clock measuring time in hours and minutes.



NOTE

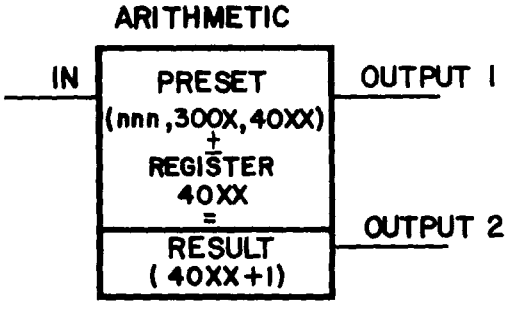
For greater long-term accuracy, adjust timer between 598 and 602 using an accurate clock.

Figure 7-6. Time-of-Day Clock

7.6.5 Arithmetic Operations

The arithmetic operations consist of addition and subtraction.

The format of the arithmetic operations is shown below:



where:

Input

The arithmetic operation is performed when power is connected to the input. The input circuit is normally open.

Preset

The upper element of an arithmetic operation is a preset value. The preset can be a fixed value (up to 999) or a register. When the preset is a register (reference number 300X or 40XX), the content of the register (a value up to 999) is used as the preset value.

Holding Register

A register that holds the result of another logic operation (reference number 40XX).

Result Register

A register in which the answer to the arithmetic operation is stored. The MICRO 84 automatically assigns a reference number to this register. The number assigned is the holding register reference number plus one (e.g., 40XX + 1). When assigning a reference number to a holding register, make sure that there is an unassigned reference number, one larger than the one being assigned, for use by the controller.

Output 1

The status of the output is determined by the result of the arithmetic process.

Output 2

The status of the output is determined by the result of the arithmetic process.

NOTE

An output is energized only if the input is also energized. When power is removed from the input, the output is de-energized.

7.6.5.1 Addition

The addition function adds the preset value to the contents of the holding register and stores the result in the result register. Based on the results of the addition, one of two output conditions exists:

- If the sum exceeds 999, Output 1 is energized (Output 2 is de-energized) and the remainder (the sum minus 1000) is stored in the result register.
- If the sum is less than 999, Output 2 is energized (Output 1 de-energized).

NOTE

The result register contains only the result of the latest calculation.

7.6.5.2 Subtraction

The subtraction function compares the value in the preset with the value in the holding register and places the value of the difference in the result register. Based on the results of the compare, one of three output conditions exists:

- If the preset is greater than the holding register, Output 1 is energized (Output 2 de-energized).
- If the preset is less than the holding register, Output 2 is energized (Output 1 de-energized).
- If the preset is equal to the holding register, both Output 1 and Output 2 are energized.

7.6.6 Sequencers (Drum Programmers)

The MICRO 84 Programmable Controller can be used to produce a drum type programmer on a circumferential row-by-row (sequencer) basis. Each circumferential row can operate independently or dependently with reference to the surrounding rows. The controller can contain up to eight sequencers with up to sixteen contacts for each. Reference numbers for sequencer contacts start with the digit 2 and are in the form 2RXX. The significance of the remaining three digits of a sequencer contact reference number is as follows:

Sequencers are controlled by the value (00-16) that is contained in a sequence register (reference numbers 4001-4008).

Sequencer Register Reference Number: 400Y

Sequencer Contact Reference Number: 2RXX

where:

R = 1 thru 8 (Sequencer reference register number 4001-4008)

XX = 01 thru 16 (Sequencer contact number)

Sequencer operation is controlled by a numeric value (1 thru 16) placed in a specific sequencer register (4001-4008) by any of the non-relay devices such as a counter, timer, or arithmetic operation. The value placed in the sequencer reference register results in a single sequencer contact (2RXX) being energized (contact closes). All other contacts are de-energized. As an example, refer to Figure 7-7.

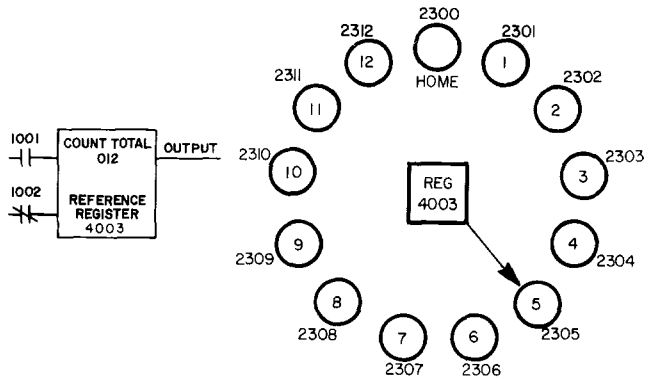


Figure 7-7. Sequencer Example

Each time input "count" contact (1001) is closed, the value in register 4003 increases by one. This is equivalent to moving the stepping switch one position (clockwise). If register 4003 contains the value five, sequencer contact 2305 is energized. When input 1002 is energized, the counter is reset to zero and the stepping switch goes to home position (no contacts closed).

Each sequencer is completely independent since each is controlled by a separate register (4001-4008). Each sequencer contact reference number (2RXX) can be used one or more times anywhere in the logic where a contact is appropriate.

By using various logic techniques, any sequencer can be made to skip steps and to jump forward or backward as necessary. Only one sequencer contact (per sequencer) is in effect at any one time (i.e., contact points between the jump points are not affected by the jump). The keystrokes required to program a typical sequence are given in Appendix B.

SECTION 8
USING THE P371 PROGRAM PACK

The MICRO 84 P371 Program Pack (see Figure 8-1) incorporates a memory that is identical to the user's portion of the controller memory. It is this portion of the controller memory that contains the user-generated logic program. The program pack allows the user to dump (store) this logic program for use as a backup to the current system. The program pack also allows a logic program generated on one controller to be transferred (loaded) into another controller with a minimum of effort.

NOTES

Only the user-generated logic program is dumped to the program pack. The contents of all holding registers, and the redefinition of the A and B keys are not transferred.

Be sure the Program Pack used matches the controller (i.e., only use the 1K P371-001 with M84A-001 and the 2K P371-002 with M84A-002).

Before transferring the user's logic program, the MICRO 84 automatically erases the memory that is to receive the program (e.g., when dumping a logic program to the program pack, the first step that the controller performs is to erase the memory in the program pack).

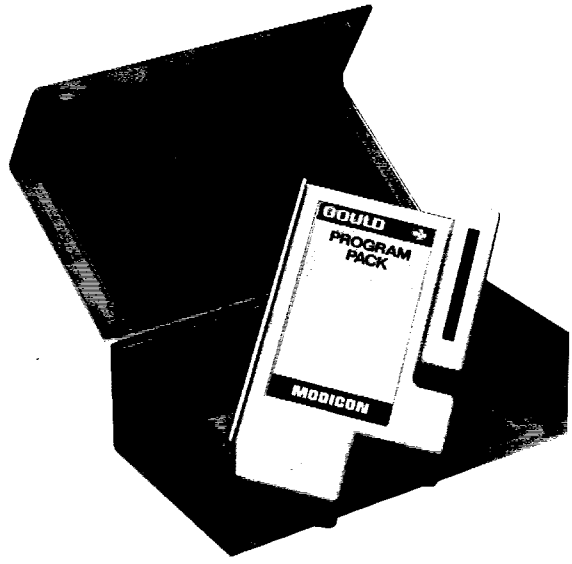


Figure 8-1. Program Pack

8.1 P371 PROGRAM PACK INSERTION

The program pack fits into a slot in the bottom of the MICRO 84 Controller. Insert the male connector on the program pack into the slot on the controller. Push on the base of the program pack so that it mates firmly with the controller. When not actively using the Program Pack, remove it from the controller and store it in a safe place. Do not press the orange button while inserting the pack into the mainframe.

NOTE

When recording and then reloading a program into the same controller, remove the P371 Program Pack and then reinsert it before loading the program. This procedure prevents the occurence of false error conditions.

8.2 DUMPING (STORING) A PROGRAM TO THE P371 PROGRAM PACK

Dumping a user's logic program from the MICRO 84 controller to the program pack is initiated by a sequence of keystrokes entered into the P370 programmer. The necessary steps and keystrokes to dump a program are given below.

1. Enter the Supervisory mode and stop the controller (controller RUN light goes out).

[SUPV] [1] [ENTER] [ENTER]

2. Dump the user's logic program into the program pack. During this process, the WRITE light on the program pack illuminates.

[5] [ENTER] [ENTER]

Wait until the WRITE light goes out before proceeding.

3. Restart the controller. The controller's RUN light illuminates.

[2] [ENTER] [ENTER]

4. Return the P370 Programming Panel to the EXAMINE mode.

[RESET]

8.3 LOADING A PROGRAM FROM THE P371 PROGRAM PACK

The button located on the program pack's bottom activates the transfer of program data from the program pack to the controller.

1. Pressing the button once causes the following action:

   a. The controller stops, turning off the RUN light on the controller.

   b. Initiates the program data transfer. After the data is transferred, the DONE light, located on the program pack, turns on.

      If the DONE light does not turn on, it indicates that the load was not successful. Retry loading.

2. Pressing the button the second time causes the following action:

   a. The DONE (green) light turns off.

   b. The controller's RUN light turns on, indicating that the controller has restarted.

3. If the P370 Programmer was attached during the loading sequencer, press SHIFT/RESET. This is necessary so that the programmer can reflect the new controller logic.