SECTION 1
INTRODUCTION TO RELAY LOGIC

The 584L Programmable Controller is programmed using basic relay logic programming language. The logic elements used include relay contacts, coils, references, registers, and function blocks. The controller's logic is structured into networks and segments, and programmed into the 584L via a P190 Programmer. The controller scans each network and solves the logic which can control the input to other logic in the program, or control an output (i.e., turning ON a switch, stopping a process, resetting a meter, etc.).

1.1 NETWORKS

A network is a set of interconnected logic elements which represents all or part of the user's 584L program. Each network has a maximum width of 10 nodes and a maximum length of 7 nodes. An eleventh column is provided exclusively for coils. See Figure 1-1.

A network can contain any combination of relay contacts, coils (eleventh column only), counters, timers, and arithmetic, data transfer (DX), and special function blocks. The logic can occupy the whole network area or just a portion of it.

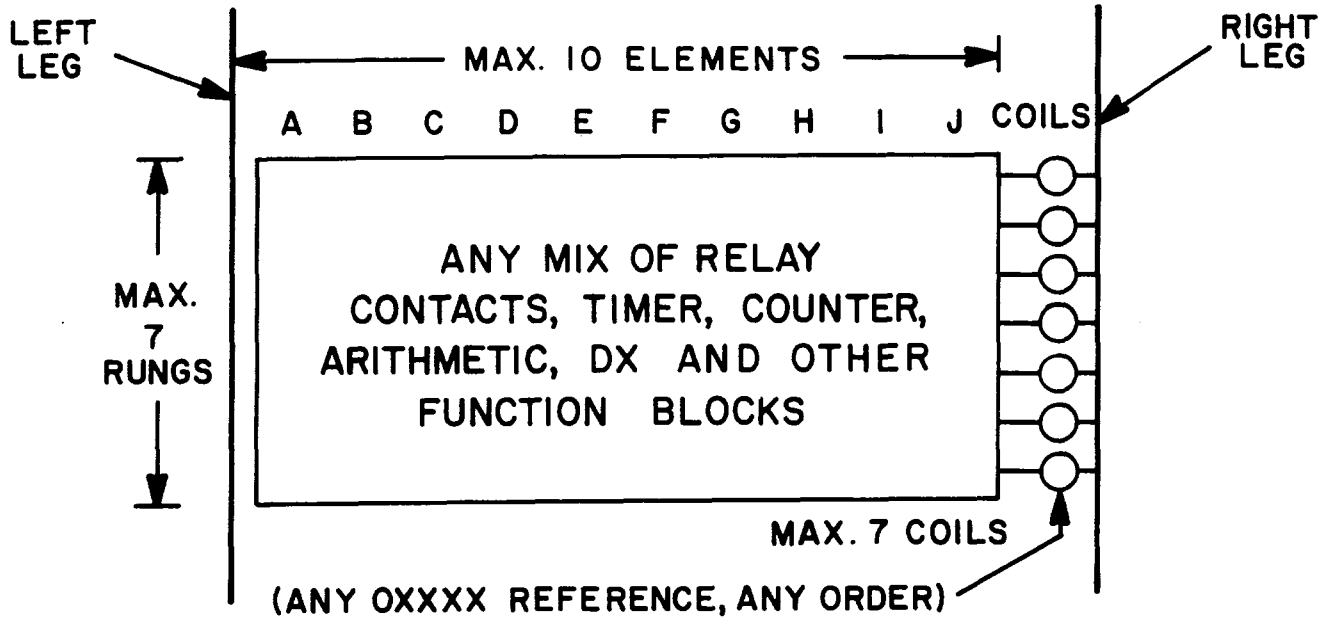

Figure 1-1. Network Parameters

1.2 SEGMENTS

Networks are separated into groups called segments. Each segment controls two I/O channels — 256 inputs and 256 outputs; 128 inputs and 128 outputs for each channel. The number of segments available is equal to the number of I/O channels divided by two. See Appendix A to determine the number of I/O channels needed for the 584L program and thereby the number of segments available.

The logic contained in each segment should control the I/O Channels it represents. For example, if logic in segment 3 controls an output in Channel 1, the output is not activated until the controller starts its next scan and scans segment 1. If the logic for that output had been placed in segment 1, the output would have been activated on the same scan.

1.3 REFERENCES

Reference numbers are used to identify relay contacts, coils, inputs, outputs, and registers. There are four types of references; each one has a different code digit to identify which type it is. This digit is the first of five consecutive digits. The reference types and their functions are listed in Table 1-1.

Table 1-1. References

0XXXX — Coil/Discrete Output

- A discrete (ON/OFF) signal that is controlled by logic.
- Can be used to drive a real output through an output module.
- Can be used internally to drive one or more contacts in user logic.

1XXXX — Discrete Input

- Status of the input is controlled by an input module.
- Used to drive contacts in user logic.
- Can be used repeatedly in the program.

30XXX — Input Register

- A numerical input from an external source (i.e., thumbwheel, analog signal, or high speed counter).
- Sixteen consecutive discrete signals.
- Can be binary or binary coded decimal (BCD).

4XXXX — Holding/Output Register

- Used to store numerical information, decimal or binary, in the controller.
- Can output numerical information to an output module.

NOTE

These numbers refer to actual registers within the controller which contain numerical values or ON/OFF conditions. An X is any digit, 0 through 9; however, it may have a specific limit (e.g., max. 5) as designated in the configuration table.

1.4 RELAY CONTACTS

The relay contact is the basic programming element. It can be referenced to either a logic coil (0XXXX) or a discrete input (1XXXX). The contact is opened, no power passing through, or closed, power passing through, when a certain condition exists (i.e., logic coil is energized or de-energized; input signal is ON or OFF).

Relay contacts can be normally open, normally closed, or transitional.

1.4.1 Normal Contacts

The two most commonly used contacts are:

Normally Open (NO) Contact 

Normally Closed (NC) Contact 

When the coil or discrete input to which a contact is referenced is ON, the normally open (NO) contact is closed and passes power, and the normally closed (NC) contact is open and does not pass power. When the coil or discrete input is OFF, the NO contact is open and does not pass power, and the NC contact is closed and passes power. See Table 1-2.

Table 1-2. Normally Open and Normally Closed Contacts

	NO Contact	NC Contact
Coil or Discrete Input is ON	passes power	does not pass power
Coil or Discrete Input is OFF	does not pass power	passes power

1.4.2 Transitional Contacts

A transitional contact is turned ON by the transition, OFF to ON or ON to OFF, of the coil or discrete input to which it is referenced. It is not affected by the ON or OFF state of the logic coil or discrete input after the transition.

Sometimes it is necessary for a function to be performed only once. In this case a transitional contact is used because it only transitions once each scan.

Since there are two different transitions (OFF to ON and ON to OFF), there are two transitional contacts:

Transitional Contact (OFF to ON) 

Transitional Contact (ON to OFF) 

1.4.3 Inserting Contacts

A relay contact is inserted into a program by positioning the cursor over the desired location, entering a 0XXXX or 1XXXX reference into the Assembly Register (AR) and pressing the appropriate software label key. To change the type of relay

contact, position the cursor over the contact to be changed and press the desired software label key. To change the reference numbers below a contact, position the cursor over the contact, enter a new value into the AR, and press the ENTER key.

1.5 VERTICAL AND HORIZONTAL SHORTS

Vertical and horizontal shorts are simply straight line connections between contacts.

Vertical shorts are used to connect contacts and function blocks one above the other in a network. Vertical shorts can also be used to connect inputs or outputs in a function block to create either/or conditions. When two contacts are connected by vertical shorts, a vertical short on each side, power is allowed to pass through if either, or both, contacts receive power.

Enter a vertical short into a program by positioning the cursor over the node to the left of and above the location desired for the short and press the VERTICAL SHORT software label key. A vertical short is cleared by pressing the VERTICAL OPEN software label key. A vertical short does not take any user memory.

Horizontal shorts are used in combination with vertical shorts to expand logic within a network without breaking the power flow. They can be used to create either/or conditions using basic relay contacts. For example, if one line of logic contains two relay contacts, and the line below it only contains one contact, a horizontal short is placed beside the single contact. See Figure 1-2. A vertical short is used to connect the horizontal short to the top logic line. Power passes through to energize the coil if the two top contacts are energized or if the bottom contact is energized.

Enter a horizontal short into a program by positioning the cursor over the node desired for the short and pressing the " ——— " software label key. To clear a horizontal short, press the DELETE NODE software label key. A horizontal short uses two words of memory.

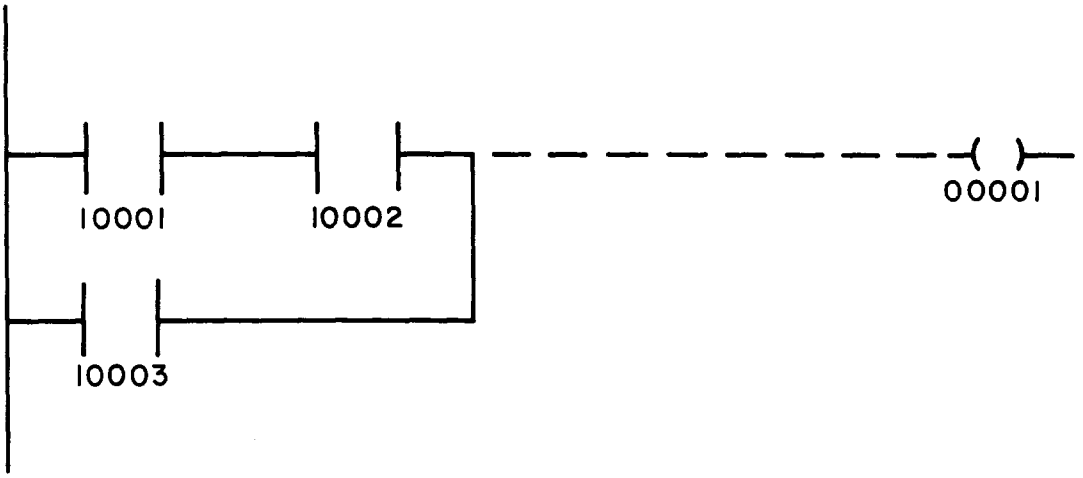

Figure 1-2. Shorts

1.6 COILS

A coil is used to activate logic within the user's program and/or to control an output circuit. It is represented by a 0XXXX reference number and either of two symbols.

A Normal Coil, —( )— , is turned OFF if power is removed and later restored.

A Latched Coil, —(L)— , retains its previous state if power is removed and later restored.

Coils are located in the far right (eleventh) column of a network. Each network can contain a maximum of seven coils.

Each 0XXXX reference can be used as a coil only once, but can be referenced to any number of relay contacts. Output coils are generally given the lower 0XXXX reference numbers and internal coils are given the higher 0XXXX reference numbers. It is not an error to intermix output and internal coil reference numbers since the 584L is able to output any valid coil.

A logic coil is inserted into a program in the same way as a relay contact except that the cursor does not have to be over column eleven. The cursor can be directly beside the last logic element in a row. When the coil software label key is pressed, dashed lines are inserted and the coil is placed in the eleventh column. The only reference allowed is a 0XXXX reference, unlike relay contacts which allow 0XXXX or 1XXXX references.

1.7 DISABLE/ENABLE

Any logic coil or discrete input in the user's program can be disabled and forced ON or OFF. To disable a coil or discrete input, do the following:

1. Ensure that Memory Protect is OFF on both the P190 Programmer and the 584L PC.

2. Position the cursor at the bottom of the P190 screen.

3. Enter the coil reference number or discrete input reference number into the Assembly Register (AR).

4. Press the ERASE/GET key.

The reference number appears at the cursor position with its state, ON or OFF, and the following software labels appear on the screen:

ENABLE	DISABLE	FORCE ON	FORCE OFF

5. Press the DISABLE software label key.

The coil or input is now DISABLED ON or DISABLED OFF, depending on its state when the DISABLE key was pressed.

Press the FORCE ON or FORCE OFF software label keys to change the state of the coil or input. Press the ENABLE software label key to enable the coil or input. If the coil or input is enabled, it cannot be forced ON or forced OFF; it retains the state it had when the ENABLE software label key was pressed.

Disabling a coil or input causes the programmed logic to bypass that coil or input. The coil's state or the input's state is now controlled by the user through the FORCE ON and FORCE OFF software label keys. Memory Protect must be OFF.

WARNING

There are exceptions to the logic bypassing a disabled coil. All the DX function blocks which allow a coil in the destination node override the coil if it is disabled. This can cause personal injury if the user assumes a coil has disabled an operation and repairs are being made, when the coil's state can change as a result of a particular function.

1.8 CONTROLLER SCAN

The 584L PC scans the networks in the user's program to solve the logic. The scan starts at the top left of a network and goes from top to bottom in each column working from the left column towards the right column. See Figure 1-3.

Although coils are placed in the eleventh column and are solved there, if the logic controlling them is in column ten, coils can be solved in other columns as well. If the logic controlling a particular coil is in column 6 and dashed lines connect the logic to the coil, that coil is solved in column 7. For example, in Figure 1-4, coil 00036 is solved before contacts I0012 and I0024, and coil 00033.

The scan starts in segment 1 network 1 and continues in segment 1 network 2. When the scan reaches the end of segment 1, all the logic for segment 1 is solved and the inputs are read for segment 2. The scan continues at the first network of segment 2. Segment 1 outputs are serviced and segment 2 logic is scanned and solved. The inputs are read for segment 3. This is illustrated in Figure 1-5.

The controller continues its scan until all the segments and networks have been scanned. It then returns to segment 1 network 1 and services the last network's outputs, thereby starting the cycle over again.

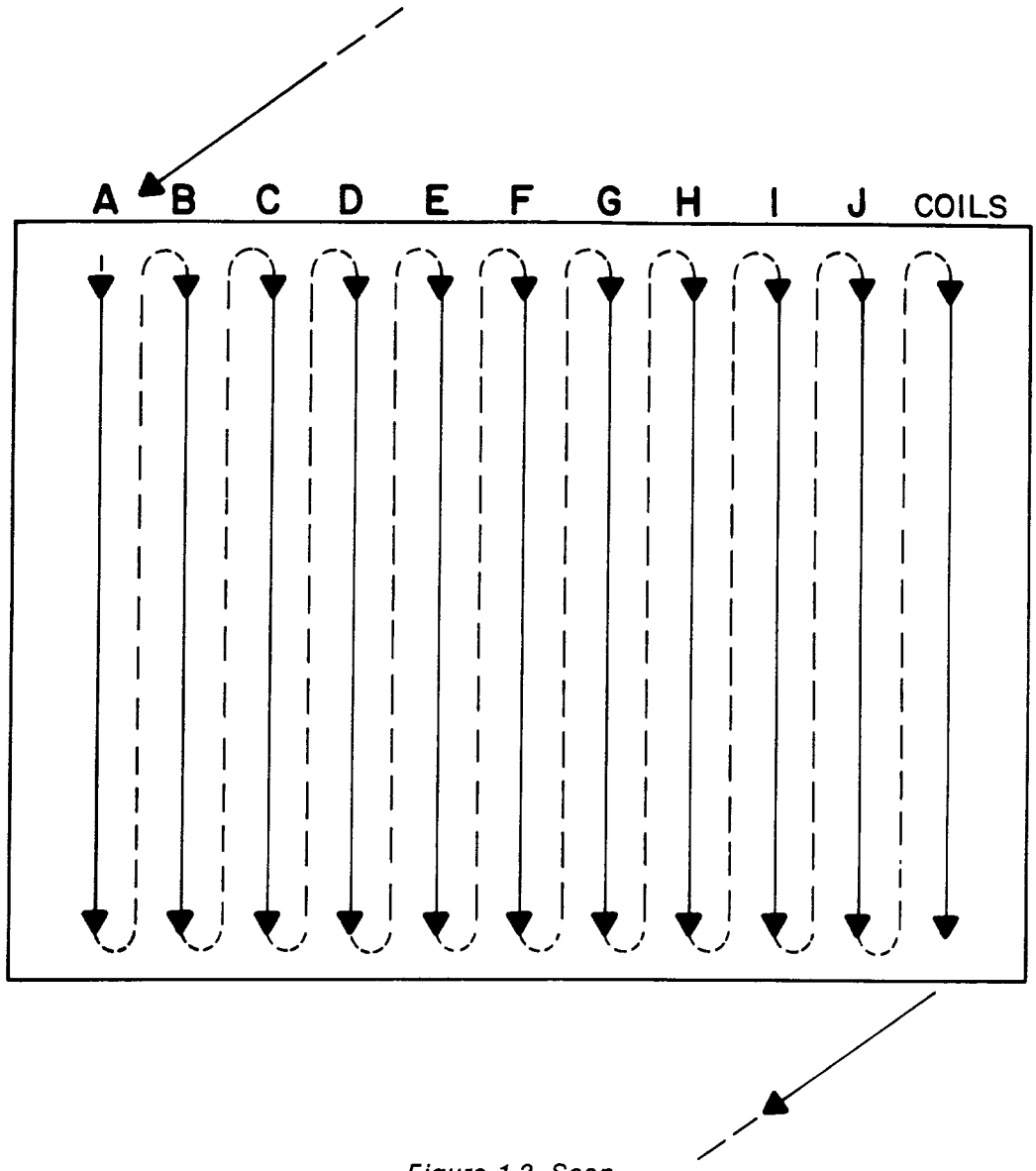

Figure 1-3. Scan

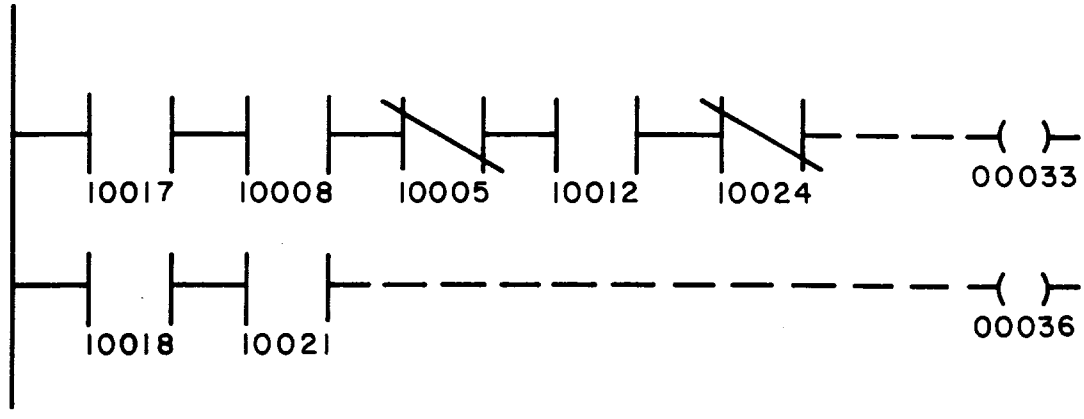

Figure 1-4. Order of Coil Solving

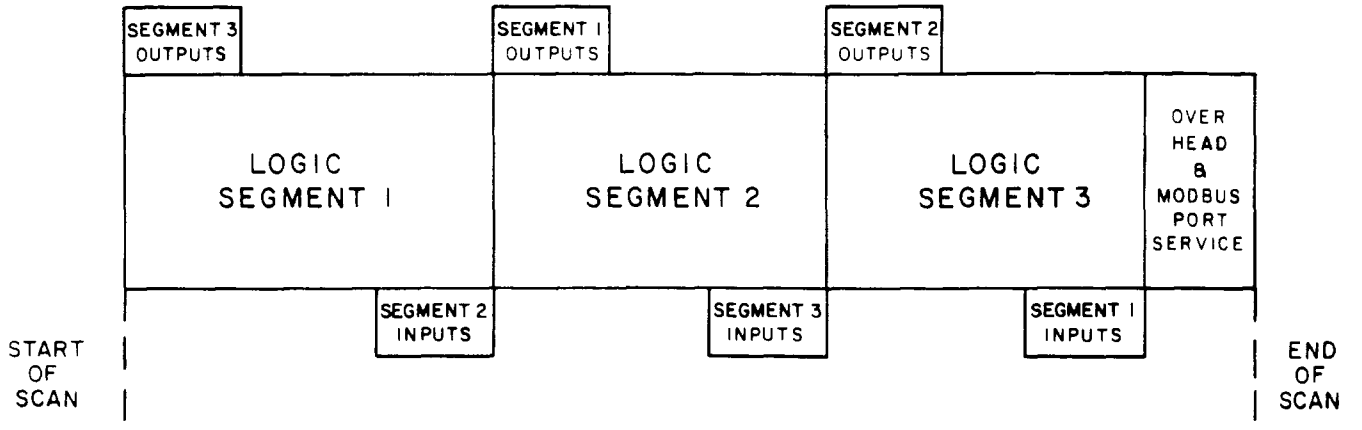

Figure I-5. Sample Logic Scan with I/O Servicing

1.9 PROGRAMMING THE 584L

User programs for the 584L PC are entered using a P190 Programmer. See the P190 CRT Programming User's Manual for installation instructions. The following must be done to enter a program:

1. The 584L PC must be configured to the user's specifications. Appendix A provides the Configuration and Traffic Cop information needed to accomplish this.

2. Insert the 584L Programmer Tape (T584-001) into the PI90 tape drive (front upper right corner).

3. Press the red INIT and INIT LOCK keys on the P190 panel simultaneously.

4. Enter a unit number into the AR. The unit number can be within the range of 1 to 247. It refers to the location of the controller in the data line communicating with the P190.

5. Press the ATTACH software label key. The following PC status software labels appear on the screen:

These software label keys are used to change the 584L PC's status and can be reached at any time while programming by pressing the SHIFT and RESET/EXIT keys simultaneously.

6. Stop the 584L by pressing the STOP 584L software label key.

7. Press the EXIT key followed by the START NEXT key to begin entering a program.

A power rail appears on the left of the P190 screen and the following software labels are displayed:

Each software label key, when pressed, brings up another set of software labels. Figure 1-6 contains all the software label keys needed to enter a program illustrated in a broken down flow chart. To return to the original set of software labels, shown at the top of Figure I-6, from any level of software labels except the PC status level, press the CHG NODE (Change Node) key on the P190 panel.

To start a network, press the START NEXT key. A power rail appears on the left side of the P190 screen and the cursor is located in the top left node. A network can now be entered.

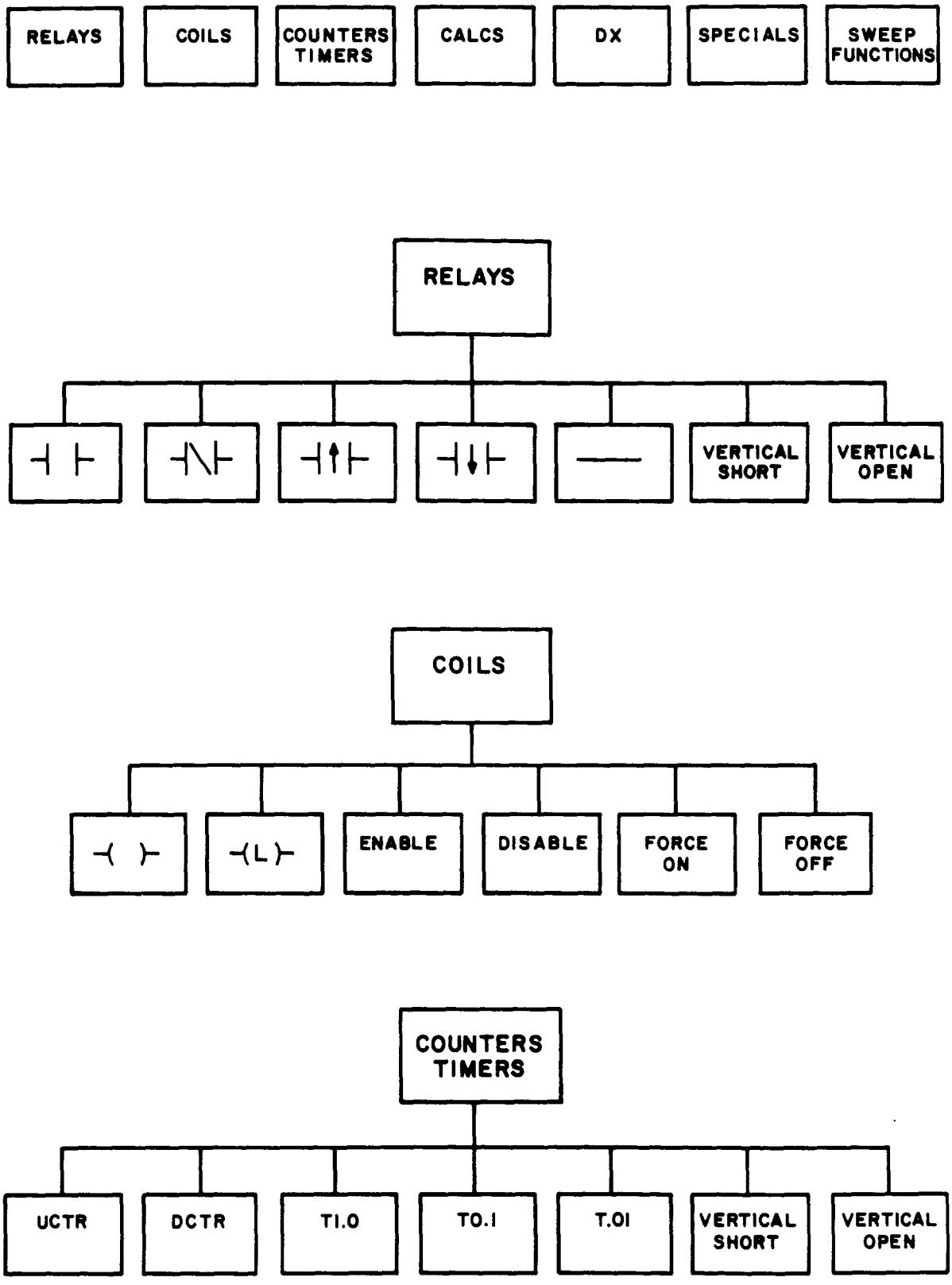

Figure 1-6. Software Label Flow Diagram

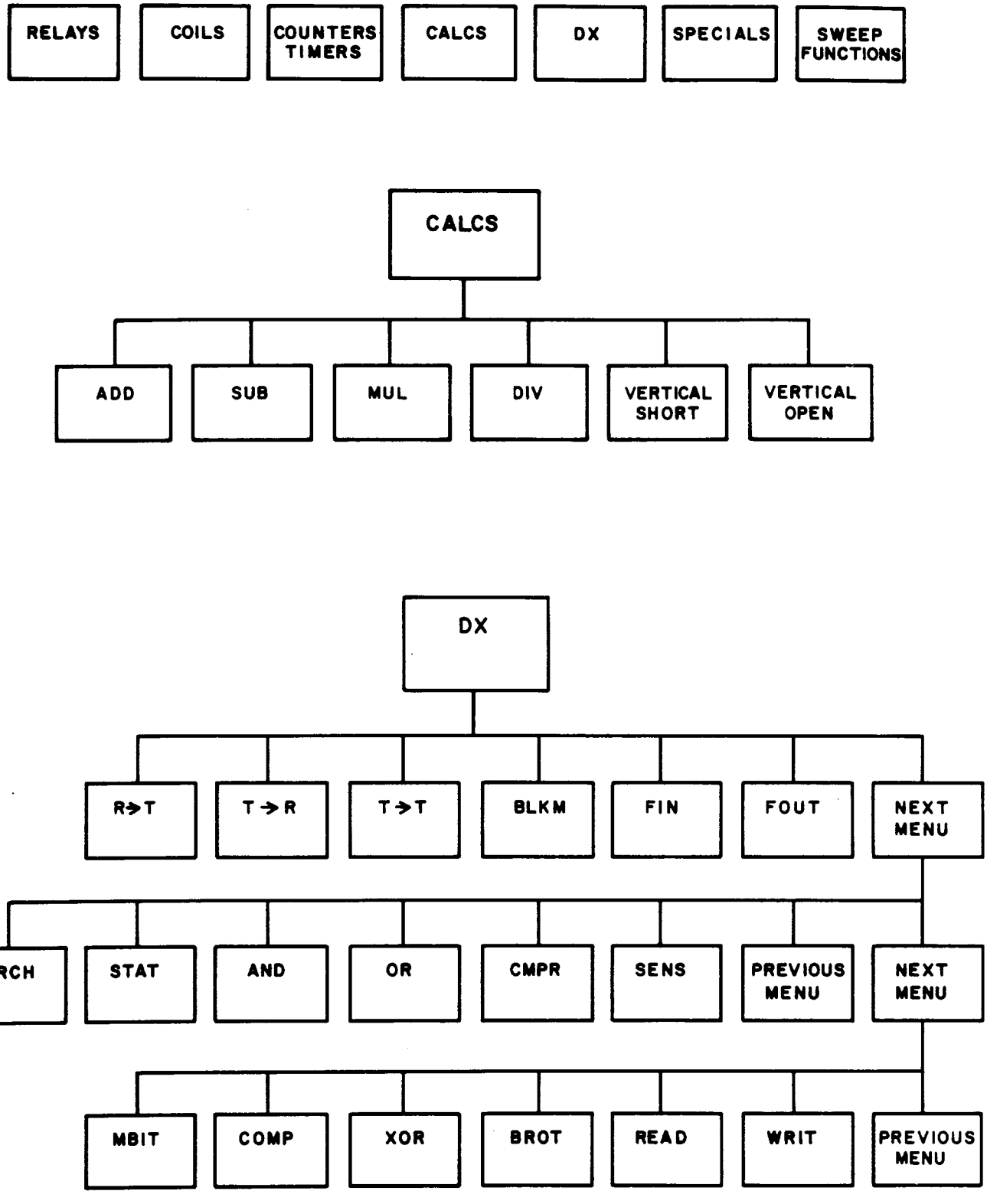

Figure 1-6. Software Label Flow Diagram (Cont.)

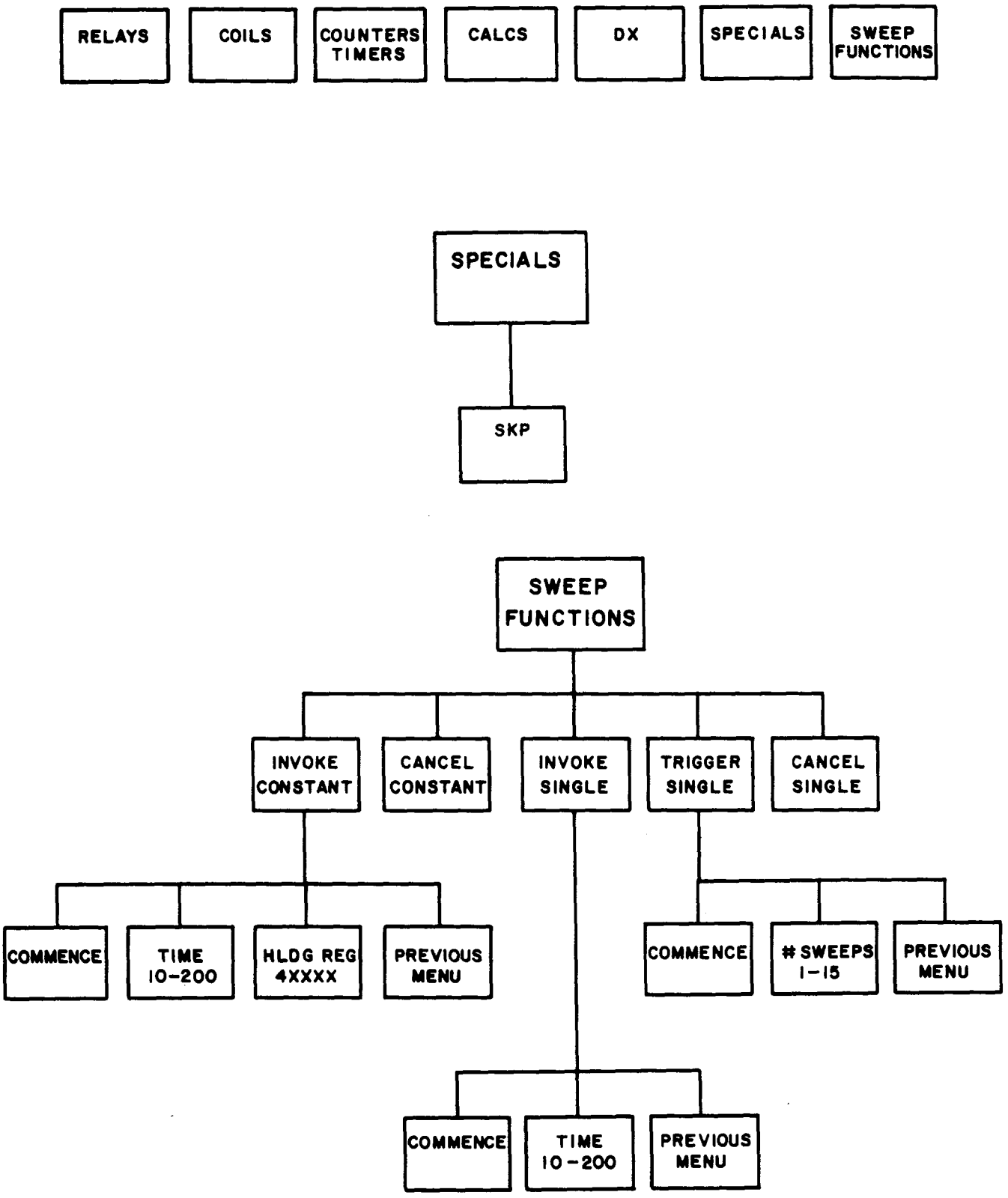

Figure 1-6. Software Label Flow Diagram (Cont.)

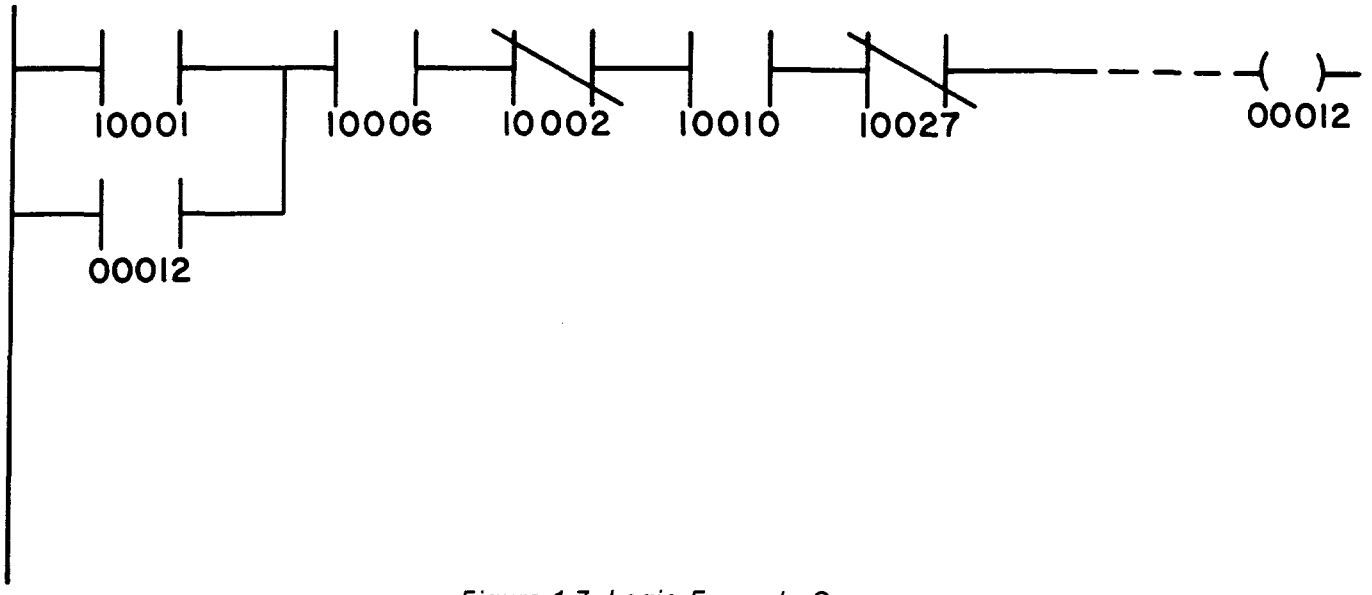

Figure 1-7. Logic Example One

Enter the logic in Figure 1-7, as follows:

1. Press the RELAYS software label key.

   Software labels of the relay contacts appear on the screen.

2. Enter 10001 into the Assembly Register (AR).
3. Press ⊣ ⊢ .
4. Press → .
5. Enter 10006 into the AR.
6. Press ⊣ ⊢ .
7. Press → .
8. Enter 10002 into the AR.
9. Press ⊣/⊢ .
10. Press → .
11. Enter 10010 into the AR.

12. Press —| |— .
13. Press → .
14. Enter 10027 into the AR.
15. Press —|/|— .
16. Press → .
17. Press the CHG NODE (Change Node) key.

    The original set of software labels appears.
18. Press COILS.

    Software labels of coils appear on the screen.
19. Enter 00012 into the AR.
20. Press —( ) .
21. Press → .

    The cursor moves to the first node in the same row (wrap-around).
22. Press VERTICAL SHORT.
23. Press ↓ .

    (Enter 00012 into the AR.)
24. Press —| |— .

    This network is complete. To start a new network press the START NEXT key.

NOTE

To get to a previous network, press the SHIFT and GET PREV/GET NEXT keys simultaneously. To go to the next network, press the GET PREV/GET NEXT key.

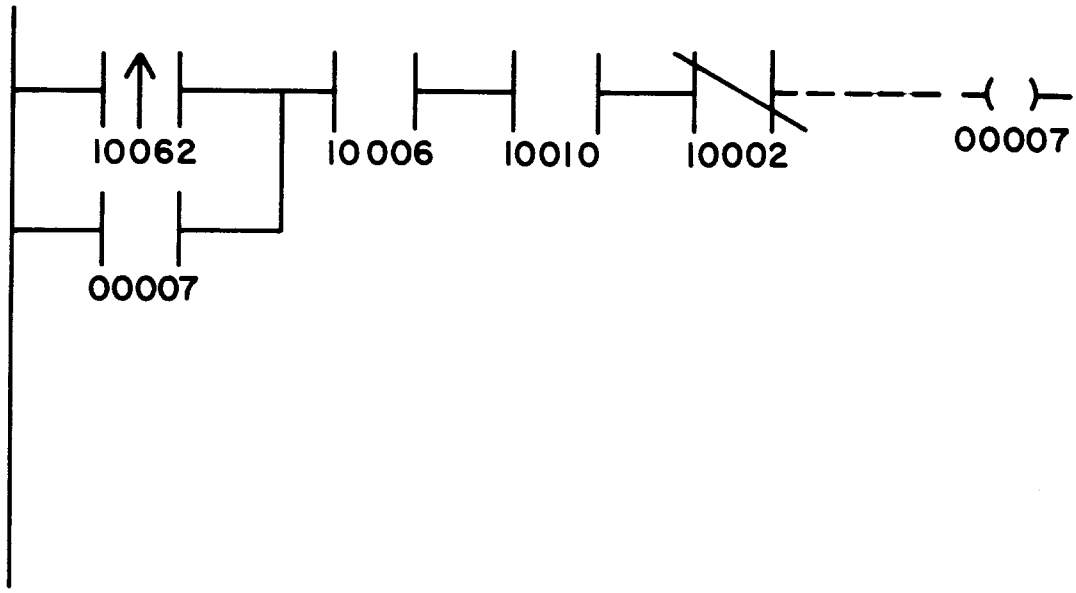

Figure 1-8. Logic Example Two

Enter the logic in Figure 1-8 as follows:

1. Press the RELAYS software label key. (If the RELAYS software label is not displayed, press the CHG NODE key first.)
2. Enter 10062 into the AR.
3. Press ⊣↑⊢ .
4. Press → .
5. Enter 10006 into the AR.
6. Press ⊣ ⊢ .
7. Press → .
8. Enter 10010 into the AR.
9. Press ⊣ ⊢ .
10. Press → .
11. Enter 10002
12. Press ⊣/⊢ .
13. Press → .
14. Press the CHG NODE key.

15. Press COILS.
16. Enter 00007 into the AR.
17. Press —( )— .
18. Press → .
19. Press RELAYS.
20. Press VERTICAL SHORT.
21. Press ↓ .

    (Enter 00007 into the AR.)
22. Press —| |— .

This network is complete.

1.10 BASIC LOGIC EXAMPLES

1.10.1 Oscillator

The following paragraphs provide an explanation of the logic illustrated in Figure 1-9 to program an oscillator.

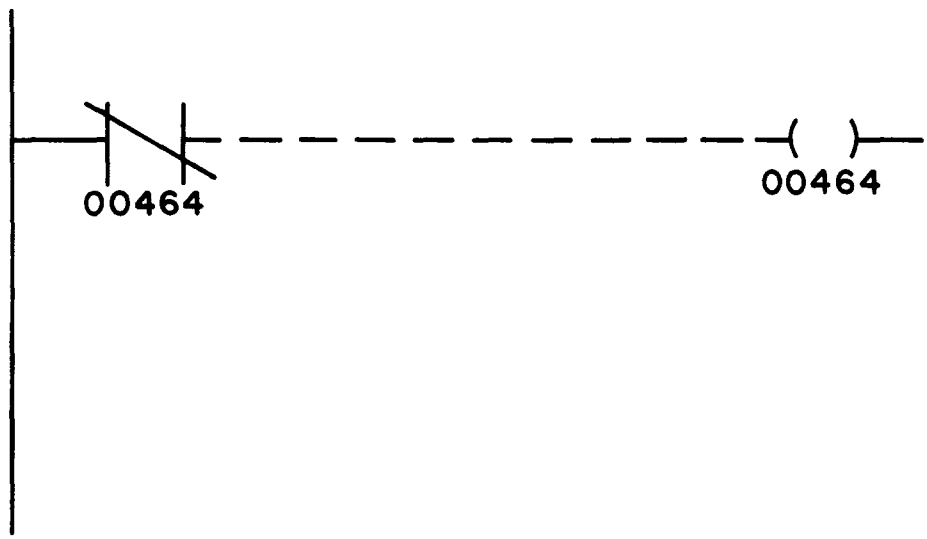

Figure 1-9. Oscillator

Many applications require the use of a signal which oscillates ON-OFF-ON-OFF. (On the first scan the coil is ON; on the second scan the coil is OFF; on the next scan the coil is ON; OFF, ON, OFF, etc.). Since the signal is ON every other scan, it can be used to flash lights or cause something to take place every other scan.

An oscillator is made using a logic coil and a normally closed contact with the same reference number.

1.10.2 Cascaded Logic

Very often, with a complex control function, more than ten nodes are needed to satisfy the function. In such a case, an internal coil is used to represent a partial result. To continue the logic flow, use a contact which is referenced to the internal coil. The second series of nodes can be ended with an output coil to represent the result of the logic or an internal coil to extend the logic again. The logic can be extended within a network and/or to another network.

NOTE

It is best to use a new network to continue cascaded logic. If a new network is not used, resulting logic solutions may be different than those desired. This is due to the order in which logic is solved.

The following paragraphs provide an explanation of the cascaded logic illustrated in Figure 1-10.

The first ten nodes in the top network illustrated in Figure 1-10 activate coil 00059. The logic is extended by referencing coil 00059 to the normally open contact in the next network. The logic is output by latched coil 00032.

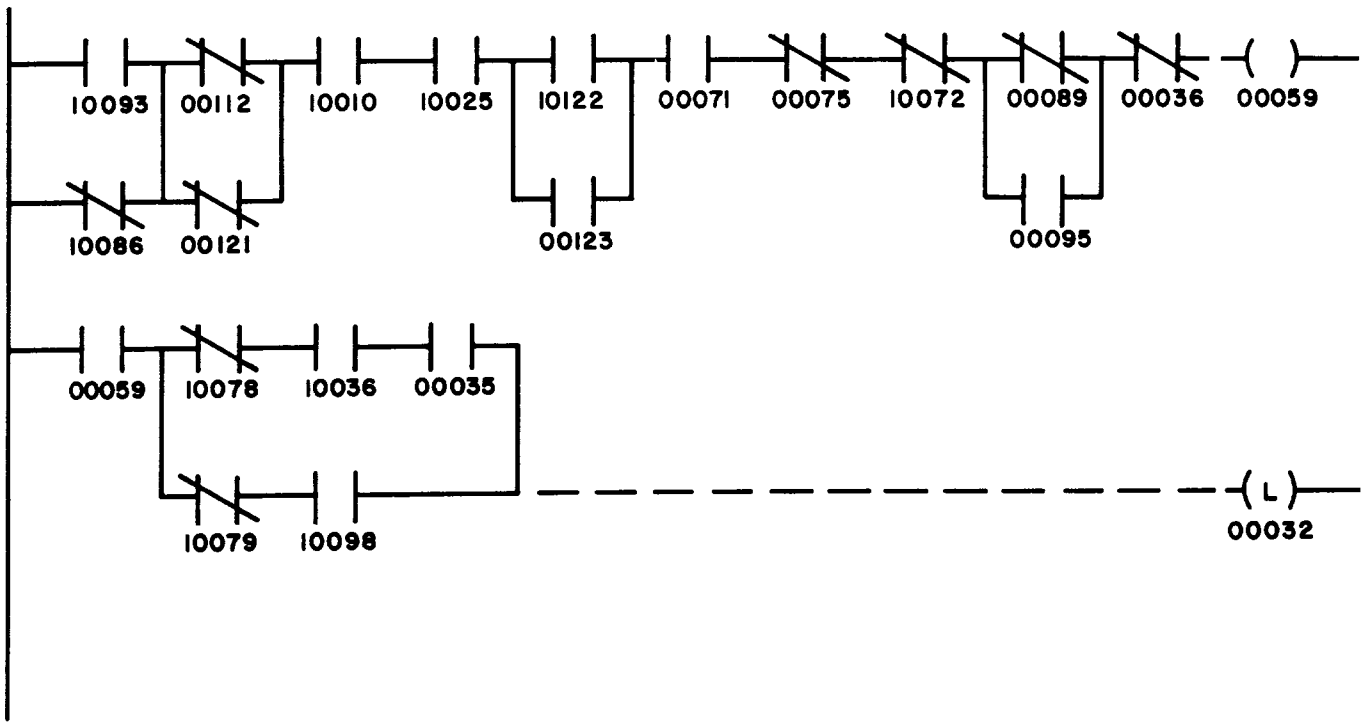

Figure 1-10. Cascaded Logic