SECTION 2 BASIC PROGRAMMING FUNCTIONS

This section describes how to edit a network as well as how to program a counter or timer. Instructions are also provided to perform arithmetic functions. These functions can be used independently or with the more advanced functions which are introduced in Sections 3 through 5.

Each function block in this section uses registers to hold and store data. A register is a location in the controller's memory in which a numerical value is stored. This value can be binary or binary coded decimal (BCD). In a 584L PC, the maximum decimal value is 9999 and the maximum number of bits is sixteen.

A function block is inserted into a program by entering the value desired for the top node of the function block into the AR and pressing the appropriate software label key. The function block appears with the desired value in the top node, and question marks in the other node(s).

How to edit a function block is explained in Table 2-1.

Table 2-1. Editing a Function Block

Type of Edit	Instructions
To replace the question marks with a value.	1. Position the cursor over the question marks. 2. Enter a value into the AR. 3. Press the ENTER key.
To change a value in a function block.	1. Position the cursor over the value. 2. Enter the new value into the AR. 3. Press the ENTER key.
To change the type of function block.	1. Position the cursor in the top node of the block. 2. Press the desired software label key.
To delete a function block.	1. Position the cursor in the top node of the block. 2. Press the DELETE NODE key.

2.1 EDITING

Once a whole or partial network has been entered, spacing changes can be made. This section highlights the four types of changes.

To make changes in a network:

1. Ensure that the Memory Protect key on the P190 is OFF and that Memory Protect on the 584L is OFF.
2. Display the network needing changes on the P190 screen.
3. Press the EXIT key.
4. Press the EDIT NETWORK software label key.

The following software labels are displayed:

EXPAND HORIZONTAL	EXPAND VERTICAL	COMPRESS HORIZONTAL	COMPRESS VERTICAL

To return to the programming software keys (e.g., RELAYS, COILS, etc.), press the CHG NODE (Change Node) key.

The Memory Protect keylock is located on the lower right front of the P190 Programmer, beneath the tape drive. The Memory Protect key is used to prevent accidental or unauthorized changes to the 584L PC's memory. If the key is in a vertical position, as shown in Figure 2-1a, Memory Protect is ON and the user's logic cannot be altered by an external device, only examined.

The key is turned clockwise to the unlock position as shown in Figure 2-1b. In this case, Memory Protect is OFF and changes can be made to the user's logic.

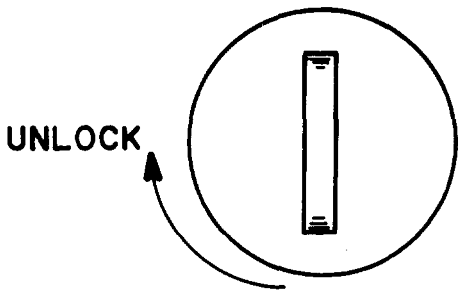


Figure 2-1a

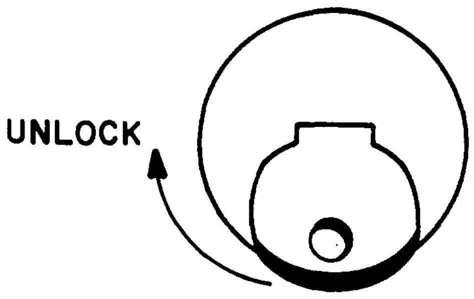


Figure 2-1b

Figure 2-1. Memory Protect

2.1.1 Expand Horizontal

This software label key is used to create a column in a network. When the key is pressed, the programming elements at the cursor position, and below, above, and to the right of the cursor, are moved one column to the right.

The logic in Figure 2-2 shows a network before pressing the EXPAND HORIZONTAL software label key. The shaded area is the cursor. The revised network is shown in Figure 2-3.

NOTE

The cursor can be placed anywhere in the column to be created. It does not have to be over an element. There must be a blank column at, or to the right of, the cursor or the EXPAND HORIZONTAL is not allowed.

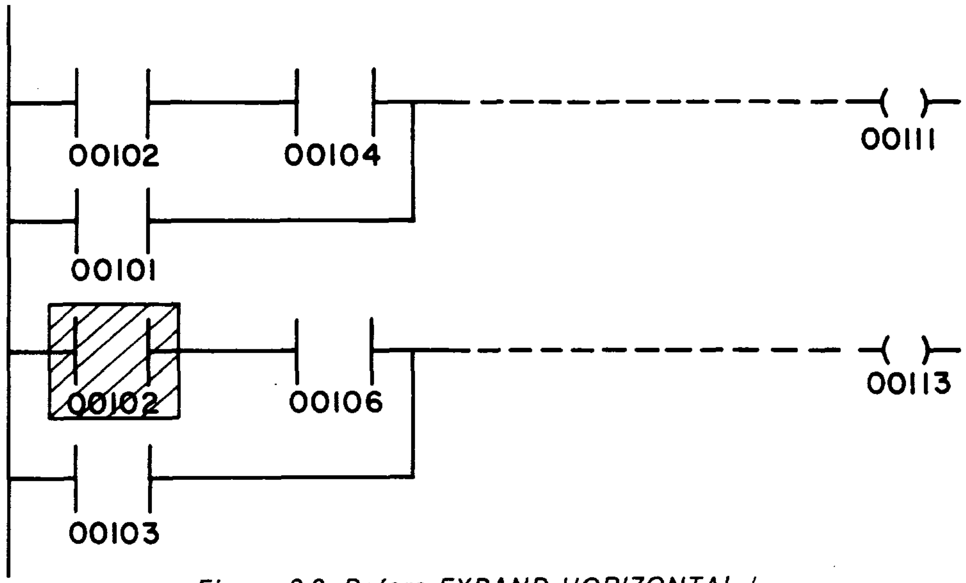


Figure 2-2. Before EXPAND HORIZONTAL / After COMPRESS HORIZONTAL

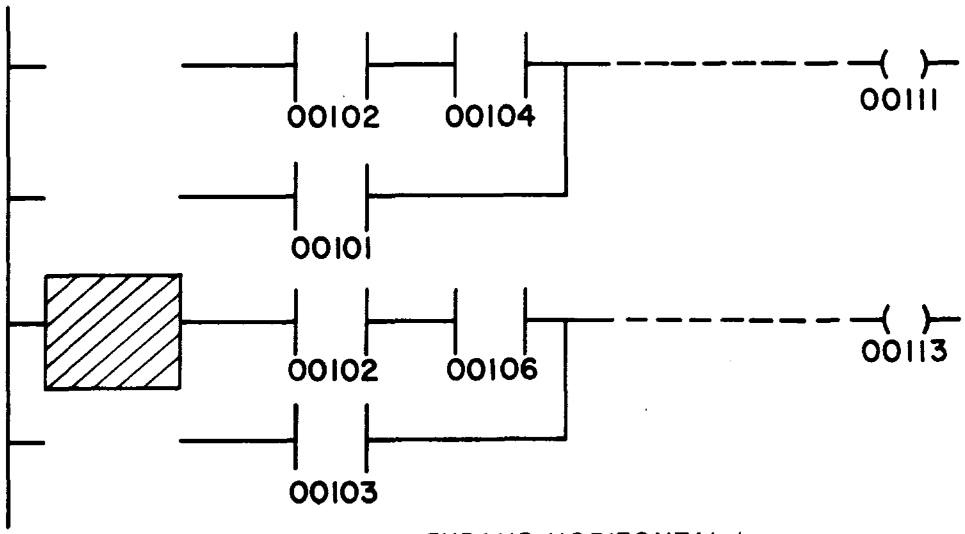


Figure 2-3. After EXPAND HORIZONTAL / Before COMPRESS HORIZONTAL

2.1.2 Compress Horizontal

The result of pressing this software label key is the opposite of pressing EXPAND HORIZONTAL. When the key is pressed, the column in which the cursor is located is deleted.

The logic in Figure 2-3 shows a network before pressing the COMPRESS HORIZONTAL software label key. The revised network is shown in Figure 2-2.

NOTE

COMPRESS HORIZONTAL is not allowed if the cursor is located over a programming element or if the column to be deleted contains a programming element.

2.1.3 Expand Vertical

This software label key is used to create a row in a network. When the key is pressed, the programming elements at the cursor position, and below, to the right, and to the left of the cursor, are moved one row down.

The logic in Figure 2-4 shows a network before pressing the EXPAND VERTICAL software label key. The revised network is shown in Figure 2-5.

NOTE

The cursor can be placed anywhere in the row to be created. It does not have to be over an element. Also, there must be a blank row under the last row containing programming elements or the EXPAND VERTICAL is not allowed.

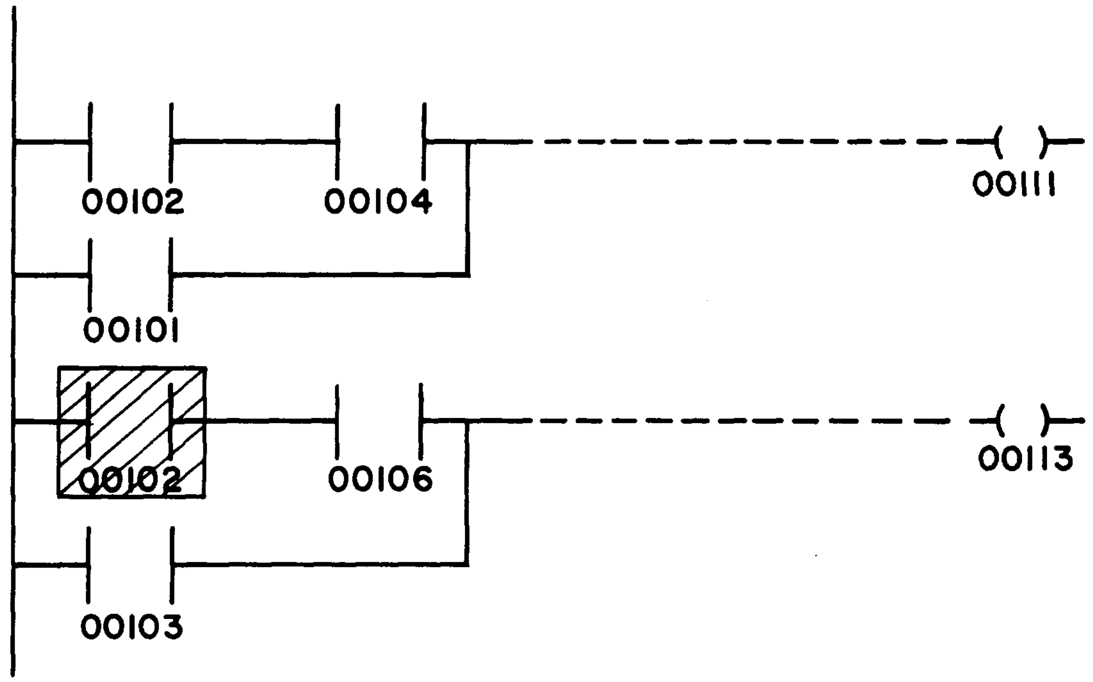


Figure 2-4. Before EXPAND VERTICAL / After COMPRESS VERTICAL

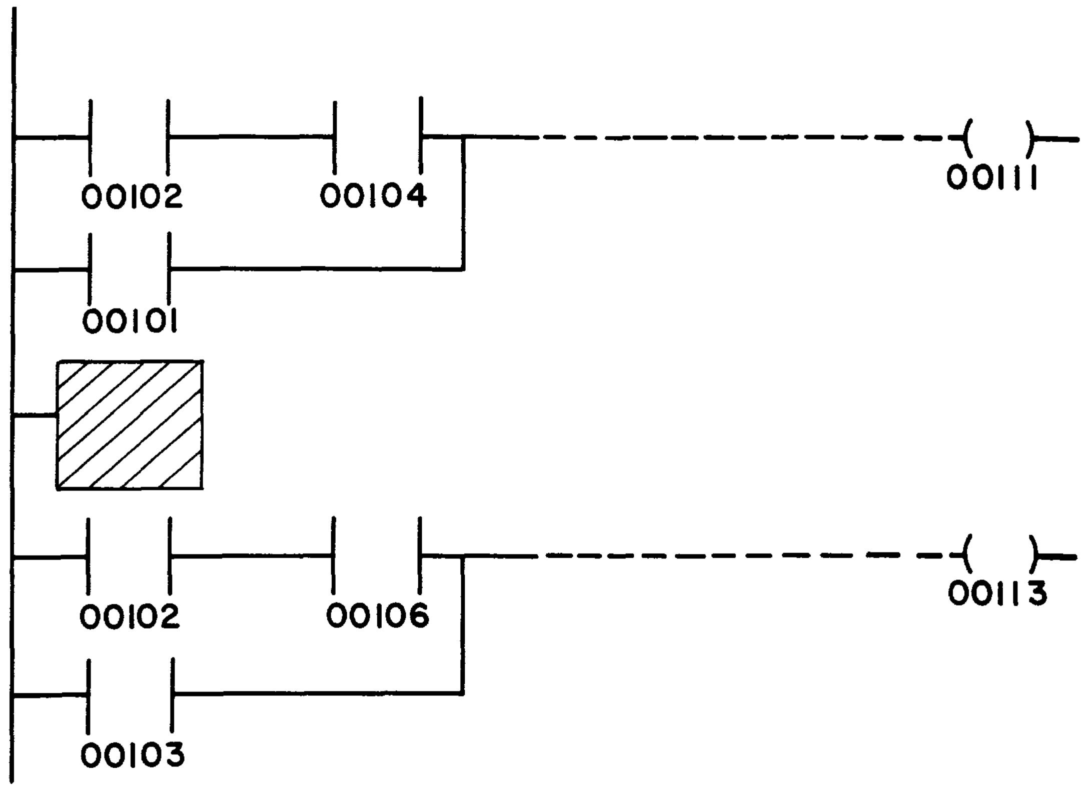


Figure 2-5. After EXPAND VERTICAL / Before COMPRESS VERTICAL

2.1.4 Compress Vertical

The result of pressing this software label key is the opposite of pressing EXPAND VERTICAL. When the key is pressed, the row in which the cursor is located is deleted.

The logic in Figure 2-5 shows a network before pressing the COMPRESS VERTICAL software label key. The revised network is shown in Figure 2-4.

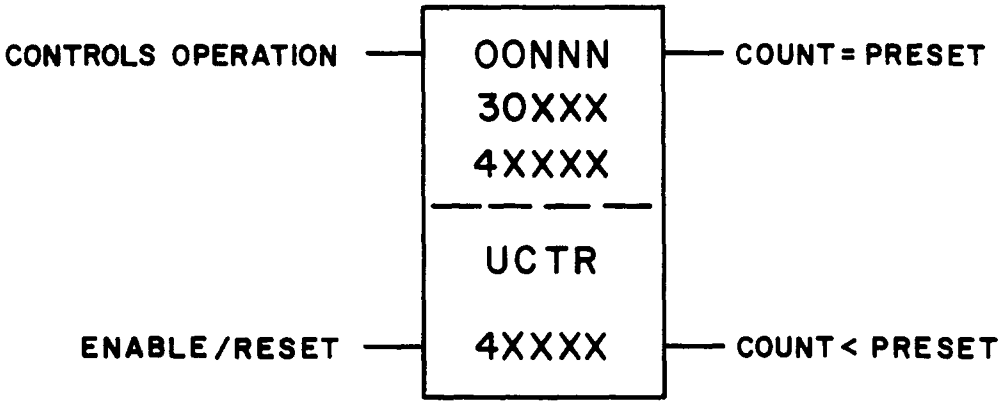
NOTE

COMPRESS VERTICAL is not allowed if the cursor is placed over a programming element or if the row to be deleted contains a programming element.

2.2 COUNTERS

2.2.1 Up Counter (UCTR)

FUNCTION
The Up Counter function counts the OFF to ON transitions of the control input. This counter increases by one upon each positive transition of the control input.



FUNCTION BLOCK
- The top node contains the preset value for the counter. It can be a constant, up to 999 in a Level 1 584L PC or up to 9999 in a Level 2 584L PC, a 30XXX input register reference, or a 4XXXX holding register reference. The contents of these register references or holding registers are the preset value.

- The bottom node contains the symbol UCTR and a unique 4XXXX holding register reference. This holding register contains the count value and increases, starting at zero, upon each OFF to ON transition of the control input.

NOTE

The controller sets the count value to the preset if an attempt is made to enter a value greater than the preset.

INPUTS
- The top input controls the operation. When it receives power, transitions from OFF to ON, the count value increases by one.

- The bottom input, when receiving power, enables the counter. When the input is not receiving power, the count value is reset to zero and any transitions of the top input are ignored.

OUTPUTS
- The top output passes power when the count value is equal to the preset value.

- The bottom output passes power when the count value is less than the preset value.

NOTE

Only one output passes power at a time.

EXAMPLE
The following paragraphs provide a detailed explanation of the logic used in Figure 2-6.

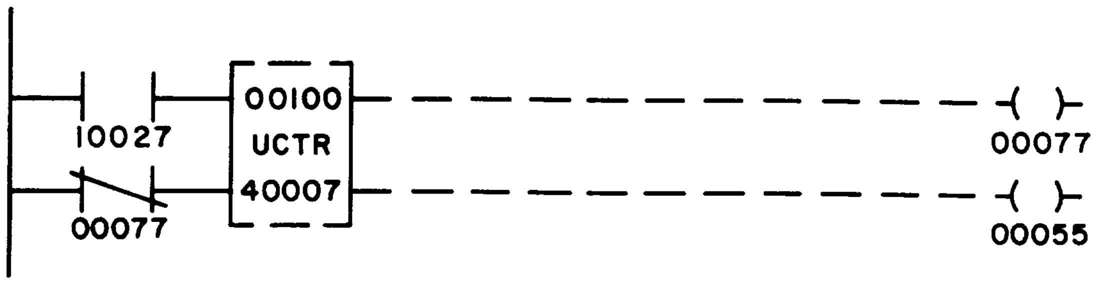


Figure 2-6. Up Counter

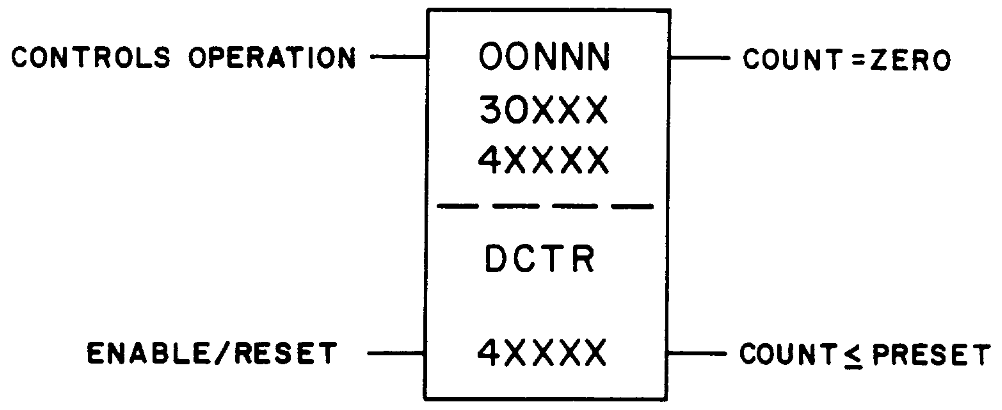
When input 10027 is energized, the top input receives power and, since the bottom input is also receiving power, the counter is enabled and counting begins.

Each time input 10027 transitions from OFF to ON, the value in register 40007 increases by one. When this value reaches 100, the top output passes power. Coil 00077 is energized and coil 00055 is de-energized. The bottom input loses power when coil 00077 is energized (the normally closed contact does not pass power), and the counter value is reset to zero.

On the next scan that input I0027 transitions OFF to ON, coil 00077 is de-energized, thereby energizing input 00077 and re-enabling the counter.

2.2.2 Down Counter (DCTR)

FUNCTION
The Down Counter function counts the OFF to ON transitions of the control input. The counter decreases by one upon each positive transition of the control input.



FUNCTION BLOCK
- The top node contains the preset value for the counter. It can be a constant, up to 999 in a Level 1 584L PC or up to 9999 in a Level 2 584L PC, a 30XXX input register reference, or a 4XXXX holding register reference. The contents of this register are the preset values.

- The bottom node contains a unique 4XXXX holding register reference. This holding register contains the count value and decreases, starting at the preset value, upon each OFF to ON transition of the control input.

NOTE

The controller sets the count value to the preset if an attempt is made to enter a value greater than the preset.

INPUTS
- The top input controls the operation. When it receives power, transitions from OFF to ON, the count value decreases by one.

- The bottom input, when receiving power, enables the counter. When the input is not receiving power, the count value is reset to the preset value and any transitions of the top input are ignored.

OUTPUTS
- The top output passes power when the count value is equal to zero.

- The bottom output passes power when the count value is less than or equal to the preset value but not equal to zero.

NOTE

Only one output passes power at a time.

EXAMPLE
The following paragraphs provide a detailed explanation of the logic used in Figure 2-7.

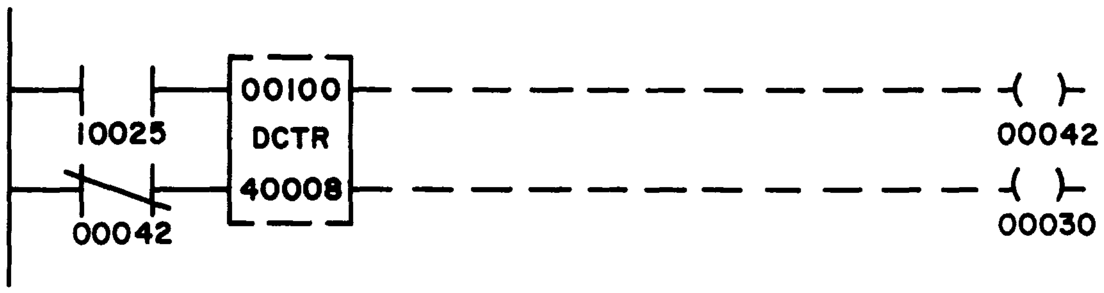


Figure 2-7. Down Counter

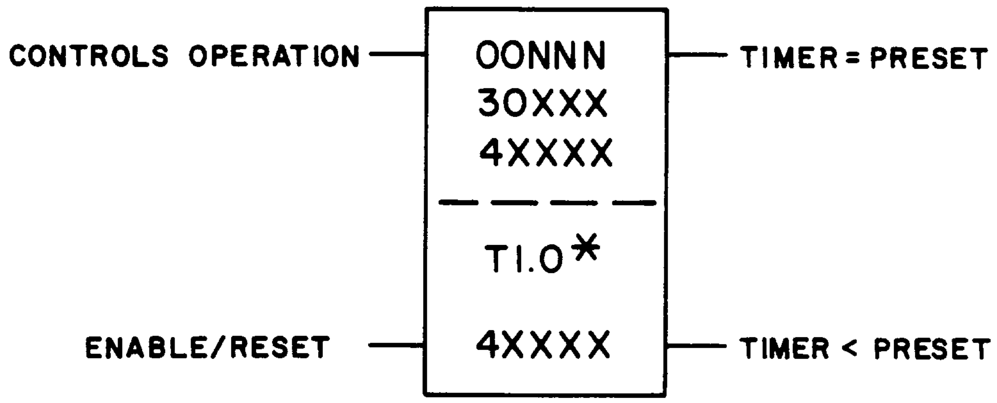
When input 10025 is energized, the top input receives power and, since the bottom input is also receiving power, the counter is enabled and down-counting (decreasing) begins.

Each time input 10025 transitions from OFF to ON, the value in register 40008 decreases by one (this value starts at the preset). When this value reaches zero, the top output passes power. Coil 00042 is energized and coil 00030 is de-energized. The bottom input loses power when coil 00042 is energized (the normally closed contact does not pass power), and the counter value is reset to the preset (100).

On the next scan input that 10025 transitions OFF to ON, coil 00042 is de-energized, thereby energizing input 00042 and re-enabling the counter.

2.3 TIMER (T1.0*)

FUNCTION
The Timer function uses any one of three clocks in the 584L PC to record time. The 584L is capable of counting time in seconds, tenths of seconds, and hundredths of seconds.



* T1.0	=	seconds;
T0.1	=	tenths of seconds;
T.01	=	hundredths of seconds.

FUNCTION BLOCK
- The top node represents the preset value for the timer. It can be a constant, up to 999 in a Level 1 584L PC or up to 9999 in a Level 2 584L PC, a 30XXX input register reference, or a 4XXXX holding register reference. The contents of this register are the preset value.

  For example, if the preset value is 009, it is either 9 seconds (T1.0), .9 seconds (T0.1) or .09 seconds (T.01). If the value is 050, it is either 50 seconds (T1.0), 5 seconds (T0.1), or .5 seconds (T.01). (Hint: take the preset value and divide by either 1, 10, or 100 to find the value it represents.)

- The bottom node contains a unique 4XXXX holding register reference. This register holds the timer value and increases in time, starting at zero and going up to the preset value, as long as both the top and bottom inputs are receiving power.

NOTE

The controller sets the timer value to the preset if an attempt is made to enter a value greater than the preset.

INPUTS
- The top input controls the operation. When it receives power and the timer is enabled, the timer value increases time. When the top input loses power, the timer stops increasing.
- The bottom input, when receiving power, enables the timer. When this input is not receiving power, the timer value is reset to zero. The timer does not increase in time if only the top input is receiving power.

OUTPUTS
- The top output passes power when the timer value equals the preset value.
- The bottom output passes power when the timer value is less than the preset value.

NOTE

Only one output passes power at a time.

EXAMPLE
The following paragraphs provide a detailed explanation of the logic used in Figure 2-8.

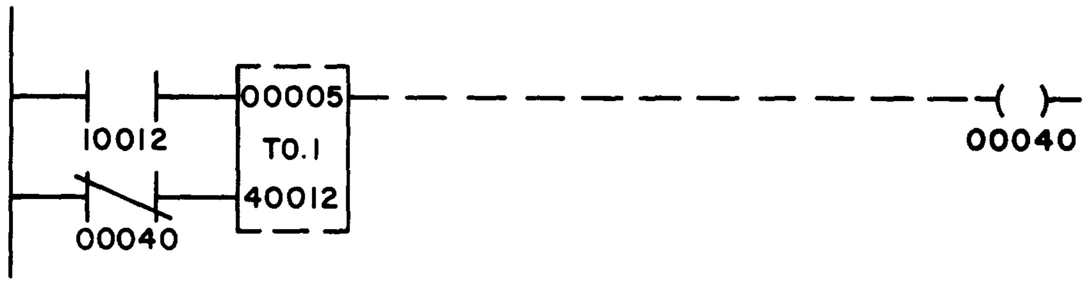


Figure 2-8. Timer

When input 10012 is energized, the top input receives power and, since the bottom input is also receiving power, the timer is enabled and register 40012 begins accumulating time in tenths of seconds. When the value in register 40012 equals 5 (.5 or ½ second), the top output passes power and energizes coil 00040. The bottom input loses power when coil 00040 is energized (the normally closed contact does not pass power), and the timer value (register 40012) is reset to zero.

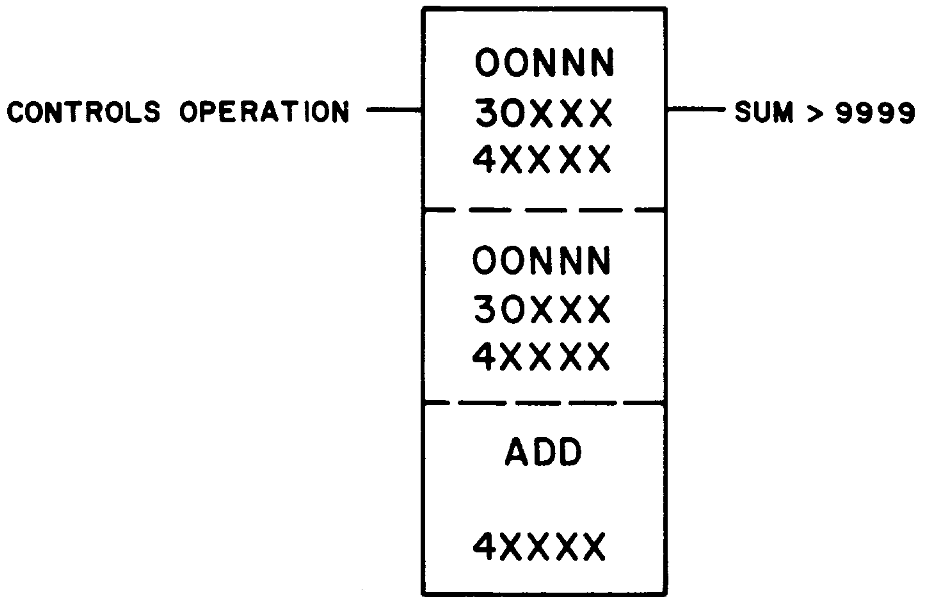
On the next scan that input 10012 is energized, coil 00040 is de-energized, thereby energizing input 00040 and re-enabling the timer.

2.4 ARITHMETIC FUNCTIONS

The following arithmetic functions are available: Addition, Subtraction, Multiplication, and Division. All four function blocks occupy three nodes in a network and place the result of the operation in the bottom node's holding register. The top input of each function block controls the operation; when it is receiving power the function is performed.

2.4.1 Addition (ADD)

FUNCTION
The Add function adds two values together (top and middle nodes) and places the sum in a holding register (bottom node).



FUNCTION BLOCK
- The top and middle nodes contain values which can be either constants, up to 999 in a Level 1 584L PC or up to 9999 in a Level 2 584L PC, a 30XXX input register reference, or a 4XXXX holding register reference. The top and middle node values are added together when the control input receives power.

- The bottom node contains 4XXXX holding register reference. This holding register holds the sum of the top and middle node values.

INPUT
- The top input controls the operation. When it receives power, the value in the top node is added to the value in the middle node and the sum is placed in the bottom node's holding register.

OUTPUT
- The top output passes power when the sum is greater than 9999; it indicates that a 1 should be placed in front of the result located in the holding register. For example, if 6500 is added to 5000, the result in the holding register is 1500 and the top output passes power. The actual sum is 11,500.

NOTE

Only the top input and top output are used for this function.

EXAMPLE
The following paragraphs provide a detailed explanation of the logic used in Figure 2-9.

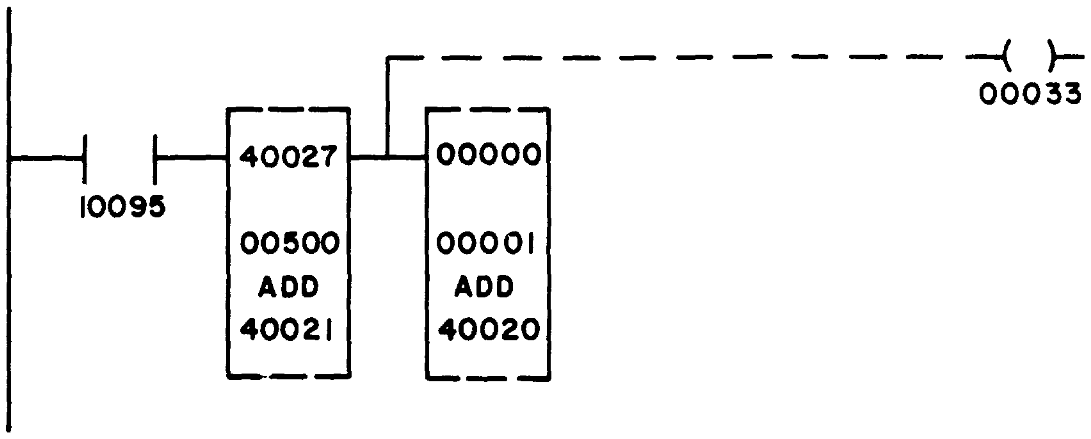


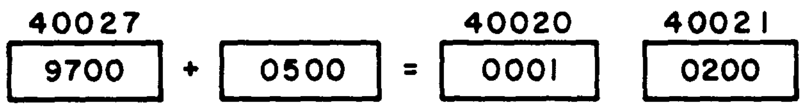
Figure 2-9. Addition

When input 10095 is energized, the top input of the first add block receives power and the content of register 40027 is added to the fixed value 00500. The sum is placed in register 40021.

If the content of register 40027 is 9700, the result placed in register 40021 is 0200 and the top output passes power. Coil 00033 is energized and the top input of the second add block receives power. This function places a one in register 40020 so the sum of the two numbers is read properly.

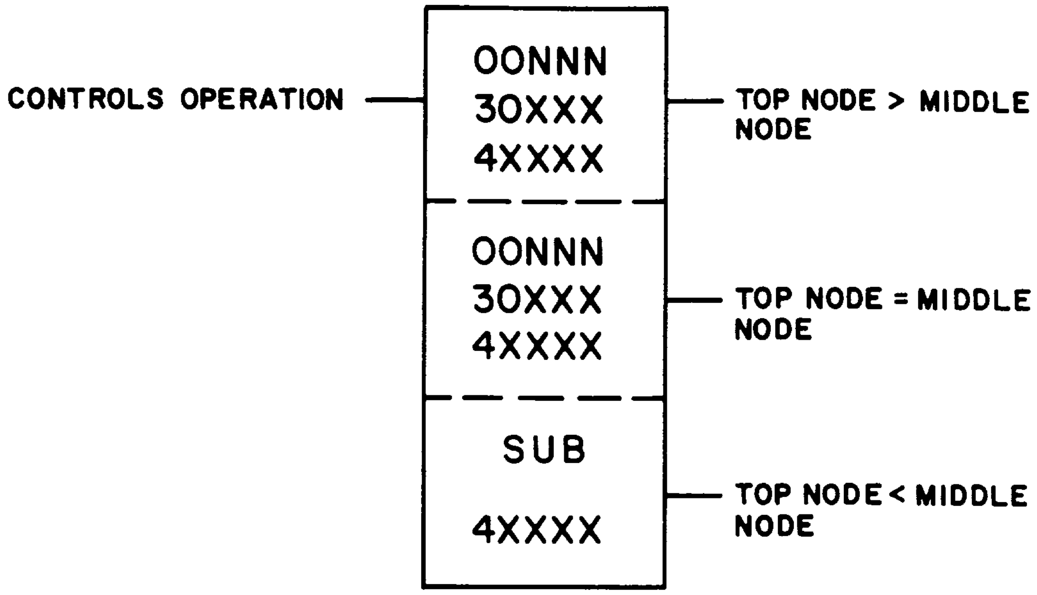
NOTE

The add block may give incorrect results with sums over 20,000.



2.4.2 Subtraction (SUB)

FUNCTION
The Subtract function subtracts the value in the middle node from the value in the top node and places the difference in the bottom node's holding register.



FUNCTION BLOCK
- The top node contains a value which can be a constant, up to 999 in a Level 1 584L PC or up to 9999 in a Level 2 584L PC, a 30XXX input register reference, or a 4XXXX holding register reference. The value in the middle node is subtracted from this value when the control input receives power.

- The middle node contains a value which can be a constant, up to 999 in a Level 1 584L PC or up to 9999 in a Level 2 584L PC, a 30XXX input register reference, or a 4XXXX holding register reference. This value is subtracted from the value in the top node.

- The bottom node contains the symbol SUB and a 4XXXX holding register reference. This holding register contains the difference between the top and middle node values.

INPUT
- The top input controls the operation. When it receives power, the value in the middle node is subtracted from the value in the top node, and the difference is placed in the bottom node's holding register.

OUTPUTS
- The top output passes power when the value in the top node is greater than the value in the middle node.

- The middle output passes power when the value in the top node equals the value in the middle node.

- The bottom output passes power when the value in the top node is less than the value in the middle node.

NOTE

The use of any or all outputs is optional. Two outputs can be connected together with vertical shorts to create a greater than or equal to function using the top and middle outputs, or a less than or equal to function using the middle and bottom outputs (i.e., these may be required for set point control or alarm limits).

EXAMPLE
The following paragraphs provide a detailed explanation of the logic used in Figure 2-10.

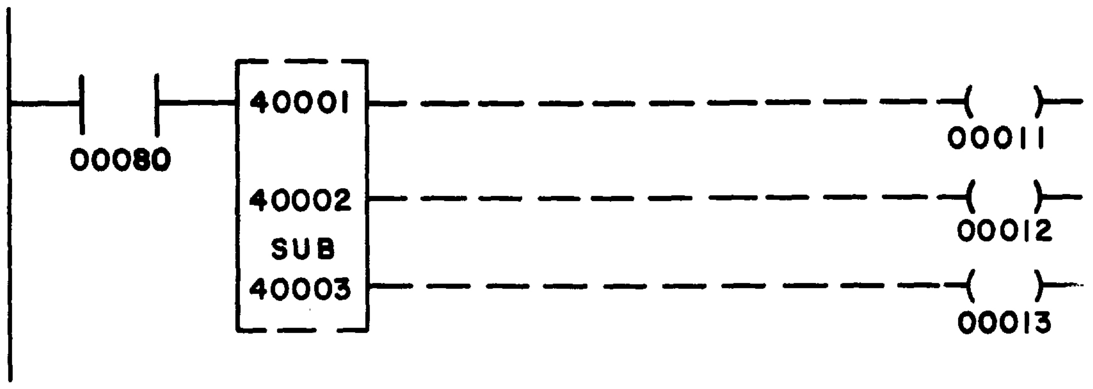


Figure 2-10. Subtraction

When contact 00080 is energized, the top input of the function block receives power and the subtract is performed. The value in register 40002 is subtracted from the value in 40001 and the result is placed in register 40003.

If the value in 40001 is greater than the value in 40002, the top output passes power and energizes coil 00011. This indicates a normal subtract function.

If the value in 40001 is equal to the value in 40002, the middle output passes power and energizes coil 00012. This indicates a difference of zero.

If the value in 40001 is less than the value in 40002, the bottom output passes power and energizes coil 00013. This indicates that the answer to the subtract should be negative.

NOTE

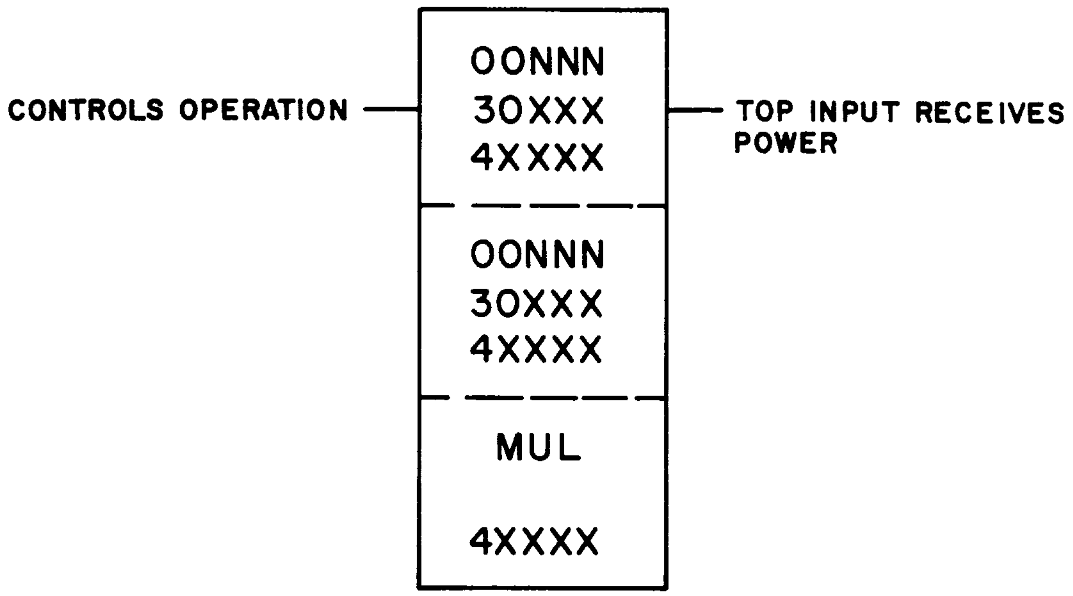
The value placed in register 40003 is the absolute value of the difference — no sign is associated with the content of register 40003.

If the value in 40001 is 9000 and the value in 40002 is 0500, the result placed in 40003 is 8500.

40001		40002		40003
9000	–	0500	=	8500

2.4.3 Multiplication (MUL)

FUNCTION
The Multiply function calculates the product of the values in the top and middle nodes and places the answer in two consecutive holding registers referenced to in the bottom node of the function block. Two consecutive holding registers are used because the product of two 4-digit numbers can be up to eight digits in length.



FUNCTION BLOCK
- The top and middle nodes each contain a value which can be a constant, up to 999 in a Level 1 584L PC or up to 9999 in a Level 2 584L PC, a 30XXX input register reference, or a 4XXXX holding register reference. The top and middle node values are multiplied.

- The bottom node contains both the symbol MUL and a 4XXXX holding register reference. This holding register holds the high order portion of the product even if it is zero. The next consecutive holding register (4XXXX + 1) holds the low order portion of the product. For this reason, the last available holding register cannot be used.

INPUT
- The top input controls the operation. When it receives power, the value in the top node is multiplied by the value in the middle node and the product is placed in the bottom node's holding register(s).

OUTPUT
- The top output passes power when the top input receives power. This allows the function blocks to be cascaded within a network.

NOTE

Only the top input and the top output are used.

EXAMPLE
The following paragraphs provide a detailed explanation of the logic used in Figure 2-11.

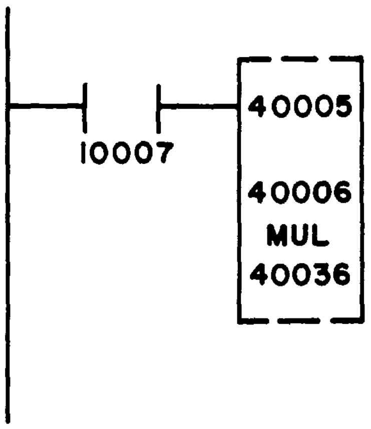
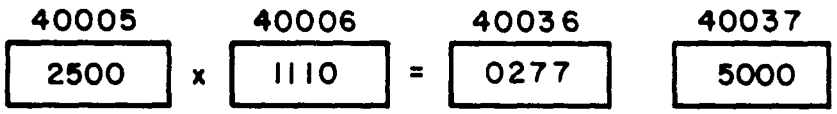


Figure 2-11. Multiplication

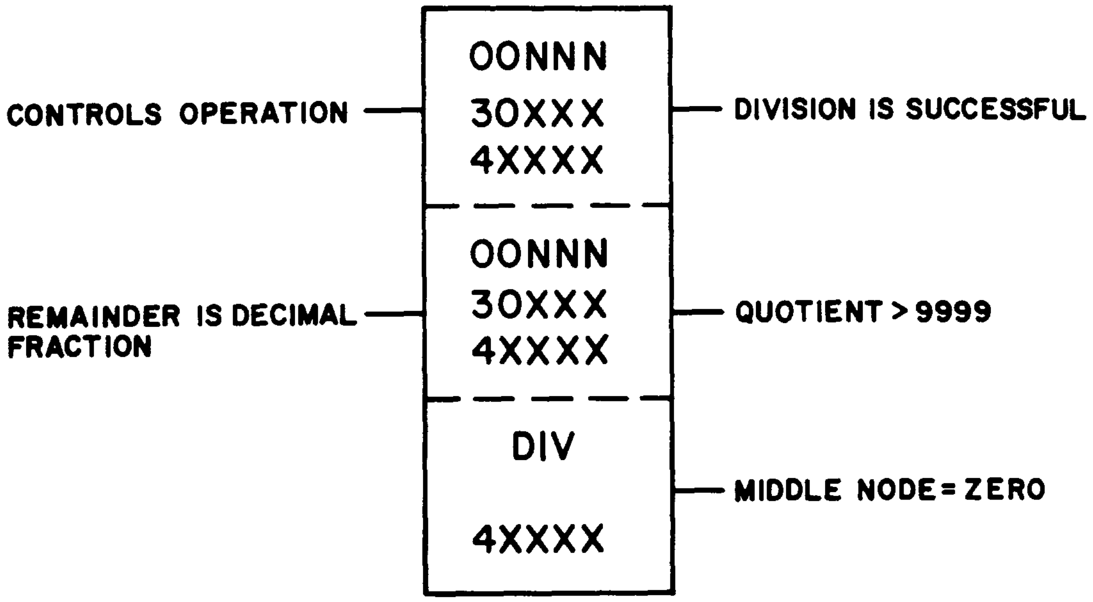
Multiply operates upon two four-digit numbers to produce an eight-digit product. In this example, when input 10007 is energized, the top input receives power and the value in register 40005 is multiplied by the value in register 40006. The resulting product is stored in registers 40036 and 40037.

If registers 40005 and 40006 contain the values 2500 and 1110 the product is 2,775,000. Thus, register 40036 is loaded with the value 0277 and register 40037 is loaded with the value 5000. The two registers are multiplied and the product is stored every scan input 10007 is energized.



2.4.4 Division (DIV)

FUNCTION
The divide function divides the value in the top node by the value in the middle node and places the quotient in the bottom node's holding register.



FUNCTION BLOCK

- The top node is the dividend. It can be a constant, up to 999 in a Level 1 584L PC or up to 9999 in a Level 2 584L PC, a 30XXX input register reference or a 4XXXX holding register reference. Two consecutive registers are used and both are needed for a double precision number. If a single precision number is desired and registers are being used, fill the first register with zeros. When a register reference is used, the next register is implied; therefore, the last available register, input or holding, cannot be used in this node.

- The middle node is the divisor. It contains a constant, up to 999 in a Level 1 584L PC or up to 9999 in a Level 2 584L PC, a 30XXX input register reference, or a 4XXXX holding register reference.

- The bottom node contains the symbol DIV and a 4XXXX holding register. This holding register holds the result of the division. The next holding register (4XXXX + 1) holds the remainder; therefore, the bottom node cannot contain the last available holding register.

INPUTS

- The top input controls the operation. When it receives power, the value in the top node is divided by the value in the middle node, and the quotient and remainder are placed in two consecutive holding registers.

- The middle input, when receiving power, causes the remainder to be a decimal fraction. If it does not receive power the remainder is a whole number. For example, 10 divided by 3 has a decimal remainder of .3333 and a whole number remainder of 1.

OUTPUTS

- The top output passes power when the division is successful. If the middle or bottom outputs pass power, the division is unsuccessful and the top output does not pass power.

- The middle output passes power when the quotient is greater than 9999. If this output passes power, zeros are placed in the bottom node's holding registers.
- The bottom output passes power when the divisor (middle node) equals zero. If this output passes power, zeros are placed in the bottom node's holding registers.

EXAMPLE
The following paragraphs provide a detailed explanation of the logic used in Figure 2-12.

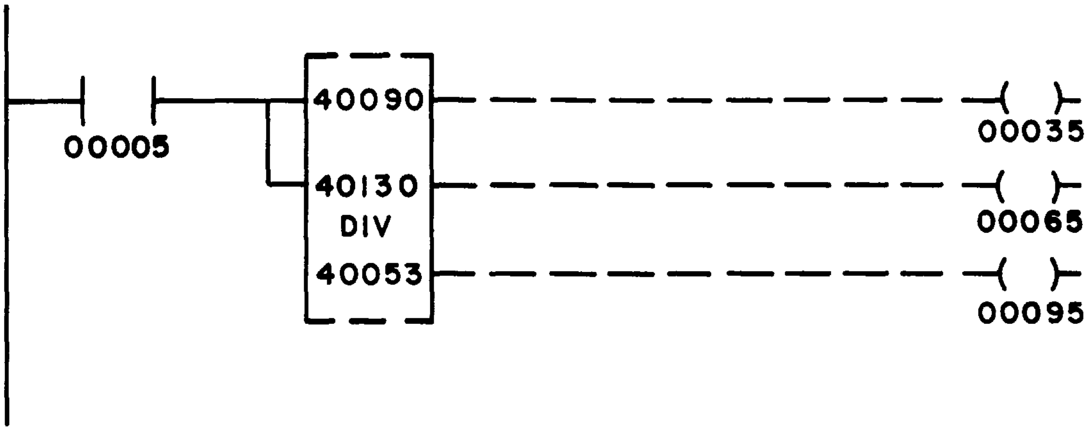


Figure 2-12. Division

When contact 00005 is energized, the top input receives power and the content of registers 40090 and 40091 (double precision number) is divided by the content of register 40130. The result is placed into register 40053 and the remainder into register 40054. Since the middle input is receiving power, the remainder is a decimal fraction rather than a whole number.

If the double precision number is 1,234,567 (40090 = 0123, 40091 = 4567) and the divisor (40130) is 0236, the value 5231 is placed in register 40053 and the decimal remainder 2161 is placed in register 40054.

$$\begin{array}{ccccccccc} 40090 & & 40091 & & 40130 & & 40053 & & 40054 \\ \boxed{0123} & & \boxed{4567} & \div & \boxed{0236} & = & \boxed{5231} & \bullet & \boxed{2161} \end{array}$$

If the middle input was not receiving power, the whole remainder 0051 would be placed in register 40054. Coil 00035 is energized if the division is successful.

Coil 00065 is energized if the quotient is greater than 9999. Coil 00095 is energized if the divisor (middle node) equals zero.

2.5 SUMMARY OF BASIC FUNCTIONS

	TOP NODE	MIDDLE NODE	BOTTOM NODE	TOP INPUT	MIDDLE INPUT	BOTTOM INPUT	TOP OUTPUT	MIDDLE OUTPUT	BOTTOM OUTPUT
UCTR	00NNN, 30XXX, or Preset	— 4XXXX	4XXXX	Control	—	Enable/Reset	Count = Preset	—	Count Preset
DCTR	00NNN, 30XXX, or 4XXXX Preset	—	4XXXX	Control	—	Enable/Reset	Count = Zero	—	Count Preset
TIMER	00NNN, 30XXX, or 4XXXX Preset	—	4XXXX	Control	—	Enable/Reset	Timer = Preset	—	Timer Preset
ADD	00NNN, 30XXX, or 4XXXX	00NNN, 30XXX, or 4XXXX	4XXXX	Control	Not Used	Not Used	Sum > 9999	Not Used	Not Used
SUB	00NNN, 30XXX, or 4XXXX	00NNN, 30XXX, or 4XXXX	4XXXX	Control	Not Used	Not Used	Top Node > Middle Node	Top Node = Middle Node	Top Node < Middle Node
MUL	00NNN, 30XXX, or 4XXXX	00NNN, 30XXX, or 4XXXX	4XXXX	Control	Not Used	Not Used	Top Input Receiving Power	Not Used	Not Used
DIV	00NNN, 30XXX, or 4XXXX	00NNN, 30XXX, or 4XXXX	4XXXX	Control	Remainder is Decimal Fraction	Not Used	Division is Sucessful	Quotient > 9999	Middle Node Equals Zero

2.6 LOGIC EXAMPLES

2.6.1 Real Time Clock

The following paragraphs provide a detailed explanation of the logic illustrated in Figure 2-13 to program a real time clock.

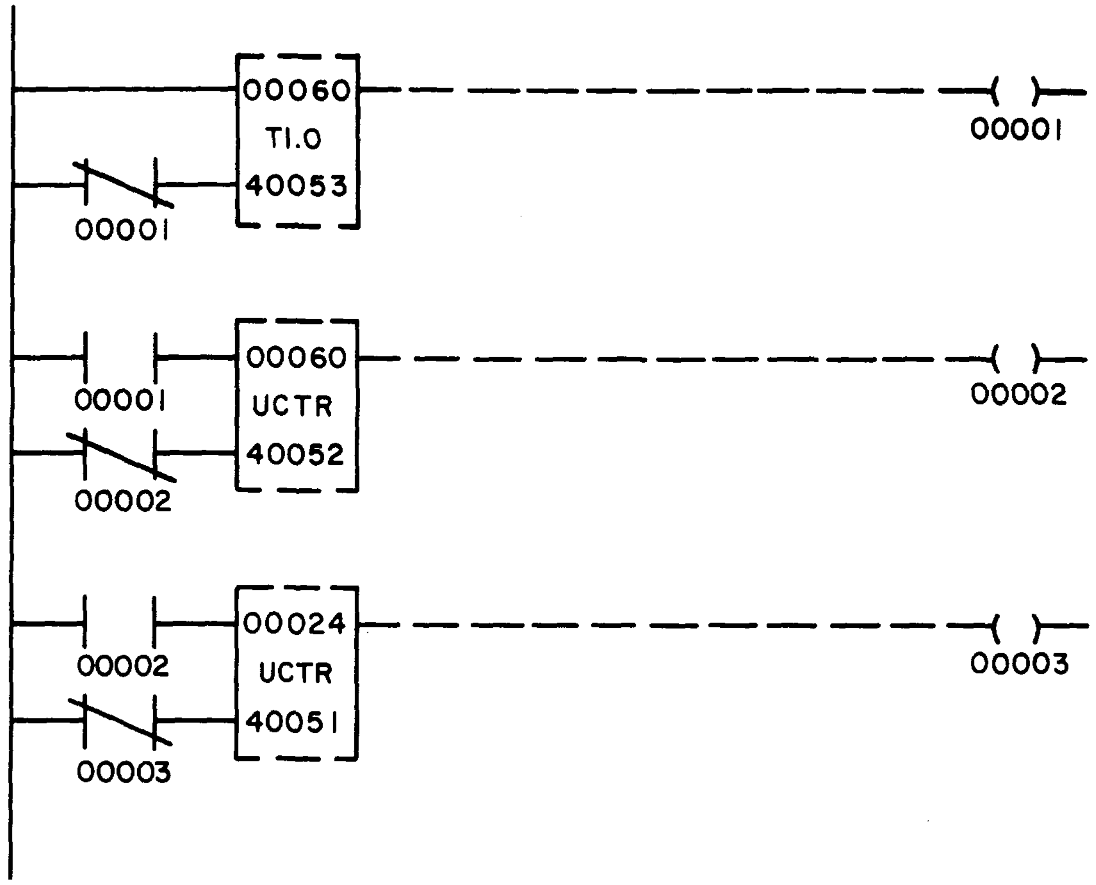


Figure 2-13. Real Time Clock

The top network in this example is the 1 minute timer. At the beginning of the logic solving, coil 00001 is OFF so both the top and bottom inputs of the timer are receiving power. Register 40053 starts increasing time in seconds until it reaches 60. At this point, the top output passes power and energizes coil 00001. Register 40053 is reset and the counter in the second network (40052) increases by one, indicating that one minute has elapsed.

Since the timer in network 1 is no longer equal to the preset, coil 00001 is de-energized and the timer resumes increasing time. Once the value in 40052 reaches 60, indicating 60 minutes, the top output passes power and energizes coil 00002. Register 40052 is reset and the counter in the third network (40051) increases by one, indicating that one hour has passed. The correct time of day can be read in registers 40051 — 40053 in hours, minutes, and seconds.

2.6.2 Fahrenheit to Centigrade Conversion

The following paragraphs provide a detailed explanation of the logic illustrated in Figure 2-14 to program a Fahrenheit to Centigrade temperature conversion.

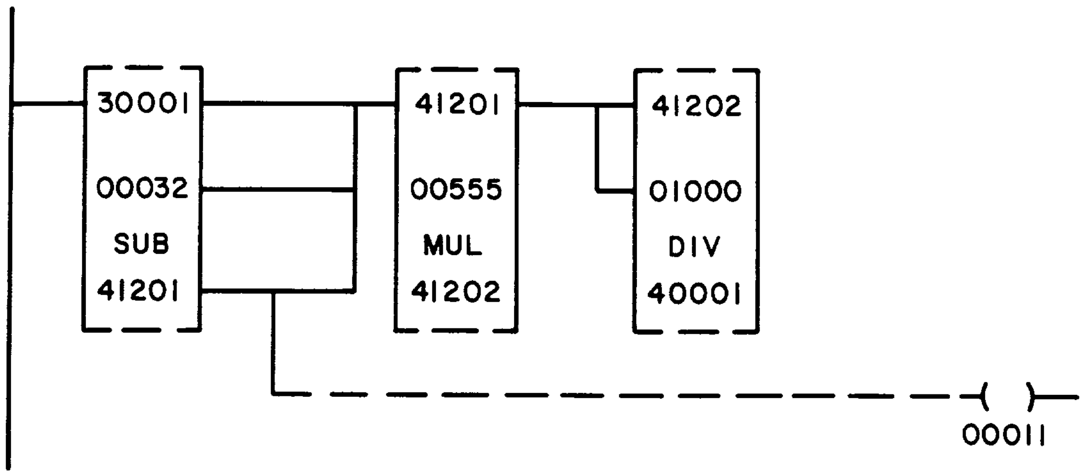


Figure 2-14. Fahrenheit to Centigrade Conversion

In this example, when the top input of the subtract function block receives power, the number 32 is subtracted from the value in register 30001 (degrees fahrenheit). The difference is placed in register 41201.

Whether the answer is positive, negative, or zero, the top input of the multiply function block receives power. If the answer to the subtract is negative, coil 00011 is energized to indicate a negative value. The value in 41201 is multiplied by 555 and the product is placed in register 41202.

The top input of the divide function block receives power and the value in 41202 is divided by 01000. The temperature conversion in degrees centigrade is placed in 40001.

2.6.3 Double Precision Add

The following paragraphs provide a detailed explanation of the logic illustrated in Figure 2-14 to program a double precision add.

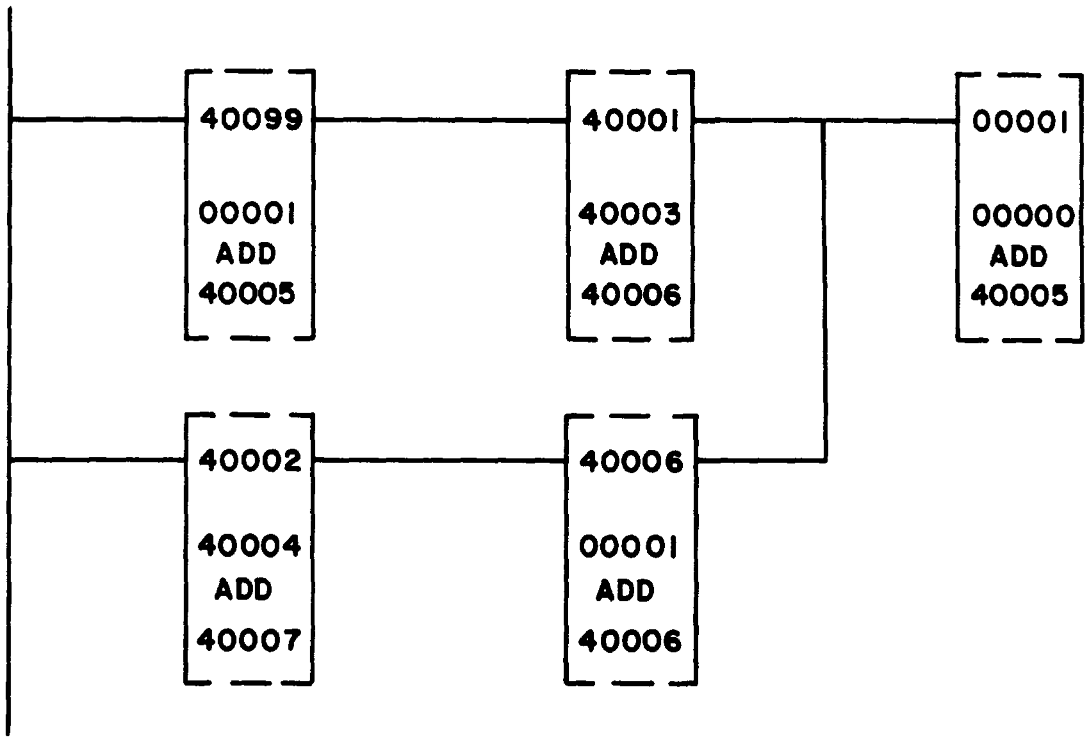


Figure 2-15. Double Precision Add

In this example, an eight-digit number in registers 40001 and 40002 is added to an eight-digit number in registers 40003 and 40004. The nine-digit result is placed in registers 40005-40007. The ADD function block at the top left of the network simply clears register 40005 to zero on each scan.

Since an overflow condition exists power is passed to the next function block. If the result of addition in this block is greater than 9999, the top output passes power to place a one in register 40005.

The second half of the addition takes place in bottom left function block, and passes power if an overflow condition exists. If an overflow exists, register 40006 is incremented by one to account for it. If 40006 overflows when this one is added to it, power is passed to increase 40005 to one.

If the following values are in the registers:

40001	40002		40003	40004		40005	40006	40007
9760	3842	+	6553	8317	=	0001	6314	2159

2.6.4 Subtract Greater Than 9999

In the Subtract function block, it is possible to have a value as large as 32,767 in the top node. This is possible if the register value is in binary format. There are sixteen bits available in a register. The first (most significant) bit is the sign bit, zero being negative and one being positive. The other fifteen bits, when all ones, equal 32,767.