beginning of this section have been modified to suit various customer applications; this has resulted in a number of available MOPS as listed in Table 9.

NOTE

See Appendix F for list of 384 executive (TEF) programs.

Any program listed can be supplied with any 184/384 Controller that has sufficient core memory. These programs can also be loaded via the Telephone Interface or Tape Loader. Additional MOPS/TEF programs are periodically developed based on customer requirements and, once developed, are made available to any 184/384 Controller user.

Table 9. Model 184 Controller Executive Program

MOPS 1: Relays, Timers, and Counter

Model	No. of Lines	No. of Holding Registers	Discrete I/O (Channel)	Latches	Min. Core Size	Special Functions
1	224	32	224 I 224 O (1,2)	224 2001- 2224	1K	
2	464	100	256 I 256 O (1,2)	304 2001- 2304	2K	
3	640	100	352 I 352 O (1,2,3)	32 2001- 2032	3K	
4	736	100	256 I 256 O (1,2)	32 2001- 2032	3K	
5	464	100	256 I 256 O (1,2)	304 2001- 2304	2K	Timers: 1/100 and 1/10 sec. divide panel timer labels by 10.
6	512	78	256 I 256 O (1,2)	256 2001- 2256	2K	
7	736	100	128 I 128 O (1)	160 2065- 2224	3K	
8	640	500	352 I 352 O (1,2,3)	32 2001- 2032	3K	Timers: 1/100 and 1/10 sec divide panel timer labels by 10.

3.3 ADVANCED CONCEPTS

3.3.1 Registers

The MODICON 184/384 Controller accepts, stores, performs operations on, and outputs data using state-of-the-art processing hardware and software. The degree to which a Controller can manipulate this data and produce the various outputs is determined by its own executive program. As described previously, the 184 MOPS programs can be modified to suit various applications. In addition to modifying the number of discrete I/O, logic lines, and latches, the MOPS can be modified to provide numerical inputs and outputs.

An example of input registers would be to connect a four-digit BCD thumbwheel from an operator's panel to an input module. Thus, by proper programming, the operator can be provided with control over the logic by adjusting the thumbwheels. An output register can be similarly used to provide numerical information to the operator via a four-digit BCD display.

In all cases, each register I/O is provided with up to four digits. Since each BCD digit requires four wires, a complete 16-circuit I/O module is used for each register I/O. The same type of I/O modules are used for registers as are used for discrete I/O; the selection of the specific I/O modules depends on the equipment to which the module is to be connected. See Appendix B for a summary of the available I/O modules.

When used as register I/O, reference numbers of the form 30xx identify input registers and 40xx identify output registers. Typically channels III and IV are dedicated to providing register I/O when the MOPS has been modified to provide this capability; channels I and II remain for discrete I/O (see Table 7).

Depending on the MOPS selected, there are a number of options to the reference numbers for channels III and IV as shown in Table 10. All 384 TEF programs provide register I/O (see Appendix F). The three examples shown in this table are for the executives currently written; most executives provide 256 discrete I/O and 16 register I/O.

Normally, I/O registers are assumed to be coded BCD; however, with a minor change to the MOPS, any register (input or output) can be coded binary. Binary registers are useful if the register is connected to discrete devices in groups of 16 in lieu of numerical devices. The same I/O modules are used for registers (either BCD or binary) as are used for discrete, one module for each register. Registers, either input or output, are wired as shown in Figure 55; power terminals 1, 2, 7, 12, and 17 are connected to reference power depending on the type of I/O module used.

NOTE

On DC modules, terminals 7, 12, and 17 are internally connected to dc common (terminal 2), and do not require external connections.

Output registers that are not used for outputting data can be used to store internal data, along with the remaining holding registers. References to logic lines, discrete inputs, latches, input registers, or holding registers can be made as many times as necessary; there are no limitations on how many times any reference is used.

As can be seen from Table 10, up to 16 input registers and 16 output registers can be provided by standard executives. Figure 56 is a block diagram of the Controller's memory, illustrating how these registers relate to the total system.

Internally, up to 999 holding registers (4001-4999), depending on the executive selected, are available to store numerical data. All holding registers can store up to four digits (maximum value 9999); the value in registers 4001-4016 can also be supplied to the real world and as such are a special type of holding registers, referred to as output registers. Holding registers are inherently retentive on power failure; unless some positive action is taken, they will permanently retain their content.

3.3.2 184 Executive Program (MOPS)

Two groups of MOPS have been developed that provide register I/O; these are MOPS 2 and MOPS 3. The MOPS 1 family has been previously discussed and provides logic lines that simulate the effect of relays, timers, and counters with discrete I/O only. The MOPS 2 family provides, in addition

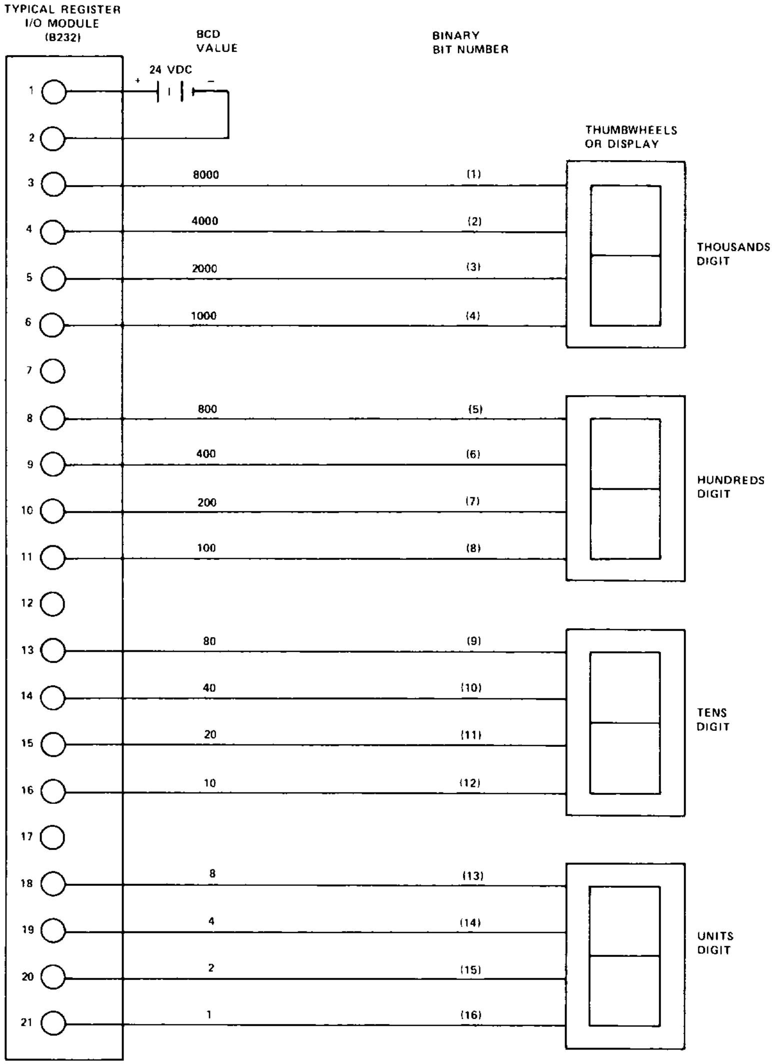


Figure 55. Wiring of Input/Output Registers

to register I/O, the calculate capability and remote set of timers and counters (see Section 3.4) as well as the relay, timing, and counting features of the MOPS 1 level. Finally, the MOPS 3 family provides all the capabilities of the MOPS 2 plus the Data Transfer features (see Section 3.5).

Each MODICON Operation System (MOPS) is designed for a specific size of core memory. This is the minimum core size required; larger core size can accommodate MOPS designed for small core memories.

NOTE

See Appendix F for list of 384 executive programs (TEF).

Table 10. I/O Channels III & IV, Optional Register Assignments

Channel III Index	256 Discrete & 16 Register Input	256 Discrete & 16 Register Output	320 Discrete & 12 Register Input	320 Discrete & 12 Register Output	352 Discrete & 10 Register Input	352 Discrete & 10 Register Output
1	3001	4001	1257-1272	257-272	1257-1272	257-272
2	3002	4002	1273-1288	273-288	1273-1288	273-288
3	3003	4003	1289-1304	289-304	1289-1304	289-304
4	3004	4004	1305-1320	305-320	1305-1320	305-320
5	3005	4005	3001	4001	1321-1336	321-336
6	3006	4006	3002	4002	1337-1352	337-352
7	3007	4007	3003	4003	3001	4001
8	3008	4008	3004	4004	3002	4002
Channel IV Index						
1	3009	4009	3005	4005	3003	4003
2	3010	4010	3006	4006	3004	4004
3	3011	4011	3007	4007	3005	4005
4	3012	4012	3008	4008	3006	4006
5	3013	4013	3009	4009	3007	4007
6	3014	4014	3010	4010	3008	4008
7	3015	4015	3011	4011	3009	4009
8	3016	4016	3012	4012	3010	4010

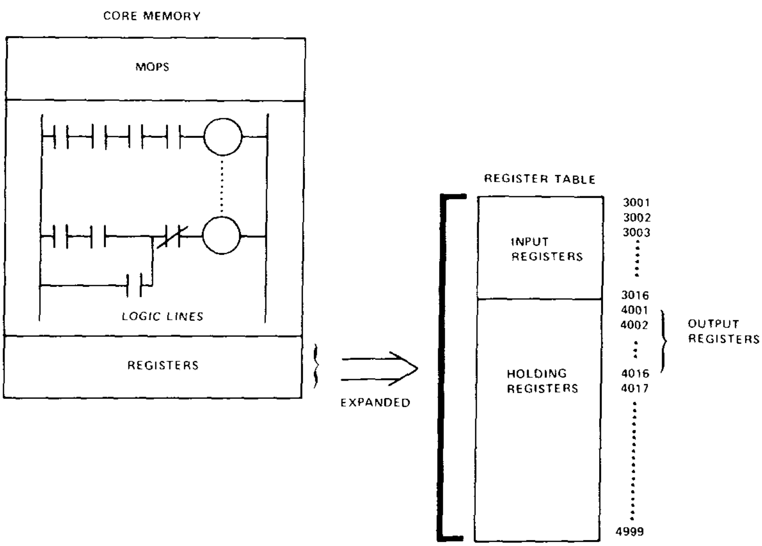


Figure 56. Block Diagram of Core Memory

Each executive resides in core memory and allocates the core memory into areas for storage of the user's logic, registers, and I/O status (see Figure 57). This allocation defines how many logic lines and registers are provided, what type of I/O is assumed, and which lines are latchable. The coil status of all lines are referred to by reference numbers beginning with the digit zero (0xxx). This reference begins at 0001 for the first line and extends to the last line provided by the MOPS. The lower-numbered lines are outputtable and the higher-numbered lines are internal.

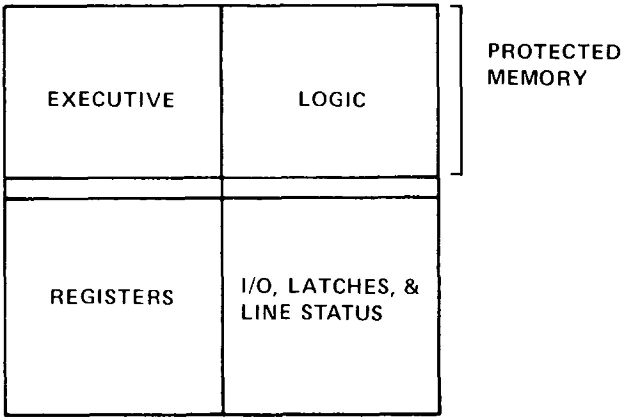


Figure 57. Core Memory Allocation

NOTE

These executive programs are constantly being generated, adding additional capabilities and features. Contact nearest MODICON sales office for latest information.

3.3.3 Additional Programming Panel Features

The Programming Panel (see Figure 25) can be utilized not only to enter logic line data and control disable status of line coils and discrete inputs (see Section 3.1.5), but also provides the ability to monitor other references used in the logic program. The line number thumbwheels are used to determine which reference is being displayed.

At any time, any logic line can be examined by entering the line number (0xxx) on the thumbwheels. The upper half of the selected element pushbutton will light to indicate power flow through relay contacts from left to right; the coil light will be ON if the line's coil is energized. Each element of the line can be examined for proper programming by selecting the element (A, B, C, or D) and observing the reference number display and contact type that is lit. Again, changes can be made only with memory protect OFF; however, monitoring can occur at any time.

If the status of a discrete input is required, the input reference (1xxx) is entered as if it was a line number on the LINE NUMBER thumbwheels. The output coil will be ON if the input is energized, and OFF if the input is de-energized; disregard the element power lights. Similarly, any latch can be examined by entering its reference (2xxx) on the LINE NUMBER thumbwheels and observing the output coil. The contents of the input registers or holding registers can be examined by entering their reference (30xx or 4xxx) on the LINE NUMBER thumbwheels and observing the reference number display. The contents of a register is displayed as a four-BCD-digit magnitude.

If a holding register (4xxx) is being displayed, all four relay contact types will light to indicate that the contents of this register can be altered if desired; input registers (30xx) obtain their contents from the outside world and cannot be altered by the Controller. New contents for a register is entered onto the REFERENCE NUMBER thumbwheels, and any contact type may be depressed to cause this data to be loaded into the holding register.

NOTE

Memory Protect does *not* have to be OFF to change the contents of the holding registers.

If a logic line in the Controller is forcing a number into the register or limiting its value, the manual load of this register will be only temporary; the reference display will indicate the acceptance of the value and then the result of the operation on it by the logic line. The manual load takes place at the end of the scan when the Programming Panel is serviced (see Table 20) and its value remains in the register until the logic line that affects the value is solved.

There are two methods of loading holding registers; one by entering the register reference on the LINE NUMBER thumbwheels as discussed above; the other indirectly through the logic line where the register is referenced, as discussed in Section 3.2.2. In the D element of any non relay line, if decimal points appear to the left of each digit of the reference (total four decimal points), this indicates that the holding register is used in the D element of another logic line. The reference is accepted as a valid reference, but the user is cautioned that the contents of this register may be altered by more than one logic line.

If values are entered on the LINE NUMBER thumbwheels that exceed the limits established by the executive, an error code (⊔22⊔) will be shown on the REFERENCE NUMBER display. Thus, if line number 740 or latch 2512 is entered on the LINE NUMBER thumbwheels, and only 640 lines and 128 latches (2001-2128) are provided, all panel lights will be extinguished and the reference number display will contain a ⊔22⊔.

This feature can be used to determine which executive is in the Controller. Increase the line display by hundreds from zero (e.g., 0100, 0200, 0300, etc.) until a ⊔22⊔ appears; reduce the line number by one hundred to extinguish this display and begin incrementing the line number by tens (e.g., 0600, 0610, 0620, etc.). When the ⊔22⊔ reappears, reduce the line number by ten and begin incrementing by units. When the last valid line is determined, add one to it for the WDT line and this is the number of lines the executive provides. A similar operation can be performed to determine how many holding registers (4xxx) are provided; no incrementing of the result is required similar to that accomplished for the WDT line. Using these two parameters (number of lines and number of registers), available executives in Tables 9, 11, and 12 (for 184 controllers) or Appendix F (for 384 Controllers) can be scanned to determine possible executives. To determine which specific executive is installed, the differences between the possibilities are determined (i.e., location of latches, number of discrete inputs, etc.), and these references examined to determine which set of references are valid.

3.4 REMOTE SET AND CALCULATE CAPABILITY

3.4.1 Remote Set Timers and Counters

Since all MOPS 2, MOPS 3 and TEF level executives are provided with register I/O, their timer or counter logic lines can have their preset adjusted by the contents of either holding or input registers. A typical remote set timer or counter line is shown in Figure 58.

Table 11. Model 184 Controller Executive Program

MOPS 2: Relays, Timers, Counters, and Calculators

Model	No. of Lines	No. of Holding Registers	Discrete I/O (Channel)	Register I/O (Channel)	Latches	Min. Core Size	B+C Coil Option, ON When B=C	Special Functions
1	400	100	256 I 256 O (1,2)	16 I 16 O (3,4)	368 2001- 2368	2K		
2	672	999	256 I 256 O (1,2)	16 I 16 O (3,4)	80 2001- 2080	4K		
3	400	100	256 I 256 O (1,2)	16 I 16 O (3,4)	368 2001- 2368	2K	X	
4	672	999	256 I 256 O (1,2)	16 I 16 O (3,4)	80 2001- 2080	4K	X	
5	672	999	256 I 256 O (1,2)	16 I 16 O (3,4)	80 2593- 2672	4K	X	
6	592	999	352 I 352 O (1,2,3)	8 I 8 O (4)	80 2513- 2592	4K		Seconds Timer Replaced by Down Counter
7	592	999	352 I 352 O (1,2,3)	8 I 8 O (4)	80 2513- 2592	4K	X	Seconds Timer Replaced by Down Counter
8	400	989	256 I 256 O (1,2)	16 I 16 O (3,4)	368 2001- 2368	4K	X	Seconds Timer Replaced by Down Counter
9	800	300	224 I 224 O (1,2)	8 I 8 O (3)	0	4K		
10	672	999	320 I 320 O (1,2,3)	12 I 12 O (3,4)	32 2001- 2032	4K	X	
11	768	300	256 I 256 O (1,2)	8 I 8 O (3)	0	4K	X	K112 Compatible
12	672	999	352 I 352 O (1,2,3)	10 I 10 O (3,4)	0	4K	X	

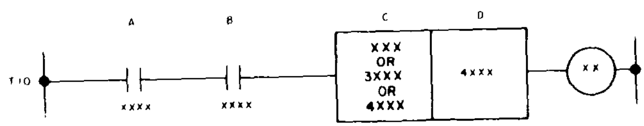


Figure 58. Typical Remote Set

The operation of these logic lines is the same as previously discussed, except for the preset value. The preset in the C element can still be a fixed quantity up to 999; but now it can also be the contents of an input register (30xx) or a holding register (4xxx). Thus, if a preset greater than 999 is required, the value up to 9999 can be placed in a holding register and that register referred to as the preset in the C element of any timer or counter line.

Example of a Remote Set Timer (see Figure 59)

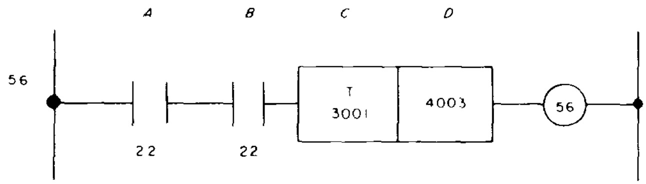


Figure 59. Remote Set Timer Line

1. Set Line Number 56 on LINE NUMBER switches.
2. Select range by pressing TIMER TENTHS or TIMER SECONDS.
3. Press A element pushbutton.
4. Set REFERENCE NUMBER switches to 0022.
5. Press Normally-Open Series ELEMENT TYPE pushbutton.
6. Press B element pushbutton.
7. Leave REFERENCE NUMBER switches set to 0022.
8. Press Normally-Open Series ELEMENT TYPE pushbutton.
9. Press C element pushbutton.
10. Set REFERENCE NUMBER switch to 3001.
11. Press any of the ELEMENT TYPE pushbuttons.
12. Press D element pushbutton.
13. Dial 4003 into REFERENCE NUMBER switches and press any element type.
14. If the DISABLE light is lit, press it to turn it OFF, unless the line is to be left disabled.

Example of a Remote Set Counter (see Figure 60)

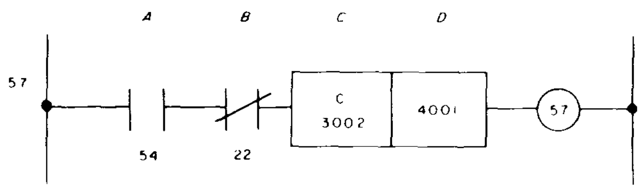


Figure 60. Remote Set Counter Line

1. Set Line Number 57 on LINE NUMBER switches.
2. Press COUNTER pushbutton.
3. Press A element pushbutton.
4. Set REFERENCE NUMBER switches to 0054.
5. Press Normally-Open Series ELEMENT TYPE pushbutton.
6. Press B element pushbutton.

7. Set REFERENCE NUMBER switches to 0022.
8. Press Normally-Closed Series ELEMENT TYPE pushbutton.
9. Press C element pushbutton.
10. Set REFERENCE NUMBER switches to 3002.
11. Press any one of the ELEMENT TYPE pushbuttons.
12. Press D element pushbutton.
13. Dial 4001 on the REFERENCE NUMBER switches and press any element type.
14. If the DISABLE light is lit, press it to turn if OFF, unless the line is to be left disabled.

3.4.2 Addition (B+C) and Subtraction (B−C)

The MOPS 2, MOPS 3, and TEF level executives are all provided with the calculate capability. Any legal line can be converted to a calculate line (B+C or B−C) by selecting the appropriate pushbutton on the Programming Panel. Table 11 lists the currently available executives (MOPS 2) that provide relay, timer, counter, and calculate lines; MOPS 3 programs are listed in Table 12. See Appendix F for list of available 384 TEF Programs.

The usefulness of the holding registers described previously is most apparent when using the Controller's calculate capabilities. Such registers (4xxx) may contain numbers which the user can add to or subtract from each other. Figure 61 is a typical calculate logic line.

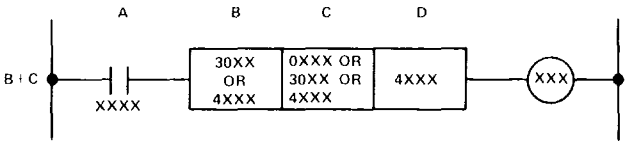


Figure 61. Typical Arithmetic Line

A single series contact (A position) is used to control the operation; this contact may be either normally-open or normally-closed. The B position then requires the entry of a register location (input or holding) — the one to which another number is to be added or from which one is to be subtracted. The C position then requires entry of a fixed number or register (input or holding) location being added or subtracted. The constant value entered into the C position may range from 0 to 999. Finally, in the D position, the holding register (a 4xxx reference) is specified where the result is to be stored.

NOTE

The values in the B and C element register are not altered by the calculate line; the value in the D element register will be replaced with the results of this line. The calculate function is accomplished every scan the A element is closed (passing power).

Again, as with any element having data (box symbol) in a line, no relay information is to be specified; all relay contact types will light and pushing any one will enter thumbwheel data from the reference number into the element selected.

In most cases, the numbers entered in both B and C positions will be 30xx and 4xxx register references rather than real numbers. This makes it much easier and more flexible for handling various functions. The Controller automatically finds the number(s) kept in the specified locations and adds or subtracts them as commanded.

Coil Activation

While the operation of the calculate line — when the A contact is closed — will take place continuously (adding, subtracting, and storing numbers), the energizing of the output coil will only occur selectively.

1. Energize if: in subtraction, the number in B is greater than or equal to the number in C (i.e., result positive, zero being a positive number).

2. Energize if: in addition, the sum of B and C is greater than 9999. (Calculation range has been exceeded.) This is a standard option.

NOTE

If the resultant sum is greater than 10,000 the amount of difference (over 10,000) will be placed in the storage location specified by the D element. For example, if 4999 is added to 6000, then the coil will be energized, and the value 0999 will be stored in the location specified by D.

3. Energize if: in addition, B exactly equals C. This is the B=C coil option (see Table 11, Special Functions). Of course, the addition is still accomplished even if B exactly equals C.

As with other functions, the data stored in the location specified by 4xxx is available for other purposes.

*2 and 3 are mutually-exclusive depending on executive program selected. Any MOPS will have only one B+C coil option. See Tables 11 and 12; unless otherwise indicated, the B+C calculate line will have the standard option.

NOTE

See Appendix F for discussion of B+C coil options available with 384 TEF executives.

Set Points (Compare)

To perform comparison of two numbers, the subtraction capability is useful as shown in Figure 62.

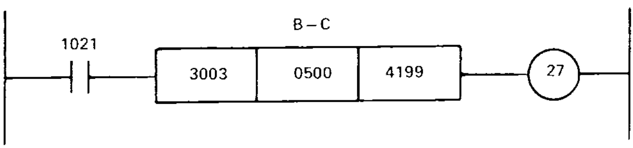


Figure 62. Sample Set Points

The variable data from an input or holding register is addressed in element B, and the set point is loaded into element C. The D element register is loaded with the result of the subtraction and can be used to indicate how close the variable data is to the set point. If the number addressed by element B becomes equal to or greater than the set point in element C, the coil output is energized.

In this example, monitoring begins as soon as input 1021 is energized. With this input energized, whenever the value in input register 3003 becomes equal to or greater than 500 (the set point), coil 27 is energized. The contents of register 4199 represents how close the variable data is to the set point (500).

As many set points as desired can be established on any single input signal, multiple inputs, or internal data, limited only by the number of logic lines available. Set points can be fixed (up to 999), under operator control (input registers up to 9999), or stored internally (holding registers up to 9999). The coil state can be reversed (ON below or at set point and OFF above set point) by placing the set point in the B element register and the variable data in the C element.

Register-to-Register Move

The Calculate B+C logic line can be used as a register-to-register move as illustrated in Figure 63.

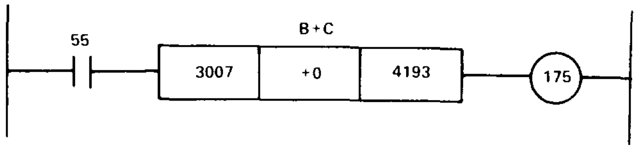


Figure 63. Sample Register to Register Move

Whenever line 55 is energized, the contents of 3007 is copied into register 4193. If line 55 is a one-shot, the contents of 3007 is sampled and held in register 4193 until the next strobing (energizing) of line 55 causes a new sample to be taken. This technique is useful when sampling of register contents is required or if a register is to be loaded with a fixed value stored in the Controller (e.g., forcing pointers to values such as 4, 10, 17, etc.).

Clearing a Register to Zero

The Calculate B−C logic line can be used to clear a register to zero as illustrated in Figure 64.

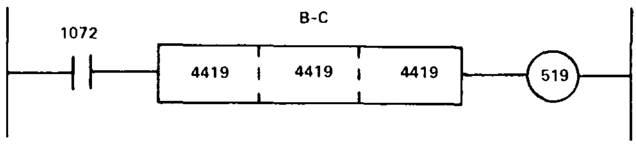


Figure 64. Sample Clearing of a Register

Whenever input 1072 is energized, the contents of 4419 is subtracted from itself, the result will always be zero, and the zero is placed into register 4419. As long as input 1072 is energized, register 4419 will be forced to zero by line 19 every scan; there is no other use for a B−C logic line with the above format. This technique is useful to clear accumulators to zero, force pointers to start of a table, clear displays, etc.

Double-Precision Add

If an executive has the standard coil option for the B+C logic lines (coil ON if sum exeeds 9999), a double-precision add can be developed as shown in Figure 65.

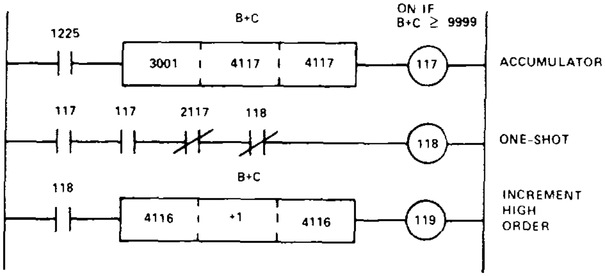


Figure 65. Sample Double-Precision Add

Whenever input 1225 is energized, the contents of 3001 is added to the contents of register 4117, and the sum stored in 4117. The contents of 4117 will continue to accumulate unless cleared to zero by a B−C logic line elsewhere in the logic. If 1225 remains closed for more than one scan, the contents of 3001 will be added to 4117 more than once. Whenever the summation accomplished by line 117 exceeds 9999, the coil is energized and line 118 fires a one-shot. Line 119 adds one to the high-order accumulator (register 4116) whenever the one-shot fires, i.e., whenever line 117 overflows. The one-shot ensures that the high-order accumulator is incremented only once on each overflow; it is not required if the A element reference in the accumulator line (i.e., input 1225) is a one-shot itself, or if continuous adding is accomplished and two successive overflows are possible. When clearing the accumulator to zero, both the high-order (register 4116) and the low-order (register 4117) values must be set to zero.

Detecting Equality with B+C Overflow Coil Option

If an executive is designed with the standard B+C coil option (i.e., coil ON with overflow, NOT when B=C), and detection of exact equality is required, two B−C logic lines can be used as illustrated in Figure 66.

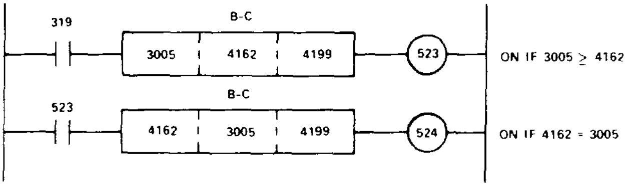


Figure 66. Sample Detection of Equality with B+C Overflow Coil Option

When line 319 is energized, line 523 will subtract the contents of register 4162 from 3005; if the contents of 3005 are equal to or greater than the contents of 4162, coil 523 will be energized and line 524 accomplished. If, and only if, the contents of 4162 exactly equal 3005, will coil 524 come ON; if the contents of 3005 are greater than the contents of 4162, coil 524 will be OFF.

To further illustrate this technique, assume values for the contents of 3005 and 4162 for the three cases: 3005 > 4162, 3005 < 4162, and 3005=4162. Assume the value in 3005 is 100, and the contents of 4162 is 80 (case 1). Line 523 subtracts 80 from 100, resulting in a positive 20 in 4199 and coil 523 is turned ON. Line 524 then subtracts 100 from 80, resulting in a negative 20 in 4199 and coil 524 is OFF. If 3005 contains 100, and 4162 contains 125 (case 2), line 523 subtracts 125 from 100, resulting in a negative 25 in 4199 and coil 523 is OFF. Line 524 is not accomplished and its coil will be OFF. If 3005 contains 100 and 4162 contains 100 (case 3), line 523 subtracts 100 from 100, the result being a positive zero in 4199, and coil 523 is energized. Line 524 subtracts 100 from 100 and the result is also a positive zero in 4199, and coil 524 will be ON. Coil 524 will come ON only if the contents of 3005 exactly equal the contents of 4162. Note that the difference, i.e., how close they are to being equal as an absolute number, is available from 4199.

Detecting Overflow with B+C Equal Coil Option

If an executive is designed with the optional B=C coil status for B+C logic lines, and detection of overflow is required, a B−C logic line and a relay line can provide this signal as illustated in Figure 67.

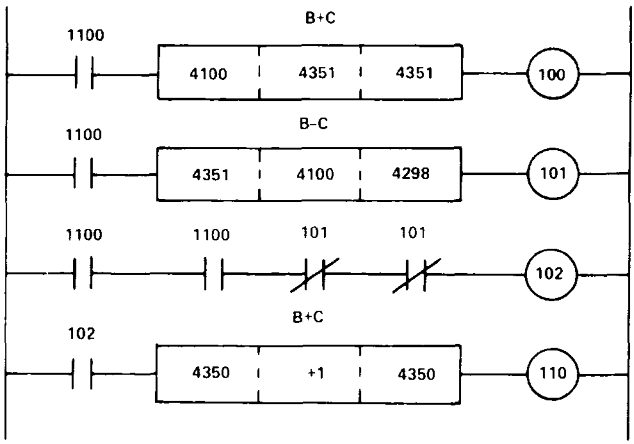


Figure 67. Sample Detection of Overflow with B+C Equal Coil Option

The B+C calculate line (100) is used to accumulate, in 4351, the values in 4100 whenever input 1100 closes. Line 101 checks to ensure the sum is always greater than or equal to the incremented quantity in 4100. If the accumulator (4351) becomes less than the incremental value, i.e., overflow just occurred and the less-significant portion is in 4351, coil 101 will *not* be energized and coil 102 will be energized to indicate overflow. If the accumlator is equal to the incremental value, i.e., accumulation just started and this is the first summation performed after 4351 was reset to zero, coil 101 will be energized and no overflow indicated. The overflow indication can be used to form a double-precision add, as is accomplished by line 110.

Entering a Calculate Line

1. Set the Line Number on the LINE NUMBER switches.
2. Put the Controller MEMORY PROTECT switch in the OFF position.

3. Press either Calculate B+C or Calculate B−C pushbutton as desired. It will light and all other LINE TYPE lights will go out.
4. Press the A element pushbutton. The A element lamp will light and the B, C, and D lights will be OFF.
5. Set the REFERENCE NUMBER thumbwheel switches to the line number, input number, or latch that is to operate the contact in the A position.
6. Press either the Normally-Open or Normally-Closed Series ELEMENT TYPE pushbutton. The ELEMENT TYPE pushbutton that is pressed will light and the REFERENCE NUMBER display will show the number that has been entered.
7. Select the B element position. It will light and the A will go out, and all four ELEMENT TYPE pushbuttons will light.
8. Enter, on the REFERENCE NUMBER switches, the data location for the B element. Depress any contact type pushbutton.
9. Repeat steps 7 and 8 for the C and D positions.

NOTE

Either an input register (30xx), or holding register (4xxx) can be used in either the B or C element. In addition, a constant value of from 0 to 999 may be entered in the C element rather than a remote data (register) location number. A holding register (4xxx) must be placed in the D element.

10. If the DISABLE pushbutton is lit, and not specifically desired, press it to turn if OFF.

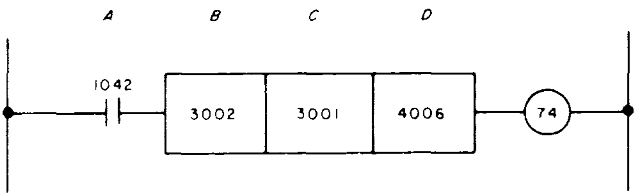
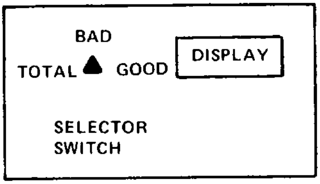


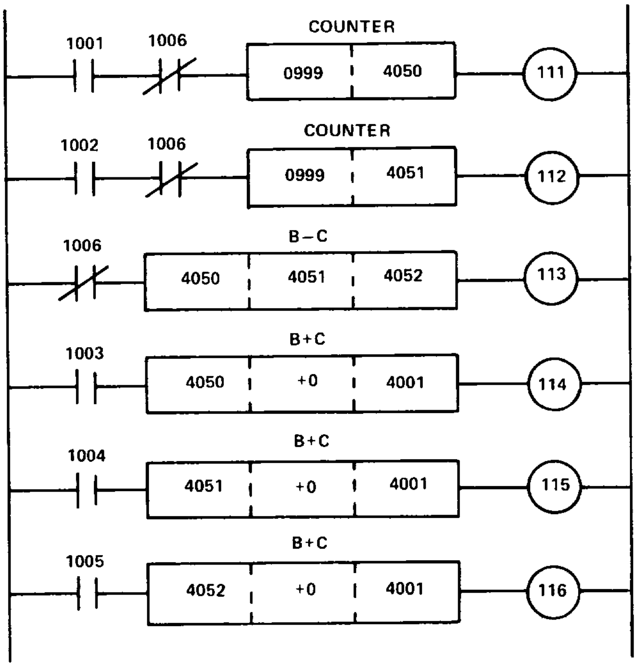
Figure 68. Calculate B+C Line

Entry of a B+C Calculate (Figure 68)

1. Set Line Number switches to 0074.
2. Press Calculate B+C pushbutton.
3. Press A element pushbutton.
4. Set REFERENCE NUMBER switches to 1042.
5. Press Normally-Open Series ELEMENT TYPE pushbutton.
6. Press B element pushbutton.
7. Set REFERENCE NUMBER switches to 3002. Press any ELEMENT TYPE pushbutton.
8. Repeat steps 6 and 7 for C element 3001, and D element 4006.
9. If the DISABLE line is lit, and not specifically desired, press it to turn it OFF.



OPERATOR'S PANEL



<u>INPUT ASSIGNMENTS</u>

1001 = PARTS PRODUCED
1002 = BAD PARTS DETECTED
1003 = DISPLAY TOTAL PARTS
1004 = DISPLAY BAD PARTS
1005 = DISPLAY GOOD PARTS
1006 = CLEAR ALL COUNTS TO ZERO

<u>REGISTER UTILIZATION</u>

4001 = DISPLAY

4050 = # OF TOTAL PARTS
4051 = # OF BAD PARTS
4052 = # OF GOOD PARTS

Figure 69. Example of Time Shared Display

EXAMPLE III. Time-Shared Display

As a review of the calculate capability, a method of time-sharing an output register that is connected to a display can be developed (see Figure 69). This will allow displaying many groups of data via a single output register. Assume input 1001 is energized every time a part is processed by a machine; input 1002 is energized every time a bad part is detected. The operator must be able to select either total parts, good parts, or bad parts for displaying; only one parameter will be displayed at a time.

Line 111 counts the number of total parts and line 112 counts the number of bad parts. Line 113 takes the difference between total parts and bad parts and places the result (good parts) into register 4052. When the selector switch is placed in the total position, input 1003 is energized, causing line 114 to move the current total count (stored in register 4050) into the display, driven from register 4001. When the selector switch is placed in the bad position, input 1003 is de-energized, and input 1004 is energized. This input causes line 115 to move the number of bad parts stored in register 4051 into the display via register 4001; line 114 will not move any data since input 1003 is not energized. Similarly, line 116 moves the number of good parts from register 4052 into the display.

If additional data is to be displayed (e.g., machine-up time), another position of the selector switch must be provided, with another input, and another calculate line similar to lines 114-116. Additional logic (e.g., a timer) must be included to develop the additional parameter to be measured. For other more sophisticated methods of driving display, see EXAMPLE IV.

3.5 DATA TRANSFER

In addition to the usual relay contact circuits described previously, and the counting, timing, and arithmetic functions, the storage location concept allows the MODICON 184/384 Controller another versatile capability: the manipulation of the contents of such storage registers, either in whole or in part, under the control of other logic. These data transfer functions are still controlled by relay logic.

The data transfer function capability provides the designer with the ability to move or transfer data in large blocks, perform extended arithmetic operations, and bit manipulations within the Controller utilizing only one line of logic. This capability is provided in the 184 Controller only by MOPS 3 level executive program (see Table 12) written for 4K memory systems. At present, the data transfer (DX) capability is divided into four groups:

Code	**Function**
1YXX	Move
2YXX	Matrix Handling
3YXX	Extended Arithmetic
4YXX	Printer

Each function group is provided as software (programmed) subroutines within the executive. All executive programs at the MOPS 3 level include various combinations of these function groups; the Move capability is very basic and is included in all MOPS 3 executives.

Since executives with data transfer functions occupy a portion of the available core memory and are larger than the less sophisticated MOPS, the quantity of logic lines and holding registers can be somewhat reduced. However, the reduced number of lines resulting from a larger executive are more than compensated for by the added power capability of the DX lines.

NOTE

See Section 3.6 for complete discussion of the Printer DX line function.

NOTE

All 384 Controllers are provided with all DX capabilities except PID without effect on logic line or register quantity. See Appendix F.

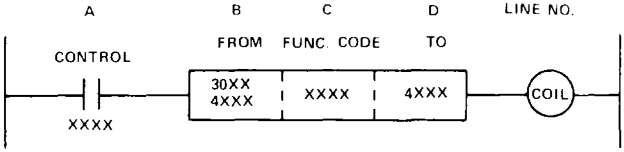


Figure 70. Typical DX Line

A typical Data Transfer (DX) logic line is shown in Figure 70. The contact in the A element activates the DX line; no operation is performed until this contact is closed (passing power through the A element). Either series-open or series-closed contacts, with any legal discrete reference can be utilized in the A element. The B element specifies where the data is to be obtained; any valid input register, output register, or holding register can be utilized in the B element. Data is copied from the location specified in the B element; the contents of the B element register are not altered.

The C element is a functional code; it is *not* a register, storage address, latch, or input. Valid functional codes are provided below as part of the discussion of each functional group.

The D element register specifies where the data or action is to be received or take place. Any valid output or holding register can be utilized in the D element. Since the data transfer into the register specified in the D element is destructive (i.e., the old data is lost and the new data retained), input registers whose contents are controlled by an external device cannot be used in the D element. The status of the coil varies within the functional groups and thus will be discussed separately as part of each group. These coil status are important in the design of supporting relay logic control.

Registers in the B and D elements can refer by inference to more than one register. In these cases, the additional registers will be in sequence following the register specified by the DX line. For example, registers 3001, 3002, and 3003 can be referred to by the appropriate functional code when register 3001 is specified in the B element. How many registers and under what conditions they are referred to is discussed as part of each functional code. The above discussion should be assumed to be applicable to each functional code unless specifically exempted.

It is important that these general concepts be understood prior to discussing the specific functional codes. Since all registers affected by the DX function are not always referred to in the logic line, it is desirable to prepare a map of register utilization to ensure the overlapping of registers does not occur unless desired by the user.

3.5.1 MOVE (Group 1YXX)

This function group allows tables of data to be built in consecutively numbered registers. The length of the table is always specified in the last two digits (XX) of the functional code (C element); thus, the maximum length of any table is 99 registers. Data can be moved from a register to a table, from a table to a register, or from a table to a table.

NOTE

Minimum table length is two registers, except 384 TEF programs which allow function codes 1001 and 1101 — see Appendix F.

Table 12. Model 184 Controller Executive Programs

MOPS 3: Relays, Timers, Counters, Calculators, and DX Functions

Model	No. of Lines	No. of Holding Registers	Discrete I/O (Channel)	Register I/O (Channel)	Latches	B+C Coil Option, ON When B=C	Move	Basic Matrix	Mult./ Divide	P500 Printer	Special Functions
4	640	931	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2001- 2128		X				
8	512	900	256 I 256 O (1,2)	16 I 16 O (3,4)	240 2001- 2240		X			X	
9	704	240	256 I 256 O (1,2)	16 I 16 O (3,4)	64 2001- 2064		X			X	Extended Sweep
10	640	795	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2001- 2128		X	X			
11	608	300	256 I 256 O (1,2)	16 I 16 O (3,4)	160 2449- 2608	X	X			X	
12	592	865	256 I 256 O (1,2)	16 I 16 O (3,4)	176 2001- 2176	X	X		X	X	
13	432	999	256 I 256 O (1,2)	16 I 16 O (3,4)	336 2001- 2336	X	X		X	X	
14	496	500	256 I 256 O (1,2)	16 I 16 O (3,4)	272 2001- 2272		X			X	
15	592	841	352 I 352 O (1,2,3)	10 I 10 O (3,4)	80 2513- 2592	X	X		X		
16	640	795	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2513- 2640	X	X	X			

Table 12. Model 184 Controller Executive Programs (continued)

MOPS 3: Relays, Timers, Counters, Calculators, and DX Functions

Model	No. of Lines	No. of Holding Registers	Discrete I/O (Channel)	Register I/O (Channel)	Latches	B+C Coil Option, ON When B=C	Move	Basic Matrix	Mult./ Divide	P500 Printer	Special Functions
17	640	931	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2001- 2128	X	X				Extended Sweep
18	608	300	320 I 320 O (1,2,3)	12 I 12 O (3,4)	96 2513- 2608	X	X			X	K112 Compatible
20	512	950	256 I 256 O (1,2)	16 I 16 O (3,4)	256 2001- 2256	X	X	X	X		
21	592	486	256 I 256 O (1,2)	16 I 16 O (3,4)	176 2001- 2176	X	X	X	X		Extended Sweep
22	720	700	256 I 256 O (1,2)	16 I 16 O (3,4)	48 2001- 2048	X	X				
23	688	624	320 I 320 O (1,2,3)	12 I 12 O (3,4)	16 2001- 2016	X	X		X		
24	640	795	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2001- 2128		X	X			Extended Sweep
25	608	788	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2481- 2608		X	X			Improved Matrix Additional 30XX References
26	704	195	256 I 256 O (1,2)	16 I 16 O (3,4)	64 2225- 2288	X	X		X	X	
27	624	300	352 I 352 O (1,2,3)	10 I 10 O (3,4)	48 2577- 2624	X	X			X	K112 Compatible

Table 12. Model 184 Controller Executive Programs (continued)

MOPS 3: Relays, Timers, Counters, Calculators, and DX Functions

Model	No. of Lines	No. of Holding Registers	Discrete I/O (Channel)	Register I/O (Channel)	Latches	B+C Coil Option, ON When B=C	Move	Basic Matrix	Mult./ Divide	P500 Printer	Special Functions
28	512	930	256 I 256 O (1,2)	16 I 16 O (3,4)	256 2001- 2256	X	X	X	X		
29	640	795	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2001- 2128		X	X			Additional 30XX References
30	720	689	256 I 256 O (1,2)	16 I 16 O (3,4)	48 2001- 2048	X	X				Extended Sweep
31	640	795	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2001- 2128		X	X			Extended Sweep Additional 30XX References
32	464	700	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2001- 2128	X	X	X		X	Improved Matrix Extended Sweep Additional 30XX References
33	320	950	256 I 256 O (1,2)	16 I 16 O (3,4)	320 2001- 2320	X	X		X	X	Extended Sweep PID
34	800	439	224 I 256 O (1,2)	16 I 16 O (3,4)	0	X	X				Extended Sweep
35	432	999	256 I 256 O (1,2)	16 I 16 O (3,4)	336 2001 - 2336		X		X	X	
36	416	999	256 I 256 O (1,2)	16 I 16 O (3,4)	352 2001 - 2352	X	X		X	X	Checksum
37	512	900	256 I 256 O (1,2)	16 I 16 O (3,4)	240 2001 - 2240		X			X	Extended Sweep

Table 12. Model 184 Controller Executive Programs (continued)

Model	No. of Lines	No. of Holding Registers	Discrete I/O (Channel)	Register I/O (Channel)	Latches	B+C Coil Option, ON When B=C	Move	Basic Matrix	Mult./ Divide	P500 Printer	Special Functions
38	432	989	256 I 256 O (1,2)	16 I 16 O (3,4)	336 2001 - 2336	X	X		X	X	K112 Compatible
39	592	850	352 I 352 O (1,2,3)	10 I 10 O (3,4)	80 2001 - 2080	X	X	X			Improved Matrix Standard I/O is 256 discrete, 16 register.
40	640	795	256 I 256 O (1,2)	16 I 16 O (3,4)	128 2001 - 2128	X	X	X			
41	592	513	256 I 256 O (1,2)	16 I 16 O (3,4)	176 2001 - 2176	X	X	X	X		Additional 30XX References
42	512	944	256 I 256 O (1,2)	16 I 16 O (3,4)	256 2257 - 2512	X	X	X	X		Improved Matrix Additional 30XX References
43	288	950	256 I 256 O (1,2)	16 I 16 O (3,4)	288 2001 - 2288	X	X		X	X	PID, Guarded Lines
44	704	795	320 I 320 O (1,2,3)	12 I 12 O (3,4)	0	X	X	X			
45	400	951	256 I 256 O (1,2)	16 I 16 O (3,4)	240 2001 - 2240	X	X	X	X		Sequencer (DX Code 29XX), 128 steps (2241-2368)

NOTES: 1. All MOPS 3 executives require 4K core size.
2. Executive programs marked "Extended Sweep" allow the processor to have a scan time greater than 200 ms and service I/O twice a scan

In all functional codes, except 16XX, the register specified in the D element will contain a pointer indicating where in the table(s) the *last* operation occurred. If this pointer is a zero, no operation has occurred; if it is a one, the first register in the table has been operated on and the next operation will be accomplished on the second register. The contents of the pointer will be automatically incremented by one at the completion of an individual move.

The coil will be energized when the A element is closed (passing power) *and* the contents of the pointer register equals the length of the table as specified in the functional code. When the A element is opened (not passing power) *after* the coil has been energized, the coil will be de-energized and the pointer set to zero.

Since the pointer is stored in a register, it can be altered by other logic lines, such as calculate lines, to force the move to occur on specific registers within the table without operating on previous registers. To accomplish this, the pointer should be forced to the desired location in the table *minus* one.

For example, if the operation is to be done on the fifth register of a table, the pointer should be forced to four. If the pointer is forced to a value equal to or greater than the table length, and the A element is closed, the coil will come ON and no action will take place, except to force the pointer to the maximum length of the table.

The following discussion assumes that no operation other than the move is performed on each table. In some cases, a register-to-register move without pointer is desired. In these cases, a calculate (B+C) line is used where the B element contains the source of the data, the C element a zero, and the D element the receiving location.

NOTE

Do not use a calculate line to move data whose magnitude is greater than 9999 (e.g., binary information), unless the 384 Binary B+C option is selected — See Appendix F.

10XX — Table-to-Register Move (Incremental)

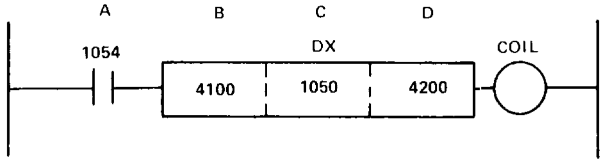
NOTE

The table-to-register moves (10XX and 11XX) are useful to change the preset on timer/counter lines as the operation sequences through finite steps, drive devices in groups of 16 wired to output registers, search a table by its contents, or to simulate a drum programmer.

This code causes one register in a table of registers to be copied into a specified register upon closure of the A element contact. To transfer the next register in the table, the A element contact must be opened and then closed again. As an example, refer to Figure 71.

Assume that input 1054 is not energized and register 4200 contains a zero. On the first scan that input 1054 is energized, the contents of register 4100 will be transferred to register 4201 and the contents of register 4200 incremented by one to 1. The values in the table (4100- 4149) will not be altered; however, the previous contents of 4201 will be lost.

On subsequent scans, no operation will be performed until 1054 is de-energized and then re-energized. If register 4200 contained the number 49 and 1054 was energized, the contents of register 4149 would be copied into register 4201, the contents of register 4200 would be incremented to 50, and the coil would be ON. The coil would remain ON until 1054 is de-energized, at which time register 4200 would be cleared to zero and the coil turned OFF; the contents of 4201 are not altered when the pointer (4200) is cleared to zero. Note that the B element reference (register 4100) actually refers to 50 sequential registers (4100-4149) since the table length is coded in the function code as 50, and the D element reference (register 4200) refers to two sequential registers (4200 and 4201).



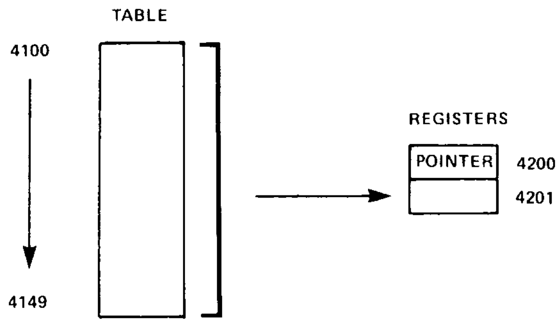


Figure 71. Sample Table to Register Move

11XX — Table-To-Register Move (Continuous)

This code causes one register in a table of registers to be copied into a specific register at the rate of one register per scan as long as the A element contact is closed (passing power). The operation of this function is very similar to code 10XX discussed above, except that the A element does not have to be cycled ON-OFF-ON.

For example, assume that Figure 71 has a functional code of 1150 and a 32 in register 4200. On the first scan that input 1054 is energized, register 4132 will be copied into register 4201 and the contents of register 4200 incremented by one to 33. On the next scan, assuming input 1054 remains energized, register 4133 will be copied into register 4201 and the contents of register 4200 incremented to 34. This operation will continue until either input 1054 is de-energized or the end of the table is reached. If, after 10 scans, input 1054 is de-energized, the number 42 will be in register 4200 and register 4201 will contain the contents of register 4141. When input 1054 is re-energized, the move will commence from where it was and copy register 4142 into 4201, incrementing the contents of 4200 to 43.

Once the end of the table is reached, register 4200 will contain a 50, the contents of register 4201 will be the same as register 4149, and the coil will be ON. Only after input 1054 is de-energized will the coil be OFF and register 4200 cleared to zero; register 4201 will still contain the contents of register 4149 until another move is performed.

12XX — Register-to-Table Move (Incremental)

NOTE

The register-to-table moves (12XX and 13XX) are useful to load tables with new data, retain multiplexed input data, or store error information for future use.

This code causes the contents of one register to be copied into a table of registers upon closure of the A element contact. To load the next register of the table in sequence, the A element contact must be opened and closed again. As an example, refer to Figure 72.

Assume that input 1034 is not energized and register 4001 contains the number 11. On the first scan that input 1034 is energized, the contents of register 3001 will be copied into register 4013 and the contents of register 4001 will be incremented by one to 12.

On subsequent scans, no operation will be performed until 1034 is de-energized and then re-energized. If register 4001 contained the number 14 and 1034 was energized, the contents of register 3001 would be copied into register 4016, the contents of register 4001 would be incremented by one to 15, and the coil would come ON. The coil will remain ON until output 1034 is de-energized, at which time register 4001 would be cleared to zero, and the coil turned OFF.

Note that the B element refers to only one register and the D element refers to 16 sequential registers, one for the pointer and the table of length 15 immediately following the pointer.

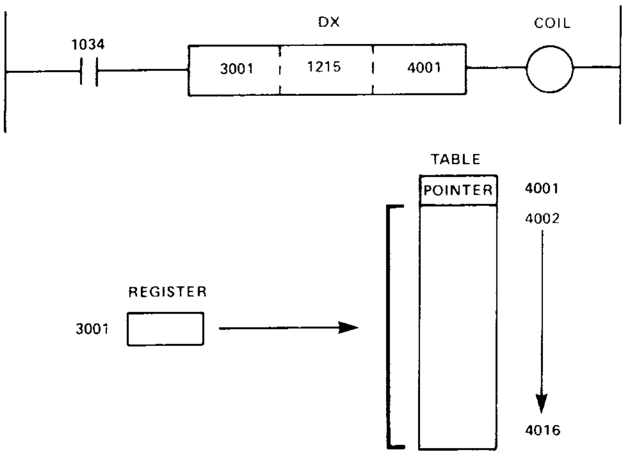


Figure 72. Sample Register to Table Move

13XX — Register-to-Table Move (Continuous)

This code causes a register to be copied into a table of registers at the rate of one register per scan as long as the A element contact is closed (passing power). The operation of this function is very similar to code 12XX discussed above except that the A element contact does not have to be cycled ON-OFF-ON.

For example, assume that Figure 72 has a functional code of 1315 and a 5 in register 4001. On the first scan that input 1034 is energized, register 3001 will be copied into register 4007 and the contents of the register 4001 is incremented to 6.

This operation will continue loading the table from register 3001 until either input 1034 is de-energized to halt operation where it is, or the end of the table is reached. Once the end of the table is reached, register 4001 will contain a 15, register 3001 would have been copied into register 4016, and the coil will be ON. No further moves are possible until input 1034 is de-energized to clear register 4001 to zero and turn the coil OFF.

14XX — Table-to-Table Move (Incremental)

NOTE

The table-to-table moves (14XX and 17XX) are useful to load working registers with various recipe-type data or drive outputs in multiple groups of 16 devices, each group wired to a register output.

This code causes the contents of a register in a table of registers to be copied into a corresponding register of another table upon closure of the A element contact. To transfer the next register in the table, the A element must be opened and then closed again. As an example, refer to Figure 73.

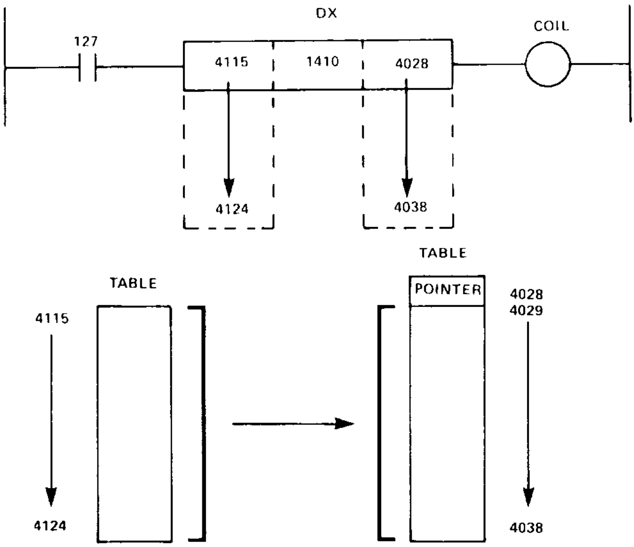


Figure 73. Sample Table to Table Move

NOTE

In Figure 73, the tables are symbolically appended to the DX line, elements B and D. Many designers use this technique to document DX moves. However, it is to be emphasized that only the first register of each table is actually entered into the DX line.

Assume that line 127 is not energized and register 4028 contains the number 2. On the first scan that line 127 is energized, the contents of the third register of the table starting at 4115 (i.e., register 4117) will be copied into the third register of the table starting at 4029 (i.e., register 4031) and the contents of register 4028 will be incremented by one to 3. Since there is only one pointer register, transfers will always take place into table locations with the same element number as the source.

If it was desired that the transfer be offset in the example of Figure 73 so that the third element of table 4115 be transferred into the first element of table 4029, the B element register should be changed to 4117 and the functional code changed to 1408 (table length of eight). A similar alteration of the receiver table is not possible without additional operations, since the pointer location would also change. Thus, if it were desired to transfer the first element of table 4115 into the third element of table 4029, a functional code of 1408 and D element register of 4030 would result in the pointer (value 0-8) being written over the second element of table 4028. In this case, the contents of 4030 should be saved before the move and restored after the move is completed.

Note that the B element refers to 10 sequential registers and the D element to 11 sequential registers where a functional code of 1410 is utilized. The coil will be energized when the pointer equals the specified table length and the A element contact is closed (passing power). The pointer is cleared to zero and the coil de-energized after it was energized if the A element contact is opened when the end of the table is reached.

17XX — Table-to-Table Move (Continuous)

This code causes the content of one register in a table of registers to be copied into the corresponding register of another table at the rate of one register per scan as long as the A element contact is closed (passing power). The operation of this function is very similar to code 14XX discussed above except that the A element does not have to be cycled ON-OFF-ON.

For example, assume that Figure 73 has a functional code of 1710 and a four in register 4028. On the first scan that line 127 is energized, register 4119 will be copied into 4033 and the contents of register 4028 incremented by one to 5. On the next scan, assuming line 127 remains energized, register 4120 will be copied into register 4034 and the contents of register 4028 incremented to 6. This operation will continue until either line 127 is de-energized or the end of the table is reached. Once the end of the table is reached, register 4028 will contain a 10; table 4115 will be completely copied into table 4029, and the coil will be ON. Only after line 127 is de-energized will the coil be OFF and register 4028 cleared to zero.

15XX — First In, First Out (FIFO) Load

NOTE

The FIFO moves (15XX and 16XX) are useful to temporarily store data that may occur in large groups and provide it in a slower more continuous rate or to move data associated with equipment synchronized to the external movement of a conveyor or tranfer system (e.g., a 99-stage shift register with 16 bits available in each stage).

This code causes the contents of a register to be copied into the last available register of a table when the A element contact is closed; the new data is thus immediately stacked above any existing data. For subsequent moves, the A element contact must be opened and then closed again. The coil will come ON after the last available register in the table is utilized (table full) and will be OFF only after data is removed from the table by 16XX function; the coil status is not affected by the condition of the A element contact. For example, refer to Figure 74, line 316.

Assume that line 275 is not energized and there are still three previous entries in the FIFO stack; thus the pointing register (4100) will contain the number 3. On the first scan that line 275 is energized, the contents of register 4011 will be copied into the FIFO table immediately above the existing thee entries, thus into register 4117 (4100 + 20 − 3) and the contents of register 4100 will be incremented by one to indicate four valid entries now in the table.

The data in register 4011 will be retained and the previous contents of register 4117 destroyed. If register 4100 contained the number 19 before the move, after the move the coil for line 316 would be ON and the number 20 would be in register 4100. The coil would remain ON regardless of the condition of the A element contact and all future moves into this table would be ignored until the contents of register 4100 was reduced to less than 20.

Note that the B element reference is to a single register and that the D element is to 21 registers.

16XX — First In, First Out (FIFO) Remove

This code, upon closure of the A element contact, causes the contents of the last register in a table to be copied into a specific register, and the contents of all remaining registers in the table containing valid data are moved down by one to their next registers. For subsequent moves, the A element contact must be opened and then closed again. The coil will come ON after the last valid entry in the table is removed (table empty) and will be OFF only after data is entered into the table by a 15XX function; the coil status is not affected by the condition of the A element contact. For example, refer to Figure 74, line 317.

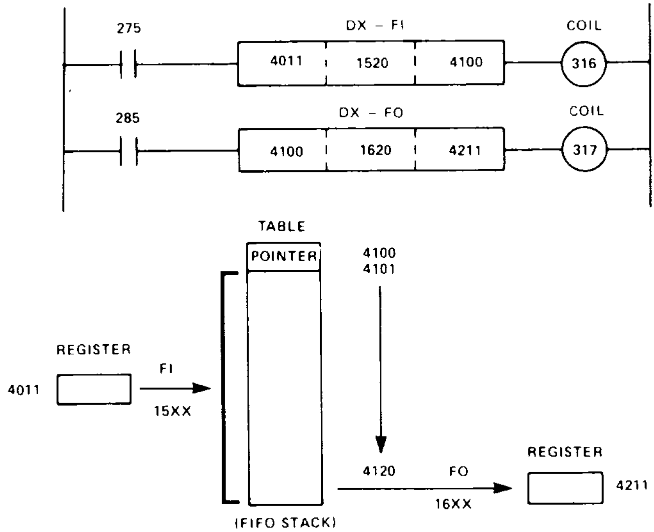


Figure 74. Sample FIFO Stack

Assume that line 285 is not energized and register 4100 contains a 7, indicating that there are seven valid entries in the FIFO stack. On the first scan that line 285 is energized, the contents of register 4120 will be transferred to register 4211, the six remaining valid registers will be copied into their following registers (i.e., 4119 into 4120, 4118 into 4119, etc.), and register 4100 will be decremented by one to 6. If register 4100 contained a one before the move, after the move the coil for line 317 would be ON regardless of the condition of the A element contact and all future moves out of this table would be ignored, until the contents of register 4100 was increased above zero.

Note that the B element refers to 21 registers and the D element to 1. This function is the only one wherein the B element reference is to a pointer. Since the pointer reference keeps track of how many valid registers there are in the FIFO stack, it must be shared between the 15XX and 16XX functional codes.

When data is removed from the FIFO stack, the uppermost register containing valid data is copied into a lower register, but its contents will remain in the previous register. Since the pointer is decremented by one, the uppermost register that contained valid data, and now contains invalid data, will be rewritten on the next load (15XX) move.

NOTE

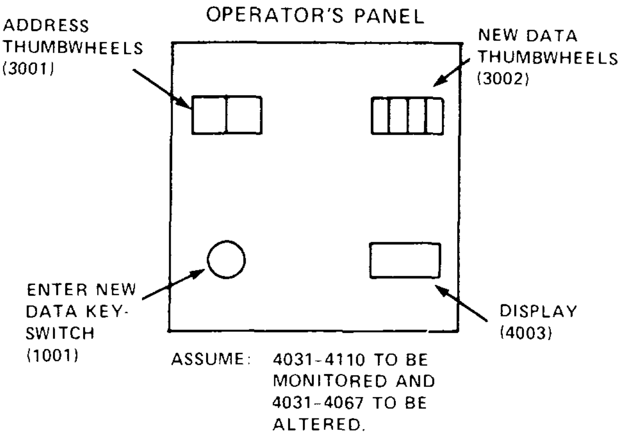
Function code 17XX is discussed after function code 14XX.

NOTE

Function codes 18XX and 19XX (384A and 384B only) are discussed in section 3.7.

EXAMPLE IV — Operator Monitor and Change of Registers

In many cases, the operator must be provided with the capability of monitoring and possible altering the contents of a large number of registers within the Controller without using the Programming Panel. These registers can be presets on timer/counter lines, current time or counts, set points for compares, recipe data, etc. The quantity of registers that are to be altered can be the same as or fewer than the monitored registers. A typical operator's panel and logic lines are shown in Figure 75.



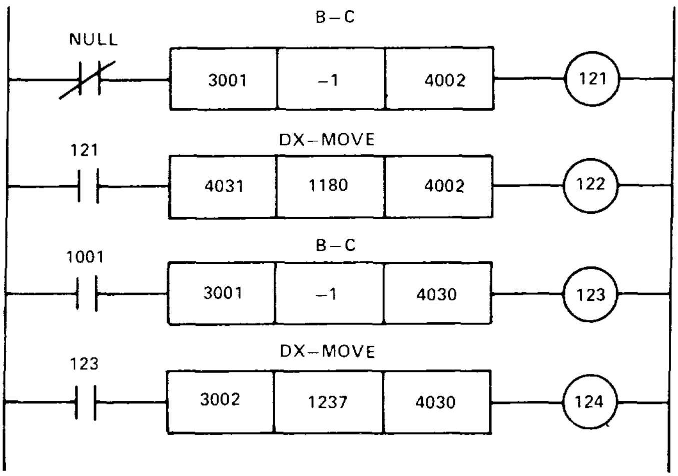


Figure 75. Example: Operator Monitor and Change of Registers

The operator's panel contains two sets of thumbwheels; in this example one set of two digits (0-99) wired to input register 3001 and the other of four digits (0-9999) wired to input register 3002. A single BCD display of four digits is wired to output register 4003, and a keylock switch to provide security is wired to discrete input 1001. If more than 99 registers are to be monitored, the thumbwheels connected to 3001 can be increased to three digits (0-999).

In this example, numerical data is placed in registers 4031-4110. If these registers contained binary status used to drive outputs, or with matrices, input register 3002 should be coded binary and connected to 16 separate toggle switches; output register 4003 should also be coded binary and connected to 16 separate status lamps.

Since the A element of line 121 is referenced to a coil that is never energized (i.e., null data relay line, input that is not wired up, etc.), line 121 always obtains the contents of 3001, subtracts 1 from it, and forces the pointer register 4002 to that value. Line 122, every scan that the A element is closed, obtains from the table of length 80 (starting with 4031), the element referred to by the pointer and places it in 4003 for display. Line 122 is inhibited from progressing through the table at one element per scan as it normally would, since line 121 is always forcing the pointer back to its required value. However, every time a new value is entered into 3001, the corresponding element is automatically extracted from the table; no action is required by the operator, other than to change the value on the thumbwheel connected to 3001. If the thumbwheels are set to zero, no action is desired since there are no elements in the table with the number or address of zero. Line 121 will still take the zero on 3001, subtract one from it, and place the result (a 1) into 4002. However, since this is a *negative* one, its coil does not come ON and no move is performed by line 122; whatever value was in the display remains there. If the value on 3001 exceeds 80 (the length of the table which the operator is allowed to monitor), no moves are performed since the pointer will be forced to 80 or greater, and line 122 is limited by the DX code to 80 registers.

Altering the data is performed in a similar manner. The operator enters on 3001 the element he would like to change, views its current contents on 4003, enters the new value on 3002, and closes the keylock switch. If he does not have a key, he cannot make changes. When the keylock switch (input 1001) is closed, line 123 forces the pointer and line 124 moves the data from 3002 into the table; the display will automatically verify entry of the new data. Again, entry of data into element zero is prevented by coil 123 used in the A element of line 124 and entry of data into registers beyond 4067 (table length 37) is inhibited by the DX function code.

This monitoring capability requires only four logic lines and provides monitoring/altering capability for up to 99 consecutive registers. If more than 99 registers are required, additional capacity can be added at the rate of five logic lines per additional 99 registers or fraction thereof. This monitoring scheme does not in any way affect the use of these registers as presets (C element) in timers/counters, set points (C element) of calculate lines, current times or counts (D element), etc. They can and should be used by other logic lines in the program as required. To be fully confident in this example, select some values between 1 and 80 and test the effects of these logic lines.

Compare the efficiency of this method using Data Transfer with Example III where calculate lines are utilized. These four logic lines using DX provide monitoring and altering of up to 99 registers; 99 calculate lines provide monitoring only of 99 registers.

3.5.2 Matrix Handling (Group 2YXX)

These codes provide the capability to manipulate data as binary bits, either in large segments or as individual bits. The following functions are available with this group: