SECTION III
BASIC CONTROL DESIGN

This section includes the following information:

1. Important machine concepts which, without describing detailed internal circuitry or design, will enable the reader to better understand subsequent descriptions of the MODICON 184/384 Controller functions.
2. A brief description of the physical actions using the Programming Panel required for entering and altering stored logic and data.
3. A detailed discussion of each of the three major functions (basic logic, calculate, and data transfer) available with the MODICON 184/384 Controller (depending on MOPS/TEF options). These discussions include
   a. Detailed description of how the function operates within the Controller configuration.
   b. Step-by-step instructions on how to enter each line type using the Programming Panel.
   c. Several examples of entry of typical ladder diagram logic for the function.
   d. Complete example illustrating an application of the logic just discussed. If these examples are understood, comprehension of the subject matter can be assured.
4. A discussion of the P500 Printer option, in detail, including its various character and control formats.
5. A detailed discussion of new Data Transfer capabilities available only with 384A Controllers.

3.1 IMPORTANT MACHINE CONCEPTS

3.1.1 Controller Reference Numbers

Throughout the programming of any Model 184/384 Controller, four-digit reference numbers are utilized to build the user's logic. These references are divided into two broad categories: discretes and registers. Discrete references are used for individual items that can be either ON or OFF, such as limit switches, pushbuttons, relay contacts, etc. Register references are used to store numerical values such as counts or times; all register references are four BCD digits long (maximum value 9999). Since there are four bits per BCD digit, registers can also be 16 bits of data.

Only five types of references are required to program the 184/384 Controller. Any specific reference can be used as many times as required by the particular application; there are no limitations on the number of times a reference is used. References are defined as follows:

0xxx — logic line numbers/discrete outputs
1xxx — discrete inputs
2xxx — latches
30xx — input register
4xxx — holding register/output register

The address index previously discussed as part of the I/O configuration is very important in establishing proper references. A typical I/O allocation as shown in Table 7, provides 352 discrete inputs and 352 discrete outputs; the maximum number of inputs and outputs can vary as discussed in Section 3.3.2. The output module placed in channel I and indexed as number 1, will output (or copy) the status of the coils associated with logic lines 1-16; output module 2 in channel I will reflect the status of lines 17-32, etc. Inputs are numbered starting with input 1001.

Table 7. Typical I/O Configuration — 352 Discrete I/O

	SLOT	INPUT	OUTPUT
CHAN. I	1	1001—1016	0001—0016
	2	1017—1032	0017—0032
	3	1033—1048	0033—0048
	4	1049—1064	0049—0064
	5	1065—1080	0065—0080
	6	1081—1096	0081—0096
	7	1097—1112	0097—0112
	8	1113—1128	0113—0128
CHAN. II	1	1129—1144	0129—0144
	2	1145—1160	0145—0160
	3	1161—1176	0161—0176
	4	1177—1192	0177—0192
	5	1193—1208	0193—0208
	6	1209—1224	0209—0224
	7	1225—1240	0225—0240
	8	1241—1256	0241—0256
CHAN. III	1	1257—1272	0257—0272
	2	1273—1288	0273—0288
	3	1289—1304	0289—0304
	4	1305—1320	0305—0320
	5	1321—1336	0321—0336
	6	1337—1352	0337—0352

The input module placed in channel I and indexed as number 1 will provide the Processor with status of the first 16 discrete inputs (1001-1016) per the devices connected to its 16 input circuits; input module 2 in channel I will provide the status of the inputs 1017-1032, etc. The circuits at the top of the I/O module (low terminal numbers) utilize the lower-numbered references, and those circuits at the bottom are assigned to the higher reference numbers allotted to that module. See Appendix B for additional details.

Since the I/O reference numbers are established by the index pin, and NOT the physical location of the I/O module, the I/O modules can be placed in any location provided they are properly indexed. Only those references required need be placed in the I/O configuration; inputs do not have to start at 1001, nor do outputs have to begin at line 1. Different types of I/O modules can be intermixed in any channel without regard to the voltage type.

3.1.2 The MODICON Ladder Line

As previously discussed, the MODICON 184/384 Controller controls the user's equipment by a program stored in the Processor's core memory and operating with the I/O. In block diagram form this can be illustrated as shown in Figure 22. All 184/384 Controllers have a program loaded by MODICON into a small portion of the Processor's memory. This program, which is inaccessible to the Programming Panel, is referred to as the executive or 184 MOPS (MODICON Operating System)/384 TEF (Three Eighty Four). The remaining portion of memory is referred to as the user area. It is in this section that the user enters his control program with the Programming Panel or another auxiliary device.

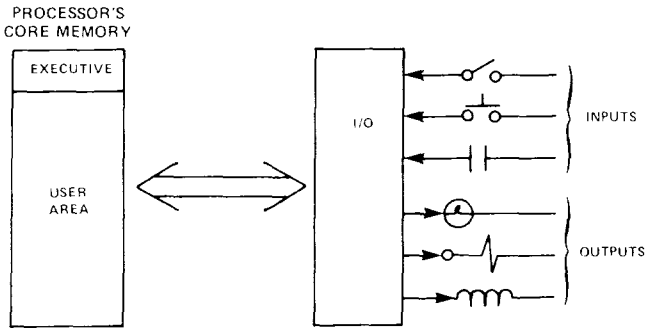


Figure 22. Block Diagram, Operation of I/O with Core Memory

All user programs are entered in a four-element line format as follows:

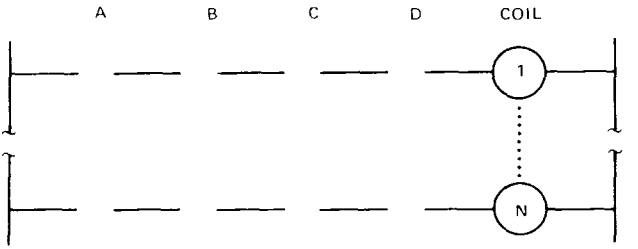


Figure 23. General Four Element Line Format

The number of lines depends on the core memory size and the specific executive installed. All lines are solved one at a time in a very short time, such that, as far as the control system is concerned, all lines are solved simultaneously. The operation of solving all lines sequentially is referred to by several terms such as scanning, scan rate, sweep time, etc.

Any single four-element line may do a number of things. It may be used to specify common relay logic functions; a timing or counting function; an arithmetic operation such as addition, subtraction, or comparison of two numbers; or a number of data manipulations covering a wide spectrum of control and data handling operations.

The type of line being entered into the Processor must therefore be identified as it is entered. Logic lines can be envisioned as rungs on a conventional ladder diagram; each rung having four elements or positions and one coil.

Utilizing the Programming Panel, the sequence for entering any one line is as follows:

1. Number (identify) the line (coil No.);
2. Select (identify) the type of line it is;
3. Enter the four elements desired.

It should be pointed out that the word "element" may not only specify a single relay contact type (e.g., a series or parallel relay contact), but may also mean entering a number — either for timing, counting, or another purpose. Any line can be converted to any valid function provided by the executive program installed. Figure 24 shows the typical MODICON Ladder Line.

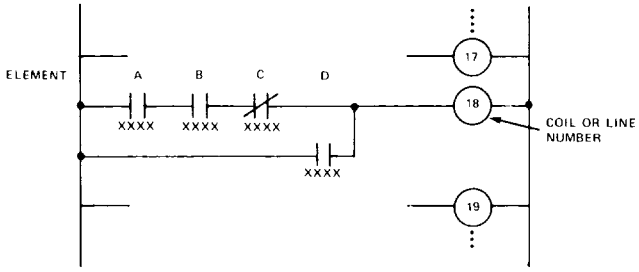


Figure 24. Typical MODICON Ladder Line

It should be noted that any line will always require four elements, even if one or two of these is not needed. (In this case, provision is made to simply duplicate an element as will be explained later.) Logic lines are defined as having exactly four elements and one coil; Their quantity does *not* depend upon number of parallel paths to a coil.

The x's shown below each element are important. Each element is associated with a reference number which must be entered into the processor along with the element. For example, it is seen in the illustration that element A is normally-open series contact. But what input or line activates it? If an input activates a contact, it is specified by a number of the form 1xxx, which the user enters *before* pushing the button which identifies the element type. Pushing the element type enters the reference and the contact type into the Controller's memory.

Figure 25 shows the MODICON Programming Panel with its pushbutton and thumbwheel switches. Referring to the sample ladder line shown previously, and to the Panel keyboard, the complete sequence of operations is as follows:

NOTE

The flow of operations proceeds logically from left to right.

1. Using the thumbwheel switches under LINE NUMBER, enter the number of the line, shown within the relay coil symbol of Figure 24 (which will also represent the output of the line).
2. Push one of the function-select pushbuttons next to the thumbwheels. This will automatically cause the Controller to know what function the line is to accomplish.

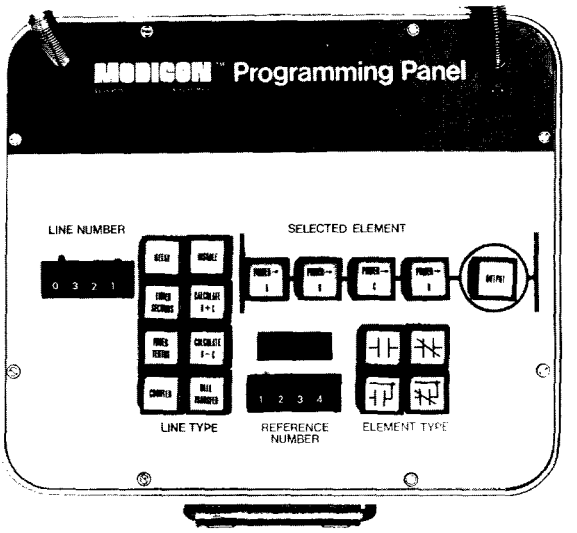


Figure 25. MODICON 184/384 Controller Programming Panel

3. Next, begin the sequence of four-element entries by depressing the pushbutton associated with the element to be programmed. It will light, indicating that the element has been selected and displayed. Elements do NOT have to be programmed in sequence.
4. Using the REFERENCE NUMBER thumbwheel switches, enter the number needed for that particular element's function. This could be either an input or output identifying number, or — in the case of timing or counting functions — the number required for counting, calculating, etc.
5. Push one of the ELEMENT TYPE pushbuttons. This will enter the information (contact type for relay contacts and reference number) into the Processor. At the same time, the number entered into the thumbwheels will show on the illuminated display directly above the REFERENCE NUMBER switches so that the user can observe that it is correct.

NOTE

Errors are easily corrected.

6. Repeat steps 3, 4, and 5 for each of the remaining three elements.

The Programming Panel error checks the line number entries and, when a contact type is depressed, the reference number. Invalid data is ignored and an error code is displayed in the reference number display. Possible error codes are listed in Table 8.

Table 8. Error Codes

␣11␣	**MEMORY PROTECT ON**
␣22␣	**INVALID LINE NO.**
␣33␣	**INVALID REFERENCE**
␣44␣	**ILLEGAL LINE TYPE**
␣55␣	**INVALID CONTACT**
␣66␣	**DATA TRANSFER ILLEGAL REFERENCE**
␣77␣	**DATA TRANSFER ILLEGAL TYPE**

With the entry of the last element, the procedure is complete, and the user may go on to the next line.

NOTE

Whenever data is being entered into the Controller with the Programming Panel (line type, references, contact type, disable status, etc.), this data is entered directly into the core memory of the Controller. If power should be interrupted prior to completion of the programming, whatever data has been entered will be retained. No additional processing is required, such as further assembling of data; whatever data the user enters is the data stored for use by the controller.

One of the valuable aspects of the MODICON Ladder Line Concept is that the coil number of any one line may be used (when entering element thumbwheel data) as a reference for any relay contact element. In other words, the output of one line may be used to control or operate an element or contact in another line, without regard for the number of times it is used.

Lines may be changed (either in whole or in part) at any time, using the Programming Panel, and any line may be tested by simulating inputs and outputs. The OUTPUT pushbutton is lighted when the coil is ON.

Once all lines are entered into the Processor, it is ready for a final checkout. For a complete description and specifications for the Programming Panel, see Auxiliary Units, Appendix A.

3.1.3 Scan

The MODICON 184/384 Controller examines (solves) each logic line in numerical sequence. Line one is the first line to be solved on each scan, followed by line two, three, etc., until all available logic lines are solved. Then the Controller goes back and solves line one again. This fixed scanning occurs at a very rapid rate from the time power is applied to the Processor until power is removed.

Since the scanning rate is very fast, it appears to an operator that all logic lines are solved simultaneously. The result of each logic line is immediately available to subsequent lines, regardless of whether this result is a change in coil state or a change in stored numerical value.

All inputs and outputs are updated once per scan concurrent with the line solving. The time from solving any individual line on one scan until that line is again solved on the next scan is defined as the "scan time" of the Controller.

3.1.4 Memory Protect

The MODICON 184/384 Controller is provided with a Memory Protect hardware feature designed to prevent accidental or unauthorized changes to part of the core memory. When the Memory Protect keylock switch (see Figure 10a) is placed in the ON position, neither the user's logic nor the executive can be altered by any external device, such as the Programming Panel, Tape Loader, Telephone Interface, or Computer Interface. The executive or logic data can be examined; but it cannot be altered. Thus, by placing Memory Protect ON and removing the key, maintenance personnel can use the Programming Panel to monitor the system, but they cannot make unauthorized changes. Only specific personnel who are provided access to the key can change the system.

Note that the Memory Protect feature protects the executive and the user's logic, but does not protect those elements that normally change — such as registers and I/O status.

3.1.5 Disable/Enable

To simplify the checkout and maintenance of a control system using the 184/384 Controller, a special feature is incorporated into all Controllers. This feature is called the Disable function. The Disable feature is operational only if Memory Protect is OFF. Any logic line can be examined by using the Programming Panel as discussed previously in Section 3.1.2. When a logic line has been selected (i.e., line number entered on line number thumbwheels), its coil status can be disconnected from the logic entered in elements A-D by depressing the DISABLE pushbutton on the Programming Panel. If the coil was OFF when the pushbutton was depressed, it will remain OFF; if it was ON, it will remain ON. The coil is no longer controlled by the program in the Controller; but is now controlled by the OUTPUT (i.e., coil) pushbutton on the Programming Panel. To change the coil state (e.g., from OFF to ON or from ON to OFF), the OUTPUT pushbutton is depressed once for each change.

When disabled, the logic line's coil, all references to this line in the ladder diagram, and any outputs driven from this coil via properly indexed output modules, will be affected solely by the OUTPUT pushbutton. The internally programmed logic still remains in the Controller and will re-establish control when the line is enabled; but this internal logic has been completely bypassed for this line by the disable function. The disable status of any logic line is permanent until altered by the Programming Panel. The line number can be altered, other lines disabled, power interrupted, memory protect turned ON, or any other change made to the system without affecting the disable status of any lines; any line disabled either ON or OFF will retain that state until changed by the Programming Panel.

The disable status of any line can be examined by entering the line number on the appropriate thumbwheels of the Programming Panel. The DISABLE pushbutton will light if the line is disabled, and be OFF if the line is enabled. If disabled, the OUTPUT (coil) pushbutton will light if the line is disabled ON and not light if the line is disabled OFF. A logic line can be enabled such that control is returned to the Controller's stored logic if the DISABLE pushbutton is depressed a second time. The disable light should go OFF if the line is successfully enabled. Memory protect must be OFF to enable lines; only the line currently being displayed will be enabled.

In addition to the logic lines, discrete inputs can also be disabled in a manner similar to the logic lines. The input to be disabled is entered on the Programming Panel's line number thumbwheels as if it were a logic line. The DISABLE pushbutton is depressed; this removes control of that input from the "real world" and assigns that control to the OUTPUT (coil) pushbutton on the Programming Panel. The input can now be forced either ON or OFF by depressing the OUTPUT pushbutton. All logic lines that use this discrete input will now respond to the disable status and not the real world. The disable status is permanent and can be altered only by the Programming Panel with the memory protect OFF. At any one time, as many logic lines and discrete inputs as desired can be disabled either ON or OFF.

NOTE

> Since the disable status is permanent, a record should be maintained of all disabled lines and inputs, so that they can be enabled at a later date. A ladder listing will show the disable state of any line or input.

In checking out a system, the disable function can be used to verify the proper wiring and operation of all discrete outputs. Each output is entered as a line number on the Programming Panel and then disabled. The OUTPUT (coil) pushbutton can be cycled ON-OFF-ON-OFF, etc., and proper operation of the discrete device observed. It is recommended that the line be enabled before the next output is tested to prevent undesirable disable statuses from occurring.

If an input such as a limitswitch fails to operate properly, its effect can be temporarily simulated by disabling the input and forcing it to the required state (ON or OFF). This is particularly useful if the input is on a remote (e.g., 2000 feet away), and the improper signal is preventing the control system from functioning. However, since the disable feature for either inputs or outputs is a very powerful function and can cause catastrophic results if improperly used, the keylock memory protect can be used to ensure changes of the disable states are made only by qualified and authorized personnel.

3.2 BASIC PROGRAMMING

All Controllers are provided with the capability to program or simulate the function of relays, timers, and counters. Any available logic line can be converted to a relay, timer, or counter line by selecting the appropriate pushbutton on the Programming Panel. There are basically three executives that provide only relay, timers, and counters designed for 184 core memory sizes 1K, 2K, and 3K, whose characteristics are as follows:

Program	No. of Lines	No. of Holding Registers	Discrete I/O (Channel)	Latches	Min. Core Size
MOPS 1 Mod 1	224	32	224/I 224/O (1,2)	224 2001-2224	1K
MOPS 1 Mod 2	464	100	256/I 256/O (1,2)	304 2001-2304	2K
MOPS 1 Mod 3	640	100	352/I 352/O (1,2,3)	32 2001-2032	3K

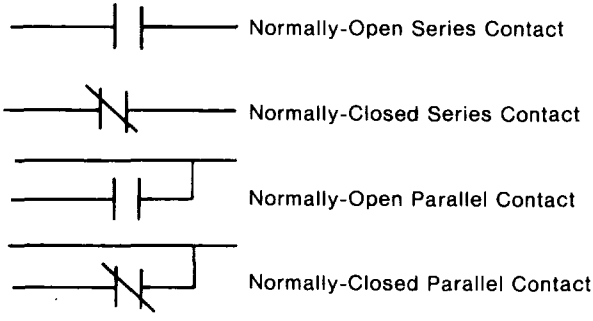
NOTE

Similar executives are available for the 384 Controller; these are called TEF programs — see Appendix F.

As can be seen, the input, output, and logic line capacity is a function of memory size. The previous discussion has defined inputs, outputs, and logic lines in a general sense; the following description will cover the programming of relays, latch relays, timers, and counters in detail. The 2K program, MOPS 1 Mod 2, will be used to illustrate the programming techniques.

3.2.1 Relays

Control logic for the 184/384 Controller is very similar to a conventional relay schematic, as shown previously. Four contact types are used:



Parallel contacts must *always* return to the left-hand leg. The B, C, and D positions can be any contact type, but the A position (for any function) must *always* be a series contact.

Since discrete inputs from user equipment are always coded 1xxx, the user may have his inputs to the Processor so labeled for ease in installation and wiring.

Coil outputs (0xxx) may be used directly to drive output circuits, or indirectly as contacts in other lines, or both. Line coils can be outputs if they are within the limits established by the MOPS and if an output module is addressed to respond to their status.

Example Relay Logic

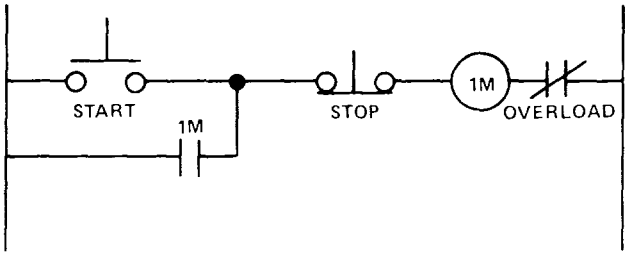


Figure 26. Sample Relay Logic

If the logic in Figure 26 were to be implemented in the 184 or 384 Controller, the control elements must be connected to the input circuits in the I/O configuration and outputs assigned. Any available inputs of the proper voltage level can be used; for this example assume the START pushbutton is wired to input 1001, the STOP pushbutton to input 1002, and the overload detection to input 1006. Output number 1 is assigned to drive the motor starter (M1). The resultant internal logic to be programmed by the user is shown in Figure 27.

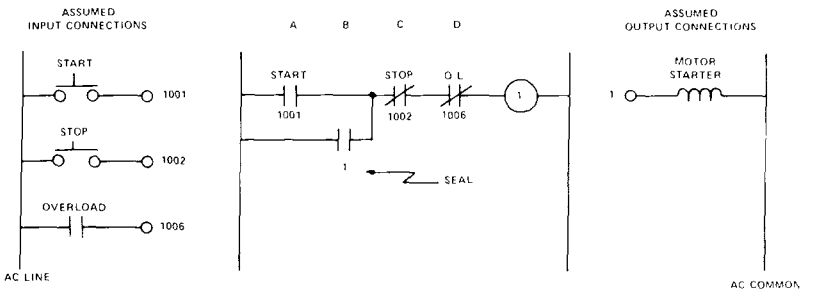


Figure 27. Sample Relay Program

NOTE

If the STOP pushbutton is wired normally closed to the input module, such that the input is normally energized, the C element of line 1 should be programmed with a normally-open contact. The normally energized reference (1002) will close the normally-open contact and allow the motor to start, unless the STOP pushbutton is depressed.

When the START pushbutton is depressed, input 1001 is energized and output 1 will be turned ON, unless the STOP pushbutton is depressed, energizing 1002 and opening the C element contact or if an overload is detected, energizing 1006 and opening the D element contact.

NOTE

Since logic lines are solved from left to right (A to D element), the C or D element, if series, will override the affect of the A and B elements. Thus, if both the START and STOP pushbuttons are energized, the STOP will take precedence and coil 1 will not be energized.

Once coil 1 is energized, the parallel B element will allow the START pushbutton to be released without de-energizing coil 1; thus the B element becomes a "seal" on input 1001. In the event of a power failure or energizing of the C or D reference, the seal will drop out and the circuit will not energize until the A element once again is energized.

NOTE

Seals are not retentive upon power failure; latches are retentive.

As previously defined, MOPS 1 Mod 2 provides 464 logic lines, 256 outputs. The 464 logic lines can be considered as two types of lines — output lines and internal lines — as follows:

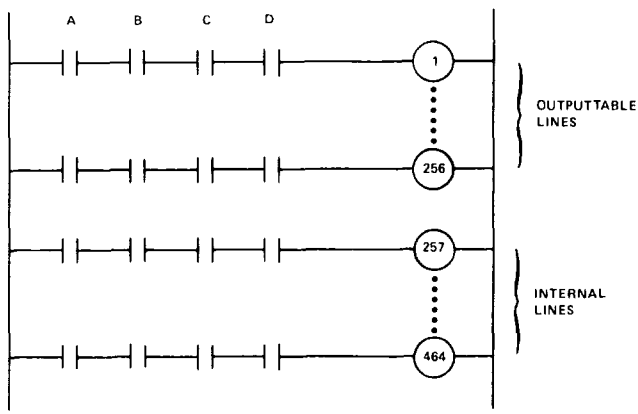


Figure 28. Definition of Output and Internal Logic Lines

Any logic lines in the range 1-256 can directly operate outputs if an output module is installed and properly indexed in the I/O section. Logic lines 257-464 cannot directly control an output and thus are used for internal control. Any logic line coil not used to control an external output can still be used to perform internal control functions. If the maximum number of outputs (including future requirements) is 200, lines 1-200 can be designated for external output control, and lines 201-464 for internal logic.

Extended Logic

If more than four elements are required in a relay line, an internal line is used to extend the number of elements that effectively control an output. As an example, assume the control logic in Figure 29 is to be implemented:

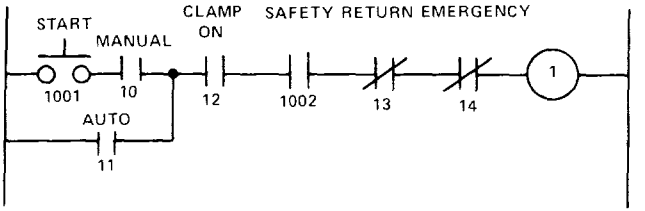


Figure 29. Sample Extended Relay Logic

Typical logic within the Controller to implement the above function would be as shown in Figure 30.

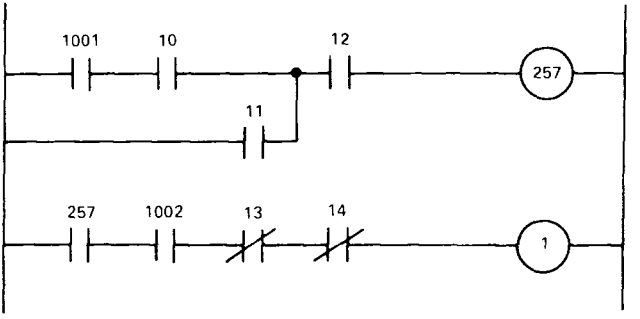


Figure 30. Sample Extended Relay Program

Latches

To build a latch circuit, the contact for a seal is normally entered at the B position. The C position is normally reserved to provide a latch contact.

The latches are designed to retain the state of a line to which they are referenced in the event of power failure. Such contacts are codes with the numbers 2xxx, where the line to which they are referenced is inserted in place of the xxx's. These latches are controlled and subject to logical changes in a similar manner as other discrete references, but permit conditions prior to the power loss to be reinstated on restoration of power.

The latch will return a line to the state it held prior to the power failure, provided that the D element contact is closed upon power-up. Latches must be programmed to be effective.

NOTE

Only specific lines are latchable as provided by the executive. For example, referring to the MOPS summary previously provided under 3.2, MOPS 1 Mod 2 provides 304 latches, specifically 2001-2304; thus lines 1-304 only are latchable.

For example, refer to Figure 31. If line 1 was energized prior to a power failure, when power is restored line 1 will automatically be energized provided the stop input is not energized. The D element contact, since it is in series, will override even the latch upon power-up. If line 1 was de-energized upon power failure, no action is taken by the latch upon restoration of power.

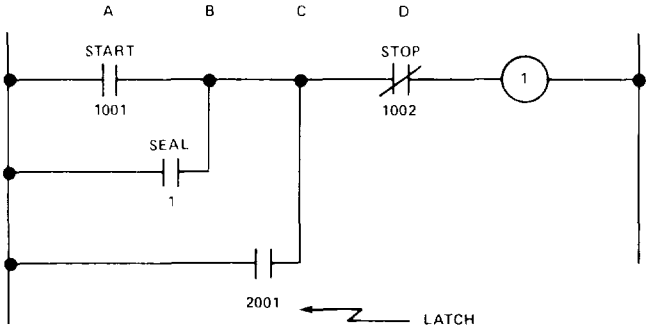
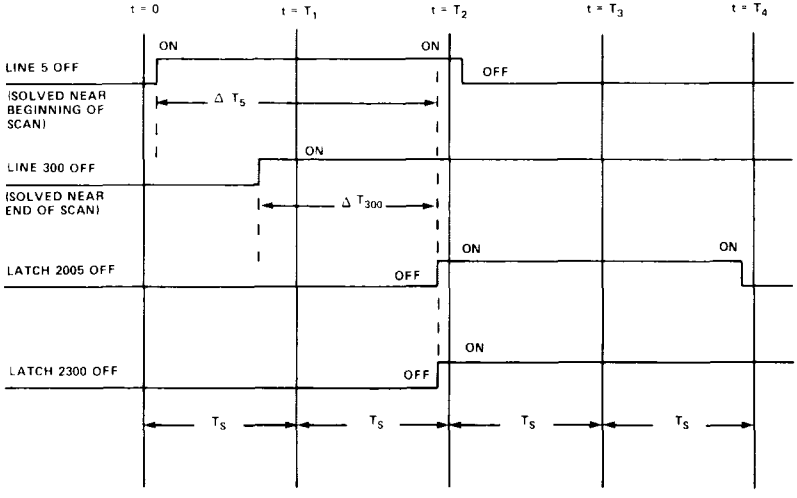


Figure 31. Sample Latch Program

Since latches are updated slightly delayed from the lines to which they are referenced, they are also called "delayed outputs." All latches are updated at the end of each scan, with the status of the lines to which they are referenced as the lines were at the *beginning* of the scan. Figure 32 will illustrate the exact time delay for lines 5 and 300, assuming there are 464 lines in the system and the MOPS allow these lines to be latchable.



NOTES:
1. T_1, T_2, T_3, T_4, etc., are the times at the end of scans 1, 2, 3, 4, etc.
2. T_S is the scan time.
3. ΔT_5 and ΔT_{300} is the delay in updating latches 2005 and 2300, respectively.
4. ΔT_n is always greater than T_S and less than $2T_S$, with latches referenced to lower-numbered lines (e.g., 2005) approaching $2T_S$ and latches on higher-numbered lines (e.g., 2300) approaching T_S.

Figure 32. Latch Timing Diagram

If the A element reference of line 1 in the previous example is guaranteed to always be ON for more than two scans, so that the latch will have time to set, the seal in the B element is not required. Normally, how long a pushbutton is depressed cannot be guaranteed; thus, it is good practice to program both the seal and the latch reference.

WARNING

If an output is only latched (not sealed and latched), and the A element contact is closed for less than two scans, the output can oscillate at the scan rate continuously until the stop is depressed.

Redundant Contacts

If, in the previous example, the overload sensor as well as the latch were required, an internal line could be used and referenced in the output line (see Figure 33). This effectively uses the internal line as a carry and increases the number of available control elements in the output line to seven.

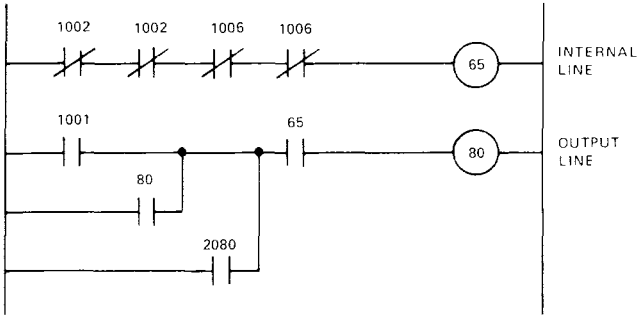


Figure 33. Sample Redundant Contacts

Note that the references for the STOP pushbutton (1002) and the overload detector (1006) are duplicated to fill in all four element positions. Every logic line must have exactly four elements; redundant contacts are used if necessary to complete the logic line.

Watchdog Timer Line

The last line of every MOPS or TEF is a relay line programmed as a Watchdog Timer (WDT) line by the executive; it cannot be altered by the user. The WDT line is four normally-closed contacts in series and referenced to itself as shown in Figure 34.

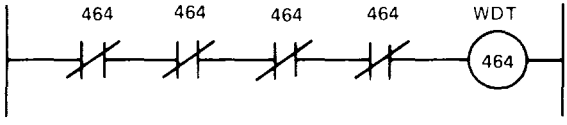


Figure 34. Sample Watchdog Timer Line

This line is a multivibrator or square-wave generator. It is ON for one scan, OFF for one scan, ON for one scan, OFF for one scan, etc., as long as the Controller is running. It takes two scans for the WDT to complete a full cycle (ON-OFF-ON); this coil reference can be used in logic lines when a square-wave generator is required to turn elements ON-OFF-ON-OFF-ON (blink error lights, etc.).

NOTE

When an executive program is listed as providing n lines (e.g., 464 lines), the user can use n−1 lines (e.g., 463 lines); the WDT is the nth (e.g., 464th) line and CANNOT be altered, only referenced.

The hardware in the Processor monitors the WDT line and is designed such that if the WDT does not change states every 200 ms, the Controller is shut down (outputs OFF, run light OFF). This ensures that if there is a hardware failure, and the Controller cannot solve a line or stops scanning, a known state will be reached (i.e., simulated power failure) with all outputs OFF and the RUN light extinguished.

Processor Fault Indications

If an external indication is required when the Processor shuts down, one output line should be programmed with the WDT references as shown in Figure 35.

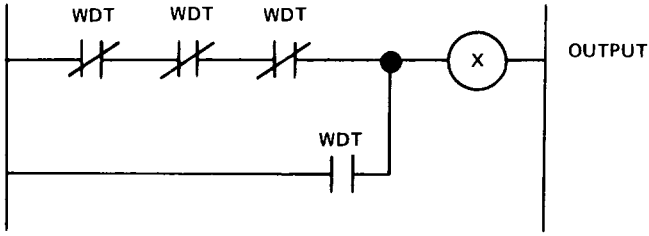


Figure 35. Sample Fault Indicator

This output will be ON as long as the Processor is running and OFF if AC power is removed or if a Processor error has been detected and shut-down accomplished. Note that this output will also be OFF if only the output module fails.

Critical output modules can be monitored by connecting an always ON output on that module to an input circuit; the input will always be ON unless either the output or input module fails. Critical input modules can be monitored by wiring an input on that module directly to power; the input will always be ON unless the input module fails.

Null Data

When a Controller is shipped from the MODICON factory, or if a new executive is loaded from the Service Center, all logic lines (except the WDT lines) are coded as relay lines with null data and all registers contain the value zero, Null data is exactly as shown in Figure 36.

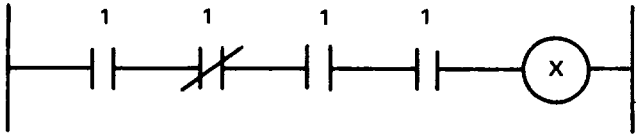


Figure 36. Null Data

NOTE

Any deviation from this line, either in references or contact types, whether or not it affects the operation of the line, will not be interpreted as null data.

Using null data, regardless of the state of coil number 1, will guarantee that all lines so coded will never be energized. This ensures that when a controller is placed in a system and power applied, no outputs will be energized until the user installs his program. When a system is "dumped" into the MODICON Service Center and a ladder listing made of the program, all lines coded as null data will be printed as a blank line. Thus, if all unused lines are coded with null data, spare lines are readily identifiable from the ladder diagram.

One-Shots

The unique timing for latches can be used to great advantage. Relay lines that are valid for exactly one scan are useful to control shift registers, various data transfer functions, some calculate applications, etc. They are called one-shots and can be built as illustrated in Figure 37 in one line using latches.

When line 57 goes from OFF to ON (scan No. 1), line 42 is not affected since this line has already been solved. On the next scan (scan No. 2), line 42 is energized and, at the end of scan No. 2, latch 2057 gets set; on the following scan (scan No. 3), line 42 is de-energized by the setting of latch 2057 on the previous scan, and remains de-energized until line 57 goes from OFF to ON again. The ON to OFF transition of line 57 does not affect the one-shot. Thus line 42 is energized for exactly one scan, from the time it is solved during scan No. 2 until line 42 is again solved on the next scan (No. 3).

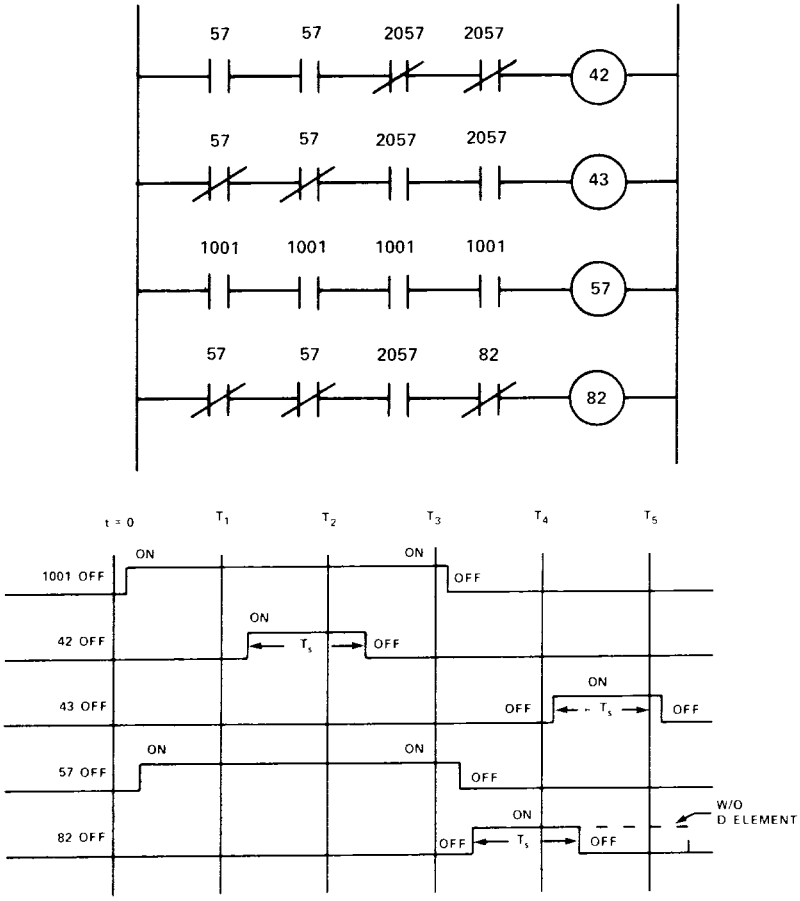


Figure 37. Typical One-Shots

The order of solution (assigment of line numbers) is very important since if line 57 were solved before the one-shot, the one-shot would be valid for two successive scans. Line 43 of Figure 37 is a one-shot triggered when line 57 goes from ON to OFF; line 82 is a one-shot triggered when line 57 goes from ON to OFF and is located after line 57. The normally-closed D element contact of line 82 ensures that this line will be valid for just one scan.

The input references (1001) in line 57 are used only to provide ON/OFF control of this line for illustrative purposes. If a one-shot is desired when an internal line goes from either OFF to ON or ON to OFF, and that line is latchable, there is no requirement to copy its status into another line; use this line and its latch as shown in either lines 42, 43, or 82.

First Scan Indicator

In some applications, actions must be taken following a power failure to clear registers, outputs, etc., but only on the first scan following a power failure. After the first scan, the elements that were cleared are controlled by other logic lines — the clear control is eliminated. The line illustrated in Figure 38 should be programmed after all clear operations have been performed. For the first scan only, all coil references are assumed to be OFF until their respective lines are solved; normally-closed references to line 420 will pass power only on the first scan.

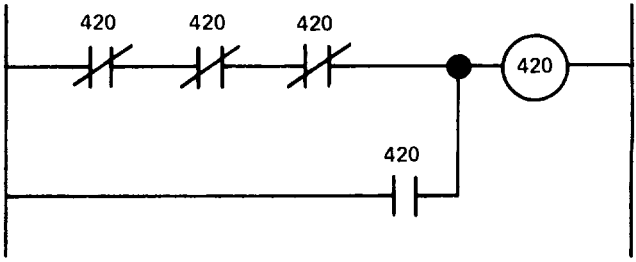


Figure 38. Sample First Scan Indicator

Entering a RELAY Line with the P112 Programming Panel

1. Set the line number on the LINE NUMBER switches.
2. Put the Controller MEMORY PROTECT switch in the OFF position.
3. Press the RELAY pushbutton. It will light and all other line type lights will go OFF.
4. Press the A element pushbutton. It will light and the B, C, and D lights will be OFF.
5. Set, on the REFERENCE NUMBER switches, the line number, input number, or latch that is to control the A position contact.
6. Press the desired ELEMENT TYPE pushbuttons to designate the type of contact to be used in the A position. The ELEMENT TYPE pushbutton will light, and the REFERENCE NUMBER display will show the reference number that has been entered.
7. Repeat steps 4, 5, and 6 for the B, C, and D element positions.
8. If the DISABLE pushbutton is lit, and not specifically desired, press it to enable the line.

NOTE

The elements of any logic line can be programmed in any order.

Programming Example (Memory Protect OFF)

Entry of a Relay Line (see Figure 39).

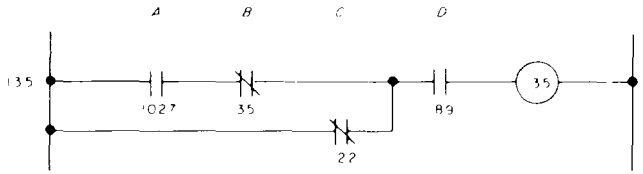


Figure 39. Relay Line Example

1. Set Line Number 135 on LINE NUMBER switches.
2. Press RELAY pushbutton.
3. Press A element pushbutton.
4. Set REFERENCE NUMBER switches to 1027
5. Press Normally-Open Series ELEMENT TYPE pushbutton.
6. Press B element pushbutton.
7. Set REFERENCE NUMBER switches to 0035.
8. Press Normally-Closed Series ELEMENT TYPE pushbutton.
9. Press C element pushbutton.
10. Set REFERENCE NUMBER switches to 0122.
11. Press Parallel Normally-Closed ELEMENT TYPE pushbutton.
12. Press D element pushbutton.
13. Set REFERENCE NUMBER switches to 0189
14. Press Normally-Open Series ELEMENT TYPE pushbutton.
15. If the DISABLE light is lit, press it to turn it OFF unless the line is to be left disabled.

EXAMPLE I — Shift Register

As a review of relay logic, a shift register can be constructed (see Figure 40), and its operation analyzed. Line 201 is basically a one-shot energized when line 202 is transitioned from OFF to ON; line 202 is controlled solely by input 1001, the shift command. Data is entered into stage one (line 200) from input 1003 and is shifted up to lower-numbered lines every time the shift command is cycled OFF to ON.

Each stage of the shift register is basically the same; reload strobe (A element), data (B element), seal (C element), and clear strobe (D element). When a shift has been commanded, line 201 becomes energized for exactly one scan; this clears the contents of all stages of the shift register via their D elements. Refer to Figure 41 for exact timing information, assuming stage 3 was ON and stage 4 was OFF prior to the shift command.

The latch reference to line 201 is utilized as the reload strobe (A element) since it is delayed from coil 201, but is also a one-shot. Thus, immediately after the one-shot (line 201) is fired, the reload strobe is energized; each stage can be reloaded since line 201 is no longer energized (it is a one-shot) and the D elements have returned to their de-energized state (closed).

The first stage (line 200) will be loaded with new data available from input 1003. The remaining stages are loaded with the data that *was* in their previous stages; latch references are used in the B elements, since the previous stages have been cleared, but their latches have not, and will still contain the previous stage's data. The C element seals in and holds the data once entered into the stage, since data entry is accomplished with one-shots.

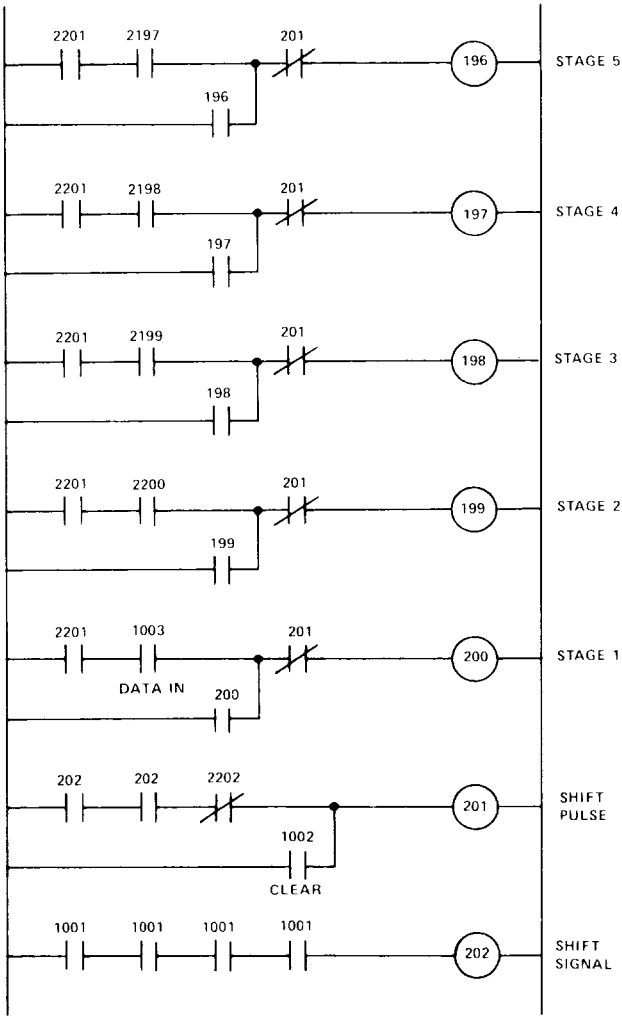


Figure 40. Example: Shift Register

Input 1002 in the D element of line 201 is used as a master clear to force all stages of the shift register to zero (OFF). When this input energized, the D elements of all stages are opened to clear the stages. Since this input will be energized for more than one scan, the stages' latches are also cleared. When input 1002 is turned OFF, stage one will be loaded with the contents of 1003 and all other stages will be OFF.

Additional stages can be added using only one latchable line per stage, following the format of lines 196-200. Since seals are utilized in all stages, if a power failure occurs, all stages will be cleared to zero (OFF). If a retentive shift register is required, either two lines per stage are required or a matrix shift register can be utilized (see EXAMPLE VI).

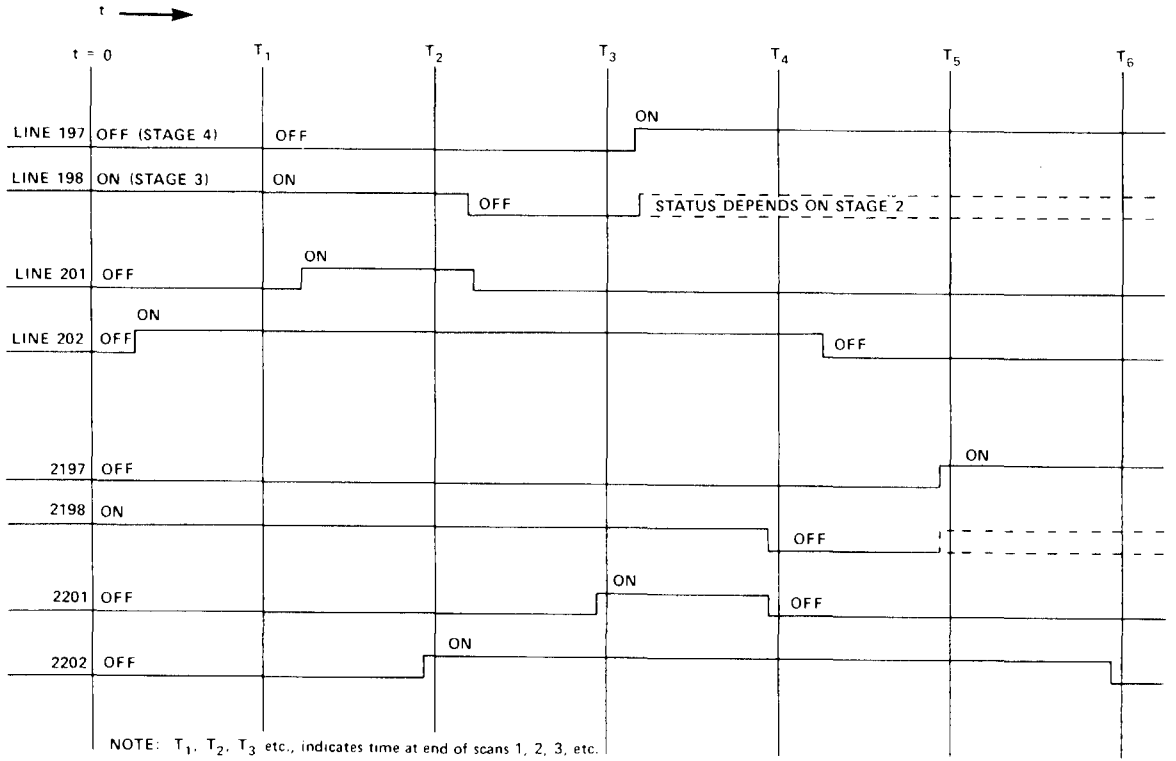


Figure 41. Timing Diagram for Shift Register

3.2.2 Timers

The 184 Controller is provided with its own internal clock which is able to generate pulses in seconds or tenths of seconds. This clock operates from the line frequency (50 or 60 Hz) of the power source connected to the main power supply. A minor change to the MOPS is required to convert from a 60 Hz to a 50 Hz executive, and vice-versa.

NOTE

The 384 Controller utilizes a crystal controlled clock for its timing signals; no changes to the TEF are required for 60 or 50 Hz operation.

Any logic line can be made a timing line (seconds or tenths of seconds) by pressing the appropriate button on the Programming Panel.

NOTE

The basic accuracy of each timer is its time increment (seconds or tenths of seconds) minus one scan time.

A typical timer line is illustrated in Figure 42.

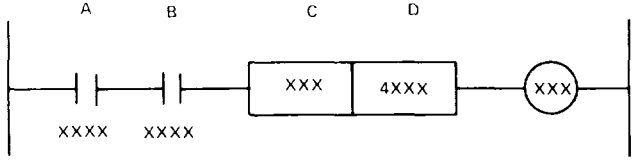


Figure 42. Typical Timer Line

The A element position specifies the series contact to be used to activate the line. The B position is used to specify what signal will reset the timer to zero; the timer is enabled when B closes and resets when B is open. The C position contains the preset time which can be 0-999 seconds for timers incrementing in units of seconds, or 0-99.9 seconds for timers incrementing in tenths of seconds. The D position specifies a storage location within the Controller where the current time is retained.

The coil of the timer line will be energized when the accumulated time is equal to the preset vale. The coil will be de-energized and the accumulated time set to zero whenever the contact in the B element is opened. No further timing is possible if the coil is energized.

With the B element closed, the timer accumulates time whenever the A element closes. If A should open when B is closed, the amount of time accumulated will be stored in the Processor so that time is not lost. Time will continue to be accumulated from its previous value when the A contact is again closed. Whenever the B element opens, regardless of the condition of the A element, or the current time or the coil state, the current time is forced to zero and will remain at that value until the B element is again closed.

NOTE

The relay contacts in the A and B element can be either normally-open or normally-closed; however, they both must be series contacts.

Specifying Time

The C element position is reserved for entering the desired amount of time, in seconds or tenths of seconds, to which the timer is to time. The

lowest number to be entered on the thumbwheels will be on the right-most thumbwheel. Whether it will specify up to 9 seconds or 0.9 seconds will depend on which function button was pushed when selecting the line type. If the tenths of seconds timer is selected and a fixed preset utilized, the decimal point will appear in the reference display of the Programming Panel.

When the C element button is pushed, all the contact type lights will come ON. Once the desired time limit (preset) has been entered on the REFERENCE NUMBER thumbwheel switches, pushing *any* of the relay buttons will cause the number to be entered and the user may proceed to the next element.

Storing Time

The number to be entered in the D position, however, is always a storage register (4xxx).

To explain the concept of registers, it is appropriate to again examine the core memory in block diagram format:

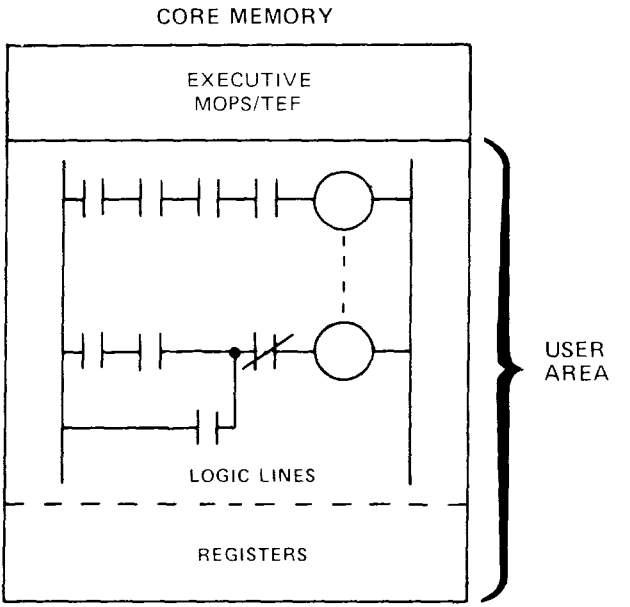


Figure 43. User Area of Core Memory

Registers are locations in the user's area of memory where numbers, up to four digits (9999), are stored; each register is a 16-bit word. However, a simple definition would be a "mailbox" or "bucket" where information (in this case, time) is permanently stored. The registers are referred to or named by reference numbers beginning with 4001 and continuing consecutively to the maximum established by the executive (4100 for MOPS 1 Mod 2).

NOTE

Holding Registers are inherently retentive upon power failure.

The storage location numbers always begin with the reference number 4 and (depending on the executive option) may include as many as 999 such

locations. In general, however, all such locations will not be used to store timing information, since these locations are useful for a great many other purposes, as will be seen later. The contents of any storage location may easily be examined (see Section 3.3.3).

Again, as with the C element procedure, the reference number is entered on the thumbwheels. When the D element is selected, all of the relay buttons will light, and the user need only push one (any) contact type to enter the reference number for the storage location, which is then displayed in the readout.

NOTE

If any element selected by the Programming Panel is a register, its contents can be displayed by holding the selected element pushbutton down. The contents can be altered by entering the new value onto the reference thumbwheels and depressing any contact type while holding the selected pushbutton down.

If the D element register number appears in the Programming Panel readout, with decimal points between each digit, the register location is also being used in the D element of another logic line. This is only a warning to the user that the contents of this register can be altered by more than one operation; it is accepted as a valid reference. Approximately 2 to 5 seconds may be required for these decimal points to appear.

Figure 44 shows an example timing line:

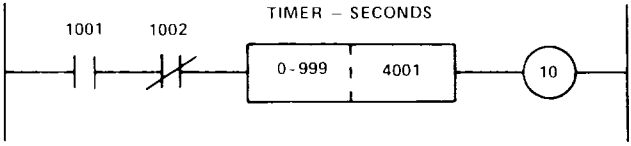


Figure 44. Example Timing Line

When 1001 is energized while 1002 is not energized (B element passing power), the timer will operate and increment the current time in register (or mailbox) 4001 at the rate of one increment every second that 1001 is closed. When the elapsed time in register 4001 equals the preset time in the C element, the line coil (No. 10) will come ON and the timer stops timing. The coil will remain ON (including after a power failure) until power through the B element is interrupted by energizing input 1002. Once the coil is energized, the status of the A element will have no effect on the coil.

The following examples illustrate various ways the basic timer line (either seconds or tenths of seconds — they are completely interchangeable) can be programmed to provide different results.

ON-Delay Energizing Timer

Referring to Figure 45, when line 83 is energized, both the A and B elements of line 101 are closed (i.e., passing power), and the timer begins accumulating time. After 15 seconds (the preset time entered into the C element of this example), coil 101 is energized and remains energized until line 83 is de-energized. If line 83 is de-energized any time prior to the timer reaching its preset, coil 101 will not come ON and the timer is reset to zero. The timer type (seconds or tenths of seconds), the preset time (15 seconds in this example), and the method of starting the timer (another logic line coil, or an input, or a latch) were all selected above for illustrative purposes only and can be altered if desired to suit other applications.

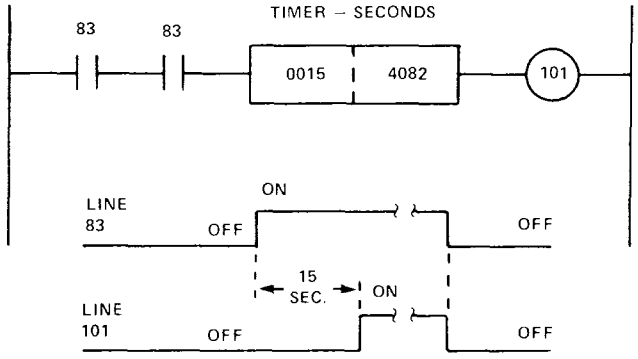


Figure 45. ON-Delay Energizing Timer

ON-Delay De-energizing Timer

The timer (Figure 46) operates similar to the previous timer, except that it is controlled by an input and operates upon one-tenth of a second intervals. The additional relay line (line 103) inverts the effect of the timer to produce the de-energizing affect. As long as the timer is OFF, line 103 will be ON; 25.6 seconds after input 1053 is energized, the timer reaches its preset and energizes its coil, causing line 103 to be de-energized. As soon as input 1053 is de-energized, the timer is reset to zero turning its coil OFF, thus allowing line 103 to be energized again. As an option, other logic control can be placed in line 103 in place of three of the normally-closed contacts referenced to the timer, line 102.

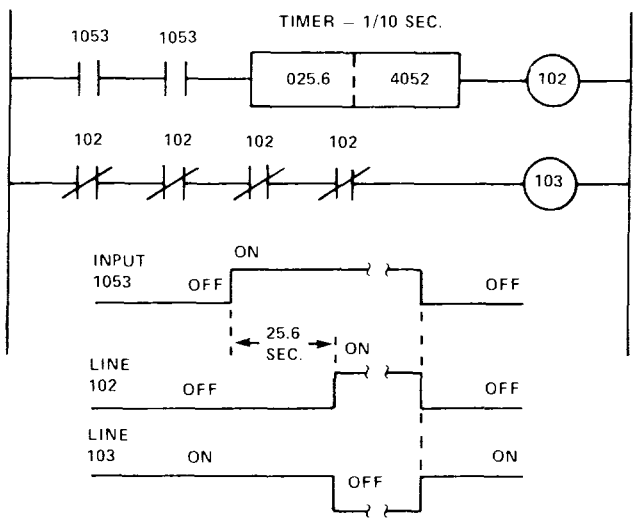


Figure 46. ON-Delay De-Energizing Timer

OFF-Delay Energizing Timer

The ON-delay energizing timer previously discussed can be converted to an OFF-delay timer by merely using normally-closed contacts as shown in Figure 47

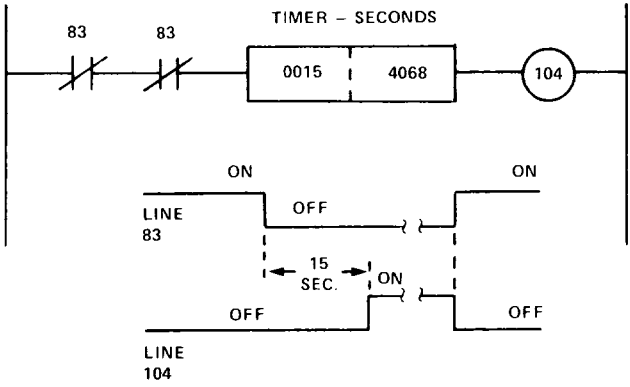


Figure 47. OFF-Delay Energizing Timer

The only difference in operation is that when line 83 is *de-energized,* the timer begins to increment time. The timer line 104 will energize its coil 15 seconds after line 83 is de-energized. Whenever line 83 is energized (opening the B element contact), the timer is reset to zero and its coil is turned OFF.

OFF-Delay De-energizing Timer

Similarly, the ON-delay de-energizing timer previously discussed can be converted to an OFF-delay timer as illustrated in Figure 48.

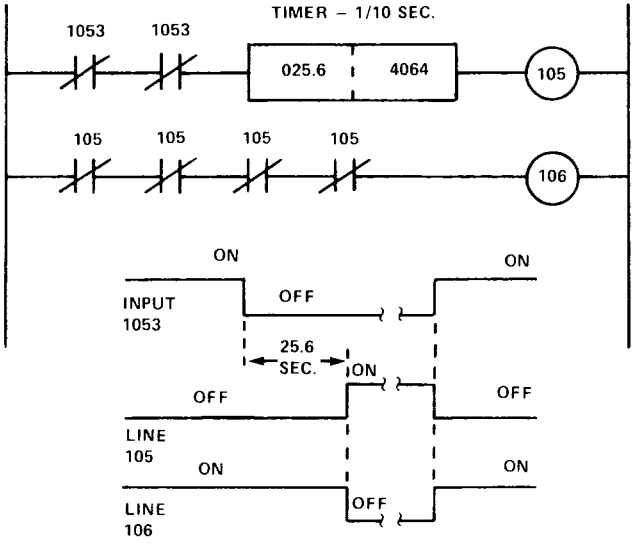


Figure 48. OFF-Delay De-Energizing Timer

When input 1053 is *de-energized*, the timer begins to accumulate time and 25.6 seconds later energizes its coil, causing line 106 to be turned OFF. Whenever input 1053 is energized (opening the B element contact), the timer is reset to zero, its coil turned OFF, and line 106 energized.

Accumulator Timer

In the previous examples, the same reference was used in both the A and B elements of the timers. This caused the timer to be reset to zero whenever it stopped timing. However, the flexibility of the timer line allows it to be converted to a timer that accumulates time by only using different references in the A and B elements as shown in Figure 49.

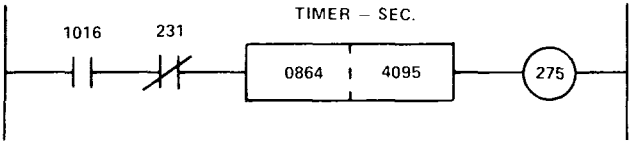


Figure 49. Accumulator Timer

As long as input 1016 is energized, the timer will accumulate time up to its preset of 864 seconds (14.4 minutes). Whenever input 1016 is de-energized, the timer stops timing but will retain its current time stored in register 4095. When input 1016 is again energized, the time begins to accumulate from the previously retained value. Thus, input 1016 can be cycled ON-OFF-ON many times, and the timer will accumulate only the time it is energized until its preset value is reached, at which time the timer stops timing and energizes its coil. The current time is retained even through power failures and is reset to zero only by opening the B element contact (energizing line 231 in this example). Every time line 231 is energized, the contents of register 4095 will be forced to zero and remain at zero until line 231 is de-energized, again closing the B element.

Self-Resetting Timers (Oscillators)

In Figure 50, the timer is programmed to reset itself by using its coil references as the control on the normally-closed B element contact. As long as coil 99 remains energized (closing the A element) the timer will accumulate time in tenths of seconds up to its preset of 18.5 seconds. When the preset is reached, the coil is energized and, on the *next* scan, the B element is opened, resetting the timer to zero and turning its coil OFF. The timer is thus re-enabled and begins to accumulate time from zero. The coil of this timer line (coil 111 of Figure 50) thus is energized for exactly one scan every 18.5 seconds that coil 99 is energized. This output can be considered a series of pulses, each one scan wide, separated by the preset time.

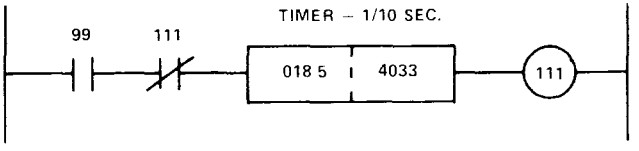


Figure 50. Self-Resetting Timer

3.2.3 Counters

A counter line operates in the same manner as a timer, and Figure 42 describes both. However, instead of pushing one of the two timer buttons on the Programming Panel, the COUNTER button is pushed after the line is identified on the LINE NUMBER thumbwheels.

As with timers, the event to be counted is received through the A position contact, and the counter is reset to zero through the B position contact.

NOTE

The counter is incremented by one when the A element contact is transitioned from open (not passing power) to closed (passing power) state provided the B element is closed.

The number of events to be counted (up to 999) is entered through the REFERENCE NUMBER thumbwheels when specifying the C position. Entry of data via the C and D positions is like that for timers, any contact type button can be pushed.

Also, as with timers, the number of events counted is stored in the Processor at a storage location whose reference number is 4xxx. This register must be entered at position D. Completion of the full count will cause the coil output to be energized and counting stopped.

Cascaded Counters

The information stored in the 4xxx locations is available for different functions. Also, the coil of a timer line might be used as the A contact for a counter, and the accumulated count used to reset both timer and counter. This technique can be utilized to extend the range of timers and counters by cascading them. The example in Figure 51 is of two counters cascaded to provide a preset of 25,000 (50 X 500) plus a timer and two counters to provide time in seconds (4051), minutes (4052), and hours (4053).

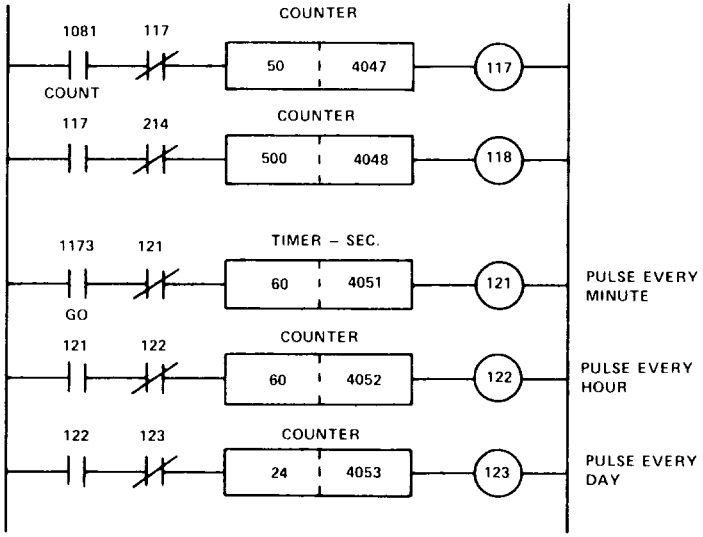


Figure 51. Cascaded Counters

Entering a Timer or a Counter 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 the TIMER SECONDS or TIMER TENTHS or COUNTER pushbuttons as desired. It will light and all other line type lights will go out.
4. Press the A element pushbutton. The A lamp will light and the B, C, and D lights will be OFF.
5. Dial on the REFERENCE NUMBER switches 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 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. Repeat steps 5 and 6 for the B element position.
8. Press the C element button. It will light and B will go out, and all four ELEMENT TYPE pushbuttons will light.
9. Set on the REFERENCE NUMBER switches the preset time or count value and press any one of the ELEMENT TYPE pushbuttons. Maximum value is 999.
10. Press the D element position. Select on the REFERENCE NUMBER switches the location where the accumulated time or count will be stored. Press any ELEMENT TYPE pushbutton.
11. If the DISABLE pushbutton is lit, and not specifically desired, press it to enable the line.

Example of a Timer Line 53 (see Figure 52)

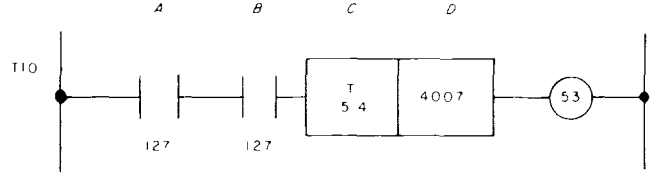


Figure 52. Timer Line

1. Set Line Number 53 on LINE NUMBER switches.
2. Press TIMER TENTHS pushbutton.
3. Press A element pushbutton.
4. Set REFERENCE NUMBER switches to 0127.
5. Press Normally-Open Series ELEMENT TYPE pushbutton.
6. Press B element pushbutton.
7. Leave REFERENCE NUMBER switches set to 0127.
8. Press Normally-Open Series ELEMENT TYPE pushbutton.
9. Press C element pushbutton.
10. Set REFERENCE NUMBER switches to 0054.
11. Press any one of the ELEMENT TYPE pushbuttons.
12. Press D element pushbutton.
13. Set REFERENCE NUMBER switches to 4007 and press any element type.

14. If the DISABLE lamp is lit, press it to turn it OFF, unless the line is to be left disabled.

Example of a Counter Line (see Figure 53)

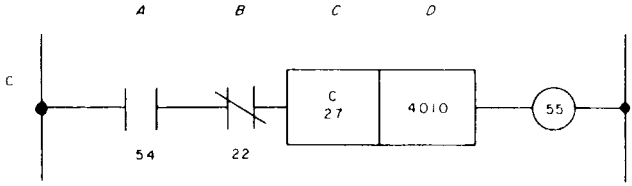


Figure 53. Counter Line

1. Set the Line Number 55 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 0027.
11. Press any of the ELEMENT TYPE pushbuttons.
12. Press the D element pushbutton.
13. Dial 4010 in the 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 II — Scan Time Evaluator

As a review of timers and counters, a circuit can be built to evaluate the average scan time of a Controller while it is operating (see Figure 54). These lines can be added to an operating system to evaluate its scan time and then removed when the final system is delivered. Any convenient spare lines can be used; they do not have to be consecutive. However, they should be programmed in the order provided. Any convenient two spare registers can be used for the D element of the counter and timer.

Line 100 is a one-shot fired when line 101 is transitioned from OFF to ON; line 101 is basically controlled by input 1001. An input is not required, just a control signal that can be turned OFF to ON; another line or an input can be disabled OFF to ON to control the one-shot. As soon as the one-shot is fired, line 102 is energized and sealed. The normally-closed D element contact in line 101 ensures that once line 102 is energized, the one-shot cannot be fired a second time. As soon as line 102 is turned ON, line 103 begins to pulse exactly as the Watchdog Timer (WDT) line cycles ON-OFF-ON.

The counter (line 104) counts these controlled pulses from line 103 until 500 pulses are detected. When the counter's preset is reached, coil 104 is energized, dropping the seal on line 102, stopping the pulses, and resetting the counter to zero. Since it takes two scans for the WDT to cycle ON-OFF-

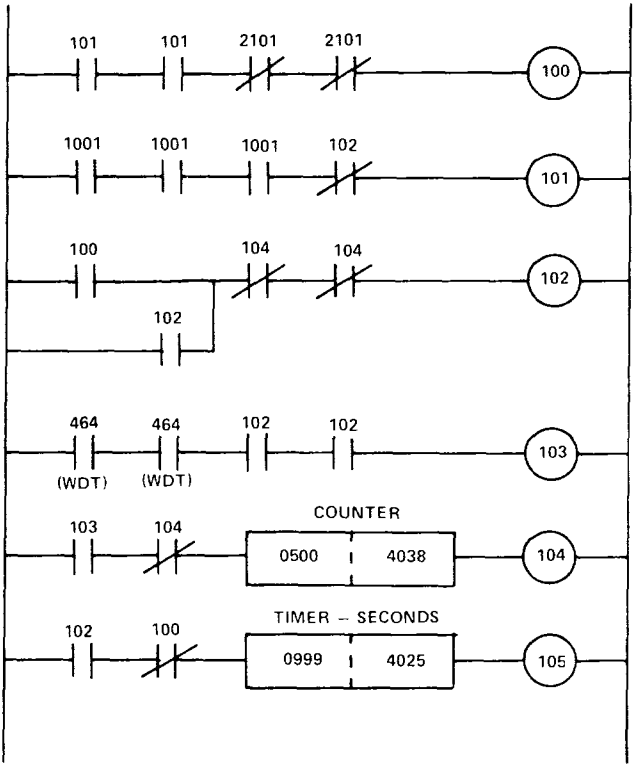


Figure 54. Example: Scan Time Evaluator

ON (which the counter counts as one count), it will take 1000 scans for 500 counts to be detected. The timer starts timing as soon as line 102 is energized (start of pulses), and stops when line 102 is de-energized (1000 scans have been detected). Thus, the contents of register 4025 (D element of line 105) will be the time in seconds it took the Controller to go through 1000 scans, or the average scan time in milliseconds for each of these 1000 scans. The timer is reset to zero only at the start of the evaluation; thus the value in register 4025 will be available any time after completion of the evaluation. The preset in line 105 is established as high as possible since the timer should stop only when line 102 is de-energized, and not at any specific value.

The one-shot is optional since the control reference (e.g., input 1001) can be used directly in the A element of line 102 and the B element of line 105, provided it is very short in duration. As long as this reference is ON, or if it is energized a second time, the timer is reset to zero but the counter continues to count. Thus, to reduce error, a short duration reference, i.e., a one-shot, must be used. The D element of line 101 ensures that the timer cannot be reset to zero while the scan evaluation is in progress (i.e., line 102 ON).

Summary

One of the major advantages of the 184/384 Programmable Controller is the flexibility the executive program provides. Since this program is in core memory, it can be altered to change the Controller's characteristics without modifying the hardware. The three basic MOPS programs listed at the

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

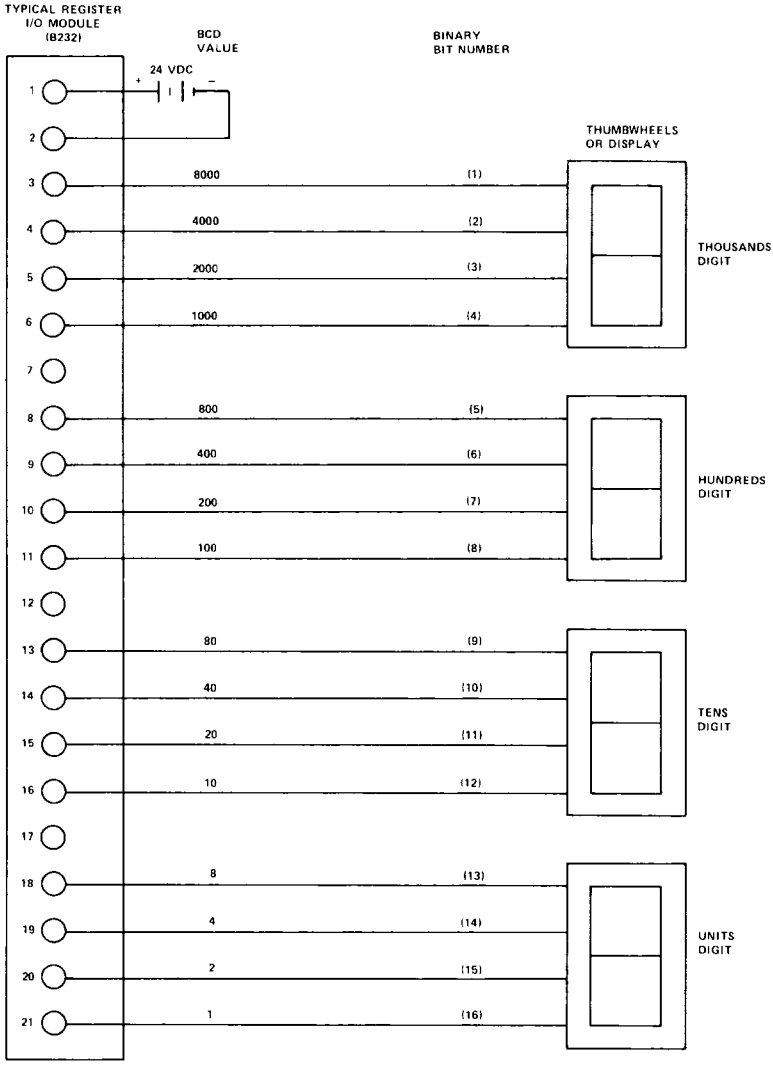


Figure 55. Wiring of Input/Output Registers

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		320 Discrete & 12 Register		352 Discrete & 10 Register	
	Input	Output	Input	Output	Input	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

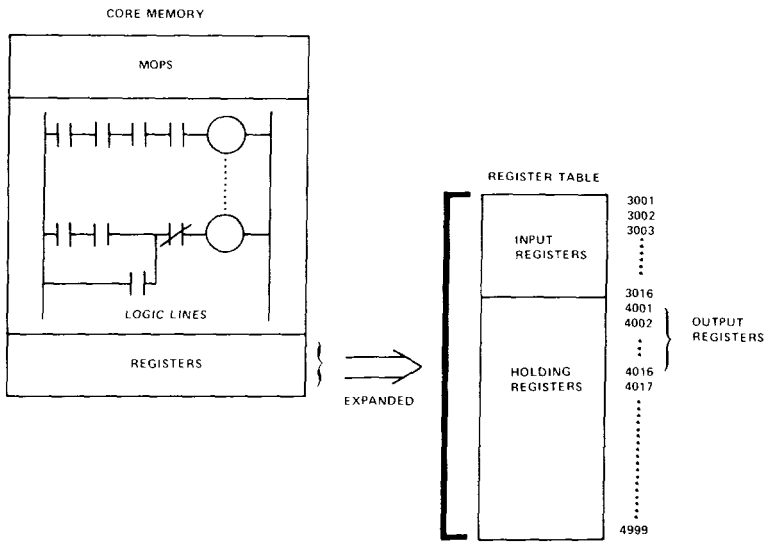


Figure 56. Block Diagram of Core Memory

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.

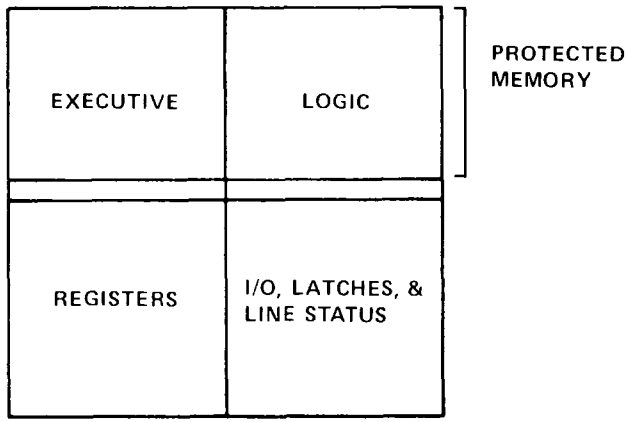


Figure 57. Core Memory Allocation

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	999	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	

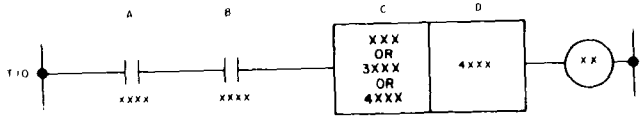


Figure 58. Typical Remote Set

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)

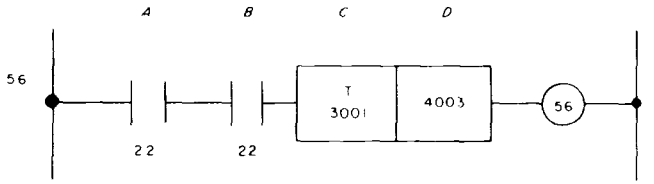


Figure 59. Remote Set Timer Line

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)

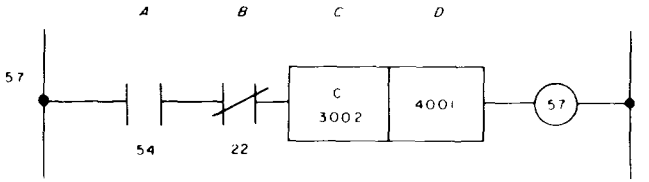


Figure 60. Remote Set Counter Line

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.

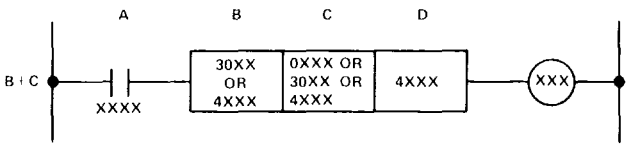


Figure 61. Typical Arithmetic Line

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.

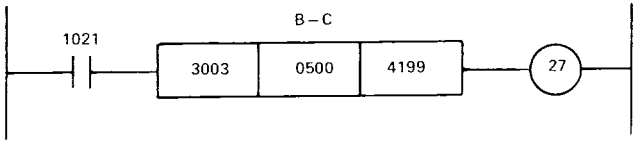


Figure 62. Sample Set Points

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.

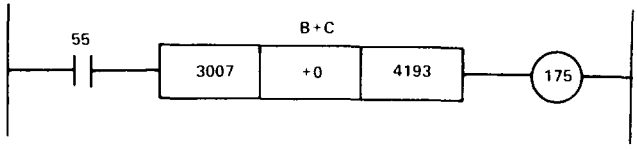


Figure 63. Sample Register to Register Move

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.

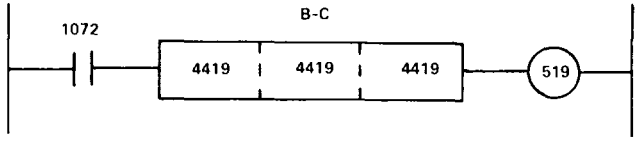


Figure 64. Sample Clearing of a Register

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.

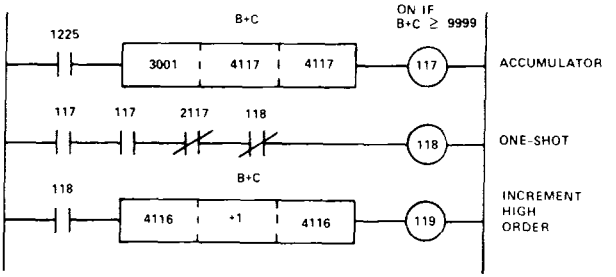


Figure 65. Sample Double-Precision Add

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.

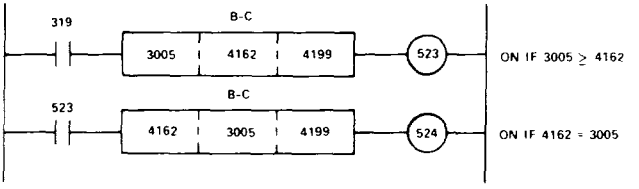


Figure 66. Sample Detection of Equality with B+C Overflow Coil Option

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.

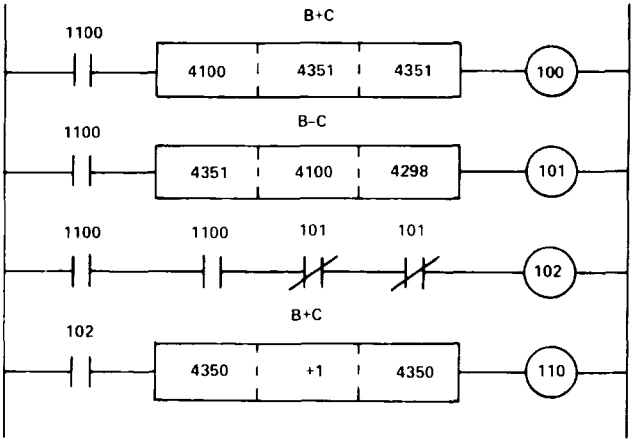


Figure 67. Sample Detection of Overflow with B+C Equal Coil Option

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.

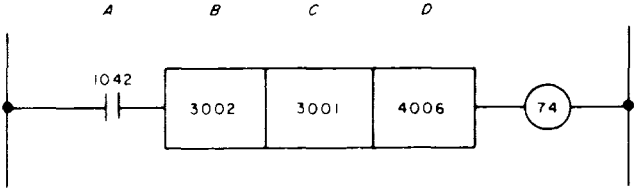
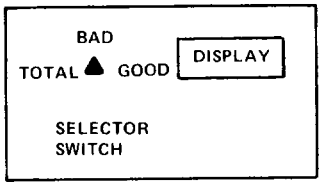


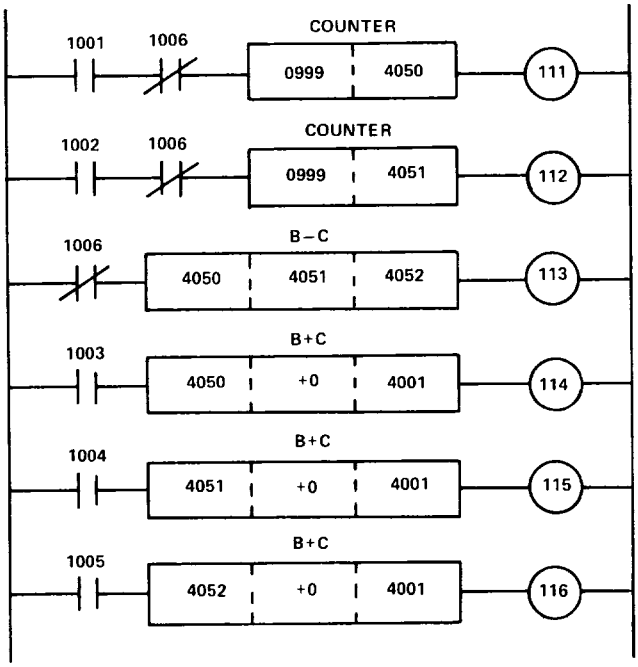
Figure 68. Calculate B+C Line

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



INPUT ASSIGNMENTS

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

REGISTER UTILIZATION

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.

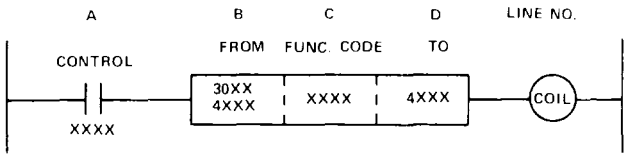


Figure 70. Typical DX Line

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).

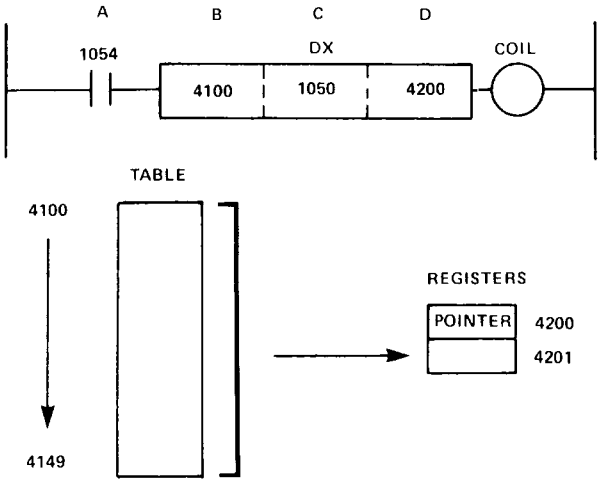


Figure 71. Sample Table to Register Move

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.

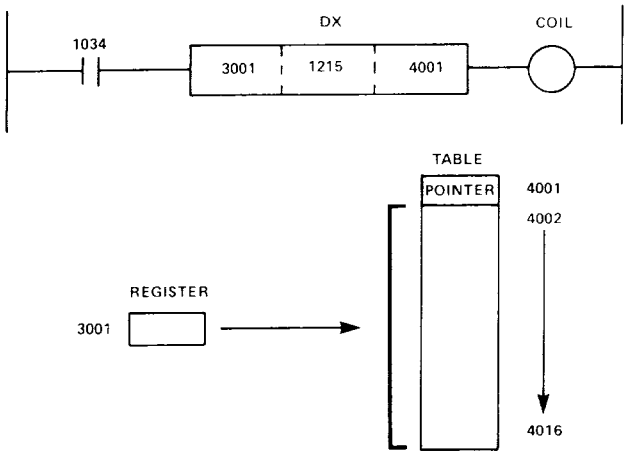


Figure 72. Sample Register to Table Move

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.

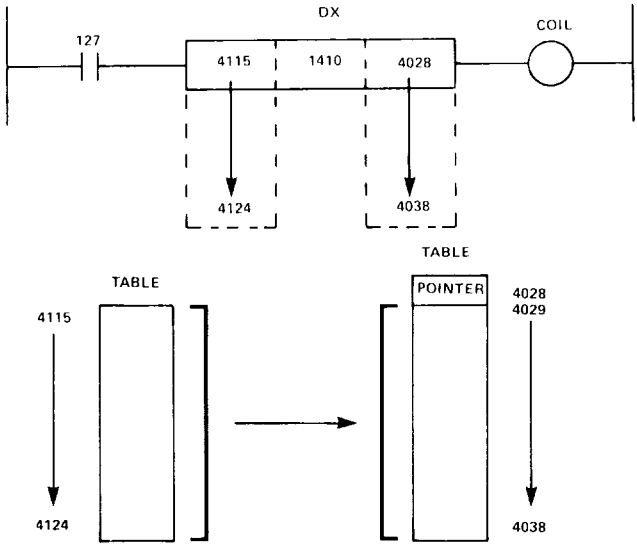


Figure 73. Sample Table to Table Move

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.

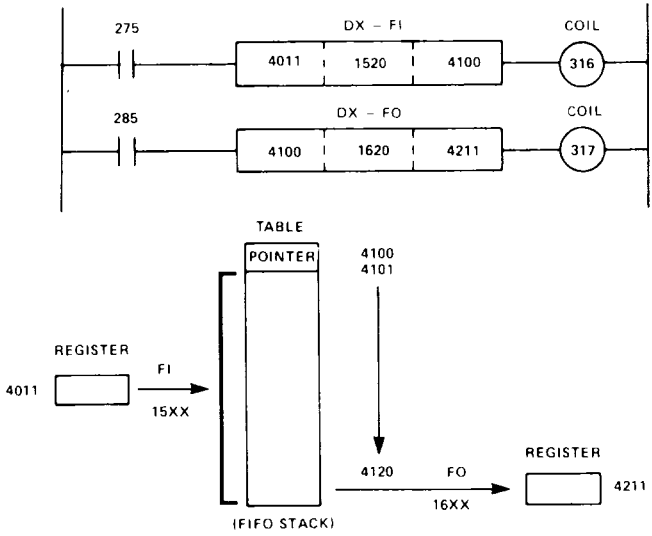


Figure 74. Sample FIFO Stack

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

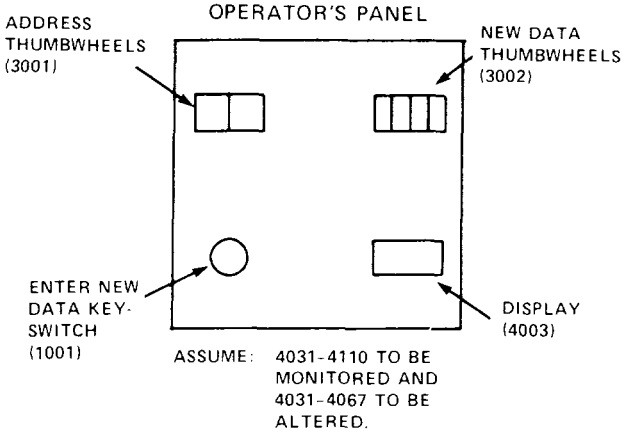
Function code 17XX is discussed after function code 14XX.

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.



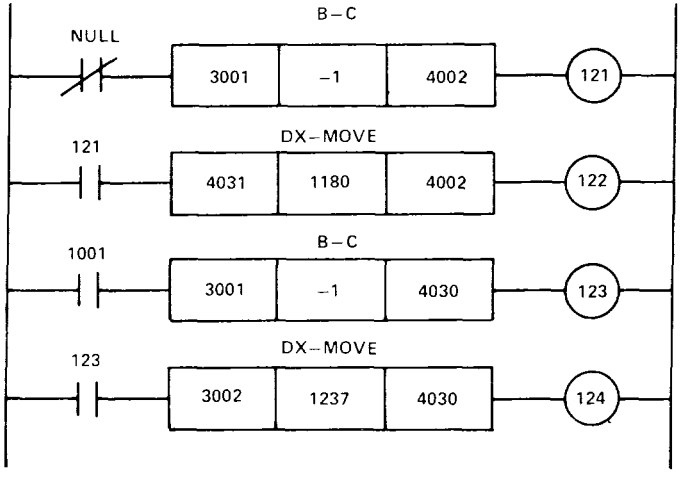


Figure 75. Example: Operator Monitor and Change of Registers

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:

Basic		Improved	
20XX	Logical AND	25XX	Complement
21XX	Logical OR (Inclusive)	26XX	Logical OR (Exclusive)
22XX	Matrix Compare	27XX	Rotate Left
23XX	Clear Bit or Sense Bit	28XX	Rotate Right
24XX	Set Bit or Sense Bit	29XX	Sequencer Move

NOTE

All 184 Controllers with matrix provide the basic capability; only those executives specifically noted in Table 12 as having the improved capability provide codes 25XX-28XX. All 384 Controllers are provided with both Basic and Improved Matrix capabilities regardless of TEF selected. Only 384A and 384B Controllers can utilize code 29XX, unless specifically noted in Table 12.

Matrices are defined as sequential registers, each of 16 bits, up to a maximum of 99 registers (1584 bits). The size of the matrix in registers is defined by the XX digits of the functional codes. The individual bits of a matrix are numbered 1 through 1584 depending on their location in the matrix. The numbering begins at the high-order bit of the first register and continues left to right, as one would read lines of a page in a book, until the low-order bit of the last register is reached. The last bit of each matrix will be evenly divisible by 16.

For example, Figure 76 illustrates the numbering of a 5-register (80-bit) matrix. The quantity of bits in any matrix is always mutiples of 16; the smallest matrix is one register (16 bits) in length. A bit can be described as ON, ENERGIZED, VALID, SET, or TRUE if its numberical value is one (1); OFF, DE-ENERGIZE, INVALID, CLEAR, or FALSE can be used to describe a bit whose numerical value is zero (0).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	FIRST REGISTER
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	=2 REGISTER
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	=3 REGISTER
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	=4 REGISTER
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	=5 REGISTER

Figure 76. Typical Matrix Bit Numbering

When receiving binary data via input registers (30XX), including analog inputs, these registers must be coded in the I/O Allocation Table (part of the executive) as binary registers. If they are not coded as binary register, the data obtained from the input modules will first have its bits interpreted as a BCD number. See 4.1.4 for additional details. A similar modification must be made to provide binary (NOT BCD) data to the output modules from holding registers (40XX). Figure 77 provides two examples of outputting binary data from register 4208 via a BCD coded register (4006).

Note that if the resultant magnitude of the 16 bits in any register is greater than 9999, an incorrect BCD display will occur. Also viewing a binary register from the Programming Panel results in a binary-to-BCD conversion. Table 13 summarizes the resultant BCD display for various bit configuration. If output register 4006 of Figure 77 was coded as binary in the I/O Allocation Table, the bit pattern in 4208 would be copied into 4006 without change.

All Matrix operations are accomplished in their entirety every scan the A element is closed (passing power), regardless of the length of the matrices.

Table 13. P112 Hexadecimal Symbols

Bit Configuration	Display	
0000	0	
0001	1	
0010	2	
0011	3	
0100	4	
0101	5	
0110	6	
0111	7	
1000	8	
1001	9	
1010	⊏	(10)
1011	⊐	(11)
1100	⊔	(12)
1101	⊑	(13)
1110	⊢	(14)
1111	(blank)	(15)

20XX — Logical AND of Two Matrices

NOTE

The Matrix AND and OR operations are useful to construct masks within the Controller as well as to move blocks of data in one scan.

This code, when the A element is closed, causes the contents of the matrix referred to in the B element to be logically ANDed with the contents of the matrix referred to in the D element and the result stored in the D element matrix. This operation is accomplished on a bit-by-bit basis and is done every scan as long as the A element is passing power; the entire AND operation is done once each scan. The contents of the B element matrix is retained and the previous content of the D element matrix is destroyed and replaced with the result of the AND operation.

For a one (1) to appear in the D element matrix after the AND operation, a one (1) must appear in both the B element matrix and the previous D element matrix; in all other cases, a zero (0) will appear in the resulting D element matrix. The coil has no significance, and is OFF in all cases. As an example, refer to Figure 78.

The B element refers to three consecutive registers (4166-4168) as does the D element (4009-4011) since the functional code has defined each matrix as three registers in length. In all cases, matrix AND will be accomplished on matrices of equal length. For this example, on the first scan that input 1021 is energized, a bit-by-bit AND will be performed between registers 4166-4168 and registers 4009-4011, and the result stored

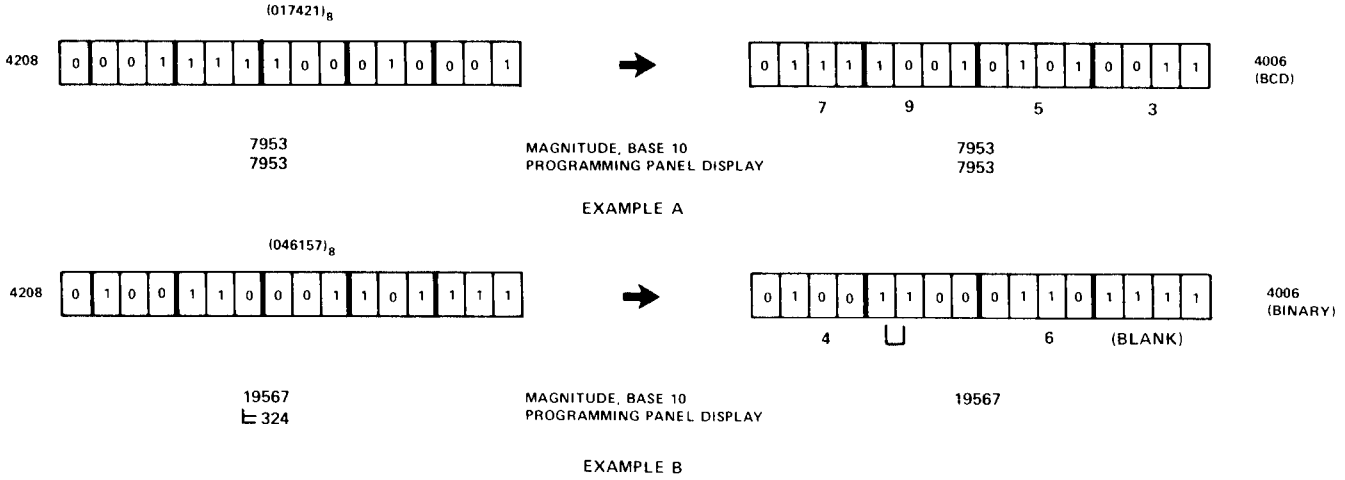


Figure 77. Programming Panel Display of Binary Data

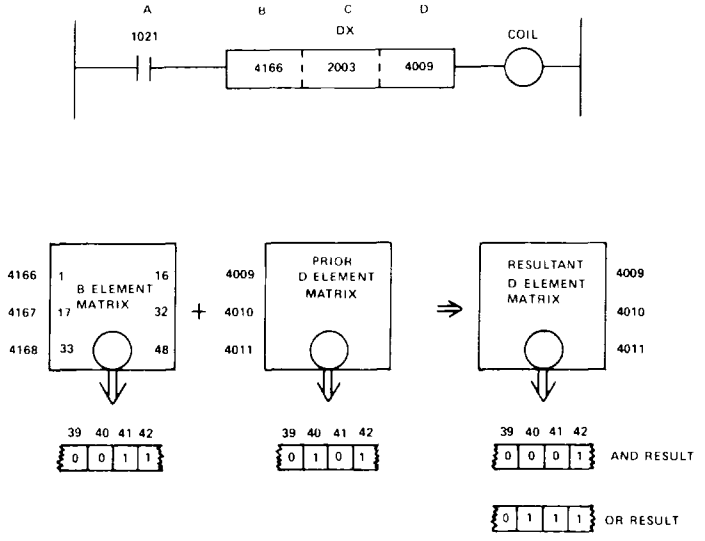


Figure 78. Sample Matrix AND and OR

in registers 4009-4011 assuming these output registers are coded for binary data. All bits in the matrices will be ANDed each scan the A element is passing power.

On subsequent scans, an AND operation is performed between the B element matrix and the resultant D element matrix; as long as the B element matrix does not change, the resultant D element matrix will not be altered.

If a one (1) in the B element matrix is changed to zero (0), the D element matrix will have a zero (0) in that location, even if the bit in the B element matrix is changed back to a one (1). Specific examples are provided by observing the operation of bits 39-42. Since there is a zero (0) in the B or previous D element matrices (or both) for bits 39-41, the resultant D element matrix contains a zero (0) for these bits. Only bit 42 has a one (1) in both matrices and thus has a one (1) in the resultant matrix. At no time will the coil be energized.

21XX — Logical OR (Inclusive) of Two Matrices

This code, when the A element is closed, causes the contents of the matrix referred to in the B element to be logically ORed with the contents of the matrix referred to in the D element, and the result stored in the D element matrix. This function operates similar to code 20XX discussed previously, except that a logical OR is performed between the two matrices. As an example, assume that the functional code of Figure 78 was 2103. The only difference in operation would be the specific results; bits 39-42 of the resultant D element matrix would contain a 0, 1, 1, 1, respectively.

Since only bit 39 has a zero (0) in both the B element matrix and the prior D element matrix, it is the only bit to contain a zero in the resultant matrix; all other bits are set to one (1). Thus, the logical OR will result in a one (1) if either matrix contains a one (1); it will result in a zero (0) only if *both* matrices contains a zero (0). Note that since this is an *inclusive* OR, a bit that is one (1) in both matrices will be a one (1) in the resultant matrix. Again the coil will be OFF in all cases. On subsequent scans, if the B element matrix has a zero changed to a one, the D element matrix will have a one in that location, even if the bit in the B element matrix is changed back to a zero.

22XX — Matrix Compare of Two Matrices

NOTE

The matrix compare is useful to monitor the status of large numbers of inputs or states as well as to detect changes in bit status. It is a very useful function.

This code, when the A element is closed, causes the contents of two matrices to be compared on a bit-by-bit basis. The compare operation will continue until a miscompare is observed, which will cause the coil to be energized and the compare terminated, or until the end of the matrices is reached. If the matrices compare exactly, the coil remains OFF and the next line of logic is performed.

The B element register refers to the first register of one matrix; the remaining registers of this matrix must follow this reference in ascending order. The functional code in the C element defines both the operation and the length in registers (XX) of each matrix. The D element register refers to a pointer register where the bit number that is currently being compared is stored; registers for the second matrix of the compare must follow this pointer register in consecutive order. Thus the B element register refers to XX registers and the D element register refers to XX+1 registers.

The contents of the pointer register is incremented by one before a compare is accomplished and will not be changed when a miscompare is encountered. Thus the bit number causing the miscompare is available from this pointer register after completion of a compare with coil ON; if no miscompare was detected (coil OFF), the pointer register will contain the bit number of the last bit in the matrix plus one.

NOTE

If more than one miscompare occurs in a matrix, the coil will remain ON for successive scans until the end of the matrix is reached, and the pointer register will contain (for one scan) each successive bit number that miscompares.

The compare is always begun at the bit referred to by the contents of the pointer register *plus* one (1), unless this value exceeds the number of bits in the matrix, in which case the comparison begins with the first bit of each matrix. The contents of the pointer register will exceed the number of bits in the matrix if the compare is restarted after a successful compare (i.e., next scan, if A element contact remains closed), since the pointer register would contain the bit number of the last bit in the matrix plus one, and that value would be incremented by one at the start of the next compare; or the pointer register could be forced to any value by other logic lines. The contents of either matrix are *not* altered by this function; only the pointer and coil status change.

As an example, refer to Figure 79 and assume register 4372 contains a zero. When input 1056 is turned ON, the comparison will begin with bit one (1) and continue until either a miscompare or the end of the matrix is encountered. Assume that at bit 23, a miscompare occurs; either a one (1) is in the B element matrix and a zero (0) in the D element matrix or vice-versa. The coil will be energized and the scan continued with the next line of logic; the pointer register will retain the value 23.

If input 1056 remains ON, and the contents of register 4372 has not been altered, on the next scan the comparison will be restarted from where it was last terminated, by comparing bit 24. If the second miscompare was encountered at bit 56, for example, the compare will be terminated a second time, the coil will remain energized, and register 4372 will contain the value 56.

Note that the location of the first miscompare is lost unless it is copied to another location (e.g., register-to-table move, continuous, controlled by the compare coil) before the line is solved again. If no miscompares are located, register 4372 will contain the value 81 and the coil will be OFF.

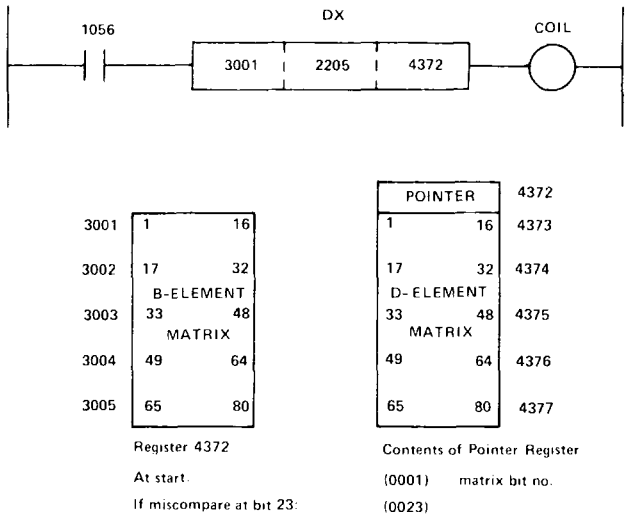


Figure 79. Sample Matrix Compare

On the next scan, assuming input 1056 is still energized, register 4372 will be incremented to 82 and, since this exceeds the size of the matrix as defined by the functional code, it will be reset to one (1) and a comparison will begin again at the first bit.

Note that, in general, either matrix of the compare can be referred to by either the B or D element; however, since the holding register whose value this code alters is associated with the D element, input registers must be referred to only by the B element.

If the last bit of the matrices miscompares, the next compare will be accomplished starting at the beginning of the matrices. Thus the end of the matrices (coil OFF) will not be detected between compares. Using the previous example, if only bit 80 miscompares, the holding register will always contain an 80 and the coil will be ON; the coil will not oscillate between ON (miscompare) and OFF (end of matrix). A calculate line can be used to detect when the pointer is equal to or greater than the number of bits in the matrix; this indicates the compare is at the end of the matrix.

If all bits agree, the entire matrices will be compared (monitored) every scan that the A element is closed regardless of the length of the matrices (maximum 1584 points).

23XX — Clear Bit or Sense Bit

NOTE

The matrix clear and set operations are useful to build matrices within the Controller as well as examine individual bits of any matrix and provide a coil reference for use as a control element in relay symbology.

This code, when the A element is closed, causes a single bit in a matrix to be cleared to zero (0) regardless of its previous value (either a 1 or a 0). The bit number is contained in the B element register, the matrix length in registers is defined by XX of the functional code, and the specific matrix location starts with the D element register.

The operation is performed continuously upon closure of the A element contact; a series of bits can be cleared by changing the value in the B element register with a table-to-register move or by using an input register in the B element. If the A element contact is NOT closed, this line will sense the state of the bit referred to by the contents of the B element register, but not alter it. The coil will be ON if it is a one (1) and OFF it it is a zero (0); the coil will *always* reflect the status of the B element bit, regardless of the state of the A element contact.

The coil status prior to closure of the A element contact will be that of the bit to be operated on and, after closure of the A element contact, the coil will be OFF (since the bit is to be cleared). When utilized as a sense line, an input register can be placed in the D element, since no alteration of this register is required. If an input register is utilized in the D element, closure of the A element contact will have no affect on either the coil status or the matrix contents. The B element register refers just to itself, and the D element register to XX registers — the size of the matrix. As an example, refer to Figure 80.

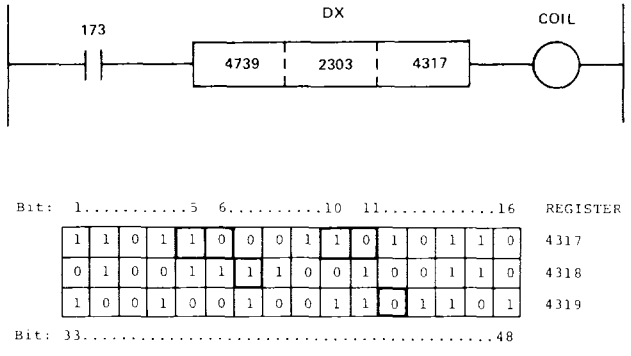


Figure 80. Sample Matrix Clear Line

Assume that line 173 is OFF and register 4739 contains the value 5. The coil will be ON, sensing the one (1) in the matrix at bit 5. When line 173 is ON, this bit in register 4317 will be cleared to zero (0) and the coil will be OFF. If the value in register 4739 is changed to 10, the tenth bit will be cleared to zero (0).

CAUTION

If an input register is to be used in the B element and is connected to thumbwheel switches, intermediate bits between 5 and 10 could be cleared if the A element contact remains closed while the input register is changed.

If the A element contact is referenced to a null data line, an input that is not utilized, or to an AND or OR logic line (i.e., some reference that will never be energized), this function can be used as a sense line. If line 173 of Figure 80 is an OR logic line (DX line with a 21XX functional code), and register 4739 contains a 23, the coil will be ON and no clearing operation is possible; if register 4739 contains a 44, the coil will be OFF.

If the contents of the B element register exceeds the number of bits in the matrix as defined by the functional code (or is zero), no operation is performed and the coil will be OFF regardless of the state of the A element contact.

24XX — Sense Bit or Set Bit

This code, when the A element is closed, causes a single bit in a matrix to be set to one (1) regardless of its prior value. Its operation is very similar to the 23XX clear function discussed above, except that the bits are set to one (1) in lieu of zero (0). Either functional codes, 23XX or 24XX, provide exactly the same sense functions.

For example, assume the functional code of Figure 80 is 2403, line 173 is OFF, and register 4739 contains the value 6. Initially, the coil will be OFF reflecting the status of the referenced bit. When line 173 becomes valid, bit 6 in register 4317 will be set to one (1) and the coil energized, again reflecting the status of the bit. If the value in register 4739 is changed to 11, bit 11 in register 4317 will also be set to one (1).

If the A element is referenced to a line or input that will not be energized, and register 4739 contains a 23, the coil will be ON and no setting operation is possible; if register 4739 contains a 44, the coil will be OFF.

25XX — Matrix Complement

NOTE

The complement capability is useful to prepare data for comparison when the data is exactly reversed from the source data (e.g., output matrix driving valves, compared to limitswitch inputs from valves, which close when the valve is de-energized).

This code, when the A element is closed, causes the contents of the matrix referred to in the B element to be complemented and the result placed in the D element matrix. Every bit in the B element matrix has its value reversed, zeros replaced with ones and ones replaced with zeros, and the result placed in the D element matrix. The entire matrix is complemented every scan that the A element is closed (passing power). The contents of the B element matrix is retained and the previous contents of the D element matrix is destroyed and replaced with the result of the complement operation. The coil has no significance, and is OFF in all cases.

It is possible to specify the same matrix in both B and D elements, resulting in the complement replacing the source matrix. However, if this is desired, the A element should be referenced to a one-shot to prevent the matrix from being complemented a second time during the next scan. As an example, refer to Figure 81.

The B element refers to three consecutive registers (4134-4136) as does the D element (4479-4481), since the function code has defined each

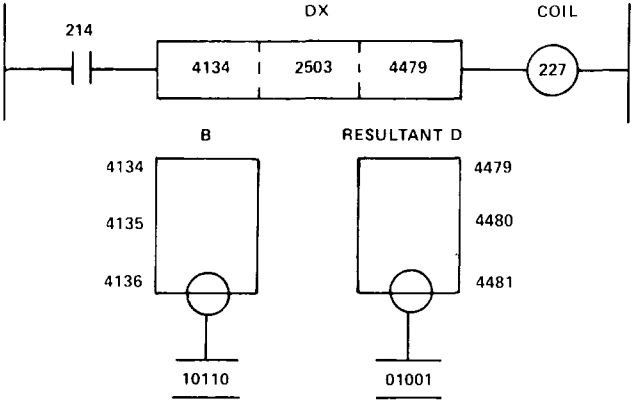


Figure 81. Sample Matrix Complement

matrix as three registers in length. In all cases, matrix COMPLEMENT will be accomplished on matrices of equal length. For this example, on the first scan that line 214 is energized, all bits in registers 4134-4136 will be complemented, and the result placed in registers 4479-4481. The contents in registers 4134-4136 will not be altered and the previous contents in 4479-4481 will be destroyed and replaced with the result of the complement operation. On subsequent scans, the complement operation will be performed as long as the A element passes power. As long as the B element matrix is not altered, the results in the D element matrix will not be altered from its initial value. Note the illustrated effect of five bits in register 4136 and their corresponding results in register 4481.

If the D element reference was changed to 4134 (same matrix referred to in both the B and D matrices), on the first scan 214 was energized, the 10110 in register 4136 will be replaced with 01001. On the next scan, assuming 214 remains energized, 4136 is again complemented, and 10110 (the original value) is placed in register 4136. Thus, every scan, the bits in each register will oscillate between ones and zeros; the result when 214 is de-energized cannot be guaranteed. To prevent this oscillation when using the same matrix in the B and D elements, the A element should be referred to a one-shot that is valid for exactly one scan. In all cases, the coil will not be energized.

26XX — Logical Exclusive OR of Two Matrices

NOTE

The exclusive OR is useful to compare two matrices in one scan and allow specific identification of those bits that miscompared at some future time. After an exclusive OR is performed, a one indicates a miscompare and a zero a compare. These miscompares can be located by comparing (22XX code) the result with a zero matrix (a matrix containing all zeros).

This code, when the A element is closed, causes the contents of the matrix referred to in the B element to be logically ORed (exclusively) with the contents of the matrix referred to in the D element, and the result stored in the D element matrix. This function operates similar to code 20XX and 21XX discussed previously, except the logical Exclusive OR is performed between the two matrices. Since an Exclusive OR is accomplished every scan the A element remains closed, oscillations can occur wherever the B matrix bits are ones; thus the A element references should *always* be to a one-shot. As an example, refer to Figure 82.

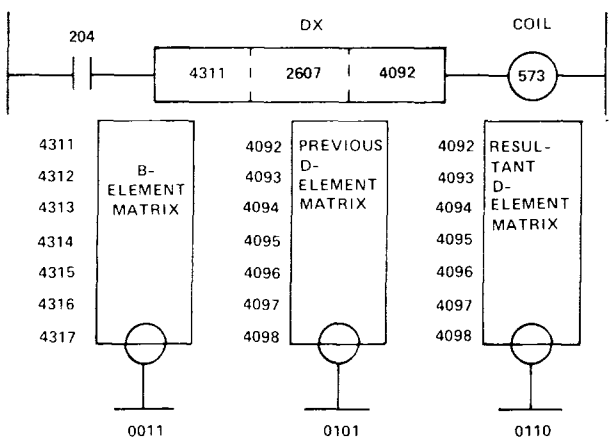


Figure 82. Sample Matrix Exclusive OR

The B element refers to seven consecutive registers (4311-4317) as does the D element (4092-4098), since the function code has defined each matrix as seven registers in length. In all cases, matrix Exclusive OR will be accomplished on matrices of equal length. For this example, on the first scan that line 204 is energized, a bit-by-bit Exclusive OR will be performed between registers 4311-4317 and registers 4092-4098. Note the specific example of four bits provided in registers 4317 and 4098. The result is a one only if there is a one in either the B element matrix or the D element matrix, but not (excluding) if they are both ones.

If line 204 remains energized on the next scan, another Exclusive OR is performed between the retained (not changed) contents of the B element matrix and the new D element matrix. The four bits in register 4097 will become 0101, the original values; to prevent this, line 204 should be a one-shot valid for exactly one scan. The coil on line 573 has no significance and is never energized.

27XX — Rotate Left

NOTE

The Rotate Left and Rotate Right are both useful to form single-bit retentive shift registers or to shift data as required for reformatting.

This code, when the A element is closed, causes the contents of the matrix referred to by the B element to be rotated one position to the left and the result placed in the D element matrix. All bits are shifted down one position, e.g., the status of bit 30 is placed in bit 29, 29 in 28, 17 in 16, 2 in 1, etc. The status of bit 1 is carried around (true rotate not just shift) and placed in the last bit of the matrix. The coil reflects the status of this bit carried around; the coil will be ON if the bit is a one, and OFF if the bit is a zero. For example, refer to Figure 83.

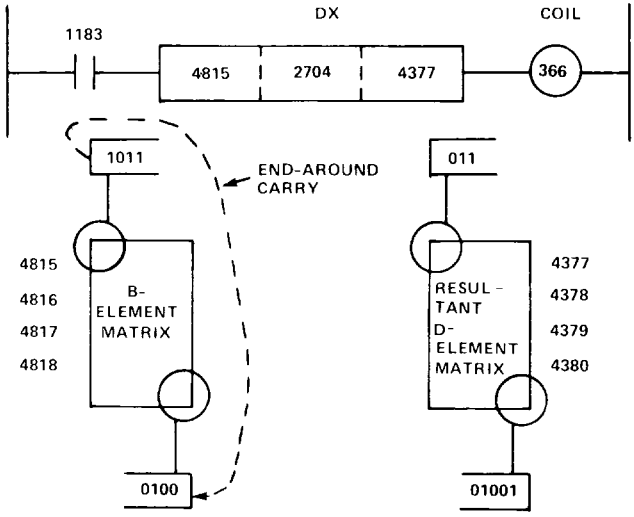


Figure 83. Sample Matrix Rotate Left

The B element refers to four consecutive registers (4815-4818) as does the D element (4377-4380), since the function code has defined each matrix as four registers in length. In all cases, matrix Rotate will utilize matrices of equal length. For this example, during the first scan input 1183 is energized, the contents of registers 4815-4818 will be rotated one position to the left and the result placed in registers 4377-4380; the contents of the B element matrix is not altered and the previous contents of the D element matrix is destroyed and replaced by the result of the rotate. The coil will be ON since the end-around carry bit was a one.

As long as input 1183 remains closed, the rotate will be performed and the coil status not altered, since on every scan the rotate goes back to the B matrix for its initial data. Note the specific status of bits 1-4 and 61-64 of the B element matrix. Bits 61-64 are shifted down to 60-63 and placed in the D element matrix; bit 1 of the B element matrix is placed in bit 64 of the D element matrix; bits 2-4 of the B element matrix are placed in bits 1-3 of the D element matrix.

28XX — Rotate Right

This code, when the A element is closed, causes the contents of the matrix referred to in the B element to be rotated one position to the right and the result placed in the D element matrix. This function operates similar to code 27XX except that the bits are shifted up, status of bit 1 into bit 2, 2 into 3, 16 into 17, 25 into 26, 32 into 33, etc., and the last bit end-around carried to bit 1. The coil is still set to the status of this end-around carried bit.

As an example, assume the function code in Figure 83 is 2804; the only difference in operation would be the specific results. Bits 2-5 of the resultant D element matrix would be 1011; bits 62-64 would be 010; bit 1 would be zero (obtained from bit 64 of the B element matrix); and the coil would be OFF.

A continuous rotate can also be obtained if the same matrix is referred to in the B and D elements of a 27XX or 28XX code. A calculate line can be used to control how long the rotate is performed, and thus how many bits are rotated.

NOTE

Function Code 29XX (384A and 384B only) is discussed in Section 3.7.

Additional 30XX References

Some of the executives that provide matrix capabilities also provide a capability referred to as Additional 30XX References. Input registers normally start at channel III and up to 16 are provided with any executive (3001-3016). However, it is possible to develop and executive that provides input registers in all four channels, maximum 32 input registers (3001-3032). Thus these additional references start at 3033, which has no possible meaning relative to the hardware I/O.

The contents of 3033 are controlled by the coil status of logic lines 1-16. If the coil of line one is ON, bit one in register 3033 will be a 1; if coil one is OFF, bit one will be a 0. The same technique is used to determine that status of bits 2-16 from coils 2-16. These additional references continue at 16 lines per register, until the last logic line is used. For example, if the MOPS provides 608 lines, the additional references used for coil status will be 3033-3070; bit 16 of register 3070 is controlled by line 608, the WDT line.

With additional references, the logic to drive discrete outputs (e.g., lines 1-256) can be built with simple relay logic; but their status is also available in registers 3033-3048 for monitoring by matrix compare lines. The additional references bridge the gap between discretes and registers.

The next additional reference after the logic lines (e.g., 3071) reflects the status of inputs 1001-1016. If input 1001 is energized, bit one in register 3071 is a 1; if 1001 is de-energized, bit one is a 0. These references continue at 16 inputs per reference until the limit of discrete inputs is reached

(e.g., 3071-3086 if 256 inputs are provided). Discrete inputs are thus available as discrete references and as a bit in a register for matrix operations. For an example of additional references, see Table 14 which assumes 608 lines and 256 discrete inputs.

EXAMPLE V — Monitoring of Discrete Inputs Versus Outputs

Assume discrete outputs 81-160 are used to drive 80 solenoid valves with standard relay logic. On each valve is a limitswitch that closes when the valve goes to the energized position. A monitoring system is required that constantly compares the outputs against the limitswitches to ensure all energized valves go to their proper position and that, when de-energized, the valves do not remain in the energized positions.

Table 14. Example Additional 30XX References (608 Lines, 256 Discrete Inputs)

Register	Contents Controlled by Lines	Register	Contents Controlled by Lines	Register	Contents Controlled by Inputs
3033	1-16	3052	305-320	3071	1001-1016
3034	17-32	3053	321-336	3072	1017-1032
3035	33-48	3054	337-352	3073	1033-1048
3036	49-64	3055	353-368	3074	1049-1064
3037	65-80	3056	369-384	3075	1065-1080
3038	81-96	3057	385-400	3076	1081-1096
3039	97-112	3058	401-416	3077	1097-1112
3040	113-128	3059	417-432	3078	1113-1128
3041	129-144	3060	433-448	3079	1129-1144
3042	145-160	3061	449-464	3080	1145-1160
3043	161-176	3062	465-480	3081	1161-1176
3044	177-192	3063	481-496	3082	1177-1192
3045	193-208	3064	497-512	3083	1193-1208
3046	209-224	3065	513-528	3084	1209-1224
3047	225-240	3066	529-544	3085	1225-1240
3048	241-256	3067	545-560	3086	1241-1256
3049	257-272	3068	561-576		
3050	273-288	3069	577-592		
3051	289-304	3070	593-608		

Assuming the limitswitches are connected to discrete inputs 1129-1208 for use in the relay logic, Figure 84 is an example of the logic that will perform this monitoring. Referring to Table 14, discrete outputs 81-160 control the contents of registers 3038-3042 and discrete inputs 1129-1208 control registers 3079-3083. Line 325 clears all five registers in the compare matrix (4487-4491) to zero every scan by ANDing them with zero. The zero matrix is loaded initially with zero, and never altered; thus it always contains known zeros unless action is taken to alter its contents. Line 326 moves the contents of registers 3079-3083 (inputs 1129-1208) into the compare matrix by ORing them with a known zero placed in the matrix by line 325. This is required since an input register (3079-3038) cannot be placed in the D element of a DX compare line. Line 327 does the actual comparing between the outputs (3038-3042) and the inputs (loaded into 4487-4491). Coil 327 will be ON if any miscompares are detected and the contents of 4486 will be the bit number (valve) that caused the miscompare.

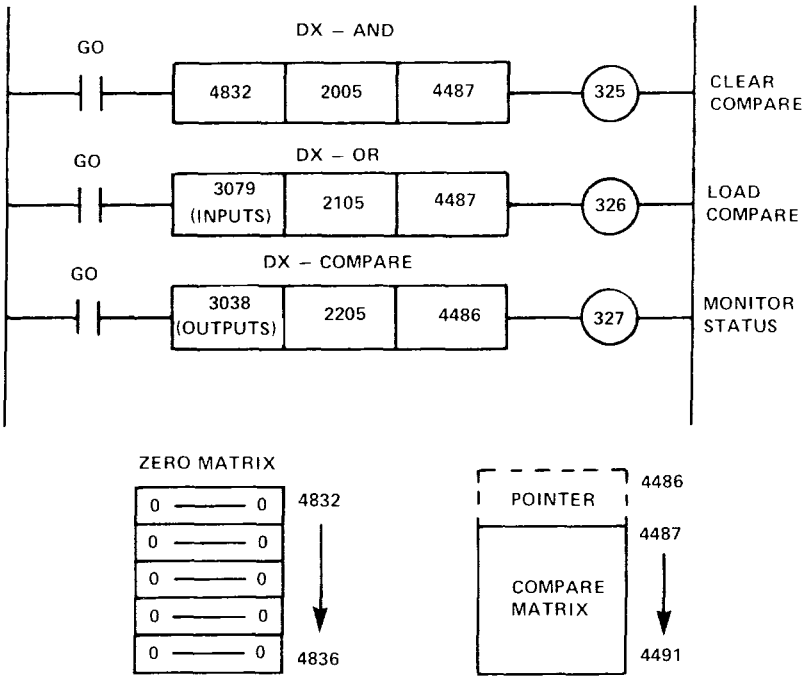


Figure 84. Example: Monitoring of Discretes

Since the matrix operations are accomplished every scan in their entirety, all inputs are compared once a scan until a miscompare is detected; one miscompare is detected every scan. The GO contact controls when the compare is enabled and can be a normally-closed contact reference to null data if constant comparisons are required. Note that the comparison is not affected by whether the output is energized or de-energized; if it does not compare (i.e., energized output and open limitswitch, or de-energized output and closed limitswitch) coil 327 will be ON. Timers can be incorporated such that a miscompare must be detected for a continuous time period (e.g., 0.5 seconds, 2.0 seconds, etc.) before action is taken. If more or less valves or other devices are to be monitored, the only action required is to change the DX function codes to adjust the size of the matrices; these three logic lines (9 words of core memory) perform the monitoring function for one to 99 registers of data (16 to 1584 devices).

If a second limitswitch is incorporated on each valve that is closed when the solenoid valve is de-energized, and open when it is energized, another comparison can be made to detect valves that "hang up" between limits. Lines 325 and 326 are duplicated with a single complement line (25XX) that will take these limitswitch inputs, complement their status, and place the result into a COMPARE matrix. A complement is required so that direct comparisons can be made to the output lines. For example, if an output is energized, a one is placed into the matrix starting at 3038; however, since these new limitswitches are open when the valve is energized (opposite of first limitswitches), a zero is placed into their input matrix. Direct comparisons will result in a miscompare whenever the valves are operating properly. To correct this, the inputs from these new limitswitches are first complemented, then direct comparisons are made and miscompares only occur when malfunctions are detected.

Various actions are possible when a miscompare is detected, as with a fault diagnostic system. The malfunction can be printed on the P500 Printer; operator alerted by lights, bells, displays, etc.; process shut down or next step not allowed; machine forced to a safe condition; etc. Which action is selected depends upon the specific application and system design requirements. Two items are available to support whatever action is taken; the compare coil (e.g., line 327) will be energized and the pointer (e.g., register 4486) will contain the bit number (related to the specific valve) that caused the miscompare.

EXAMPLE VI — Matrix Retentive Shift Register

The matrix rotate capability can be utilized to develop a retentive shift register, driven by basically three logic lines (9 words of core memory), up to 1584 stages. Either rotate function (27XX-left or 28XX-right) can be used with equal efficiency; Figure 85 illustrates a shift register built using the 2803 DX line to form a 48-station shift register. Line 497 shifts the contents of the three-register matrix, which starts at register 4757, one position to the right and places the result back into the same registers. To ensure that only one shift is made, the shift input should be a one-shot. Since the rotate function also takes the status of the last bit (e.g., 48th bit) and places it into the first bit, line 498 will always clear the first bit whenever the end-around carry is a one (coil 497 energized). Line 499 places one in the first stage of the shift register whenever commanded by the input signal. If a one is not placed in this stage by the time the next shift is performed, a zero is shifted into the shift register.

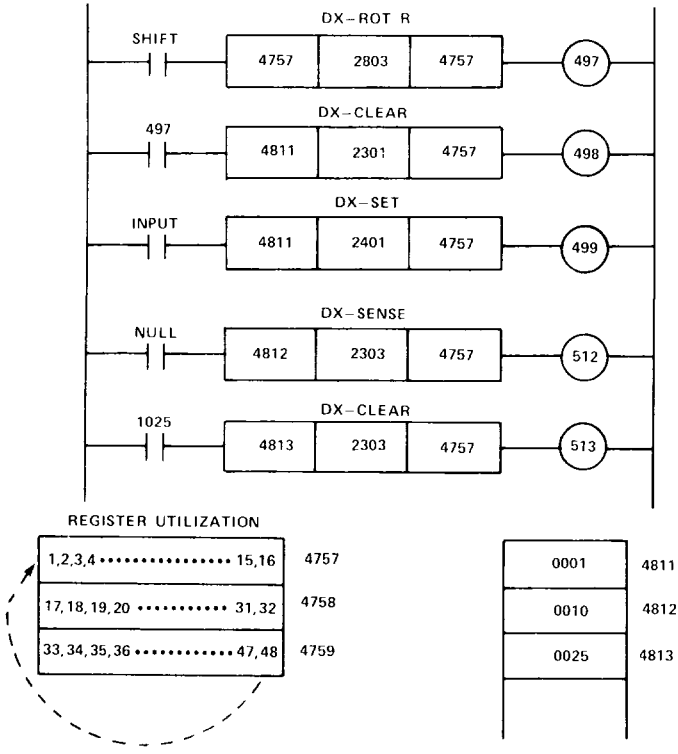


Figure 85. Example: Matrix Shift Register

Coil 497 can represent the output of the shift register if this output is required for only one scan. Otherwise, a sense line similar to line 512 can be utilized for continuous output status. In addition, a sense line can be provided to obtain the status of any stage of the shift register. Line 512 currently provides the status of the tenth stage since register 4812 contains a ten; if register 4812 contents are altered, another stage's status is obtained. Coil 512 is ON for a one in the stage and OFF for a zero. Parallel entry of any stage can also be provided with clear and set lines similar to line 513. This line is currently designed to clear stage number 25 when input 1025 is energized, since register 4813 contains a 25; a similar line with a 2403 code can be utilized to set any stage to a one. It is to be noted that lines 497-499 are the basic drive for this shift register; lines similar to 512 and 513 can be added if the application requires. If more than 48 stages are required, the DX code in line 497 is increased to provide 16 stages per additional register; a DX code of 2899 provides 1584 stages. Since the shift register is built in holding registers, it is retentive upon power failure.

3.5.3 Extended Arithmetic (Group 3YXX)

These codes provide the extended arithmetic capability so that multiplication, division, and other special functions can be accomplished. The resultant product of a multiply operation and the dividend prior to a divide operation are stored and referred to within memory as double-precision. This implies that the magnitude may exceed the capability of one register and thus two registers are allotted to store this value; one register contains the four high-order digits (tens of millions through tens of thousands) and the next register in sequence contains the four low-order digits (thousands through units). This concept is important since the result of a multiply must be evaluated from the contents of both registers, and preparation for division must include loading two registers.

When either a multiply or divide operation is performed, the coil will come ON to indicate successful accomplishment. The following are valid functional codes currently available with the extended arithmetic capability. All 184 executives with this capability are provided with the four multiply/divide codes; only selected executives also provide the special functions (see Table 12). All 384 controllers are provided with the four multiply/divide codes; the 384A and 384B are provided with all eight codes (Basic plus Special).

Basic		Special	
3000	General Multiply	34XX	PID (Seconds)
3100	General Divide	35XX	PID (Minutes)
32XX	Multiply by XX	36XX	SORT (Ascending)
33XX	Divide by XX	37XX	SORT (Descending)

The multiply/divide functions are performed only once on closure of the A element contact and are completed prior to commencing operation on the next line of logic.

3000 — General Multiply

This code causes the contents of the B element register to be multiplied by the contents of the register immediately following the B element register, and the resultant product deposited as a double-precision number in the D element registers. The multiply is accomplished only on transition of the A element from OFF to ON; the coil will come on at the completion of this line. Both the B element and the D element refer to two consecutive registers; their contents are entered and read as four BCD digits.

For example, refer to Figure 86 and assume input 1035 is not energized.

If the input registers contain the values 976 and 42 as illustrated, on the first scan that input 1035 is energized, the product (40,992) of 976 and 42

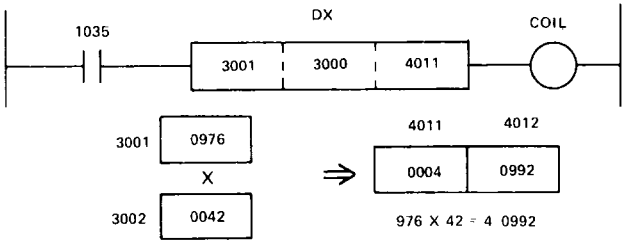


Figure 86. Sample Multiply

will be deposited into registers 4011 and 4012 as a double-precision number, and the coil energized. Note that the resulting product is separated with the four low-order digits being placed in the low-order product register (D element reference plus one) and the high-order digits (with leading zeros) placed in the high-order product register (D element reference). If the input registers change their values, input 1035 must be cycled ON-OFF-ON to produce a new product. The coil will be ON after the product is available and will remain ON until input 1035 is de-energized. Either or both B element registers can be zero.

NOTE

There are no conditions under which the coil of a multiply will not come ON as long as the A element contact is closed (passing power).

3100 — General Divide

This code causes the contents of the B element registers (double-precision) to be divided by the contents of the register referred to by the B element plus 2. The resultant quotient is stored in the D element register, with any remainder stored in the D element register plus 1. The divide operation is performed only when the A element contact is closed; its results are available prior to commencing the following line of logic.

The B element register refers to three consecutive registers and the D element register to two consecutive registers; their contents are entered and displayed as four BCD digits. The coil will come ON only if the operation is successful. Division by zero (0) or a resultant quotient too large (greater than 9999) to be stored in one register are both valid reasons for the coil to remain OFF; if the coil is not ON, the contents of any register will not be altered.

For example, refer to Figure 87 and assume line 413 is not ON.

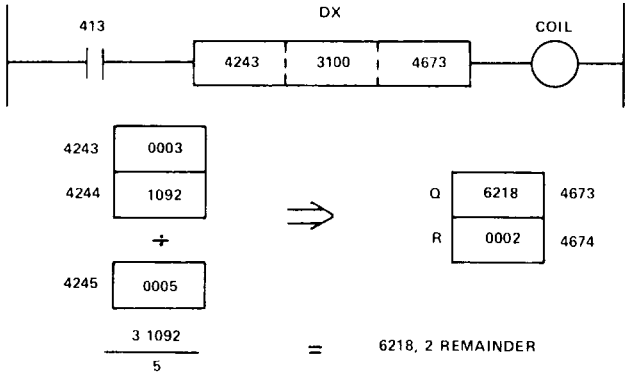


Figure 87. Sample Divide

If registers 4243 through 4245 are preloaded as indicated, on the first scan that line 413 is on, the double-precision number (31,092) in registers 4243 and 4244 will be divided by the contents (5) in register 4245. The resultant quotient (6218) will be stored in register 4673 and any remainder (2) in register 4674; the coil will be ON until line 413 is turned OFF.

If the divisor in register 4245 was a 3, the resultant quotient would be 10,364 which is too large to fit into one register; thus the coil would be OFF and the contents of all registers retained. If registers 4243-4245 change their values, a new division will take place only after line 413 is cycled ON-OFF-ON.

32XX — Fixed Multiply

This code causes the contents of the B element register to be multipied by a fixed quantity (XX), which is part of the functional code, and the resultant product is stored as a double-precision number in the registers referred to by the D element reference. The operation of this code is very similar in all respects (including coil status) to the general multiply functional code 3000 discussed previously, except the second B element register is replaced by the last two digits of the functional code.

Two limitations should be clearly understood; the multiplier must be only two digits in magnitude (99 or less excluding zero) and the multiplier cannot be altered by other logic operations. If either of these features are required, the general multiply code (3000) must be used.

For example, if the functional code of Figure 86 was 3242, the same result would be stored in registers 4011 and 4012; register 3002 is not required and can be used for another operation. Note that, for the 32XX code, the B element refers to only one register and the D element to two registers.

33XX — Fixed Divide

This code causes the double-precision contents of the B element registers to be divided by a fixed quantity (XX) which is part of the functional code, and the quotient stored in the D element register, with the remainder in the D element register plus 1. The operation of this code is very similar in all respects (including coil status) to the general divide functional code (3100) discussed above, except that the third B element register is replaced by the last two digits of the functional code.

There are two limitations. The divisor must be magnitude 99 or less, but not zero, and it cannot be altered by other logic lines. If either or both of these features is required, the general divide instruction (3100) must be used. For example, if the functional code of Figure 87 was 3305, the same results would be stored in registers 4673 and 4674; register 4245 is not required. Note that, for the 33XX code, both the B and D element refer to two registers.

3400 — PID (Proportional, Integral, Derivative) Control

This type of control is normally used for temperature, pressure, pH, etc., and other variables usually controlled by analog controls. In addition, positioning and servo loops are also ideal applications for this type of control.

The controller solves the following equation to provide the control action desired:

$$P = K_1 e + K_2 \int_0^t e\,dt + K_3 \frac{de}{dt} + K_4$$

where,

P = Output

K_1 = Gain (00.01—99.99) — proportional band is equal to $1/K_1$

K_2 = integral rate, or reset rate (0.001—9.999) repeats per *second*

K_3 = derivative time or the rate time (0001—9999) *seconds*

K_4 = Initial starting point

e = error = input minus set point

PROGRAMMING PID CONTROL

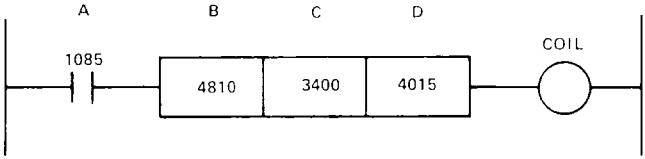


Figure 88. Sample PID Logic Line

A: Starts control when closes.

B: First register in input table (see below for input data).

C: 3400 is function code for PID control.

NOTE

Function codes 3401, 3500, and 3501 (PID-384A and 384B only) are discussed in section 3.7.

D: First register in output table (see below for output data table).

Input Table Description

Register Number (Typical)	Value in Register (Typical)	Description
4810	0517	Input (i.e., could be 517°F)
4811	0525	Set Point (i.e., could be 525°F)
4812	0025	K_1 — Gain
4813	0.050	K_2 — integral rate in repeats per *second*
4814	0000	K_3 — derivative time in *seconds* (derivative action not wanted so zeros used).
4815	0550	High limit of 550°F
4816	0500	Low limit of 500°F

NOTE

Seven sequential registers starting with the register used in the B element are set aside for PID operations when code 3400 is used. Therefore, do not use those register numbers for other functions.

Output Table (6 Registers)

Register 4015 contains the output values which can be used to position control valves or servos, etc. This output is a four-digit numerical quantity that will vary between limits established by 4815 and 4816 in this example. Maximum range of output is 0000 to 9999.

NOTE

Five register numbers following the register used in D (in this example 4016—4020) are used for internal use in PID control in the controller and cannot be used anywhere else in your program.

Additional Application Notes

1. This PID control is *direct acting*. If *reverse acting* is desired, a (B−C) calculate line can be used.
2. *Profile Control* (predetermine a non linear set point program): Can be very easily done by using a table to register DX move, storing up to 99 different set points. Linear ramp control of set points can be done in the same manner.
3. *Cascade Control:* The output of one PID line in the Controller can be the input to another PID line within the Controller by using a (B+C) calculate line to move the output value in the output register (in this example 4015) to the input register in another PID line.
4. *Adaptive Control:* The gain, reset rate and derivative time can each or all be changed at will by (B+C) calculate lines or DX transfer operations. Therefore, adaptive control can be used with no extra hardware by simply using additional programs in the controller.
5. *Feed Forward Control:* Can be accomplished very similar to adaptive control discussed above with additional programmed techniques to do the arithmetics required.
6. *High-Low Limits:* The coil of the PID line will be on when the limits are exceeded.

NOTE

Function codes 36XX and 37XX (384A and 384B only) are discussed in section 3.7.

3.5.4 Entering a Data Transfer Line

1. Set the Line Number on the LINE NUMBER switches.
2. Put the Controller MEMORY PROTECT switch in the OFF position.
3. Press the DATA TRANSFER pushbutton. 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 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 DISPLAY will show the number that has been entered.

NOTE

On 184 controllers, the Data Transfer lines can be programmed in any element order. However, the 384 controller requires the Function Code (C element) to be entered prior to programming either the B or D elements to establish proper error checking. This requirement is for DX lines only.

7. Select the B element position and enter on the REFERENCE NUMBER thumbwheels the register which is the location of the SOURCE data. Press any ELEMENT TYPE pushbutton. The REFERENCE DISPLAY will show the register that has been entered.
8. Select the C element position and enter on the REFERENCE NUMBER thumbwheels the FUNCTION CODE. Press any ELEMENT TYPE pushbutton. The REFERENCE DISPLAY will show the code that has been entered.
9. Select D element position and enter on REFERENCE NUMBER thumbwheels the register which is the DESTINATION of the data.
10. Press any ELEMENT TYPE pushbuttons. The REFERENCE DISPLAY will show the register that has been entered.
11. If the DISABLE pushbutton is lit, and not specifically desired, press it to turn it OFF.

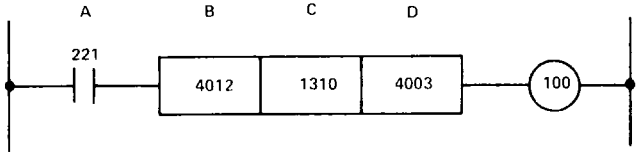


Figure 89. Data Transfer (Move) Line

1. Set Line Number switches to 0100.
2. Press DATA TRANSFER pushbutton.
3. Press A element pushbutton.
4. Set REFERENCE NUMBER switches to 0221.
5. Press Normally-Open Series ELEMENT TYPE pushbutton.
6. Press B element pushbutton.
7. Set REFERENCE NUMBER switches to 4012. Press any ELEMENT TYPE pushbutton.
8. Repeat steps 6 and 7 for the C element (1310), and D element (4003).
9. If the DISABLE Light is lit, and not specifically desired, press it to turn it OFF.

3.6 P500 PRINTER/D285 DISPLAY UNIT

3.6.1 Introduction

The P500 Printer (see Figure 90) is an industrial-environment hardcopy printer designed to provide data on the plant floor. It is capable of printing out management information such as number of parts produced, up times, efficiencies, recipe contents, etc.; or operator information such as error messages, batch completed notation, manual operations required, etc. The P500 Printer is not designed to document the user's program with ladder diagrams; this support is available from the Service Center (see 4.2.2).

The paper used in the P500 is pressure sensitive; thus the printer does not require carbon paper or ink and the associated maintenance problems. The paper is 3½ inches wide, each page approximately 5½ inches long— a convenient size to place in shirt pockets. Each page can contain up to 20 lines, each line 21 characters (see Figure 91). A summary of P500 specifications are provided in Table 15.

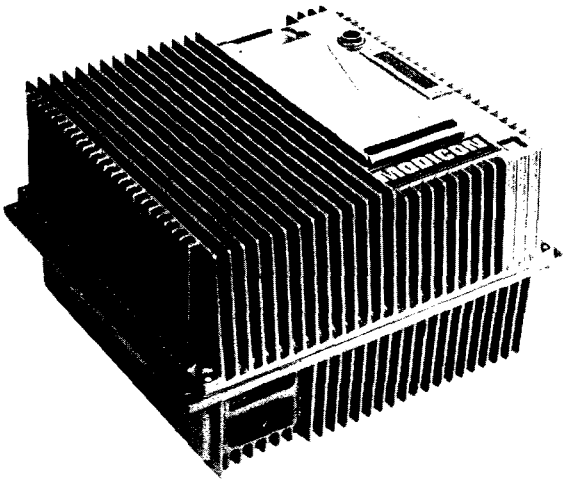


Figure 90. Programmable Printer

```
                SAMPLE

387621  14  783  9342


  DOWN  TIME  REPORT


PARTS  THIS  SHIFT  5673
EFFICIENCY     92.9%


STATUS  OF  MACHINE  617
   SEQUENCE  COMPLETE


FAILURE  OF  CARRIAGE
TO  ELEVATE  DURING
LOAD

CHECK  I/O  R  LIGHTS


CHARACTER  SET

ABCDEFGHIJKLMNOPQRSTU
VWXYZ  0123456789  ¢\>(
-:"£$%@'()*+^_./:;<=>
?
```

Figure 91. Sample Printout

Table 15. P500 Printer Specifications

Power Requirements	115 Vac ± 20%, 60 Hz (Standard) 115 Vac ± 20%, 50 Hz (Optional) 250 Volt-amps
Environmental:	
Ambient Temperature Humidity	0°C to 60°C 10% to 70% (non-condensing)
Dimensions:	19 in. x 11 in. x 17½ in.
Weight:	60 lb. (approx.)
Read-Only Memory:	108 lines of stored messages; each line contains 21 characters
Print Speed:	2 lines per second
Paper:	3½ in. wide, pressure-sensitive, fan-fold: 20 lines per page; approx. 1600 pages per box
Control I/O Level:	115 Vac or 24 Vdc
I/O Cable (supplied with P500):	25-wire, standard 12-feet long (Type W500)
Optional -Interfaces:	Parallel BCD or ASCII Serial, EIA RS-232C compatible
Model Variations:	P500-100 (115 Vac I/O, 60 Hz power) P500-200 (115 Vac I/O, 50 Hz power) P500-300 (24 Vdc I/O, 60 Hz power) P500-400 (24 Vdc I/O, 50 Hz power)
Mounting:	Standard horizontal (table-top) or optional vertical (wall-mounting)

Within the printer is a Programmable Read-Only Memory (PROM). This PROM is capable of storing 108 lines of preformatted alpha-numeric messages. These messages are programmed in accordance with the user's specification provided at the time of manufacture. The PROM can be changed by the MODICON factory. Changes are made on an exchange basis; a new message sheet is provided to MODICON, and a revised PROM (either just one chip or the entire memory) is sent to the customer. After installation, the old memory must be returned to the factory for reuse.

The P500 Printer is a general-purpose industrial unit capable of being driven from a computer, relay panel, or another controller as well as the MODICON 184/384 Controller if properly interfaced. When connected to the 184/384 Controller, the overhead control required to establish and store messages awaiting servicing by the printer, as well as providing numerical data to the printer, is all automatically handled by the DX Print capabilities.

Up to 16 printers can be connected to one 184/384 controller through the I/O and any number of logic lines can be used to drive separate messages to these printers. One printer is serviced at a time with messages in the real-time order they were energized within the Controller. Each printer connected to a 184/384 Controller requires an output register to provide data and commands to the printer, and two individual inputs to accept signals (Busy and Form Busy) from the printer. All data provided to the printer requires a positive response from the printer before another command or number is provided. This technique requires communication both ways and is called "handshaking" between the Controller and the printer.

The printer requires 115 Vac power locally to drive the print head, paper advance, DC power supply, etc., in addition to a 25-wire cable to connect to controller I/O. The AC power can be either 50 or 60 Hz and the I/O can be either 115 Vac or 24 Vdc, thus there are four models of printer available (see Table 15).

Table 16. Standard Character Set

A	Q	5	+
B	R	6	'
C	S	7	—
D	T	8	.
E	U	9	/
F	V		:
G	W	!	;
H	X	"	<
I	Y		=
J	Z	$	>
K		%	?
L	0	&	†
M	1	'	[
N	2	(	\
O	3	)	]
P	4	*	

Control Codes

Eb End of Block — indicates end of message to be printed.

Ff Form Feed — advance paper to next page; will always be executed prior to printing line in which Ff is located, regardless of Ff location in line.

Lf Line Feed — prints blank line; must be first character of a blank line in the PROM; nothing else can be programmed into this blank line.

Vd Variable Data — blanks to be filled in by data from the controller; one digit per blank.

The D285 Display Unit provides CRT display of messages using exactly the same handshaking as the P500 Printer. Any 184 or 384 controller that communicates to a P500 Printer can also control the D285 Display Unit.

3.6.2 Formatting Messages

Each printer's PROM is loaded with messages in accordance with each customer's requirements. Table 16 is a list of characters available with the standard P500. Each of the 108 lines (numbered 0 to 107 in the PROM) has space for 21 characters; Figure 92 provides a sample form completed to indicate a portion of one customer's requirements. Figure 93 shows sample forms that can be used to specify the contents of the printer PROM.

All print commands are originated within the Controller by Data Transfer (DX) logic lines. When these commands specify a message stored within the printer's PROM, they utilize the first line of the message as the identification for that message. The printer begins at the line specified and will continue printing until an End-of-Block (Eb) character is detected, which terminates the print of that message. All 21 characters in a line are printed simultaneously.

The location of the Eb characters is determined by the customer when completing the PROM specification (Figure 93); they are used to combine lines forming messages of multiple lines in length. Since only lines 0 to 99 are individually addressable, line 100 through 107 are used as a single continuation of a message that began earlier. Thus, typically, the longest message is placed at the end of the PROM to make maximum use of these continuation lines.

Message Address	COLUMNS 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
72																						P
73																						R
74																						O
75																						M
76																						
77																						6
78																						
79																						
80																						
81																						
82																						
83																						
84																						
85																						
86																						
87			P	R	O	D	U	C	T	I	O	N		S	U	M	M	A	R	Y		P
88				Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd		1	9	7	4					R
89				T	I	M	E		Vd	Vd	Vd	Vd	H	Vd	Vd	Vd	Vd	Vd	M	I	N	O
90				S	H	I	F	T		N	O		Vd									M
91	Lf																					
92	M	A	C	H	I	N	E		N	O		Vd	Vd									7
93	P	A	R	T	S		T	H	I	S		S	H	I	F	T		Vd	Vd	Vd	Vd	
94	E	F	F	I	C	I	E	N	C	Y		%			Vd	Vd		Vd	Vd		Eb	
95																						
96																						
97	Ff			D	O	W	N	T	I	M	E		S	U	M	M	A	R	Y			
98	T	Y	P	E			O	C	C	U	R			M	I	N	U	T	E	S		
99													C	U	M			W	A	I	T	P
100	T		C	H	G			Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd	R
101	M	-	R	E	P			Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd	O
102	E	L	E	C				Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd	M
103	L	O	A	D				Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd	
104	U	N	L	O	A	D		Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd	8
105								-	-	-	-		-	-	-	-				-		
106	T	O	T	A	L	S		Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd		Vd	Vd	Vd	Vd	
107	Ff	Eb																				

Figure 92. Sample P500 PROM Specification

Message Address	COLUMNS	
00		PROM 0
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		PROM 1
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		PROM 2
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		

Figure 93. Blanks for Specifying P500 PROM Messages

Message Address	COLUMNS	
36		PROM 3
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		PROM 4
49		
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		PROM 5
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		

Figure 93. Blanks for Specifying P500 PROM Messages (Cont)

Message Address	COLUMNS	
72		PROM 6
73		
74		
75		
76		
77		
78		
79		
80		
81		
82		
83		
84		PROM 7
85		
86		
87		
88		
89		
90		
91		
92		
93		
94		
95		
96		PROM 8
97		
98		
99		
N/A		
N/A		
N/A		
N/A		
N/A		
N/A		
N/A		

Figure 93. Blanks for Specifying P500 PROM Messages (Cont)

In many applications, blanks are required in the message format where numerical data from the Controller's memory is to be placed. These blanks are indicated by a special character called Variable Data (Vd) and can be used to indicate where the number of parts produced, efficiencies, up times, machine identification for diagnostics, etc., are to be printed. The location of these blanks are again per the customer's specifications (Figure 93). When the printer detects a Vd character, it requests the numerical digit (0-9) from the Controller. The Controller obtains the numerical data from registers specified in the DX print command.

Since the high-order digit (thousands digit) is provided to the printer first, most variable data blanks are left in groups of four (one register's content) to prevent restructuring of the data prior to transmission to the printer. For example, assume a blank is left for a machine number whose magnitude will be 1 to 20 (two digits maximum) and a blank is only two digits wide. If a counter is providing the machine number, the magnitude 0001 through 0020 is available in a register. However, when this register is used to fill in the blanks, the first two characters will always be zero, since the thousands and hundreds digits from the register are utilized first. This problem can be solved by either leaving four blanks so that the entire register's content is printed, or to count by hundreds with a calculate line in lieu of units as a counter does.

Two other special codes are available. The first is Line Feed (Lf), which causes a blank line to be printed (a line is skipped); Lf must be the first character of a blank line in the PROM. No other characters such as Eb, Ff, A, B, C, 3, 4, or another Lf can follow this first character. The second is a Form Feed (Ff), which causes the paper to be advanced to the top of the next page *prior* to printing the line in which Ff appears. Ff can be used to start a message on the top of a new page or force the paper up for removal after a message has been printed. Ff can occur at any character location in line; it is still executed prior to that line being printed.

All control codes occupy only one character of the 21 characters in a line, although it requires two letters to specify each code. Codes Eb, Lf, and Ff are replaced by blanks when the line is actually printed.

3.6.3 Programming DX Lines

This section applies to the Data Transfer Print capability available with some 184 MOPS 3 level execitives; see Table 12 for a list of MOPS 3 executives that provide print capabilities. All 384 controllers have DX print capabilities. These DX lines provide a simple method of outputting alpha-numeric data via the P500 Printer from the Controller. The general form of a DX line is still applicable (see Figure 70).

The A element, when closed, activates the print line and places the message in the print queue; this element has to be closed for only one scan. If the A element is cycled ON-OFF-ON while that particular print line is still in the queue (not completed at the printer), the second command will be ignored. If the A element remains closed, no additional commands will be generated when the print is complete; the A element must be cycled OFF-ON to enter a second print command after the previous print is completed.

The B element is a register, either holding or input, in which is placed the data to be printed if variable data is required by the print command. If more than four characters are required, the registers following the B element register will be used in sequence until sufficient characters are obtained.

NOTE

Characters are taken from the high-order digit first.

The C element is the DX code starting with the digit 4 (4YXX type); each code is discussed later in this section. The D element is *the* output register to which the printer is connected.

NOTE

Output registers programmed in the D element of a DX print line will automatically be permanently coded binary in the I/O Allocation Table, unless the Table is modified (see 4.1.4).

The coil on DX print lines will be energized as soon as the A element is closed (print command entered into the queue) and remains ON until the message is completed at the printer. The status of the coil is not affected by the condition of the A element, only by the status of the message in this print queue. Data in the B element table should be loaded prior to closing the A element, and not changed until the coil is de-energized. The following are discussions of the specific DX print codes available.

40PL — Print Numerical Data

NOTE

The Numerical Print capability is useful to print out report data in columns with the 41XX code providing the title and headings of the report. Do not waste PROM storage to specify format of purely numerical data.

This code causes only numerical data stored within the Controller to be printed in a specific format when the A element contact is closed. The B element register is the source of the data to be printed; if more than four digits are required by the format specified, successive registers will be utilized until the required data has been obtained. Data is supplied to the printer with the most-significant digits first.

The data format is specified in the functional code by the characters PL; P for page definition, and L for line content — see Table 17. The register in the D element identifies which output register the printer is connected to. The coil will be energized when the print line is activated and remain energized until the data is printed. As an example, see Figure 94.

Assume registers 4032-4035 contain the values 1234, 5678, 9012, and 3456, respectively. When input 1132 is energized, the printer will print one line of data and then one line feed (per page format 1 of Table 17); the line will contain four registers of data (per line format 4 of Table 17). The coil will be energized and remain ON until the print is accomplished regardless of the status of the A element contact; additional print commands of this line, while the coil is energized, will be ignored. If the data in the storage registers is changed after the A element is closed, but prior to the print being completed, the revised data may or may not be printed. Note that the B element register refers to four registers, depending on the page and line format specified in the functional code. The D element register refers only to itself.

41XX Print Specified Stored Message

NOTE

The Stored Message print capability is useful to provide report data or messages to the operator.

This code causes a message contained within the printer to be printed on closure of the A element contact. The B element register contains the numerical data (if any) to be printed in locations specified by the code Vd (variable data) in the message format; if more than four digits are required, the next register immediately after the B element register will be used. Data will be utilized with the most-significant digit first. Thus, if only one Vd code is placed in the message, the thousands digit of the register will be used. To simplify programming, whenever possible, blanks in the message should be designed as four digits in length.

The message number to be printed is specified by the last two digits (XX) of the functional code; these messages can be 00 to 99 inclusive. The message to be printed will begin at the line number specified (XX) and continue until an end-of-block (Eb) code is reached within the PROM. The D element register is the output register to which the printer is connected. The coil will remain ON until the complete message has been printed.

Table 17. P500 Printer Data Format

40PL

P = Page Format:

- 0 = Print 1 line
- 1 = Print 1 line, line feed
- 2 = 12 Line feeds, print 1 line, Form feed
- 3 = 11 lines, print 2 lines, Form feed
- 4 = 10 Line feeds, print 3 lines, Form feed
- 5 = 9 Line feeds, print 4 lines, Form feed
- 6 = 10 Line feeds, print 1 line, Line feed, print 1 line, Form feed
- 7 = 8 Line feeds, print 2 lines, Line feed, print 2 lines, Form feed

L = Line Format:

- 1 = XXXX
- 2 = XXXX XXXX
- 3 = XXXX XXXX XXXX
- 4 = XXXX XXXX XXXX XXXX
- 5 = XXXXXXXX XXXX
- 6 = XXXXXXXX XXXXXXXX

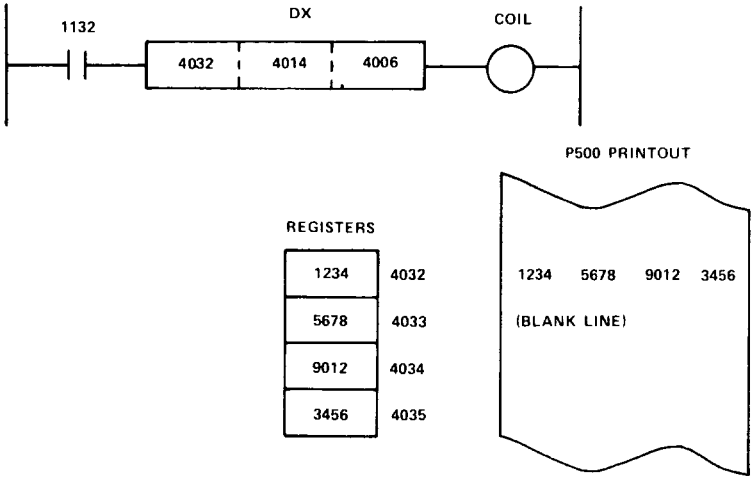


Figure 94. Sample Fixed Format Print

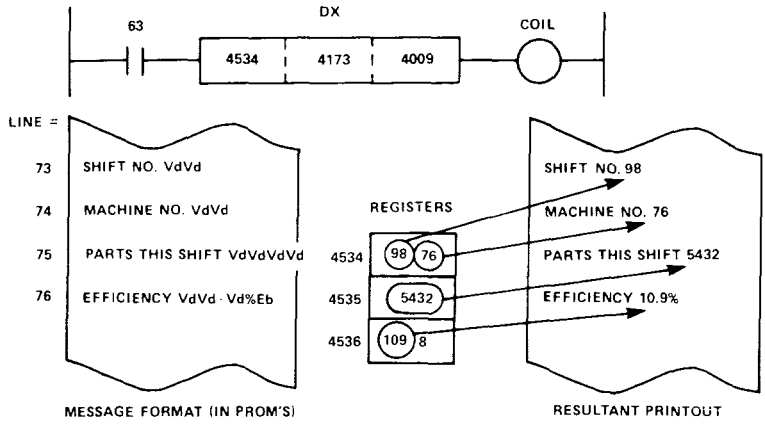


Figure 95. Sample PROM Format Print

As an example, refer to Figure 95. If registers 4534-4536 contain the data specified, when logic line 63 comes ON, message 73 will be printed with variable data blanks replaced by the contents of these registers. The coil will be ON until the message is actually printed out. If another line specified printing message 74, the printout will begin with line 74 (Machine No.) and continue to line 76. If logic line 63 remains ON at the completion of message 73, a second printing will NOT be commanded.

Note that the B element register refers to as many sequential registers as necessary to complete the variable data required by the message specified by the functional code and that the D element register refers only to itself.

4200 — Print Variable Stored Message

NOTE

The Variable Message Print capability is useful to print out error messages calculated by the Controller or messages whose addresses vary (e.g., dates printed out).

This code causes a message number contained in the B element register to be printed on the closure of the A element contact. This code operates very similar to the 41XX functional code previously discussed, except that the B element register contains, in the two least-significant digits, the message to be printed. Variable data, if any, will be obtained from the registers immediately following the B element register.

For example, assume Figure 95 contained, in the B and C elements of the logic line, the register 4533 and the functional code 4200, respectively. If register 4533 contained the value YY73 (where YY are ignored), message 73 will be printed with the data obtained from registers 4534 through 4536. By utilizing this instruction, the Controller can calculate and control both the message number and variable data to be printed.

NOTE

Procedures for entering Data Transfer lines, including DX Print lines, are provided in Section 3.5.4.

3.6.4 Connecting Printer to 184/384 Controller

Each printer connected to a Controller requires an output register and two single inputs (discrete or register) to provide communications both ways. The connections are made via a 25-wire cable (type W500, standard length 12 feet). This cable is plugged into the printer and has each of its wires on

the opposite end (Controller end) labeled with a number from 1 to 25. Table 18 summarizes the functions and connections for these wires when used with the 184/384 Controller.

The Busy and Form Busy signals are very important since they are the only method the printer has of communicating to the Controller. Whenever the printer is busy performing a function (manual paper advance, printing numbers, printing messages, etc.), the Busy signal is ON. When the printer is using its PROM for formatted messages, the Form Busy signal is ON.

If numerical data is required by the printer to complete its format (e.g., variable data blanks detected in the PROM), the Form Feed remains ON and the Busy is turned OFF.

Table 18. Printer Output Register Connections

Wire No.	Function	Printer Pin	B230 or B232 Terminal	
1	Form Select 00	A	11	NOTE: On B230 Output Module only, add jumpers between terminals 2 & 7, 7 & 12, and 12 & 17. Apply power to terminals 1 & 2 as indicated. Terminal 14 is NOT connected. Output will always be Hi for B230 only.
2	Form Select 01	B	10	
3	Form Select 02	C	9	
4	Form Select 03	D	8	
5	Form Select 04	E	6	
6	Form Select 05	F	5	
7	Form Select 06	H	4	
8	Form Select 07	J	3	
9	Space	K	20	
10	Motor Turn ON	L	15	
11	Print	M	21	
12	Load Buffer	N	16	
13	Start Form	R	18	
14	Form Feed	S	19	
25	Clear	P	13	
			B230	**B232**
23	115 Vac Hot/24 Vdc Hi (+)	EE/BB	2	1
24	115 Vac Neut/24 Vdc Rtn (−)	HH/CC	1	2
18	Busy	u	Connect to B231 or B233 Input Module. See Figures 96-98.	
19	Form Busy	v		
15	BCD/ASCII	T	Unconnected or connect to neutral/return for BCD; connect to Hi for ASCII.	
16	Orientation	U	Unconnected or connect to neutral/return for normal printing; connect to Hi for inverted printing.	
17	Paper Out	t	Remote indication of paper out.	
20	Motor ON	w	Remote indication of motor ON.	
21	DC Power ON	x	Remote indication of dc power ON.	
22	Bell	y	Drives external bell when Printer receives bell character.	

When the character is accepted by the printer, the Busy is turned ON. If additional characters are required, the printer cycles the Busy signal OFF-ON-OFF as many times as necessary, transferring one character at a time to the printer until the format is completed. Characters are transmitted at the rate of one character every two scans of the Controller.

3.6.5 D285 Connections and Programming

The D285 CRT Display Unit connects via a W280 cable and requires one output module and two input circuits. The connections are very similar to the P500 Printer and are shown in Table 18A. There are four models of D285 Display Unit offering a variety of power sources (50 or 60 Hz) and I/O voltage levels (115V or 24 Vdc). As an option, this display unit can be equipped with an I/O device section which includes: one 4-digit thumbwheels, one 4-digit LED display, three 24 Vdc lamps, two pushbuttons, one snap switch, and one key lock switch. These devices are wired into the controller's I/O section via two W280 cables (see Table 18A for connections), separately from the W280 cable that drives the CRT.

Within the D285 display unit, messages can be stored in PROM loaded by the factory. These messages are similar to the P500 PROM messages, except they are 64 characters wide. Up to 32K ASCII characters can be stored within the D285 memory depending upon model number ordered (minimum 8K memory providing 7K ASCII characters); addressing is provided for 496 unique messages (numbered 1 to 496). Messages can be of any length, limited only by amount of memory available; messages do not have to be one "line" long nor do they have to be even increments of 64 ASCII characters. Special characters are available to control message format and are used as follows:

Eb End of Block — indicates end of message to be displayed

Ff Form Feed — clears bottom of screen (ignored when used with 40PL DX code)

Lf Line Feed — displays a blank line for each line feed encountered

Vd Variable Data — blanks to be filled in by BCD data from the controller; one digit per blank

Bl Blink Text — begins flashing all characters until Sb or Eb is encountered; replaced by a space when displayed

Sb Stop Blink — stops flashing message characters; no effect if not already flashing. Replaced by a space when displayed.

Ts Top of Screen — place message at top of screen and begin schrolling down. Normal entry is at bottom schrolling up.

Tb Blink Top — place message at top of screen and begin flashing.

The 12" CRT screen provides excellent clearity to view messages up to ten feet away. Up to 16 lines, each of 64 characters, can be displayed at one time. Messages can be entered at the bottom and schrolled up or at the top and schrolled down. As an option, an ASCII port can be added to provide the messages to another device (such as ASCII compatible line printer, remote CRT, or magnetic tape recorder) as they are placed on the CRT screen. The controller can select messages for CRT screen only or for both CRT and optional device. When this option is selected, user must specify baud rate and ASCII type (RS-232 or 20 ma loop) at time of order.

Table 18A. D285 Connections Via W280 Cables

Connector Pin	Wire Color	Functions: D285 Control Cable	I/O Option Cables: Upper	I/O Option Cables: Lower
A	Wht/Blk/Grn	Load Buffer	TW BCD 1	Display BCD 1
B	Wht/Blk/Yel	Start Form	TW BCD 2	Display BCD 2
C	Wht/Blk/Orn	Clear	TW BCD 4	Display BCD 4
D	Wht/Blk/Red	SPARE	TW BCD 8	Display BCD 8
E	Wht/Blk/Brn	Space	TW BCD 10	Display BCD 10
F	Wht/Blk/Blk	Print	TW BCD 20	Display BCD 20
G	Wht/Gray	Form Feed	TW BCD 40	Display BCD 40
H	Wht/Violet	Hard Copy	TW BCD 80	Display BCD 80
J	Wht/Blue	Form Select 07	TW BCD 100	Display BCD 100
K	Wht/Green	Form Select 06	TW BCD 200	Display BCD 200
L	Wht/Yellow	Form Select 05	TW BCD 400	Display BCD 400
M	Wht/Orange	Form Select 04	TW BCD 800	Display BCD 800
N	Wht/Red	Form Select 02	TW BCD 1000	Display BCD 1000
P	Brown	Form Select 01	TW BCD 2000	Display BCD 2000
R	White	Form Select 00	TW BCD 4000	Display BCD 4000
S	Blue	Form Select 03	TW BCD 8000	Display BCD 8000
–	Wht/Brown	SPARE	SPARE	SPARE
T	Wht/Black	Form Busy	NO S4 Key Switch PB	Lamp #2
U	Gray	Busy	NO S1 Pushbutton	Lamp #3
V	Violet	Positive/Hi Voltage	NO S2 Pushbutton	Lamp #1
W	Green	Negative/Low Voltage	NC S3 Snap Switch	+24 Vdc Power
X	Yellow	SPARE	NO S3 Snap Switch	-24 Vdc Power
Y	Orange	SPARE	SPARE	Lamp Common
Z	Red	SPARE	SPARE	SPARE

The D285 Display Unit is controlled exactly as the P500 Printer is controlled via DX codes 40PL, 41XX, and 4200. Any controller capable of communicating to a P500 Printer is also capable of controlling the D285; no changes to the controller's hardware or executive program is required. There are only two exceptions unique to the D285 operation. When using DX codes 40PL, all Form Feed (Ff) executions will be ignored by the D285. This prevents the fixed format code from entering numerical data and then clearing the CRT screen, in lieu of advancing the P500's paper. As a maximum, only 21 of the 64 available characters in a D285 CRT line will be utilized by the fixed format DX code.

The second change is message addressing. D285 messages numbers 1-99 are used exactly as the P500 message are via either DX codes 41XX (XX = message number 01 to 99) or DX code 4200. To address messages above 99 (i.e. 100 — 496, since the D285 has 496 messages versus the P500's 100 message capacity), message zero is used as a converter. Within the D285, message zero can not be altered by the user and always has the form: Vd Vd Vd Vd. The content of these four blanks must be filled in by the controller with the message address (0001 — 0496) to be displayed.

For example, DX code 4100 will cause message zero to be addressed. The content of whatever register is entered into the B element of this logic line (all four BCD digits) will now be used as the message to be displayed. If this content is greater than 496 or not assigned within the PROM memory, the D285 Display Unit will display the message "ILLEGAL MESSAGE ADDRESSED". Another method of addressing messages above 99, is to use DX code 4200. The content of the B element register, two least significant digits only (units and tens) are made zero(XX00). The next register in sequence is assumed to contain the four digit address of the message to be displayed.

Other than the fixed format form feed and addressing messages above 99, all other features of DX Printing apply. Messages can be queued up in as many logic lines as necessary. They are serviced basically in logic line numerical order. Variable Data (Vd) will be filled in by BCD digits (0 — 9) from the register specified in the B element of the DX logic lines. All commands to the D285 require an active and positive response from the Display Unit, thus a two-way handshaking is employed. Busy, Form Busy, and Abort inputs must be programmed as discussed in paragraph 3.6.6 to insure proper handshaking capabilities.

If the ASCII port option is selected, to inhibit providing ASCII information to this port, pin H of the W280 cable is connected to a high voltage source. This can be controlled either by the appropriate type output module, or an external switch, or the output module driving the D285. The output module that provides commands and data to the D285 has one spare circuit that can be used via a DX Matrix (bit 10) set/clear operation if this capability is available. Otherwise, any single output circuit of the proper voltage can be used.

See Appendix A for further details on the D285 Display Unit.

3.6.6 Programming Busy, Form Busy, and Abort Inputs

The Busy and Form Busy signals from the printer can be connected to any convenient inputs of the proper voltage, either discrete or register inputs. To inform the Controller of which inputs were selected, special programming must be provided at the end of the ladder diagram. Counting back from the last line of the program (Watchdog Timer line), the WDT-4 coil must reflect

the printer's Busy signal, the WDT-3 the Form Busy signal, and the WDT-2 an optional Abort signal generated by the user. Table 19 summarizes the line assignments for these functions with executives of various length.

The Abort signal, when energized, will cause the current message being printed to be aborted and the next message in the queue utilized. Since the Abort signal only works on the OFF-ON transition of the abort coil, if more than one message is being aborted, this coil will be cycled OFF-ON-OFF-ON, (WDT references) as many times as there are messages to be aborted. If the Abort is not used, the WDT-2 line should contain null data to prevent its coil from being energized.

Table 19. Line Assignments for Printer Control Functions

Program Length	Busy	Form Busy	Abort
432	428	429	430
496	492	493	494
512	508	509	510
608	604	605	606

If the Busy, Form Busy, and Abort signals are connected to discrete inputs, Figure 96 illustrates one method of programming the WDT-4 through WDT-2 lines. Whenever the printer is busy, the input signal to the Controller

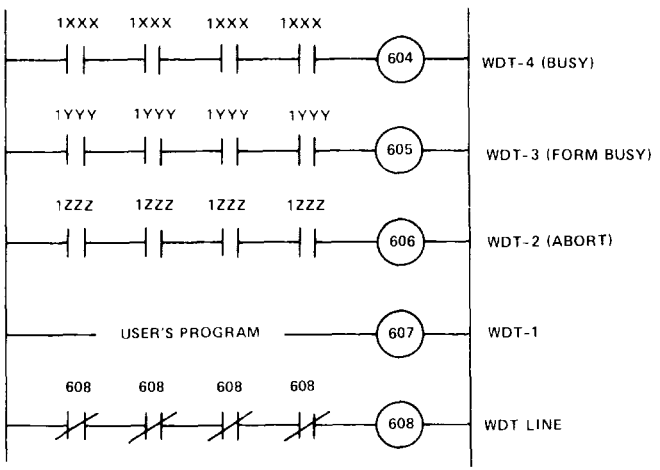


Figure 96. Typical Printer Control Lines using Discrete Inputs

(wire 18 of Table 18) is turned ON, input 1XXX to which this wire is connected becomes energized, and coil 604 is energized. Coil 604 will thus copy the Busy status of the printer via input 1XXX, which can be any discrete input to the Controller. A similar analysis can be performed for the Form Busy (1YYY) and Abort (1ZZZ) signals.

If the Busy, Form Busy, and Abort signals are connected to register inputs, a number of methods can be used to program the WDT-4 through WDT-2 lines. If matrix capabilities are available, the method in Figure 97 can be used. The three sense lines energize their coils only if the bits in the input registers (3014 and 3008) representing Busy, Form Busy, and Abort signals, are turned ON. Registers 4XXX, 4YYY, and 4ZZZ contain the bit numbers to which the Busy, Form Busy, and Abort signals are connected.

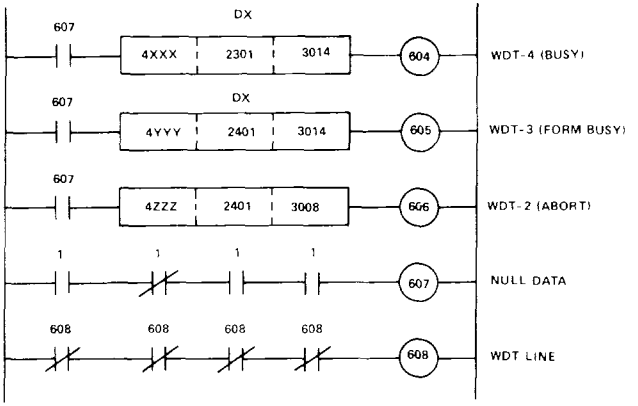


Figure 97. Typical Printer Control Lines using Matrix Register Inputs

If matrix capabilities are not available, subtraction lines similar to those in Figure 98 can be used. As an example, assume the Busy and Form Busy inputs are connected to the one and two lines of the thousands digit of BCD register 3014. The Abort signal is assumed to be a discrete input connected to 1ZZZ. Storage registers 4HHH, 4TTT, 4BBB, and 4MMM contain the values 1000, 2000, 3000, and 4000, respectively; lines 600, 601, 602, and 603 will be ON if, and only if, the contents of register 3014 exceeds or equals the values 1000, 2000, 3000, and 4000 respectively. Thus, the Busy line (604) will be ON if the thousands digit of register 3014 is a one (1) or a three (3) and the Form Busy line (605) will be ON if this digit is a two (2) or three (3).

If more than one printer is connected to the Controller, the Busy and Form Busy signals from each printer should be ORed into WDT-4 and WDT-3 lines. Thus, if any one printer is busy, the WDT-4 coil is energized, and if any printer has its Form Busy ON, the WDT-3 coil is energized. Figure 99 illustrates one method of programming these lines assuming three printers are connected and all inputs are wired to discrete input modules.

3.7 Improved Data Transfer Capabilities (384A)

The following DX functions have been developed and are available as standard features on the 384A and 384B controllers. All of the following functions are provided with these controllers in addition to all the standard DX functions previously discussed. A few 184 executives have been configured with one of these functions; see table 12 for specific details or which functions are available with 184 controllers.

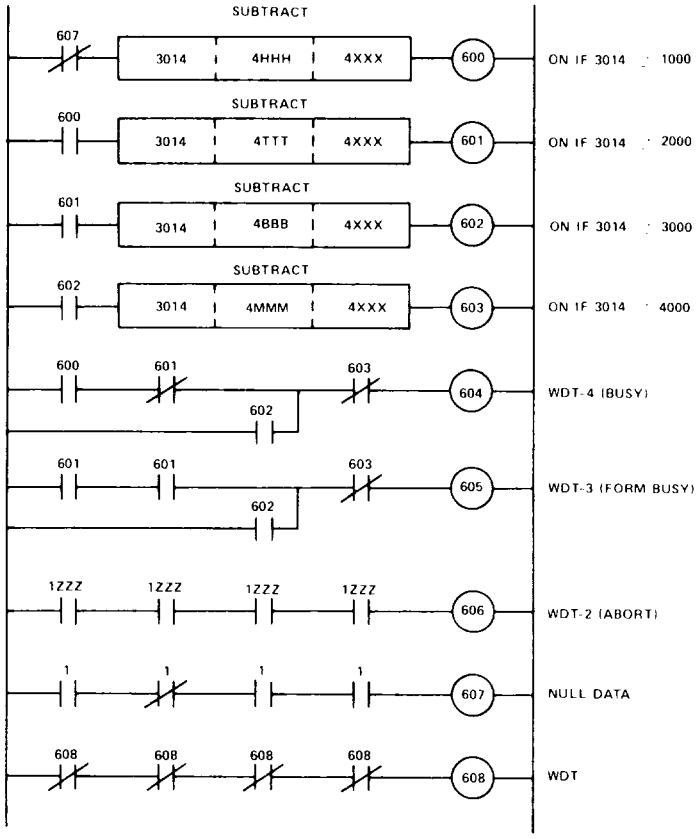


Figure 98. Typical Printer Control Lines using Register Inputs

3.7.1 Privileged Registers (5XXX references)

In many applications, the 384 controller is so efficient in its use of logic lines, that more holding registers are also desired. This is especially true of large monitoring routines and multiple recipe storage. To satisfy these requirements, the concept of "privileged" registers was developed. When memory protect is ON, these registers cannot be altered by the logic, only copied. Thus standards can be entered and used in the logic as required; however, once memory protect is ON, they cannot be altered by the logic. They can be examined at any time using the programming panel; they can never be altered directly from the programming panel regardless of the condition of memory protect. At any time, a computer via the computer interface, can alter these registers.

The privileged registers, (5XXX) are similar to holding registers (4XXX) in that they are retentive on power failure and utilize one word of memory. Thus they can contain four BCD digits (maximum value 9999) or sixteen bits of binary information. However, they can NOT be referenced in the logic as can the holding registers (4XXX). To utilize data contained in the privileged registers, they first must be moved into holding registers. Two Move functions are available for handling privileged registers: 18XX (move into 5XXX) and 19XX (move out of 5XXX).

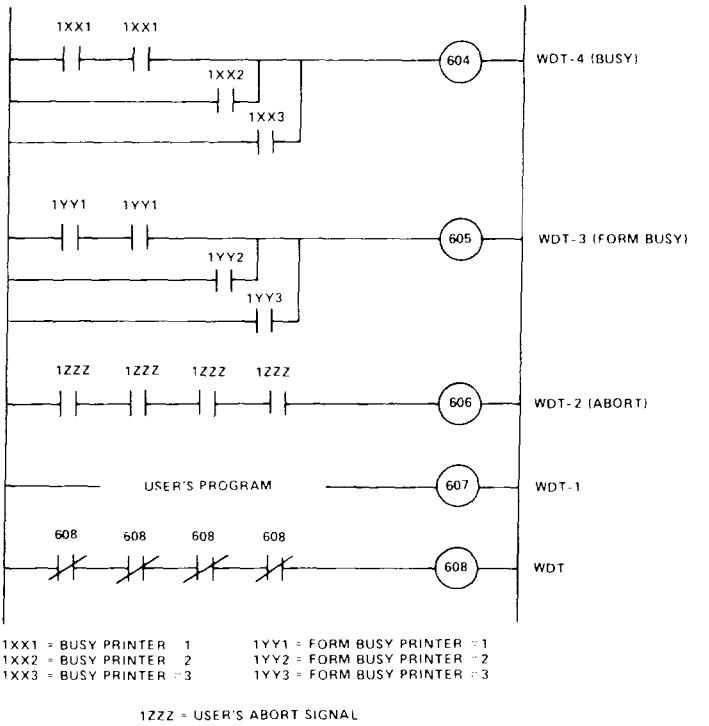


Figure 99. Typical Printer Control Lines with Multiple Printers

18XX — *4XXX/30XX to 5XXX Move (Block)*

This code causes the content of a table of input (30XX) or holding (4XXX) registers to be copied into a table of privileged registers (5XXX). This is a block move such that every scan the A element contact is closed and memory protect is OFF, the entire table will be moved. The table length is indicated by the XX of the function code. The coil always reflects the state of memory protect regardless of the condition of the A element; thus the coil will be ON when memory protect is ON, and OFF when memory protect is OFF. This move code (18XX) will not function as long as memory protect is ON. If memory protect is engaged during the block move, the move will be completed and then additional updates inhibited.

Since this is a block move, no pointer is required and all registers in the table will be moved every scan the A element remains closed. Both tables must be of the same length. Since the DX code has only two digits for table length, the maximum size of these tables is 99; the minimum table size is one. As an example, refer to figure 100.

When input 1001 is energized, 35 holding registers (4237-4271) are moved into privileged registers (5106-5140) in one scan. As long as input 1001 remains energized, registers 4237-4271 are copied into registers 5106-5140 thus updating these privileged registers every scan. No pointer is involved since all registers are updated effectively at the same time. The coil of line 731 will be energized whenever memory protect is ON regardless of the condition of the A element. If memory protect is ON, line 731 will *not* affect the contents of registers 5106-5140 even if input 1001 is energized.

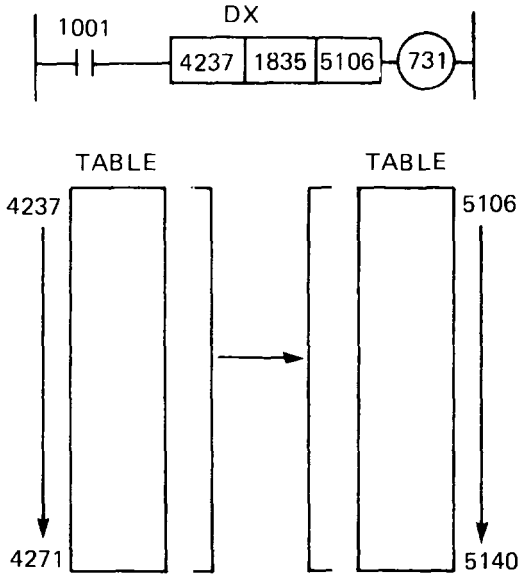


Figure 100. Sample 4XXX to 5XXX Move

19XX — *5XXX to 4XXX Move (Block)*

This code causes the contents of a table of privileged registers (5XXX) to be copied into a table of holding registers (4XXX). This is a block move such that every scan the A elements contact is closed, the entire table will be moved. This move can be accomplished regardless of the condition of memory protect. The coil will be energized when the move has been completed; since the entire table is moved every scan, the coil will be energized whenever the A contact is closed.

This is a block move and thus no pointer is required; all registers in the table will be moved every scan the A element remains closed. Both tables must be of the same length. Since the DX code has only two digits for table length, the maximum size of these tables in 99; the minimum table size is one. As an example, refer to figure 101.

When line 392 energizes its coil, 12 privileged registers (5063-5074) are moved into holding registers (4419-4430) in one scan and the coil will be energized. As long as coil 392 is energized, all 12 privileged registers will be moved into the holding registers and coil 688 will remain ON. No pointer is involved in this move since all registers are moved every scan. This move can take place only into holding registers (4XXX), not input registers (30XX), since it does alter their content.

3.7.2 Matrix Sequencer Move

The additional 30XX references discussed in section 3.5.2 form the bridge between discrete references such as line coils (0XXX) and inputs (1XXX), and register references. The DX function code 29XX forms the bridge in the opposite direction, from registers to discrete references. Inputs (1XXX) have been selected as the discrete references that can be controlled from register contents.

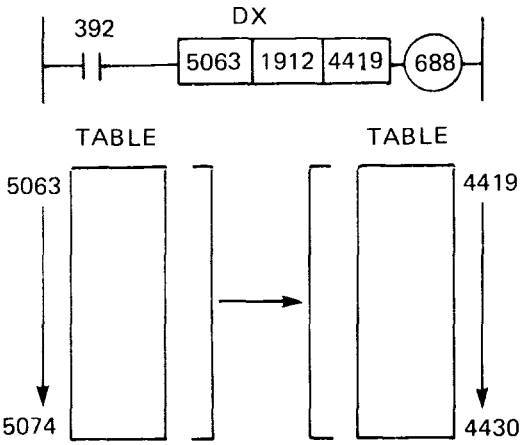


Figure 101. Sample 5XXX to 4XXX Move

These inputs are assigned reference numbers beyond those used in the Input/Output section. Exactly how many sequencer inputs are available is dependent upon the executive or TEF program selected (see Appendix F). However, as an example, assume a TEF was configured for 256 "real" inputs and 128 sequencer steps. The references for discrete inputs would be the conventional 1001-1256 and for the sequencer steps 1257-1384. The sequencer inputs can be dedicated to one sequencer or to a series of independent sequencers depending upon the application and the user's program; however, the total number of sequencer inputs (or steps), can not exceed the quantity established by the executive for this purpose.

NOTE

It is convenient, but not necessary, to assign sequencer inputs to individual sequencers in groups of sixteen.

29XX - Matrix Sequencer Move

This code causes the content of a matrix (B element) to be moved into discrete references, such that each register controls 16 "sequencer" inputs. Matrices can be of holding registers (4XXX) or input registers (30XX). The discrete references are updated every scan the A element is closed.

NOTE

The sequencer will retain this state when the A element is opened and through a power failure. They can only be altered by a 29XX DX logic line.

The XX in the DX code (C element) identifies the size of the matrix in registers and the quantity of sequencer inputs (when multiplied by 16), controlled by that logic line. The minimum size of a matrix is one register, and the maximum is the number of sequencer inputs provided by the TEF divided by 16.

The D element is the first sequencer input to be controlled by this logic line; all other sequencer inputs follow this reference in ascending numerical order. The reference entered in the D element must be a sequencer input such that, when divided by 16, will result in a remainder of exactly one. As an example, refer to figure 102.

NOTE

If an attempt to enter a reference that is not of the proper value

is made, the automatic error checking in the 384A or 384B will reduce that value to the next lowest reference that is legal.

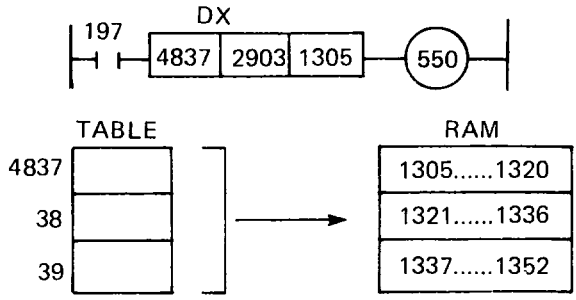


Figure 102. Example Matrix Sequencer Move

When logic line 197 energizes its coil, the content of registers 4837-4839 is moved into the RAM where it establishes the state (ON or OFF) for sequencer inputs 1305-1352. Every scan the A element is closed, these inputs will be updated with the content of registers 4837-4839. If an error is made in the programming of line 550 that the controller cannot correct for, and the A element is closed, coil 550 will be energized. Since the matrix is defined by the DX code as three registers long, it controls the status of 48 sequencer inputs. When the A element is opened, these inputs retain their state even through power failure, and can be altered only by a DX logic line utilizing a 29XX function code.

For example, if the content of the first eight bits of register 4837 was 10011101, then sequencer inputs 1305, 1308, 1309, 1310, and 1312 would be energized since they are opposite one (ON) bits and inputs 1306, 1307, and 1311 would be de-energized since they are opposite zero (OFF) bits. These sequencer inputs can be used anywhere in the user's logic to produce a sequencer or tenor drum effect. How many steps there are in each sequencer depends upon how many matrices are assigned to controlling the sequencer inputs; how many outputs are assigned to each sequencer, depends upon how many sequencer inputs are utilized (i.e., size of DX code XX value).

3.7.3 Improved PID

The basic PID function (DX code 3400) has been altered in two areas to provide a more user-related operation. The first change was to incorporate a PID function with constants in minutes instead of seconds. The second change was to make the derivative term (K_3) a function of rate of change of input and not error. Other than these improvements, the PID function operates the same as discussed for function code 3400 in section 3.5.3. These new capabilities are in addition to the 3400 code; all four PID functions (3400, 3401, 3500, and 3501) are available with the improved PID and are in the fast ROM area of the 384.

35XX — PID (Minutes)

This function operates identical to function code 34XX discussed in section 3.5.3, except that the constants are entered as minutes. The following is the definition of constants used in the seven register input table to the PID function:

first register (e.g. 4810) = Input } Same Units
second register (e.g. 4811) = Set Point }
third register (e.g. 4812) = K_1 (Gain)
fourth register (e.g. 4813) = K_2 (repeats per min. in the form XX.XX, or 00.00-99.99)
fifth register (e.g. 4814) = K_3 (minutes in the form XXX.X, or 000.0-999.9).
sixth register (e.g. 4815) = High Limit
seventh register (e.g. 4816) = Low Limit

3401 and 3501 — *PID with Modified K_3 Term*

This function operates identical to function 3400, except that the derivative term responds to the rate of change of input, not error. The PID equation now becomes:

$$P = K_1 e + K_2 \int_0^t edt + K_3 \frac{di}{dt} + K_4$$

Utilizing this function allows changes to be made to the set point, without affecting the K_3 term of the PID function. The output will still respond to the K_1 (gain) and K_2 (integral) terms; however, the oscillatory effects of K_3 will be eliminated.

3.7.4 Sort

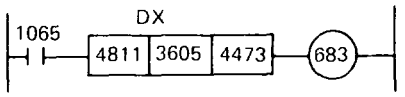
These two DX functions allow a table of registers to be sorted by their numerical contents. The entire table is sorted in one scan upon closure of the A element. The sort is performed only once upon closure of the A element; if additional sorts are required, the A element must be opened and then closed again. The first register of the table to be sorted is entered in the D element; the DX code last two digits indicate the table length. Maximum table length is 99 registers, minimum is 02.

Associated with the D element table is another table of the same length referred to in the B element. At the beginning of the sort, each register in the D element table is paired with a register in the B element table. The first registers in each table are paired together, the second registers, the third registers, etc. The sort is then performed on the D element table by its numerical value. The B element table is also sorted but only to maintain the paired relationship.

For example, if the first register in the D element table becomes the third register after the sort because of its numerical magnitude, the first register in the B element table will become the third register in its table regardless of its magnitude. The B element must be a holding register (4XXX), not an input register (30XX) since its value can be altered. The tables must be equal length and between 02 and 99 registers long. The coil will come ON if no sorting is required when the A element is closed; it remains ON as long as the A element is closed.

36XX — SORT (Ascending order)

This code causes the content of a table to be sorted by their numerical values, resulting in the smallest magnitude on the top, and the largest on the bottom. The sort is performed entirely in one scan upon closure of the A element; the coil will be energized if no sort is required. For example, refer to figure 103.



PRIOR TO SORT

4811	0101	4473	0011
12	0200	74	0007
13	0400	75	0015
14	0700	76	0004
15	2000	77	0003

AFTER SORT

4811	2000	4473	0003
12	0700	74	0004
13	0200	75	0007
14	0101	76	0011
15	0400	77	0015

Figure 103. Example Sort

When the A element is closed, the content of D element table (registers 4473-4477) are rearranged to placed them in ascending order. The lowest content (value 0003) is placed in the first register, the next to lowest in the second, etc. until the highest magnitude (value 0015) is placed in the last register. A value of zero is considered the lowest magnitude. The coil is energized if the D element table was already in ascending order when the A element was closed.

The B element Table is also rearranged; however, its result depends upon the magnitudes of the D element table. Prior to the sort, the value 0101 (first register in the B element table) is paired with the first register in the D element table (value 0011). After the sort, the value 0011 results in the fourth element of the D table; thus the value 0101 is also placed in the fourth element of the B table to maintain its pairing. All values in the B element table are similarly arranged regardless of their magnitudes.

37XX – SORT (Descending Order)

This code operates exactly the same as function code 36XX except that the D element table is arranged in *descending* order. The highest content is placed in the first register. For example, refer to figure 103 and assume the DX code was 3705. The results would be as follows when the A element is closed:

Register	Content	Register	Content
4811	0400	4473	0015
12	0101	74	0011
13	0200	75	0007
14	0700	76	0004
15	2000	77	0003

3.7.5 Guarded Lines

All 384A and 384B controllers have the capabilities to be configured with logic lines that have additional protection from change beyond Memory Protect. These lines can be located anywhere in the user's logic lines as one contiguous set. Lines are guarded in groups of sixteen, with the first line number, when divided by sixteen, resulting in a remainder of exactly one

(e.g. lines 33, 81, 129, 289, 401, etc.) Any number of lines can be guarded, but they must be in consecutive numerical order. Any 384 controller can also be provided with guarded lines, by requesting a TEF 04 executive that includes guarded lines.

Guarded lines can be viewed at any time from various programming devices (P112 Programming Panel, P140/P145 CRT's etc.) to determine their programmed content and power flow. However, if Memory Protect is turned OFF and a change made to any content of these guarded lines (i.e. disable coil, alter reference, change contact type, register address, or fixed preset, etc.) *All* guarded lines will have their coils forced OFF. Removing the change will NOT reenable the coils; the coils can be restored only by reloading the entire program. This restore can be accomplished from either the MODICON Service Center or a L206 Tape Loader. If the guarded line coils are used as normally open contacts in the normal logic as permissives, when changes are attempted to the guarded logic the permissives will be lost and the process/machine operation will stop. If Memory Protect is ON, changes to guarded lines will be inhibited and no loss of guarded coils is possible. After loss of guarded coils, turning Memory Protect ON will cause the RUN lights to go OFF; turning Memory Protect OFF will restore the RUN light.

When a system is received from MODICON, or a reload of the TEF with null data is made, no lines are guarded. The logic to be guarded is entered in the normal manner with any programming device. Once this program is installed and checked-out, the MODICON Service Center is contacted and asked to guard specific line numbers. A special code is added by the Service Center via the T152 Interface to protect the desired lines. The Service Center can also remove the guarding function if additional changes to these lines is desired; however, special identification procedures will be required to validate such a request.

In addition to logic lines, inputs can be also guarded: Once guarded, the disabled state of these inputs cannot be altered. Inputs that are enabled cannot be disabled, nor can disabled inputs be enabled. If a change to a guarded input's disabled state is attempted, all guarded lines will have their coils turned OFF. The system responds exactly the same to changing guarded inputs as it does for guarded lines.

3.7.6 Input/Output Communication Status

Four input registers after the additional 30XX references (see section 3.5.2) are allocated to recording the results of the I/O communications. If the controller is unable to communicate to an I/O module (see section 4.1.2), a one is placed in these registers; if it can communicate without six retransmissions, a zero is placed in these registers. A zero is also placed in the registers if an I/O module is not installed. The first register contains the status of channel I's I/O modules, the second channel II, etc. Within these registers, the first eight bits record the input module statuses (slots 1-8) and the last eight bits the output modules (slots 1-8).

To monitor the I/O module status, a matrix of known zero's is constructed in four holding registers: a DX compare line (2204) compares these input registers with this known standard. If an I/O module malfunctions, the compare line coil is energized and the pointer can be decoded to identify exactly which I/O module. When more than one I/O module malfunctions, the pointer will have more than one value and each can be decoded. If this standard is built in registers 4771-4774, figure 104 illustrates such a compare line for monitoring I/O modules; additional 30XX references are assumed to stop at 3095.

In addition to these four registers, a fifth register is available to indicate the status of the mainframe. Using the assumed references shown in figure 104, this fifth register would be 3100; these exact references always depend upon where the standard 30XX references end and thus the number of logic lines and discrete inputs. Five bits are used in the mainframe status register as follows:

Bit No.	ON (one value) When
1	Memory Protect is ON
2	Last reset caused by memory protect violation
3	Error detected in Logic Solver hardware
4	Programming Panel connected
5	First scan following power failure

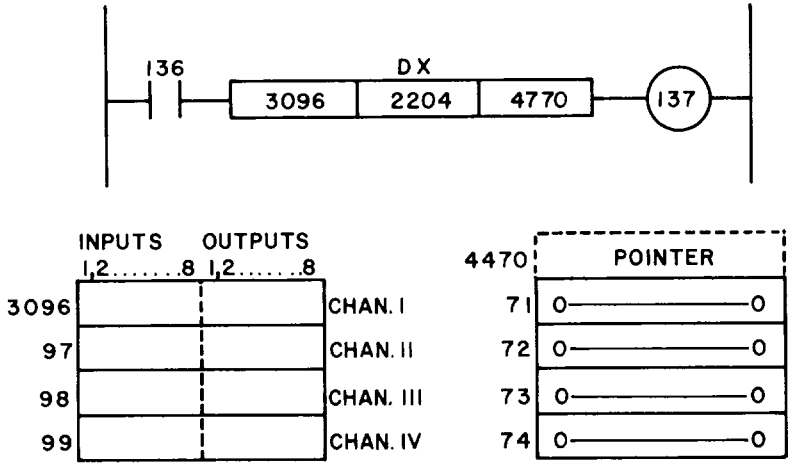


Figure 104. Example I/O Monitoring

3.8 ASCII I/O COMMUNICATIONS (384B)

The 384B Controller has all DX capabilities previously described in paragraphs 3.5, 3.6, and 3.7 for the 184, 384, and 384A Programmable Controllers. In addition, the 384B has the communications capabilities to be utilized with the 1084 Programmable Controller (including 1084 ASCII) as well as its own ASCII driving capability. Refer to the 1084 manual for complete discussion of the 1084 communications. The 384's stand-alone ASCII capability is obtained by adding four DX functions (codes 43XX, 44XX, 45LF and 46XX); these codes are discussed in the following subparagraphs. The 384B's require a TEF10 level executive to utilize the ASCII DX codes; however, they can operate on TEF08 executives if ASCII I/O is not required.

3.8.1 Introduction to ASCII Communications

ASCII I/O is obtained by using the B680 or B684 I/O module (see appendix B). When installed in I/O structure, this B680/B684 requires only one B240 or B241 slot locations. It utilizes two index pin locations, one for input capability and one for output capability, the same location as set by the user. Thus as a maximum, only eight ASCII modules (and no other modules of any type) can be installed in a single I/O channel. If less than eight ASCII modules are installed in a channel, they can be mixed with discrete and/or register I/O. Internally, each ASCII module requires four input registers

(30XX references) and four output registers (40XX references); no other module or I/O slot location should be coded to utilize data in these I/O registers. Thus, if all available register I/O is dedicated to ASCII I/O, up to eight ASCII I/O modules can be accommodated per 384B controller. Traffic cop modifications are required for each I/O slot number into which ASCII I/O modules are to be installed.

Each ASCII module is serviced separately during a 384B scan and thus they all can be active each scan. In addition, separate storage is provided for both input and output transfers, allowing simultaneous transfers (duplex operation) for each ASCII module. These ASCII communications are controlled by DX PRINT logic lines with the DX codes 43XX and 45LF (outputs) or 44XX and 46XX (inputs). As many DX logic lines can be addressed to a single ASCII module as required; however only one input and one output logic line will be actively communicating to an ASCII module at one time.

Similar to the P500 PRINT, when the A element is transistioned from open (not passing power) to closed (passing power), the print line becomes "active" and energizes its coil. If the ASCII module is available, the logic line begins commuication; if the module is already busy with another line, this line's communication is delayed until the module is available. The coil of an ASCII Print line will be energized whenever the A element is closed and de-energized when that ASCII Print operation is complete, regardless of the state of the A element.

NOTE

The A element need be closed for only one scan, and will not restart communications if it remains closed after completion of the line's function.

The first ASCII DX line activated and referenced to an ASCII module will take control of the module. Subsequent lines that are referenced to the same module, will be queued up and serviced by their line number in numerical order following the line that currently controls the ASCII module.

NOTE

Servicing these queued lines begins at the line following the currently controlling line *NOT* Line number one.

Each ASCII module in the I/O structure is provided with four consecutive output registers (40XX) for storing ASCII characters prior to their delivery to the ASCII device, and a similar number of input registers (30XX) for receipt of characters from the ASCII device. These two buffers have completely independent control and storage and can be accessed by any properly programmed logic line in the 384B controller. The only similarity is that the last two digits of the register references will be the same (i.e., 3001-3004 and 4001-4004).

NOTE

Since Traffic Cop modifications are required for all ASCII modules, it is convenient, but not necessary, to leave the normal 30XX and 40XX references to BCD/Binary numerical I/O (i.e., 3001-3016 and 4001-4016), and establish other references for the ASCII modules. Suggested references are:

MODULE	INPUTS	OUTPUTS
1	3017-3020	4017-4020
2	3021-3024	4021-4024
3	3025-3028	4025-4028
4	3029-3032	4029-4032

If more than four ASCII modules are utilized, some (or all) of the normal register I/O will be required.

The input or output buffer (four registers) are utilized in a similar operation as follows:

POINTER		Register 1
Status	Char 1	Register 2
Char 2	Char 3	Register 3
Char 4	Char 5	Register 4

The first register contains the line number which currently controls that ASCII module buffer (either input or output, but not both). A pointer of zero indicates no line has control over the buffer, it is available for use. The remaining three registers are divided in halves, and contain status relative to communication to the ASCII module or characters in the process of being transferred.

NOTE

Upon power failure, all DX ASCII PRINT logic lines have their coils de-energized, and Buffer Pointers set to zero. To abort an ASCII output, bit 1 of register 1 should be set. As long as this bit is set to a one state, all outputs to this ASCII device, will be terminated. ASCII inputs cannot be aborted.

When a properly referenced ASCII DX line is activated, it looks at the pointer (register 1) to determine if the buffer is available. If it is, it places its line number into the pointer, and loads/obtains from the buffer up to five characters. Every scan, the DX line monitors the status area and refills/empties the buffer with up to five characters if the buffer is empty/full. When the DX line has completely transferred its quantity of ASCII characters to/from the buffer, the DX line de-energizes its coil and clears the buffer pointer (register 1) to zero. The next active line referenced to this ASCII module can now take control of the buffer.

When the scan services the I/O slot to which an ASCII module is referenced, up to five characters can be transmitted to/receive from the B680/B684 I/O module. Less than five characters will be transmitted/received if the buffer is only partially full or if the B680/B684 has space for less than five characters. For example, if the scan rate of the 384B controller is 50 msec (20 scans per second), up to 100 characters can be sent/received per second. This converts to 1100 baud at eleven bits per ASCII character, (eight data, two stop, one start bits). This transmission rate is fully duplex and applies to all ASCII devices connected to the 384B controller, since each is serviced independently.

NOTE

Baud rates are a function of scan time which can depend also upon number of ASCII devices active at one time.

3.8.2 Programming DX ASCII Functions

43XX-ASCII Output

This function code copies ASCII characters stored in holding registers and provides them to a single ASCII device. The characters are assumed to be packed two per register and can be any legal ASCII character. Referring to figure 105 (line 167) the A element is a relay contact that controls when the ASCII Output is to be activated. The A element can be referenced to any line coil, discrete input, or latch reference. The B element is a pointer which indicates how many characters (NOT registers) have been provided to the output buffer; the table of registers that contain the ASCII characters must follow this register in ascending order. The B element must be a holding register (4XXX).

NOTE

The contents of the pointer will be reset to zero when the ASCII output is completed.

The C element is the DX code (43XX) where XX is replaced with the number of registers in the B element table (excluding the pointer). Thus, the maximum number of ASCII characters accessed by a 43XX DX line will be twice the XX value. The D element is the output register (40XX) to which the ASCII module is indexed. In all cases, four consecutive output registers starting at the D element register are assigned as this ASCII module's output buffer.

NOTE

No other output modules or I/O slot locations should be addressed to any of these output registers.

The coil indicates this line is busy providing ASCII characters to that particular B680/B684 module. The coil is energized when the A element is closed, and de-energized whenever the transfer to the output buffer is complete. The transfer is complete whenever the end of the table as defined by the DX code is reached or a special delimiter is encountered in the table.

NOTE

The standard delimiter is *LF* and can be altered by a Traffic Cop type change. The delimiter is outputted to the ASCII device.

When the A element of line 167 (see figure 105) is closed, coil 167 is energized. If the output buffer is available (register 4021 contains zero), the line's number is loaded into register 4021 and the first five ASCII characters are loaded from the table starting at register 4371 into the output buffer. If register 4021 did not contain zero, outputting ASCII characters is delayed until all ASCII Outputs prior to line 167 that are addressed to register 4021 are completed. Every scan when line 167 is solved, the output buffer status (bits 1-8 of register 4022) is examined until the buffer is empty. When empty, line 167 transmits the last five ASCII characters into the output buffer, de-energies its coil, and clears register 4021. If the A element is closed when its coil is de-energized, no further outputs will be commanded by line 167.

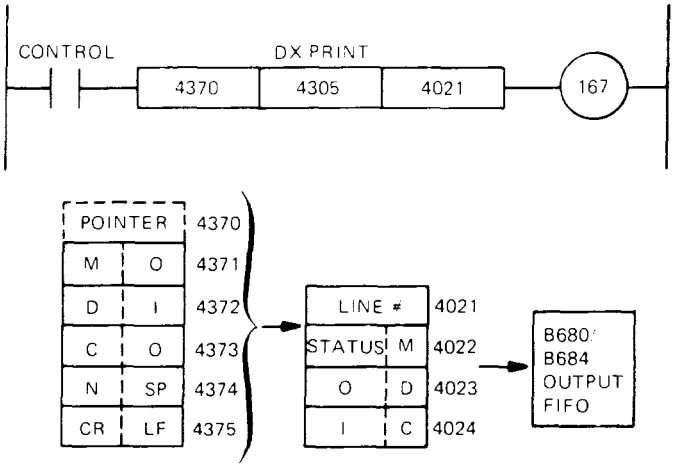


Figure 105a. ASCII Output

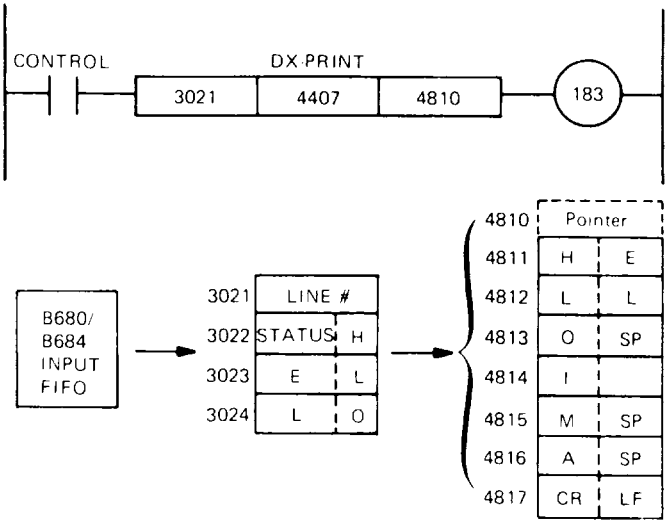


Figure 105b. ASCII Input

NOTE

If the quantity of the last set of ASCII characters loaded into the buffer is less than five (e.g., three), the status area will insure only the new characters are sent to the B680/B684 I/O module.

44XX - ASCII Input

This function code receives ASCII characters from a single ASCII device and stores them in a table of holding registers. The ASCII characters are automatically packed two per register and can be any legal ASCII character. Referring to figure 105, (line 183), the A element is a relay contact that controls when the ASCII Input is to be activated. The A element can be referenced to any line coil, discrete input, or latch reference. The B element is the input register (30XX) to which the ASCII module is indexed. In all cases, four consecutive input registers starting at the B element register are assgned as this ASCII module's input buffer.

NOTE

No other input module or I/O slot locations can be addressed to any of these input registers.

The C element is the DX code (44XX) where XX is replaced with the number of registers in the D element table (excluding the pointer). Thus the maximum number of ASCII characters loaded by a 44XX DX line will be twice the XX value. The D element is a pointer which indicates how many characters (NOT registers) have been loaded from the input buffer; the table of registers that are loaded by this line must follow this register in ascending order. The D element must be a holding register (4XXX).

NOTE

The contents of the pointer will be reset to zero when the ASCII input is completed.

The coil indicates this line is busy receivig ASCII characters from that particular B680/B684 module. The coil is energized when the A element is

closed, and de-energized whenever the transfer from the input buffer is complete. The transfer is complete whenever the end of the table as defined by the DX code is reached or a special delimiter is received from the ASCII device.

NOTE

The standard delimiter is *CR* and can be altered by a Traffic Cop type change. The delimiter is stored in the table.

When the A element of line 183 (see figure 105) is closed, coil 183 is energized. If the input buffer is available (register 3021 contains zero), register 3021 is loaded with the line's number (i.e., 183), and any ASCII characters in the buffer (up to five) are loaded into the table starting at register 4811. If register 3021 did not contain zero, receiving ASCII characters is delayed until all ASCII inputs prior to line 183 that are addressed to register 3021 are completed. Every scan when line 183 is solved the input buffer status (bits 1-8 of register 3022) is examined to determine when new input data is available. Up to five ASCII characters are received every scan; the buffer does not have to be full to be acted upon by the DX 44XX logic line. After fourteen characters are received in this particular example (DX code 4407), line 183 will de-energize its coil and clear register 3021. If the A element is closed when its coil is de-energized no further inputs will be commanded by line 183.

45LF-ASCII Numerical Output

This function code operates similar to function code 43XX, except that the data to be outputted is four BCD numerical digits per register. The content of each register of the B element table is automatically converted from binary to BCD, and each resultant BCD digit is provided to the ASCII module as an ASCII character. Digits are provided first from the high order (1000's) digit.

The format of outputting data is controlled by the L and F characters of the DX code. The L character specifies the number of lines to be outputted (1 to 9) each followed by a single carriage return and a line feed. The content of each line is controlled by the F or format character as follows:

F Code	Digits Per Column	Number of Columns	Spaces Between Columns	Registers Per Line	Characters Per Line
0	4	1	0	1	4
1	8	1	0	2	8
2	4	2	8	2	16
3	8	2	8	4	24
4	4	4	8	4	40
5	8	4	4	8	44
6	4	8	4	8	60
7	8	8	2	16	78
8	4	12	2	12	70
9	4	16	1	16	79

NOTE

Sufficient registers must be provided to support the content of the format specified.

If a line quantity of zero is specified, the ASCII output line will provide to the output buffer a single carriage return followed by the number of line feeds specified by the F character (0-9). Any spaces, carriage returns, or line

feeds required by the format of numerical ASCII output is automatically generated by DX function and does not require register storage.

NOTE

The B element register is not utilized with DX codes 450F and can be any legal register value.

The first four registers in the table associated with the BCD values to be outputted, are utilized by the ASCII output for internal statuses. These registers are not included in the table length as defined by the L and F digits of the DX code. The first register records the number of registers that have been converted to BCD values and then to ASCII characters. The second register stores various status information about the transfers and when to generate spaces or line feeds. The remaining two registers temporarily store ASCII characters after conversion and prior to their delivery to the output buffer. These registers must be allocated to the ASCII Print line, cannot be used elsewhere in the program, and must not be altered by the user.

As an example, refer to figure 106, line 362. When line 278 energizes its coil, line 362 also energizes its coil and monitors register 4025. If register 4025 contains a zero (buffer available), line 362 takes control of the buffer and begins to output the ASCII data. Since there are three lines required, each with four registers content separated by eight spaces (DX code 4534), a total of 16 (12+4 internals) registers is required in the B element table. Thus, this line provides to the ASCII device the content of registers 4235-4246 separated by appropriate spaces and line feeds. Line 266 of figure 106 will cause a single carriage return and two line feeds to be generated when line coil 283 is energized.

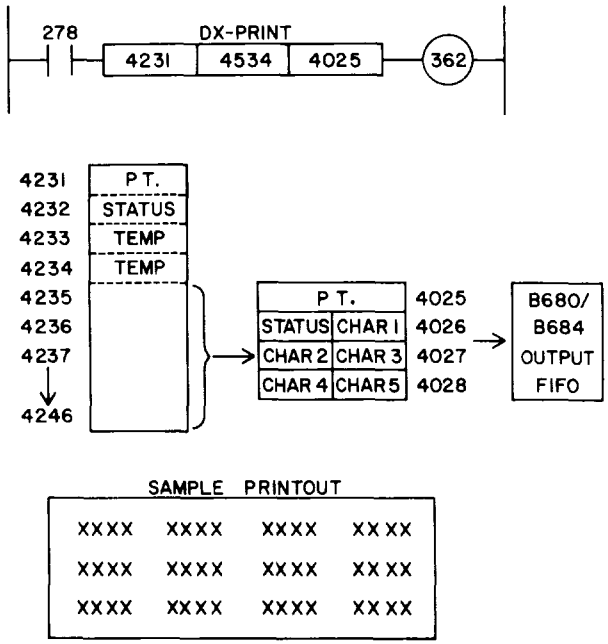


Figure 106a. ASCII Numerical Output

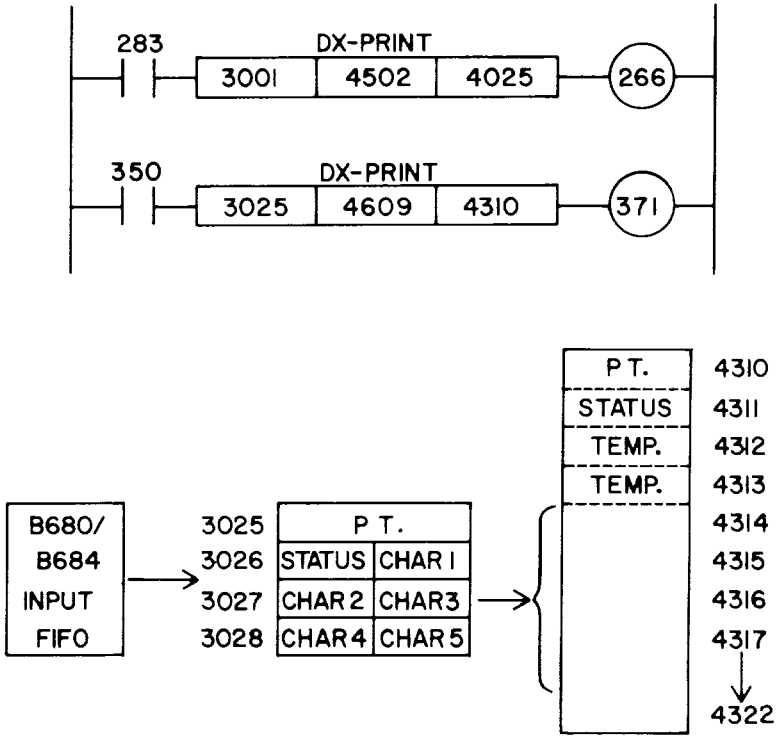


Figure 106b. ASCII Numerical Input

46XX-ASCII Numerical Input

This function code operates similar to function code 44XX except that the data received is packed up to four BCD numerical digits per register. ASCII characters other than numerical values, space, or a delimiter (carriage return) will be ignored and not effect the ASCII inputting via the 46XX code. Digits are stored in registers with each new digit causing previous digits stored in the register to be shifted to left (higher order). After four digits are received, the next register in the D element table is loaded with the next numerical character. If a space character is received after a register contains at least one BCD digit, loading that register is stopped and the next register is accessed. The loading of this table is terminated whenever the end of the table as defined by the XX of the DX code is reached, or if the delimiter is detected (CR).

As an example, refer to line 371 of figure 106. When line 350 energizes its coil, line 371 also energizes its coil and monitors register 3025. If register 3025 contains a zero (buffer available), line 371 takes control of the buffer and begins to input the ASCII data. Since there are nine registers in the D element table (plus 4 internal), up to 36 numerical digits (including zeros), can be received. The pointer in register 4310 contains the number of

registers in the table that have been loaded. The content of all registers loaded by this DX line (e.g., 4314-4322) is automatically converted to binary after all digits for that register are received.

Delimiters

There are two separate delimiters in each 384B controller; one operating on ASCII inputs and the other on ASCII outputs. The same delimiters are used for both 1084 ASCII communications as well as DX (stand-alone) ASCII. Each delimiter can be altered from the Service Center similar to changing the Traffic Cop. Unless otherwise requested, delimiters will be a Carriage Return (CR) for inputting and Line Feed (LF) for outputting. Delimiter for numerical ASCII (45LF and 46XX) can NOT be changed.