(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.

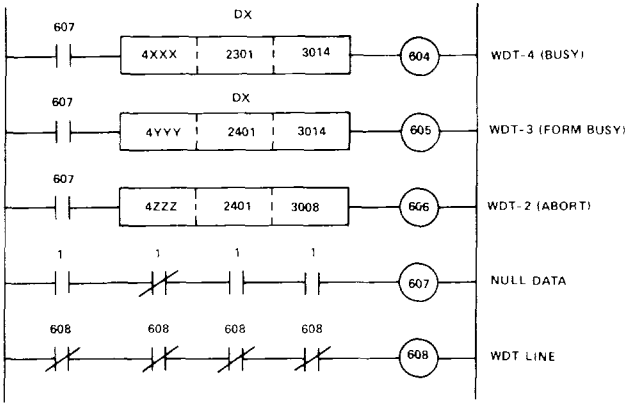


Figure 97. Typical Printer Control Lines using Matrix Register Inputs

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.

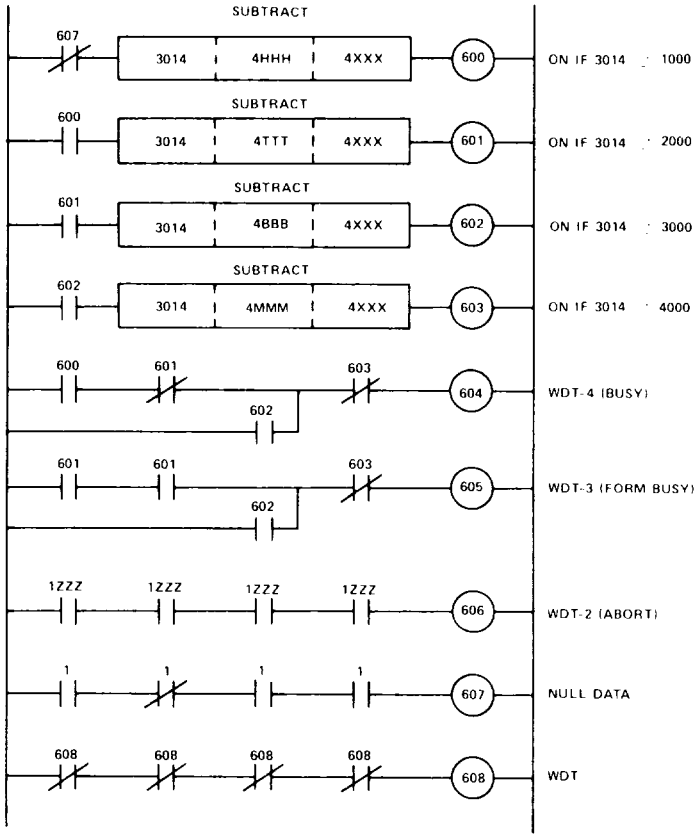


Figure 98. Typical Printer Control Lines using Register Inputs

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).

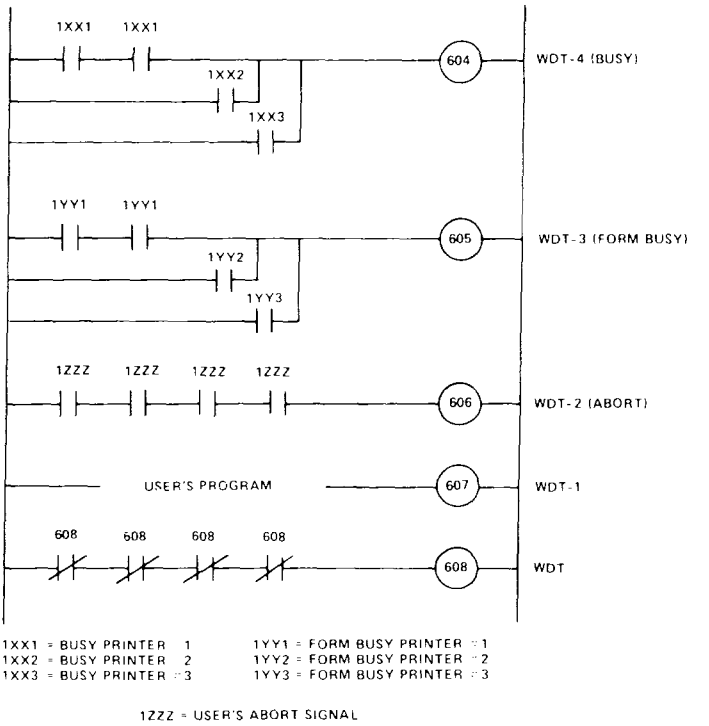


Figure 99. Typical Printer Control Lines with Multiple Printers

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.

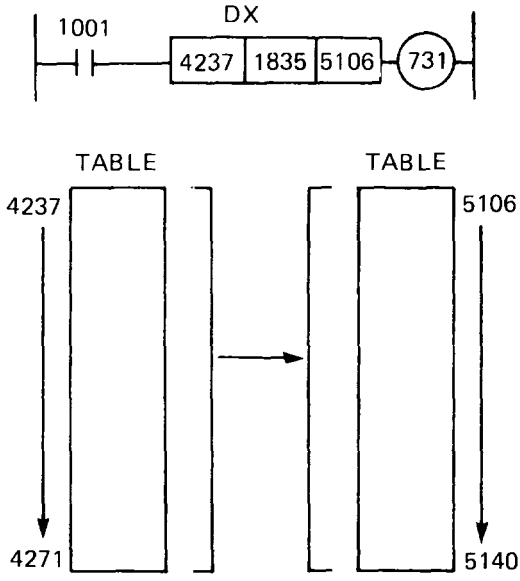


Figure 100. Sample 4XXX to 5XXX Move

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.

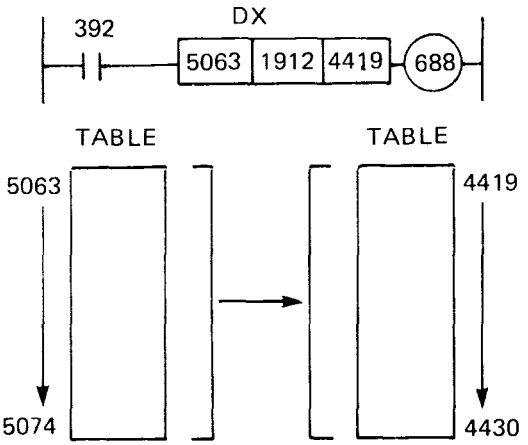


Figure 101. Sample 5XXX to 4XXX Move

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.

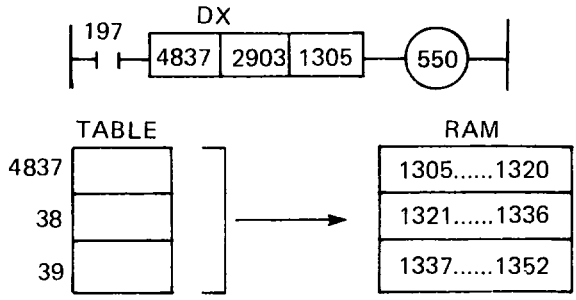


Figure 102. Example Matrix Sequencer Move

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

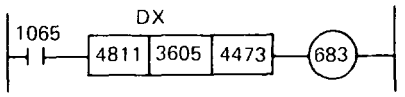
These two DX functions allow a table of registers to be sorted by their numerical contents. The entire table is sorted in one scan upon closure of the A element. The sort is performed only once upon closure of the A element; if additional sorts are required, the A element must be opened and then closed 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

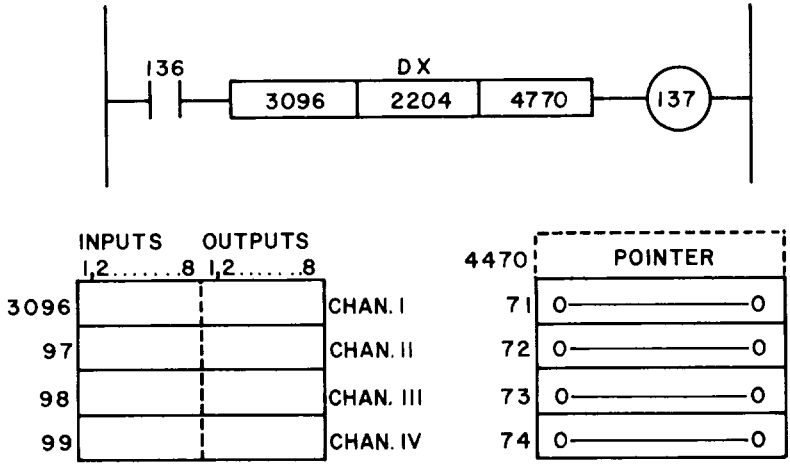


Figure 104. Example I/O Monitoring

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