SECTION 5. SPECIAL FUNCTIONS

INTRODUCTION

This Section of the manual explains all special functions for the PC-1100 and PC-1200 programmable controllers (PC). An explanation of the Battery Status Coil is also included, even though it is considered to be a dedicated function logic coil.

Table 5-1 lists the available special functions in alphabetical order. Table 5-1 lists several types of information about each function: mnemonic, description, application, and applicable programmable controller.

There are several versions of the PC-1100 and PC-1200 programmable controllers. Catalog numbers are labeled on the side of the programmable controller's enclosure. The catalog number identifies the type of programmable controller, the type of power supply used, the type of programmable function set used, and the amount of memory used:

- PC-1100-x01y
- PC-1100-x02y
- PC-1100-x03y
- PC-1100-x05y
- PC-1200-x02y
- PC-1200-x04y

where: x represents a number associated with the type of power supply that is used by the programmable controller

01, 02, 03, 04, and 05 represent the special function set used

y represents a number associated with the amount of memory available.

These catalog numbers are used to identify versions of the PC-1100 and PC-1200. In Table 5-1, if a version supports a function, an "x" is entered in the Table under the appropriate version number. If a version does not support a function, a "-" is entered. It is important to be aware of which functions are supported by the programmable controller that you are using. Since we are only concerned with which special functions are supported by the PC, the type of power supply and amount of memory are not necessary in Table 5-1.

There are some operational differences between the way a PC-1100 and a PC-1200 will solve or implement a special function. Those differences are discussed in the special function description. Changes between the PC-1100 and PC-1200 were made to enhance the user's programming capability and to have minimal impact on existing programs when updating current processors. However, there are a few functions that operate very differently. In these cases, two descriptions are presented: one for the PC-1100 and one for the PC-1200. Refer to Table 5-1 to determine if there is more than one description for a particular function.

Special Function Descriptions

The special function descriptions contained in this section appear in alphabetical order. Each function description is self-contained, in that all figure and table numbers relate only to the individual function.

To help identify which programmable controller is available for a particular special function, each special function description includes a box at the top of the page which lists the catalog numbers. The words "SUPPORTED" or "NOT SUPPORTED" beside each catalog number indicate the special function's availability.

A Note About the LT (Literal) Special Function

When programming a PC with a CRT Program Loader (NLPL-780) or older versions of the Mini Loader (NLPL-789), special functions must be entered using the Literal (LT) special function. LT is a template that, when coded with a number, will operate as a particular special function.

When programming either the PC-1100 or PC-1200 with an Advanced Program Loader (APL), special functions should be entered into the ladder program by typing in a unique two character mnemonic that represents the special function. LT can be used with the APL, but it is suggested that the mnemonic method be used. Mnemonic special functions offer checking for proper operand assignment, register types, and number of inputs.

The LT special function does not determine the validity of operand definitions. Therefore, the user is responsible for valid entries as defined in this manual. Failure to comply with defined entries can result in program or system misapplication.

However, the LT special function does allow the flexibility of using existing program loaders with newer programmable controllers. Newer programmable controllers may support special functions that are not supported by older program loaders. The LT special function can be used with the new controller to access new special functions until the program loader can be updated with the mnemonic version of the special function.

Caution

Avoid using the LT special function whenever possible. Misapplication of LT could result in program or system misapplication.

Note

For more information on Literal functions, refer to the reference sheet in this section for the LT special function.

TABLE 5-1. SPECIAL FUNCTIONS

Mnemonic	Description	Application	PC-1100				PC-1200	
			01	02	03	05	02	04
AD/SB	Add/Subtract	Math Function	x	x	x	x	x	x
AM	AND Matrix	Matrix Function	-	x	x	x	x	x
AR	ASCII Receive - PC-1100	Serial Port Command	-	x	x	-	-	-
AR	ASCII Receive - PC-1200	Serial Port Command	-	-	-	-	x	x
* AS	Ascending Sort	Table Operation	-	x	x	x	x	x
* AT	ASCII Transmit	Serial Port Command	-	x	x	x	x	x
BD/DB	Conversions (binary to decimal)	Conversions	x	x	x	x	x	x
BF	Bit Follow	Coils	x	x	x	x	x	x
* BO	Bit Operate	Table Operation	x	x	x	x	x	x
BS/BC	Latches	Coils	x	x	x	x	x	x
BT	Block Transfer	Move Function	-	x	x	x	x	x
CG	Continuous Group Select	Scan Alterantion	-	x	x	x	x	x
CM	Complement Matrix	Matrix Function	-	x	x	x	x	x
* CP	Configure Port	Serial Port Command	-	x	x	x	x	x
CR	Control Relay	Coils	x	x	x	x	x	x
* DV	Divide	Math Function	-	x	x	x	x	x
EQ/GE	Comparisons	Comparisions	x	x	x	x	x	x
* FI/FO	FIFO Stack	Stack Operation	-	x	x	x	x	x
* FI/LO	FILO Stack	Stack Operation	-	x	x	x	x	x
IM	Indirect Move	Move Function	x	x	x	x	x	x
* LC	Loop Controller	Process Control	-	-	x	-	-	x
* LR	Latch Read	I/O Function	-	x	x	x	x	x
LS	Lock Scan	Scan Alteration	x	x	x	x	x	x
LT	Literal	Special Support	x	x	x	x	x	x
MB	Move Byte	Move Function	x	x	x	x	x	x
* MP	Multiply	Math Function	-	x	x	x	x	x
* MR	Master Control Relay	Coils	x	x	x	x	x	x
MV	Move	Move Function	x	x	x	x	x	x
NR/NL	N-Bit Serial Shift Registers	Shift Registers	x	x	x	x	x	x
OM	OR Matrix	Matrix Function	-	x	x	x	x	x
OT/CT	Open Table/Close Table	Table Operation	-	x	x	x	x	x
* PT	Port Transmit	Serial Port Command	-	x	x	-	x	x
* RP	Restore Program Counter	Scan Alteration	-	x	x	x	x	x
* RW	Reset Watchdog Timer	Scan Alteration	-	x	x	x	x	x
* SK	Skip	Coils	x	x	x	x	x	x
SM	Search Matrix	Matrix Function	-	x	x	x	x	x
SP	Save Program Counter	Scan Alteration	-	x	x	x	x	x
* SQ	Square Root	Math Function	-	x	x	x	x	x
* TL/TO	Table Lookup/Table Lookup Ordered	Table Operation	-	x	x	x	x	x
* TR	Table-to-Register Move	Move Function	x	x	x	x	x	x
* TS/TT	Timers	Timers	x	x	x	x	x	x
* UA	Unit Address	Serial Port Command	-	x	x	-	x	x

"*" = modified for PC-1200, "x" = supported, "-" = not supported

TABLE 5-1. SPECIAL FUNCTIONS (Cont'd)

Mnemonic	Description	Application	PC-1100				PC-1200	
			01	02	03	05	02	04
UC/DC	Up Counter/DC Counter	Counters	x	x	x	x	x	x
* UI	Update Immediate	I/O Function	x	x	x	x	x	x
* US	Update Select	I/O Function	-	x	x	x	x	x
XM	XOR Matrix	Matrix Function	-	x	x	x	x	x
Special Coil								
CR0128	Battery Status Coil (PC-1100)	PC Maintenance	x	x	x	x	-	-
CR1024	Battery Status Coil (PC-1200)	PC Maintenance	-	-	-	-	x	x

"*" = modified for PC-1200, "x" = supported, "-" = not supported

PC-1100-x01y: SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The Add (AD) and Subtract (SB) functions add and subtract four-digit decimal numbers (0 through 9999), producing a four-digit decimal number (0 through 9999) result. Through programming, these functions can provide double-precision addition and subtraction for numbers in excess of four digits. See the AD/SB "Application" description for double-precision operation. AD/SB function symbology is shown in Figure 1.

Operand 1 and Operand 2 are added or subtracted when the enable circuit changes from non-conducting to conducting. Operand 1 comes from an input, output, or holding register. Operand 2 comes from one of these registers or is programmed as a constant. The result is placed in the destination register, which is an output or holding register.

Although the limits on the operands and the result are decimal (0000 through 9999), addition and subtraction are performed on equivalent binary numbers. For this reason, the register values for Operands 1 and 2 must be stored in binary form. Similarly, the register value of the result is in binary form. The Binary to Decimal (BD) function is used to convert binary numbers to Binary-Coded-Decimal (BCD); the Decimal to Binary (DB) function is used to convert BCD to binary form.

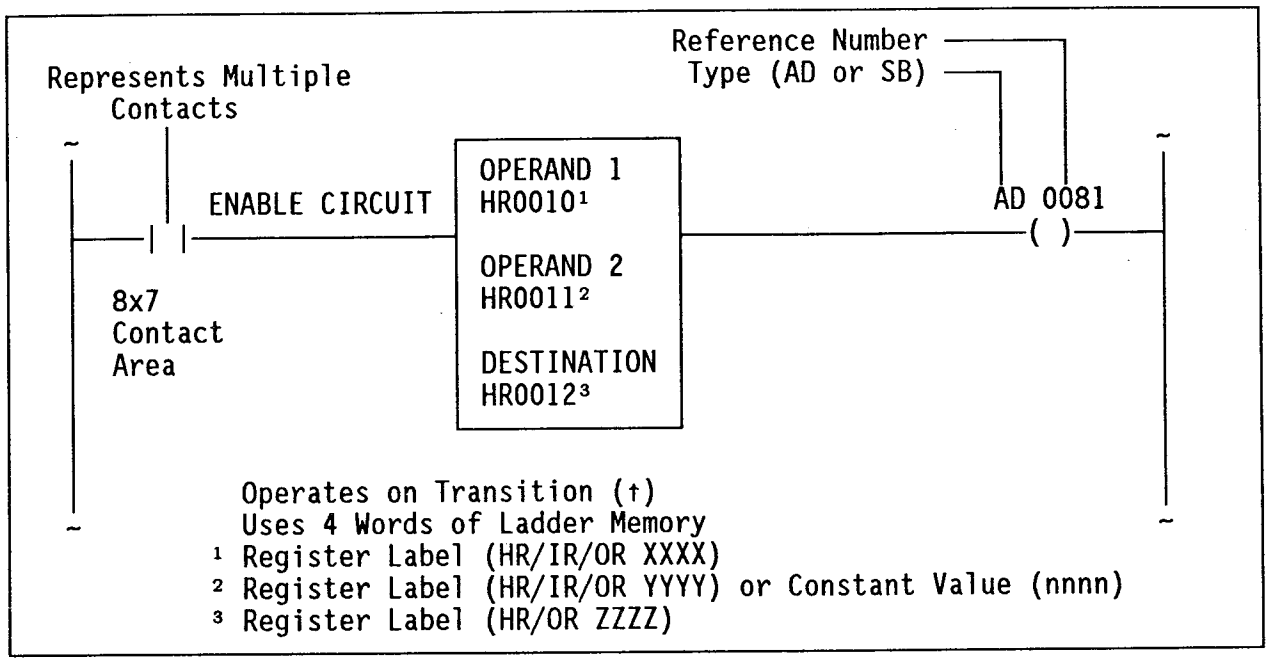


Figure 1. Add/Subtract (AD/SB)

ADD (AD)

Operands 1 and 2 are added, and the result is placed in the assigned designation register. If the result is equal to or greater than 10000, 10000 is subtracted and the coil is energized.

SUBTRACT (SB)

Operand 2 is subtracted from Operand 1, and the result is placed in the designation register. If the result is less than 0000, the SB coil energizes and the amount of underflow (i.e., amount less than 0000) is placed in the designation register.

In either case, when the enable circuit does not conduct, the coil (AD or SB) is de-energized. Forcing the coil only affects its contacts and output circuit (if any); the function continues under enable circuit control.

SPECIFICATIONS

OPERAND 1

- ADD (AD) - The first addend
- SUBTRACT (SB) - The minuend
- The Operand 1 value is held in a specified Holding Register (HR), Input Register (IR), or Output Register (OR).

OPERAND 2

- ADD (AD) - The second addend
- SUBTRACT (SB) - The subtrahend
- Operand 2 is either a constant (0001 through 9999) or is held in a specified Holding Register (HR), Input Register (IR), or Output Register (OR).

Note

Both operands must be in binary form.

DESTINATION

The Destination is a specified Holding Register (HR) or Output Register (OR).

COIL

- ADD - Operand 1 + Operand 2 = Destination register contents. The AD coil energizes if the result is greater than 9999.
- SUBTRACT - Operand 1 - Operand 2 = Destination register contents. The SB coil energizes if the result is less than 0000.

AD/SB TRUTH TABLE

See Table 1.

TABLE 1. ADD/SUBTRACT TRUTH TABLE

Enable Circuit	Result
0	None. The AD or SB coil de-energizes.
↑	AD/SB - The result is placed in the destination register. The AD coil is energized when overflow occurs; the SB coil is energized when the result is less than zero.
1	The coil remains in the last state.

APPLICATIONS

When establishing deviation alarms for a process, it may be advantageous to allow different alarm ranges around the set point rather than the same deadband on each side of the set point. AD/SB functions can establish the band or range of operation around the set point.

Figure 2 shows a circuit for one such method of establishing this range of operation. The upper and lower limits are independently established. If the upper and lower limits were always equal and opposite, a single register could be used as Operand 2 for both the AD and SB functions to save a word of memory. If the range is never adjusted. Operand 2 for both functions can be set as a constant value.

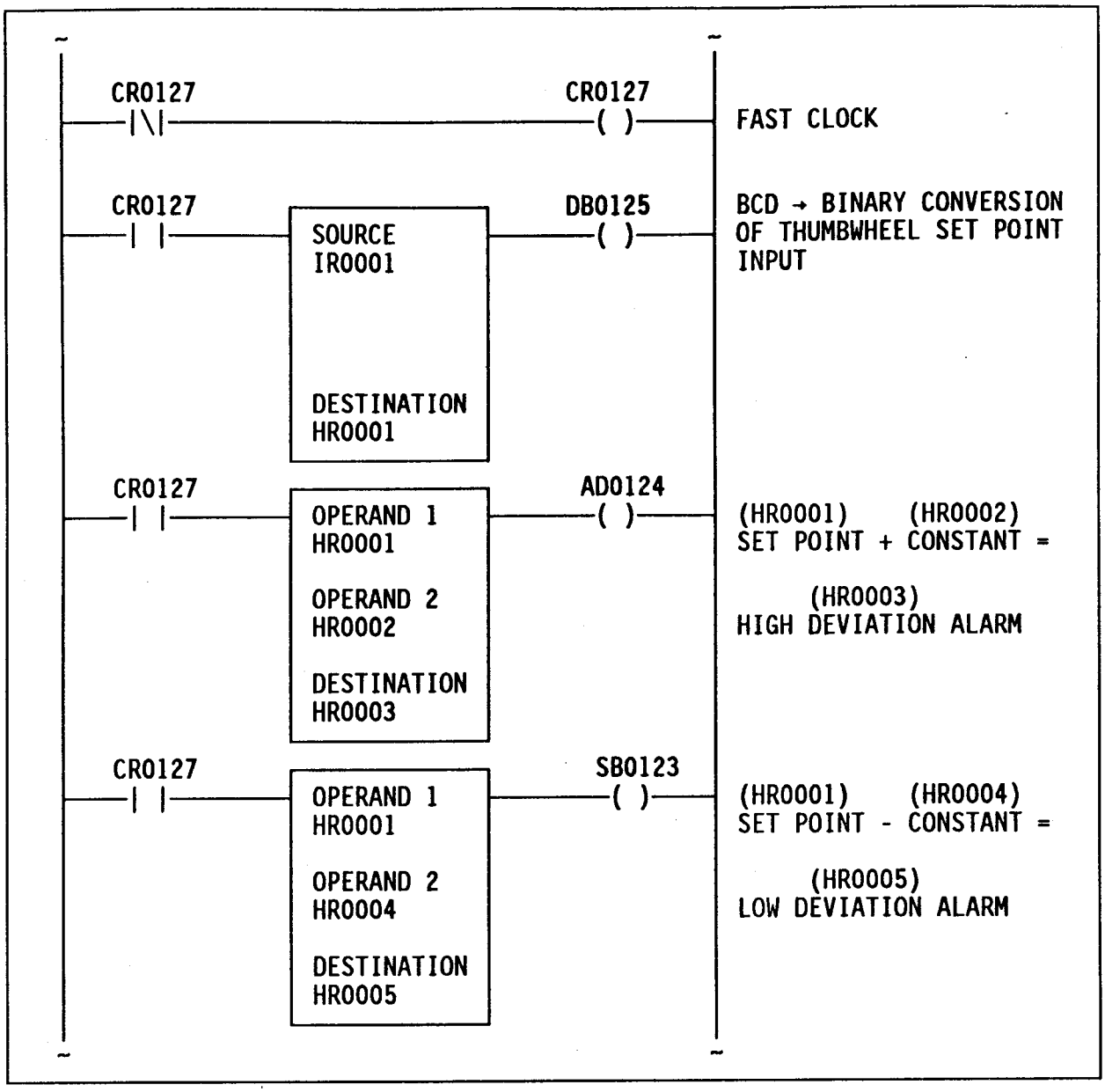


Figure 2. Establishing the Range of Operation

An example of a double-precision application for the AD function is shown in Figure 3. This programming configuration is used when numbers of more than four digits are added.

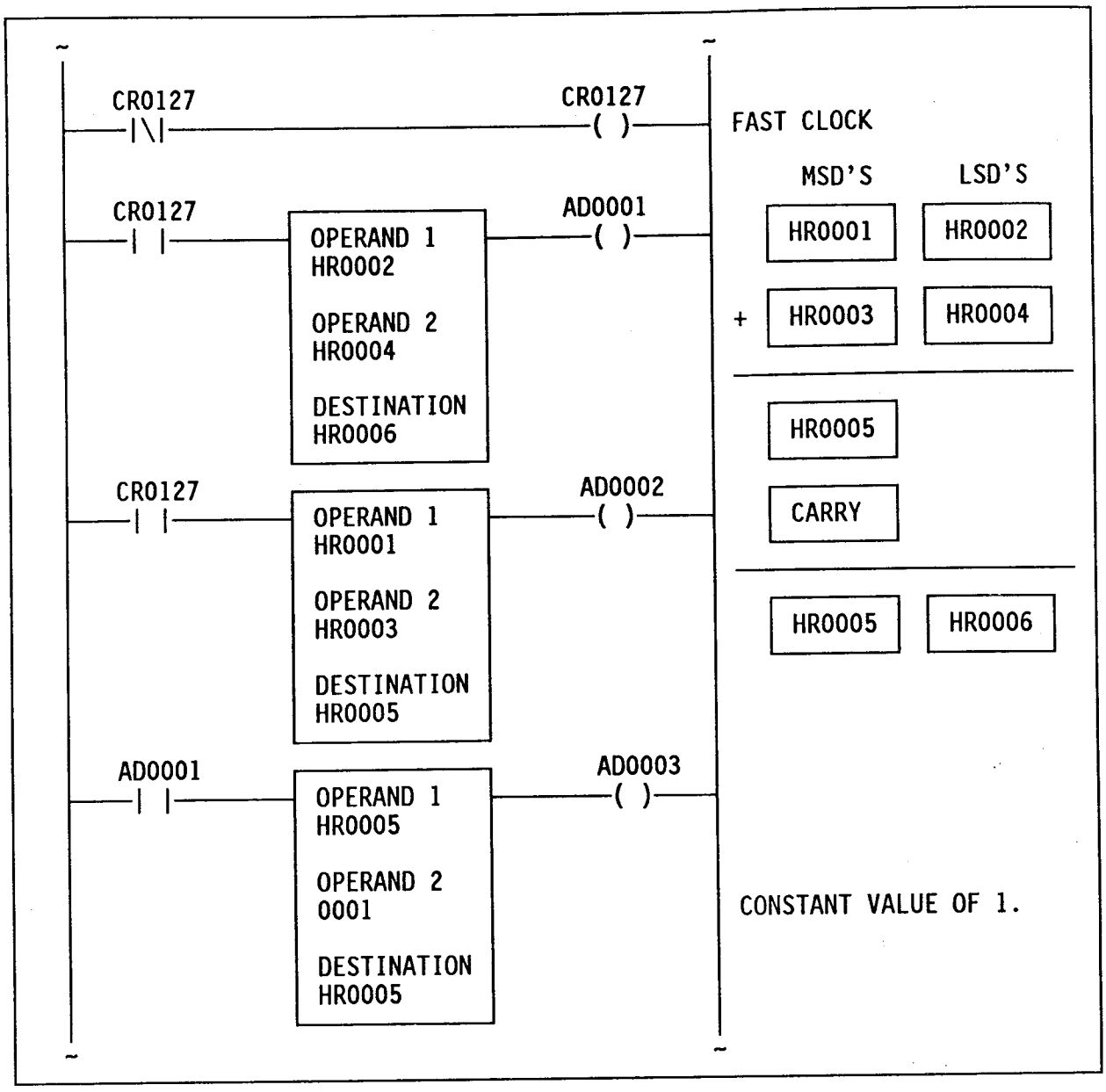


Figure 3. Double-Precision Add

AM - AND MATRIX

PC-1100-x01y: NOT SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

AND Matrix (AM) function logically AND's a pair of matrices on a bit-per-bit basis; then, it places the result in a designation matrix. A matrix is defined as a table of registers handled on a bit-by-bit basis. AM function symbology is shown in Figure 1.

The AM operation occurs when the enable circuit changes from non-conducting to conducting. The original pair of matrices is unaffected by the operation as shown in Table 1.

OP CODE

Op Code 90 defines the Literal (LT) as AM. Whether or not the Literal function should be used depends upon the capability of your program loader. It is recommended that the mnemonic function be used whenever possible. Refer to the Introduction in this Section and the LT function description.

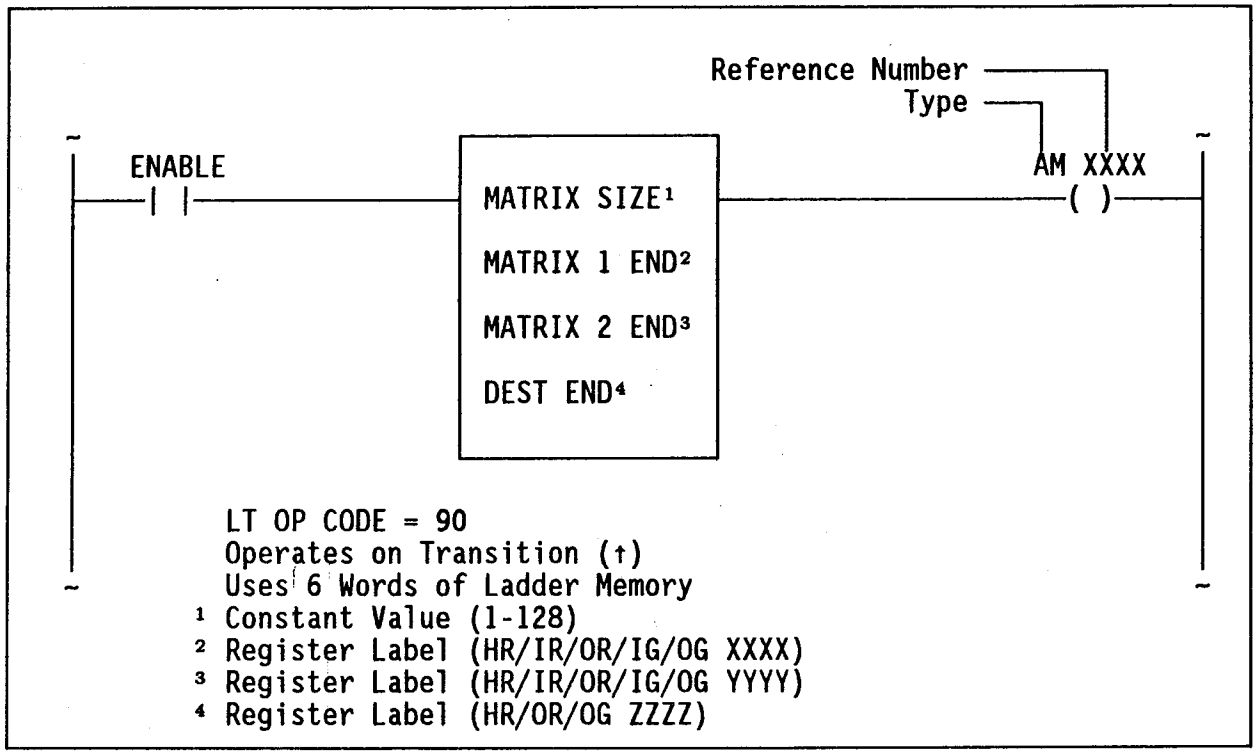


Figure 1. AND Matrix (AM)

TABLE 1. AM TRUTH TABLE SAMPLE

Matrix 1 Bit N	Matrix 2 Bit N	Destination Matrix Bit N
0	0	0
0	1	0
1	0	0
1	1	1

Note

N is the same respective bit in all three matrices.

SPECIFICATIONS

OPERAND 1 - MATRIX SIZE

The matrix size is a constant value that defines the number of registers included in the matrix. The range is 1 through 128, which is subject to the limitations shown in Table 2.

TABLE 2. AM END REGISTER

Type	PC-1100	PC-1200-1020/1040	PC-1200-1041/1041/1042	PC-1250
HR	≤ 1792 [1]	≤ 1792	≤ 1792	≤ 1792
IR	≤ 8	≤ 32	≤ 64	≤ 128
OR	≤ 8	≤ 32	≤ 64	≤ 128
IG	≤ 4	≤ 4	≤ 8	≤ 16
OG	≤ 8	≤ 32	≤ 64	≤ 128

[1] For the PC-1100, the maximum number of holding registers depends on memory size, as described in Section 4.

OPERAND 2/OPERAND 3 - MATRIX 1/MATRIX 2 END

Matrix 1 End and Matrix 2 End define the type and number of the last register in Matrix 1 End and Matrix 2 that will be AND'ed. The type and number are limited as shown in Table 2.

Note

The highest Holding Register reference number acceptable is dependent upon the memory and user program size.

OPERAND 4 - DESTINATION END

The destination end defines the type and number of the last register in the matrix containing the results of the AM function. The type and number are limited, as shown in Table 3.

TABLE 3. AM DESTINATION END REGISTER

Type	PC-1100	PC-1200-1020/1040	PC-1200-1041/1041/1042	PC-1250
HR	≤ 1792 [1]	≤ 1792	≤ 1792	≤ 1792
OR	≤ 8	≤ 32	≤ 64	≤ 128
OG	≤ 8	≤ 32	≤ 64	≤ 128

[1] For the PC-1100, the maximum number of holding registers depends on memory size, as described in Section 4.

COIL

The coil energizes when the enable circuit is in transition between off and on. The coil remains on until the enable circuit is turned off.

AM TRUTH TABLE

See Table 4.

TABLE 4. AM TRUTH TABLE

Enable	Result
0	The coil is de-energized. Matrix 1, Matrix 2, and the destination are unchanged.
↑	Matrix 1 and Matrix 2 are AND'ed and the result is placed in a destination matrix. Matrix 1 and Matrix 2 are unaffected by the operation. The coil is energized if the result is non-zero.
1	Matrix 1, Matrix 2, the destination and the coil status are unchanged.

APPLICATIONS

The AM function is used to perform a masking operation. If, for example, the register in Figure 2 is to be masked, it is desired that only the lower-eight bits of HR0001 be transmitted to OR0001. The upper-eight bits must be set to zero. The desired action can be accomplished by AND'ing HR0001 with another register configured as shown in Figure 2.

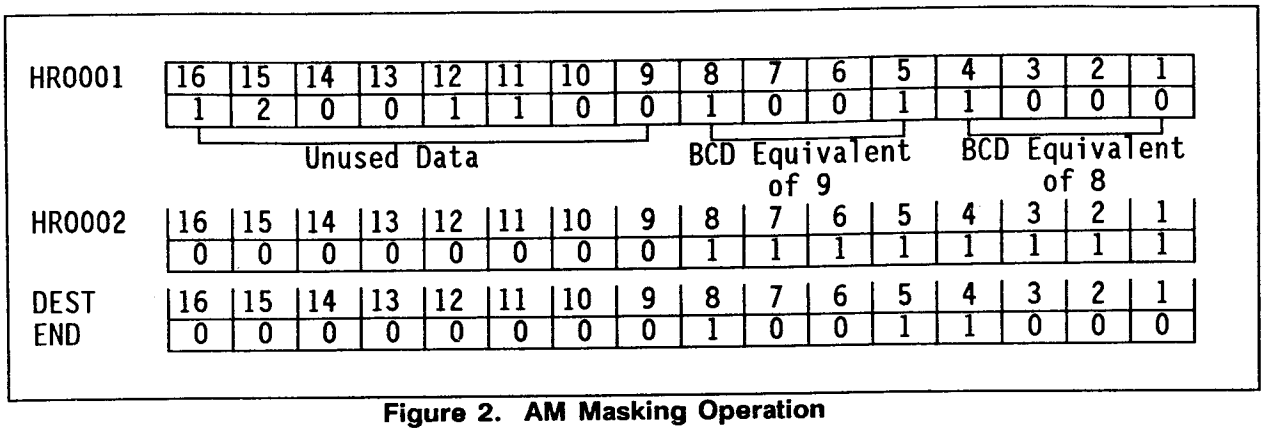


Figure 2. AM Masking Operation

Anytime a bit is AND'ed with zero, zero is the result. Anytime a bit is AND'ed with a one, the result depends upon the other bit's state as follows: 1 AND 1 = 1; 1 AND 0 = 0.

Figure 3 shows the ladder diagram for the AM function. When IN0001 is changed from open to closed, HR0001 is AND'ed with HR0002 and the results are placed in OR0001. The coil energizes if the enable circuit remains closed and the result is not zero.

Figure 4 illustrates a pair of matrices that are AND'ed together.

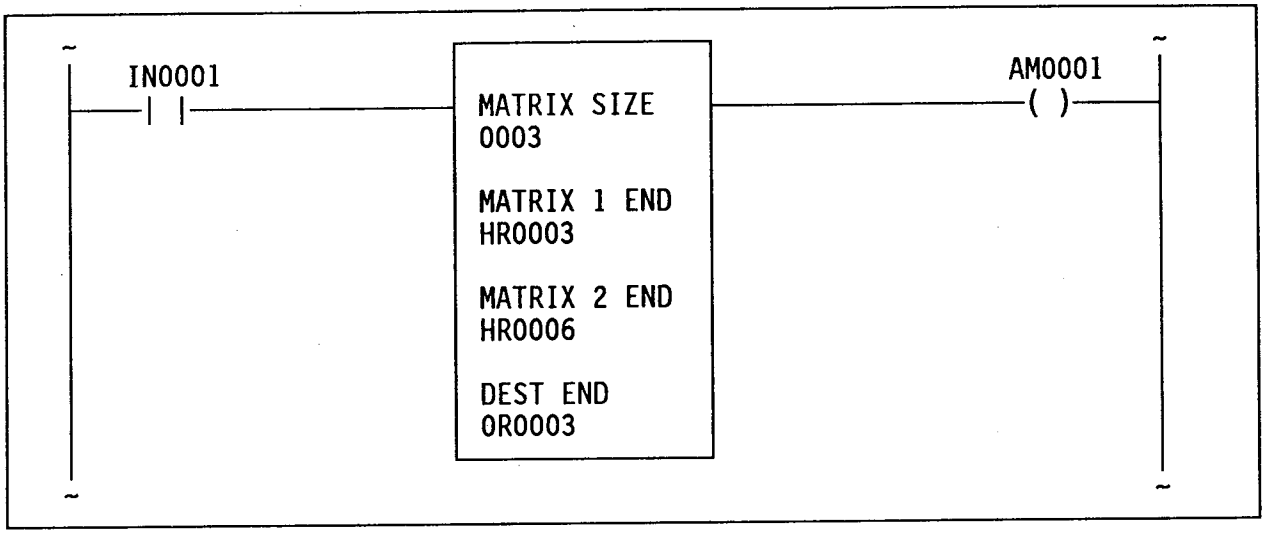


Figure 3. AM Application

MATRIX 1

BIT NO.	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
HR0001	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

BIT NO.	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
HR0002	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

BIT NO.	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
HR0003	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0

MATRIX 1 END

MATRIX 2

BIT NO.	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
HR0004	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

BIT NO.	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
HR0005	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

BIT NO.	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
HR0006	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

MATRIX 2 END

DESTINATION MATRIX

BIT NO.	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
OR0001	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

BIT NO.	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
OR0002	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

BIT NO.	48	47	46	45	44	43	42	41	40	39	68	37	36	35	34	33
OR0003	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0

DESTINATION END

Figure 4. A Pair of Matrices AND'ed

PC-1100-x01y: NOT SUPPORTED	PC-1100-x05y: NOT SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: NOT SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: NOT SUPPORTED

DESCRIPTION

This function receives asynchronous communication data from peripheral devices through Port A and stores the data in holding registers, per the specifications contained in the special function block. The Op Code for ASCII Receive is 48.

The RECEIVE ENABLE Input, when conducting, enables the special function to capture data received by Port A and places the data in the table of Holding Registers. When the input transitions from non-conducting to conducting the function status byte, port error conditions, table pointers and coil status are all reset. When the input transitions back to a non-conducting state, Port A reverts back to program loader communications Data already in temporary storage is placed in Holding Registers and pointers and status indicators are kept intact until the input transitions from non-conducting to conducting again.

Operand 1, TABLE LENGTH, can be a constant (CV), a Holding Register (HR), an Input Register (IR), an Output Register (OR), an Input Group (IG) or an Output Group (OG). This operand defines the total length of the HR table. The total length of the table equals the number of HR's for the message plus one HR for the table pointer. The maximum size is 256 registers and the maximum message length is 255 registers (510 bytes). The last HR of the table is reserved. It contains a relative byte pointer which points to the byte in the table in which the most recent data from Port A has been stored.

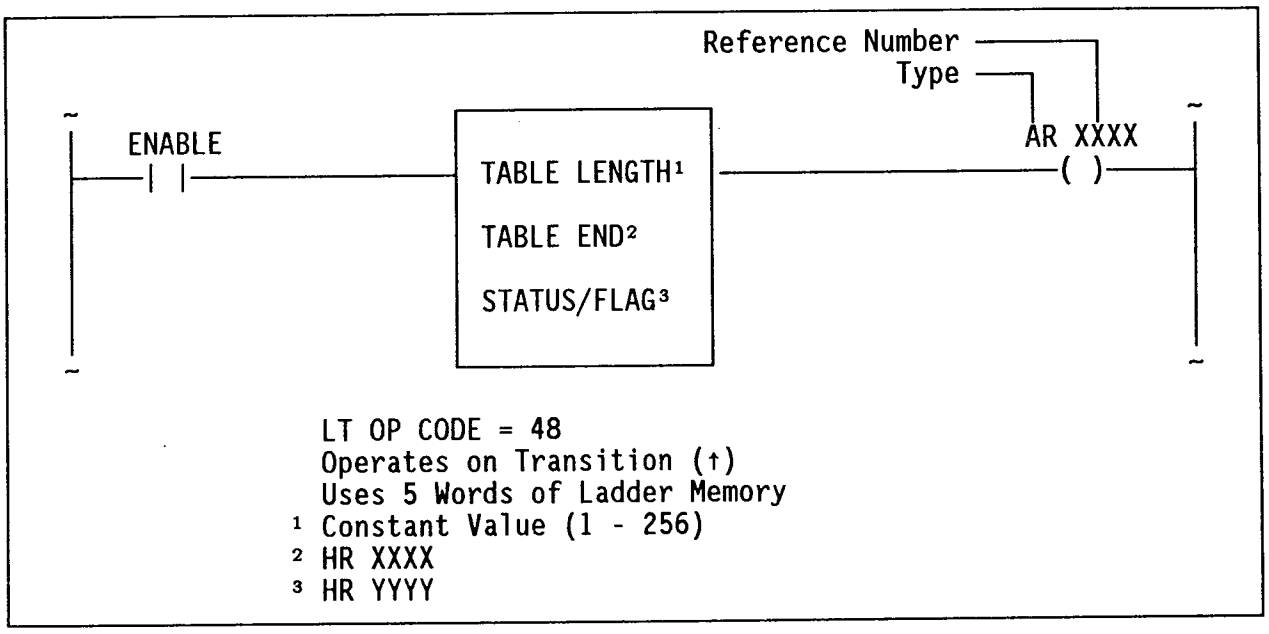


Figure 1. ASCII Receive (AR)

Operand 2, TABLE END, specifies a Holding Register whose contents define the last Holding Register of the table. The TABLE END must be specified as a Holding Register, HR.

Operand 3, STATUS/FLAG, is a dual purpose operand defined by a Holding Register. The FLAG portion of the operand contains an user defined End-Of-Message (EOM) character (optionally selected in the byte which gives error diagnostic information about the port. An error condition can be recognized by a STATUS/FLAG register value greater than 512. When an error is detected, data reception and transfer to Holding Registers is not disabled. This holding register is shown in Figure 2.

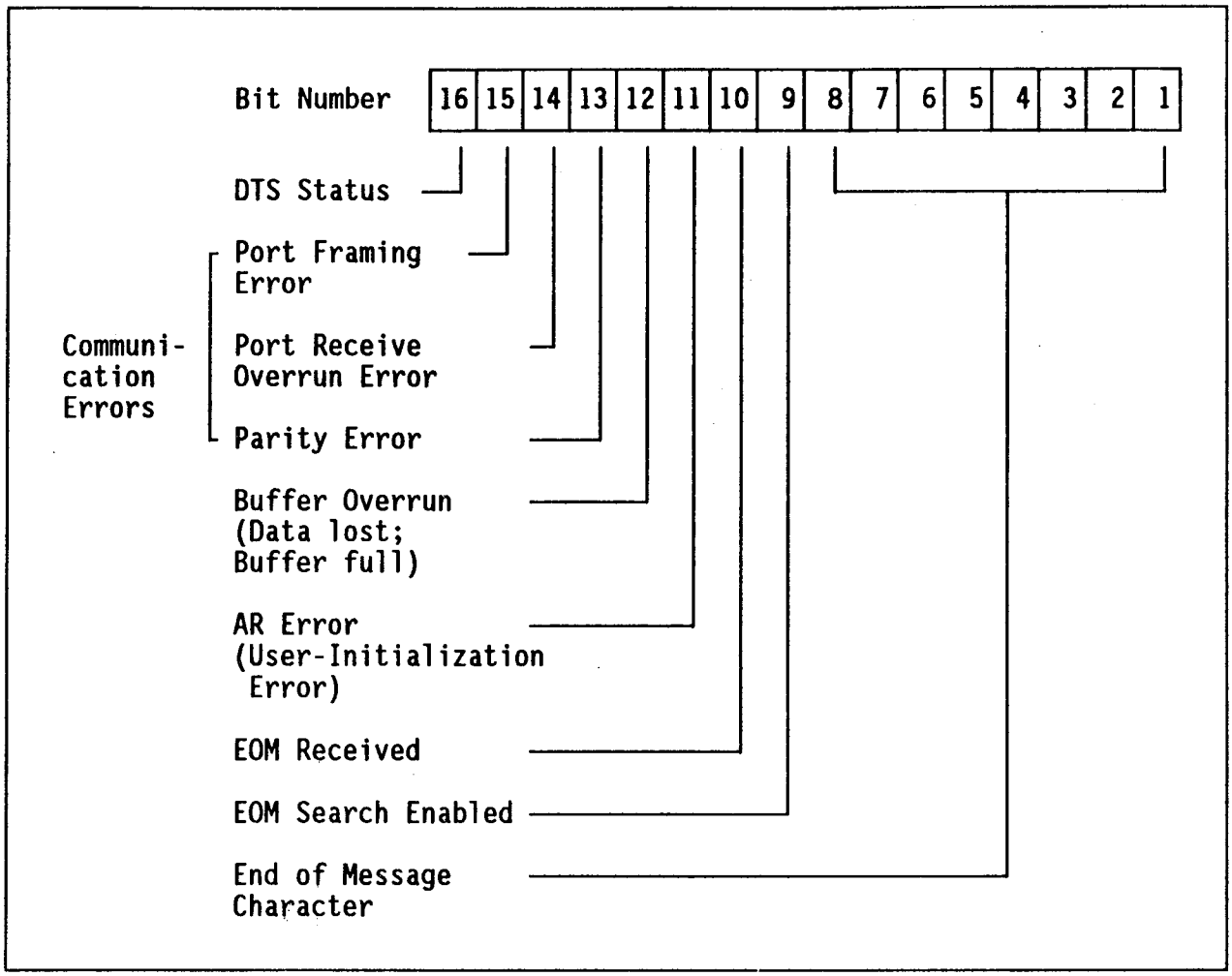


Figure 2. STATUS/FLAG Register

The ASCII Receive coil energizes to indicate that:

1. A byte received through Port A matches the EOM character or
2. The Holding Register Table is full.

When the coil is energized, Port A is disabled (i.e., Data Terminal Ready (DTR) is false). Under condition (1) the coil de-energizes when the pointer is zeroed or changed to reference a Holding Register different from the TABLE END.

OPERATION

When the ENABLE input to the function block is conducting, Port A will accept inbound data. Data format is not restricted to ASCII. Data received is stored temporarily in a 140 byte buffer. Each byte is compared to the End-Of-Message (EOM) character if that option is selected. Each time the ladder scan reaches the AR function the data in temporary storage is placed in the Holding Register table and the pointer is updated accordingly. The bytes are stored in the HR table with the first byte received being placed in the low order byte of the first holding register, the second byte received being placed in the high order byte of the first holding register, etc. When the HR table is full, the AR function is disabled and any data received by Port A will be lost.

When the ENABLE input is disabled while receiving data, the data in temporary storage will be transferred to HRs, if possible (an error bit will be set in the STATUS byte if the HR's are full. The partial message in HRs will be retained until the ENABLE line is conductive again, at which time the HR pointer will be reset and incoming data will overwrite the partial message. When the ENABLE line is not conducting or when the PC is not in the Run Mode, any incoming data will be treated as Program Loader communications.

The ASCII Receive function will be disabled (the DTR line of Port A will be set false) if:

(1) An End of Message (EOM) character is encountered,
(2) The temporary buffer is full or
(3) The Holding Register table is full.

To re-initialize communications after an EOM character has been encountered, zero the pointer.

Once the temporary buffer is full, the AR function will attempt to transfer the data to the Holding Register table and resume communications. However, if the table is full, the pointer must either be zeroed or changed to reference a HR different than the Table End to continue operation.

Note

The "Data Set Ready" (DTR) line for Port A must be held high for proper operation.

TABLE 1. AR TABLE LENGTH

Type	PC-1100 Limit
CV	≤ 256
HR	≤ 1792 [1]
IR	≤ 8
OR	≤ 8
IG	≤ 4
OG	≤ 8

[1] For the PC-1100, the maximum number of holding registers depends on memory size, as described in Section 4.

TABLE 2. AR TRUTH TABLE

Enable	Result
0	Port A continues to receive data (if the function was enabled) until either an EOM character is encountered or the table is full.
↑[1]	Table pointer, coil status, status byte and errors are reset. Data is transferred from Port A into the table of Holding Registers.
1	Port A continues to receive data (if the function was enabled) until either an EOM character is encountered or the table is full.

[1] Transition from OFF to ON.

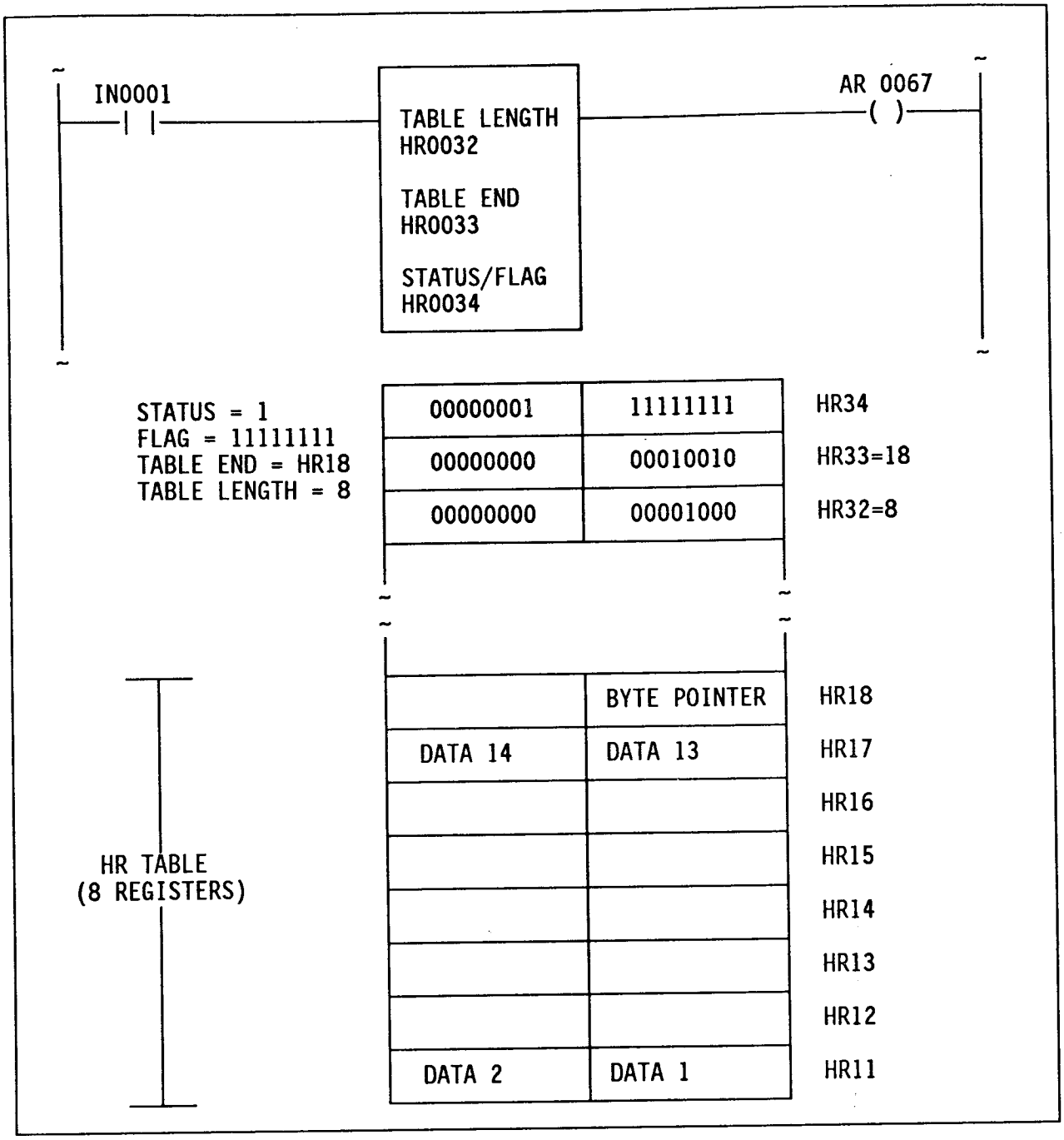


Figure 3. AR Example

AR - ASCII RECEIVE

WORKS WITH ER

PC-1200 VERSION

PC-1100-x01y: NOT SUPPORTED	PC-1100-x05y: NOT SUPPORTED
PC-1100-x02y: NOT SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: NOT SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

This function receives asynchronous communication data from peripheral devices through Port A or Port B and stores the data in holding registers, per the specifications contained in the special function block. The Op Code for ASCII Receive is 48. AR function symbology is shown in Figure 1.

The RECEIVE ENABLE Input, when conducting, enables the function to capture data received and place the data in the table of Holding Registers. When the input transitions from non-conducting to conducting the function status byte, port error conditions, table pointers and coil status are all reset. When the input is in a non-conducting state the port reverts to program loader communications. Data in temporary storage is lost when ENABLE on the ASCII Receive function goes low.

Operand 1, TABLE LENGTH, can be a constant (CV), or the contents of a Holding Register (HR), an Input Register (IR), an Output Register (OR), an Input Group (IG) or an Output Group (OG). This operand defines the total length of the HR table. The total length of the table equals the number of HR's for the message plus one HR for the table pointer. Therefore, the minimum value for TL is 2. The maximum size is 256 registers and the maximum message length is 255 registers (510 bytes). The last HR of the table is reserved. It contains a relative byte pointer which points to the byte in the table in which the most recent data has been stored.

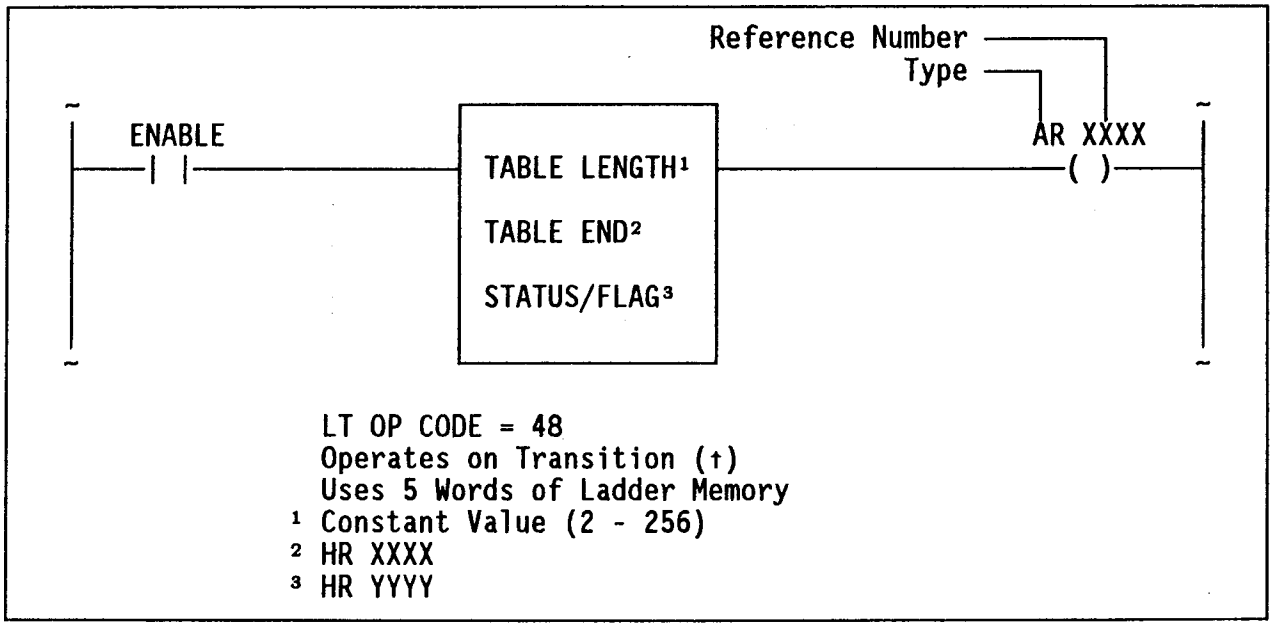


Figure 1. ASCII Receive (AR)

Operand 2, TABLE END, specifies a Holding Register whose contents define the last Holding Register of the table. The TABLE END must be specified as a Holding Register, HR.

Operand 3, STATUS/FLAG, is defined by a pair of registers: the Status/Flag and the Port Mode register. The Status/Flag register is divided into two bytes so that the most significant byte indicates the status of the function and the least significant byte holds a user defined End-Of-Message (EOM) character (optionally selected in the STATUS byte). The STATUS portion of the operand occupies the upper byte which gives error diagnostic information about the port. An error condition can be recognized by a implementing a bit pick or mask with a matrix function on the Status/Flag Register. When an error is detected, data reception and transfer to Holding Registers is not disabled. Refer to Figure 2.

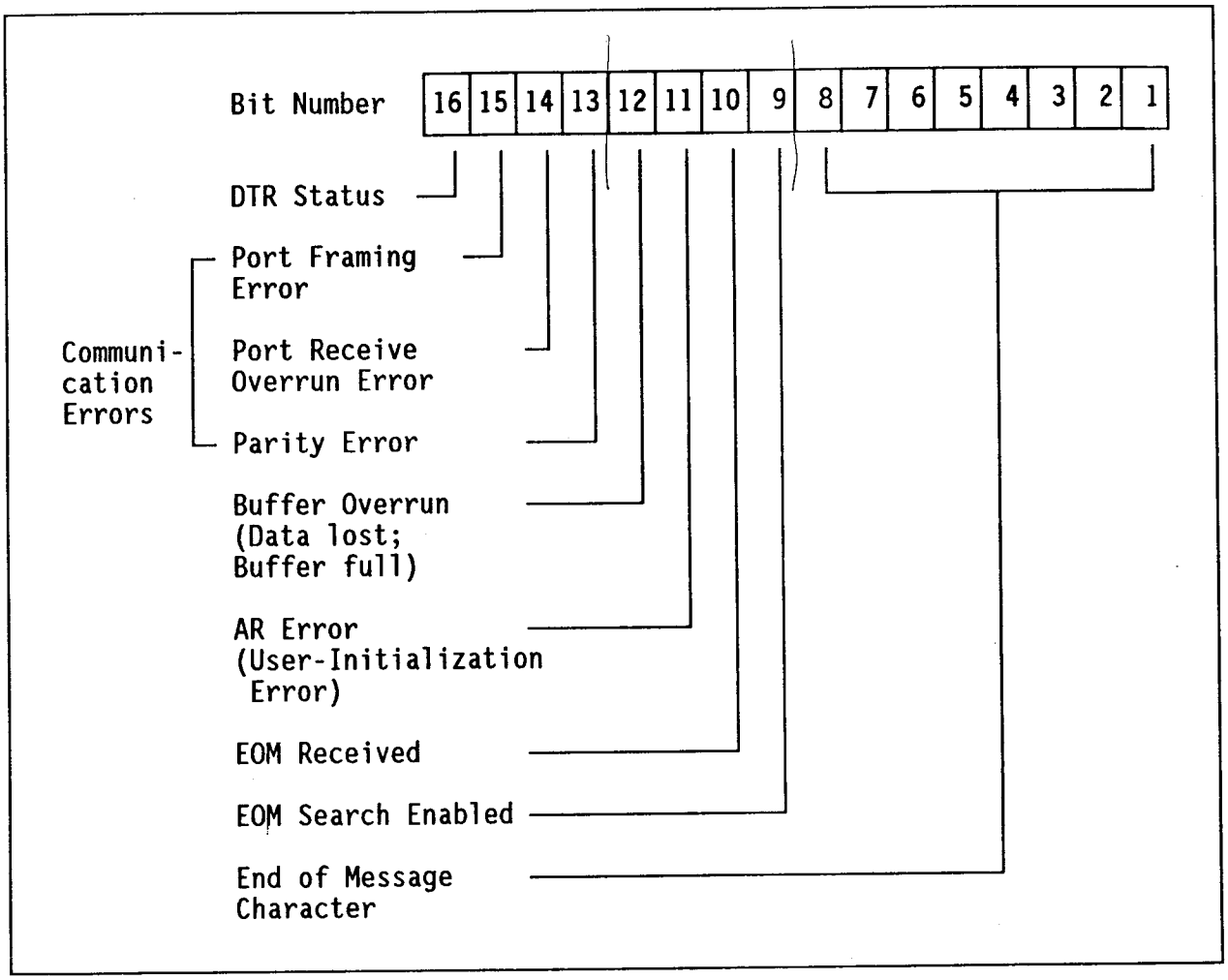
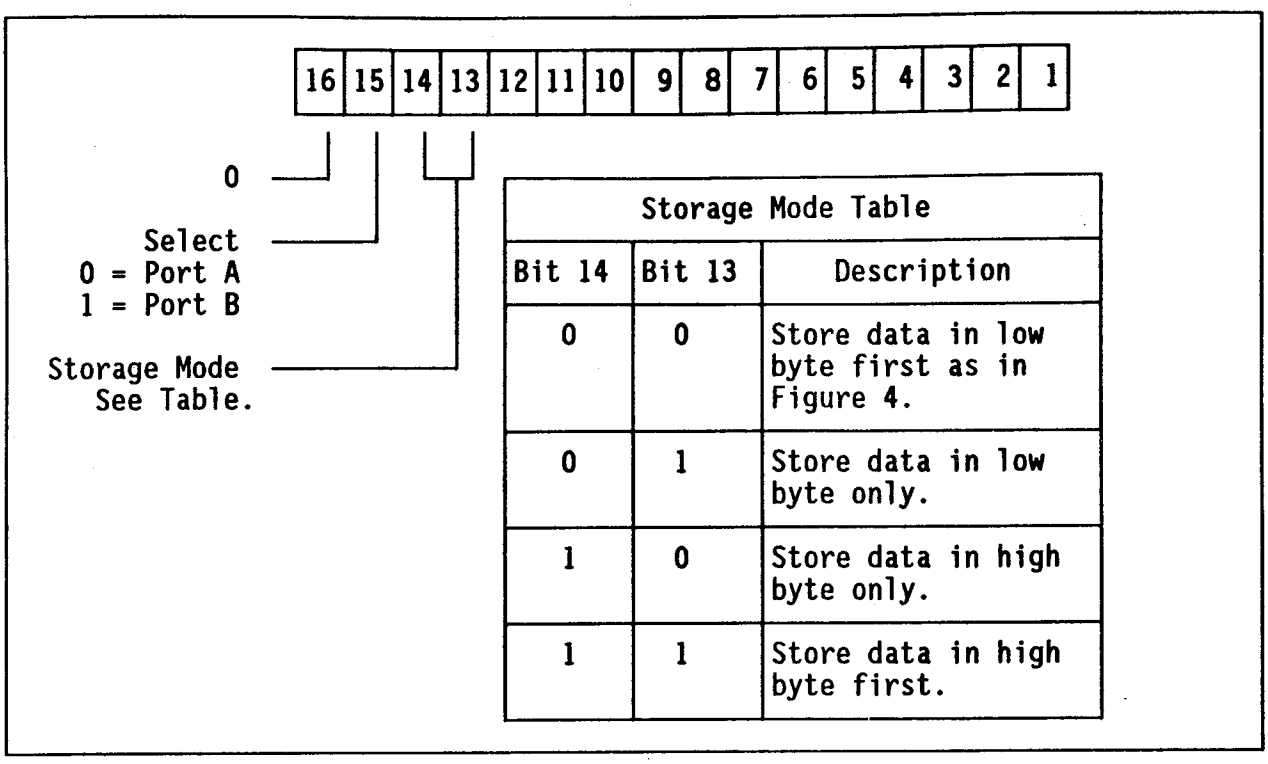


Figure 2. STATUS/FLAG REGISTER

The register preceeding Operand 3 defines which port is to be used and the method through which data is to be stored. This is known as the Port Mode register. The port can only be designated (by setting bit 15 of the Port Mode register) while ASCII RECEIVE is disabled. The port mode register is defined in Figure 3.



Storage Mode Table		
Bit 14	Bit 13	Description
0	0	Store data in low byte first as in Figure 4.
0	1	Store data in low byte only.
1	0	Store data in high byte only.
1	1	Store data in high byte first.

Figure 3. Port Mode Register

Note

DTR sets the most significant bit. See Figure 2.

The ASCII Receive coil energizes to indicate that:

1. A byte received through the port matches the EOM character or

2. The Holding Register Table is full.

When the coil is energized, the port is still enabled (i.e., Data Terminal Ready (DTR) is true). DTR will be disabled when the transparent data buffer is almost full (520 bytes). Under condition (1), the coil de-energizes when the pointer is zeroed or changed to reference a Holding Register different from the TABLE END.

OPERATION

When the ENABLE input to the function block is conducting, the port will accept inbound data. Data format is not restricted to ASCII. Data received is stored temporarily in a 520 byte buffer. Each byte is compared to the End-Of-Message (EOM) character, if that option is selected. Each time the ladder scan reaches the AR function the data in temporary storage is placed in the Holding Register table and the pointer is updated accordingly. The bytes are stored in the HR table as defined by Operand 3. When the HR table is full, the AR function is will accept up to 520 additional bytes before any data is lost.

When the ENABLE line is not conducting or when the PC is not in the Run Mode, any incoming data will be treated as Program Loader communications. If an ASCII Receive special function is not enabled, then data is treated as Program Loader information.

The ASCII Receive function will be disabled (the DTR line of the port will be set false) if the temporary buffer is full.

To re-initialize communications after an EOM character has been encountered, zero the pointer.

Once the temporary buffer is full, the AR function will attempt to transfer the data to the Holding Register table and resume communications. However, if the table is full, the pointer must either be zeroed or changed to reference a HR different than the Table End to continue operation.

Note

The "Data Set Ready" (DTR) line for the port must be held high for proper operation.

TABLE 1. AR TRUTH TABLE

Enable	Result
0	Port use reverts to Program Loader communications.
↑[1]	Table pointer, coil status, status byte and errors are reset. Data is transferred from Port into the table of Holding Registers.
1	Port continues to receive data (if the function was enabled) until either an EOM character is encountered or the table is full.
[1] Transition from OFF to ON.	

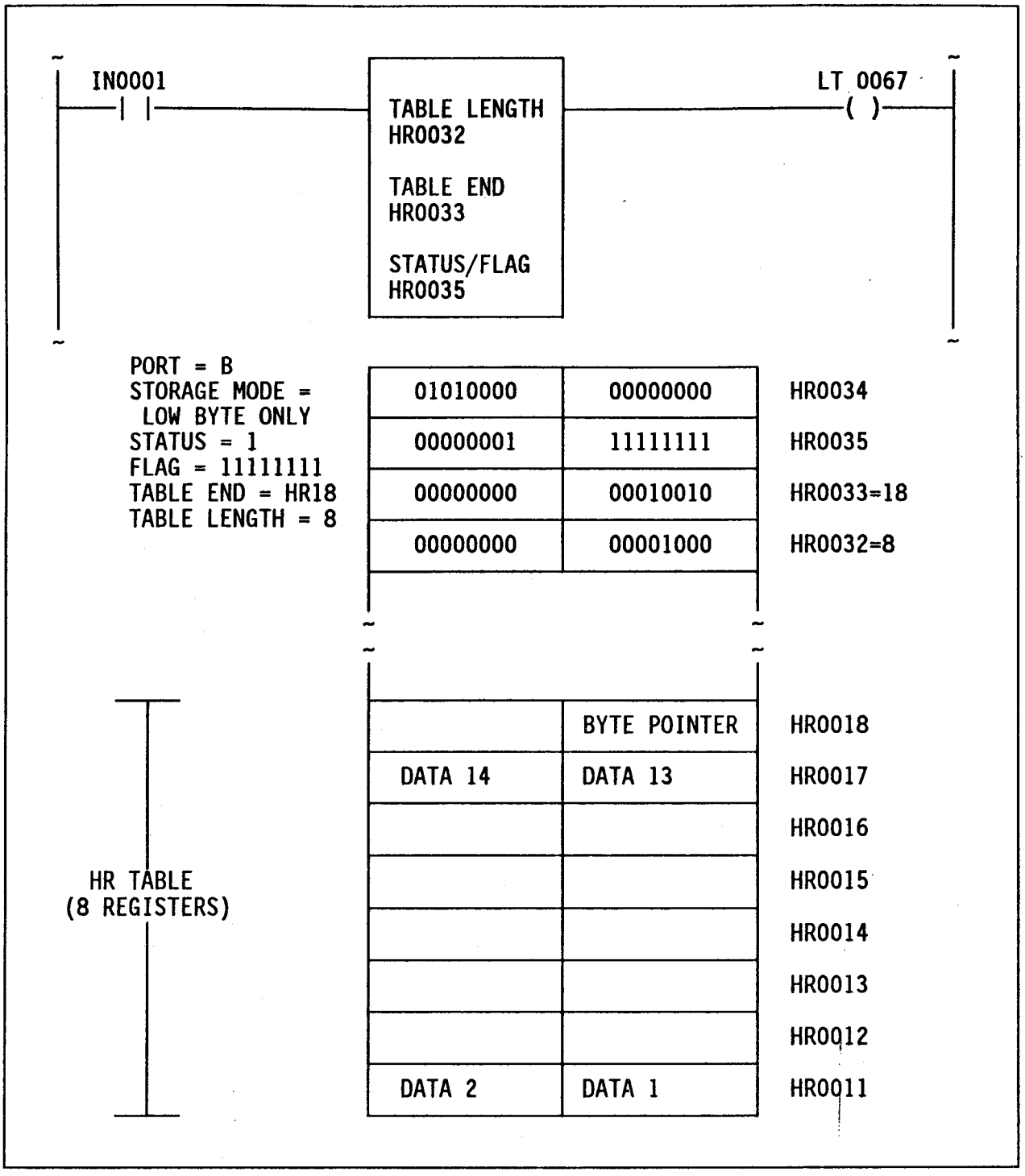


Figure 4. AR Example

PC-1100-x01y: NOT SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The Ascending Sort (AS) function arranges the data from a table of registers into ascending order (smallest-to-largest). AS function symbology is shown in Figure 1.

Data from a companion table is paired with data in the same relative position in the table being sorted. (See Figure 2.) In a PC-1100, a total of 16 extra holding registers is required for the function to operate correctly. These registers are located in the 16 registers following Table 1. In a PC1200, these registers are not used, however, the space must be allocated. Refer to Operand 2. The holding registers in Figure 2 are arranged as shown in Figure 3.

OP CODE

Op Code 58 defines the Literal (LT) as the AS function. Whether or not the Literal function should be used depends upon the capability of your program loader. It is recommended that the mnemonic function be used whenever possible. Refer to the Introduction in this Section and the LT function description.

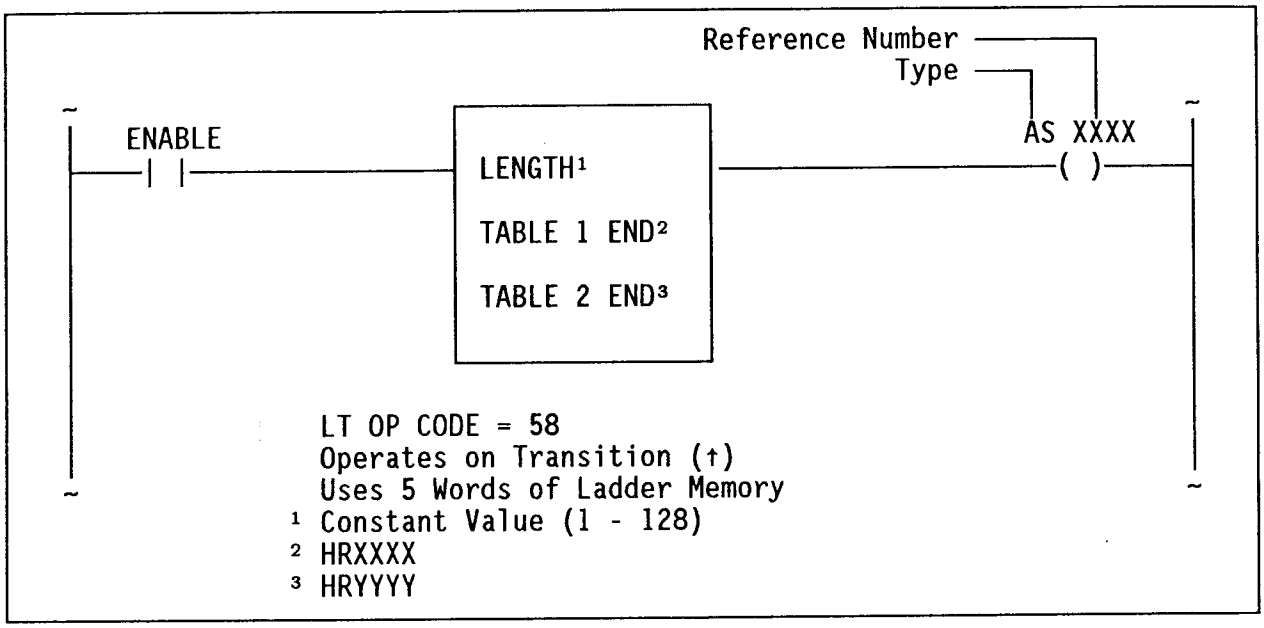


Figure 1. Ascending Sort (AS)

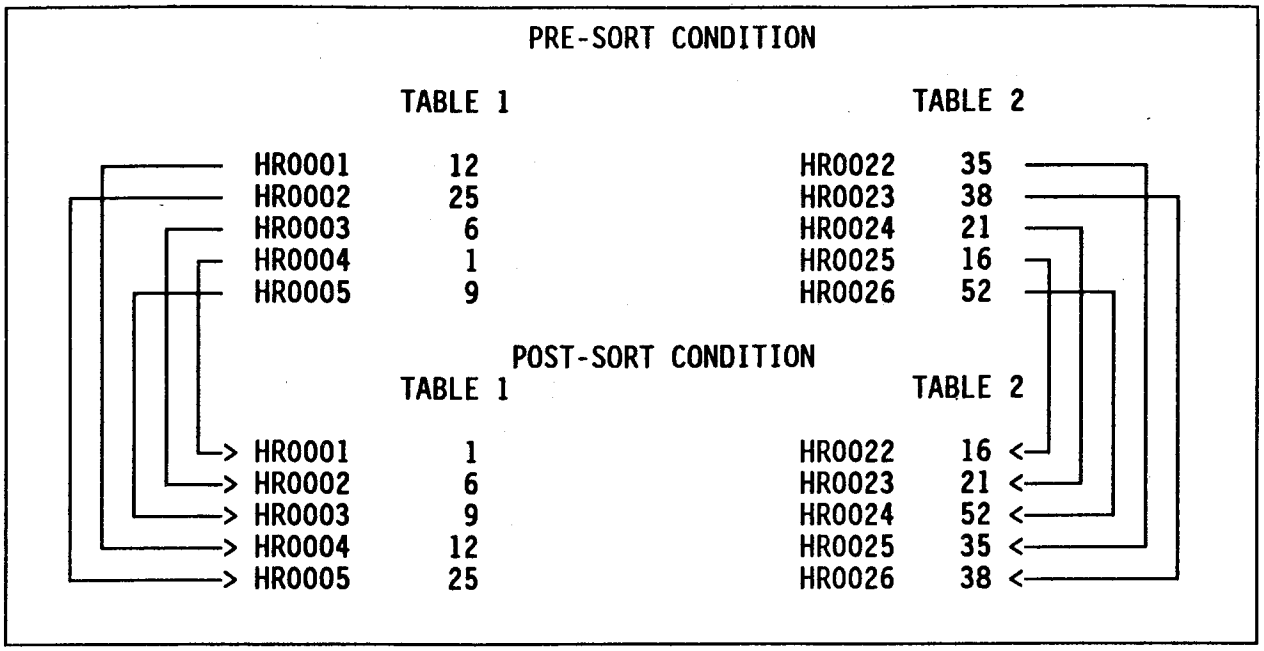


Figure 2. Sorted Holding Registers

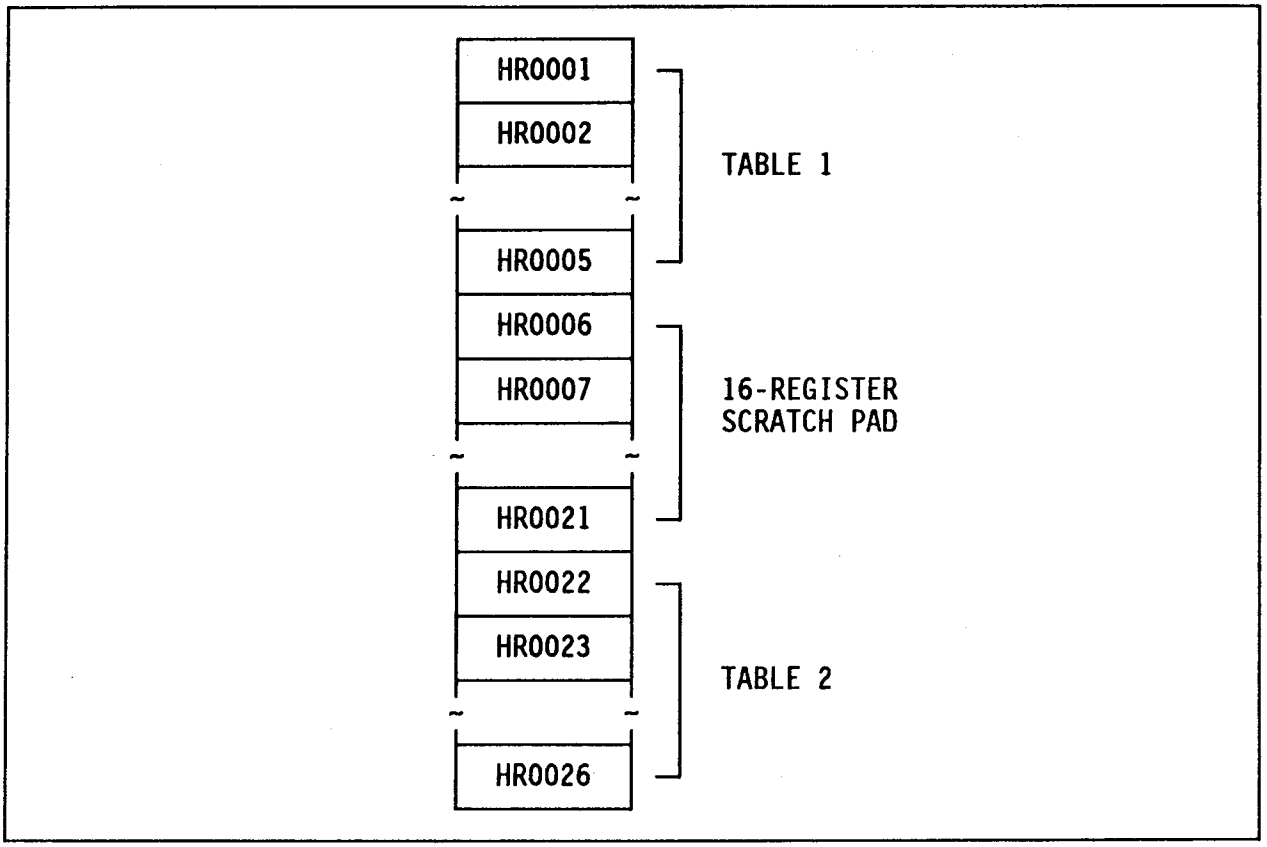


Figure 3. Holding Registers Arrangement

SPECIFICATIONS

OPERAND 1 - LENGTH

The length is a constant value in the range of 1 through 128 that defines the number of registers in use, excluding the extra registers.

OPERAND 2 - TABLE 1 END

Table 1 End specifies the last holding register in Table 1 to be sorted, plus 16 holding registers for "scratch pad" use. In the PC-1200, these holding registers are not used, but are required by the function.

OPERAND 3 - TABLE 2 END

Table 2 End specifies the last holding register in Table 2. If Operand 3 equals operand 2, only Table 1 will be sorted.

COIL

When enable is set to zero, the coil is de-energized and no sorting occurs. When enable is set to one, the AS function occurs. The coil is energized when sorting is completed.

APPLICATION

The ascending sort function (AS) is used to assemble data in an ascending order in one table and maintain a tag in the associated position in a second table. In this example, assume parts are randomly placed in 10 bins with a size and bin number entered manually in holding registers. Additions could be made to allow using thumbwheel switches and a enter pushbutton. It is desired to select and display the bin number and size of a part that most closely matches (equals or exceeds) value read by a gauge. It is assumed that all register contents are in decimal (binary form - any BCD to binary conversions required have been made before loading in assigned registers). The operator sets bin number (1 through 10) in HR0097 and size in HR0098. When input IN0001 is turned on, size is loaded into the selected location of table HR0121 through HR0130. Registers HR0101 through HR0110 contain respective bin numbers 1 through 10. To select a part gauge, size is entered in HR0116 and IN0002 is turned on. The tables are sorted and the TO function selects the size and bin number displaying them in HR0118 and HR0117. Note that the IM and TR functions have one extra position because they start with a zero position (not used) and the TO and TL functions start with one.

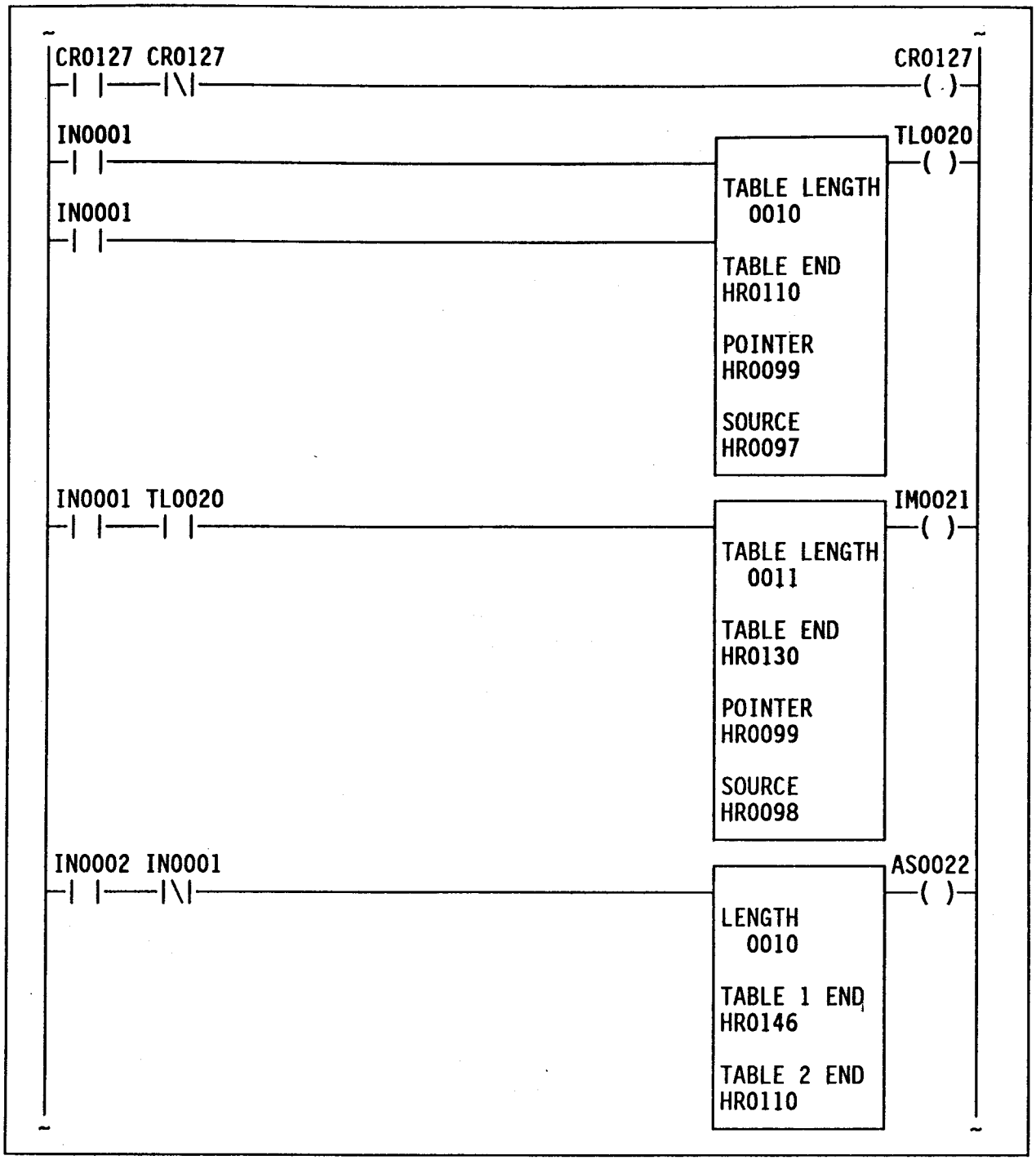


Figure 4a. AS Sorting Applicaton

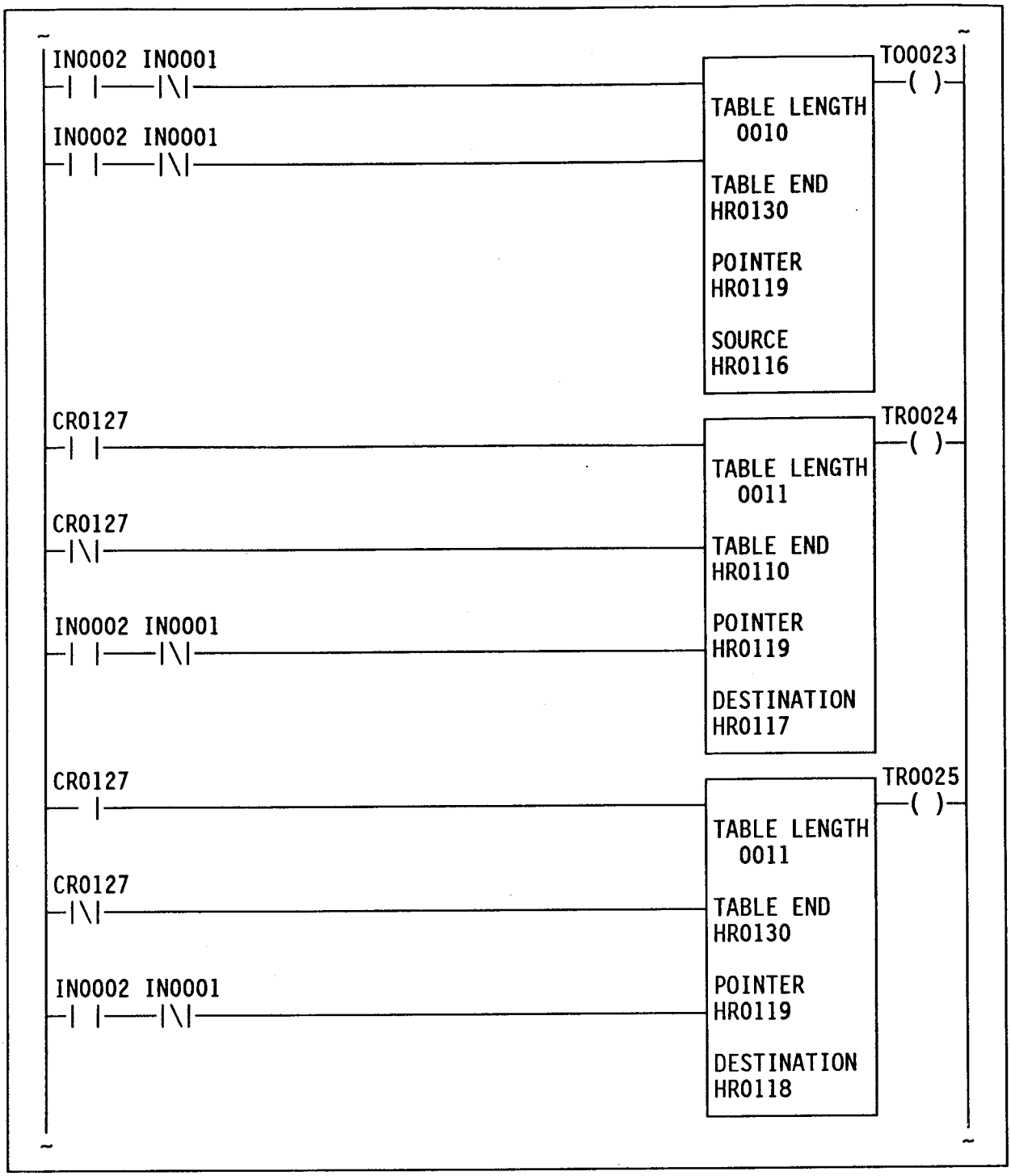


Figure 4b. AS Sorting Applicaton

AT - ASCII TRANSMIT

WORKS WITH ER

Modified for PC-1200

PC-1100-x01y: NOT SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The ASCII Transmit (AT) function allows American National Standard Code for information Interchange (ASCII) coded alphanumeric messages to be processed. These messages are transmitted from the processor directly to a printer or other suitable RS-232C compatible device that is capable of translating ASCII codes. If the processor is equipped with two ports, the function permits port selection. ASCII function symbology is shown in Figure 1.

The processor stores and works with all data in a binary format while the program loader can display or enter this data in hexadecimal (HEX), decimal, binary, or ASCII form. Table 1 defines the translation between these different representations of the ASCII character and control data set.

The ASCII message is stored in holding registers to be transmitted when this function is enabled. The ASCII characters are stored two per holding register. The upper byte (i.e., upper half of the holding register) is transmitted first. See example in Table 3.

Message information is stored in the format shown in Figure 2. The processor sends a maximum of ~~six~~ 60 ASCII characters per scan. If a special operation (B) code is encountered, the processor stops transmitting the ASCII characters in the present scan, even if less than six ASCII characters have been sent. The B

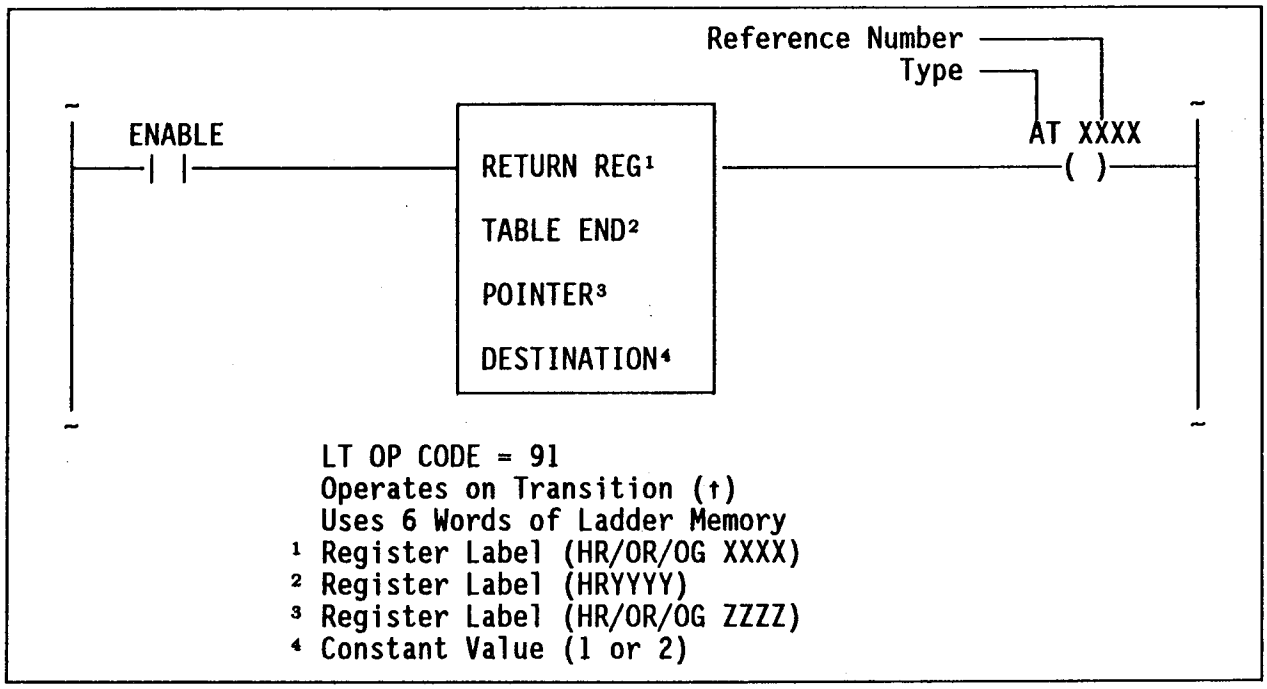


Figure 1. ASCII Transmit (AT)

code is implemented on the next scan and must be stored in the high byte. The length of the message is limited only by the number of registers available in the processor.

Table 2 defines a set of "B-code" special instruction codes unique to the Westinghouse AT special function. The B-codes must be entered and viewed in hexadecimal and are so named because they always start with the hexadecimal digit B. Their purpose is to either change the flow of ASCII transmission by jumping to an alternate message segment or to translate binary data to ASCII.

TABLE 1. ASCII CODE CONVERSIONS

Hexadecimal	Decimal	Octal	Binary	Character	Description
00	0	000	0000000	NUL	Null
01	1	001	0000001	SOH	Start of heading
02	2	002	0000010	STX	Start of text
03	3	003	0000011	ETX	End of text
04	4	004	0000100	EOT	End of transmission
05	5	005	0000101	ENQ	Enquiry
06	6	006	0000110	ACK	Acknowledge
07	7	007	0000111	BEL	Bell
08	8	010	0001000	BS	Back space
09	9	011	0001001	HT	Horizontal tab
0A	10	012	0001010	LF	Line feed
0B	11	013	0001011	VT	Vertical tab
0C	12	014	0001100	FF	Form feed
0D	13	015	0001101	CR	Carriage return
0E	14	016	0001110	SO	Shift out
0F	15	017	0001111	SI	Shift in
10	16	020	0010000	DLE	Data link escape
11	17	021	0010001	DC1	Device Control 1
12	18	022	0010010	DC2	Device Control 2
13	19	023	0010011	DC3	Device Control 3
14	20	024	0010100	DC4	Device Control 4
15	21	025	0010101	NAK	Negative acknowledge
16	22	026	0010110	SYN	Synchronize
17	23	027	0010111	ETB	End of transmission block
18	24	030	0011000	CAN	Cancel
19	25	031	0011001	EM	End of media
1A	26	032	0011010	SUB	Substitute
1B	27	033	0011011	ESC	Escape
1C	28	034	0011100	FS	File separator
1D	29	035	0011101	GS	Group separator
1E	30	036	0011110	RS	Record separator
1F	31	037	0011111	US	Unit separator
20	32	040	0100000	SP	Space
21	33	041	0100001	!	Exclamation
22	34	042	0100010	"	Double quote
23	35	043	0100011	#	Number or pound
24	36	044	0100100	$	Dollar sign
25	37	045	0100101	%	Percentage
26	38	046	0100110	&	Ampersand

TABLE 1. ASCII CODE CONVERSIONS (Cont'd)

Hexadecimal	Decimal	Octal	Binary	Character	Description
27	39	047	0100111	'	Apostrophe or single quote
28	40	050	0101000	(	Left parenthesis
29	41	051	0101001	)	Right parenthesis
2A	42	052	0101010	*	Asterisk
2B	43	053	0101011	+	Plus
2C	44	054	0101100	,	Comma
2D	45	055	0101101	-	Minus
2E	46	056	0101110	.	Period
2F	47	057	0101111	/	Slash
30	48	060	0110000	0	Zero
31	49	061	0110001	1	One
32	50	062	0110010	2	Two
33	51	063	0110011	3	Three
34	52	064	0110100	4	Four
35	53	065	0110101	5	Five
36	54	066	0110110	6	Six
37	55	067	0110111	7	Seven
38	56	070	0111000	8	Eight
39	57	071	0111001	9	Nine
3A	58	072	0111010	:	Colon
3B	59	073	0111011	;	Semi-colon
3C	60	074	0111100	<	Less than
3D	61	075	0111101	=	Equal
3E	62	076	0111110	>	Greater than
3F	63	077	0111111	?	Question
40	64	100	1000000	@	At sign
41	65	101	1000001	A	Letter A
42	66	102	1000010	B	Letter B
43	67	103	1000011	C	Letter C
44	68	104	1000100	D	Letter D
45	69	105	1000101	E	Letter E
46	70	106	1000110	F	Letter F
47	71	107	1000111	G	Letter G
48	72	110	1001000	H	Letter H
49	73	111	1001001	I	Letter I
4A	74	112	1001010	J	Letter J
4B	75	113	1001011	K	Letter K
4C	76	114	1001100	L	Letter L
4D	77	115	1001101	M	Letter M
4E	78	116	1001110	N	Letter N
4F	79	117	1001111	O	Letter O
50	80	120	1010000	P	Letter P
51	81	121	1010001	Q	Letter Q
52	82	122	1010010	R	Letter R
53	83	123	1010011	S	Letter S
54	84	124	1010100	T	Letter T
55	85	125	1010101	U	Letter U

TABLE 1. ASCII CODE CONVERSIONS (Cont'd)

Hexadecimal	Decimal	Octal	Binary	Character	Description
56	86	126	1010110	V	Letter V
57	87	127	1010111	W	Letter W
58	88	120	1011000	X	Letter X
59	89	131	1011001	Y	Letter Y
5A	90	132	1011010	Z	Letter Z
5B	91	133	1011011	[	Left bracket
5C	92	134	1011100	\	Back slash
5D	93	135	1011101	]	Right bracket
5E	94	136	1011110	↑	Up arrow
5F	95	137	1011111	←	Back arrow
60	96	140	1100000	`	Back quote or accent mark
61	97	141	1100001	a	Small letter a
62	98	142	1100010	b	Small letter b
63	99	143	1100011	c	Small letter c
64	100	144	1100100	d	Small letter d
65	101	145	1100101	e	Small letter e
66	102	146	1100110	f	Small letter f
67	103	147	1100111	g	Small letter g
68	104	150	1101000	h	Small letter h
69	105	151	1101001	i	Small letter i
6A	106	152	1101010	j	Small letter j
6B	107	153	1101011	k	Small letter k
6C	108	154	1101100	l	Small letter l
6D	109	155	1101101	m	Small letter m
6E	110	156	1101110	n	Small letter n
6F	111	157	1101111	o	Small letter o
70	112	160	1110000	p	Small letter p
71	113	161	1110001	q	Small letter q
72	114	162	1110010	r	Small letter r
73	115	163	1110011	s	Small letter s
74	116	164	1110100	t	Small letter t
75	117	165	1110101	u	Small letter u
76	118	166	1110110	v	Small letter v
77	119	167	1110111	w	Small letter w
78	120	170	1111000	x	Small letter x
79	121	171	1111001	y	Small letter y
7A	122	172	1111010	z	Small letter z
7B	123	173	1111011	{	Left brace
7C	124	174	1111100	|	Vertical bar
7D	125	175	1111101	}	Right brace
7E	126	176	1111110	~	Approximate or tilde
7F	127	177	1111111	DEL	Delete (rub out)

TABLE 2. SPECIAL OPERATIONS CODES

Code	Meaning
B000	Return to the holding register specified by the contents of the return register. If the contents equal zero, B0 is ignored (i.e. no operation).
B1XX 00YY	The ASCII character in the low byte of the B1 register is to be repeated YY times (where YY < 256 and is stored in the following register).
B200 (B903) [1] PPPP	The next holding register is a pointer (PPPP) to another holding register whose content is a number. The number is to be converted into a five-digit Binary-Coded-Decimal (BCD) number. The lower two digits are converted into ASCII codes and sent to the proper communications port.
B300 (B99F) [1] PPPP	The next holding register is a pointer (PPPP) to another holding register whose content is a number. The number is to be converted into a five-digit BCD number between -32K and +32K, depending on the most-significant bit. The sign and the five digits are converted into ASCII code and sent to the proper communications port.
B400 (B90F) [1] PPPP	The next holding register is a pointer (PPPP) to another holding register whose content is a number. The number is to be converted into a five-digit BCD number, and the lower four digits are converted into ASCII codes and sent to the proper communications port.
B500 (B91F) [1] PPPP	The next holding register is a pointer (PPPP) to another holding register whose content is a number. The number is to be converted into a five-digit BCD number and then converted into ASCII codes and sent to the proper communications port.
B600 (B95F) [1] PPPP	The next holding register is a pointer (PPPP) to another holding register whose content is a number. The number is to be converted into a five-digit BCD number. Leading zeroes of this number are blanked; the remaining digits are converted to ASCII codes and sent to the proper communications port.
B700 NNNN	Jump to the holding register whose reference number is specified in the following register. Add two to the current pointer value and store it in the return register.
B800	Message End. Reset the return register to zero.

[1] B2 through B6 are not available in Advanced II software (3.0 - 3.xx). Instead, substitute this code to achieve the function described. B2 through B6 not available in Advanced PC-1100s (PC1100-x033).

TABLE 2. SPECIAL OPERATION CODES (Cont'd)

Code	Meaning
B9XX PPPP	The next holding register is a pointer to another holding register. The contents of this register is a decimal value ≤ 65535. The five digits are converted to ASCII code and transmitted as defined by the low byte of the B9XX code. The five LSB (00 - 1F Hex) of the B9 code forms a mask with the five ASCII digits. Only those ASCII digits which correspond to a "1" in the mask will be transmitted. A "0" in the mask inhibits transmission of the ASCII digit. example: • • • HR11 B91C (1C serves as the mask) HT12 23 (Pointer) • • • HR23 12345 The binary equivalent of the mask 1C is 11100. The holding register pointed to is HR23=12345 The corresponding ASCII digits transmitted are 123. If HR11=B903, the ASCII digits transmitted in HR23 are 45. If HR11=B915, the ASCII digits transmitted in HR23 are 135. The three MSB of the low byte of the B9 code are defined as follows: Bit 8 - When set to a "1" state, the sign (as defined by Bit 16) is converted to ASCII code and transmitted. Bit 7 - When set to a "1" state, suppresses leading zeroes. Bit 6 - When set to a "1" state, suppresses leading spaces. The ASCII Transmit function continues with the HR following the Pointer Register.

TABLE 2. SPECIAL OPERATION CODES (Cont'd)

Code	Meaning
BA0Y	The low byte of the BA codes specifies the following: BA00 - The high byte will be transmitted first, followed by the low byte. BA03 - The low byte only is transmitted. BA02 - The high byte only is transmitted. BA01 - The low byte will be transmitted first, followed by the high byte. The register following the BA Code contains the number of Registers to be transmitted in the low byte, and the pointer in the high byte. High Byte — Low Byte Internal Pointer — # Registers Transmitted 01 — 1 Register Transmitted • • • FF — 255 Registers Transmitted 00 — 256 Registers Transmitted Beginning with the second register from the BA Code, transfer the binary bit pattern contained in the next 1-256 registers. When complete, the AT function resumes execution with the very next register following the binary bit pattern table of registers

TABLE 2. SPECIAL OPERATION CODES (Cont'd)

Code	Meaning
BAXX 00YY	High Byte is Internal Pointer YY defines number of registers to be. transmitted (00-FF). BAXX 00YY • • • 1-256 Registers (Binary Bit Pattern Table) • • • B800 XX-specifies order of transmission 00-High Byte Low Byte 01-Low Byte Only 02-High Byte Only 03-Low Byte High Byte Continuation of ASCII Transmit Message
BB00 PPPP	Jump to the HR whose reference number is specified in the following register (No adjustment is made to the Return Register.)

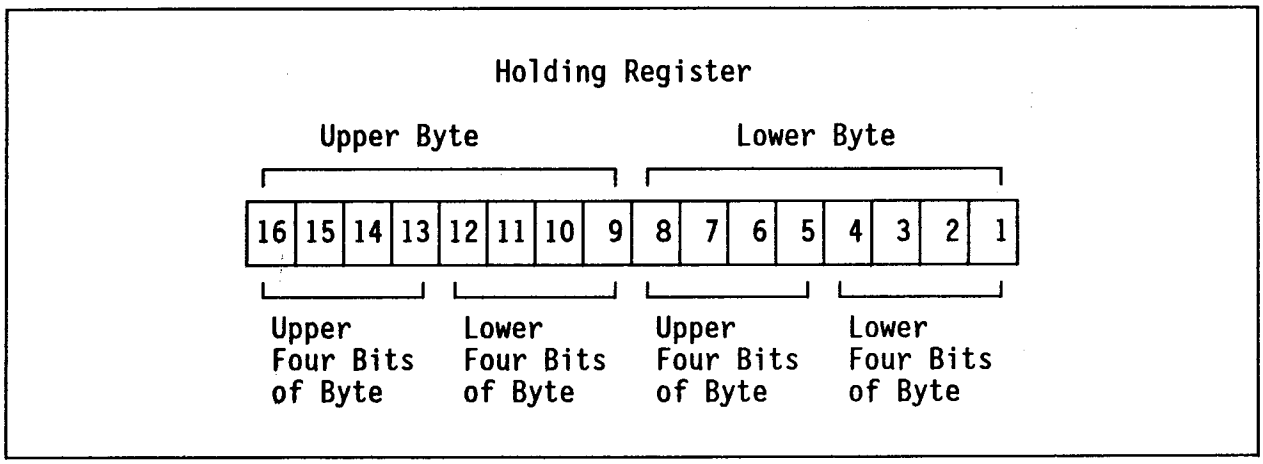


Figure 2. Character Storage Format

OP CODE

Op Code 91 defines the Literal (LT) as the AT function. Whether or not the Literal function should be used depends upon the capability of your program loader. It is recommended that the mnemonic function be used whenever possible. Refer to the Introduction in this Section and the LT function description.

SPECIFICATIONS

OPERAND 1 - RETURN REGISTER

The return register defines the location of the holding register reference number that specifies the location for return from a "B-code" subroutine. This location must be programmed, even if it is not used. It may be defined as a:

- Holding Register (HR)
- Output Register (OR)
- Output Group (OG)

In the PC-1200, if any error occurs in the execution of this function, the return register will be changed to the maximum value (e.g., 65535, FFFFH, etc.). An error will occur if the function is programmed improperly, if an illegal B-code is used, or if a non-ASCII character other than a B-Code is detected in the message.

OPERAND 2 - TABLE END

The table end is a dummy register that defines the highest holding register reference number containing message data. This is always a holding register. This is used to flag program loader to highest holding register used for program recording purposes and ladder programming.

OPERAND 3 - POINTER

The pointer points to the holding register that is being sent. The pointer may be a:

- Holding Register (HR)
- Output Register (OR)
- Output Group (OG)

OPERAND 4 - DESTINATION

The destination is a constant that specifies which port the processor will use to transmit message data. The destination for the PC-1100/PC-1200 with two ports must be set as follows:

1 = Port A
2 = Port B

COIL

When enable is set to zero, the coil is de-energized and no message is transmitted. When enable is set to one, the coil is energized and the message is transmitted.

APPLICATION EXAMPLES

Example 1

Any desired messages can be formatted and transmitted to suitable RS-232C compatible devices. Table 3 shows the message STATION 15 IS INACTIVE coded hexadecimal. The programming for the STATION 15 IS INACTIVE message is shown in Figure 3.

TABLE 3. ASCII MESSAGE TABLE

Holding Register	Hexadecimal Code		Message	
	Upper Byte	Lower Byte	Upper Byte	Lower Byte
HR0200	53	54	S	T
HR0201	41	54	A	T
HR0202	49	4F	I	O
HR0203	4E	20	N	(space)
HR0204	31	35	1	5
HR0205	20	49	(space)	I
HR0206	53	20	S	(space)
HR0207	49	4E	I	N
HR0208	41	43	A	C
HR0209	54	49	T	I
HR0210	56	45	V	E
HR0211	0A	0D	(CR)	(LF)[1]
HR0212	B8	00	End of Message	

[1] Carriage Return, Line Feed: primarily used with printers and dumb terminals. Not necessary in all ASCII transmissions. See Application Note 2.

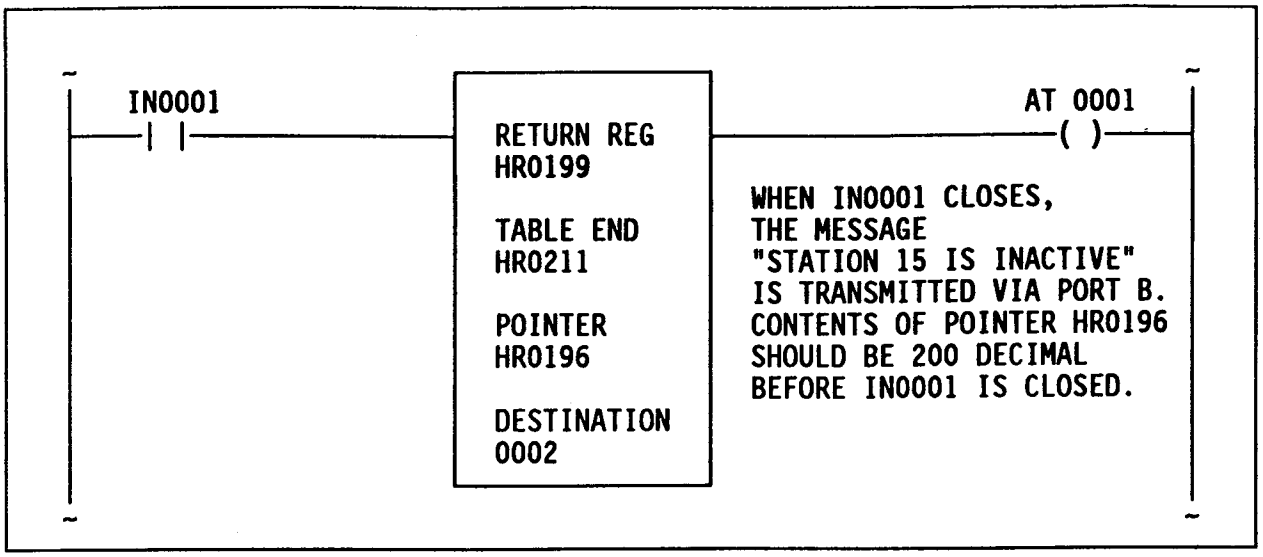


Figure 3. AT Example 1 - STATION 15 IS INACTIVE Message

Example 2

The AT function can be used to cause a graph of a function to be printed. (See Figure 4). If the correct control signals are inserted in the ASCII Message Table for the printer in use, the program will print a bar of X's that are proportioned in length to the input to the Analog-to-Digital (A-D) converter.

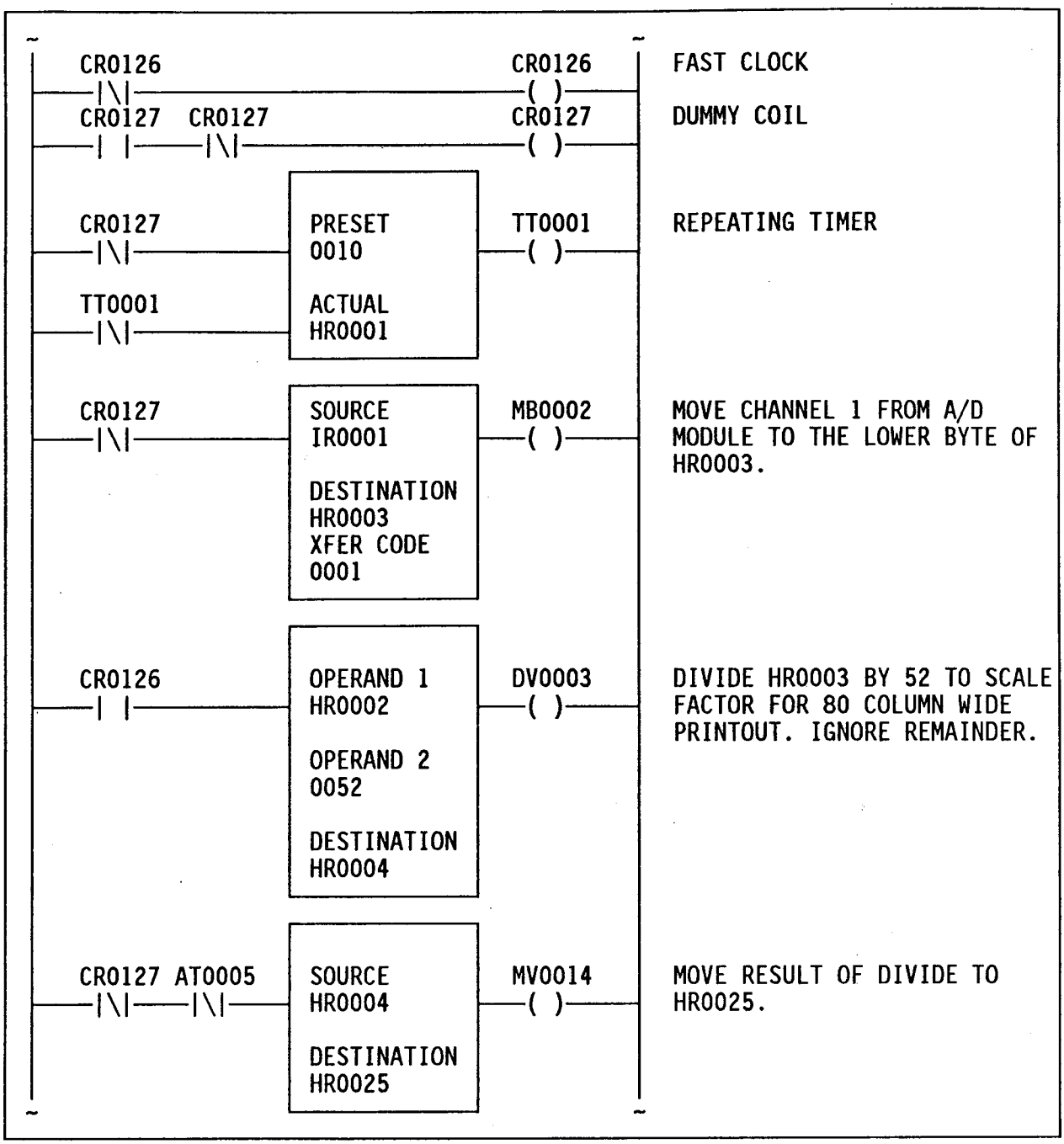


Figure 4a. AT Example 2 - Bar Graph

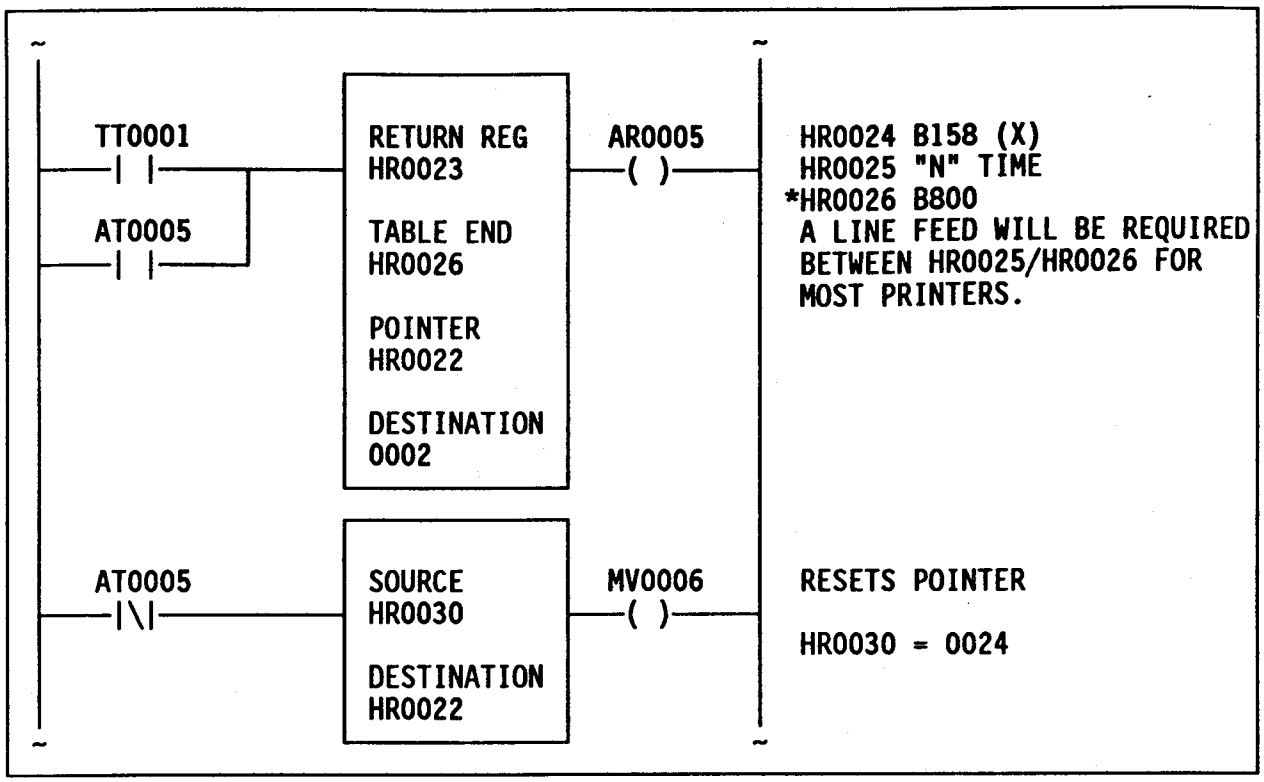


Figure 4b. AT Example 2 - Bar Graph

Example 3

This is a modification of Example 1. It adds B-codes for more detail. Both the submessage and binary-to-ASCII conversions are made. Note the use of the pointer reset circuit.

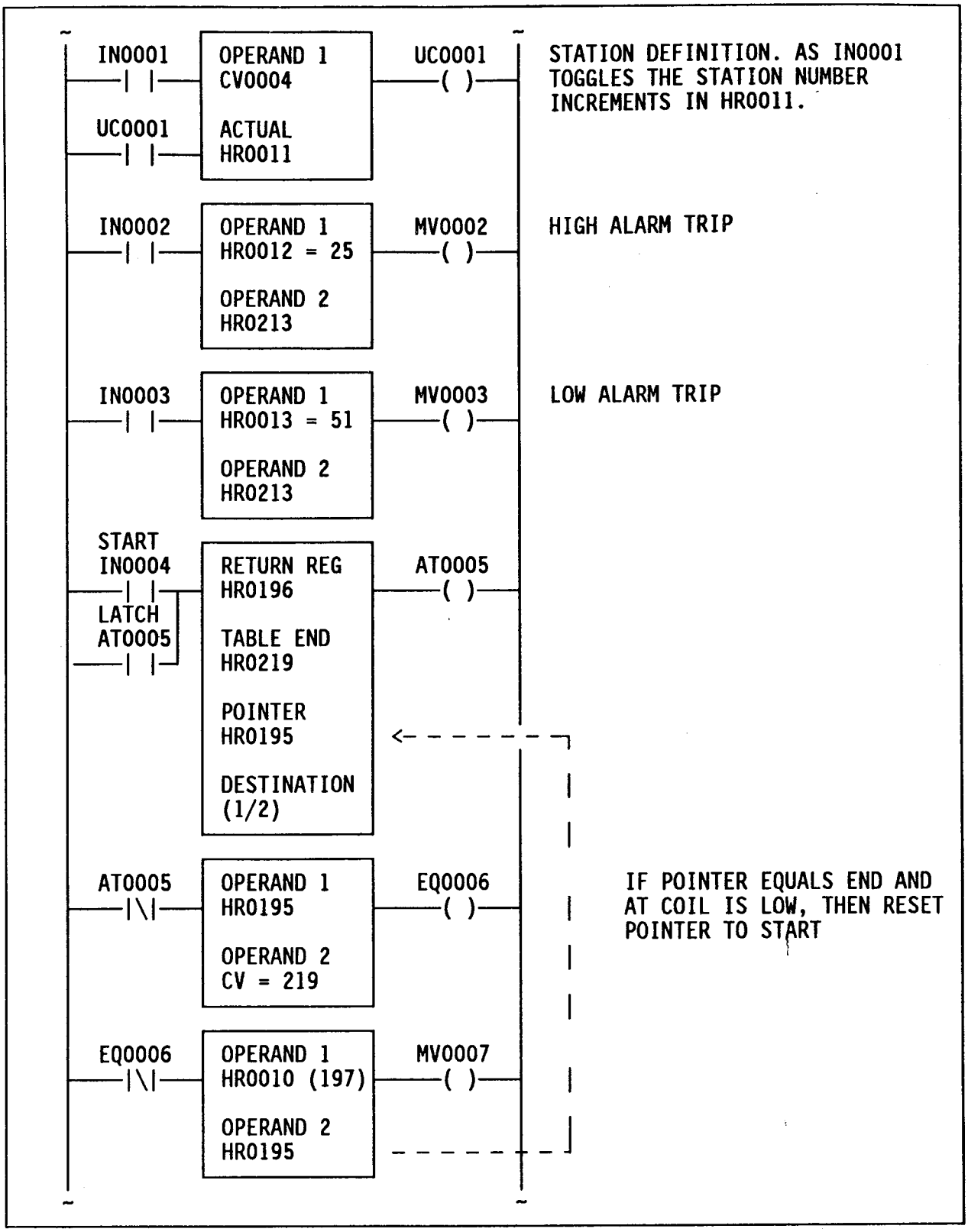


Figure 5a. Example 3 - B-Code Use

Holding Reg. Address	Hexadecimal Code		ASCII Character	
	High byte	Low byte	High Byte	Low Byte
HR0197	B1	2A	(B-code = repeat "*"	
HR0198	00	20	32 (decimal) times)	
HR0199	0D	0A	(CR)	(LF)
0200	53	54	S	T
0201	41	54	A	T
0202	49	4F	I	O
0203	4E	20	N	(SP)
0204	23	20	#	(SP)
0205	B4	00	(B-code)	
0206	(00011 DEC.)		(B Pointer)	
0207	20	49	(SP)	I
0208	53	20	S	(SP)
0209	49	4E	I	N
0210	B7	00	(B-code)	
0211	(00025/51 DEC.)		(B Pointers)	
> 0212	20	41	(SP)	A
0213	76	41	L	A
0214	82	66	R	M
0215	0D	0A	(CR)	(LF)
0216	B1	2A	(B-Code = repeat "*"	
HR0217	00	20	32 (decimal) times)	
HR0218	0D	0A	(CR)	(LF)
HR0219	B8	00	(B-Code = end)	
> 0025	20	72	(SP)	H
0026	73	71	I	G
0027	72	20	H	(SP)
0028	B0	00	(B-code = return)	
> 0051	20	76	(SP)	L
0052	79	87	O	W
0053	B0	00	(B-code = return)	

Result: When Up Counter (UC) equals 3 and IN0002 is TRUE, the AT message will be:

****
STATION # 3 IS IN HIGH ALARM
****

Figure 5b. Example 3 - B-Code Use

APPLICATION NOTES

A description of serial port and a definition of control lines can be found in Section 3 of this manual.

1. A binary number and the ASCII binary equivalent do not have the same binary representation in memory. For example:

   Decimal "2" = 0000 0010 in binary, which is processed by the computer.

   B-codes B2 through B6 translate this into binary which is used by a printer. The result is:

   Character "2" = 0011 0010.

2. The AT special function uses the serial port handshake lines. The receiving device can drive the processor's CTS line for buffer control or RTS and CTS can be strapped at the processor if the receiving device's buffer is large enough to handle the message. Failure to satisfy CTS will halt ASCII transmission.

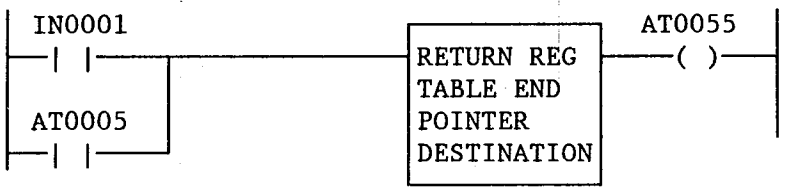
3. The pointer must be set and reset by the user's program. Example 3 shows a ladder mechanism which will reset the AT function when the pointer equals "end of message".

4. Most printers require a line feed, carriage return or a full line of characters (80 or 132) to be received before the line will be printed.

5. The use of B-codes, especially B7 and B1, can significantly reduce the use of holding registers by using a common message with variations spliced in. Refer to Example 2.

6. In order to prevent logic tearing, the pointers for B-codes should be set by the user's program before the AT function is engaged. The AT transmission may take several ladder scans and the processor does not snap-shot B-code pointers.

7. The use of a seal-up circuit on the AT enable circuit and referenced to the AT coil can be used to latch a momentary AT trip. See Table 4 and Figure 3.



8. NLSW 783U/784 APL software for programming PC-718/900/1100/1200 processors has an "offline utilities" feature which provides for ASCII entry including "B-Codes" in a word processor type format. Otherwise, registers can be monitored and changed by viewing registers in ASCII format. See the APL manual for more information.

TROUBLESHOOTING

Symptom	Possible Cause	Solution
No apparent output at printer. AT pointer advances a few, then stops.	1. Line buffer not satisfied 2. Cable wrong 3. Damaged communication ports	1. See Appliction Note 2. 2. See Section 3. 3. Beware of common mode voltage difference between peripheral and processor.
Unexpected or incorrect characters displayed on screen	1. Mismatch of baud rate or data frame between processor and peripheral	See Section 3 for DIP switch settings. See CP description. Set modes equal.
Characters drop out	1. Receiving device buffer overflow.	Establish handshakes with CTS of procssor if possible. Slow down baud rate (both devices) so buffer does not overflow.
Coil of AT drops out transmission and stops before B800 with FFFF in return register	1. Illegal ASCI character in message (above 7F)	Change to legal character
Other problems	1. Logic problem in program	Review examples. Refer to Application Notes.

PC-1100-x01y: SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The Binary to Decimal (BD) function converts binary bits to Binary-Coded-Decimal (BCD) digits; the Decimal to Binary (DB) function converts BCD digits to a binary number. BD/DB function symbology is shown in Figure 1.

BINARY TO BCD (BD)

The BD function changes up to 16 binary bits to four BCD digits (up to 9999).

The BD coil energizes when the conversion circuit conducts and the binary number exceeds 9999; the last valid number remains in the output register or group when the number exceeds 9999. The coil de-energizes when the convert circuit does not conduct. Forcing the BD coil affects only the associated contacts and any output circuit; conversions continue according to the convert circuit.

BCD TO BINARY (DB)

The DB function changes up to four BCD digits to a 16-bit binary number. The source of the BCD number and the destination of the binary result are the same registers and groups as in the BD Conversion function.

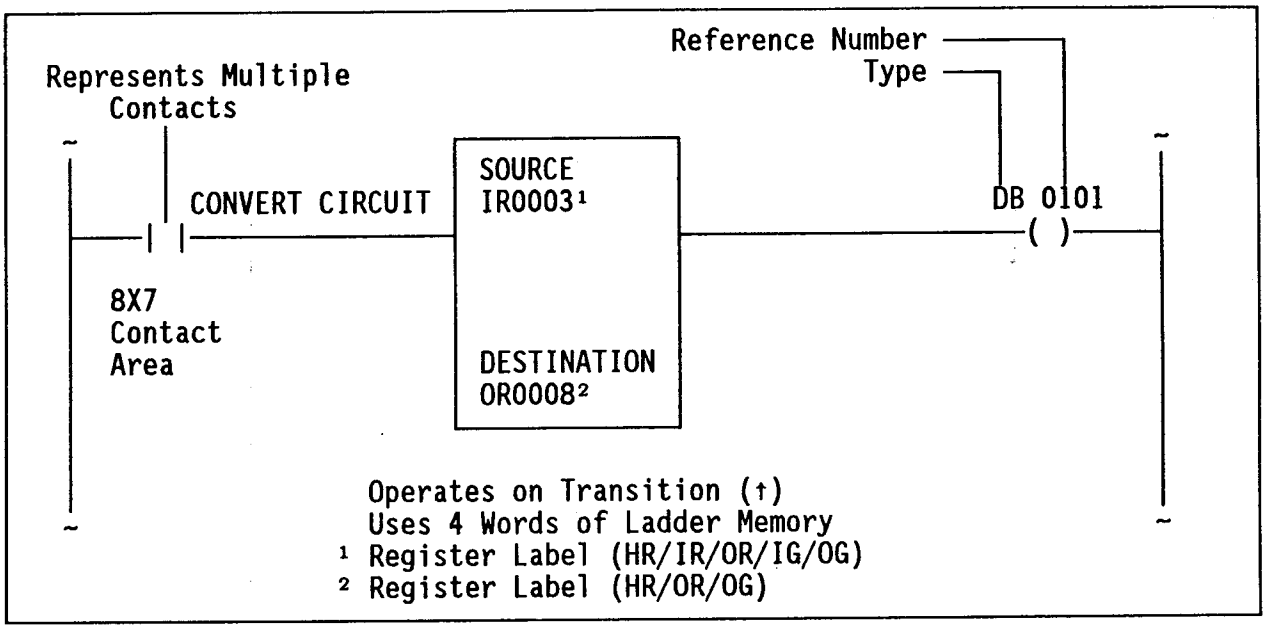


Figure 1. Binary to Decimal (BD)/Decimal to Binary (DB)

The DB coil energizes when the convert circuit conducts and the BCD number is invalid (any digit exceeding nine); the last valid number remains in the destination register or output group. The coil de-energizes when the convert circuit does not conduct; forcing the coil affects the contacts and conversions in the same manner as the BD Conversion function.

The registers for both the BD and DB functions are specified by the type and reference number; input and output groups are specified by the type and group number.

The conversion for the BD and DB functions is made when the convert circuit changes from non-conducting. The result of the conversion determines the state of the destination register or output group until a new result is produced.

SPECIFICATIONS

SOURCE

The source is the BCD or binary number to be converted. This value is held in a specified register or group:

- Holding Register (HR)
- Input Register (IR)
- Output Register (OR)
- Input Group (IG)
- Output Group (OG)

BD/DB TRUTH TABLE

See Table 1.

APPLICATIONS

Thumbwheel inputs and numerical displays are generally in BCD form. The processor operates in binary; therefore, incoming data is converted from BCD to binary, and outputs are converted from binary to BCD, as shown in Figure2.

TABLE 1. BD/DB TRUTH TABLE

Enable	Result
0	None. The coil de-energizes.
↑	Converts the value in the source register. The result is placed in the destination. In the BD function, the coil is energized when the convert circuit is conducting and the binary number is greater than 9999. In the DB function, the coil is energized when the convert circuit is conducting and the BCD number contains a digit greater than 9.
1	The coil remains in the last state.

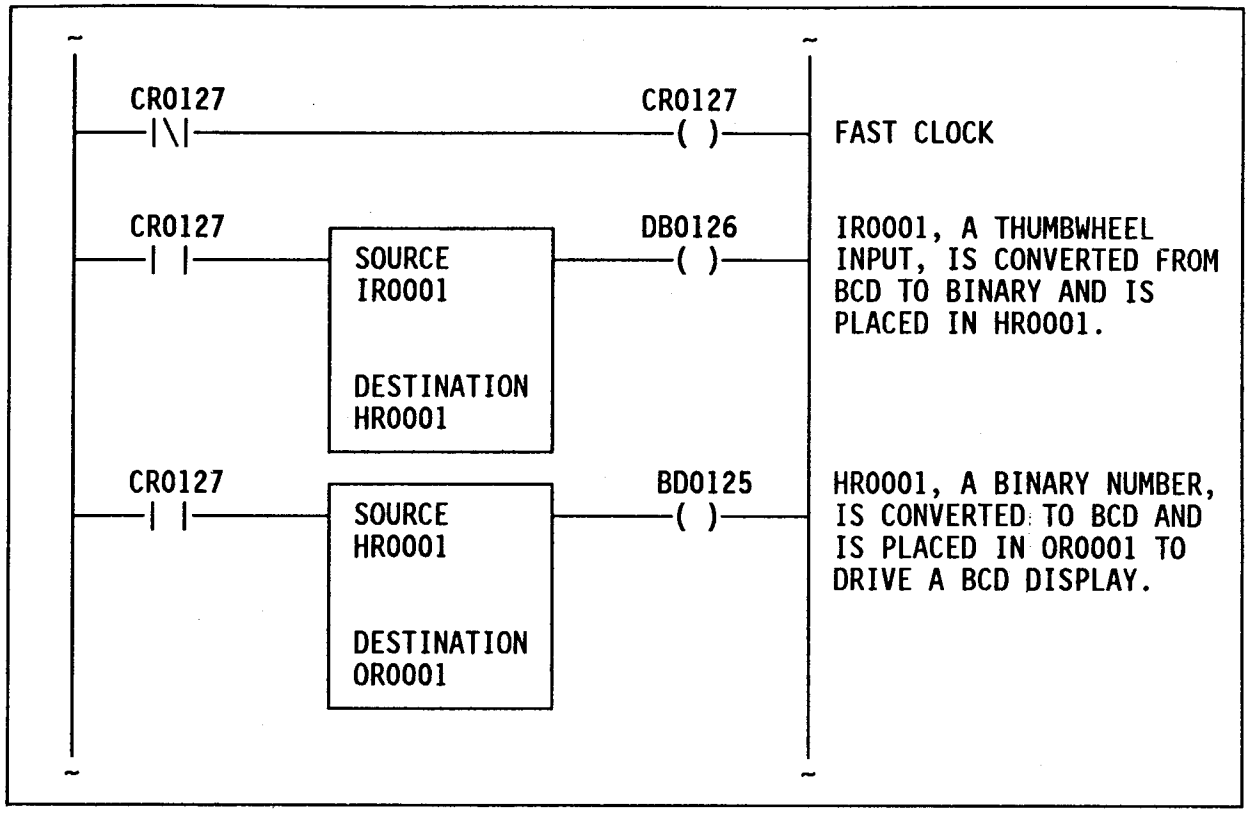


Figure 2. Number Conversions

BF - BIT FOLLOW

PC-1100-x01y: SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The Bit Follow (BF) function sets or clears a bit in either a holding or output register according to the status of a contact circuit. Activating a BF coil sets the designated bit to a logic 1. Disabling a BF coil clears the designated bit to a logic 0. BF function symbology is shown in Figure 1.

The BF function is similar to the Bit Set and Bit Clear functions, except that it does not latch. However, during a power loss, if all conditions remain unchanged, the BF function retains its state. The BF function cannot be forced.

SPECIFICATIONS

BIT NUMBER

The bit number specifies the bit controlled by the coil.

Note

Bit numbers beyond 16 (the number of bits in a single register) may be used. For example the bit cited in Figure 1 (BF0016, HR0003) could also be bit 48 of HR0001. This feature is limited to 2048 or the number of bits from the reference HR or OR to the highest available register, whichever is least.

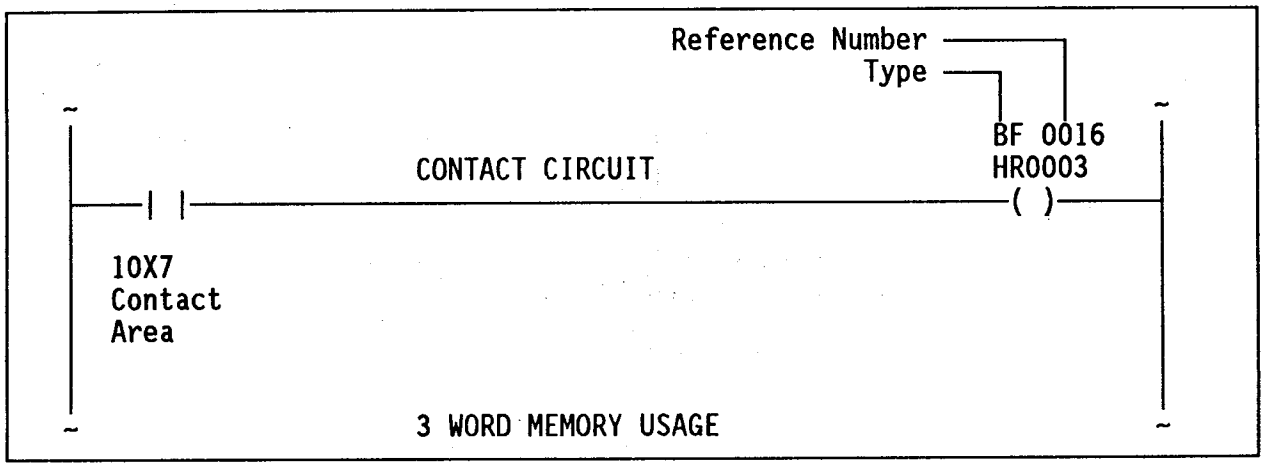


Figure 1. Bit Follow (BF)

DESIGNATED REGISTER

The designated register specifies the register in which the bit controlled by the coil is located. The register may be a Holding Register (HR) or an Output Register (OR).

BF TRUTH TABLE

See Table 1.

TABLE 1. BF TRUTH TABLE

Contact Circuit	Result
0 (Non-Conductive)	Specified bit is cleared: OFF.
1 (Conductive)	Specified bit is set to one: ON.

APPLICATIONS

The BF function can be used as a retentive relay. For example, using the program in Figure 2, if the processor loses power during normal operation, on power up, the BF coil will retain the state it was in prior to power loss. Thus, the motor will automatically restart when the power is restored.

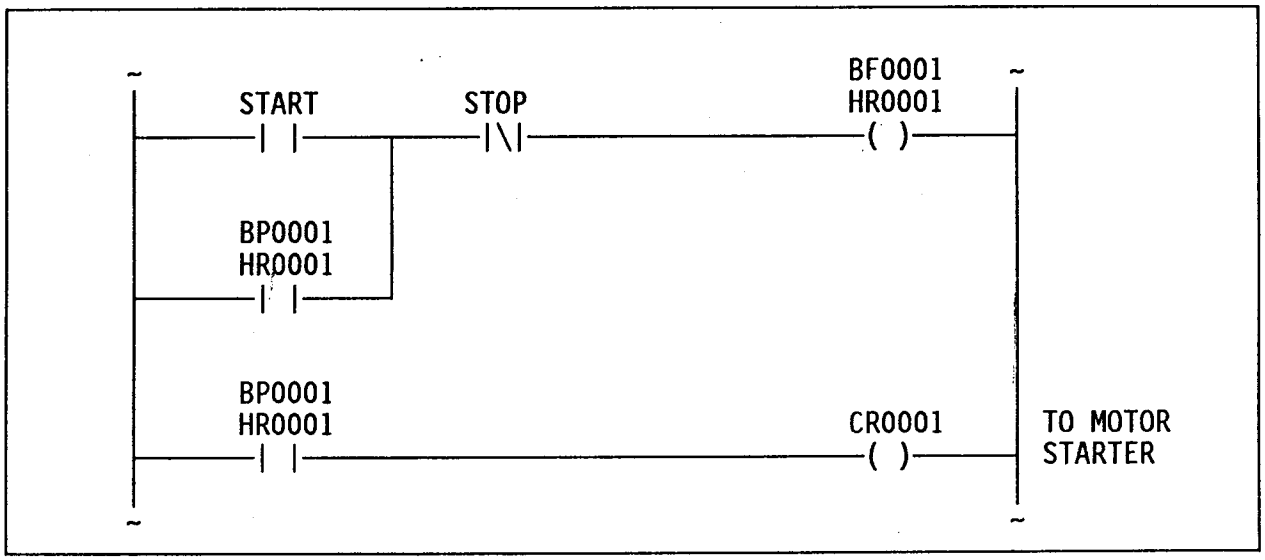


Figure 2. Retentive Relay

BO - BIT OPERATE

Modified for PC-1200

PC-1100-x01y: SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The Bit Operate (BO) function allows the user to selectively monitor or manipulate a designated bit in a predefined table or matrix. The value (up through 4096) in a pointer location specifies the bit. The state of the BO coil reflects the state of the bit. Bit manipulation is accomplished by using a set input and a reset input in accordance with Table 1. BO function symbology is shown in Figure 1.

OP CODE

Op Code 61 defines the Literal (LT) as a BO function. Whether or not the Literal function should be used depends upon the capability of your program loader. It is recommended that the mnemonic function be used whenever possible. Refer to the Introduction in this Section and the LT function description.

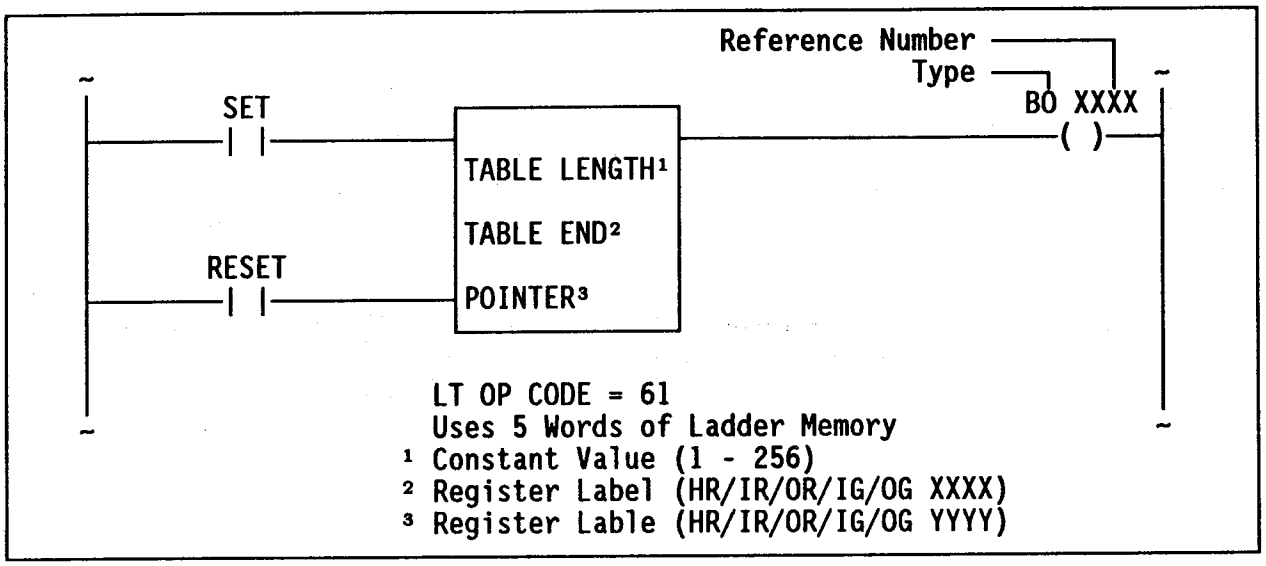


Figure 1. Bit Operate (BO)

TABLE 1. BO ACTION

Set	Reset	Bit Action/Coil State
OFF	OFF	Monitor mode - Coil follows the bit state.
ON	OFF	Set bit - Coil energizes.
OFF	ON	Reset bit - Coil de-energizes.
ON	ON	Reset bit - Coil de-energizes (reset overrides set).

SPECIFICATIONS

OPERAND 1 - TABLE LENGTH

Table length is a constant that defines the number of registers in the BO table. The range is 1 through 256, and is subject to the limitations listed in Table 2.

Note

The highest number holding register (1792) in Table 2 is limited by and dependent on memory size.

TABLE 2. TABLE LENGTH AND TABLE END LIMITS

Type	PC-1100	PC-1200-1020/1040	PC-1200-1041/1041/1042	PC-1250
HR	≤ 1792 [1]	≤ 1792	≤ 1792	≤ 1792
IR	≤ 8	≤ 32	≤ 64	≤ 128
OR	≤ 8	≤ 32	≤ 64	≤ 128
IG	≤ 4	≤ 4	≤ 8	≤ 16
OG	≤ 8	≤ 32	≤ 64	≤ 128

[1] For the PC-1100, the maximum number of holding registers depends on memory size, as described in Section 4.

OPERAND 2 - TABLE END

Table end defines the type and number of the last register in the table. The type and number are limited, as indicated in Table 2.

OPERAND 3 - POINTER

In the PC-1100, the pointer is a group that holds the number of the bit (up through 4096) to be operated. If the pointer is greater than 4096, it is set to zero, and the coil is turned OFF. The pointer is a register or group:

- Holding Register (HR)
- Input Register (IR)
- Output Register (OR)
- Input Group (IG)
- Output Group (OG)

In the PC-1200, the pointer is a group that holds the number of the bit (up through 16 times the Table Length) to be operated. In the PC-1200, if the pointer is greater than 16 times the Table Length, it is set to zero, and the coil is turned OFF. If the pointer is zero, the coil is turned OFF.

BO TRUTH TABLE

See Table 3

TABLE 3. BO TRUTH TABLE

Set	Reset	Result
0	0	The coil gives the status of the bit indicated by the pointer. 0 = De-energized 1 = Energized
0	1	The bit indicated by the pointer resets. The coil de-energizes.
1	0	The bit indicated by the pointer sets. The coil energizes.
1	1	Same as Set = 0, Reset = 1 (reset overrides set).

APPLICATIONS

The BO function monitors and operates a bit(s) in a table. The program in Figure 2 illustrates how the function is used.

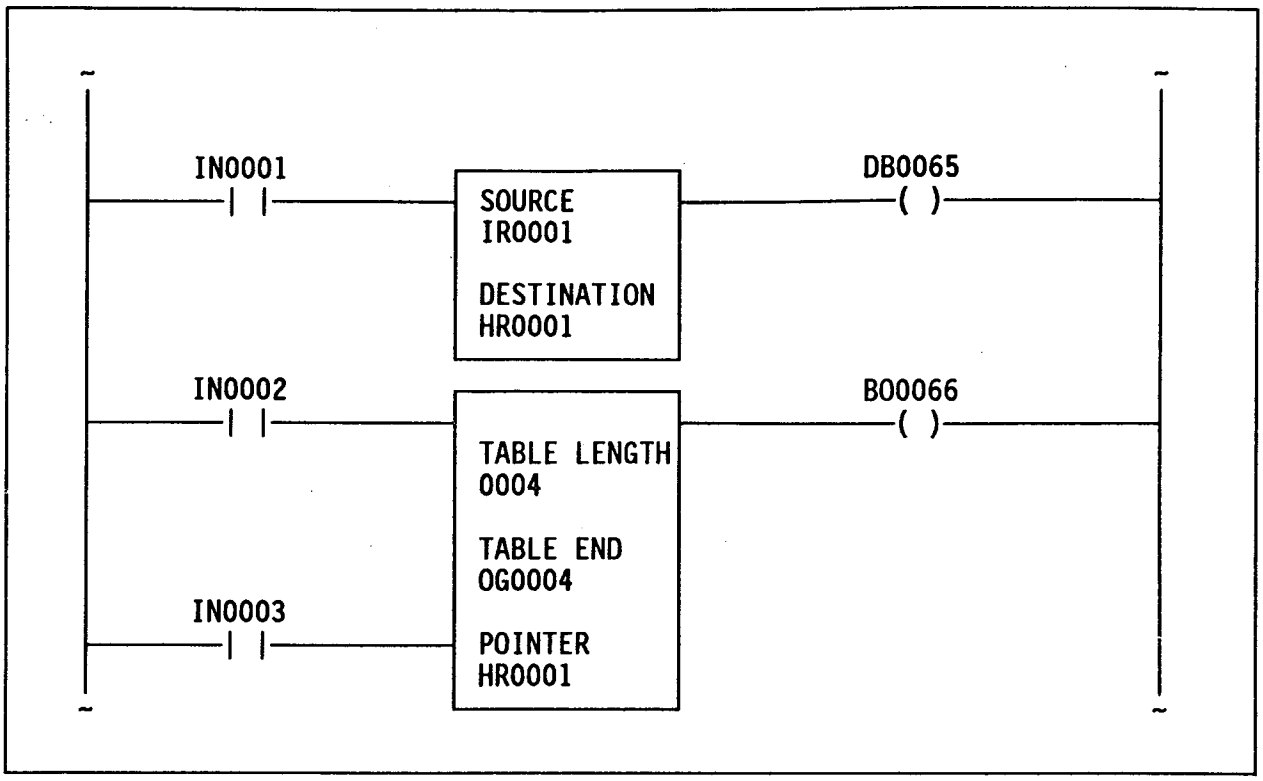


Figure 2. BO Application Program

Data for the pointer enters via a four-digit thumbwheel. DB0001 converts the data in IR0001 from Binary Coded Decimal (BCD) to binary when IN0001, an ENTER DATA pushbutton, is pressed. This circuit allows the user to turn ON any of 64 outputs. The table represents outputs CR0001 through CR0064 the total available output capacity of the PC-1100; IN0002 turns the designated output on; IN0003 turns the designated output OFF.

A similar program, shown in Figure 3, allows the monitoring of a selected input. The data from a thumbwheel set is converted from BCD to binary every other scan. The state of the input whose number is present in the thumbwheel set (1 through 64) is reflected in the state of the BO0001 coil.

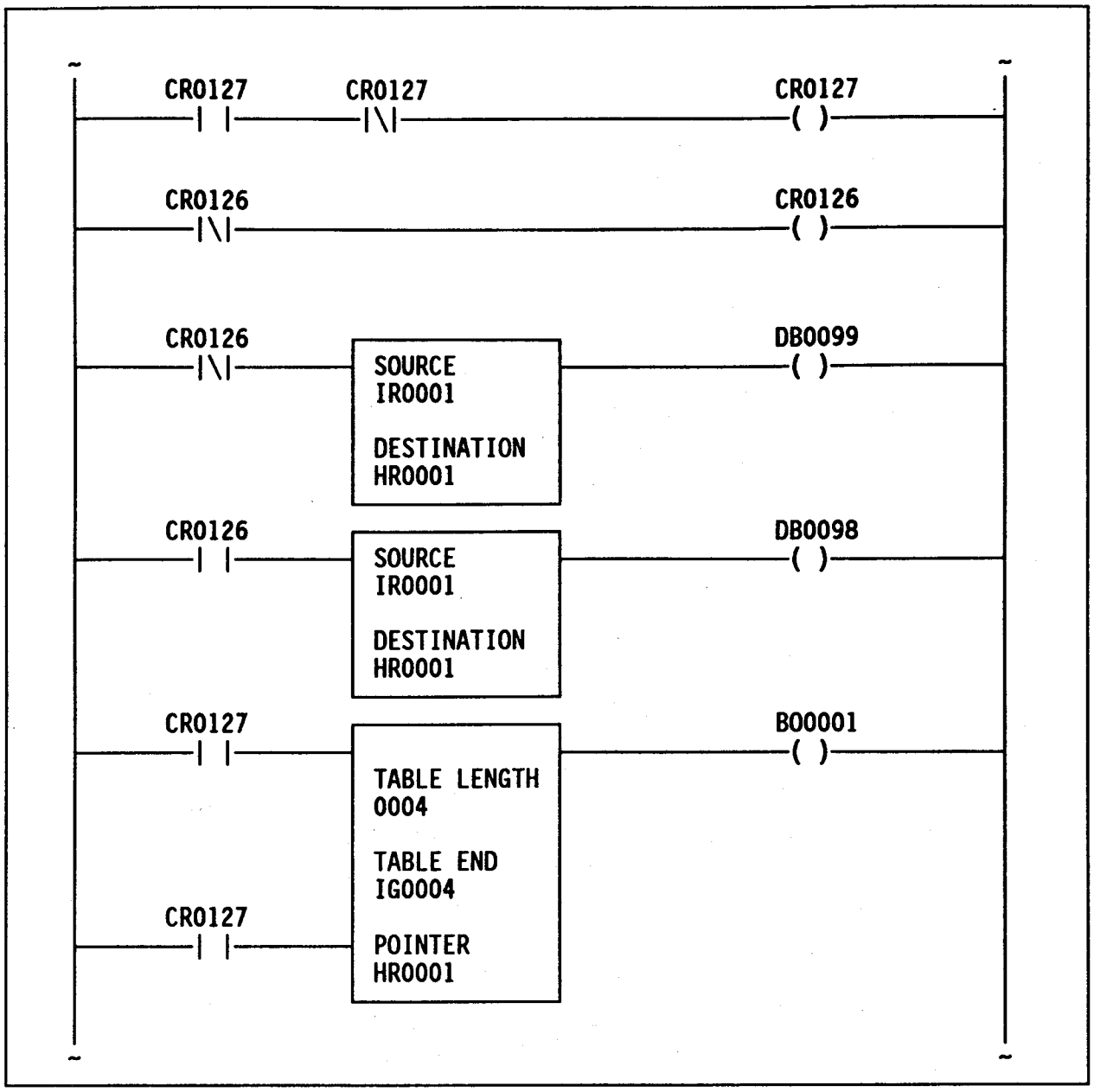


Figure 3. BO Input Monitor Application Program

PC-1100-x01y: SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The Bit Set (BS) and Bit Clear (BC) functions allow program control of a specific holding or output register bit. Both functions are effectively a latch; BS is latch ON, and BC is latch OFF. The Bit Pick (BP) contact checks the status of a latch. BS/BC function symbology is shown in Figure 1.

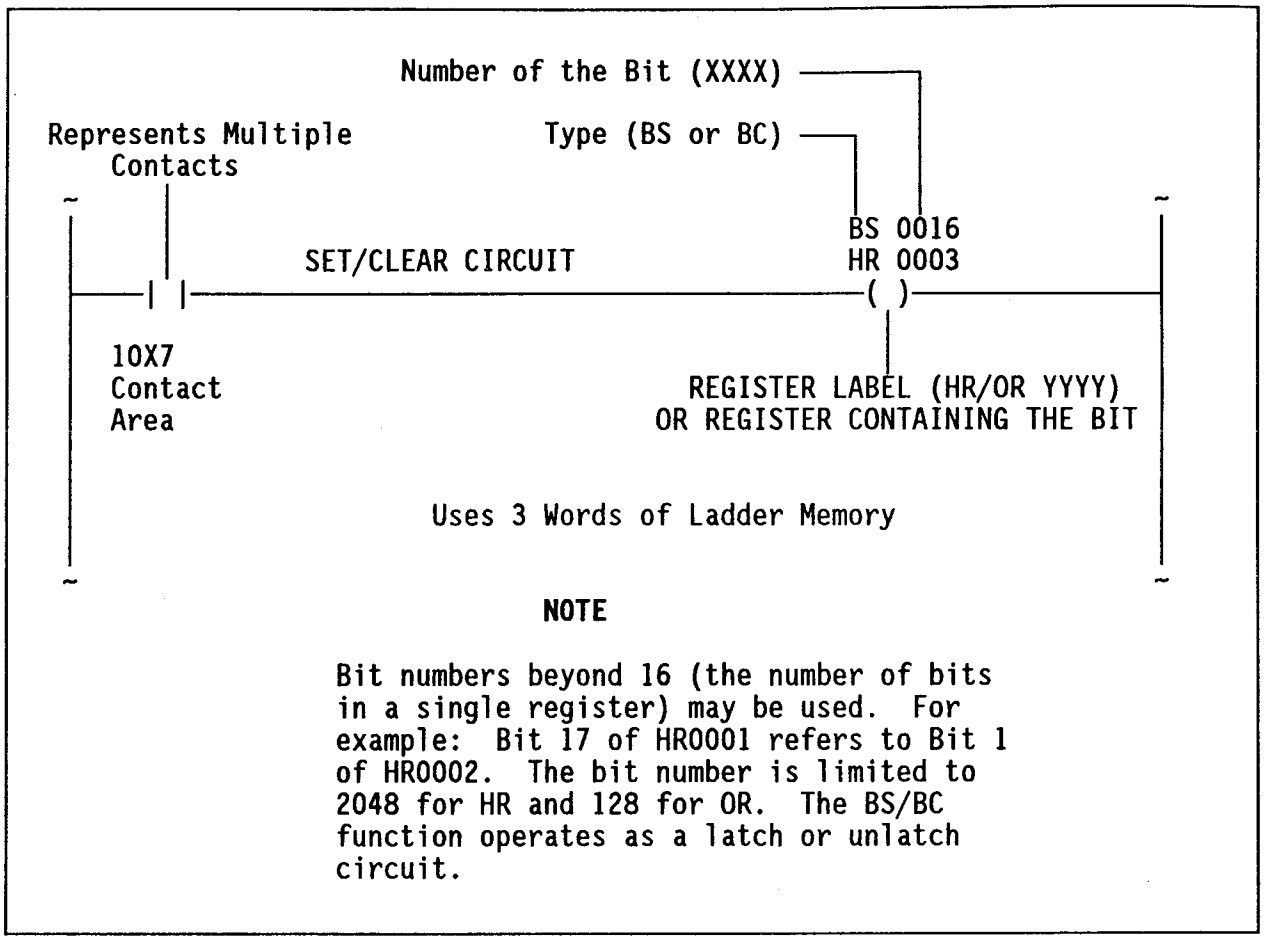


Figure 1. Bit Set (BS)/Bit Clear (BC)

SPECIFICATIONS

BS/BC CIRCUIT

- BS

  When BS is conducting, the designated bit is set and held in that condition until the circuit is opened.

- BC

  When BC is conducting, the designated bit is cleared and held in that condition until the circuit is opened.

Note

Neither function has effect when the circuit is non-conducting.

DESIGNATED REGISTER

The designated register specifies the register in which the bit is to be set or cleared. The register may be a:

- Holding Register (HR)
- Output Register (OR)

TYPE

The type designates the type of function to be performed:

- BS
- BC

NUMBER OF BIT

Number of bit specifies the bit number of the designated register to be set or cleared.

COIL

When the coil energizes, the function sets or clears the designated bit in a specified register. When the coil de-energizes, the function has no effect on the register.

Note

BS/BC operation is as follows in the Master Control Relay (MR) function:

MR ON - Normal operation.

MR OFF - BS and BC are disabled and the register bit does not change.

The bit position does not clear in the BS function. The BS or BC coil is counted as a coil to determine the range of MR.

APPLICATIONS

The BS and BC functions are used as latches by designating a holding or output register for use with the Latch functions. Setting the latch performs the Latch function; clearing the latch performs the Unlatch function The Bit Pick (BP) contact function is used to find current latch status.

In Figure 2, Bit 2 of HR0500 is set (latched) only if IN0001 is present and Bit 1 is not set to one. This figure illustrates the use of BP contacts; however, they are not essential for latching operations. Bit 1 of HR0500 is cleared (unlatched) only if IN0002 is present and if Bit 1 is already set to one. The maximum number of latches is limited only by the number of available holding and output registers.

Note

In regard to the order of execution in a program, if both IN0001 and IN0002 are ON, the remainder of the program sees the bit as OFF (except for the BC function).

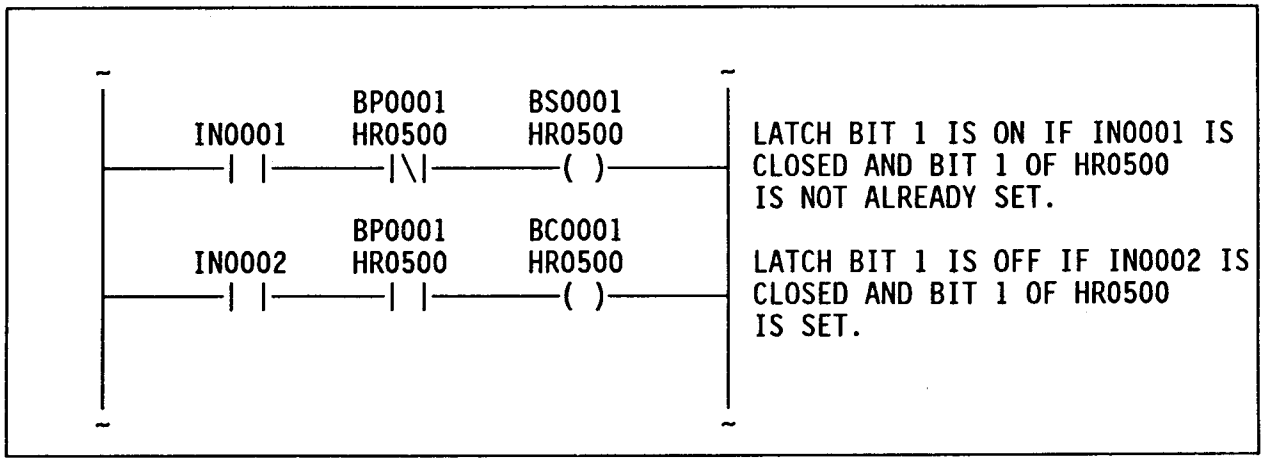


Figure 2. Bit Latch Application

BT - BLOCK TRANSFER

PC-1100-x01y: NOT SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The Block Transfer (BT) function allows the user to copy a table of registers, 1 through 256 registers in size, into a similarly-sized table of registers. The transfer occurs when the enable circuit controlling the function changes from open (non-conducting) to closed (conducting). BT function symbology is shown in Figure 1.

OP CODE

Op Code 60 defines the Literal (LT) as the BT function. Whether or not the Literal function should be used depends upon the capability of your program loader. It is recommended that the mnemonic function be used whenever possible. Refer to the Introduction in this Section and the LT function description.

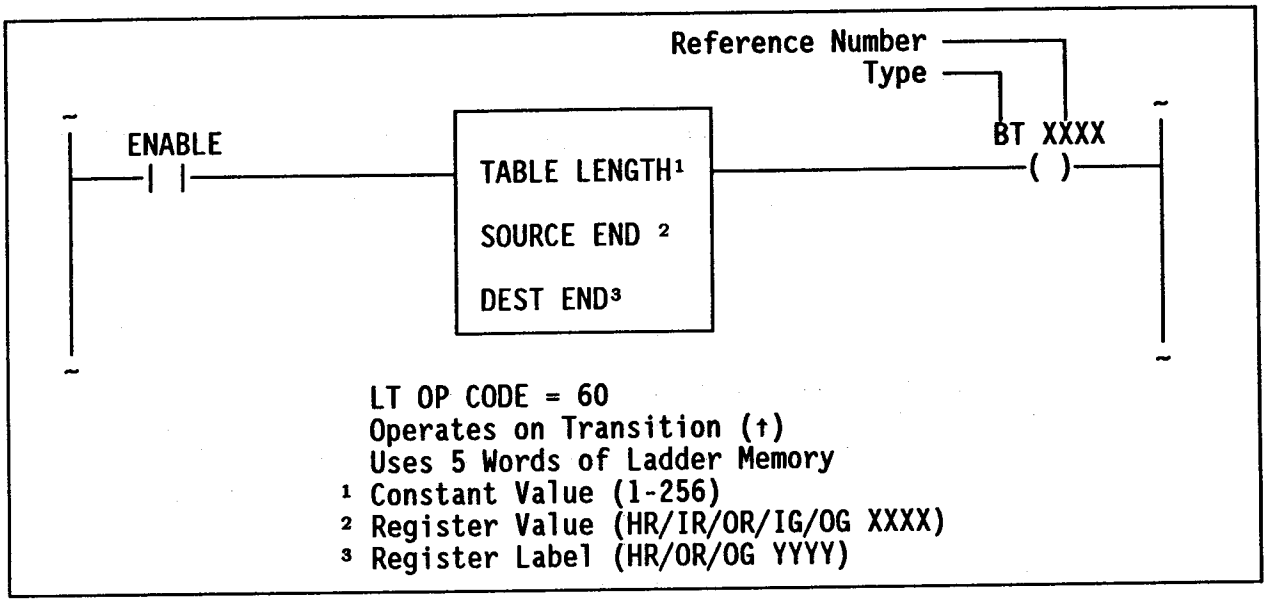


Figure 1. Block Transfer (BT)

SPECIFICATIONS

OPERAND 1 - TABLE LENGTH

The Table Length is a constant value that defines the number of registers involved in the transfer. The range is from 1 through 256, with the limits listed in Tables 1 and 2.

Note

The highest Holding Register reference number acceptable is dependent of the memory and user program size.

OPERAND 2 - SOURCE END

The source end defines the type and number of the last register in the table being duplicated. The limitations for each type of register or group are listed in Table 1.

OPERAND 3 - DEST END

The destination end defines the type and number of the last register in the table at the new location. The limitations for each type of register or group are listed in Table 2.

TABLE 1. SOURCE END LIMITS

Type	PC-1100	PC-1200-1020/1040	PC-1200-1041/1041/1042	PC-1250
HR	≤ 1792 [1]	≤ 1792	≤ 1792	≤ 1792
IR	≤ 8	≤ 32	≤ 64	≤ 128
OR	≤ 8	≤ 32	≤ 64	≤ 128
IG	≤ 4	≤ 4	≤ 8	≤ 16
OG	≤ 8	≤ 32	≤ 64	≤ 128

[1] For the PC-1100, the maximum number of holding registers depends on memory size, as described in Section 4.

TABLE 2. DESTINATION END LIMITS

Type	PC-1100	PC-1200-1020/1040	PC-1200-1041/1041/1042	PC-1250
HR	≤ 1792 [1]	≤ 1792	≤ 1792	≤ 1792
OR	≤ 8	≤ 32	≤ 64	≤ 128
OG	≤ 8	≤ 32	≤ 64	≤ 128

[1] For the PC-1100, the maximum number of holding registers depends on memory size, as described in Section 4.

BT TRUTH TABLE

See Table 3.

TABLE 3. BT TRUTH TABLE

Enable	Result
0	The coil de-energizes. The source and destination registers do not change.
↑	The coil energizes. The source table is duplicated on a register basis in the destination table.
1	The source and destination registers do not change.

APPLICATIONS

The BT function facilitates the movement of relatively large blocks of data from one location to another in processor memory.

In the example shown in Figure 2, several different output states can be monitored. The program in Figure 3 implements this concept.

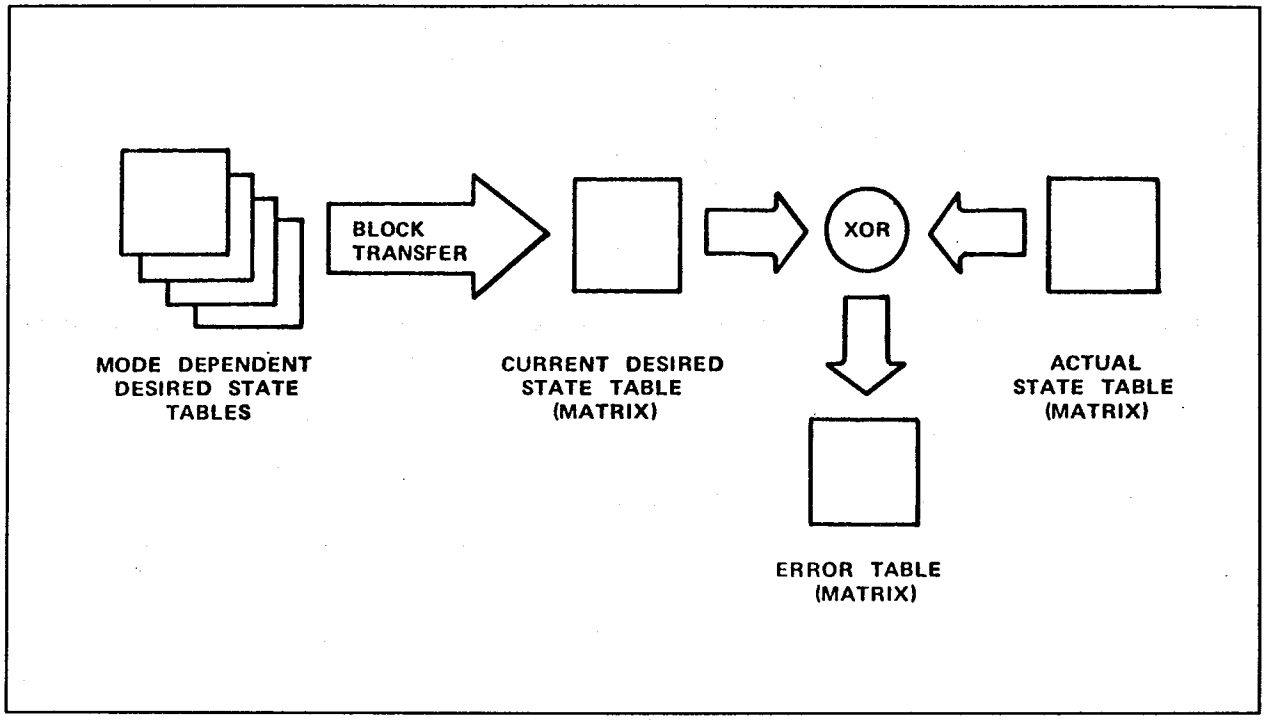


Figure 2. Monitoring Outputs Using the BT Function

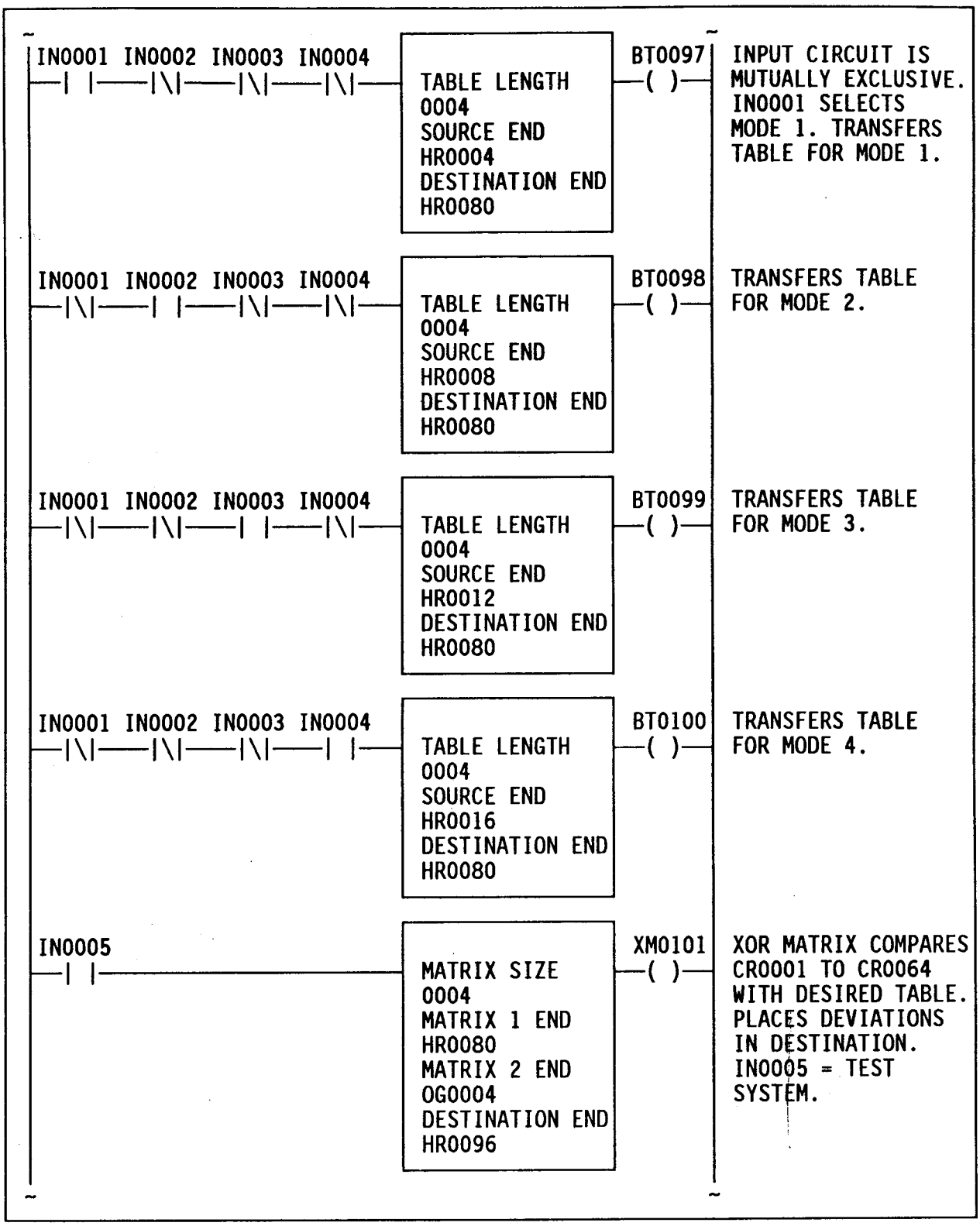


Figure 3. BT Application Program

CG - CONTINUOUS GROUP SELECT

PC-1100-x01y: NOT SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The Continuous Group Select (CG) function allows the user to change a transitional functional, or group of transitional functions to continuous operation. When the ENABLE contact closes, the specified coil or output group in the operand operates on every scan if enabled. CG function symbology is shown in Figure 1.

OP CODE

Op Code 15 defines the Literal (LT) as the CG function. Whether or not the Literal function should be used depends upon the capability of your program loader. It is recommended that the mnemonic function be used whenever possible. Refer to the Introduction in this Section and the LT function description.

SPECIFICATIONS

OPERAND 1

Operand 1 determines the type of CG function to be performed. When the operand is a constant value, the number (range of 1 through 256 for PC-1100, range of 1 through 1024 for PC-1200) determines which coil of a transitionaly type special function operates continuously when enabled. When the operand is an output group, it determines which group of 16 coils associated with the transitional special functions operates continuously when enabled. OG0001 consists of Output Coils 1 through 16, OG0002 consists of Output Coils 17 through 32, etc.

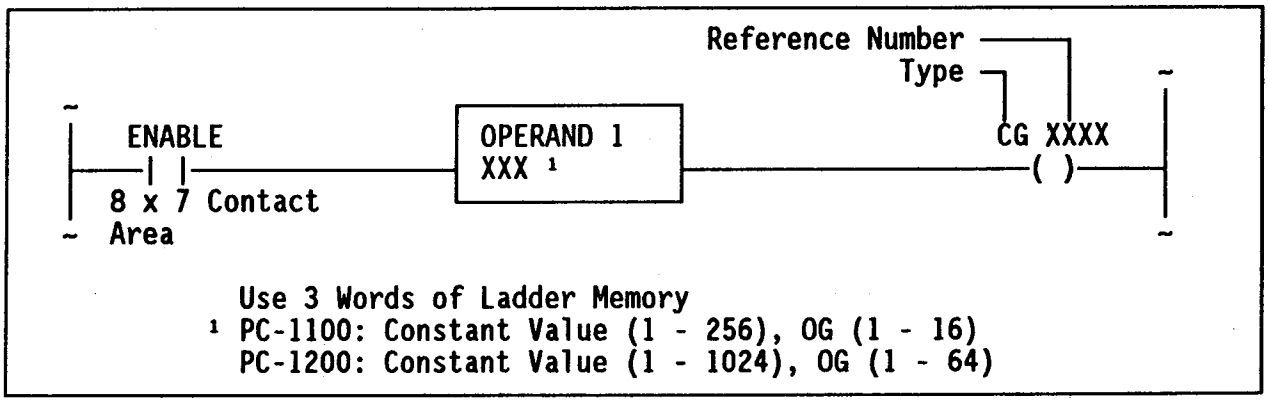


Figure 1. Continuous Group Select (CG)

CG TRUTH TABLE

See Table 1.

TABLE 1. CG TRUTH TABLE

Step	Result
0	The coil de-energizes. The specified special function or group of special functions operates normally.
1	The coil energizes. The special function or group of special functions specified by Operand 1, if enabled, operates every scan.

APPLICATIONS

When several transitionally-operated functions (e.g., AD, SB, BD, DB, etc.) are used in a program, they can all be grouped in one or two output groups and operated continuously. To select a particular group, for example, Coil Reference No. CR0017 through CR0032 (OG0002), the line shown in Figure 2 is added to the ladder diagram of the program.

The CG function is also used to cause a single function to operate continuously, when the function is enabled, by specifying a constant value to Operand 1. This value is the coil number of the continuously selected function.

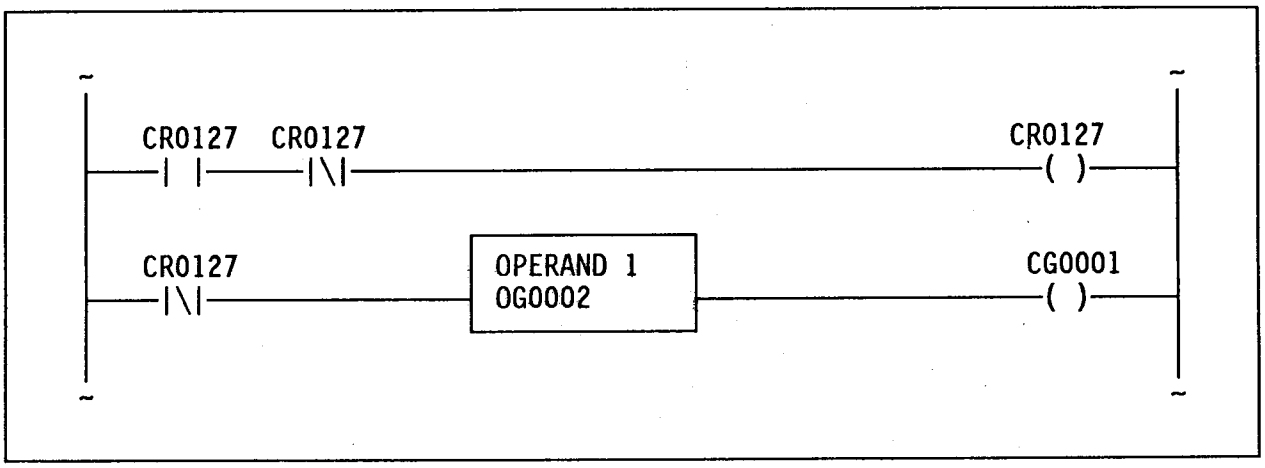


Figure 2. CG Application

CM - COMPLEMENT MATRIX

PC-1100-x01y: NOT SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

The Complement Matrix (CM) function causes the bit states of the contents of a matrix to reverse; all zeroes become one's and all one's become zeroes. The result is placed in the destination location. CM function symbology is shown in Figure 1.

The CM operation occurs when the enable circuit changes from non-conducting to conducting. The contents of the original matrix are unaffected, as shown in Figure 2.

OP CODE

Op Code 57 defines the Literal (LT) as the CM function. Whether or not the Literal function should be used depends upon the capability of your program loader. It is recommended that the mnemonic function be used whenever possible. Refer to the Introduction in this Section and the LT function description.

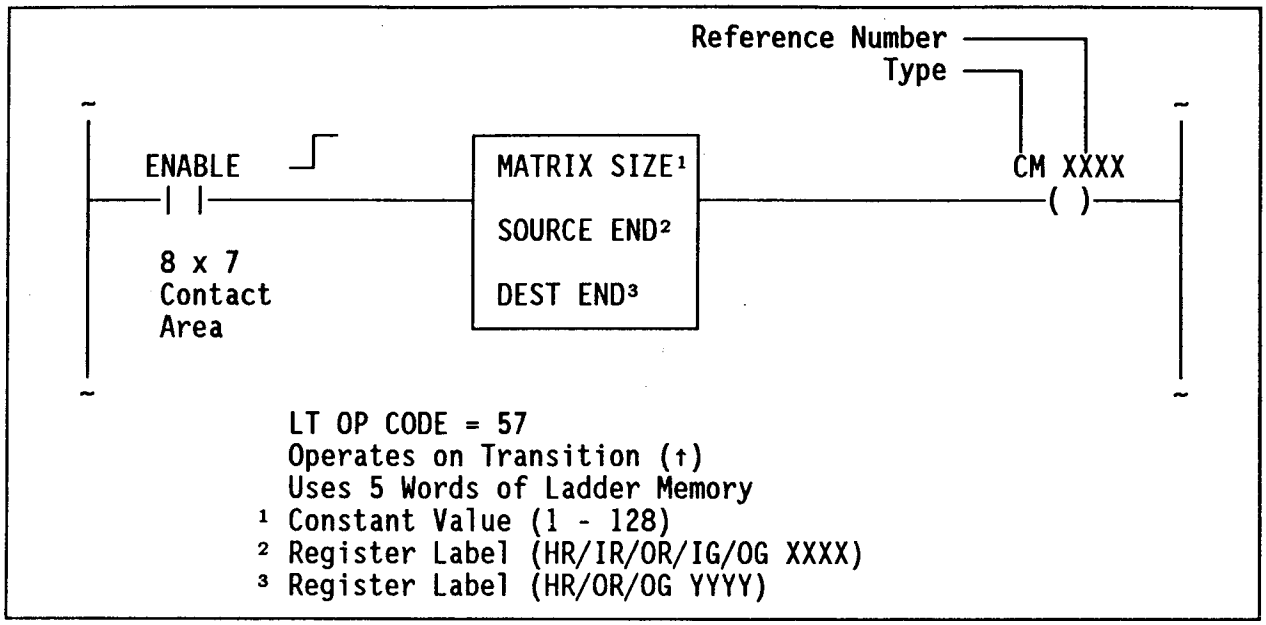


Figure 1. Complement Matrix (CM)

ORIGINAL MATRIX

	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
HR0001	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
HR0002	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

COMPLEMENTED MATRIX

	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
HR0003	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
HR0004	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0

Figure 2. Complementing a Matrix

SPECIFICATIONS

OPERAND 1 - MATRIX SIZE

The matrix size is a constant value that defines the number of registers included in the matrix. The range is 1 through 128, and is subject to the limitations of Tables 1 and 2.

Note

The highest Holding Register reference number acceptable is dependent on memory and user program size.

TABLE 1. SOURCE END/MATRIX SIZE LIMITS

Type	PC-1100	PC-1200-1020/1040	PC-1200-1041/1041/1042	PC-1250
HR	≤ 1792 [1]	≤ 1792	≤ 1792	≤ 1792
IR	≤ 8	≤ 32	≤ 64	≤ 128
OR	≤ 8	≤ 32	≤ 64	≤ 128
IG	≤ 4	≤ 4	≤ 8	≤ 16
OG	≤ 8	≤ 32	≤ 64	≤ 128

[1] For the PC-1100, the maximum number of holding registers depends on memory size, as described in Section 4.

TABLE 2. DESTINATION END/MATRIX SIZE LIMITS

Type	PC-1100	PC-1200-1020/1040	PC-1200-1041/1041/1042	PC-1250
HR	≤ 1792 [1]	≤ 1792	≤ 1792	≤ 1792
OR	≤ 8	≤ 32	≤ 64	≤ 128
OG	≤ 8	≤ 32	≤ 64	≤ 128

[1] For the PC-1100, the maximum number of holding registers depends on memory size, as described in Section 4.

OPERAND 2 - SOURCE END

The source end defines the type and number of the last register in the original matrix. The type and number are subject to the limits in Table 1.

OPERAND 3 - DESTINATION END

The destination end defines the type and number of the last register in CM, according to the limits in Table 2.

COIL

The coil follows the enable circuit.

CM TRUTH TABLE

See Table 3.

TABLE 3. CM TRUTH TABLE

Enable	Result
0	The coil de-energizes. The source matrix and destination matrix do not change.
↑	The coil energizes. The bit pattern from the source matrix reverses (i.e., 1's become zeroes; zeroes become 1's). The source matrix is unaffected.
1	The Source Matrix and Destination Matrix do not change.

APPLICATIONS

In many cases, it is important to know what is not occurring, rather than what is occurring. If, for example, there is a situation in which outputs CR0001 through CR0032 are normally concurrently ON, it is easier to check to see which outputs are not ON, instead of which outputs are ON.

Figure 3 shows a situation where CR0005 is not operating. After the matrix is complemented, the Search Matrix (SM) function with a bit register of OR0001 tells the operator that CR0005 failed to come ON.

ORIGINAL MATRIX

HR0001

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1

HR0002

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

COMPLEMENTED MATRIX

HR0003

16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

HR0004

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3. CM Application Example

Figure 4 shows the ladder diagram for the CM function. When IN0001 is operated, OG0001 and OG0002 are complemented and placed in HR0001 and HR0002. When IN0002 is operated, the bit register holds the number of the first "1" bit in the matrix. The bit register is converted from binary to Binary-Coded-Decimal (BCD) for display purposes.

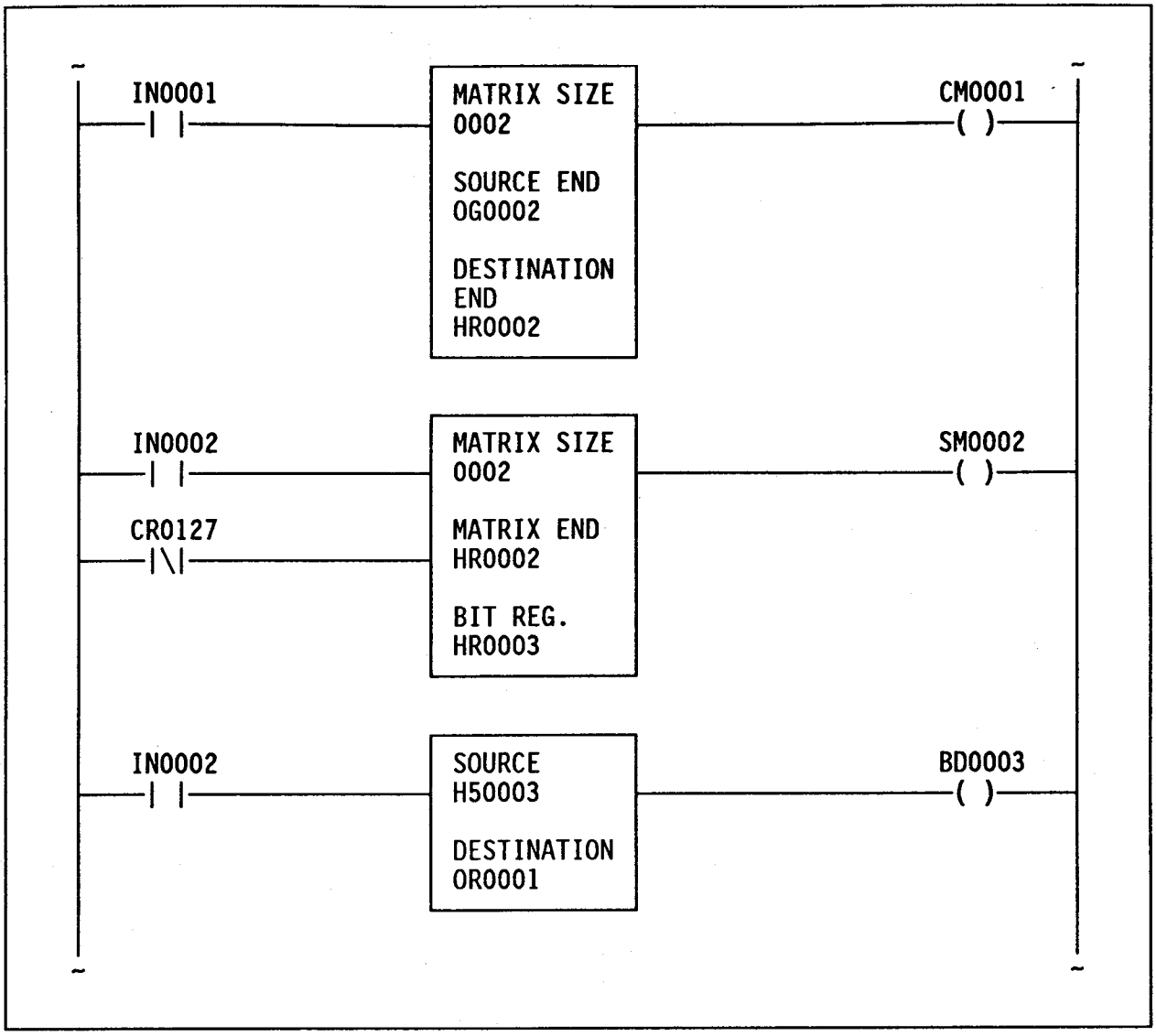


Figure 4. CM Application Program

PC-1100-x01y: NOT SUPPORTED	PC-1100-x05y: SUPPORTED
PC-1100-x02y: SUPPORTED	PC-1200-x02y: SUPPORTED
PC-1100-x03y: SUPPORTED	PC-1200-x04y: SUPPORTED

DESCRIPTION

Configure Port is one of three special functions used in networking. Refer to Section 3 for an overview of networking and to the PT (Port Transmit) and UA (Unit Address) special function descriptions in this section for additional information.

The Configure Port (CP) function is used to change the communications parameters of the A and/or B ports if other than the standard default settings are desired. Baud rate, number of stop bits, parity and number of data bits can be configured by the CP function.

In the PC-1100, a transition into Run, either by keyswitch from Stop or from a Power-up condition, enables a CP search to initialize the ports based on the CP Operand 2 configuration register and indirect operand port timing values. In addition, during Run, the port configuration and port timing can be redefined by moving an alternate legal configuration to Operand 2 and the indirect operand, and then causing the enable line to go from low to high.

In the PC-1200, a transition into Run will execute the CP function and define port setup *only if the Enable input is on*. Upon low to high transition of the Enable input, the port will be configured based on Operand 2 and the indirect operand.

If no CP special function is programmed, then Port A defaults to the DIP switch selection while Port B use the default of the program loaders: 9600 baud, 8 data bits, odd parity, and 2 stop bits. Refer to Tables 3-2 and 3-3 and Figures 3-28 through 3-30 for DIP switch selection.

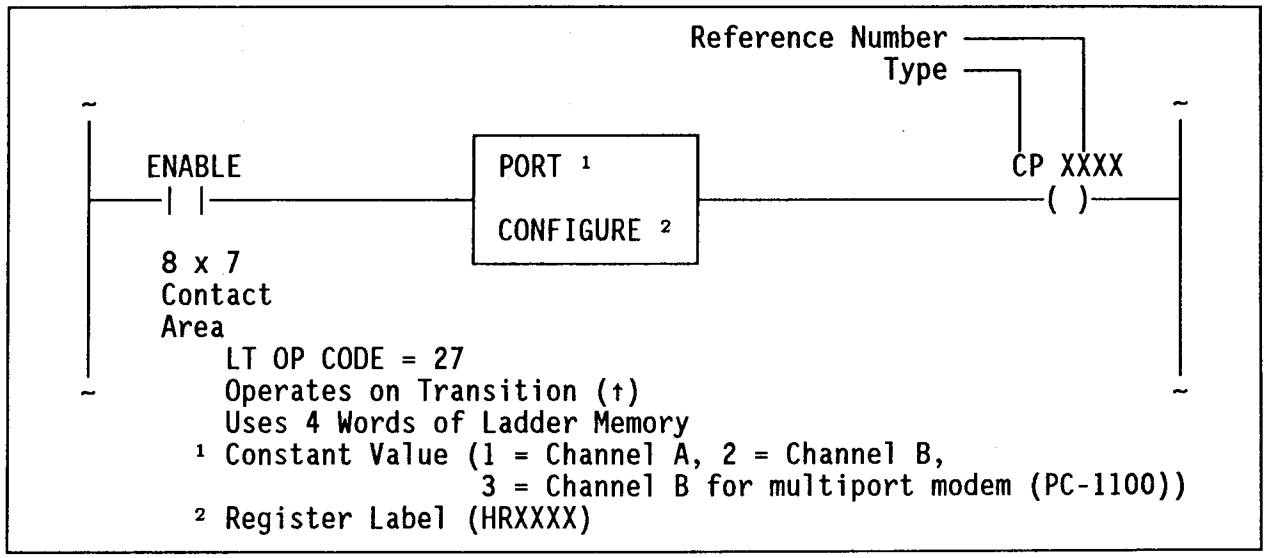


Figure 1. Configure Port Concept

A CP function is not required if the default configuration is wanted. When networking PCs (see Section 3) the B port configuration must be the same for the Master and all Slaves on the link in order to obtain communications. When changing the configuration of both the A and B ports, two CP functions, one for each port will be required.

The CP function is also used to facilitate modem communications. Port B can be configured for a transmit time delay of 0.1 to 12.8 or 25.2 (depending on software version) seconds that the PC-1100 waits before responding to the Clear-to-Send signal generated by the modem.

FAULT MODE COMMUNICATIONS

Upon power-up the PC checks for memory fault bits 12,13,15, or 16. If any of these faults exist, then on early PC-1100s:

> the PC examines the "Mode" (single-point/multi-point)switch to determine whether channel B requires the "Set PC address" command to be enabled. The PC then examines the states of the channel A and B initialization tables. If these tables match there check sum then the ports will be re initialized to these parameters.

In PC-1100 executive software levels V 3.6 or greater and all PC-1200s if a memory fault occurs then the communication initialization tables are assumed to be corrupted. If assumed or found to be corrupted then the default format is automatically invoked.

Note

Memory Faults can override CP parameters with the system resetting to Default settings.

If a CP function is not found upon power-up on the PC-1100 or if CP Enable input on the PC-1200 is low, then the default format for the ports will be used, modified according to the DIP switch baud and parity switches.

OP CODE

Op Code 27 defines the Literal (LT) as the CP function. Whether or not the Literal function should be used depends upon the capability of your program loader. It is recommended that the mnemonic function be used whenever possible. Refer to the Introduction in this Section and the LT function description.

SPECIFICATIONS

OPERAND 1 - PORT

The communication port (channel) to be configured is defined by a constant value of 1 to 3. These equate to:

Constant Value	Definition
1	Port A will be reconverted as per Operand 2
2	Port B will be reconverted as per Operand 2
3*	Port B will be reconverted as per Operand 2 plus an indirect operand is created primarily for modem use.

* On the PC-1100 this is intended to coordinate with the PT special function in turning on the hand shake lines when networking with multipoint modems and to provide for a Slave response time out variable. See discussion on Indirect Operand.

OPERAND 2 - CONFIGURE

Operand 2 defines the Configuration Register and if Operand 1 is set to a constant value of 3, then it also defines use of the Indirect Operand for Transmit delay and PT Slave response time-out. The previous register in memory with respect to Operand 2 is the location of the Indirect Operand.

Figures 2a and 2b show the programmable port configurations allowed. When entering the configuration call up the referenced register with a program loader in the Hexadecimal format. Find the appropriate selections in regards to number of data bits, stop/parity, and baud rate and enter them in the configuration register. If the configuration is left at zero the default Port configurations will be invoked.

INDIRECT OPERAND

In the PC-1100, when Port B is configured for modem networking (Operand 1 = 3), then the Indirect Operand is referenced one register previous to Operand 2.

- In PC-1100s with executive firmware level V3.6 or greater this indirect operand is split in two. The upper byte defines the Slave response time-out for PT function while the lower byte defines a Transmit Delay waited after the processor sees CTS go high.

- In earlier PC-1100s (< V3.6) the Transmit Delay and Slave Time-Out Delay are one in the same variable restricted to the lower byte with a range of 0.1 to 12.8 seconds.

In the PC-1200, when Port B is configured for modem networking (Operand 1 = 3), then the Indirect Operand is referenced one register previous to Operand 2. In the PC-1200, this operand only defines the Transmit Delay which starts from

when the processor pulls RTS high. In contrast, the Slave Response Time-Out variable is an Indirect Operand in the PT function, associated with the function it pertains to.

The following table summarizes the Indirect Operand.

PC Version	High Byte	Low Byte
PC-1100 ≥ V3.6	Slave Response Time out (0.1 - 25.5 Sec)	Transmit Delay (0.1 - 25.5 Sec)
PC-1100 < V3.6	Slave Resp. Time out/Transmit Delay (0.1 - 12.8 Sec)	
All PC-1200s	RTS to Transmit Delay (0.001 to 32.767 Sec.)	

If the Slave Response Time-Out is zero, then it defaults to 0.5 seconds. If the Transmit Delay is zero, no delay is invoked.

The Slave Response Time-out is the time a Master PC in a network waits for a response from a Slave after issuing a command. Refer to the Port Transmit (PT) special function description.

If an RS-232 interface device, such as a modem, has no RTS/CTS time delay, then the Transmit Delay can be used to delay data transfer after RTS is held high, based on the programmed value. On the PC-1100, RTS and CTS must be strapped together.

CP TRUTH TABLE

See Table 1.

TABLE 1. CP TRUTH TABLE

Enable	Result
0	The coil de-energizes. The configuration for the designated communications channel does not change.
(PC-1100)↑[1]	The coil energizes. The communications channel is initialized to the parameters specified in Operand 2.
(PC-1200)↑*	If parameters are valid, the communication channel is initialized according to those parameters and the coil energizes. If parameters are invalid, channel is not re-energized.
1	The coil status and the communications channel configuration do not change.
[1]Transition from OFF to ON.	

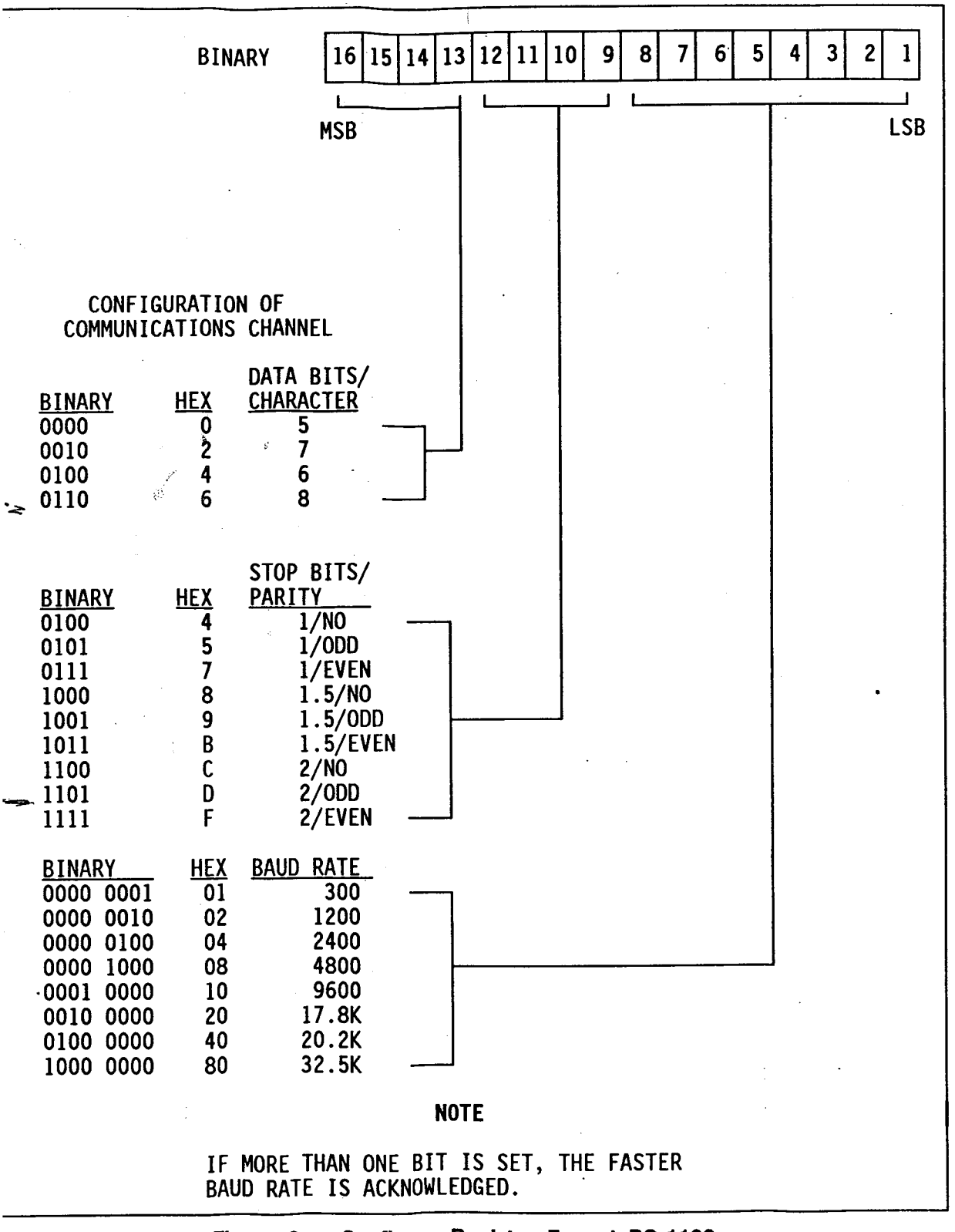


Figure 2a. Configure Register Format PC-1100

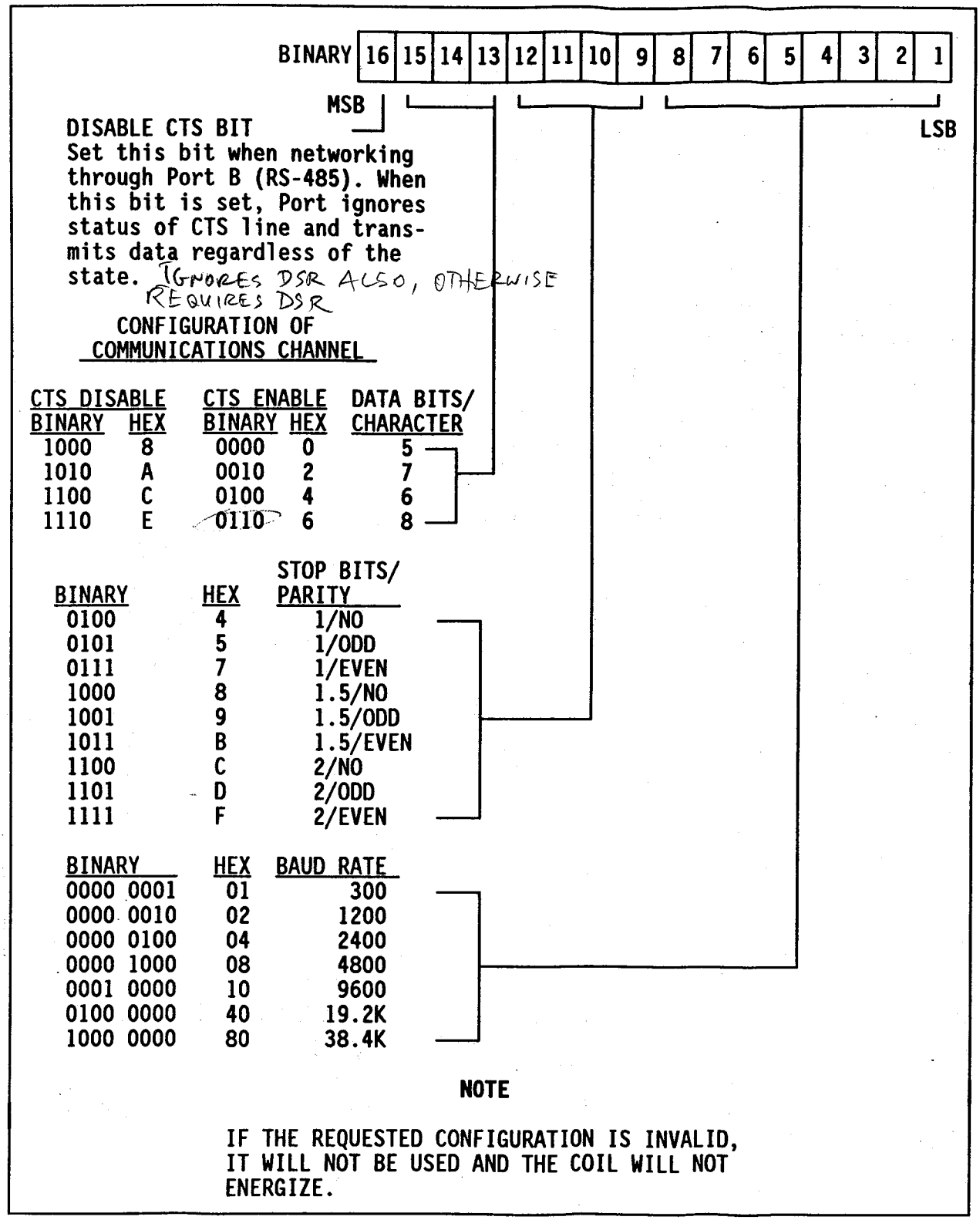


Figure 2b. Configure Register Format PC-1200

Port 6D16

APPLICATIONS

The PC communicates asynchronously at 9600 (or 1200) baud with a 12 bit data frame consisting of one start bit, 8 data bits, one odd parity bit (or not parity) and two stop bits unless otherwise defined by the Configure Port (CP) function. 1200/9600 baud and odd/no parity selections are also defined by DIP switches on the CPU board. Refer to Section 3 of this Manual.

Assume that RS-232C Communications to a printer is desired through Channel B, with the following parameters:

- 7 data bits
- 1 stop bit
- odd parity
- 9600 baud

The CP function shown in Figure 3 configures Channel B according to HR0178.

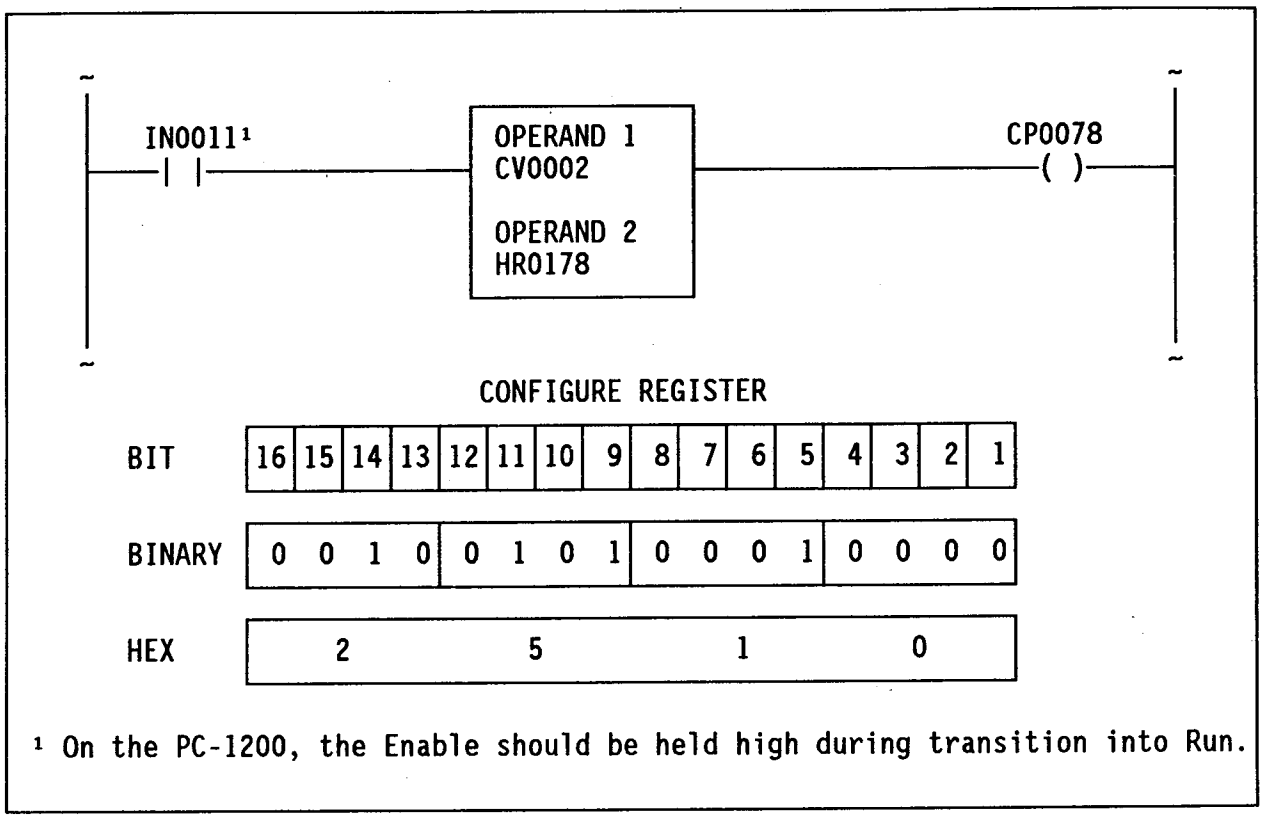


Figure 3. CP Application Example