SECTION 4. PROGRAMMABLE FUNCTIONS
PC-700/900 Programmable Controllers

4-1. USER INFORMATION

This section presents, in modular form, all the PC-700 and PC-900 programmable functions, with the exception of the Loop Controller (LC) function, which is described in Section 5. The programmable functions are listed in Table 4-1.

Mnemonic locators are provided on the outer top and bottom edges of each page to assist the user in locating each programmable function (see Fig. 4-1). The function's mnemonic is also included as part of the page number to further assist the user (see Fig. 4-1).

The PC-700/900 programmable functions are presented in the following order:

Mnemonic Locator	Function
AD/SB	Add/Subtract
AM	AND Matrix
AS	Ascending Sort
AT	ASCII Transmit
BD/DB	Conversions
BF	Bit Follow
BO	Bit Operate
BS/BC	Latches
BT	Block Transfer
CG	Continuous Group Select
CM	Complement Matrix
CR	Control Relay
CS	Continuous Select
DR	Drum Controller
DV	Divide
EQ/GE	Comparisons
FI/FO	First In Stack/First Out Fetch
FI/LO	First In Stack/Last Out Fetch
MP	Multiply
MR	Master Control Relay
MV	Move
NR/NL	N-Bit Serial Shift Registers
OM	OR Matrix
OT/CT	Open Table/Close Table
RP	Restore Program Counter
RT	Register-to-Table Move
RW	Reset Watchdog Timer
SK	Skip
SM	Search Matrix
SP	Save Program Counter
SQ	Square Root
SR/SL	Shift Registers
TL/TO	Table Lookup/Table Lookup Ordered
TR	Table-to-Register Move
TS/TT	Timers
UD/DC	Counters
UI	Update Immediate
US	Update Select
XM	XOR Matrix

Some of these programmable functions have special formats and are called Literal (LT) functions, as described in paragraph 4-2. The specifications for the PC-700/900 LT functions are shown in Table 4-2.

4

4-2. LITERAL (LT) FUNCTION

An LT function is a general, special function format that allows **Numa-Logic** Program Loaders to program new special functions without the need of specifically assigned keys, when those functions are supported by the processor being programmed. An LT function is shown in Figure 4-2.

CAUTION

Erratic processor operation results from incorrectly-entered LT functions. The number of input circuits, the assignment of operands, and register types are not checked for validity when the LT function is loaded. See Table 4-2 for futher information on each LT function within the PC-700/900 programmable functions. The reference number shown in Figure 4-2 is restricted to a maximum number of coils allowed by the processor being programmed. The Op Code shown in Figure 4-2 is restricted to a range of 11 through 128, and is subject to being supported by the processor being programmed.

TABLE 4-1. PROGRAMMABLE FUNCTIONS

SOFTWARE VERSIONS - PC-700
1.0—1.6 Extended Special Functions Less Bit Follow — Offline Programming
1.7—1.8 Extended Special Functions with Bit Follow — Offline Programming
2.X Standard Special Functions — Offline Programming
3.X Advanced Functions — Offline Programming
5.X Extended Special Functions — Online and Offline Programming
6.X Advanced II Functions — Online and Offline Programming

Function		PC-700 Software 2.X Standard Functions	PCX-700 Software 1.0—1.6 Extended Special Functions	PCX-700 Software 1.7—1.8 and 5.X Extended Special Functions	PC-700 Software 3.X Advanced Functions	PC-700 Software 6.X Advanced II Functions	PC-900A/ PC-900B Standard Functions	PC-900A/ PC-900B Extended Special Functions	PC-900B Advanced Functions
Add/Subtract	AD/SB		•	•	•	•		•	•
AND Matrix	AM				•	•			•
Ascending Sort	AS					•			
ASCII Transmit	AT					•			•
Bit Follow	BF			•	•	•	•	•	•
Bit Operate	BO				•	•			•
Block Transfer	BT				•	•			•
Comparisons	EQ/GE		•	•	•	•		•	•
Complement Matrix	CM				•	•			•
Continuous Group Select	CG					•			•
Continuous Select	CS				•	•			
Control Relay	CR	•	•	•	•	•	•	•	•
Conversions	BD/DB		•	•	•	•		•	•
Counters	UC/DC	•	•	•	•	•	•	•	•
Divide	DV		•	•	•	•		•	•
Drum Controller	DR		•	•	•	•	•	•	•
First In Stack/First Out Fetch	FI/FO				•	•			•
First In Stack/Last Out Fetch	FI/LO				•	•			•
Latches	BS/BC	•	•	•	•	•	•	•	•
Loop Controller	LC					•			
Master Control Relay	MR	•	•	•	•	•	•	•	•
Move	MV		•	•	•	•		•	•
Multiply	MP		•	•	•	•		•	•
N-Bit Serial Shift Registers	NR/NL				•	•			•
Open Table*/Close Table*	OT/CT					•			
OR Matrix	OM				•	•			•
Register-to-Table Move	RT				•	•			•
Reset Watchdog Timer	RW								•
Restore Program Counter	RP								•
Save Program Counter	SP								•
Search Matrix	SM				•	•			•
Shift Registers	SR/SL	•	•	•	•	•	•	•	•
Skip	SK	•	•	•	•	•	•	•	•
Square Root	SQ					•			
Table Lookup/Table Lookup Ordered	TL/TO					•			
Table-to-Register Move	TR				•	•			•
Timers	TS/TT	•	•	•	•	•	•	•	•
Update Immediate	UI					•	•	•	•
Update Select	US					•			
XOR Matrix	XM				•	•			•

*Not in Version 6.0.

TABLE 4-2. LITERAL FUNCTION SPECIFICATIONS

	Mnemonic	Op Code	Number of Inputs	Input Designation	Operand Number	Operand Function	Memory Use	Data Storage and Register Options					
								CV	HR	OR	IR	OG	IG
AND Matrix	AM	90	1	Enable	1	Matrix Size	6	•					
OR Matrix	OM	89	1		2	Matrix 1 End			•	•	•	•	•
XOR Matrix	XM	88	1		3	Matrix 2 End			•	•	•	•	•
					4	Destination End			•	•		•	
Ascending Sort	AS	58	1	Enable	1	Length	5	•					
					2	Table 1 End			•				
					3	Table 2 End			•				
ASCII Transmit	AT	91	1	Enable	1	Return Register	6		•	•		•	
					2	Table End			•				
					3	Pointer			•	•		•	
					4	Destination		•					
Bit Operate	BO	61	2	Set	1	Table Length	5	•					
					2	Table End			•	•	•	•	•
				Reset	3	Pointer			•	•	•	•	•
Block Transfer	BT	60	1	Enable	1	Table Length	5	•					
					2	Source End			•	•	•	•	•
					3	Destination End			•	•		•	
Close Table	CT	93	1	Enable	1	Table Length	6	•					
					2	Table End			•				
					3	Pointer			•	•	•	•	•
					4	Destination			•	•	•	•	•
Complement Matrix	CM	57	1	Enable	1	Matrix Size	5	•					
					2	Source End			•	•	•	•	•
					3	Destination End			•	•		•	
Continuous Group Select	CG	15	1	Step	1	Operand 1	3	•				•	
Continuous Select	CS	11	1	Step	1	Completes Function	3	•					
First Out Fetch	FO	85	1	Step	1	Table Length	6	•					
Last Out Fetch	LO	86	1		2	Table End			•	•		•	
					3	Pointer			•	•			
					4	Destination			•	•		•	
Loop Controller	LC	94	4	Manual	1	Set Point	6	•	•	•	•	•	•
				Auto	2	Process Variable			•	•	•	•	•
				Cascade	3	Output			•	•		•	
				Calculate	4	Loop Table End			•				
N-Bit Serial Shift Right	NR	50	3	Shift	1	Table Length	5	•					
N-Bit Serial Shift Left	NL	51	3	Serial In	2	Table End			•	•			
				Reset	3	N Bit Field		•	•	•	•		
Open Table	OT	92	1	Enable	1	Table Length	6	•					
					2	Table End			•				
					3	Pointer			•	•	•	•	•
					4	Source			•	•	•	•	•
Register-to-Table Move	RT	80	3	Step	1	Table Length	6	•					
First In Stack	FI	81	2		2	Table End			•	•		•	
				Reset	3	Pointer			•	•			
				Enable	4	Source			•	•	•	•	•
Reset Watchdog Timer	RW	14	2	Trigger	1	Operand 1	3		•				
				Reset									
Restore Program Counter	RP	13	1	Enable	1	Operand 1	3		•				
Save Program Counter	SP	12	1	Enable	1	Operand 1	3		•				

TABLE 4-2. LITERAL FUNCTION SPECIFICATIONS (Cont'd)

	Mnemonic	Op Code	Number of Inputs	Input Designation	Operand Number	Operand Function	Memory Use	Data Storage and Register Options					
								CV	HR	OR	IR	OG	IG
Search Matrix	SM	56	2	Step	1	Matrix Size	5	•					
					2	Matrix End			•	•	•	•	•
				Reset	3	Bit Register			•	•		•	
Square Root	SQ	19	1	Calculate	1	Source	4		•	•	•		
					2	Destination			•	•			
Table Lookup	TL	82	2	Step	1	Table Length	6	•					
Table Lookup Ordered	TO	83			2	Table End			•	•	•	•	•
				Reset	3	Pointer			•	•		•	
					4	Source			•	•	•	•	•
Table-to-Register Move	TR	84	3	Step	1	Table Length	6	•					
					2	Table End			•	•	•	•	•
				Reset	3	Pointer			•	•			
				Enable	4	Destination			•	•		•	
Update Select	US	25	1	Enable	1	I/O Quarters	4	•					
					2	Dummy Operand		•					

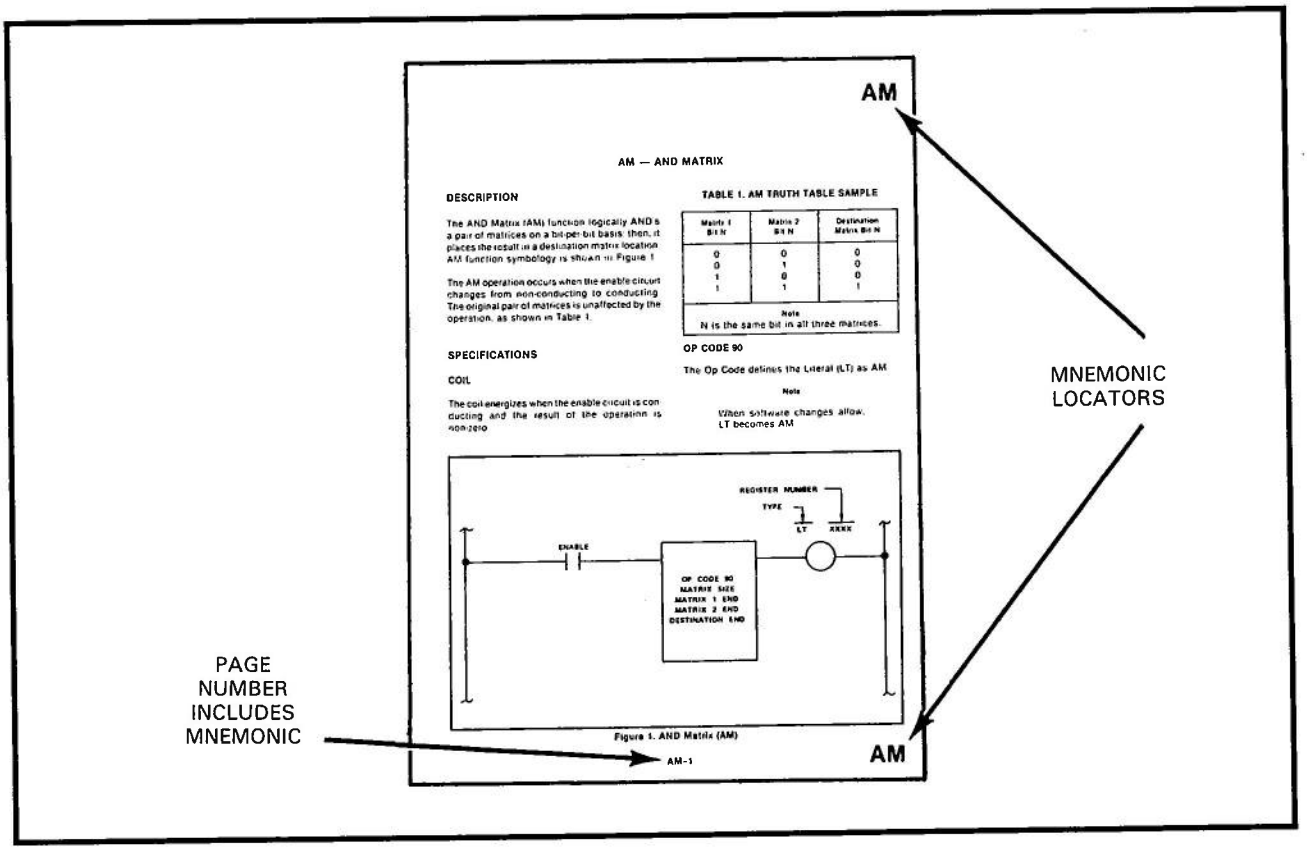


Figure 4-1. Mnemonic Locators and Page Number

AD/SB

SUBTRACT (SB)

Operand 2 is subtracted from Operand 1, and the result is placed in the destination 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
- The 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).

DESTINATION

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

Note

Both operands must be in binary form.

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 a set point for a process, it may be advantageous to allow process operation within a band around the set point rather than restricting operation to the set point alone. The 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.

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.

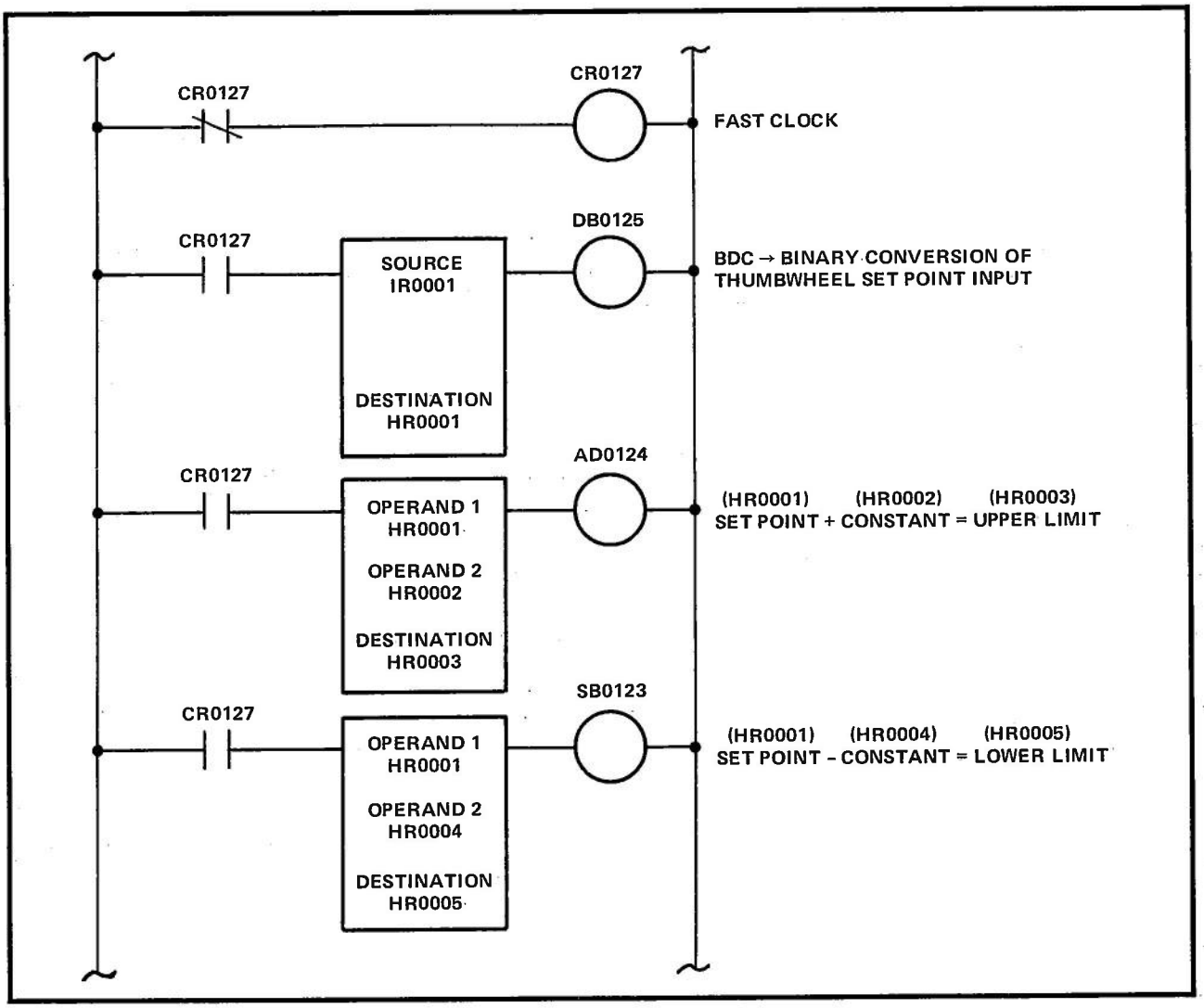


Figure 2. Establishing the Range of Operation

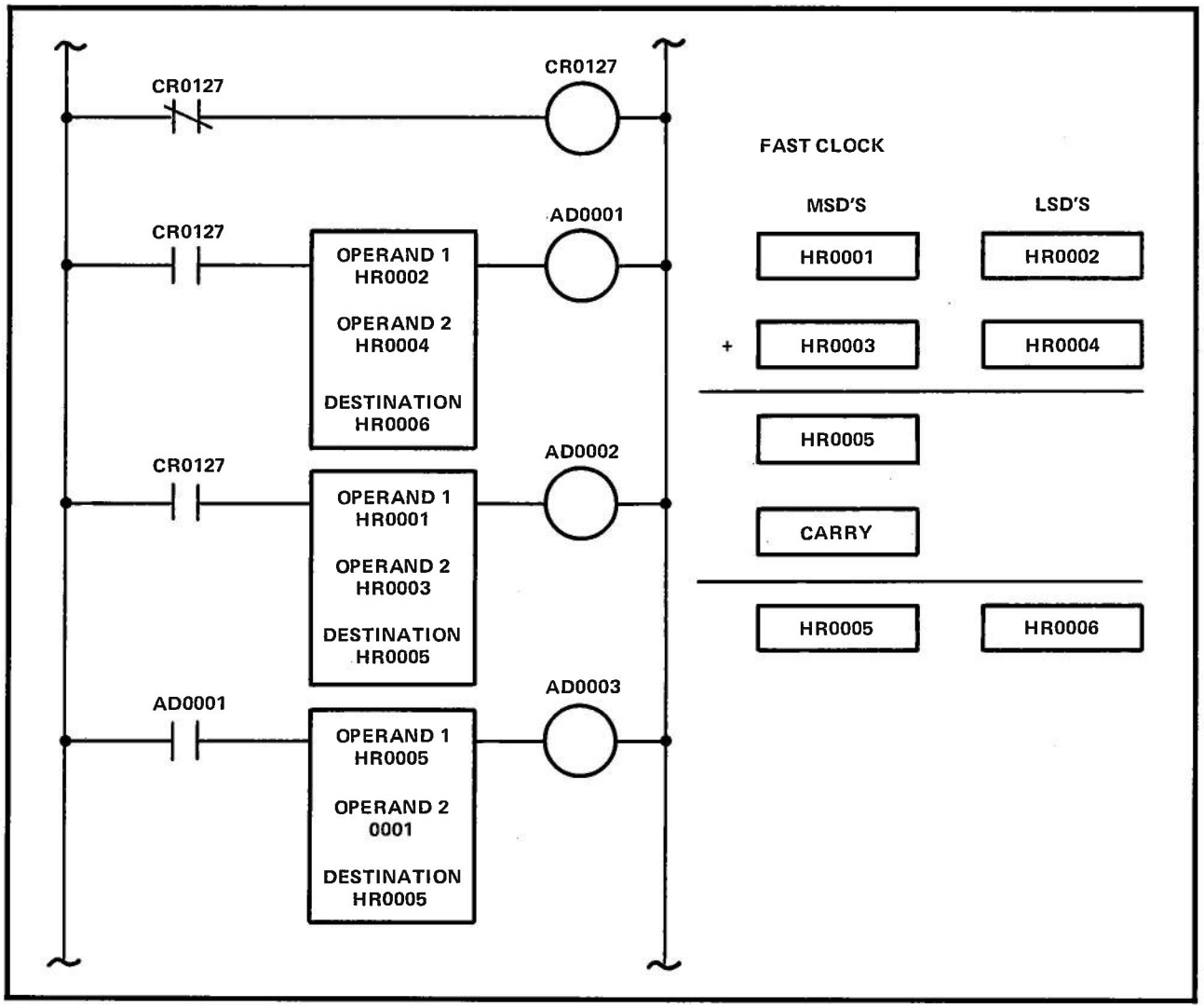


Figure 3. Double-Precision Add

AM — AND MATRIX

DESCRIPTION

The AND Matrix (AM) function logically AND's a pair of matrices on a bit-per-bit basis; then, it places the result in a destination matrix location. 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.

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 bit in all three matrices.

SPECIFICATIONS

COIL

The coil energizes when the enable circuit is conducting and the result of the operation is non-zero.

OP CODE 90

The Op Code defines the Literal (LT) as AM.

Note

When software changes allow, LT becomes AM.

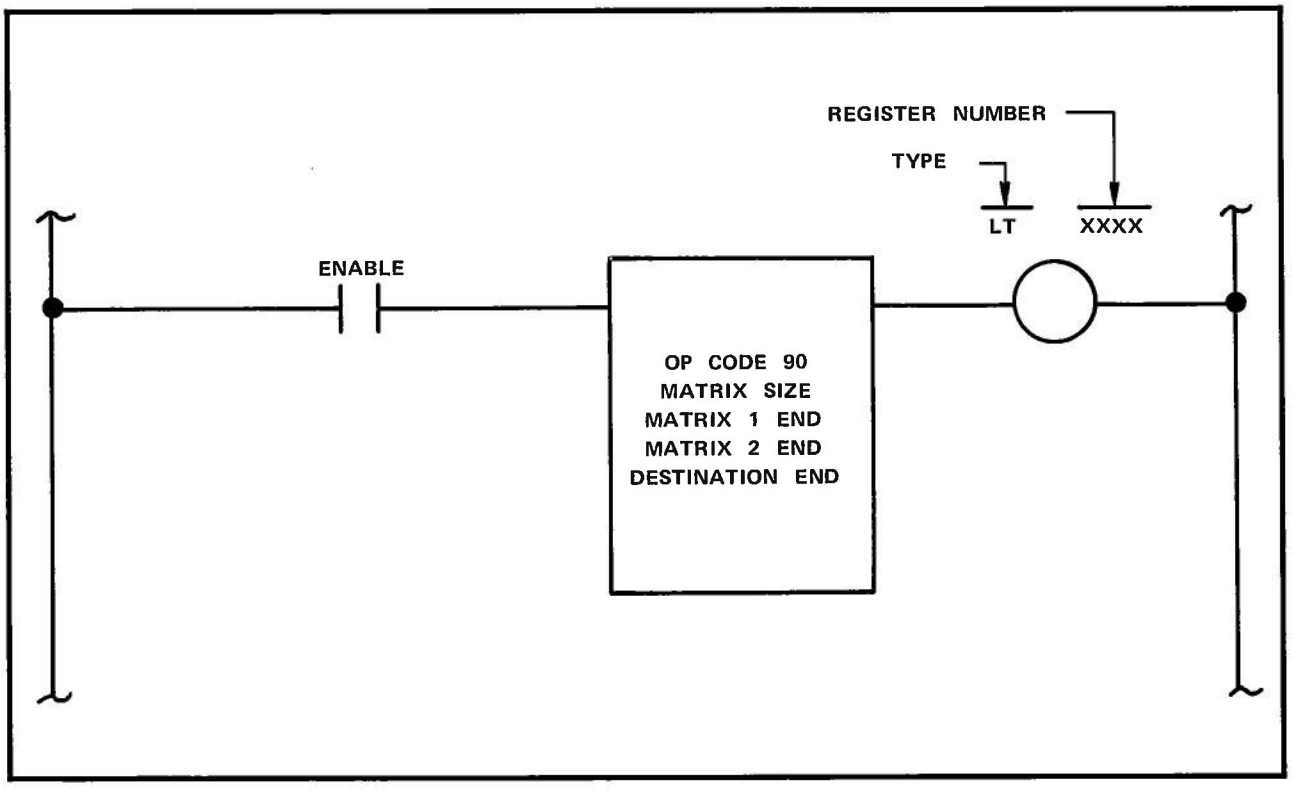


Figure 1. AND Matrix (AM)

AM

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	Limit
HR	⩽ 1792
IR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
OR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
IG	⩽ 16 (PC-700) ⩽ 8 (PC-900 A/B)
OG	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)

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

Note

The highest number holding register is limited by and dependent upon the memory size.

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	Limit
HR	⩽ 1792
OR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
OG	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)

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.
1	The result is placed in the destination matrix. The coil is energized if the result is non-zero.

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

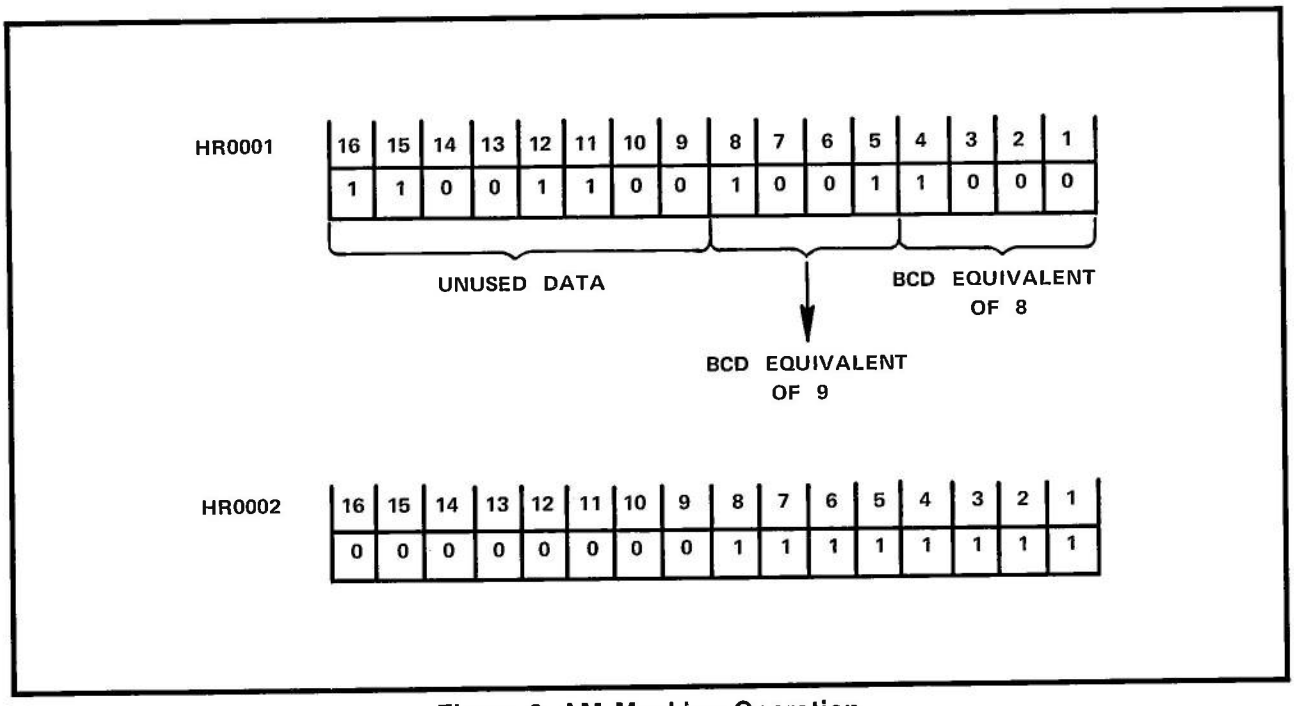


Figure 2. AM Masking Operation

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.

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.

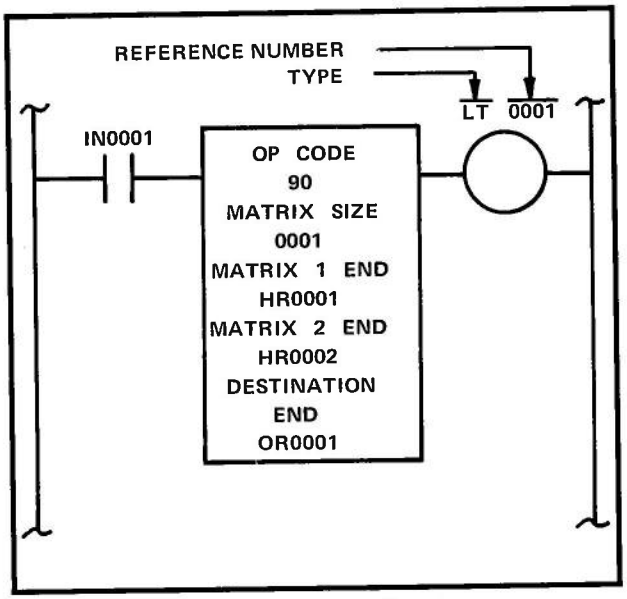


Figure 3. AM Application

MATRIX 1

HR0001

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

HR0002

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

HR0003

48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0

MATRIX 1 END

MATRIX 2

HR0004

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

HR0005

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	1	1	1	1

HR0006

48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

MATRIX 2 END

DESTINATION MATRIX

OG0001

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

OG0002

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	1	0	1	0

OG0003

48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
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

AS — ASCENDING SORT

DESCRIPTION

The Ascending Sort (AS) function arranges the data from a table of registers into ascending order (smallest-to-largest). This function can take more than one scan for completion. 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.) A total of 16 extra holding registers is required for "scratch pad" purposes. The scratch pad is located in the 16 registers following Table 1. The holding registers in Figure 2 are arranged as shown in Figure 3.

SPECIFICATIONS

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.

OP CODE 58

The Op Code defines the Literal (LT) as the AS function.

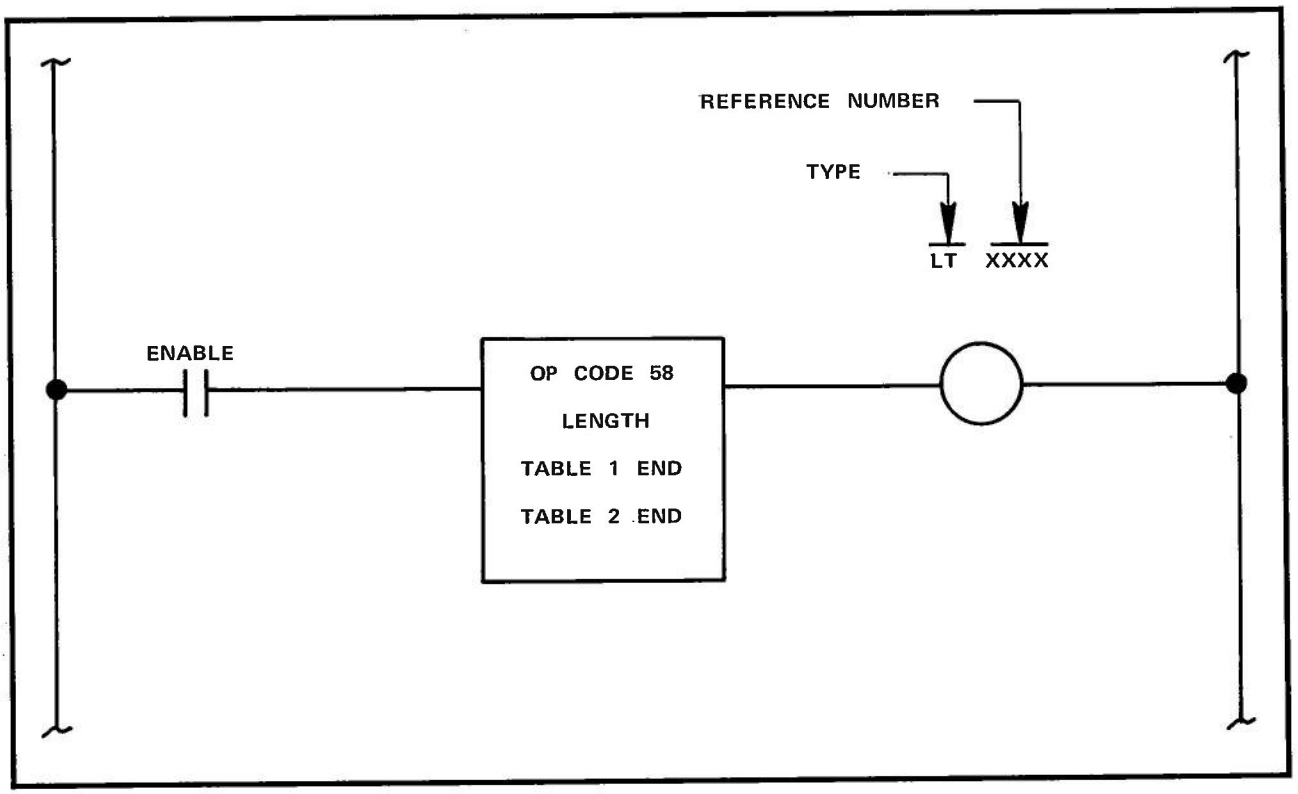


Figure 1. Ascending Sort (AS)

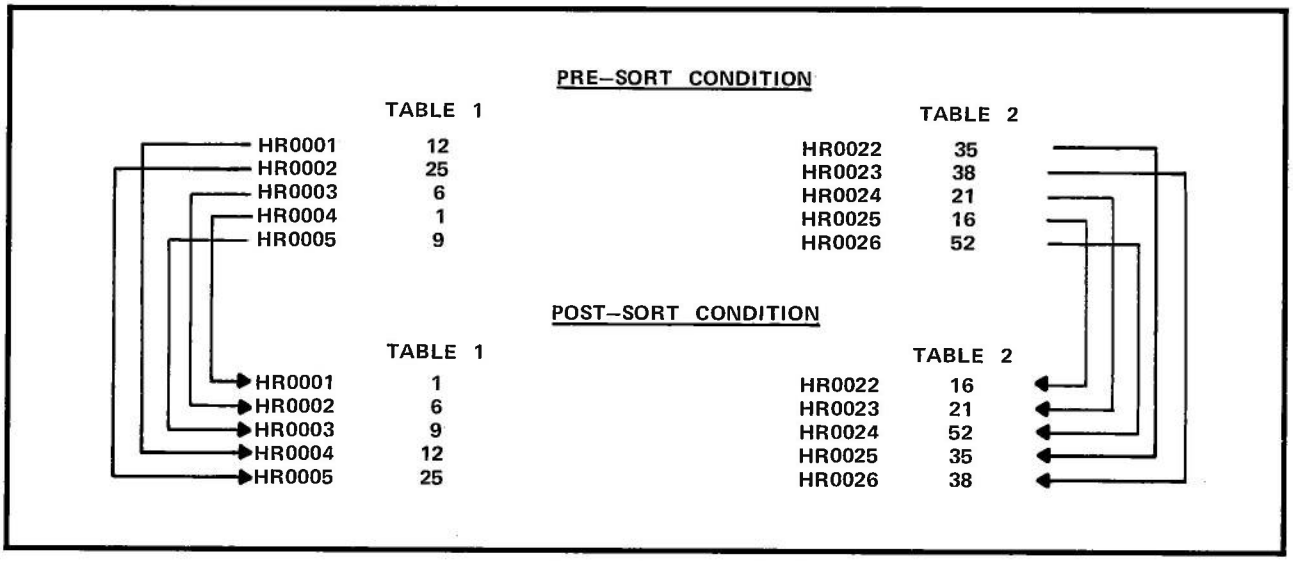


Figure 2. Sorted Holding Registers

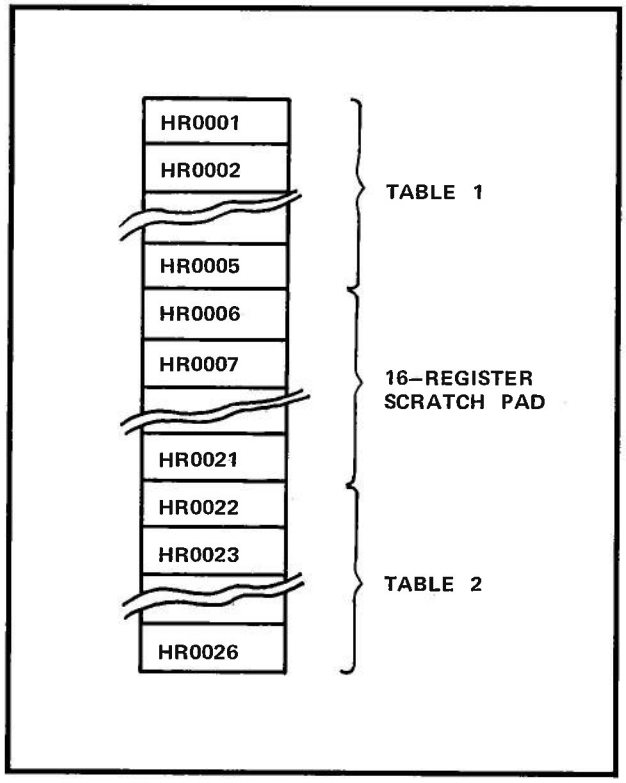


Figure 3. Holding Registers Arrangement

Note

When software changes allow, LT becomes AS.

LENGTH

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

TABLE 1 END

The Table 1 End specifies the last holding register in Table 1 to be sorted, plus 16 holding registers for "scratch pad" use.

TABLE 2 END

The Table 2 End specifies the last holding register in Table 2.

APPLICATIONS

The AS function is used to assemble data into meaningful order. Three machines are equipped with a fault closing and a fault register. The machines identify themselves as part of the fault register output. See Figure 4.

When a fault occurs, it is placed in a register with a time tag placed in a companion register. Since the three machines will develop time/fault tables that are not in time order for the entire installation, the AS function is used to order the data according to time. Figure 5 illustrates the required program without the 24-hour clock feature.

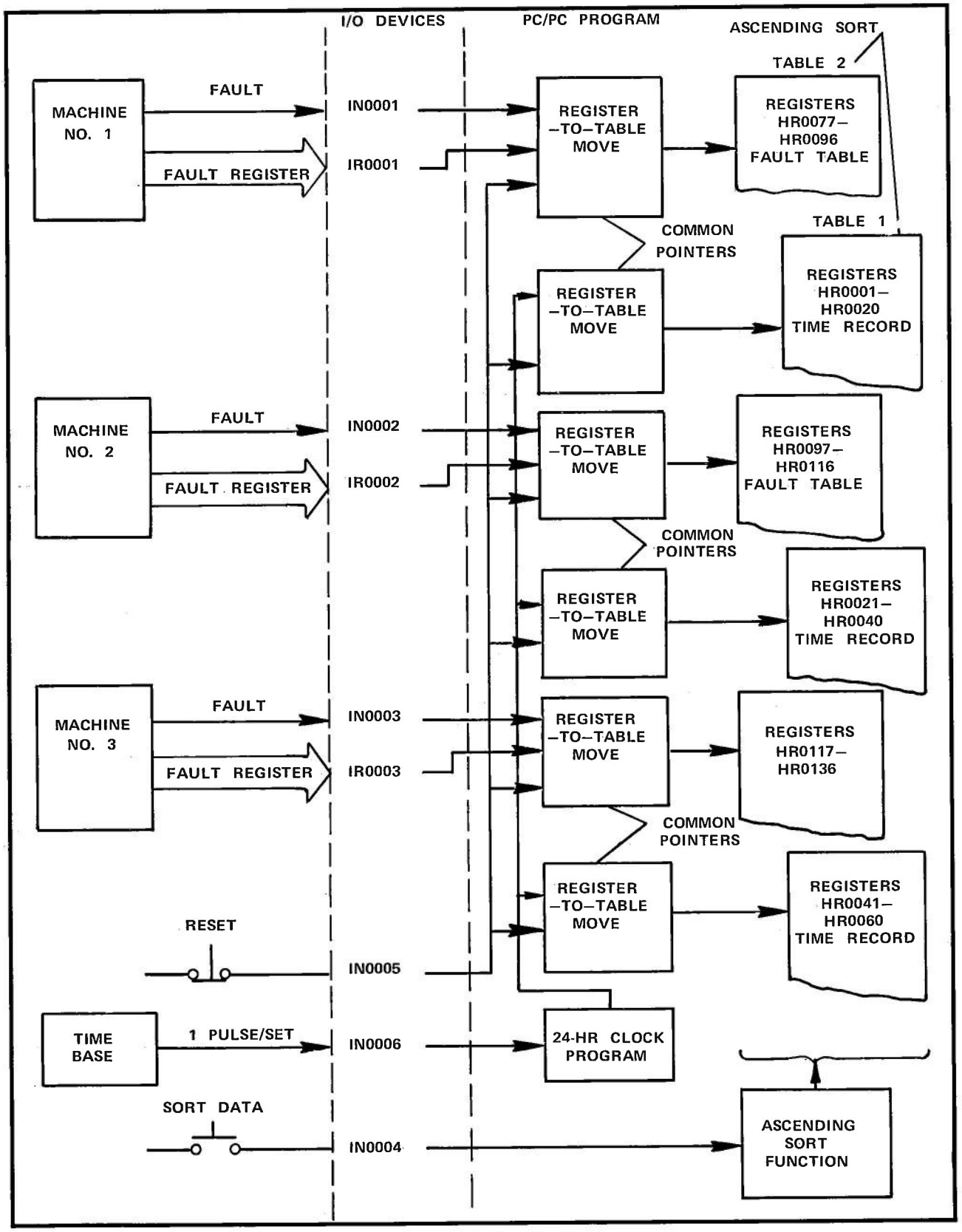


Figure 4. AS Application

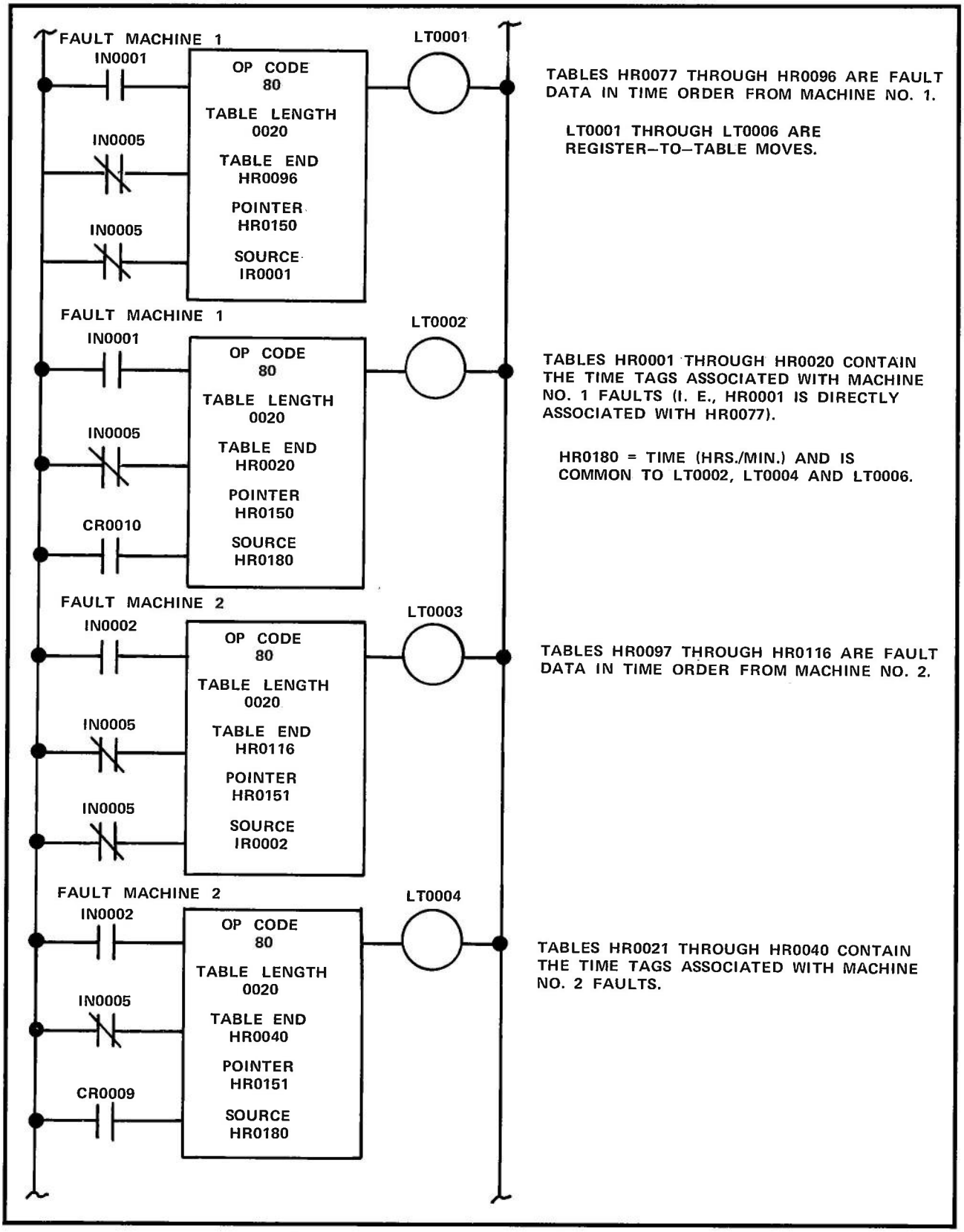


Figure 5a. AS Fault Application Program

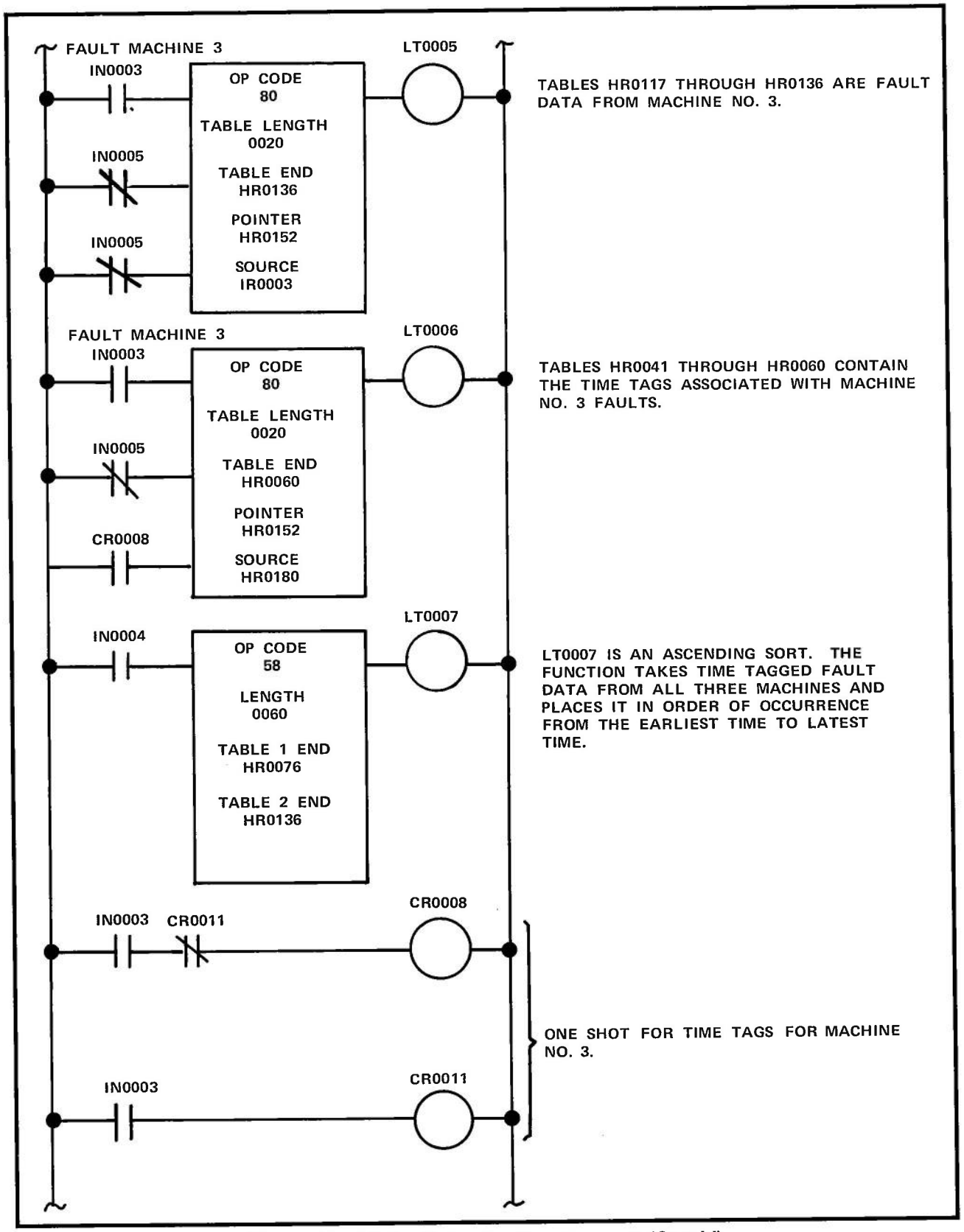


Figure 5b. AS Fault Application Program (Cont'd)

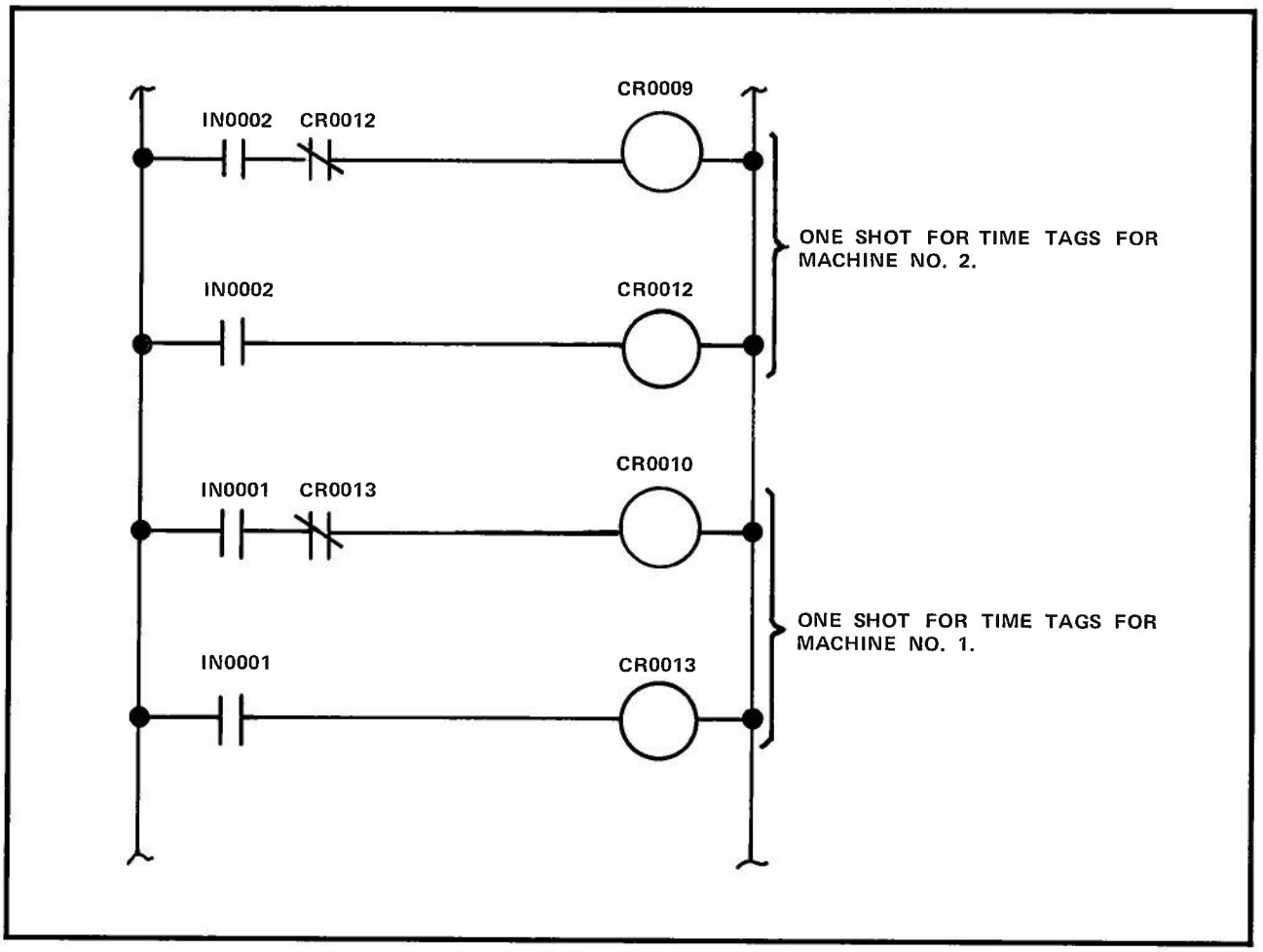
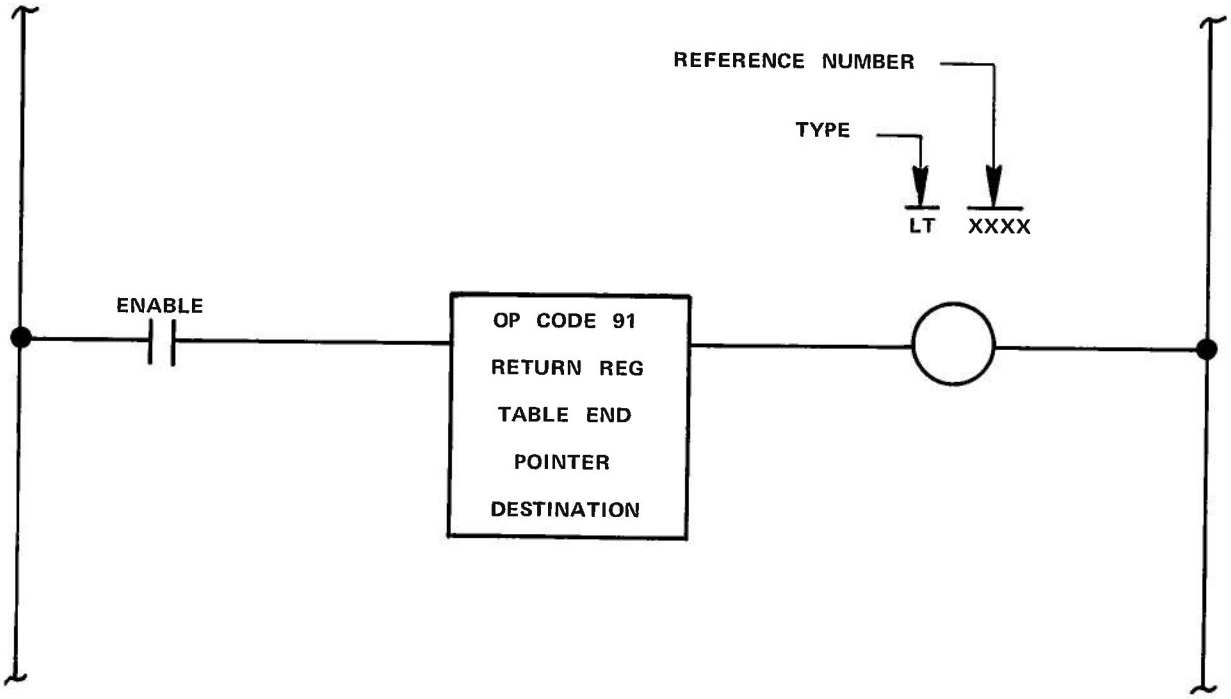


Figure 5c. AS Fault Application Program (Cont'd)



AT — ASCII TRANSMIT

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 ASCII message is stored in holding registers to be transmitted when this function is enabled. The ASCII codes are stored two per holding register. The upper byte (i.e., upper half of the holding register) is transmitted first. See Table 1.

Tables 2 and 3 show the control characters and special codes that are used to formulate messages.

Message information is stored in the format shown in Figure 2. The processor sends a maximum of six 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 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.

Note

ASCII characters above 7F are not valid. When they are encountered, transmission is terminated.

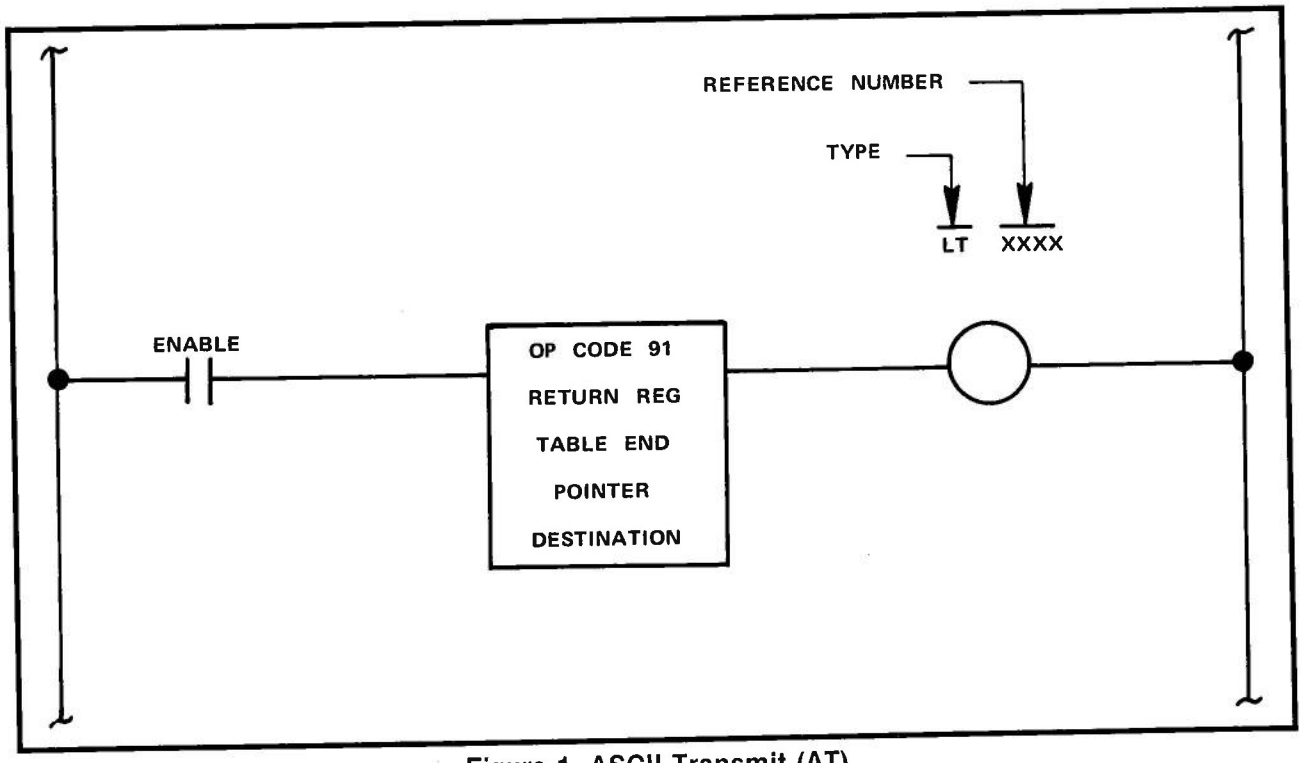


Figure 1. ASCII Transmit (AT)

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
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

TABLE 1. ASCII CODE CONVERSIONS (Cont'd)

Hexadecimal	Decimal	Octal	Binary	Character	Description
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
56	86	126	1010110	V	Letter V
57	87	127	1010111	W	Letter W
58	88	130	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

TABLE 1. ASCII CODE CONVERSIONS (Cont'd)

Hexadecimal	Decimal	Octal	Binary	Character	Description
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)

SPECIFICATIONS

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.

TABLE 2. CONTROL CHARACTERS

Hex	Mnemonic	Meaning
00	NUL	Null
01	SOH	Start of heading (CC)
02	STX	Start of text (CC)
03	ETX	End of text (CC)
04	EOT	End of transmission (CC)
05	ENQ	Enquiry (CC)
06	ACK	Acknowledge (CC)
07	BEL	Bell
08	BS	Backspace (FE)
09	HT	Horizontal tabulation (FE)
0A	LF	Line feed (FE)
0B	VT	Vertical tabulation (FE)
0C	FF	Form feed (FE)
0D	CR	Carriage return (FE)
0E	SO	Shift out
0F	SI	Shift in
10	DLE	Data link escape (CC)
11	DC1	Device Control 1
12	DC2	Device Control 2
13	DC3	Device Control 3
14	DC4	Device Control 4
15	NAK	Negative acknowledge (CC)
16	SYN	Synchronous idle (CC)
17	ETB	End of transmission block (CC)
18	CAN	Cancel
19	EM	End of medium
1A	SUB	Substitute
1B	ESC	Escape
1C	FS	File separator (IS)
1D	GS	Group separator (IS)
1E	RS	Record separator (IS)
1F	US	Unit separator (IS)
7F	DEL	Delete

OP CODE 91

The Op Code defines the Literal (LT) as the AT function.

RETURN REGISTER

The return register defines the location of the holding register reference number that specifies the location for return from a 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)

TABLE END

The table end defines the highest holding register reference number containing message data. This is always a holding register.

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)

Note

The pointer must be set to the beginning of the message before executing the function. No transmission will occur with the pointer equal to zero.

DESTINATION

The destination is a constant that specifies which port the processor will use to transmit message data. The destination for the PC-900B must be set to one. The destination for the PC-700 with two ports must be set as follows:

1 = Program Loader Port
2 = Computer Interface Port

TABLE 3. SPECIAL OPERATION CODES

Code	Meaning
B0	Return to the holding register specified by the contents of the return register. If the contents equal zero, B000 is ignored (i.e., no operation).
B1	The ASCII character in the low byte of the register is to be repeated N times (where N < 256 and is stored in the following register).
B2	The next holding register is a pointer 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.
B3	The next holding register is a pointer to another holding register whose content is a number. The number is to be converted into a five-digit BCD number between −32 K and +32 K, depending on the most-significant bit. The sign and the five digits are converted into ASCII code and sent to the proper communications port.
B4	The next holding register is a pointer 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.
B5	The next holding register is a pointer 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.
B6	The next holding register is a pointer 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.
B7	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.
B8	Message End. Reset the return register to zero.

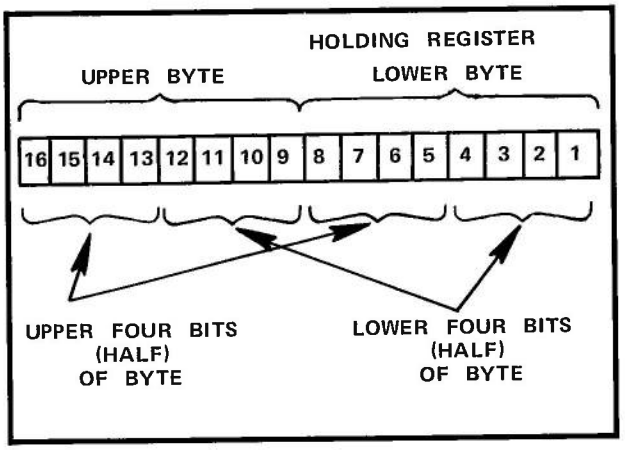


Figure 2. Character Storage Format

APPLICATION

Any desired messages can be formatted and transmitted to suitable RS-232C compatible devices. Control commands are ignored because they are dependent upon the device used.

The PC-700 and PC-900, as shipped, are set to operate at 9600 baud with a 12-bit data frame (one start bit, eight data bits, one odd parity bit, and two stop bits). Set the printer to match this baud rate and data format.

The controller transmits data on Pin 2 with Pin 7, the signal ground or common.

Pin 20, data terminal ready, is set high by the PC any time power is on.

Pin 6, data set ready, must be pulled high by the printer or by jumpering to Pin 20 of the PC.

When the AT function is enabled, Pin 4 is set high. If Pins 5 and 6 are high, data is transmitted.

Connect Pin 5 to the "not full buffer" signal from the printer. Optionally, jumper Pin 4 and Pin 5 on the PC to function without a "handshake".

Table 4 shows the message **STATION 15 IS INACTIVE** coded in hexadecimal. The programming for the **STATION 15 IS INACTIVE** message is shown in Figure 3.

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.

TABLE 4. ASCII MESSAGE TABLE

Holding Register	Hexadecimal Code	Message
HR0200	5354	(ST)
HR0201	4154	(AT)
HR0202	494F	(IO)
HR0203	4E20	(N)
HR0204	3135	(15)
HR0205	2049	(I)
HR0206	5320	(S)
HR0207	494E	(IN)
HR0208	4143	(AC)
HR0209	5449	(TI)
HR0210	5645	(VE)
HR0211	0D0A	Carriage Return
HR0212	B800	End of Message

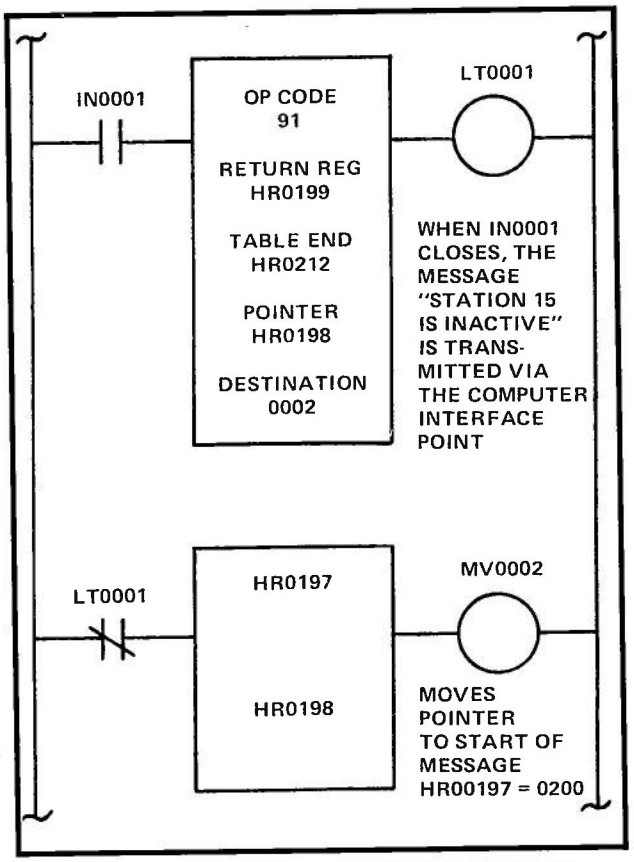


Figure 3. AT Example — STATION 15 IS INACTIVE Message

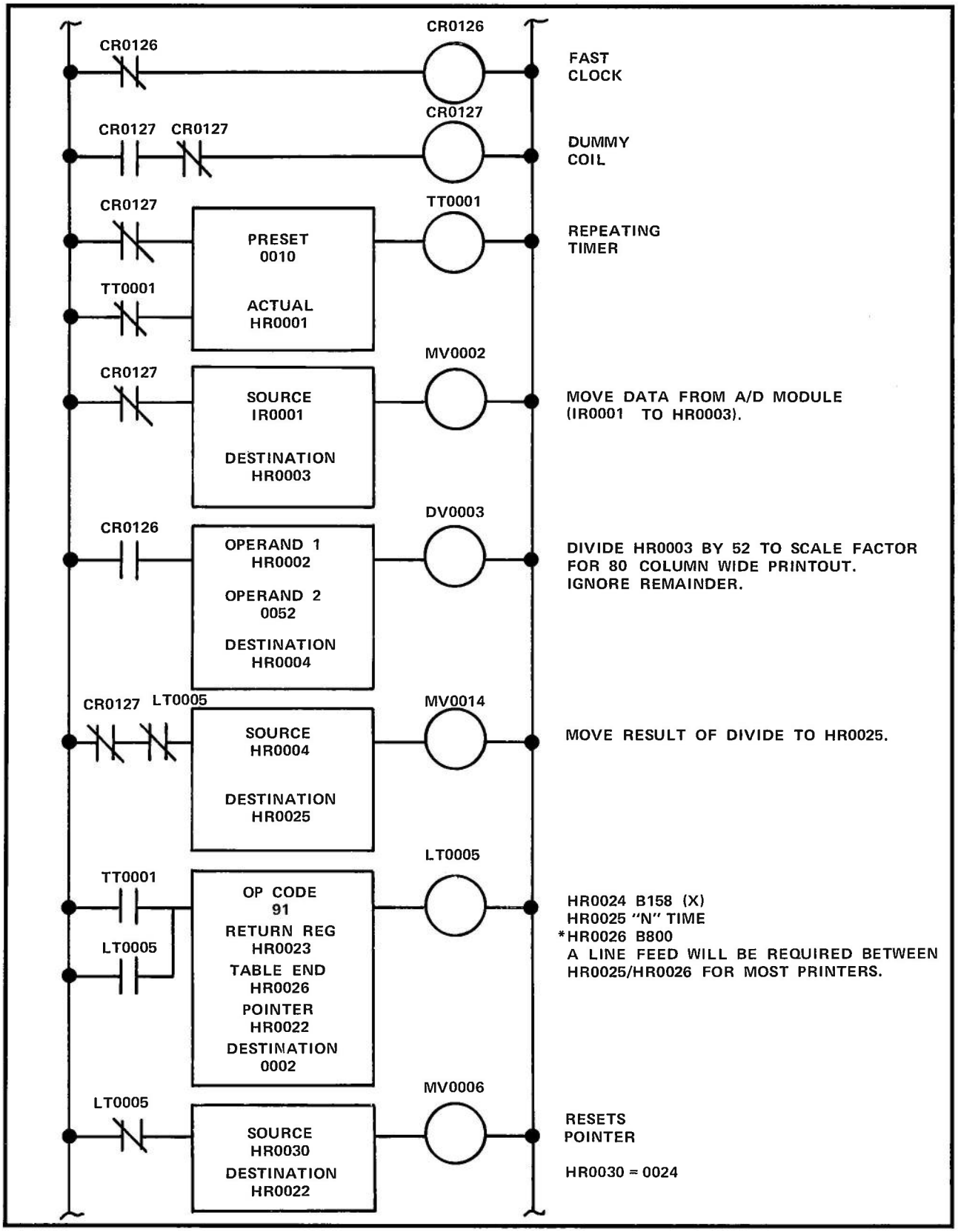


Figure 4. AT Program

BD/DB — CONVERSIONS

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.

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.

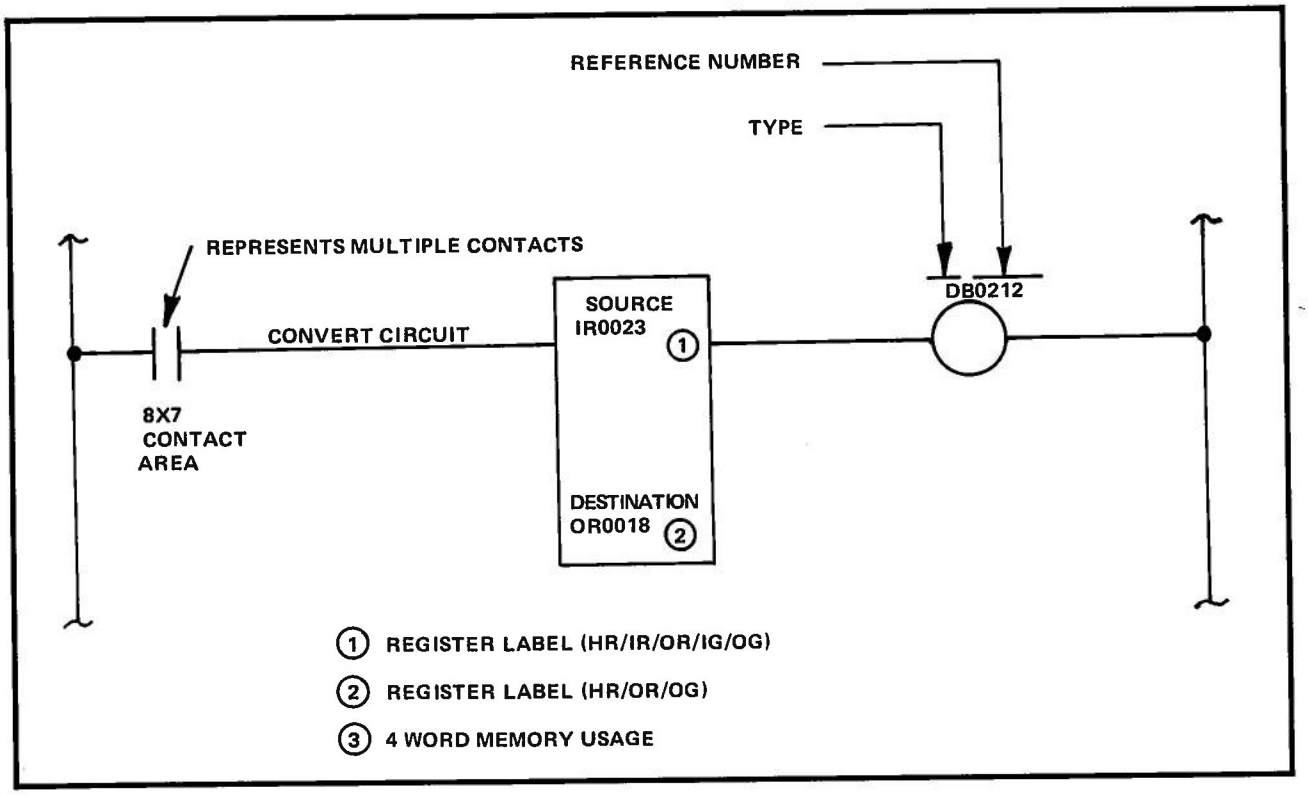


Figure 1. Binary to Decimal (BD)/Decimal to Binary (DB)

BD/DB

The conversion for the BD and DB functions is made when the convert circuit changes from non-conducting to 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)

DESTINATION

The destination is the location where the conversion results are placed. This value is held in a specified register or group:

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

BD/DB TRUTH TABLE

See Table 1.

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.

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 Figure 2.

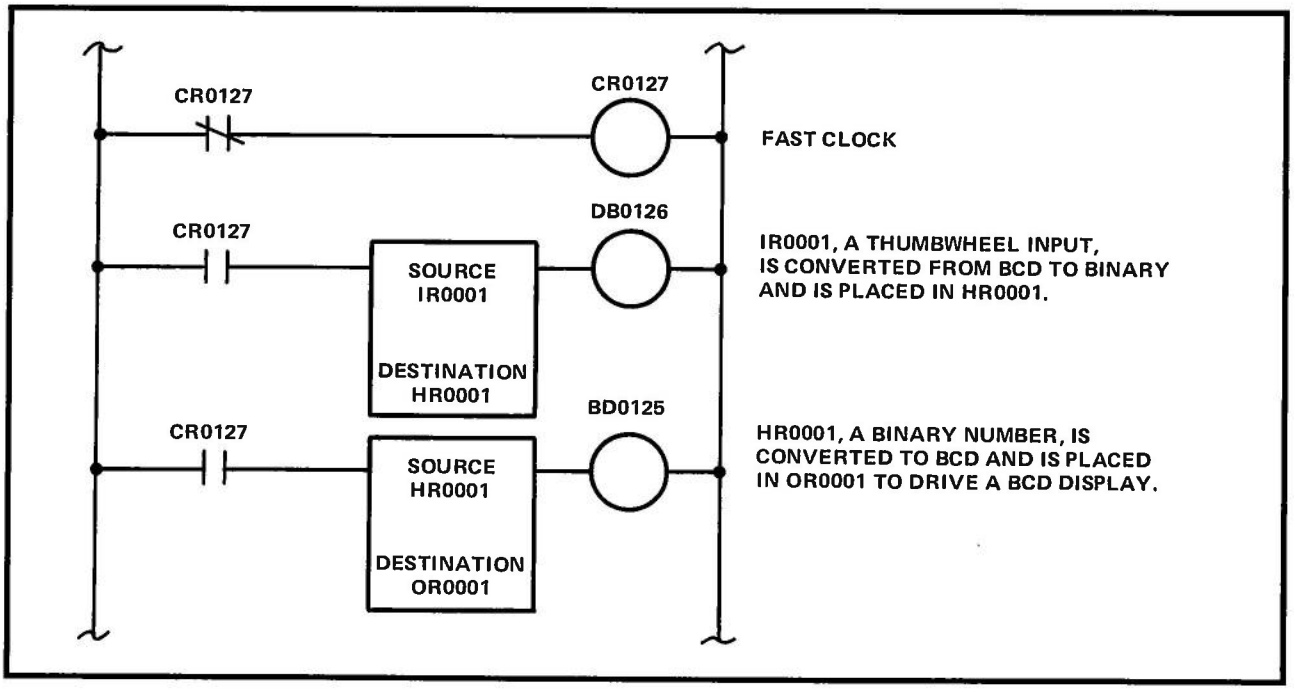


Figure 2. Number Conversions

BF — BIT FOLLOW

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.

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) could also be Bit 48 of HR0001. This feature is limited to 2048 bits for Holding Registers (HRs) and as follows for Output Registers (ORs):

PC-700 = 512 bits
PC-900A = 128 bits
PC-900B = 256 bits

SPECIFICATIONS

BIT NUMBER

The bit number specifies the bit controlled by the coil.

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).

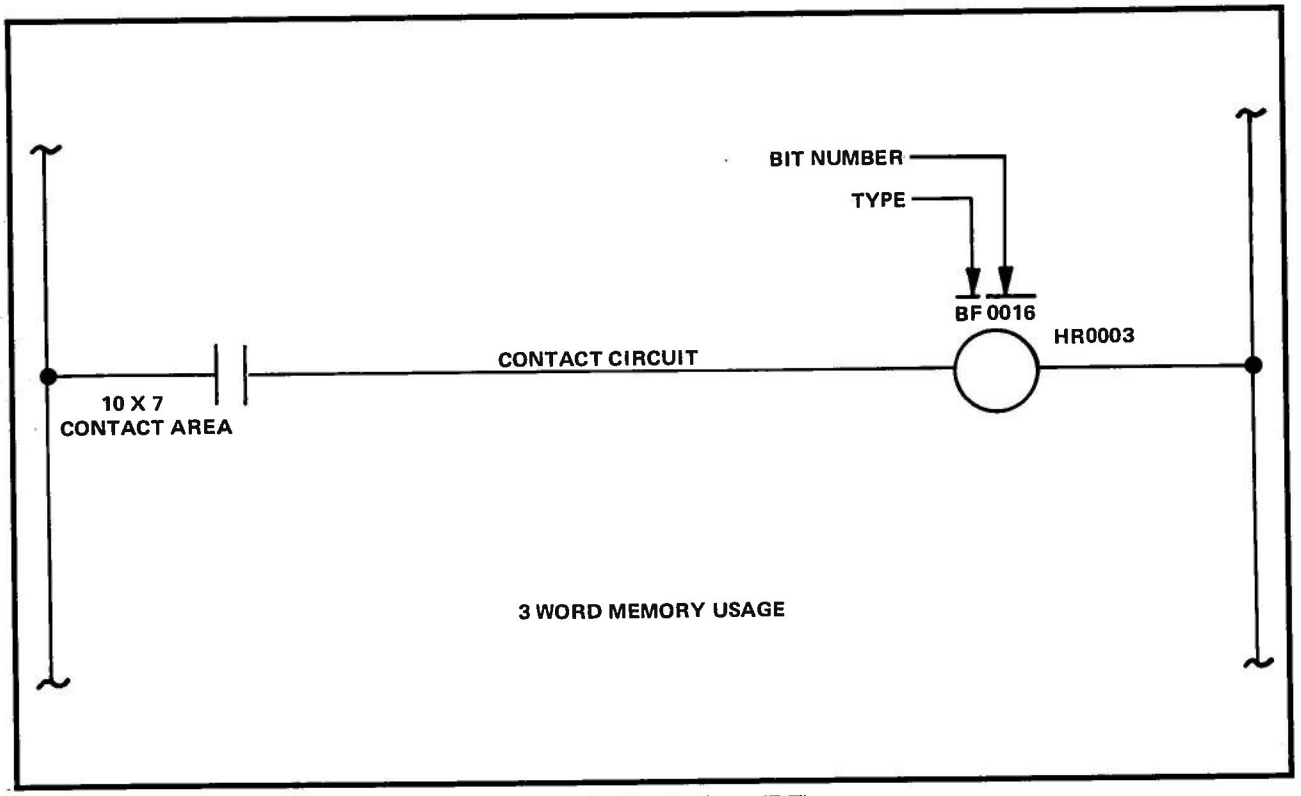


Figure 1. Bit Follow (BF)

BF TRUTH TABLE

See Table 1.

TABLE 1. BF TRUTH TABLE

Contact Circuit	Result
0 (Open)	Specified bit is cleared.
1 (Closed)	Specified bit is set to one.

APPLICATIONS

The BF function can be used to extend the number of discrete outputs available beyond that of the processor in use. Figure 2 is an example of extending the discrete I/O in a PC-700. Additionally, the BF function can be used to extend register inputs beyond the limit of the particular processor in use, as shown in Figure 3.

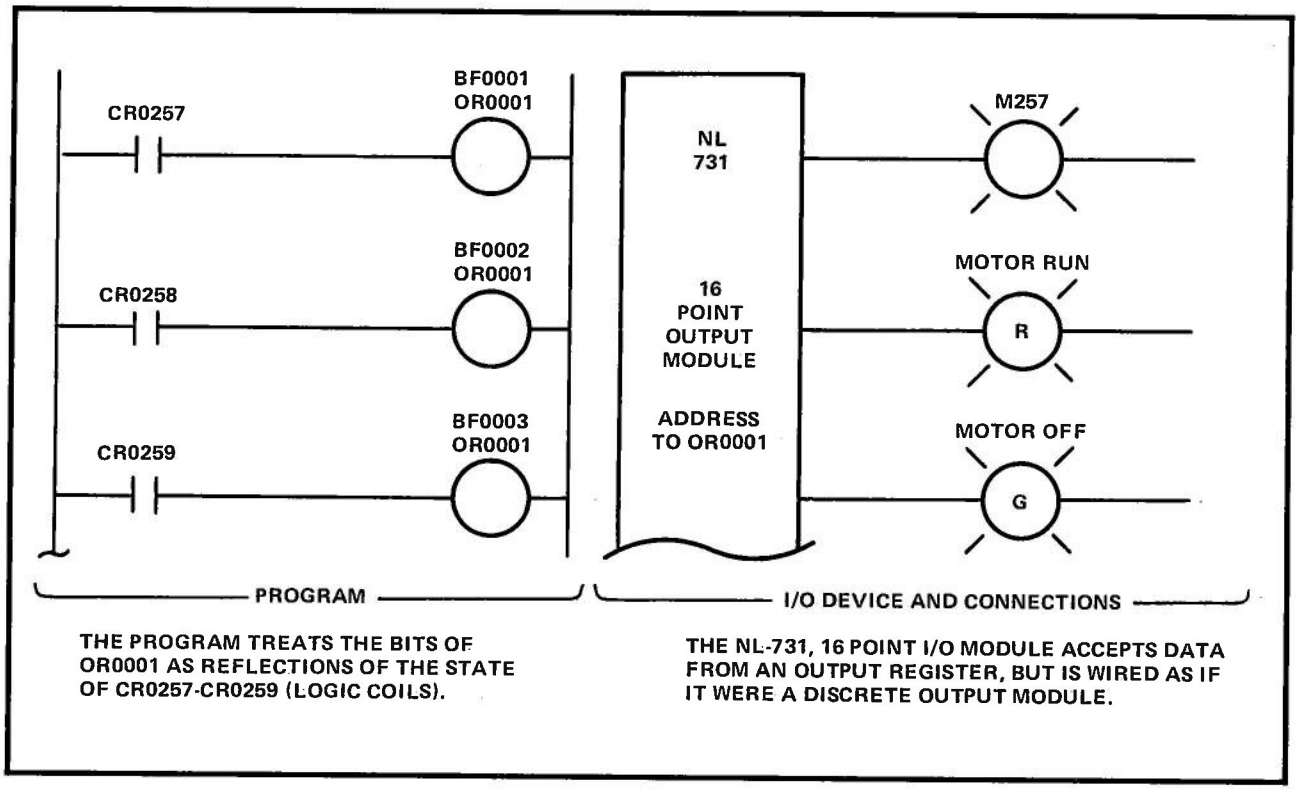


Figure 2. Extending Discrete I/O

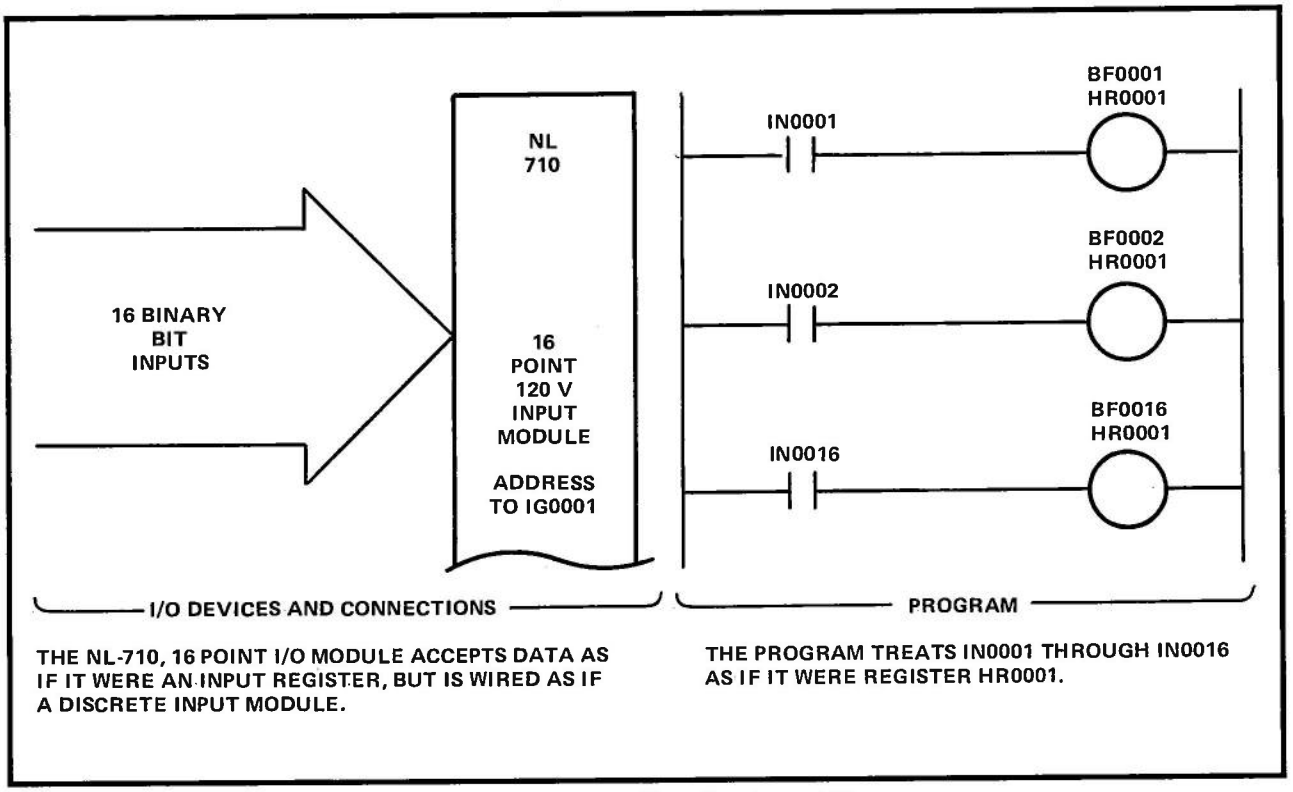


Figure 3. Extending Register I/O

BO — BIT OPERATE

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.

SPECIFICATIONS

OP CODE 61

The Op Code defines the Literal (LT) as a BO function.

Note

When software changes allow, LT becomes 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).

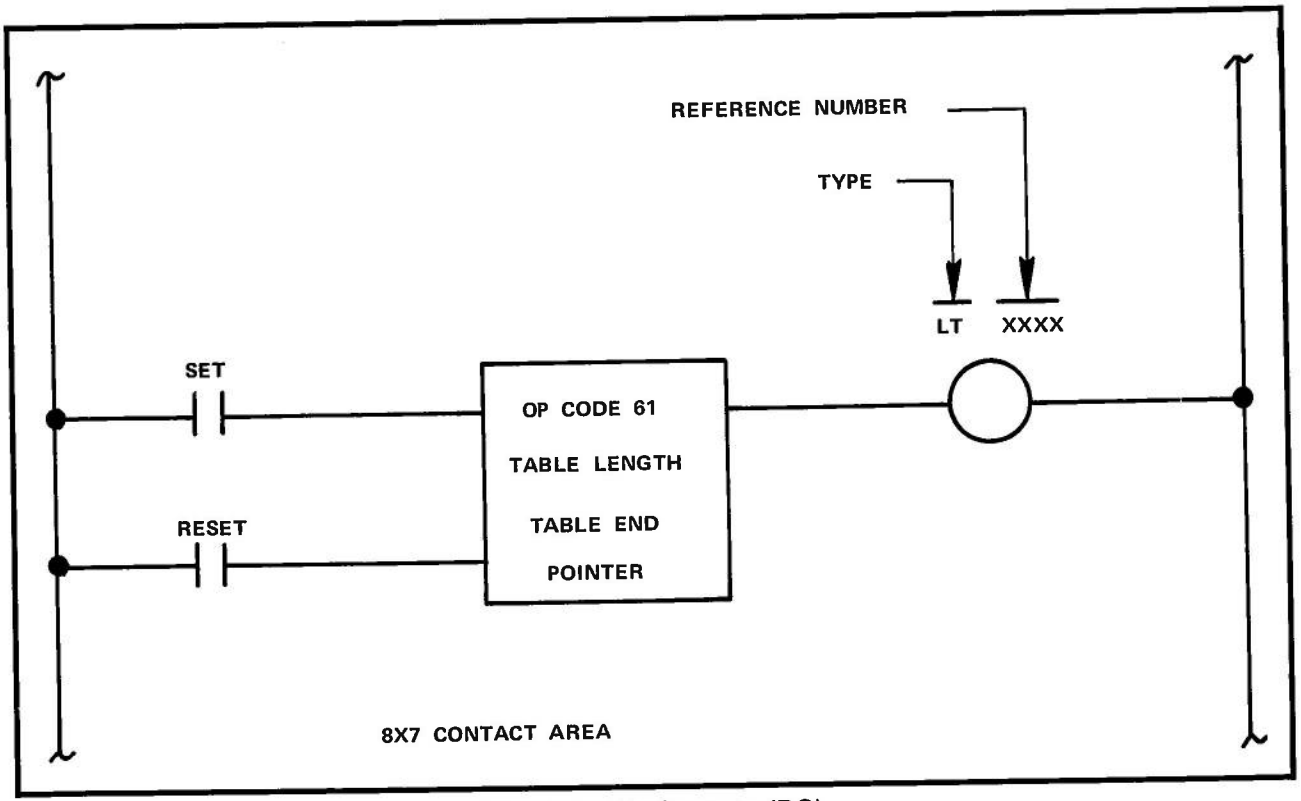


Figure 1. Bit Operate (BO)

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 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.

TABLE 2. TABLE LENGTH AND TABLE END LIMITS

Type	Limit
HR	⩽ 1792
IR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
OR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
IG	⩽ 16 (PC-700) ⩽ 8 (PC-900A/B)
OG	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)

POINTER

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)

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.

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 256 outputs. The table represents outputs CR0001 through CR0256, the total available output capacity of the PC-700. 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 continuously converted from BCD to binary. The state of the input whose number is present in the thumbwheel set (1 through 256) is reflected in the state of the LT0001 coil.

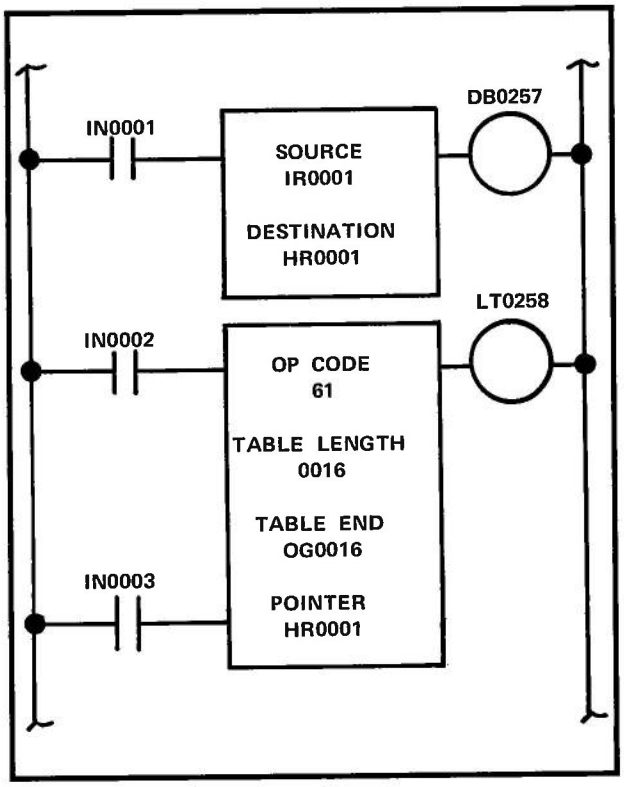


Figure 2. BO Application Program

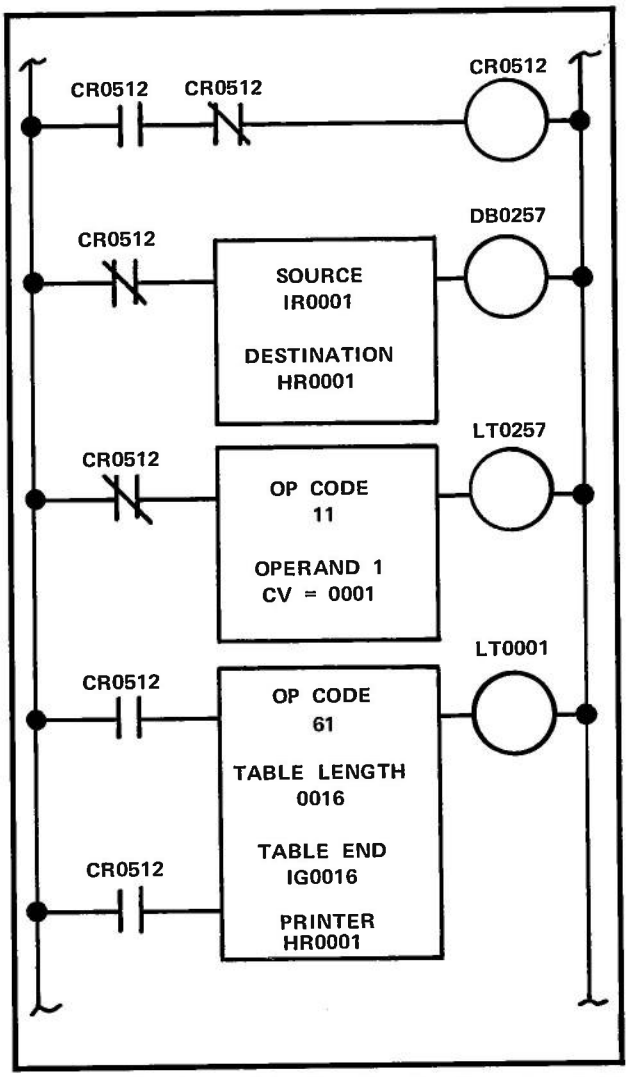


Figure 3. BO Input Monitor Application Program

BS/BC — LATCHES

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.

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)

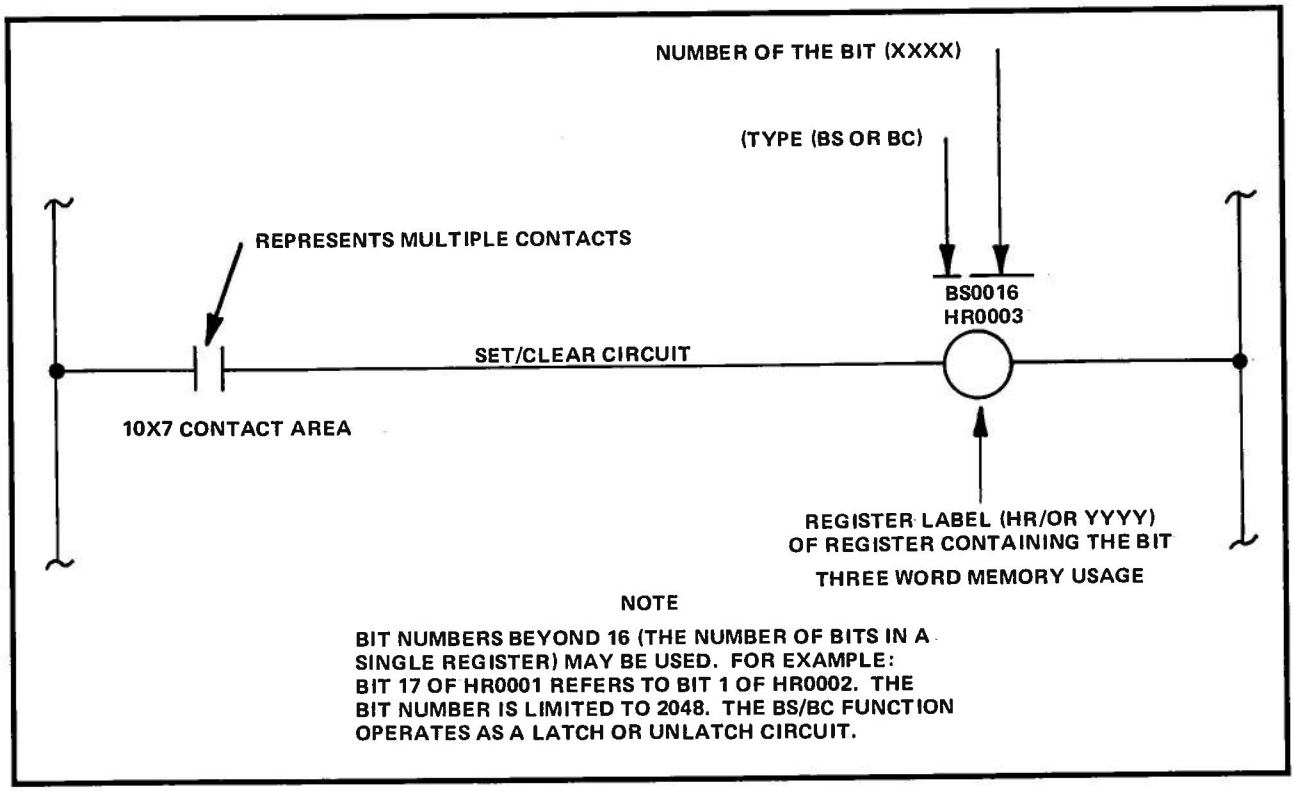


Figure 1. Bit Set (BS)/Bit Clear (BC)

BS/BC

TYPE

The type designates the type of function to be performed:

- BS
- BC

NUMBER OF BIT

Number of bit specifies the bit number 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 1 of HR0500 is set (latched) only if IN0001 is present and Bit 1 is not set to one. This figure illistrates 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).

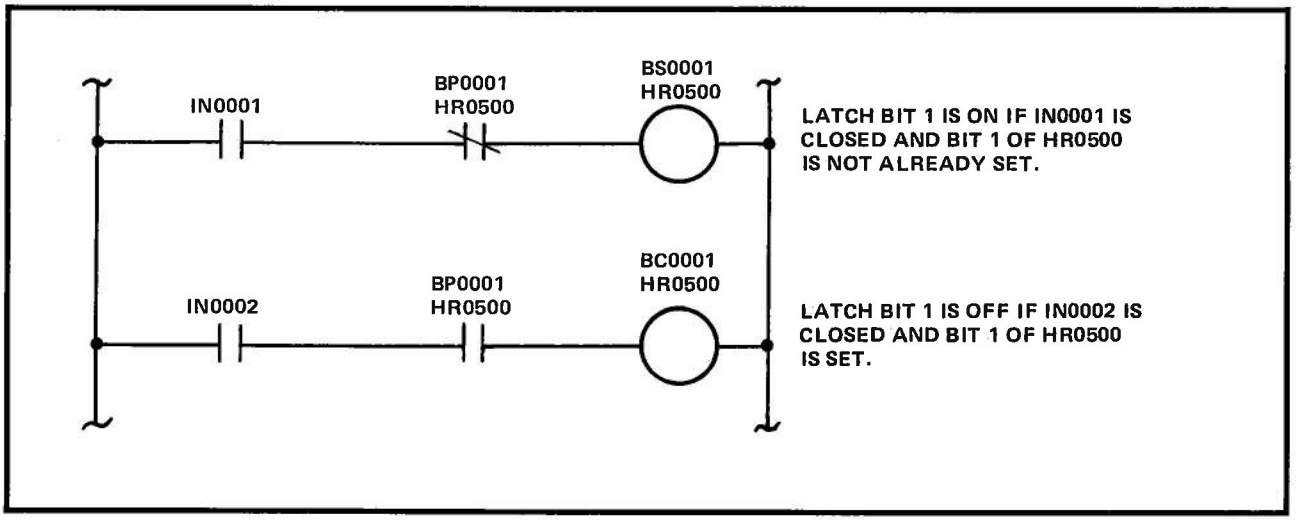


Figure 2. Bit Latch Application

BT — BLOCK TRANSFER

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.

SPECIFICATIONS

OP CODE 60

The Op Code defines the Literal (LT) as the BT function.

Note

When software changes allow, LT becomes BT.

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 number holding register is dependent upon memory size.

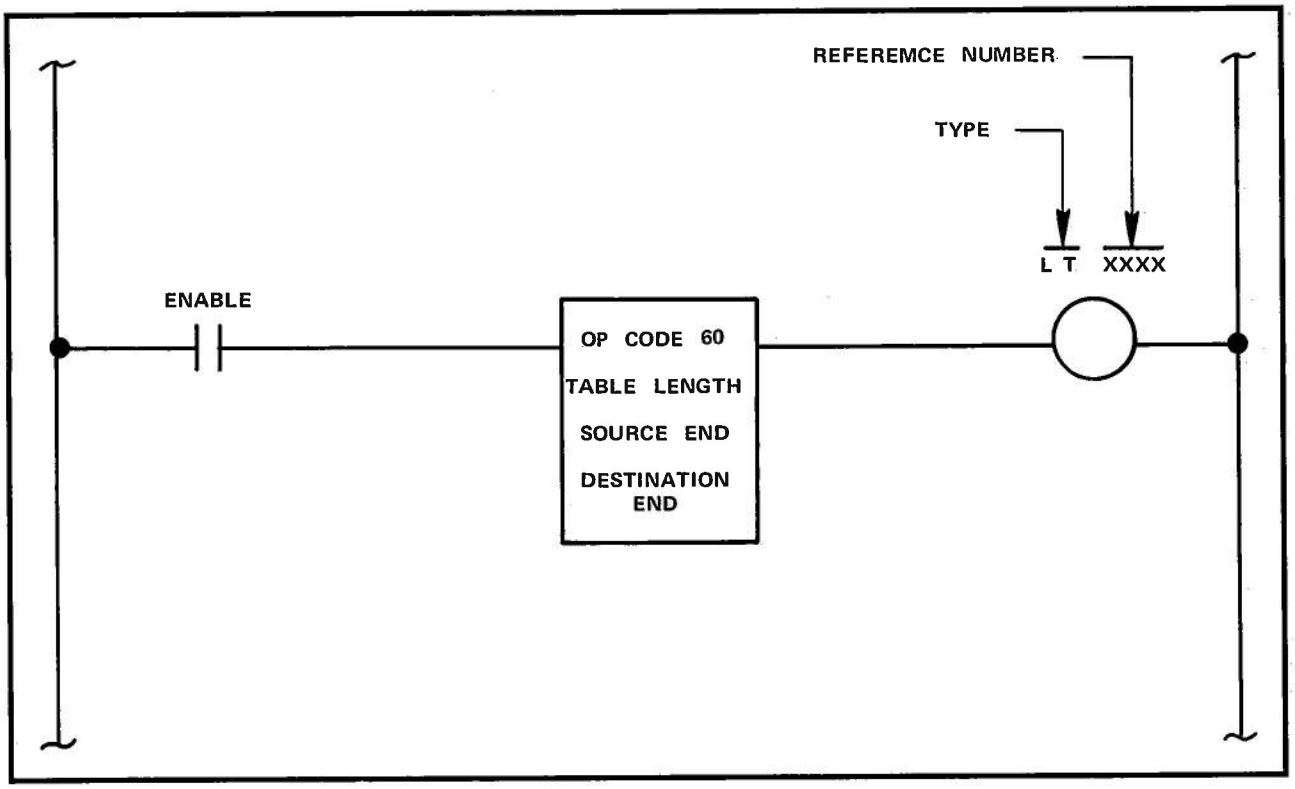


Figure 1. Block Transfer (BT)

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.

TABLE 1. TABLE LENGTH/SOURCE END LIMITS

Type	Limit
HR	⩽ 1792
IR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
OR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
IG	⩽ 16 (PC-700) ⩽ 8 (PC-900A/B)
OG	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)

DESTINATION 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 2. DESTINATION END LIMITS

Type	Limit
HR	⩽ 1792
OR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
OG	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)

BT TRUTH TABLE

See Table 3.

TABLE 3. BT TRUTH TABLE

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

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.

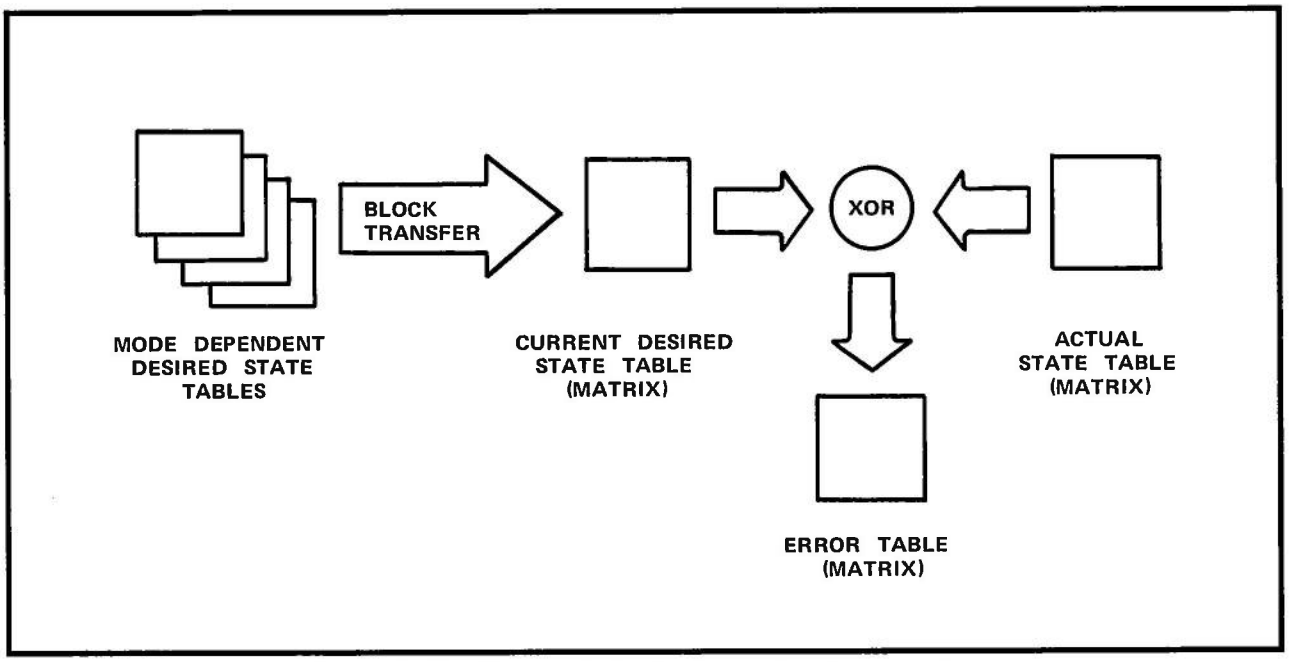


Figure 2. Monitoring Outputs Using the BT Function

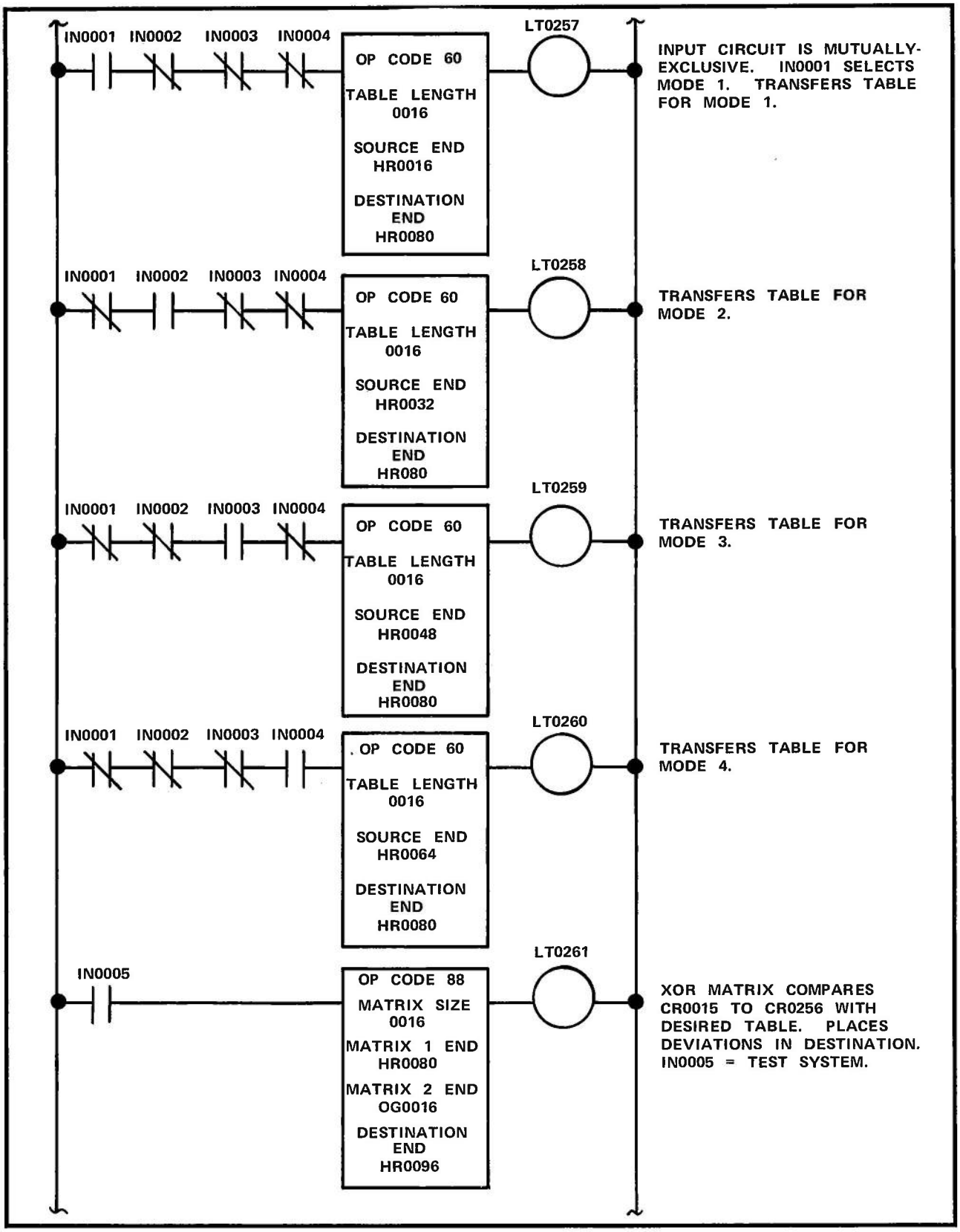


Figure 3. BT Application Program

CG — CONTINUOUS GROUP SELECT

DESCRIPTION

The Continuous Group Select (CG) function allows the user to change a transitional function, or group of transitional functions, to continuous operation. When the step 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.

SPECIFICATIONS

OP CODE 15

The Op Code defines the Literal (LT) as the CG function.

Note

When software changes allow, LT becomes CG.

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) determines which coil operates continuously when enabled. When the operand is an output group, it determines which group of 16 coils operates continuously when enabled. OG0001 consists of Output Coils 1 through 16, OG0002 consists of Output Coils 17 through 32, etc.

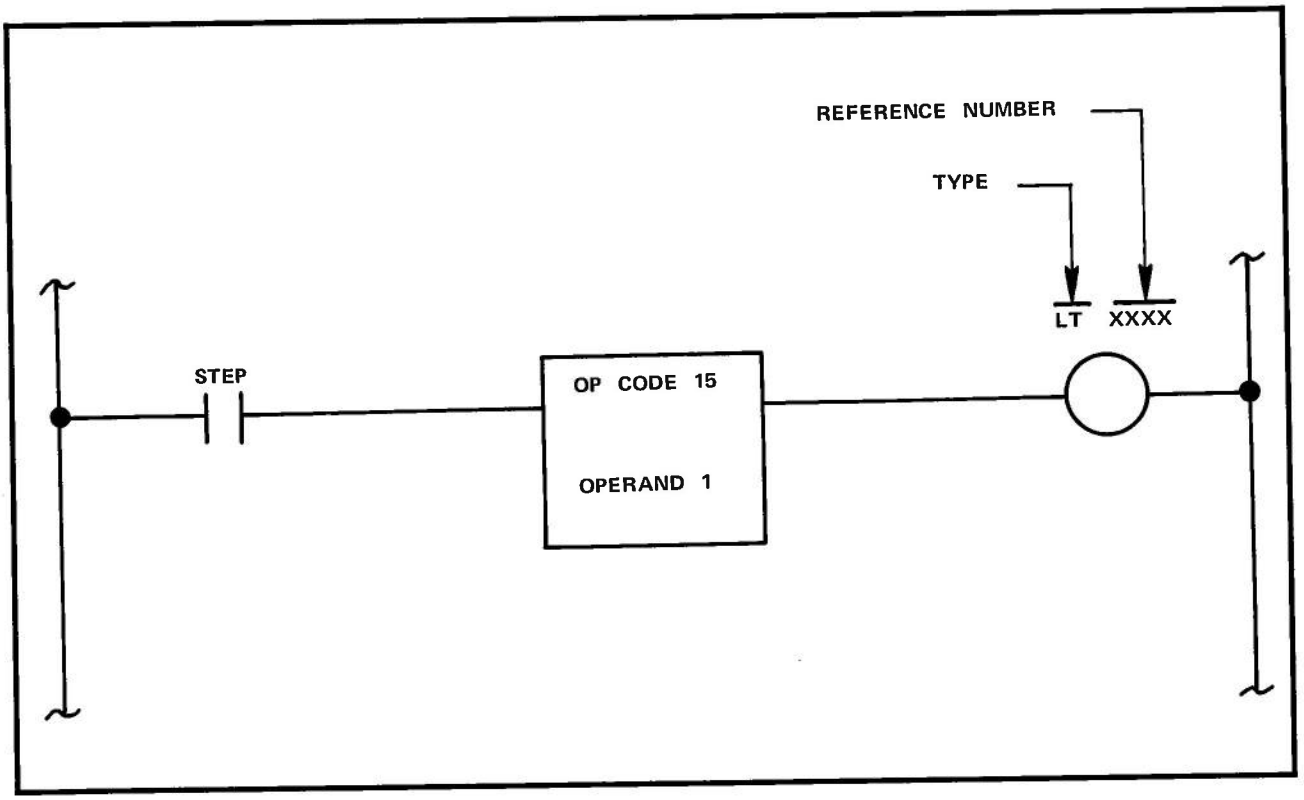


Figure 1. Continuous Group Select (CG)

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.

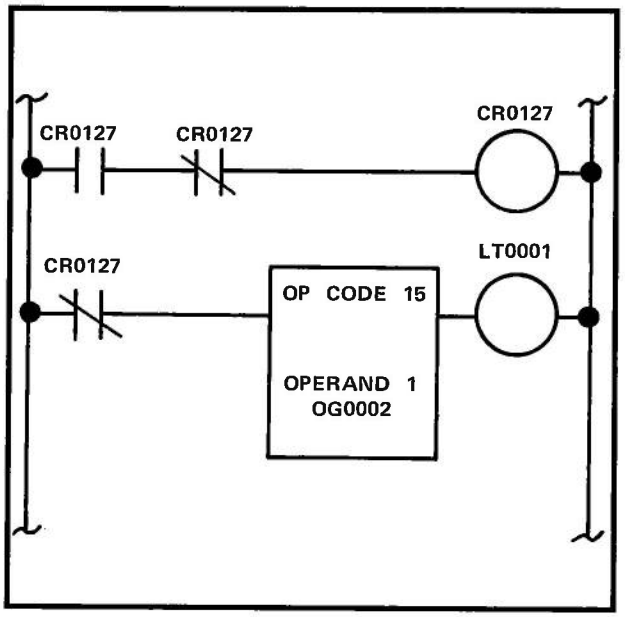


Figure 2. CG Application

CM — COMPLEMENT MATRIX

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.

SPECIFICATIONS

COIL

The coil follows the enable circuit.

OP CODE 57

The Op Code defines the Literal (LT) as the CM function.

Note

When software changes allow, LT becomes CM.

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 number of the holding registers is limited by, and dependent on, memory size.

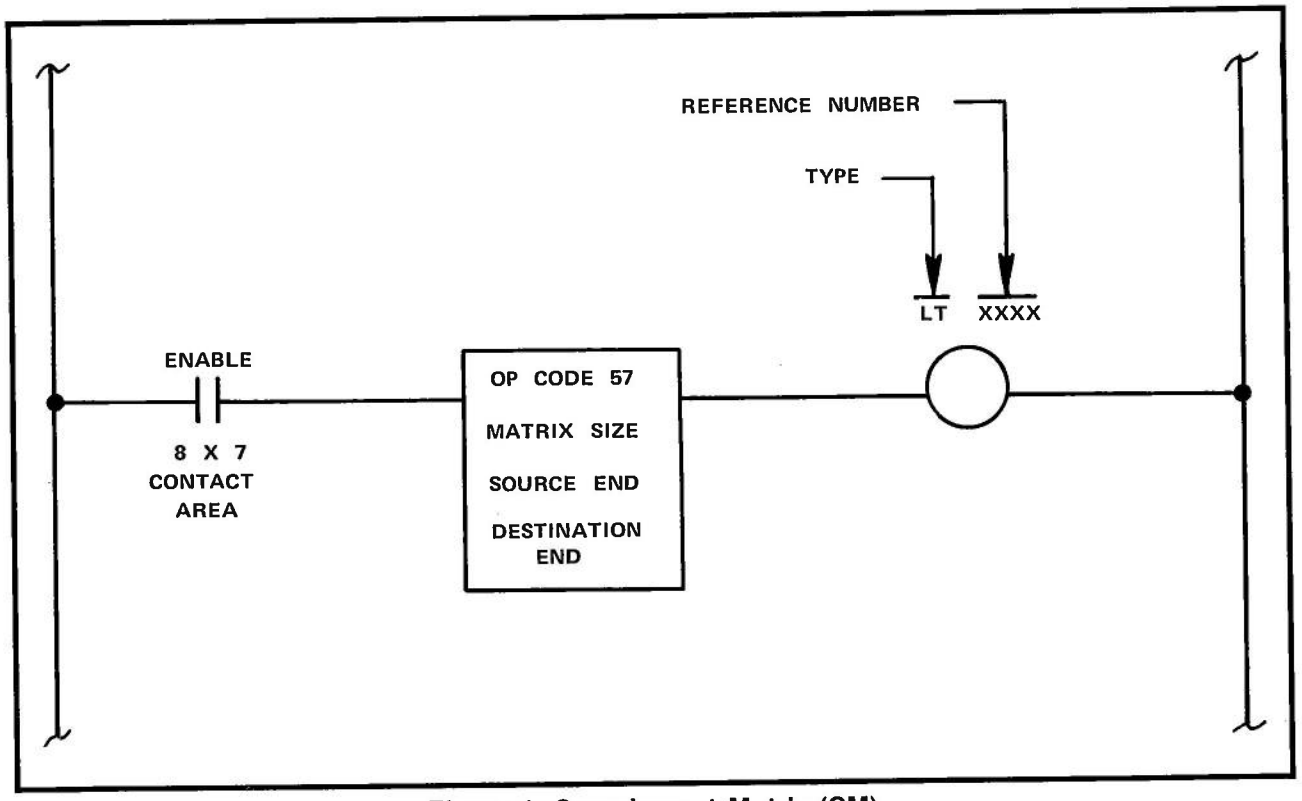


Figure 1. Complement Matrix (CM)

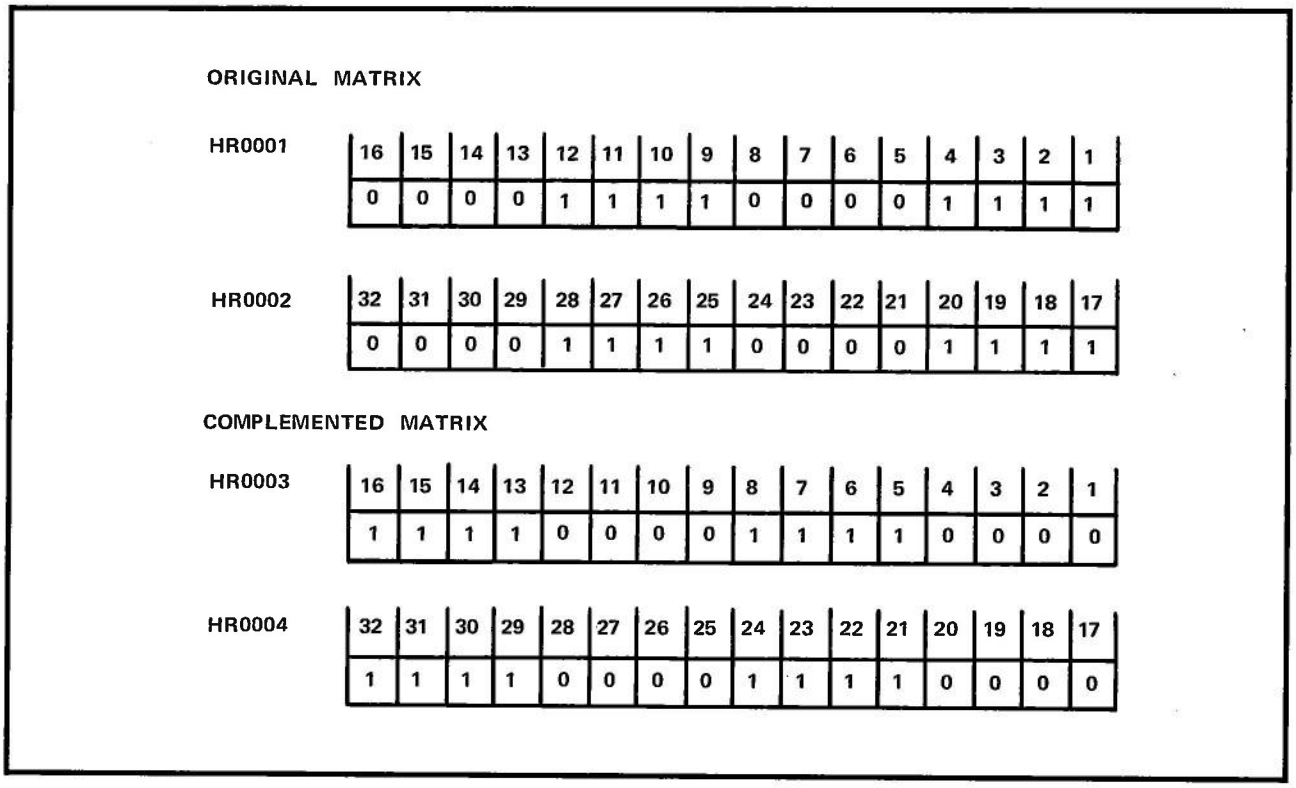


Figure 2. Complementing a Matrix

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.

TABLE 1. SOURCE END/MATRIX SIZE LIMITS

Type	Limit
HR	⩽ 1792
IR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
OR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
IG	⩽ 16 (PC-700) ⩽ 8 (PC-900A/B)
OG	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)

DESTINATION END

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

TABLE 2. DESTINATION END/MATRIX SIZE LIMITS

Type	Limit
HR	⩽ 1792
OR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
OG	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)

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 coil energizes. The complement of the source matrix is in the destination matrix.

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.

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.

ORIGINAL MATRIX

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

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

COMPLEMENTED MATRIX

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

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

Figure 3. CM Application Example

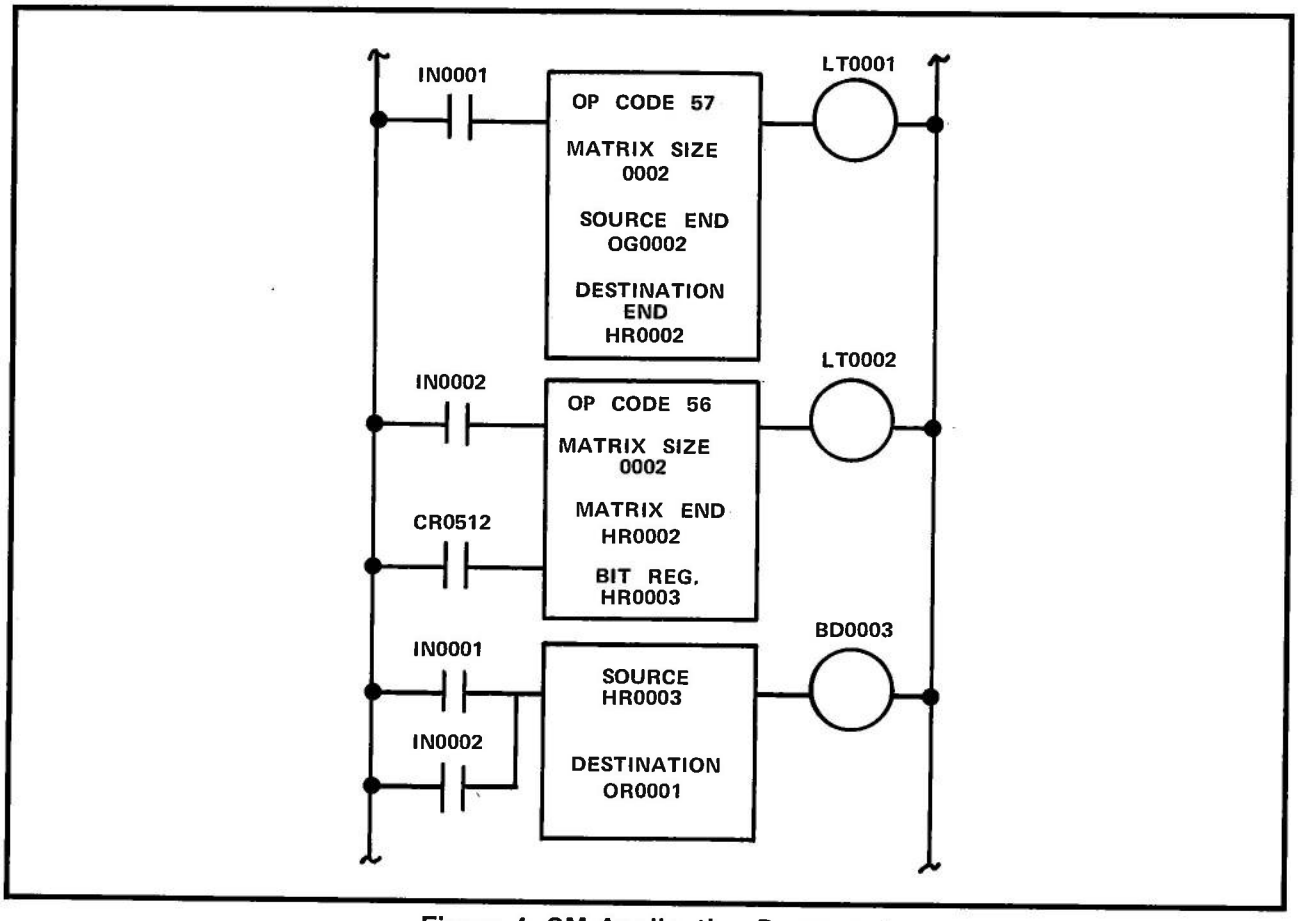


Figure 4. CM Application Program

CR — CONTROL RELAY

DESCRIPTION

Control Relay (CR) coils control contacts and output circuits. Each coil controls all contacts with the same reference number as the coil. Output coils can control output circuits and have reference numbers from one to a limit that depends on memory size. Logic coils only control contacts and have reference numbers limited to the values listed in Table 1. CR function symbology is shown in Figure 1.

Note

When necessary, output coils that do not operate circuits operate as logic coils. It is wise to allow a gap in the numbering between output coils and logic coils to accommodate program expansion.

Each CR coil responds directly to its contact circuit. When the contact circuit conducts, the coil is energized. When the contact circuit does not conduct, the coil is de-energized. When the coil is energized, normally-open (NO) contacts are closed, and normally-closed (NC) contacts are open. When the coil is de-energized, NO contacts are open and NC contacts are closed.

APPLICATIONS

Figure 2 shows a control relay start/stop circuit application. As shown, when pushbutton IN0001 is pressed, CR0001 energizes through the NC contact of IN0002. This causes the CR0001 NO contact to close, holding Coil CR0001 energized until IN0002 is pressed to open the circuit.

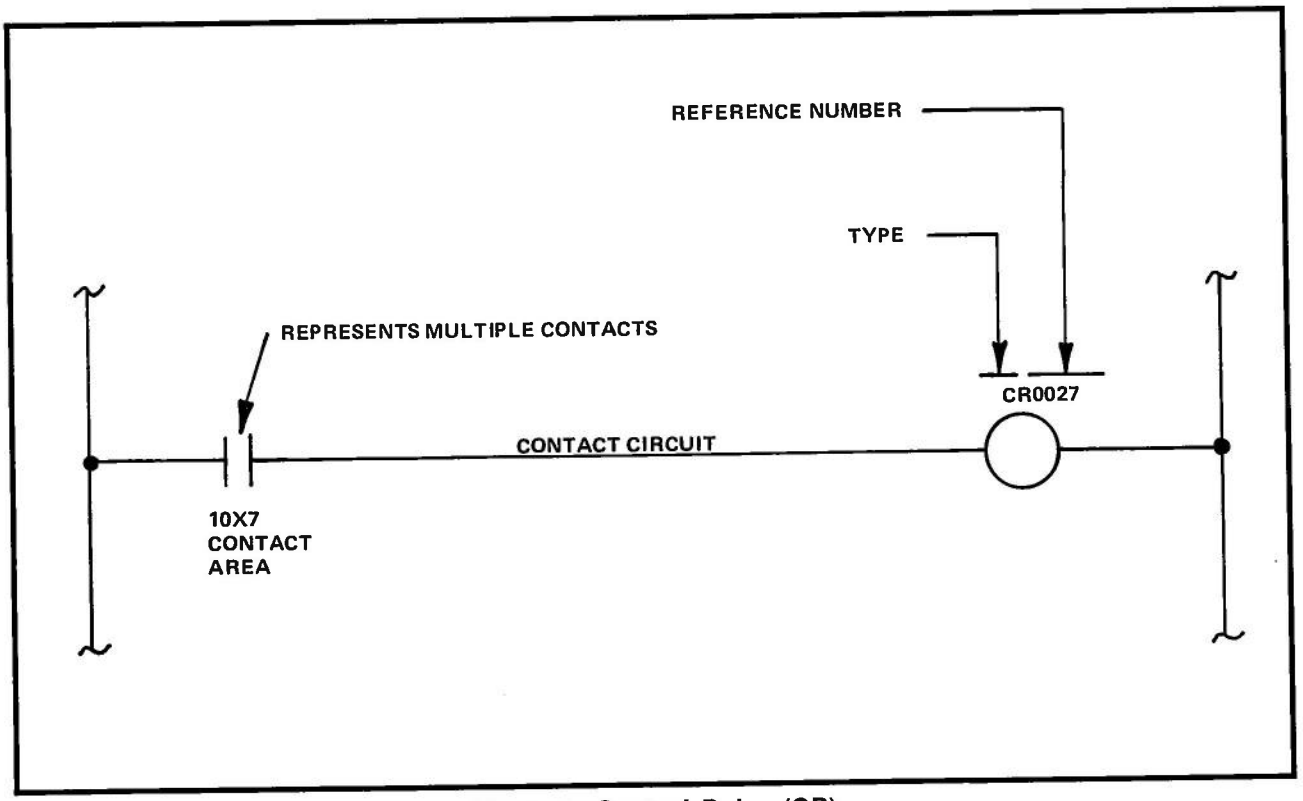


Figure 1. Control Relay (CR)

TABLE 1. COIL REFERENCE NUMBERS

Processor	Memory Size	Reference Numbers	
		Output Coil	Logic Coil
PCX-700 Software †1.X (Extended) †3.X (Advanced) †4.X (Advanced) 5.X (Extended) 6.X (Advanced II)	All (to a maximum of 8192)	1 through 256	257 through 512
PC-700 †Software 2.X standard special functions	256 512 1024 2048 through 4096	1 through 64 1 through 218 1 through 192 1 through 256	65 through 512 129 through 512 193 through 512 257 through 512
PC-900A †All software	All	1 through 127*	None
PC-900B All software	All	1 through 127*	129 through 256

*CR128 is a battery-status coil.
† = No longer available; for reference only.

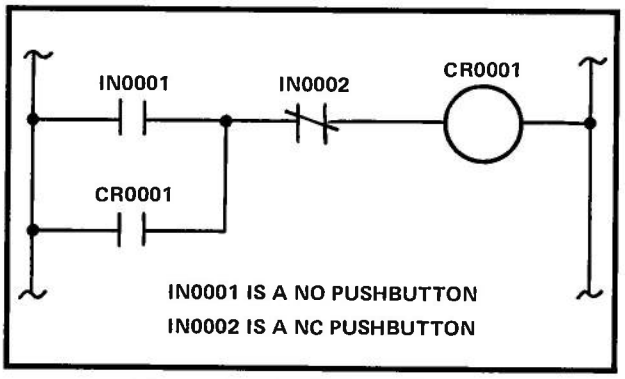


Figure 2. Start/Stop Circuit

The oscillator circuit application of a control relay is shown in Figure 3. Upon the initial scan, the programmable controller detects the NC contacts of CR0002, energizing the CR0002 coil on the next scan. The NC contacts of CR0002 are then solved open, causing coil CR0002 to subsequently de-energize. In this way, the CR0002 coil is energized every other scan.

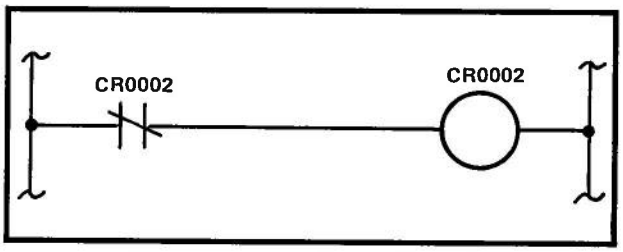


Figure 3. Oscillator Circuit

Figure 4 shows the CR dummy coil circuit application. As shown, CR0003 is always OFF, unless forced ON. This configuration allows the CR0003 contacts to be used whenever a circuit is to be always open (NO contact) or always closed (NC contact). Not programming CR0003 has the same effect as shown in Figure 4; the CR0003 contacts can be used for always ON or always OFF circuits. However, when not programmed, the state of CR0003 is not documented, and it is not immediately apparent that the coil is used as a dummy circuit. Unless

programmed as shown, CR0003 could be forced ON, staying ON even with a force deleted. When programmed, the coil turns OFF when the force command is removed.

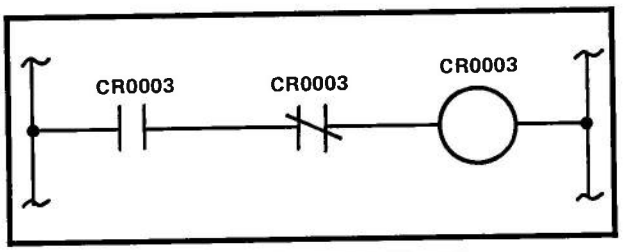


Figure 4. Dummy Circuit

CS — CONTINUOUS SELECT

DESCRIPTION

The Continuous Select (CS) function allows the user to change a transitional function to a continuous function. When the step contact closes, the function specified by the special function's coil reference number is activated on every scan. CS function symbology is shown in Figure 1.

SPECIFICATIONS

OP CODE 11

The Op Code defines the Literal (LT) as a CS function.

Notes

1. When software changes allow, LT becomes CS.

2. Place the CS function, providing it is programmed as a Literal, after the function being changed to prevent contacts from being labeled as LT.

OPERAND 1

Operand 1 is assigned a constant value of 1. It is not used in the CS operation and only serves to complete the special function.

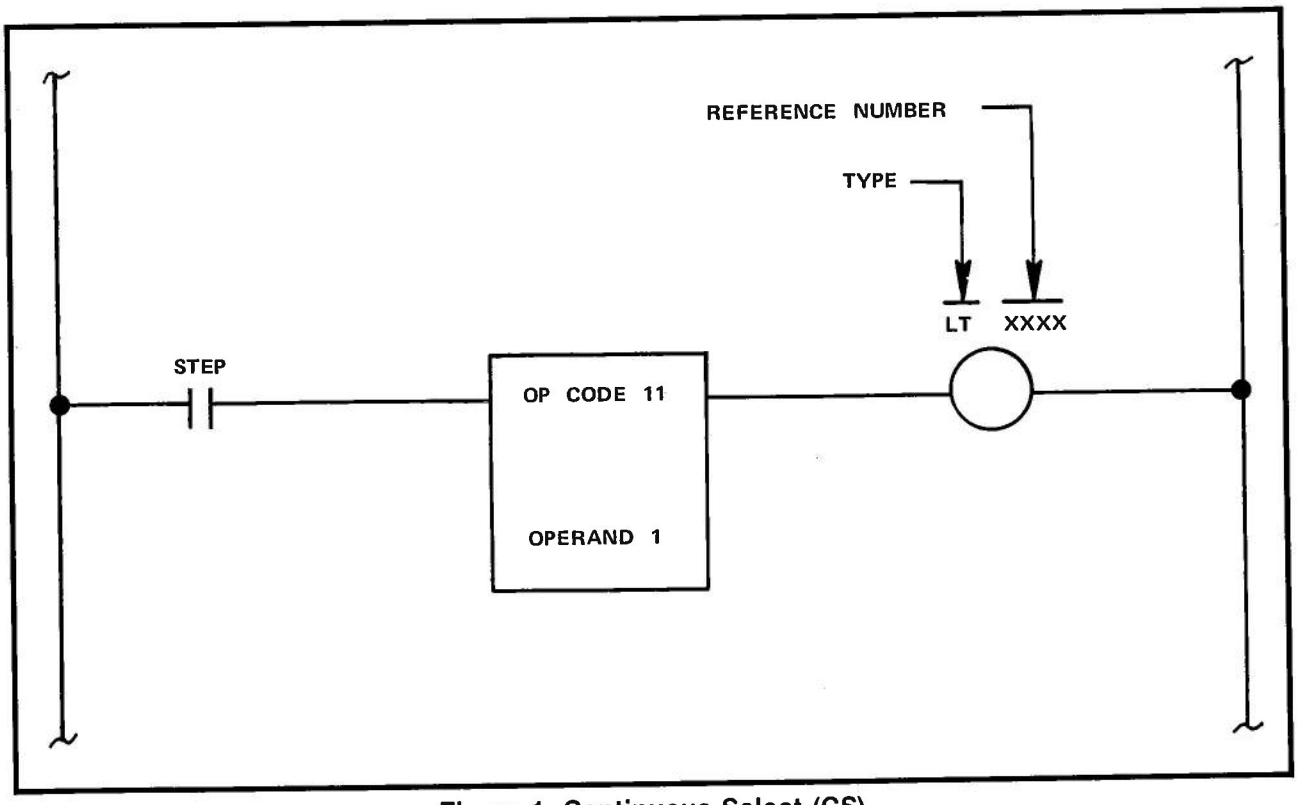


Figure 1. Continuous Select (CS)

CS

COIL

The coil must have the same reference number as the coil of the function being changed from incremental to continuous.

CS TRUTH TABLE

See Table 1.

APPLICATIONS

The CS function causes a normally transitional function (e.g., the Decimal-to-Binary [DB] function) to operate on every scan.

As shown in Figure 2, each time the processor scans the program, the contents of IR0001 are converted from Binary-Coded-Decimal (BCD) to binary and placed in HR0001.

TABLE 1. CS TRUTH TABLE

Step	Result
0	Coil de-energizes. Similarily numbered special function operates normally.
1	Similarily numbered special function operates each scan. Coil never energizes.

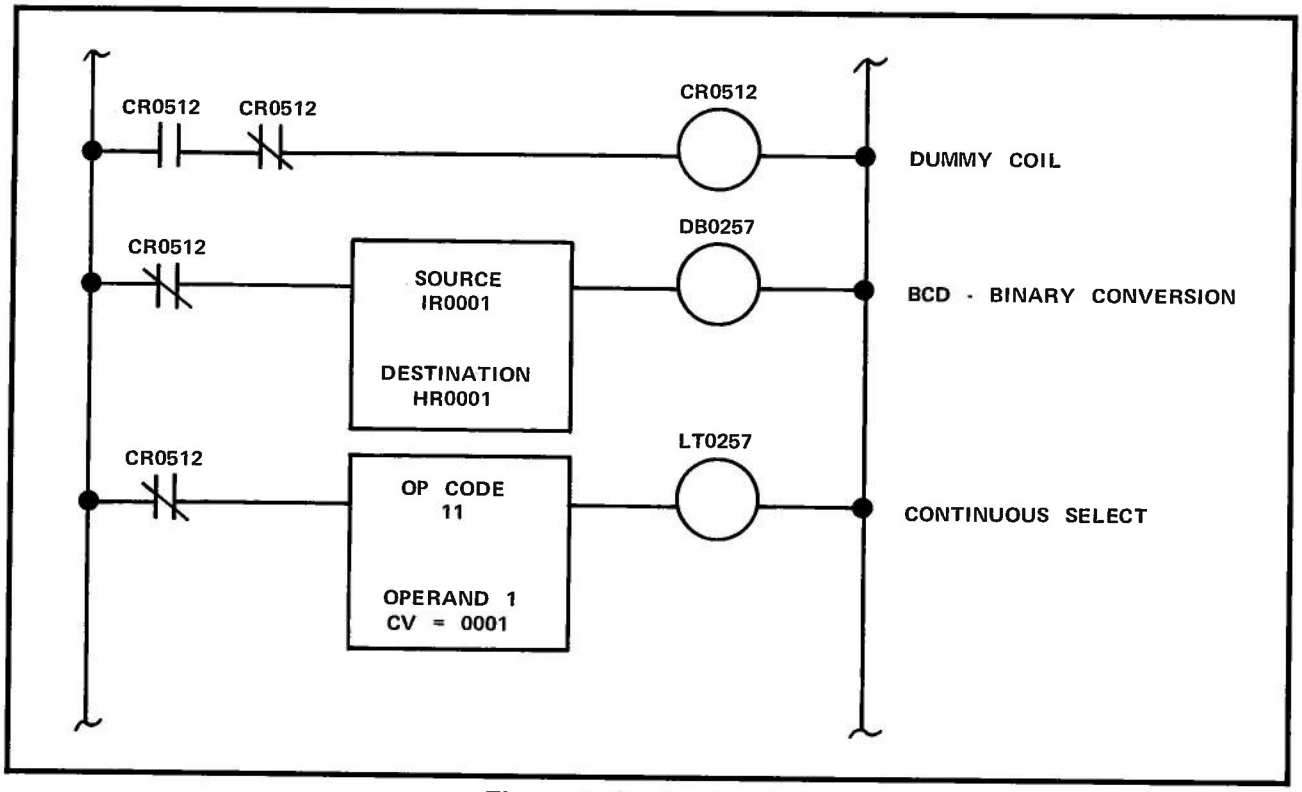


Figure 2. CS Application

DR — DRUM CONTROLLER

DESCRIPTION

The Drum Controller (DR) function allows the user to program a sequence of states for a register or a group of 16 outputs. This sequence has up to 128 steps which define the state of each of the 16 bits contained in the register or group of outputs. DR function symbology is shown in Figure 1.

The DR function consists of the following:

- Data Registers — A group of holding registers assigned to contain the output data.
- Step Pointer — A holding or output register assigned to contain the current step number.
- Destination — An output register or output group assigned to contain the current output states.

Three contact circuits control the operation of the DR function: the step circuit, the reset circuit, and the enable circuit. The step pointer moves to the next step each time the step circut changes from non-conducting to conducting with reset and enable conducting, and resets to zero when the reset circuit is not conducting. The data contained in the register selected by the step pointer's step number is transferred to the destination when the enable circuit conducts. If it does not conduct, the destination contains the same data as when the enable circuit changed from conducting to non-conducting.

The reference number of the first register and the total number of steps specify the holding registers sequentially assigned to store the output states for each step. A constant specifies the total number of steps (1 through 128), and a reference number specifies the first register in a block of holding registers. The additional

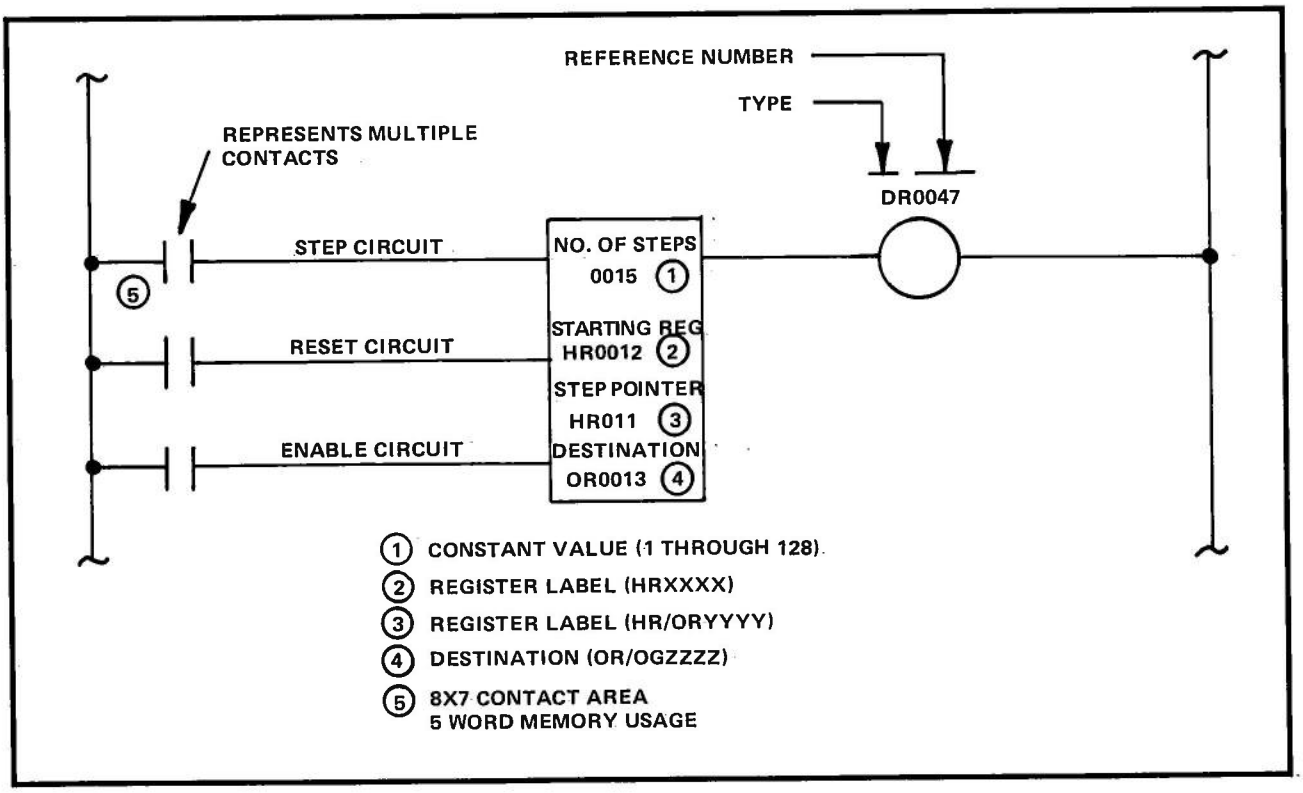


Figure 1. Drum Controller (DR)

registers in the block are automatically designated according to the total number of steps, and are assigned the reference numbers immediately following the first holding register reference number.

The DR coil energizes when the step number increments to the specified total number of steps. Forcing a DR coil affects only the contacts associated with the coil; the DR function continues to operate normally.

During normal operation, the step circuit periodically changes from non-conducting to conducting. At this time, the step number increases by one, so that the step pointer references the next block of output states. These states are then transferred to the destination, as shown in Figure 2. As this process continues, the pre-programmed bit patterns in the data registers appear sequentially in the destination, as each is selected by incrementing the step number in this way.

The length of the DR function is set by specifying the total number of steps. When the step number plus one equals the specified number, the next change from non-conducting to conducting in the step circuit resets the step pointer to Step 000.

Figure 3 shows how DR functions are chained to increase the number of available steps:

1. A normally-open (NO) contact from the previous DR coil appears in the enable and reset circuits of the next DR.
2. A normally-closed (NC) contact from each DR coil (except for the last DR function in the chain) appears in its own enable circuit.
3. A normally-closed (NC) contact from the last DR coil appears in the reset circuit of the first DR function in the chain.

When chained, the last register of the previous DR and the first register of the next DR should be the same.

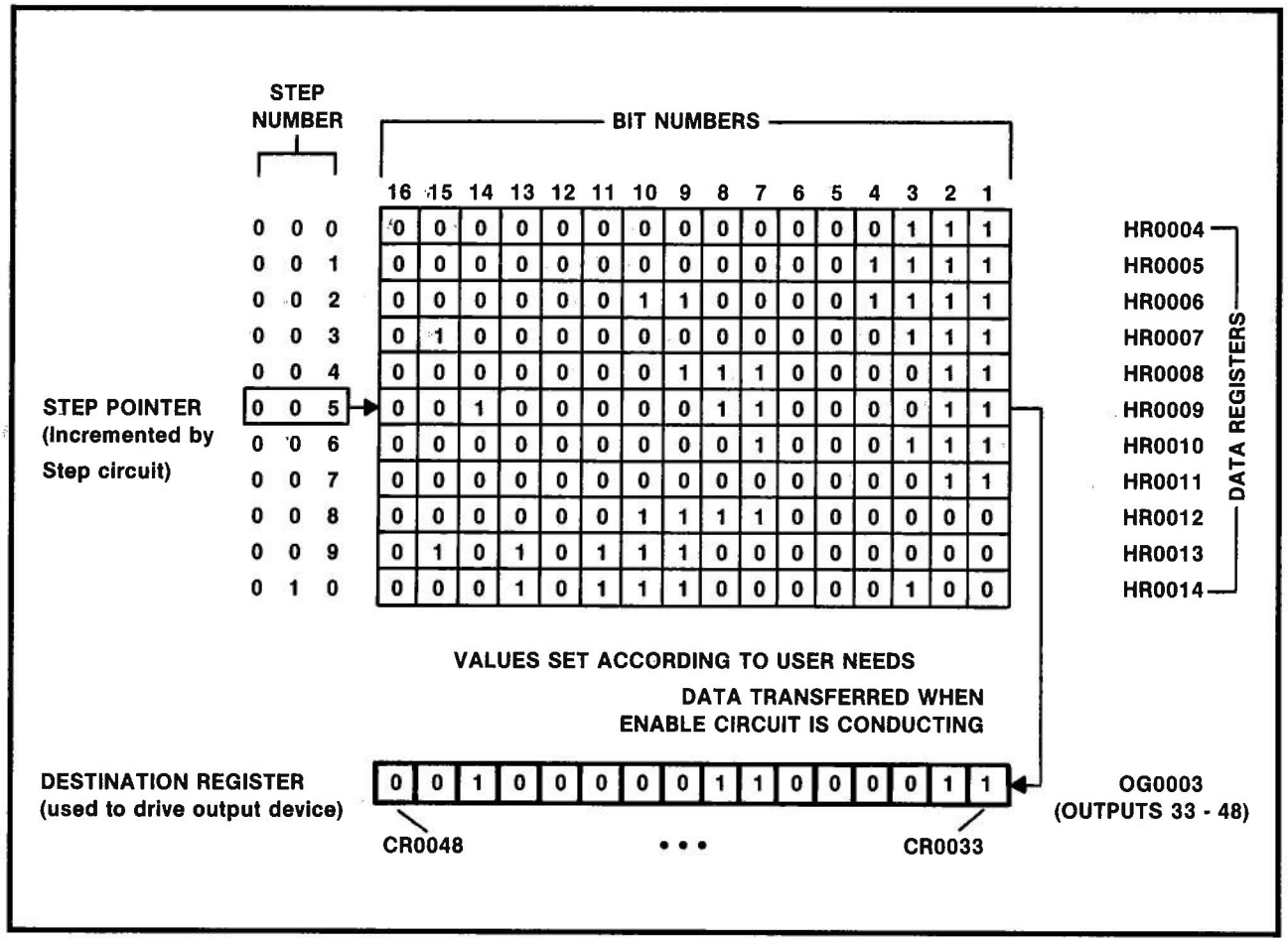


Figure 2. Eleven-Step DR

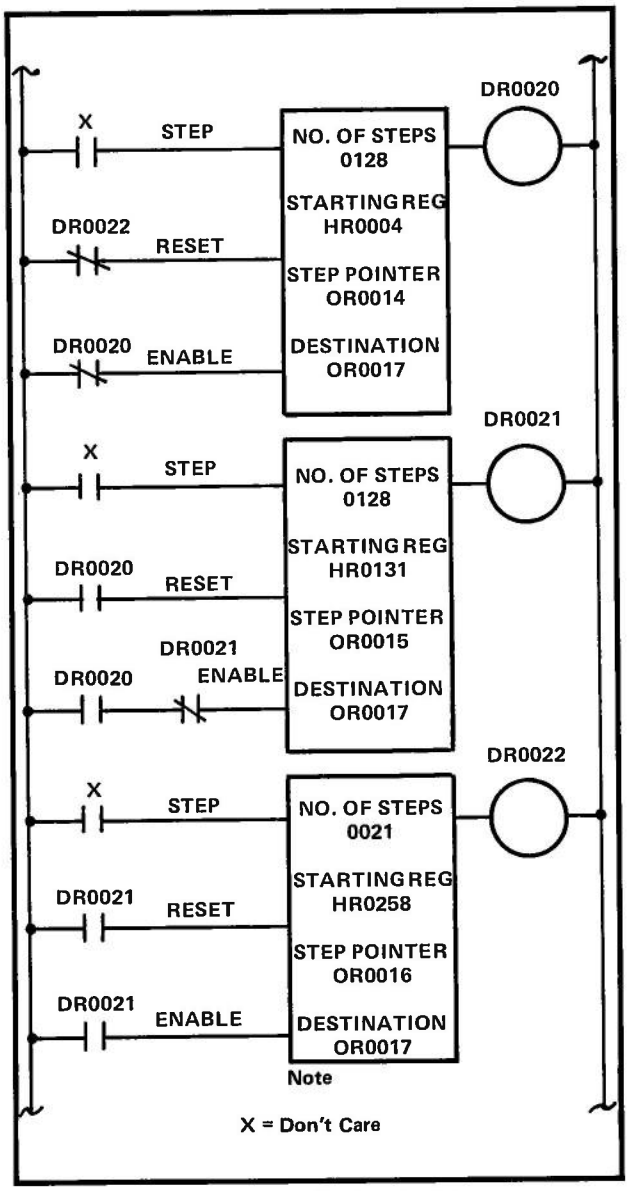


Figure 3. Chained DR

SPECIFICATIONS

NUMBER OF STEPS

The number of steps is a constant value from 1 through 128 defining the length (in steps) of the DR function sequence.

STARTING REGISTER

The starting register is the first holding register in the consecutive sequence containing the desired data for each step in the DR function sequence.

STEP POINTER

The step pointer specifies the register that contains the number of the current step in the DR sequence. One of the following registers may be used:

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

DESTINATION

The destination contains the data in the current DR function sequence step. This holding area is one of the following:

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

COIL

The coil energizes when the step number plus one equals the specified total number of steps.

DR TRUTH TABLE

See Table 1.

APPLICATION

In its simplest form, DR drives up to 16 discrete outputs through as many as 128 steps, and is programmed as shown in Figure 4.

DR does not have to use the step circuit, and can be readily manipulated by manipulating the step pointer, as shown in Figure 5.

In some situations, it may be required to have a DR automatically operate at different intervals between steps. This can be accomplished by using two DRs, as shown in Figure 6.

TABLE 1. DR TRUTH TABLE

Enable Circuit	Reset Circuit	Step Circuit	Result
0	0	X	The step pointer is held at zero; no data is moved.
0	1	X	The step pointer is held at the current step; no data is moved.
1	0	X	The step pointer is reset to zero; data is moved from Step 0 to the destination.
1	1	0	The step pointer is held at the current step; data is moved.
1	1	↑	The step pointer is incremented by one; data is moved from the new step to the destination.
1	1	1	The step pointer is held at the current step; data is moved from the current step to the destination.

X = Don't Care ↑ = Transition from OFF to ON

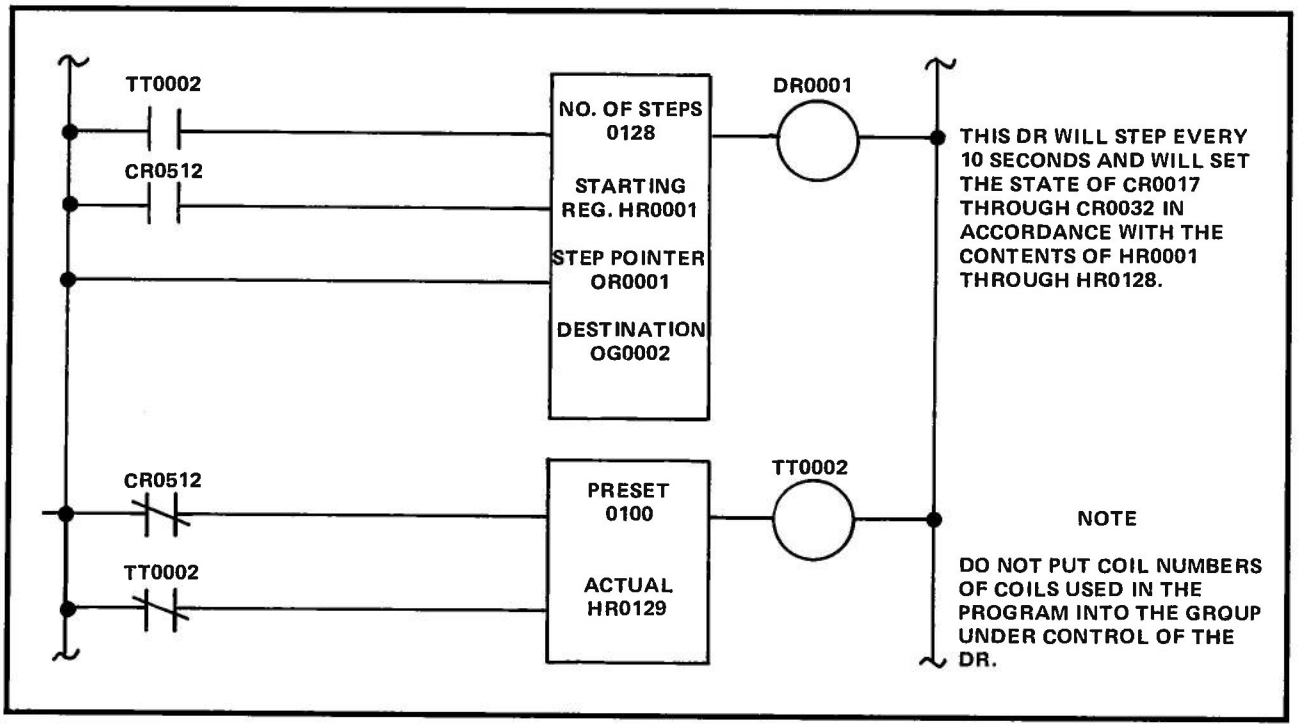


Figure 4. DR Program

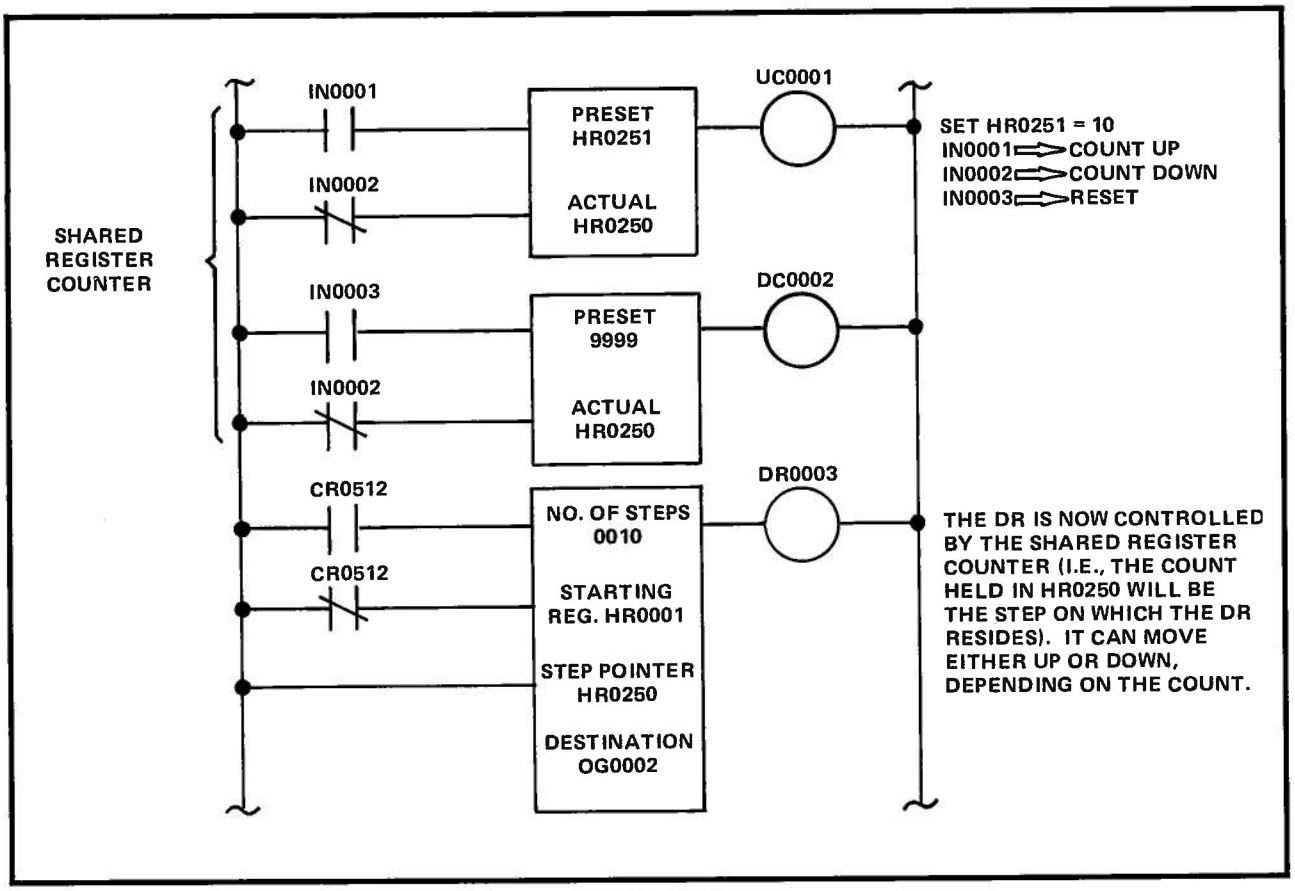


Figure 5. DR Application

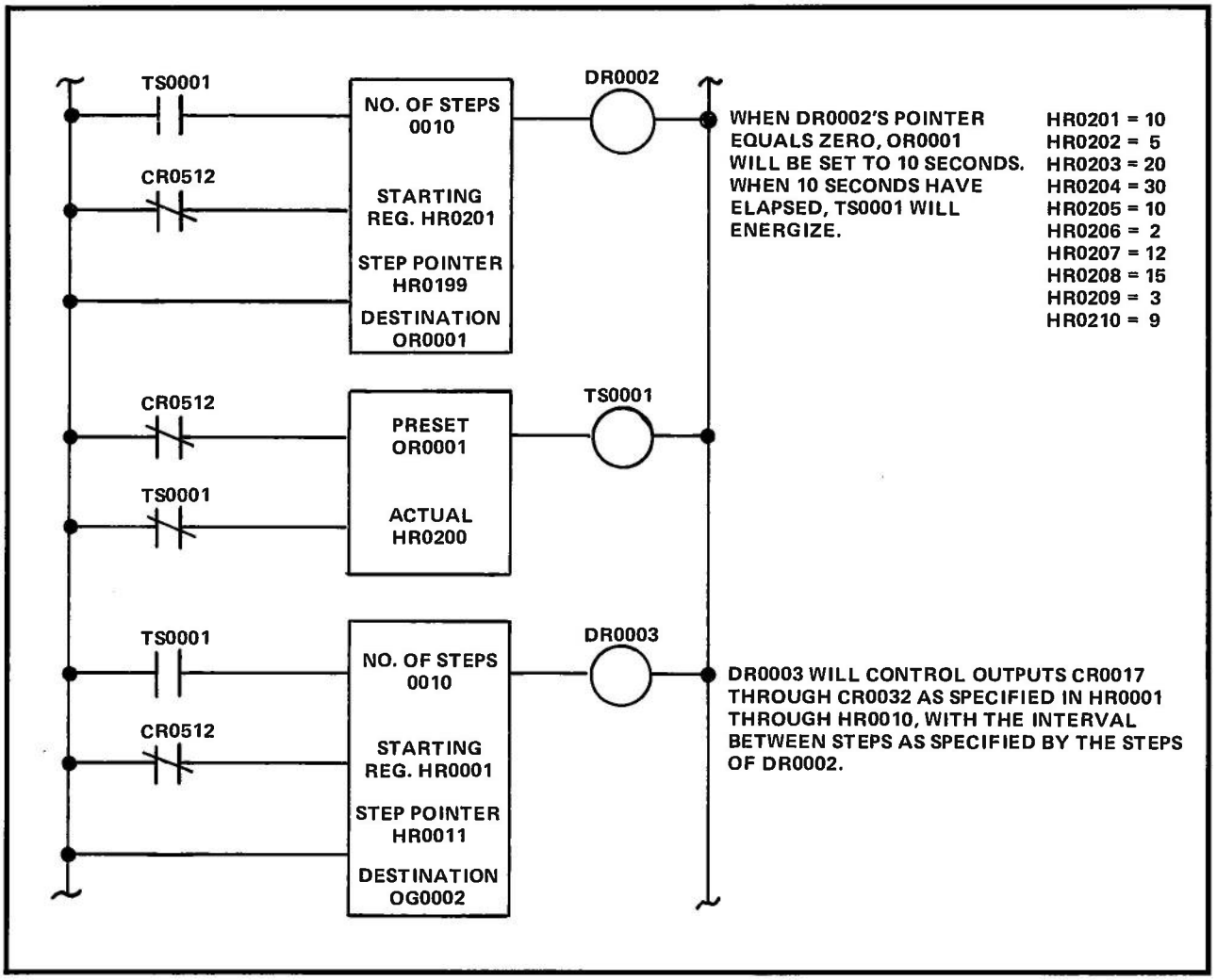


Figure 6. Application Using Two DRs

DV — DIVIDE

DESCRIPTION

The Divide (DV) function divides an eight-digit decimal number (up to 99,999,999) by a four-digit decimal number (up to 9999), resulting in a number of up to four digits. DV function symbology is shown in Figure 1.

Operand 1 is divided by Operand 2 when the enable circuit changes from non-conducting to conducting. Operand 2 comes from an input, output, or holding register or is programmed as a constant. Operand 1 comes only from a pair of these registers. Both Operand 1 registers are specified by a single reference label that automatically designates both the register to which it refers (most-significant register), and the register of the same type having the next highest reference number (least-significant register). For example, if Operand 1 is 00782493, the register with the lower reference number contains 0078, and the register with the next highest reference number contains 2493.

Although the limits on the operands and the result are decimal, division is performed using binary numbers. Therefore, the register values for Operand 1 and Operand 2 must be stored in binary form. If a number is originally in Binary-Coded-Decimal (BCD) form, it may be converted to binary form by using the Decimal-to-Binary (DB) function. If the result is needed in BCD form, the Binary-to-Decimal (BD) function is used.

Each of the registers for Operand 1 is limited to 9999. Therefore, numbers originally in binary form and in excess of 9999 cannot be used for Operand 1 in a DV function.

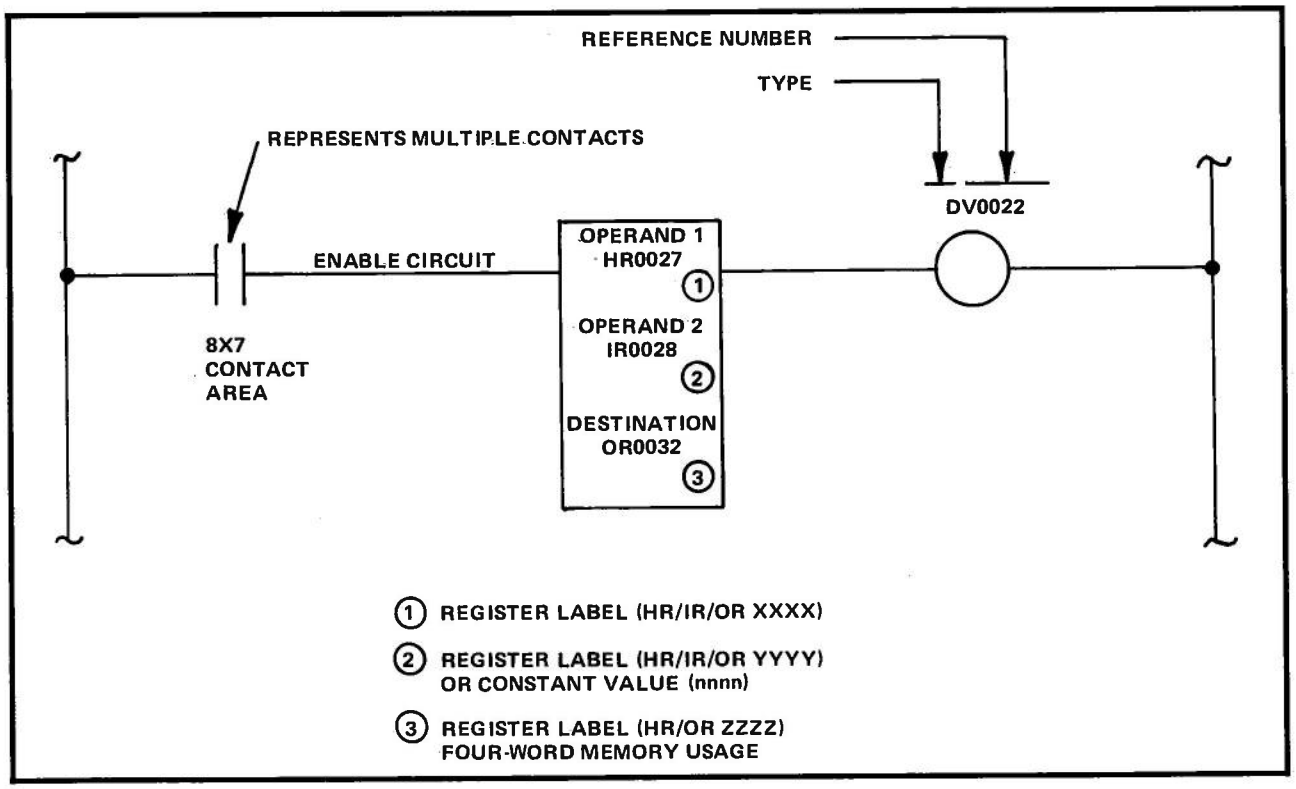


Figure 1. Division (DV)

Note

All registers are limited to 9999, including the first register of the result.

The division result is placed in a destination register, which is composed of a consecutive pair of output or holding registers. The first destination register contains the result; the second destination register contains the remainder. The destination register pair is designated by a single reference label, which automatically designates the register to which it refers and the register of the same type having the next highest reference number.

The DV coil energizes if Operand 2 equals 0000, or if the result is greater than 9999. The coil energizes to indicate that an error has been detected. If there is an error, the destination register is not updated and contains the same numbers as prior to enabling the function.

SPECIFICATIONS

OPERAND 1

Operand 1 is the dividend contained in a consecutive pair of locations. The first location contains the four most-significant digits; the second location contains the four least-significant digits. This value is held in a specified pair of Holding Registers (HR), Input Registers (IR), or Output Registers (OR).

OPERAND 2

Operand 2 is the divisor. This value is a constant (0001 through 9999) or may be held in a specified Holding Register (HR), Input Register (IR), or Output Register (OR).

DESTINATION

The destination is a consecutive pair of register locations. The first location contains the result; the second location contains the remainder. The destination is a specified pair of Holding Registers (HRs) or Output Registers (ORs).

Note

Both operands must be in binary form.

DATA SUMMARY

The Operand 1 range is 0000 through 99,999,999. The Operand 2 range is 0000 through 9999. Operand 1 divided by Operand 2 equals the destination (0 through 9999 with a remainder).

DV TRUTH TABLE

See Table 1.

TABLE 1. DV TRUTH TABLE

Enable Circuit	Result
0	None. The DV coil de-energizes.
↑	Operand 1 is divided by Operand 2. The result is placed in the destination register. The coil energizes when the circuit is enabled and either Operand 2 is zero or the result is greater than 9999; the destination retains the last valid computed value.
1	None. The coil retains the state caused by the transition.

APPLICATIONS

The DV function is most useful in scaling internal quantities for application to an analog output module that accepts a decimal range from 0 through 4095. If the processor operates on a number in the range of 0 through 8190, the number is scaled for analog output operation by dividing by 0002, as shown in Figure 2.

Figure 3 shows a 16-bit word divided into two eight-bit words.

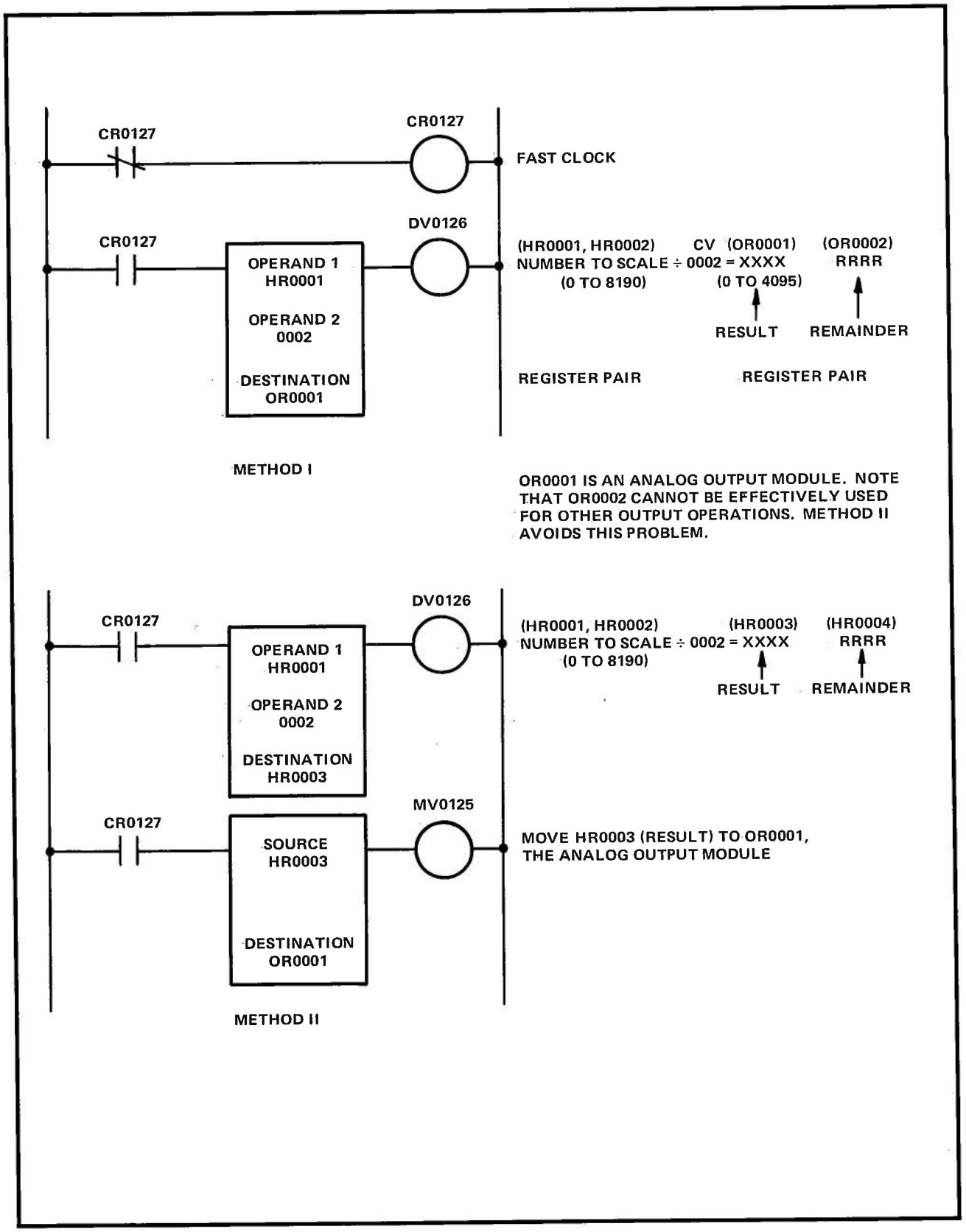


Figure 2. DV Scale Factoring

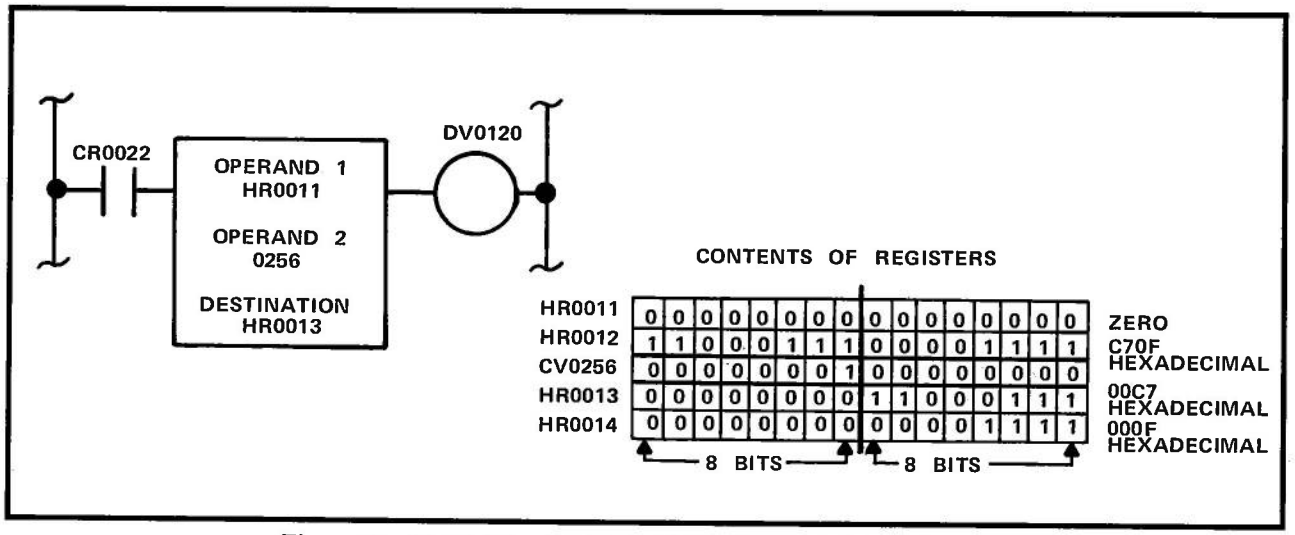


Figure 3. 16-Bit Word Divided into Two, Eight-Bit Words

EQ/GE — COMPARISONS

DESCRIPTION

There are two Comparison functions: Equal To (EQ) and Greater Than Or Equal To (GE). The EQ/GE functions operate a coil by comparing two four-digit decimal numbers. EQ/GE function symbology is shown in Figure 1.

The two numbers, Operand 1 and Operand 2, are compared when the enable circuit conducts. Operand 1 comes from an input, holding, or output register. Operand 2 comes from one of these registers, or is programmed as a constant value.

Any comparison ($<$, $>$, $=$, $\neq$, $\leqslant$, $\geqslant$) is made by carefully choosing the type of contacts used or by combining the contacts from both the EQ and GE functions. For example, if the comparison function is A equals B (A = B) and if the enable circuit conducts, the normally-closed (NC) contact is closed when A does not equal B. Similarly, the NC contacts from a GE coil indicate when A is less than B (A < B). A comparison of A greater than B (A > B) can be generated by the series combination of a normally-open (NO) GE contact and an NC EQ contact. A comparison of A less than or equal to B ($A \leqslant B$) can be generated by the parallel combination of an NC GE contact and an NO EQ contact. When comparison functions are used together in this manner, identical contact circuits should be used for both functions.

Table 1 lists the six possible comparisons and the condition of the circuit when the equation is true.

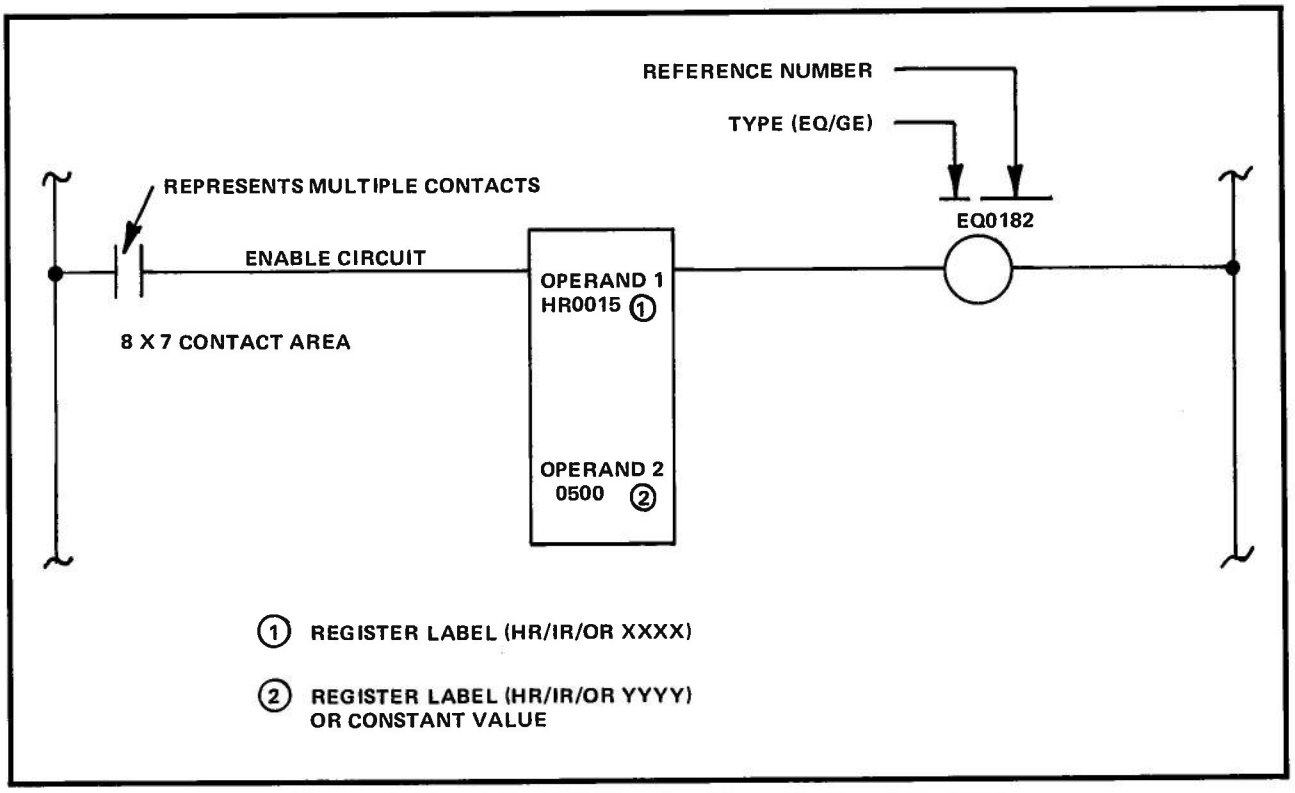


Figure 1. Equal To (EQ)/Greater Than Or Equal To (GE)

EQ/GE

TABLE 1. COMPARISON FUNCTIONS

Function	Equation	Circuit (Conducts When Equation is True)
Equal (EQ)	A = B	EQ
Not Equal	A ≠ B	EQ
Greater Than or Equal To (GE)	A ≥ B	GE
Less Than	A < B	GE
Greater Than	A > B	GE EQ
Less Than or Equal To	A ≤ B	GE EQ

EQ

When the enable circuit conducts and Operand 1 equals Operand 2, the EQ coil energizes. If Operand 1 and 2 are not equal, or if the enable circuit does not conduct, the coil de-energizes.

GE

When the enable circuit conducts and Operand 1 is greater than or equal to Operand 2, the GE coil energizes. If Operand 1 is less than Operand 2, or if the enable circuit does not conduct, the coil de-energizes.

SPECIFICATIONS

ENABLE CIRCUIT

When the enable circuit conducts, Operand 1 is continuously compared to Operand 2. When the circuit does not conduct, the function disables and the coil de-energizes.

OPERAND 1

Operand 1 is the number to be compared (A). This value is held in a specified register:

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

OPERAND 2

Operand 2 (B) is the number to which Operand 1 (A) is compared. This value is a constant (0001 through 9999) or is held in a specified register:

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

COIL

The EQ coil energizes when the enable circuit is conducting and Operand 1 equals Operand 2.

The GE coil energizes when the enable circuit is conducting and Operand 1 is greater than or equal to Operand 2.

Both functions de-energize at all other times.

APPLICATIONS

The GE/EQ functions allow monitoring of an analog input within a range of values. In the Add (AD)/Subtract (SB) functions, a method of setting a range around a set point is discussed. Using the limit registers established in that example, HR0003 holds the upper limit, and HR0005 holds the lower limit, as shown in Figure 2.

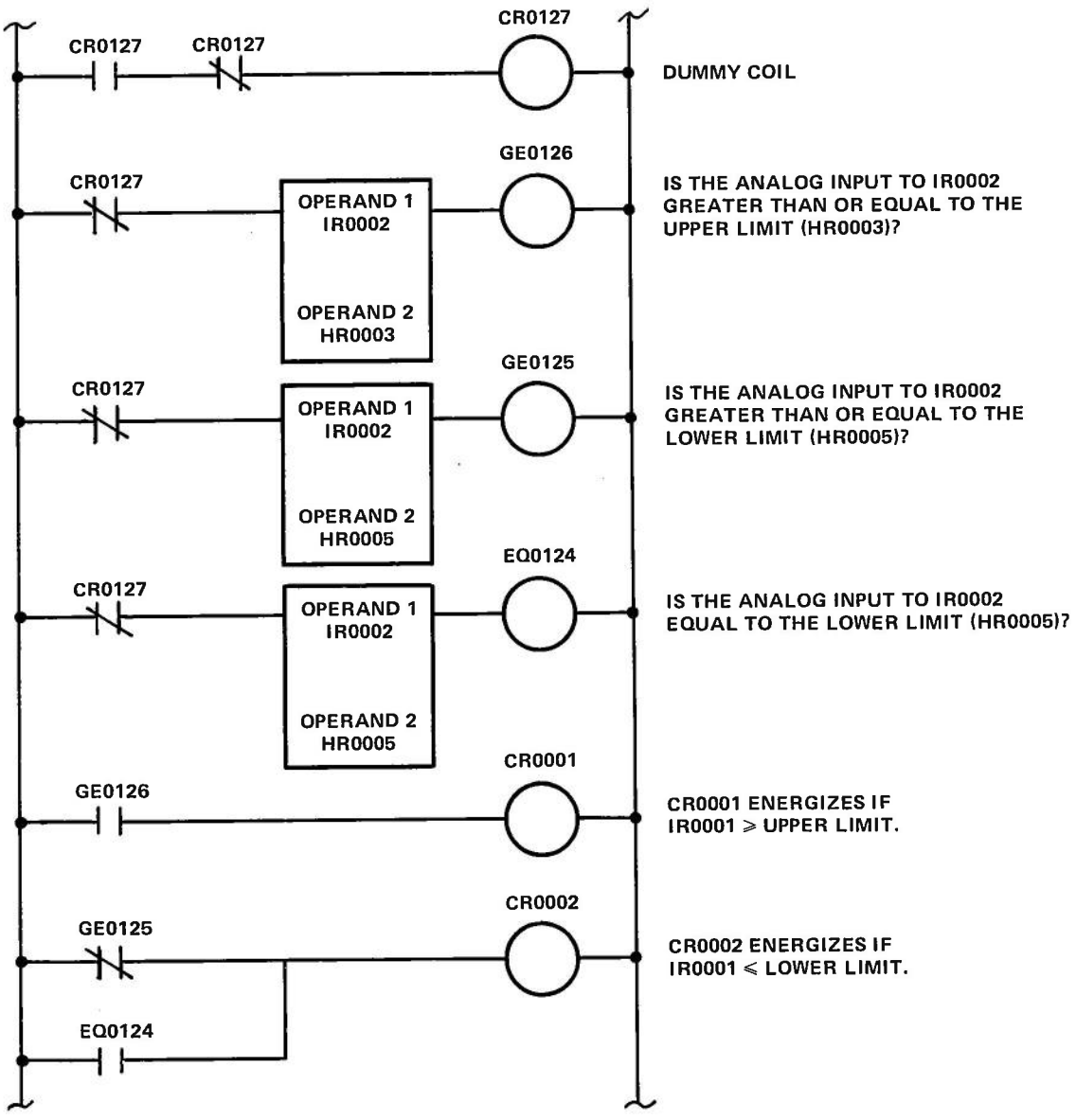


Figure 2. Limit Comparisons

FI/FO — FIFO STACK

DESCRIPTION

The FIFO Stack function is a combination of two functions: the First In Stack (FI) function, and the First Out Fetch (FO) function. In FIFO Stack, the first data entered is the first data retrieved. FI function symbology is shown in Figure 1; FO function symbology is shown in Figure 2.

Note

The FI and FO functions are designed to be used together.

Figure 3 uses a card playing analogy to demonstrate the FIFO Stack concept. Individual cards are stacked on a table; then, they are dealt from the bottom of the stack. When Card No. 1 is removed, Card No. 2 occupies the bottom position.

SPECIFICATIONS

OP CODE 81/85

The Op Code defines the Literal (LT) as either an FI or FO function.

Note

When software changes allow, LT becomes FI and FO.

TABLE LENGTH

The table length is a constant value that defines the size of the FIFO Stack (i.e., number of registers). Both functions should be programmed with the same table length. The size of the FIFO Stack is limited, as specified in Table 1, with a maximum of 256.

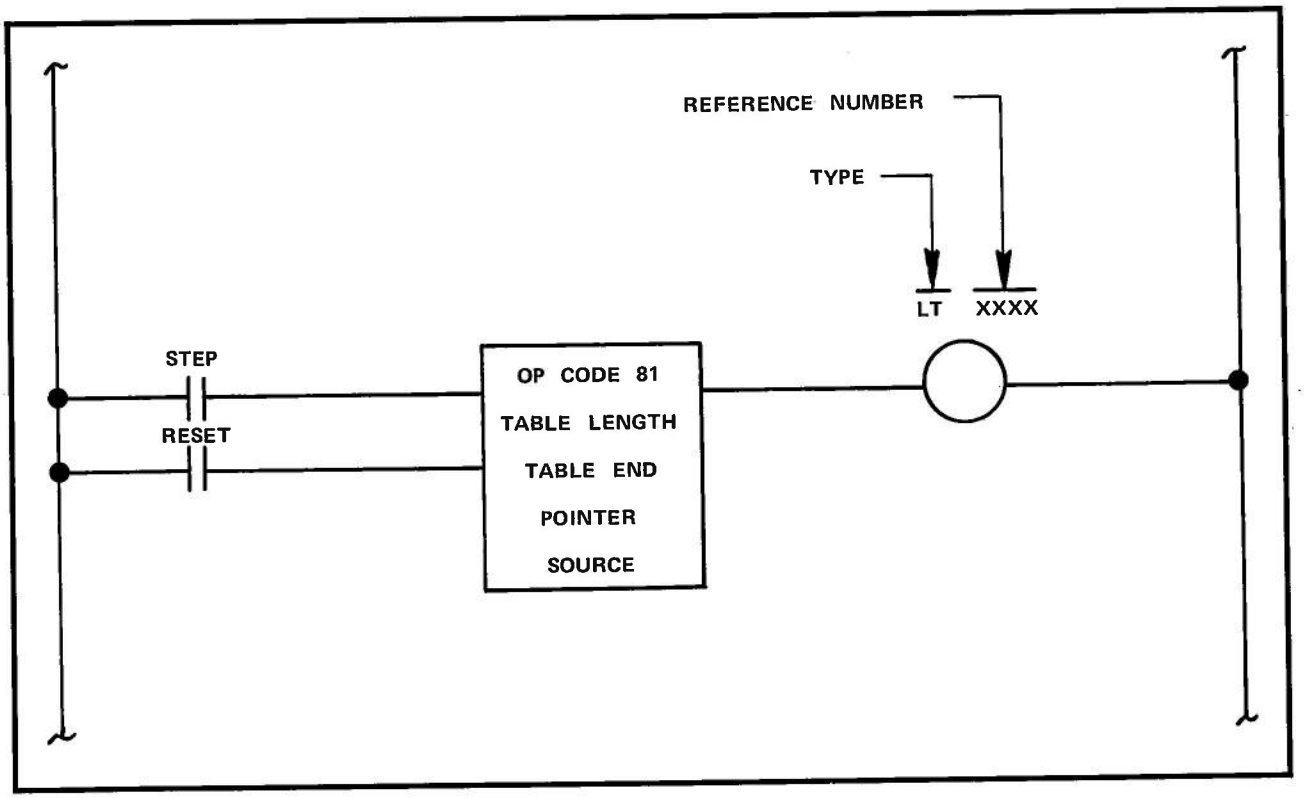


Figure 1. FI Function

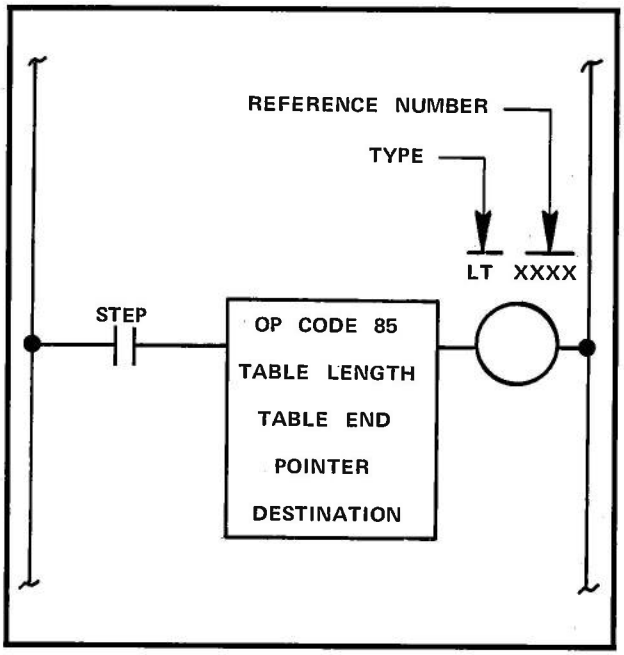


Figure 2. FO Function

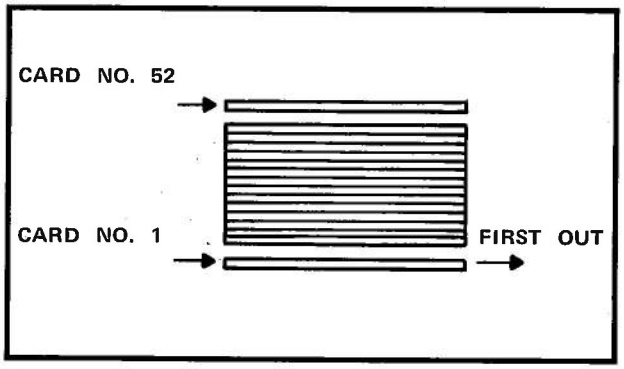


Figure 3. FIFO Stack Concept

Note

The highest available number of the holding registers in Table 1 is 1792, and is dependent on memory size.

TABLE END

The table end defines the type and number of the last register in the FIFO Stack, which is subject to the limits in Table 1. Both functions should be programmed with the same table end.

TABLE 1. TABLE LENGTH AND TABLE END LIMITS

Type	Limit
HR	⩽ 1792
OR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
OG	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)

POINTER

The pointer is a register that contains the number of the FIFO Stack location. This register must be the same in both functions:

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

SOURCE

The source in the FI function is the location from which the FIFO Stack is filled. This location is a specified register or group:

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

DESTINATION

The destination is used in the FO function to determine the location to which data is moved from the FIFO Stack. This location is a specified register or group:

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

FI TRUTH TABLE

See Table 2.

TABLE 2. FI TRUTH TABLE

Step	Reset	Result
X	0	The coil de-energizes. The pointer sets to zero.
↑	1	The source moves to the position in the stack designated by the pointer. After the move, the pointer increments by 1.
1	1	The coil energizes when the pointer value equals the stack size, and de-energizes when the pointer value is less than the stack size.

FO TRUTH TABLE

See Table 3.

TABLE 3. FO TRUTH TABLE

Step	Result
0	The coil de-energizes. No data transfers.
↑	Data transfers from the table end position in the destination. The contents of the table shift down one place. The pointer decrements by 1.
1	The coil energizes when the pointer equals zero.

APPLICATIONS

The FIFO Stack is used in a situation where a machine performs a controlled sequence with variable data. For example, a machine that paints an object that comes in several colors can be programmed for the day's production. The operator need only enter the color choices, and the program retrieves the choice for the machine in the proper order. (See Figure 4.)

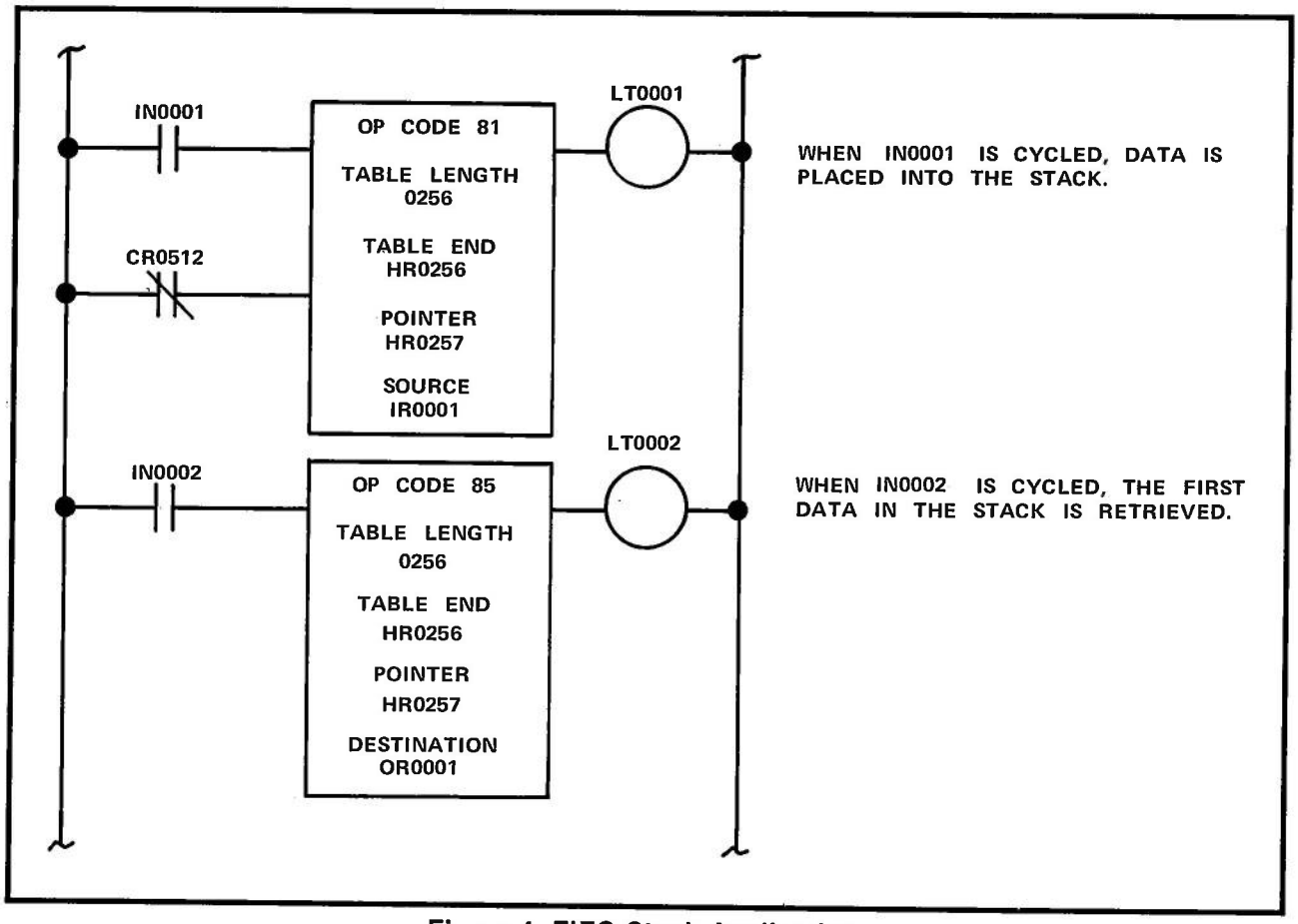


Figure 4. FIFO Stack Application

FI/LO — FILO STACK

DESCRIPTION

The FILO Stack function is a combination of two functions: the First In Stack (FI) function, and the Last Out Fetch (LO) function. In the FI/LO functions, the first data entered is the last data retrieved. FI function symbology is shown in Figure 1; LO function symbology is shown in Figure 2.

Note

FI and LO functions are designed to be used together.

Figure 3 uses a card playing analogy to demonstrate the FILO Stack concept. Individual cards are stacked on a table, and dealt from the top of the deck. When Card No. 52 is removed, Card No. 51 occupies the top position. Card No. 1 is the last card removed.

SPECIFICATIONS

OP CODE 81/86

The Op Code defines the Literal (LT) as either an FI or LO function.

Note

When software changes allow, LT becomes FI and LO.

TABLE LENGTH

The table length is a constant value that defines the size (i.e., number of registers) of the FILO Stack. Both functions should be programmed with the same table length. The size of the FILO Stack is limited, as specified in Table 1, with a maximum of 256.

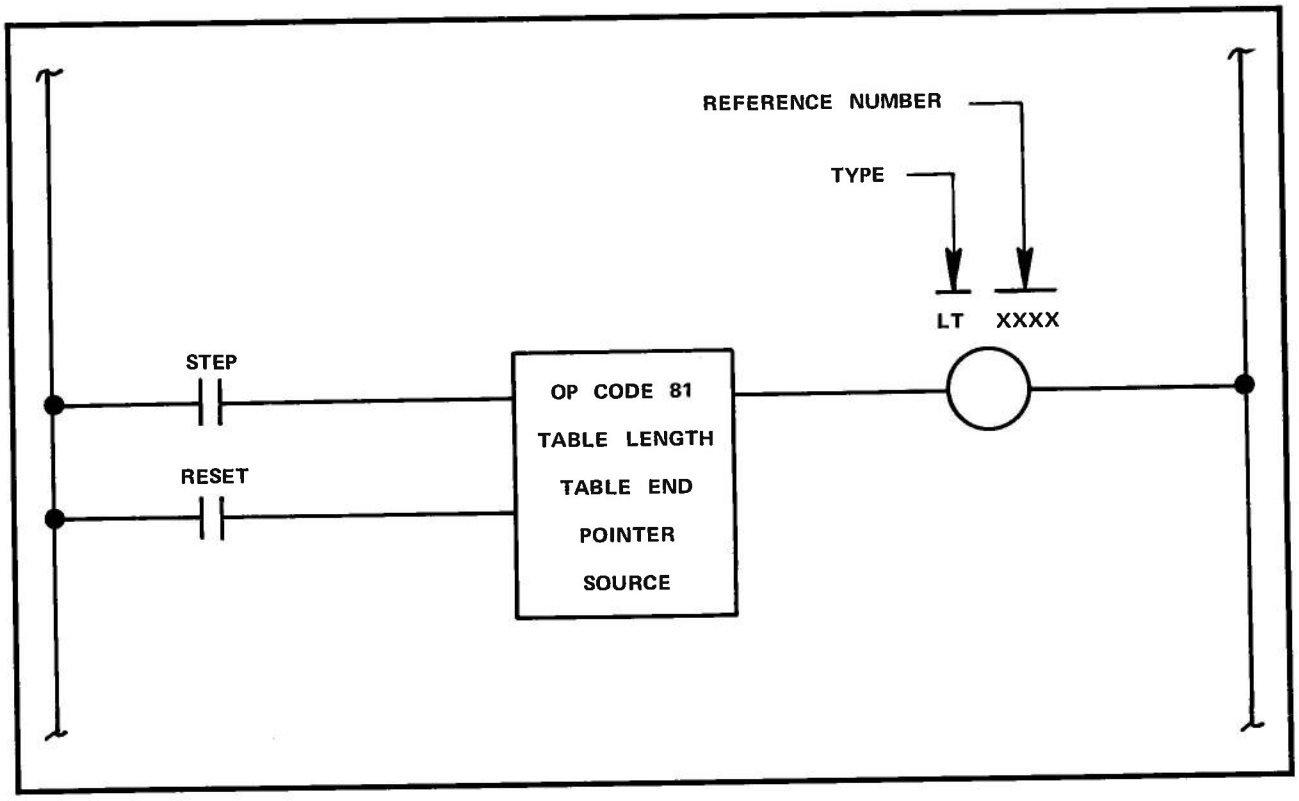


Figure 1. First In Stack (FI)

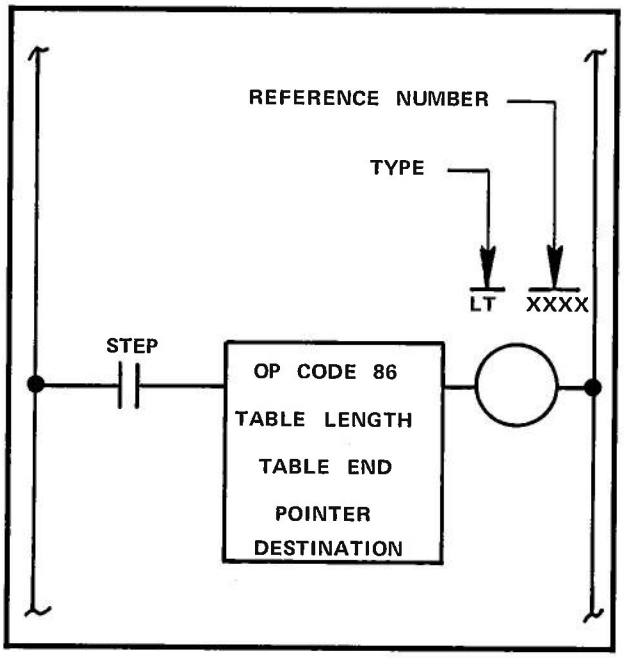


Figure 2. Last Out Fetch (LO)

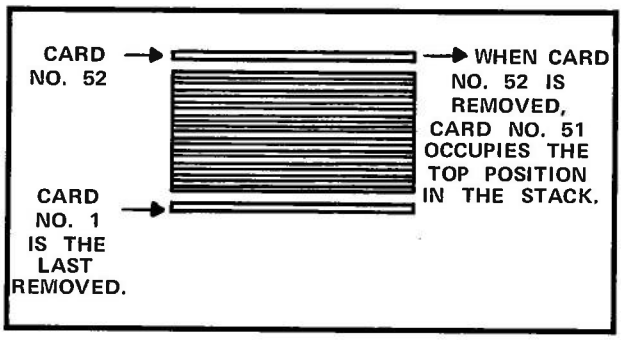


Figure 3. FILO Stack Concept

Note

The highest available number of the holding registers in Table 1 is 1792, and is dependent on memory size.

TABLE END

The table end defines the type and number of the last register or group in the FILO Stack and is subject to the limits in Table 1. Both functions should be programmed with the same program end.

TABLE 1. TABLE LENGTH AND TABLE END LIMITS

Type	Limit
HR	⩽ 1792
OR	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)
OG	⩽ 32 (PC-700) ⩽ 8 (PC-900A) ⩽ 16 (PC-900B)

POINTER

The pointer is a register that contains the number of the current FILO Stack location. This register must be the same in both functions:

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

SOURCE

In the FI function, the source defines the location from which a stack is filled. This location is a specified register or group:

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

DESTINATION

The destination is used in the LO function to define the location to which data is moved from the FILO Stack. This location is a specified register or group:

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

FI TRUTH TABLE

See Table 2.