File No. 7070/7074-01
GA22-7003-6

IBM Systems Reference Library

IBM 7070-7074 Principles of Operation

This manual provides a complete description of the 7070-7074 systems, including optional features. Each operation code is explained and illustrated. Priority processing, scatter read-write, block transmission, zero suppression, the customer engineering console, and other features of the 7070-7074 are explained in detail.

This publication, Form A22-7003-6, is a reprint of Form A22-7003-5. It incorporates Technical Newsletter N22-0173-0

Copies of this and other IBM publications can be obtained through IBM Branch Offices.
Address comments concerning the contents of this publication to:
IBM Corporation, Customer Manuals, Dept. B98, PO Box 390, Poughkeepsie, N.Y. 12602

Contents

IBM 7074 Data Processing System

IBM 7070-7074 Data Processing Systems

The IBM 7070 and 7074 are solid-state electronic data processing systems for commercial and scientific applications. Efficient, high-speed components are controlled by instructions embodying great power and versatility. Series of instructions written in mnemonic and symbolic languages are translated by Applied Programming programs into decimal-number machine language instructions and placed in core storage. Such stored programs then control the input, processing, and output of data. A priority processing feature permits multiple operations, involving input-output units, to occur simultaneously with processing of the main program.

IBM Applied Programming packages of thoroughly tested programs, uniquely fitted to a wide variety of applications, contribute greatly to the system's outstanding ratio of profit to its users.

IBM 7074 production rates are two to ten times faster than those of the IBM 7070. An installed 7070 may be field-converted to a 7074. Programs for the 7070 are fully compatible with the 7074 and may be used without modification.

The IBM 7601 Arithmetic and Program Control, and the IBM 7301 Magnetic Core Storage, with its IBM 7602 Core Storage Control, are the center of activity in the 7070 system. The 7601 is the central processing unit (CPU) for the 7070 system. The CPU for the 7074 is the 7104 High-Speed Processor.

The CPU for each system contains the registers, accumulators, and circuitry for executing arithmetic and logical instructions and for addressing data and instructions.

All information entering and leaving the system passes through the 7301 Core Storage. The data lines and channels that connect system components are represented schematically in Figure 1.

Configurations of the IBM 7070-7074

An important aspect of 7070-7074 systems is their modular design. The range from medium-scale through large-scale systems is covered by a variety of configurations. Card input-output systems may be expanded to use an intermediate tape-and-card system. Random access disk storage may be added. Or, a system may have a tape-oriented configuration and use only magnetic tape units for high-speed input and output operations (plus the IBM 7501 Console Card Reader for entering exception data). A tape-oriented system may be used in conjunction with the IBM 1401 Data Processing System for high-volume applications. The 1401 then controls and performs all auxiliary operations, such as conversion of punched cards to magnetic tape, and magnetic tape to punched cards or printed report. Tape written by a 1401 system can be used on the 7070-7074, and vice versa.

The following equipment is available for use in a 7070 or 7074 system configuration:

IBM NO.	NAME	MAX NO. IN SYSTEM
729	Magnetic Tape Unit	40
1301	Disk Storage	5
1302	Disk Storage	
1414	Input-Output Synchronizer, Model 6	4
7150	Console Control Unit	1
7301	Magnetic Core Storage	
	7070: Model 1 or 2	1
	7074: Model 3 or 4	1
7340	Hypertape Drive (7074 only)	20
7500	Card Reader	3
7501	Console Card Reader	1
7400	Printer	3
7550	Card Punch	
7600	Input-Output Control	1
7601	Arithmetic and Program Control (7070)	1
7104	High-Speed Processor (7074)	1
7602	Core Storage Control	1
7603	Input-Output Synchronizer	1
7604	Tape Control	2
7631	File Control	2
7640	Hypertape Control (7074 only)	1
7740	Communication Control	1
7750	Programmed Transmission Control	1
7802	Power Converter	1
7900	Inquiry Station	10
7907	Data Channel	1

IBM 729 Magnetic Tape Units: As many as 40 magnetic tape units, in four strings of as many as ten units each, can be used.

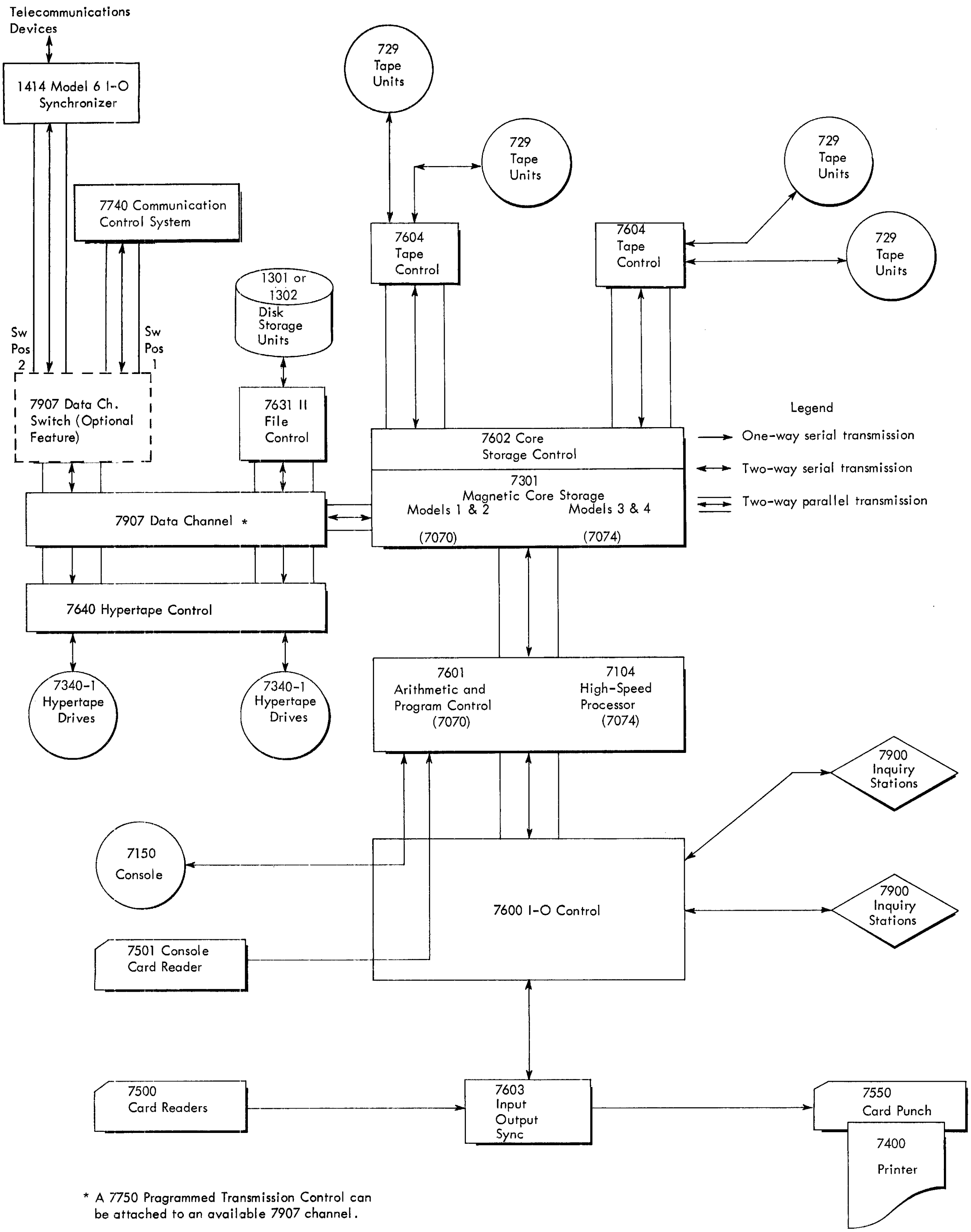
IBM 1301 Disk Storage: A system can include as many as five 1301 disk storage units with a storage capacity of up to 432 million positive digits.

IBM 1302 Disk Storage: A 7074 can have as many as five 1302 Disk Storage units with a storage capacity of up to 1,808 million positive digits.

IBM 1414 Model 6 Input-Output Synchronizer: This unit provides synchronizing and translating circuitry for telecommunication devices attached to telephone or telegraph facilities.

IBM 7150 Console: This unit provides manual control of the system, display of core-storage words, and typed output under control of the stored program.

IBM 7301 Core Storage: The 7070-7074 contains magnetic core storage of either 50,000 numeric digits



* A 7750 Programmed Transmission Control can be attached to an available 7907 channel.

Figure 1. System Components

Page Missing From Original

Character	Card Code	Core Storage 2-Digit Alphamerical Code	Magnetic Tape BCD Code	Core Storage 1-Digit Numerical Code	Notes
Blank		00	CA		8
	12-3-8	15	CBA821		
⌑ or)	12-4-8	16	BA84		
	12-5-8	17	CBA841		1
	12-6-8	18	CBA842		1
GM ‡	12-7-8	19	BA8421		1
& or +	12	20	BA		
$	11-3-8	25	B821		
*	11-4-8	26	CB84		
	11-5-8	27	B841		1
	11-6-8	28	B842		1
–	11	30	CB		7
/	0-1	31	A1		
,	0-3-8	35	A821		
% or (	0-4-8	36	CA84		
	0-5-8	37	A841		1
	0-6-8	38	A842		1
SM	0-7-8	39	CA8421		1, 5
# or =	3-8	45	C821		
@ or !	4-8	46	84		
	5-8	47	C841		1
	6-8	48	C842		1
TM	7-8	49	8421		1, 6
$\overset{+}{0}$	12-0	60	BA82		2
A	12-1	61	CBA1		
B	12-2	62	CBA2		
C	12-3	63	BA21		
D	12-4	64	CBA4		
E	12-5	65	BA41		
F	12-6	66	BA42		
G	12-7	67	CBA421		
H	12-8	68	CBA8		
I	12-9	69	BA81		
$\overset{-}{0}$	11-0	70	CB82		2
J	11-1	71	B1		
K	11-2	72	B2		
L	11-3	73	CB21		
M	11-4	74	B4		
N	11-5	75	CB41		
O	11-6	76	CB42		
P	11-7	77	B421		
Q	11-8	78	B8		
R	11-9	79	CB81		
RM ╪	0-2-8	80	CA82		3
S	0-2	82	A2		
T	0-3	83	CA21		
U	0-4	84	A4		
V	0-5	85	CA41		
W	0-6	86	CA42		
X	0-7	87	A421		
Y	0-8	88	A8		
Z	0-9	89	CA81		
0	0	90	82	0	7
1	1	91	C1	1	7
2	2	92	C2	2	7
3	3	93	21	3	7
4	4	94	C4	4	7
5	5	95	41	5	7
6	6	96	42	6	7
7	7	97	C421	7	7
8	8	98	C8	8	7
9	9	99	81	9	7
Delta Δ	11-7-8		CB8421		4
Alpha α	12-11				7

Figure 3. IBM 7070-7074 Coding System

NOTES:

1. Cannot be read into the IBM 7500 Card Reader, nor are they translated on output to the 7500 Card Punch, 7400 Printer, 7900 Inquiry Station, or the Console typewriter.
2. Cannot be read by the 7500 Card Reader unless they are wired as the units position of numeric words. Similarly, on punching or printing operations, these codes (60, 70) require special wiring.
3. This code cannot be wired to read or punch on the unit record equipment.
4. Generated by the system controls on write operations, and not translated on read operations. This card code (11-7-8) cannot be entered through the 7500 Card Reader.
5. The tape segment mark Write instruction (TSM) writes a single-character tape record that causes a condition six tape final status word (end of segment — EOS) when it is read; it is not placed in core storage. However, the tape segment mark can be read into core storage as a character within a tape record having a length of five or more characters. Also, the tape segment mark can be written from core storage as part of a tape record.
6. The tape mark is handled the same as the tape segment mark.
7. Only the numerals 0 through 9 and special characters +, –, and alpha can be read in the 7501 Console Card Reader. The 7500 Card Reader does not read the 12-11 alpha punch.
8. Note that no invalid (nontranslatable) 2-digit combinations are included. Any invalid 2-digit combination in core storage would be written on tape as a check (C) bit.

instructions in a sequential fashion; each instruction is located in a word with an address numbered one higher than the last instruction. The address of each succeeding instruction is obtained by means of the instruction counter in the CPU. The program can interrupt the sequence by changing the contents of the instruction counter, either directly or as the result of a logical decision.

INSTRUCTION FORMAT

Each instruction word in a 7070-7074 program consists of ten digits and sign. The sign of an instruction can be plus or minus, but not alpha. The digit positions are numbered from left to right: 0 1 2 3 4 5 6 7 8 9. The 0 position of a 7070-7074 word is the high-order or left-most position; 9 is the low-order or units position. The general format of the eleven positions in an instruction is:

S01	Operation code (S indicates sign)
23	Indexing word
45	Control
6789	Address

OPERATION CODE

The operation code (the sign in conjunction with positions 0-1) specifies the operation to take place. For example: the operation code +24, ADD TO ACCUMULATOR 2, adds the contents of the word specified by the address portion of the +24 instruction to whatever amount is already in accumulator 2. The sum of the operation remains in the accumulator.

Some of the operation codes are augmented by other digit positions in the instruction. In these *augmented* codes, the additional digit positions specify variations upon the basic operation code. For example +69, UNIT RECORD CONTROL, is used for card input, punched/ printed ouput, and console typewriter operations. The numeric value of the digit in position 5 of a +69 instruction defines the operation: read a card, punch/ print, etc. The numeric value of the digit in position 4 of a +69 instruction identifies the particular input or output unit involved.

CONTROL — FIELD DEFINITION

Many of the instructions process a portion of a word as easily as a full word. Positions 4 and 5 specify what part of the addressed word is to be used. The digit in position 4 denotes the starting position, the high-order position of the field. The digit in position 5 specifies the lower-order position. This is called *field definition.* A single position is field-defined by placing the same digit in positions 4 and 5 of the instruction. For example, 99 defines the units position of the word addressed. When field definition is used, the digit in position 4 of the instruction can never be higher in value than the digit in position 5. (Field definition does not cross word boundaries.)

The field definition feature permits several fields, with like sign, to be stored in a single word. Each field is readily available for processing. Whenever a portion of a word is operated upon in this manner, the sign of the word is the sign for all fields in the word.

CONTROL OTHER THAN FIELD DEFINITION

With most of the augmented codes, the control portion of an instruction (positions 4 and 5) does the augmenting — denoting the specific operation of the several that are indicated by the operation code.

In instructions that specifically operate on index words, positions 4 and 5 specify the index word to be operated on. (Positions 2-3 refer to the index word, and may be used to modify the address portion of the instruction.)

ADDRESS

The address portion of an intruction, positions 6-9, usually refers to the storage location of the data. In an accumulator addition operation, for example, positions 6-9 specify the address of the data to be added; in a store operation, the location in which the data are to be stored is specified. In branch operations, the address portion of the instruction may be the location of the next instruction. An example of this is +30, BRANCH IF ZERO IN ACCUMULATOR # (mnemonic BZ3). If a nonzero number is in accumulator 3 (regardless of sign), the program executes the instruction in the next sequential location. If the accumulator is zero, the next instruction comes from the location specified in the address portion of the BZ3 instruction.

In some operations, the address portion of an instruction contains the actual number to be processed, rather than a storage location. The four-digit number in positions 6-9 of the instruction is used as a factor in the operation. This number is always considered plus for these operations.

With some of the augmented codes, the address portion of the instruction provides the augmenting information.

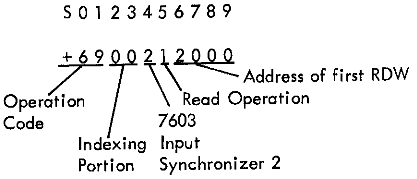
All input-output operation instructions contain the address of a record definition word in positions 6-9. Instructions that operate on more than one storage word use a record definition word to define a block of storage words.

Record Definition Words

A record definition word (RDW) contains the first and last addresses of a block of storage words to be read into, or written from. Positions 2-5 specify the location of the first word in the block and positions 6-9 specify the location of the last word in the block. For example, RDW +0031463155 defines a block of ten words in locations 3146 through 3155.

An RDW that has identical start and stop addresses defines a single word in core storage. For example, RDW +0044554455 specifies location 4455 only.

The record definition word itself is stored in a location specified in the instruction. For example, the unit record read instruction (UR) for a 7500 Card Reader operation has this format:



Where word 2000 contains −0030013016, execution of this UR instruction places the information from the card in core storage locations 3001 through 3016.

Note that the *first* RDW is specified in the instruction. If the RDW has a plus sign, the CPU automatically uses the RDW in the next sequential location for transmission of another block of data. In the example cited above, if the RDW in location 2000 has a plus sign, the CPU next uses the RDW in location 2001. An RDW with a minus sign is always the last RDW in the series; if the first RDW has a minus sign, it will be the only RDW used by the instruction.

A series of RDW's can define blocks of storage that do not adjoin each other. In the example below, the record definition words define a block of 20 words in locations 7320 to 7339, another block in locations 8450 to 8469, and a third block in locations 9240 to 9259. One instruction can move all three blocks.

RDW LOCATION	RDW CONTENT
	S 0 1 2 3 4 5 6 7 8 9
1000	+ 0 0 7 3 2 0 7 3 3 9
1001	+ 0 0 8 4 5 0 8 4 6 9
1002	− 0 0 9 2 4 0 9 2 5 9

Index Words

A single program or sub-program can be used to process varying batches of data by changing the address portion of the instructions involved. Index words in the 7070-7074 provide an efficient, flexible means of address modification. Indexing adds or subtracts a previously established number to or from the address portion of an instruction. These previously established number(s) are stored in core storage locations 0001-0099, and are called index words.

Positions 2 and 3 of an instruction specify which of the 99 index words is to be used. (Code 00 means no indexing.) When specified, indexing occurs as part of the execution of the instruction. Positions 2-5 and the sign of the designated index word are added algebraically to the address portion of the instruction, positions 6-9 (considered plus), and this new address is used for the operation.

The other six digit positions of index words are available to the programmer as storage; positions 6-9 of index words are often used for constants, decrements, and limits. Index word locations not needed in the program can be used as normal core storage words.

A minus index word causes the address in the instruction to be reduced by the value in the indexing portion of the index word (positions 2-5). If the indexing portion has a greater numeric value and is minus, the tens complement of the difference is obtained. For example, if positions 6-9 of an instruction contain 1875 and are indexed by the value −2000, the resultant address is 9875, rather than 0125 (9875 is the tens complement of 0125; 0125 + 9875 = 10,000).

Positions 6-9 of every instruction can be modified by indexing, regardless of whether they represent an address, a four-digit factor, are part of the operation itself, or are not used at all.

An index word can be plus or minus, but not alpha. An alpha index word specified in positions 2-3 of the instruction causes an error stop, whether positions 6-9 of the instruction are used or not.

Any time there is a value other than 00 in the index word portion (positions 2-3) of an instruction, time is taken for indexing. This is true even if instruction positions 6-9 are not used in the operation; the indexing operation takes place at the beginning of the execution of each instruction.

For most operation codes, indexing adds 36 microseconds to 7070 instruction execution time. However, there are 21 operations for which indexing adds only 24 microseconds:

+00	Halt and Branch (HB)
−00	Halt and Proceed (HP)
+01	Branch (B)
−01	No Operation (NOP)
−03	Sense Mode for Sign Change (SMSC)
−03	Halt Mode for Sign Change (HMSC)
−03	Branch if Sign Change (BSC)
−10, −20, −30	Branch if Minus in Accumulator # (BM1, BM2, BM3)
+11, +21, +31	Branch if Overflow in Accumulator # (BV1, BV2, BV3)
+40	Branch if Low (BL)
−40	Branch if High (BH)
+41	Branch if Field Overflow (BFV)
+41	Sense Mode for Field Overflow (SMFV)
+41	Halt Mode for Field Overflow (HMFV)
−41	Branch if Equal (BE)
+51	Branch if Alteration Switch On (BAS)
+51	Branch if Channel Busy (BCB)

On the 7074 system, indexing requires 5 microseconds.

Effective Address

The *effective address* of an instruction is the particular area of core storage actually addressed after the indexing word and field definition specifications have been taken into consideration. As many as eight positions of an instruction may be used to define the specific digit positions of specific core storage to be used by the instruction. Positions 6-9 contain an address. Positions 2-3 designate the presence or absence of an index word. Positions 4-5 define the digit positions that are to be used (09 in these positions denotes a full word).

Indicators

The IBM 7070-7074 contains ten indicators that are turned on automatically by conditions that occur during execution of the stored program:

Accumulator 1 overflow
Accumulator 2 overflow
Accumulator 3 overflow
Floating-decimal overflow
Floating-decimal underflow
Sign change
Field overflow
High (compare)
Equal (compare)
Low (compare)

An overflow condition occurs when the results of programmed calculations exceed the capacity of an accumulator or of the field specified to receive data. Floating-decimal overflow indicates an attempt to develop a characteristic greater than 99. Floating-decimal underflow indicates an attempt to develop a characteristic less than 00. Floating-decimal-point operations are available on the 7070-7074 as an optional feature.

The indicators can be tested at any time by the program. With the exception of the compare indicators, each indicator is automatically turned off by the operation that tests it. The compare indicators are reset by the next compare operation.

Electronic Switches

Three words in core storage are used as electronic switches. Each digit position in these words is considered to be a switch that is ON if it contains a digit 1 through 9; a zero indicates OFF. Specific instructions turn these switches on or off, and/or test their condition.

Alteration Switches

Four manual keys on the 7150 operator's console perform alteration switch functions. A specific instruction in the program tests the setting of one of the four alteration switches. If the switch is in the ON position, execution of the stored program is altered. This ability can be used to include or exclude portions of a program.

Instructions involving the use of alteration switches, indicators, and electronic switches may be found in the Logic Codes section of this manual.

Programming Features

Autocoder Programming Systems

Three levels of the Autocoder programming systems are provided for the IBM 7070-7074 Data Processing Systems. These are called Basic Autocoder, Four-Tape Autocoder and Autocoder.

BASIC AUTOCODER

Basic Autocoder is a programming system developed to simplify the preparation of programs for the 7070-7074. The major advantages of such a programming system are:

1. Each 7070-7074 operation has a mnemonic representation. For example, the programmer instructs the computer to store the quantity in accumulator 2 by writing ST2; this is the mnemonic for the STORE ACCUMULATOR 2 operation. The programmer does not need to know that the machine language code for the ST2 operation is +22.
2. Each instruction has a unique mnemonic representation, even though some machine-language operation codes (positions S, 0, 1) are the same. For example, the +69 card control operation code has seven different mnemonics.
3. Data to be processed are referred to symbolically, using names or other meaningful designations.
4. Instructions are not assigned core storage locations by the programmer; thus the addition and deletion of instructions entail no re-assignment of addresses.
5. Each routine or portion of a program can be written independently of the others with no loss of efficiency in the final program.

Writing a program in Basic Autocoder language relieves the programmer of many tedious clerical tasks. These tasks are turned over to the 7070-7074 and the Basic Autocoder Processor. The processor is a stored program, provided by IBM, that accepts the user's program written in Basic Autocoder language, translates the mnemonic codes into machine language, assigns core storage addresses to the instructions and the symbolic data references, and assembles a finished machine language program, called the *object* program.

Also, the processor checks for common coding errors, and notes these by means of console messages while continuing the translation process.

The Basic Autocoder Processor can assemble programs for use with any configuration of the 7070-7074 system. Translation is on a one-for-one basis; each line of coding produces a single instruction or storage assignment. Special statements to the processor program help the programmer to define areas of core storage and particular items within the area. The Basic Autocoder Processor requires only minimum system equipment: 5,000 words of core storage, one card reader, and one card punch. Basic Autocoder also works with one or two 729 Magnetic Tape Units. The addition of a printer makes it possible to obtain a listing of the assembled program.

FOUR-TAPE AUTOCODER

The Four-Tape Autocoder Processor includes the capabilities of Basic Autocoder and has substitution-type *macro-instructions* that can call for a series of instructions from a "library" of subroutines. It is used with systems having four or five magnetic tape units.

AUTOCODER

Autocoder is used with systems having at least six magnetic tape units. It contains powerful generator-type macro-instructions, extensive control operations over processing, program reassembly capabilities, multifile procedures, many output options, and a highly developed means of describing data.

Autocoder considers generator-type macro-instructions in relationship to the data involved in the operation and generates a set of instructions to handle the operation in the most efficient way. The generated program is comparable to what an experienced programmer would produce. Provision is made for the programmer to extend the Autocoder language by adding his own macro-instructions.

Program Testing

IBM utility programs provide the programmer with three types of frequently used routines. Some utility programs perform *housekeeping* functions such as setting storage to zeros or loading programs. Other utility programs handle file conversion problems such as tape to printer. Another type of utility program simplifies program testing.

By arranging a sequence of appropriate utility programs, a program may be tested with a minimum of manual operations at the console. The following 7070-7074 utility programs are available:

1. Condensed card load
2. Load program relocator
3. Zero storage program
4. Zero storage program (alternate)
5. Tape mark
6. Tape rewind
7. Tape file generator
8. Snapshot
9. Storage print
10. Tape print
11. Branch trace
12. Tape duplication
13. Tape compare
14. Unload storage
15. PAT (procedure for automatic testing)
16. TAT (tape auto testing program)

Fortran Systems

Fortran (Formula Translation) is a universal programming language long in use with IBM Data Processing Systems. Virtually any numeric procedure can be expressed in the Fortran language. Arithmetic formulas are stated in a language closely resembling that of mathematics. Fortran is a problem-oriented language; it is independent of the language format of any particular computer.

The IBM Basic Fortran program translates Fortran language statements (source program) into Autocoder (symbolic) language. Used in conjunction with the Basic Autocoder program, the symbolic statements produced by Basic Fortran are converted into a 7070-7074 machine language program (object program).

Basic Fortran requires a system with one IBM 7500 Card Reader and one IBM 7550 Card Punch. The Fortran program is more powerful than Basic Fortran and requires a system with six tape units.

Input-Output Control System

The IBM 7070 Input-Output Control System (IOCS) provides input and output routines for reading and writing card and tape records. The most important advantages of IOCS are reduced programming effort, efficiency, tested routines free of errors, and standardization of record handling.

IOCS automatically schedules simultaneous reading, writing, and processing functions. Macro-instructions are provided for processing data records that are to be written on, or read from, tape. Each data record is handled sequentially even though the data records may be blocked on tape. Blocking of output data records and deblocking of input data records is handled automatically by IOCS.

Through the use of tape label records written magnetically at the beginning and end of reels of tape, each reel is identified and checked before it is used in the program. IOCS error routines automatically correct tape and unit record errors whenever possible.

IOCS permits a program to be interrupted at any time and later resumed at the point of interruption by the use of checkpoint and restart routines.

The functions provided by IOCS are incorporated into the user's program during assembly by the IBM 7070 Autocoder Processor, or by the IBM 7070 Four-Tape Autocoder Processor.

Other programming features of the 7070-7074 are discussed in individual sections of this manual. They are: Block Transmission, Scatter Read-Write, Table Lookup, Priority Processing, and Floating-Decimal-Point Arithmetic (optional).

Machine Requirements for Programs and Programming Systems

Figure 4 specifies the input and output units required for the listed programs; the information does not refer to the user's program that is produced. For example, Fortran requires six tape units to process the user's source program, which, in turn, may require a different number of tape units; the object program produced may be written for use on a system that has only one tape unit. Similarly, programs written for a system having many tape units may be assembled with Basic Autocoder on a system without tape units.

Most of the programs allow a tape unit to be substituted for card input and output equipment. The programs which use magnetic tape units will function most efficiently when the tape units are connected to utilize two channels; however, the programs can be used with systems having only one tape channel.

Program Name	IBM 7500 Card Reader	IBM 7400 Printer	IBM 7550 Card Punch	IBM 729 Tape Unit	Additional Units or Features (Optional)
Basic Autocoder	1		1		1 IBM 7400 or IBM 7550
	1		1	1	
			1	2	1 IBM 7500
Four-Tape Autocoder				4	1 IBM 7500, 1 IBM 7400, 1 IBM 7550
Autocoder				6-10	1 IBM 7500, 1 IBM 7400, 1 IBM 7550
Basic Fortran	1		1		Floating Decimal
Fortran				6-10	1 IBM 7500, 1 IBM 7400, 1 IBM 7550, Floating Decimal
Input/Output Control System (Note A)					
Sort 90	1 (Note B)			4-16	
Merge 91	1 (Note B)			1-26	
Report Program Generator				6-10	1 IBM 7500, 1 IBM 7400, 1 IBM 7550
Spool System Two of these programs may be used simultaneously	1			1	
		1		1	
			1	1	
Utility Programs					
Condensed Card Load Load Program Relocator Zero Storage Programs Tape Mark and Tape Rewind	1 (Note B)				
Snapshot Storage Print Branch Trace	1 (Note B)	1 (Note C)			
Tape Print	1 (Note B)	1 (Note C)		1 min.	
Tape File Generator	1 (Note C)			1 min.	
Tape Duplication	1 (Note B)			2	
Tape Compare	1 (Note B)	1 (Note C)		2	
Punch Storage	1 (Note B)		1 (Note C)		
PAT	1 (Note B)	1 (Note C)			Tape Units
Simulation of the 650 on the 7070	Units must correspond to 650 system to be simulated				

Notes:

A. Requirements for assembly are the same as for the Autocoder system used to assemble the source program.

B. A tape unit or an IBM 7501 Console Card Reader may be substituted.

C. A tape unit may be substituted for this unit.

Figure 4. Input-Output Equipment Required for IBM 7070-7074 Applied Programming Programs

Optional Features

Additional Core Storage

This optional feature increases total storage capacity of a 7074 system from 50 to 200 percent by providing up to 20,000 additional words in increments of 5,000 or 10,000 words. With maximum additional storage, the 7074 has total storage capacity of 30,000 words. Core storage cycle time for additional storage is the same as for normal storage – four microseconds per ten-digit word.

Address Assignments

In a 7074 system with 10,000 words of storage, the four-digit storage addresses range from 0000 through 9999. Five-digit addresses are necessary, however, when the 7074 is operating in additional storage mode. Address assignments for storage of more than 10,000 words are:

Storage Capacity	General Storage	Acc 1	Acc 2	Acc 3	Prog Reg	Instr Counter
15,000 words	00000-14999	99991	99992	99993	99995	99999
20,000 words	00000-19999	99991	99992	99993	99995	99999
25,000 words	00000-24999	99991	99992	99993	99995	99999
30,000 words	00000-29999	99991	99992	99993	99995	99999

If an address is used that is greater than the range of addresses of the attached core storage modules, an illegal address error indication occurs.

Only 99991, 99992, and 99993 can be used as data addresses. Other uses of 9999x as an address result in an illegal address error indication.

Program Compatibility

Equipping the 7074 with additional core storage does *not* obsolete present programs with four-digit addresses. A special electronic switch, the additional storage switch, allows the 7074 to operate with either four- or five-digit addresses. The switch is program-controlled and must be on when the 7074 is executing instructions with five-digit addresses. When the switch is off, the 7074 recognizes only four-digit addresses. The new instruction that controls the switch is explained in detail under "Additional Storage Control Instruction." The additional storage switch turns off automatically when electrical power is first applied to the 7074 system. Whenever the switch is on, a new console light labeled "Additional Storage Mode" is also on.

Instruction Format for Five-Digit Addresses

When the 7074 is in normal storage mode (5,000 or 10,000 words), its instructions have the same format as 7070 instructions. In additional storage mode (15,000 to 30,000 words), the format is:

S01 No change.

23 Index word. The values 00, 01, 02 . . . 08, 09 in positions 2-3 specify *no* indexing. In this case, position 3 is the high-order (ten thousands) digit of a five-digit address whose four low-order digits are in positions 6-9. The values 10 through 99 in positions 2-3 specify indexing, and in this case the indexing portion of the specified index word is a *five*-position field (positions 1-5) that has the same sign as the index word. The algebraic addition of this five-position field and the four-position field, considered positive, in 6-9 of the instruction gives a five-digit effective address.

45 No change.

6789 This is the four-digit field that contains the four low-order digits of a five-digit address, if the instruction is not indexed; is added to the five-digit field in positions 1-5 of the specified index word, if the instruction is indexed.

A tape operation is unaffected by a change in mode during the operation.

Additional Storage Control Instruction

The new augmented code, +04, controls the additional storage switch by turning it on, turning it off, or testing it and branching if it is on. The format of the additional storage control instruction is:

S01 +04

23 Index word. See "Instruction Format for Five-Digit Addresses."

4 0, 1, or 2, depending upon the specific operation.
- 0 – BASS – Test the switch and branch if it is on.
- 1 – ASSN – Turn on the switch (additional storage mode).
- 2 – ASSF – Turn off the switch (normal storage mode).

5 Not used. Can contain any digit.

6789 This is the four-digit field that:

a. Contains the four low-order digits of a five digit address, if the instruction is not indexed.

b. Is added to the five-digit field in positions 1-5 of the specified index word, if the instruction is indexed.

If the index word is minus and the indexing portion is greater in value than that in positions 6-9 of the instruction, the tens complement of the difference is obtained.

For example:

Instr pos 6-9	9000	9000	9000
Index word pos 1-5	+12000	−12000	+07000
Address result	21000	97000	16000
Instr pos 6-9	9000	9000	9000
Index word pos 1-5	−07000	+99000	−99000
Address result	02000	08000	10000

In the second example, the address 97000 would cause an illegal address error indication.

7074 Timing: 6 microseconds.

Record Definition Words

Record definition words (RDW's) must be stored in addresses 00000 through 09999 when the 7074 is operating in additional storage mode; an attempt to locate an RDW outside this area causes an RDW program check. The RDW start address occupies positions 1-5 of the word, and positions 6-9 are the four low-order digits of the stop address. If the four-digit number in the stop address part of the RDW is smaller in value than the four low-order digits of the start address, the high-order (ten thousands) digit of the stop address is considered to be one greater than the high-order digit of the start address. For example, in the RDW−01 8750 8500, the full five-digit stop address is 28500. This mode of operation limits the size of a block of data referenced by an RDW to 10,000 words. Also, the check for start address greater than stop address does not function when the 7104 is operating in additional core storage mode.

During normal mode operation of an additional storage 7074, position one of RDW's is ignored in 7604 operations, but is processed in 7907 operations.

Additional Storage Status Words

For 729 tape operations in additional storage mode, the format of the initial status word (ISW) is unchanged in the sign position. Position 0 of the ISW defines the specific operation that occurs and positions 1-5 contain the address of the next sequential instruction after the tape instruction. Positions 6-9 of the ISW contain all significant digits of the address of the first record definition word used in the operation.

The final status word (FSW) format for 729 tape operations in additional storage mode is unchanged in positions S, 0, and 1. Positions 2-5 of the FSW contain the four low-order digits from the working address portion of the record definition register at completion of the operation. The high-order digit of the working address can be determined from the contents of the last record definition word used in the operation. Positions 6-9 of the FSW contain the four low-order digits *plus 1* of the address of the last record definition word used in the operation.

Priority Operations

At the time of a priority interrupt, if the machine is in additional storage mode, the contents of the five digit instruction counter are placed in positions one through five of index word 97. Upon execution of a priority release instruction, if the effective address in the PR instruction is 00097 and there is no priority waiting, the address in positions one through five of IW97 is sent to the instruction counter, and the next instruction is taken from that address. If the effective address in the PR instruction is *not* 00097 and there is no priority waiting, IW97 is not changed, the five-digit effective address is sent to the instruction counter, and the next instruction is taken from that address.

If another priority is waiting, and the effective address in the PR instruction is 00097, the waiting priority routine is started. IW97 is unchanged. If another priority is waiting, and the effective address is *not* 00097, positions one through five of IW97 are replaced by the effective address and the waiting priority routine is started.

Table Lookup

In all table lookup operations executed in additional storage mode, the location of the found table argument is placed in positions one through five of IW98 when the operation is ended. The remaining positions of IW98 are not changed, and its sign is set to plus.

Index Word Codes

Index word codes, which manipulate index words, have the same format in additional storage mode as other instructions. Some index word codes, however, operate in special ways when the 7074 is in additional storage mode. Also, note that only index words 10 through 99 are used in additional storage mode.

Branch Compared Index Word (−43 BCX): For the comparison, position 1 of the specified index word is ignored.

Branch Decremented Index Word (−49 BDX): For this operation, position 1 of the specified index word is not involved in the manipulation.

Branch Incremented Index Word (+49 BIX): Positions 1-5 of the specified index word are incremented by +1, but position 1 is ignored in the comparison of indexing portion with non-indexing portion. The no-branch condition occurs only when the difference between the indexing portion (2-5) and non-indexing portion (6-9) is +0001. Any other difference value causes the branch condition.

Branch and Load Location in Index Word (+02 BLX): The five digits from the instruction counter replace the contents of the indexing portion (1-5) of the specified index word.

Branch if Indexing Portion in Index Word Is Non-Zero (+44 BXN): For the non-zero test, position 1 of the specified index word is ignored.

Index Word Add to Indexing Portion (+47 XA): The five-digit address in the instruction is added algebraically to the indexing portion (1-5) of the specified index word.

Index Word Subtract from Indexing Portion (−47 XS): The five-digit address in the instruction is subtracted algebraically from the indexing portion (1-5) of the specified index word.

Index Word Zero and Add to Indexing Portion (+46 XZA): The five-digit address in the instruction replaces the contents of the indexing portion (1-5) of the specified index word, and the sign of the index word is set to plus.

Index Word Zero and Subtract from Indexing Portion (−46 XZS): The five-digit address in the instruction replaces the contents of the indexing portion (1-5) of the specified index word, and the sign of the index word is set to minus.

Index word codes BXM (−44), XSN (+48), XL (+45), XLIN (−48), and XU (−45) operate the same way in additional storage mode as in normal storage mode.

IBM 7150 Console

The 7150 Console of a 7074 system with additional core storage has two new features:

1. A control light labeled "additional storage mode" that turns on whenever the additional storage switch turns on. The light remains on as long as the 7074 is operating in additional storage mode.
2. A fifth address stop dial on the operating panel. This dial is used to specify the high-order (ten thousands) digit of the address stop address.

On a machine stop, five digits from the instruction counter and eleven digits with sign of the program register are printed out. The eleventh digit is the high-order digit of the data address and is printed after position 5 and before position 6 of the actual instruction. This digit is also printed on a manual display of the program register.

In order to display a core storage word while in additional storage mode, a five-digit address must be keyed.

Read Binary Tape Feature

This optional feature for the 7074 Data Processing System allows the 729 Magnetic Tape Units of the 7074 to read the odd-parity binary tapes prepared by IBM 704, 709, or 7090 systems or by analog-digital converters for these systems. The feature includes a new instruction (Tape Read Binary — TRN) and special binary-to-octal translating circuitry. Only tape channel 1 of the system can have the optional feature.

The binary-to-octal translating circuitry converts binary data in a tape column to two octal characters. Each octal character is represented by a decimal digit (0, 1, 2, 3, 4, 5, 6, or 7) in core storage. As shown in Figure 5, the high-order octal character represents the three upper binary data bit positions, and the low-order octal character represents the three lower binary data bit positions.

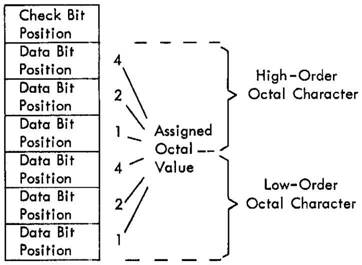


Figure 5. IBM 729 Octal Tape Character

Because of the one-for-two relationship between tape columns and octal characters, the information from five tape columns fills one core storage word. The machine automatically furnishes a plus sign for the octal storage word. Octal arithmetic operations in the decimal-logic 7074 must be based on programmed subroutines.

The format of the new Tape Read Binary (TRN) instruction is:

S Plus (normal condition priority) or minus (unusual condition priority).
0 Always the digit 8.
1 Always the digit 8.
23 Indexing word.
4 0-9 (729 tape unit).
5 Always the digit 1.
6789 Address of the first (or only) record definition word.

7074 Timing: 19 microseconds.

Interval Timer Feature

The interval timer, an optional feature for the 7070 and 7074 systems, is a program-controlled device for measuring data processing time.

Function

The timer records the length of an interval of time by adding "1" to the three-position timer register every 30 seconds while the 7070-7074 is in ready status. The register records up to 999 of the 30-second time increments — the equivalent of 8 hours and 19.5 minutes — and resets itself automatically after 999 increments. (The timer register also resets itself automatically whenever power is removed from the system.) The contents of the timer register can be examined at any time by means of a new augmented instruction. The instruction performs two functions, one of which is to store the contents of the register in a core storage word. The instruction places the three digits of the timer register in the three low-order positions (7-9) of the storage word, fills the remaining positions (0-6) with zeros, and gives the word a plus sign. The register contents remain unchanged. The contents of the storage word can be displayed on the 7070-7074 console typewriter in either of two ways: manually, by the operator, or by means of the TYP instruction. The other function of the new instruction is to reset the timer register to zeros.

Installing the interval timer feature on a 7070-7074 system requires the presence or concurrent installation of the console card reader attachment (Feature Code 239), but not necessarily the 7501 Console Card Reader.

Control Instruction

The format of the interval timer control instruction is:

S01 +69
23 Index word (01-99)
4 Always the digit 9
5 Operation
0 — ITZ — Interval Timer Zero
1 — ITS — Interval Timer Store
6789 Either (1) the address of an RDW with a minus sign if the operation is ITS, or (2) any combination of four digits if the operation is ITZ. For the ITS operation the start address of the RDW specifies the location of the storage word into which the register contents will be stored. The RDW stop address can consist of any four digits, providing it is greater than or equal to the start address.

Execution of the interval-timer-store operation requires 1.3 milliseconds, and if the operation occurs when the timer register is incrementing, up to 9 milliseconds additional time is required. The interval-timer-zero operation requires 60 microseconds (average). At completion of either the store or zero operation, the stored program advances to the next instruction in sequence; if an error occurs during either operation, the program halts.

Other Optional Features

Other optional features are discussed in other sections of this manual: Floating-Decimal-Point Arithmetic has its own section; additional 729 Tape Attachment and 729 Tape Switching with Switch Control Console are in the section on 729 Magnetic Tape Unit.

Operations Involving Accumulators

Included in this section are all the operations that involve the use of one or more of the accumulators, with the exception of BRANCH ON ACCUMULATOR contents or sign, and the floating-decimal instructions. (Table look-up operations use accumulator 3.)

Figure 6 is a categorized list of the codes in this section, with the operation codes, names, and Autocoder mnemonics.

Zero Accumulator # and Add
+13, +23, +33 ZA1, ZA2, ZA3

Machine Description: The specified accumulator is set to zero. The field-defined portion of the word addressed by positions 6-9 (indexable) is brought to the accumulator. The units position of the field always enters the units position of the accumulator, and the sign of the accumulator is made the same as that of the data word.

Instruction Format

S Always +.
0 Designates the accumulator: 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3.
1 Always 3.
23 Indexing word.
45 Field definition.
6789 Address of data word (indexable).

Examples: To move all of word 0500 to accumulator 1:

S01	*23*	*45*	*6789*
+13	00	09	0500

CATEGORIES	OP CODES	NAMES	MNEMONICS
Reset accumulator	+13, +23, +33	Zero accumulator # and add	ZA1, ZA2, ZA3
	−13, −23, −33	Zero accumulator # and subtract	ZS1, ZS2, ZS3
Addition and subtraction	+14, +24, +34	Add to accumulator #	A1, A2, A3
	−14, −24, −34	Subtract from accumulator #	S1, S2, S3
Multiplication	+53	Multiply	M
Division	−53	Divide	D
Absolute Value	+16	Zero accumulator 1 and add absolute	ZAA
	−16	Zero accumulator 1 and subtract absolute	ZSA
	+17	Add absolute to accumulator 1	AA
	−17	Subtract absolute from accumulator 1	SA
Store	−11, −21, −31	Zero storage and store accumulator #	ZST1, ZST2, ZST3
	+12, +22, +32	Store accumulator #	ST1, ST2, ST3
	−12, −22, −32	Store digits from accumulator # and ignore sign	STD1, STD2, STD3
Add to storage	+18, +28, +38	Add to storage from accumulator #	AS1, AS2, AS3
	−18, −28, −38	Subtract accumulator # from storage	SS1, SS2, SS3
	+19, +29, +39	Add to absolute storage from accumulator #	AAS1, AAS2, AAS3
Shift	+50	Shift control:	
		Shift right accumulator #	SR1, SR2, SR3
		Shift right and round accumulator #	SRR1, SRR2, SRR3
		Shift left accumulator #	SL1, SL2, SL3
		Shift left and count accumulator #	SLC1, SLC2, SLC3
	−51	Coupled shift control:	
		Shift right coupled	SR
		Shift right and round coupled	SRR
		Shift left coupled	SL
		Shift left and count coupled	SLC
		Shift right from point accumulator #	SRS
		Shift left from point accumulator #	SLS

Figure 6. Accumulator Operation Codes

After this operation is completed, the contents of word 0500 and accumulator 1 are identical.

To zero accumulator 2 and then add positions 3-5 of word 1645:

S01 23 45 6789
+23 00 35 1645

Contents of word 1645: −81345 60193. Contents of accumulator 2: −00000 00456, regardless of contents prior to instruction.

7074 Timing: 8 microseconds.

7070 Timing: The duration of this code depends on the number of digits in the field. A full word takes only 36 microseconds, however, because the digits do not have to be shifted in the arithmetic register (Figure 7).

Number of Digit Positions	1	2	3	4	5	6	7	8	9	10
Microseconds	36	48	48	48	60	60	60	72	72	36

Figure 7. IBM 7070 Timing — Zero Accumulator# and Add

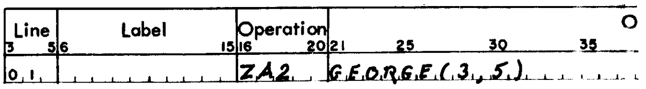
Comments: Although these codes involve accumulators, they are not really arithmetic operations; there is no addition or subtraction of two values. The accumulator always takes the sign of the data word—plus, minus or alpha, regardless of the size of the field defined.

Accumulator addresses (9991, 9992, or 9993) can be used, in which cases the addressed accumulator is treated as a core-storage word. If the same accumulator is used, as in ZA1 from accumulator 1, for example, the instruction has the effect of moving the field to the low-order portion and resetting the other positions. For example, if accumulator 1 has −54557 98643, and this instruction is given:

S01 23 45 6789
+13 00 03 9991

The result in the accumulator is −00000 05455. If field definition were 69 instead of 03, the result would be −00000 08643.

Autocoder Examples (Figures 8 and 9): The second example given for this code would be written symbolically as shown in Figure 8. The assembly program produces code +23 from the ZA2 operation. Field definition (3, 5) specifies positions 3-5 of an area previously defined as GEORGE. If GEORGE is a complete word, as in this example, field definition in the assembled instruction is the same (35). If GEORGE had been defined as the six low-order positions of a word, field definition would be 49. The address of that word is placed in positions 6-9 of the assembled instruction.



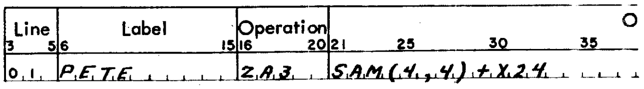
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0,1,		ZA2	GEORGE(3,5)

Figure 8

To reset accumulator 3 and add position 4 of SAM, indexed (instruction PETE; Figure 9): In the example, assume that SAM has been defined as a complete word and its location is 1789. Instruction PETE is assembled as:

S01 23 45 6789
+33 24 44 1789

The number in positions 2-5 of IW 24 is added algebraically to the 4-digit number 1789, each time this instruction is executed.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0,1,	PETE	ZA3	SAM(4,4)+X24

Figure 9

Zero Accumulator # and Subtract

−13, −23, −33 ZS1, ZS2, ZS3

Machine Description: The field-defined portion of the word addressed by positions 6-9 (indexable) is brought to the accumulator specified by position 0, to the low-order portion if less than 10 digits are defined. If the operand is + or −, the accumulator gets the opposite sign; if the operand is alpha, the accumulator gets an alpha sign.

Instruction Format: Same as ZA#, except for the sign.

Examples: To move all of word 0500 to accumulator 1 and change the sign:

S01 23 45 6789
−13 00 09 0500

Contents of 0500: +54380 02004. Contents of accumulator 1 after the operation: −54380 02004, regardless of previous contents.

To move the three high-order positions of word 1762 to accumulator 3, and change the sign:

S01 23 45 6789
−33 00 02 1762

Contents of 1762: −06000 00000. Contents of accumulator 3 after the operation: +00000 00060.

Timing: Same as ZERO ACCUMULATOR # AND ADD.

Comments: The sign of an accumulator can be changed by using this operation code with its own accumulator. The instruction:

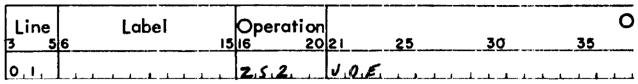
S01 23 45 6789
−23 00 09 9992

changes the sign of accumulator 2 from + to − or vice-versa. There is no change if the sign is alpha.

Whenever the sign of the operand is alpha, ZS# and ZA# produce the same result.

Autocoder Example (Figure 10): Assume that JOE has been defined as positions 0-5 of word 1551. Because no indexing or field definition is specified in the operand, the assembled instruction is:

S01	*23*	*45*	*6789*
−23	00	05	1551



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		Z S 2	J O E

Figure 10

Add to Accumulator

+14, +24, +34 A1, A2, A3

Machine Description: The field-defined portion of the word addressed by positions 6-9 (indexable) is added to the amount in the accumulator, to the low-order portion if less than 10 digits are defined. The signs of both factors are taken into account. The result is in the accumulator after the operation.

Instruction Format

S	Always +
0	Designates the accumulator: 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3.
1	Always 4
23	Indexing word
45	Field definition
6789	Address of data word (indexable)

Examples: To add the four low-order positions of word 0540 to accumulator 2:

S01	*23*	*45*	*6789*
+24	00	69	0540

To add all of word 1781 to accumulator 3:

S01	*23*	*45*	*6789*
+34	00	09	1781

7074 Timing: 8 microseconds *true*; 10 microseconds *complement.*

7070 Timing: The duration of this instruction depends on the number of significant digits in the result, and on whether the operation is a *true add* or *complement add.* If the result is the sum of the two numbers, the operation is called *true add.* If the result is the difference of the two numbers, causing the digits in one of the factors to be complemented before entering the adder, the operation is called *complement add.* If a complement-add operation causes the accumulator to change its sign (because the original accumulator value was smaller in absolute value than the data from storage), a *recomplement* automatically takes place at the conclusion of the operation. For example:

Accumulator 2 contains the value +123
The storage word value is −127
The operation is: +24, add
to accumulator 2
The adder adds: 123
(tens complement of 127) 873
996

This is complement-add. The value 996 is the tens complement of the correct answer, −004. At the end of the operation, the 996 is recomplemented to −004.

Consider a complement-add operation that does *not* need recomplementing:

Accumulator 2 contains the value: −456
The storage-word value is: +421
The operation is: +24, add
to accumulator 2
The adder adds: 456
(tens complement of 421) 579
−(1) 035

Note that the result does not need recomplementing if the accumulator value is greater than that of the field-defined storage word. Note also that the carry 1 is not needed to obtain the correct result of +35, whereas a carry in a true-add operation is part of the total (+305 added to +759 is +1 064). Complement-add *without* recomplement thus takes less time than true-add with carry, which in turn takes less time than complement-add *with* recomplement.

Figure 11 indicates the duration in microseconds of accumulator addition/subtraction operations, as determined by field size.

Number of Digit Positions	1	2	3	4	5	6	7	8	9	10
True Add to Accumulator	48	48	48	60	60	60	72	72	72	84
Complement Add to Acc.	36	48	48	48	60	60	60	72	72	72
If Recomplement	60	60	72	84	84	96	108	108	120	132

Figure 11. IBM 7070 Timing — Add to Accumulator#

Comments: An add operation obtains either the sum or the difference of the two factors, depending on the signs: the sum if the signs are the same, the difference if they are different. Three factors influence the value and sign of the result in an add instruction:

1. The sign of the storage word (+ or −)
2. The sign of the accumulator, prior to the operation (+ or −)
3. The operation (add or subtract)

Figure 12 is a chart showing the signs and values of the results of these combinations.

ORIGINAL SIGN OF ACCUMULATOR	SIGN OF DATA WORD	OPERATION	VALUE OF RESULT	SIGN OF RESULT
+	+	Add	Sum	+
+	−	Add	Difference	Sign of Greater Value
−	+	Add	Difference	Sign of Greater Value
−	−	Add	Sum	−

Figure 12. Add to Accumulator# – Plus and Minus Factors

If either factor has an alpha sign its value is considered plus in an arithmetic operation. Regardless of the value of the result, the sign of the result is alpha if either factor is alpha (Figure 13).

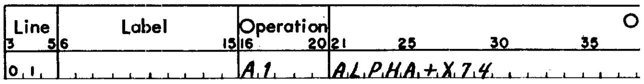
Any time the contents of an accumulator are brought to zero (by adding an amount equal to it but opposite in sign if neither sign is alpha), the sign is not changed. If a +7 from storage is added to a −7 in an accumulator, the result in the accumulator is −0.

ORIGINAL SIGN OF ACCUMULATOR	SIGN OF DATA WORD	OPERATION	VALUE OF RESULT	SIGN OF RESULT
+	Alpha	Add	Sum	Alpha
−	Alpha	Add	Difference	Alpha
Alpha	+	Add	Sum	Alpha
Alpha	−	Add	Difference	Alpha
Alpha	Alpha	Add	Sum	Alpha

Figure 13. Add to Accumulator# – Alpha Factors

Autocoder Example (Figure 14): Assume that ALPHA has been defined as the five high-order positions of word 0660. The assembled instruction is:

S01 23 45 6789
+14 74 04 0660



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		A 1	A L P H A + X 7 4

Figure 14

Subtract from Accumulator

−14, −24, −34 S1, S2, S3

Machine Description: The field-defined portion of the word addressed by positions 6-9 (indexable) is subtracted from the amount in the accumulator, from the low-order portion if less than ten digits are field-defined. The signs of both factors are taken into account. The result is in the accumulator after the operation.

Instruction Format: Same as A#, except for the sign.

Examples: To subtract the five high-order positions of word 0881 from the amount in accumulator 3:

S01 23 45 6789
−34 00 04 0881

To subtract the units position of word 2320 from the amount already in accumulator 1:

S01 23 45 6789
−14 00 99 2320

Timing: Same as for A#

Comments: A subtract operation obtains the difference of the two factors if the signs are the same, and the sum of the factors if the signs are different. As with the add operation codes, the result is influenced by the sign of the storage word and the sign of the accumulator, as well as by the subtract instruction (Figure 15).

ORIGINAL SIGN OF ACCUM.	SIGN OF DATA WORD	OPER.	VALUE OF RESULT	SIGN OF RESULT
+	+	Subtract	Difference	+ if acc. value greater − if storage value greater
+	−	Subtract	Sum	+
−	+	Subtract	Sum	−
−	−	Subtract	Difference	+ if storage value greater − if acc. value greater

Figure 15. Subtract from Accumulator# – Plus and Minus Factors

If either factor is alpha, its value is considered plus in determining true or complement add. It makes the sign of the result alpha in all cases (Figure 16).

ORIGINAL SIGN OF ACCUMULATOR	SIGN OF DATA WORD	OPERATION	VALUE OF RESULT	SIGN OF RESULT
+	Alpha	Subtract	Difference	Alpha
−	Alpha	Subtract	Sum	Alpha
Alpha	+	Subtract	Difference	Alpha
Alpha	−	Subtract	Sum	Alpha
Alpha	Alpha	Subtract	Difference	Alpha

Figure 16. Subtract from Accumulator# – Alpha Factors

Any time the contents of an accumulator are brought to zero (if neither factor is alpha), the sign is not changed. If a +7 in storage is subtracted from +7 in an accumulator, the result in the accumulator is +0.

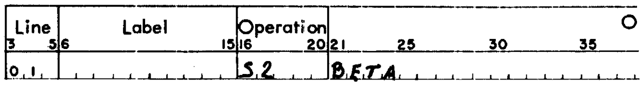
An accumulator can be brought to zero by a subtract instruction, with the address of that accumulator, regardless of its sign. For example, to bring accumulator 1 to zero:

```
S01  23  45  6789
-14  00  09  9991
```

The sign of the accumulator is unchanged.

Autocoder Example (Figure 17): Assume that BETA has been defined as positions 5-9 of word 3778. The assembled instruction is:

```
S01  23  45  6789
-24  00  59  3778
```



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		S 2	BETA

Figure 17

Multiply

+53 M

Machine Description: The arithmetic unit can multiply a ten-digit number by a ten-digit number to produce a 20-digit product. Either factor can be plus or minus, and the sign of the product conforms to the algebra rules: plus times plus equals plus, plus times minus equals minus, and minus times minus equals plus. If either sign is alpha, the sign of the product is alpha.

The multiplicand must be placed in accumulator 3 in a previous instruction. The multiply operation multiplies this number by the field-defined portion of the word addressed by positions 6-9 (indexable). The product is developed in accumulators 1 and 2, but is not affected by the previous contents of either accumulator; they are automatically reset before the multiply operation begins.

At the completion of the operation, the low-order portion of the product is in accumulator 2, with the units position in the units position of the accumulator. (If the product is ten digits or less, the entire product is in accumulator 2.) The multiplicand is in accumulator 3 at completion of the operation.

Instruction Format

S01	+53
23	Indexing word
45	Field definition of the multiplier
6789	Core-storage address of the multiplier

Examples: To multiply the 10-digit number in word 1230 by the 10-digit number in word 1735:

```
S01  23  45  6789
+33  00  09  1230  (multiplicand to acc. 3)
+53  00  09  1735  (20-digit product, devel-
                   oped in acc. 1 and 2)
```

To multiply positions 0-6 of word 3310 by positions 3-9 of word 3316:

```
S01  23  45  6789
+33  00  06  3310  (multiplicand to acc. 3)
+53  00  39  3316  (14-digit product, 10 low-
                   order positions in acc. 2, 4
                   high-order positions in acc.
                   1)
```

The duration of a multiply operation depends on the values of the multiplier digits as well as the number of them. The following formulas apply.

7074 Timing: In microseconds:

Exact: $26 + 2N^1 + 4N^2 + 6N^3$

Average: $26 + 3.4N$

N^1 = number of 1's, 2's, and 4's in multiplier
N^2 = number of 3's, 5's, 6's, and 8's in multiplier
N^3 = number of 7's and 9's in multiplier
N = number of significant digits in multiplier

7070 Timing: In microseconds:

$48(N^1 + 2N^2 + 3N^3) + 4N^4 + 4N^5 + 180$

In multiplier:

N^1 = number of 1, 2, 4 digits
N^2 = number of 3, 5, 6, 8 digits
N^3 = number of 7, 9 digits
N^4 = number of zeros
N^5 = number of zero groups

If the total is not a multiple of 12, use next higher multiple of 12.

A zero group is a zero or two-or-more adjacent zeros. For timing purposes, the multiplier is considered a ten-digit number. Therefore, if field definition specifies a multiplier of less than ten digits, the high-order zeros constitute a zero group. Also, the number of zeros in the multiplier is equal to the difference between ten digits and the number of positions field-defined.

Comments: A number is multiplied by itself, if accumulator 3 is the address of the multiply instruction. If the value N is in word 0504, these two instructions obtain N^2:

```
     S01  23  45  6789
ZA3  +33  00  09  0504
M    +53  00  09  9993
```

If this instruction is next, N^3 is obtained:

```
M    +53  00  09  9992
```

Repeating this instruction obtains N^4, etc. If the contents of accumulator 2 are stored in accumulator 3, after each multiplication, the values obtained are N^2, N^4, N^8, etc., as long as the product values do not exceed ten digit positions in size.

Polynomial nesting is the term given to adding variables to increasing powers of a constant:

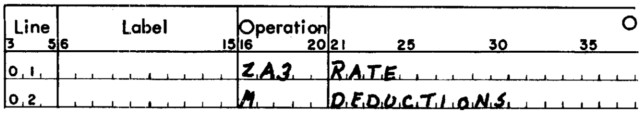
$$a + bx + cx^2 + dx^3 + ex^4$$

This can be performed efficiently on the 7070, because of the fact that the multiplicand (x) needs to be brought to accumulator 3 only once. Each product is the multiplier to obtain the next product.

INSTRUCTIONS	RESULT IN ACCUMULATOR 2
Zero-add x to Acc. 3	
Multiply by e	ex
Add d to Acc. 2	$d + ex$
Multiply x by contents of Acc. 2	$dx + ex^2$
Add c to Acc. 2	$c + dx + ex^2$
Multiply x by contents of Acc. 2	$cx + dx^2 + ex^3$
Add b to Acc. 2	$b + cx + dx^2 + ex^3$
Multiply x by contents of Acc. 2	$bx + cx^2 + dx^3 + ex^4$
Add a to Acc. 2	$a + bx + cx^2 + dx^3 + ex^4$

Autocoder Example (Figure 18): Assume that RATE has been defined as positions 0-4 of word 2468, and DEDUCTIONS as positions 8 and 9 of the same word. The assembled instructions are:

S01	*23*	*45*	*6789*
+33	00	04	2468
+53	00	89	2468



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		ZA3	RATE
0 2		M	DEDUCTIONS

Figure 18

Divide

−53 **D**

Machine Description: A 20-digit dividend is divided by a ten-digit divisor, to obtain a ten-digit quotient and a ten-digit remainder, with full sign control. Prior to the divide instruction, the dividend must be in accumulators 1 and 2, with the high-order portion in accumulator 1. The sign of accumulator 1 determines the sign of the dividend, in all cases. The divisor is the field-defined portion of the word addressed by positions 6-9 (indexable).

The absolute value of the divisor must be greater than the absolute value of the portion of the dividend in accumulator 1, at the start of the divide operation. If this rule is not observed, the machine attempts to develop a quotient of more than ten digits, causing the quotient to overflow and execution of the program to stop.

At the conclusion of the divide operation, the quotient is in accumulator 2, the remainder is in accumulator 1, and the divisor is in accumulator 3.

Instruction Format

S01	−53
23	Indexing word
45	Field definition of divisor
6789	Address of divisor (indexable)

Examples: To divide a 20-digit number, in words 1656 and 1657 (high-order portion of dividend in 1656), by positions 5-9 of word 1707:

S01	*23*	*45*	*6789*
+13	00	09	1656
+23	00	09	1657
−53	00	59	1707

To divide a 12-digit number by a four-digit number; the ten low-order positions of the dividend are in word 2400, indexed by positions 2-5 of IW 62. The two high-order dividend digits are in positions 8-9 of the next word, 2401 indexed by IW 62; and the divisor is in positions 4-7 of that same word:

S01	*23*	*45*	*6789*
+23	62	09	2400
+13	62	89	2401
−53	62	47	2401

7074 Timing: In microseconds:

Exact: $26 + 2N^0 + 6N^1 + 8N^2$

Average: $26 + 2N^0 + 6.8N$

N^0 = number of high order zeros in quotient

N^1 = number of 1's, 2's, 3's, 8's, 9's, and 0's (except high order zeros) in quotient

N^2 = number of 4's, 5's, 6's, and 7's in quotient

N = number of significant digits in quotient

7070 Timing: Division timing is determined by the sum of the quotient digits. For example, the sum of the digits in a quotient of 51326 is: $5 + 1 + 3 + 2 + 6 = 17$. The formula is as follows: Total number of microseconds $= 192 + 48\ (10 +$ sum of the quotient digits).

Comments: A divide operation always divides the entire contents of accumulator 3 into the entire contents of accumulators 1 and 2. For this reason, accumulators 1 and 2 must each be programmed to contain part of the dividend, or all zeros, prior to the divide instruction. Note that the sign of accumulator 2 in a divide operation is ignored; the sign of accumulator 1 is considered the sign of the entire 20-digit dividend. If the dividend is ten digits or less, and goes entirely in accumulator 2 for the divide operation, accumulator 1 must be set to zero and given the correct sign.

Decimal-point alignment is often the determining factor in positioning the dividend in accumulators 1 and 2. The dividend can be positioned correctly by use of this formula, in which D represents the number of digit positions to the *right* of the decimal:

D in the divisor
+D desired in the quotient
=D in the dividend

If the quotient is to be half-adjusted, the dividend should have one additional position to the right of the decimal. For example, total hours are divided into total miles, to obtain velocity:

The hours value has two decimal places:	hh.hh
It is desired to obtain velocity to three places:	vv.vvv
The miles are to two places:	mmm.mm

D in divisor h = 2
D desired in quotient v = 3
+1 for half-adjusting = 1
total D needed for dividend m = 6

The dividend itself has only two positions to the right of the decimal. Therefore, it should be positioned: mmm.mm0000.

The greater the dividend can be made in relation to the value of the divisor, the more exact the quotient will be. For example, the ten-digit number 00000 22000, divided by 00000 00007, is 00000 03142. The ten-digit number 22000 00000, divided by 00000 00007, is 03142 85714. Keep in mind, however, that the absolute value of that portion of the dividend in accumulator 1 must be less than the absolute value of the divisor. (Accumulator 1 cannot be used for the divisor, because of this rule; accumulator 2 or 3 can, however.)

The sign of the quotient is determined by the signs of the divisor and dividend. If they are the same, the quotient is plus; if they are different, the quotient is minus. The remainder always has the same sign as the dividend (Figure 19).

If either the divisor or dividend is alphabetic, the quotient is alphabetic (Figure 20).

If the divisor is:	and the dividend is:	the quotient is:	and the remainder is:
+	+	+	+
+	−	−	−
−	+	−	+
−	−	+	−

Figure 19. Divide—Plus and Minus Factors

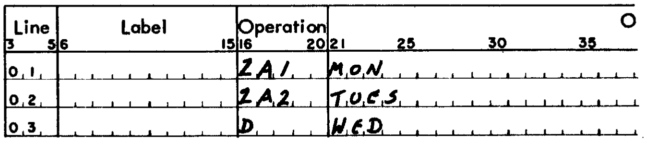
If the divisor is:	and the dividend is:	the quotient is:	and the remainder is:
+	Alpha	Alpha	Alpha
−	Alpha	Alpha	Alpha
Alpha	+	Alpha	+
Alpha	−	Alpha	−
Alpha	Alpha	Alpha	Alpha

Figure 20. Divide—Alpha Factors

Autocoder Example (Figure 21):
MON: Positions 6-9 of word 2101
TUES: Word 2102
WED: Positions 0-5 of word 1789

The assembled instructions are:

S01	*23*	*45*	*6789*
+13	00	69	2101
+23	00	09	2102
−53	00	05	1789



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		ZA1	MON
0 2		ZA2	TUES
0 3		D	WED

Figure 21

Zero Accumulator 1 and Add Absolute

+16 **ZAA**

Machine Description: The field-defined portion of the storage word addressed by positions 6-9 (indexable) is brought to accumulator 1, to the low-order portion if less than ten digits are field-defined. Regardless of the storage-word sign, +, −, or alpha, the accumulator becomes plus.

Instruction Format

S01	+16
23	Indexing word
45	Field definition
6789	Address of data word

Examples: To move the four high-order positions of word 1689 to accumulator 1, and give it a plus sign:

S01	*23*	*45*	*6789*
+16	00	03	1689

Contents of word 1689: −34517 78400. Contents of accumulator 1 after operation: +00000 03451, regardless of previous contents.

Timing: Same as ZA#.

Comments: This instruction can be used with accumulator 1 only. Thus, the number 1 is not needed in the Autocoder mnemonic. The value in accumulator 1 can be guaranteed to have a plus value, regardless of its previous sign, if 9991 is the address of a ZAA instruction:

S01	*23*	*45*	*6789*
+16	00	09	9991

Autocoder Example (Figure 22): The actual address 1537 is used. The assembled instruction is:

S01	*23*	*45*	*6789*
+16	00	59	1537

Line	Label	Operation	O
3 5	6 15	16 20	21 25 30 35
01		ZAA	1537(5,9)

Figure 22

Zero Accumulator 1 and Subtract Absolute

−16 **ZSA**

Machine Description: The field-defined portion of the storage word addressed by positions 6-9 (indexable) is brought to accumulator 1, to the low-order portion if less than 10 digits are field-defined. Regardless of the storage-word sign, +, −, or alpha, the accumulator becomes minus.

Instruction Format: Same as ZAA, except for the sign.

Example: To move all of word 1648 to accumulator 1, and make it minus:

S01 *23* *45* *6789*
−16 00 09 1648

Contents of word 1648: −46175 81421. Contents of accumulator 1 after the operation: −46175 81421, regardless of previous contents.

Timing: Same as ZA#.

Comments: This instruction can be used with accumulator 1 only. Thus, the number 1 is not needed in the Autocoder mnemonic.

Autocoder Example (Figure 23): The assembled instruction is:

S01 *23* *45* *6789*
−16 39 09 1919

Line	Label	Operation	O
3 5	6 15	16 20	21 25 30 35
01		ZSA	1919+X39

Figure 23

Add Absolute to Accumulator 1

+17 **AA**

Machine Description: The field-defined portion of the word addressed by positions 6-9 (indexable) is added to the number already in accumulator 1. The number from storage is always considered plus, regardless of its sign in storage; the number in accumulator 1 can be plus, minus, or alpha. The result is in the accumulator after the operation. If accumulator 1 is alpha, the result is alpha.

Instruction Format

S01	+17
23	Indexing word
45	Field definition
6789	Address of data word

Examples: To add the absolute value of positions 5-9 of word 1617 to the amount already in accumulator 1:

S01 *23* *45* *6789*
+17 00 59 1617

Contents of accumulator 1 before the operation: +00000 05456. Contents of word 1617: −00000 00302. Contents of accumulator 1 after the operation: +00000 05758.

To add the absolute value of positions 0-2 of word 3735 to the amount already in accumulator 1:

S01 *23* *45* *6789*
+17 00 02 3735

Contents of accumulator 1 before the operation: −00000 00021. Contents of word 3735: −03500 00000. Contents of accumulator 1 after the operation: +00000 00014.

Timing: Same as ADD TO ACCUMULATOR #. True-add or complement-add is determined solely by the sign of the accumulator value: if it is plus or alpha, the operation is true-add; if it is minus, the operation is complement-add.

Comments: This operation can be used with accumulator 1 only. The sum of the two values is obtained if the accumulator is plus or alpha prior to the operation, and the difference is obtained if the accumulator is minus. If the storage-word sign is alpha, its value is considered plus, but its sign is ignored. If the accumulator is alpha, its value is considered plus, and the resultant sign is alpha.

Autocoder Example (Figure 24): Assume that BETA has been defined as positions 4-7 of word 1671.

The assembled instruction is:

S01 *23* *45* *6789*
+17 75 47 1671

Line	Label	Operation	O
3 5	6 15	16 20	21 25 30 35
01		AA	BETA+X75

Figure 24

Subtract Absolute from Accumulator 1

−17 **SA**

Machine Description: The field-defined portion of the word addressed by positions 6-9 (indexable) is added to the number already in accumulator 1. The number from storage is considered minus, regardless of its sign in storage. The number in accumulator 1 can be plus, minus, or alpha. The result is in the accumulator after the operation.

Instruction Format: Same as AA, except for the sign.

Examples: To subtract the absolute value of positions 7-9 of word 3331 from the amount already in accumulator 1:

S01 *23* *45* *6789*
−17 00 79 3331

Contents of accumulator 1 before the operation: +00000 00456. Contents of word 3331: +63457 19406. Contents of accumulator 1 after the operation: +00000 00050.

To subtract the absolute value of word 3518 from accumulator 1:

S01 *23* *45* *6789*
−17 00 09 3518

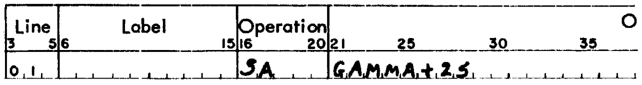
Contents of accumulator 1 before the operation: +00000 01234. Contents of word 3518: +00000 71234. Contents of accumulator after the operation: −00000 70000.

Timing: Same as ADD TO ACCUMULATOR #. True-add or complement-add is determined solely by the sign of the accumulator value: if it is minus, the operation is true-add; if it is plus or alpha, the operation is complement-add.

Comments: The sum of the two values is obtained if the accumulator is minus prior to the operation, and the difference is obtained if the accumulator is plus or alpha. If the storage-word sign is alpha, its value is considered minus, and the sign is ignored. If the accumulator sign is alpha, its value is considered plus, and the resultant sign is alpha regardless of its value.

Autocoder Example (Figure 25): GAMMA has been defined as word 2550. The assembled instruction is:

S01 *23* *45* *6789*
−17 00 09 2575



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		SA	GAMMA+25

Figure 25

Field Overflow

Field overflow occurs in each of the following conditions:

1. A store or add-to-storage type of operation in which the number of significant digits in the accumulator is greater than field definition allows to be stored in the word.
2. An "add-to-storage" operation in which a carry is propagated beyond the high-order position of the field defined.

When either situation arises, only the digits that field definition allows are stored. The *field overflow* indicator is turned on, and the machine either continues or stops, depending on the setting of the field-overflow stop/sense switch. This switch is set to "sense" or "stop" by augmented code +41: sense mode for the field overflow (SMFV) and halt mode for field overflow (HMFV). If the field-overflow indicator is on, and the program changes the field-overflow stop/sense switch to STOP, the machine immediately stops. The operation that tests the indicator (BFV, branch if field overflow, +41), turns it off if it is on.

Sign Change

If a store or add-to-storage type of operation brings data to a field-defined portion of a storage word, and the remaining positions of the word are not set to zero, a difference in sign between the storage word and the data brought to it is detected by the machine. The field is stored in the word, and the sign is changed. The *sign-change* indicator is turned on, and the program either stops or continues, depending on setting of the sign-change sense/stop switch. This switch is set to SENSE or STOP by an augmented operation code, −03: sense mode on sign change (SMSC) and halt mode on sign change (HMSC). If the sign-change indicator is on, and the program changes the sign-control stop/sense switch to STOP, the machine immediately stops. The operation that tests the indicator (BSC, branch if sign change, −03), turns it off if it is on.

Zero Storage and Store Accumulator # −11, −21, −31 ZST1, ZST2, ZST3

Machine Description: The field-defined portion of the storage word addressed by positions 6-9 (indexable) is replaced by an equivalent number of digits from the specified accumulator, from the low-order portion if less than ten digits are field-defined. The remaining positions of the storage word are set to zero, and the sign is set to the sign of the accumulator. Field overflow is possible, but sign change is not.

Instruction Format

S	Always minus
0	Designates the accumulator: 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3.
1	Always 1
23	Indexing word
45	Field definition
6789	Address of storage word in which the data are to be stored

Examples: To zero storage word 1860 and store data from accumulator 2, into positions 3-4:

S01 *23* *45* *6789*
−21 00 34 1860

Contents of accumulator 2: +00000 00023. Contents of 1860 after the operation: +00023 00000, regardless of previous contents.

To zero storage word 2816 and store data from accumulator 1, into positions 1-4:

S01 *23* *45* *6789*
−11 00 14 2816

Contents of accumulator 1: −00000 54231. Contents of 2816 after the operation: −04231 00000, regardless of previous contents. The field-overflow indicator is turned on, because there are five significant digits in the accumulator and only four digits are field-defined.

7074 Timing: 8 microseconds full field; 12 microseconds not full field.

7070 Timing:

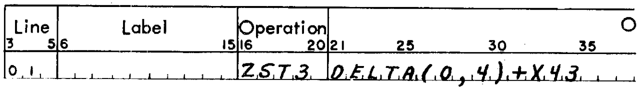
No. of Digit Positions	1	2	3	4	5	6	7	8	9	10
Microseconds	48	48	60	60	60	72	72	72	84	36

Comments: If there are more significant digits in the accumulator than field-definition allows to be stored, the *field-overflow* indicator is turned on. The correct number of digits is stored.

Note that the contents of the storage word prior to this operation are lost completely and have no effect on the contents of the word after the operation.

Autocoder Example (Figure 26): Delta has been previously defined as word 1500. The assembled instruction is:

S01 23 45 6789
−31 43 04 1500



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		ZST3	DELTA(0,4)+X43

Figure 26

Store Accumulator
+12, +22, +32 ST1, ST2, ST3

Machine Description: The field-defined portion of the storage word addressed by positions 6-9 (indexable) is replaced by an equivalent number of digits from the specified accumulator, from the low-order portion if less than ten digits are field-defined. The sign of the storage word is set to that of the accumulator. The remaining positions of the storage word are unchanged. Field-overflow and sign-change indications are both possible.

Instruction Format

S Always plus
0 Designates the accumulator: 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3
1 Always 2
23 Indexing word
45 Field definition
6789 Address of storage word in which the data are to be stored

Examples: To store data from accumulator 1 in positions 4-7 of word 1820, set the sign to that of accumulator 1, and not disturb other positions of the word:

S01 23 45 6789
+12 00 47 1820

Contents of word 1820 before the operation: −22222 33333. Contents of accumulator 1: +00000 00678. Contents of word 1820 after the operation: +22220 67833. The sign-change indicator is turned on, because the accumulator and storage-word signs were different.

To store one digit from accumulator 3 into position 6 of word 2000, indexed by positions 2-5 of iw 54:

S01 23 45 6789
+32 54 66 2000

Contents of the storage word before the operation: +43444 53355. Contents of accumulator 3: +00010 00009. Contents of the storage word after the operation: +43444 59355. The field-overflow indicator is turned on, because the digit 1 in position 3 of the accumulator indicates that the accumulator number is larger than storage-location field.

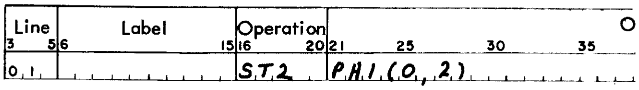
7074 Timing: 12 microseconds

7070 Timing: Same as zst#

Comments: Note that field definition locates the exact positions of the storage word to which the accumulator data go. It also defines the number of positions in the accumulator, starting from the right, that are to be stored. This is true of all store-accumulator instructions.

Autocoder Example (Figure 27): Assume that phi has been previously defined as the five low-order positions of word 4430. The assembled instruction is:

S01 23 45 6789
+22 00 57 4430



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		ST2	PHI(0,2)

Figure 27

Store Digits from Accumulator # and Ignore Sign
−12, −22, −32 STD1, STD2, STD3

Machine Description: The field-defined portion of the storage word addressed by positions 6-9 (indexable) is replaced by an equivalent number of digits from the specified accumulator, from the low-order portion if less than ten digits are field-defined. The remaining positions and sign of the storage word are unchanged.

This code differs from the store-accumulator instruction in that the sign of the accumulator is ignored. Thus, the sign-change indicator cannot be turned on by this instruction. Field overflow is possible, however.

Instruction Format: Same as st#, except for sign.

Examples: To store digits from accumulator 1 in positions 6-9 of word 4147, and ignore the sign of the accumulator:

S01 23 45 6789
−12 00 69 4147

Contents of word 4147 before the operation: @71857 59199. Contents of accumulator 1: +00000 09290.

Contents of word 4147 after the operation: @71857 59290. The sign-change indicator is not turned on, even though the signs are different. (Field overflow does not occur in this particular example.)

To store digits from accumulator 2 in positions 8-9 of word 3214 and ignore the sign of accumulator 2:

S01 23 45 6789
−22 00 89 3214

Contents of 3214 before the operation: −55555 55555. Contents of accumulator 2: +00000 00123. Contents of 3214 after the operation: −55555 55523. The field-overflow indicator is turned on, but the sign-change indicator is not.

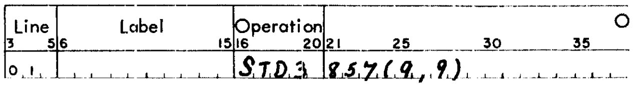
7074 Timing: 12 microseconds

7070 Timing: Same as ZST#

Comments: The only difference between this code and store accumulator is that this code cannot turn on the sign-change indicator if the accumulator and storage-word signs are different. If the signs are the same, this instruction is exactly the same as store accumulator.

Autocoder Example (Figure 28): The actual address 857 is used. The assembled instruction is:

S01 23 45 6789
−32 00 99 0857



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		STD3	857(9,9)

Figure 28

Add to Storage from Accumulator # +18, +28, +38 AS1, AS2, AS3

Machine Description: The field-defined portion of the storage word addressed by positions 6-9 (indexable) is added to the entire contents of the specified accumulator, and the result is stored in the same field of the storage word. The storage word is given the sign of the result. The accumulator value is unchanged. The sign of each factor is taken into consideration. Both sign change and field overflow are possible.

The sign-change indicator is not turned on, however, if a full word is field-defined (09 in positions 4-5). In this case, the sign of the storage word can be changed by a complement-add operation, but the sign-change indicator is not turned on. A sign change always occurs, regardless of factor signs, when the storage field is positions 1-9 of the word and the operation involves complement adding with recomplementing. A sign change always occurs, too, when (a) a numeric accumulator factor is complement-added with recomplementing to an alpha factor in storage and (b) an alpha accumulator factor is added to a numeric storage factor.

Instruction Format

S	Always +
0	Designates the accumulator: 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3
1	Always 8
23	Indexing word
45	Field definition
6789	Address of storage word that contains one of the factors prior to the operation, and the result, after the operation

Examples: To add the contents of accumulator 1 to the five low-order positions of word 3279, and store the result in those positions:

S01 23 45 6789
+18 00 59 3279

Contents of 3279 before the operation: +22222 51434. Contents of accumulator 1: −00000 00121, unchanged by the operation. Contents of 3279 after the operation: +22222 51313. Neither the field-overflow nor sign-change indicator is turned on.

To add the contents of accumulator 2 to the three low-order digits of word 2754, and store the result in those positions:

S01 23 45 6789
+28 00 79 2754

Contents of 2754 before the operation: +01230 88454. Contents of accumulator 2: +00000 01111, unchanged by the operation. Contents of 2754 after the operation: +01230 88565. The field-overflow indicator is turned on, because of the presence of four significant digits in the accumulator. There is no sign change.

To add the contents of accumulator 3 to the entire contents of word 1547:

S01 23 45 6789
+38 00 09 1547

Contents of 1547 before the operation: +00000 00567. Contents of accumulator 3: −00000 00670, unchanged by the operation. Contents of 1547 after the operation: −00000 00103. Although the sign of the storage word is changed by the operation, the sign-change indicator is not turned on, because a full word was field-defined.

7074 Timing: 12 microseconds true; 14 microseconds complement.

7070 Timing: As in the add- and subtract-to-accumulator codes, the timing of an add-to-storage instruction is determined by the number of digits field-defined and by whether the operation is true-add, complement-add, or complement-add with recomplement.

In add-to-storage operations, recomplement is required for complement-add operations in which the accumulator value is greater than the storage-word value.

Timing of add-to-storage operations is shown in Figure 29.

Number of Digit Positions	1	2	3	4	5	6	7	8	9	10
	MICROSECONDS									
True Add to Storage	48	48	60	60	60	72	72	72	84	84
Complement Add to Storage	48	48	60	60	60	72	72	72	84	84
If Recomplemented	60	72	84	84	96	108	108	120	132	132

Figure 29. IBM 7070 Timing – Add to Storage from Accumulator#

Comments: An add-to-storage type of instruction reverses the normal functions of storage word and accumulator. The accumulator contains data to be added, and the storage word becomes the "accumulator." It contains one of the factors to be added, and, after the operation, it contains the result.

The signs of both factors are considered; the sum is obtained if they are the same, and the difference is obtained if they are different. If either sign is alpha, its value is considered plus; but the sign of the result is alpha regardless of its value.

The storage word always takes the sign of the result. If less than a full word is specified by field definition, and the sign of the result is different from the sign of the original storage word, a sign-change indication is given. If the operation causes a carry to a position beyond the field defined, or the accumulator contains a larger number than field definition allows, a field-overflow indication is given.

Autocoder Example(Figure 30): KAPPA has been defined as word 1929. The assembled instruction is:

S01 23 45 6789

+38 00 59 1929

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		AS3	KAPPA(5,9)

Figure 30

Subtract Accumulator # from Storage

−18, −28, −38 SS1, SS2, SS3

Machine Description: The contents of the specified accumulator are subtracted from the field-defined portion of the storage word addressed by positions 6-9 (indexable) of the instruction. The result is stored in the same storage word, and the sign of the word is set to the sign of the result. The accumulator remains unchanged. Sign-change and field-overflow are both possible. The sign-change indicator is not turned on if a full storage word is field-defined and has its sign changed. If the subtraction involves an accumulator factor that is positive or alpha and a storage factor that is alpha and of lesser magnitude, the sign change indicator will be turned on. The number of accumulator digits involved in the subtract operation is equal to the number of digits specified by field definition. If there are significant accumulator digits to the left of those involved in the operation, the field-overflow indicator is turned on. Field-overflow occurs also if the result causes a carry beyond the high-order position of the storage word field.

Instruction Format: Same as add-to-storage, except the sign.

Examples: To subtract the contents of accumulator 3 from all of word 0950:

S01 23 45 6789

−38 00 09 0950

Contents of 0950 before the operation: +00000 12300. Contents of accumulator 3: +00000 30000, unchanged by the operation. Contents of 0950 after the operation: −00000 17700. Although the sign of the storage word is changed by the operation, the sign-change indicator is not turned on, because a full word was field-defined.

To subtract the contents of accumulator 2 from the two low-order positions of word 1781:

S01 23 45 6789

−28 00 89 1781

Contents of 1781 before the operation: +00000 00045. Contents of accumulator 2: +00000 00145, unchanged by the operation. Contents of 1781 after the operation: +00000 00000. The field-overflow indicator is turned on, because of three significant digits in the accumulator, with only two digits specified by field definition. There is no sign change.

Timing: Same as AS#

Comments: This instruction has the same relation to add-to-storage as a subtract operation has to add: the operation obtains either the sum or the difference of the two values, depending on the sign of each factor. If the signs are the same, this code obtains the difference. If the signs are different, it obtains the sum.

Autocoder Example (Figure 31): Actual address 1984 is used. The assembled instruction is:

S01 23 45 6789

−18 55 03 1984

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		SS1	1984(0,3)+X55

Figure 31

Add to Absolute Storage from Accumulator
+19, +29, +39 AAS1, AAS2, AAS3

Machine Description: The field-defined portion of the storage word addressed by positions 6-9 (indexable), considered plus, is added algebraically to the entire contents of the specified accumulator. The result is stored in the same field of the storage word; the sign of the storage word is unchanged. The accumulator is unchanged by the operation. Field overflow is possible, but sign change is not.

Instruction Format

S	Always +
0	Designates the accumulator: 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3
1	Always 9
23	Indexing word
45	Field definition
6789	Address of storage word that contains one of the factors prior to the operation, and the result, after the operation. (Value of the factor considered plus, regardless of sign of the storage word.)

Examples: To add the contents of accumulator 3 to the absolute value in positions 5-9 of word 2350, and store the result in those positions:

S01	*23*	*45*	*6789*
+39	00	59	2350

Contents of word 2350 before the operation: −98765 34343. Contents of accumulator 3: +00000 22222, unchanged by the operation. Contents of word 2350 after the operation: −98765 56565.

To add the contents of accumulator 1 to the absolute value in position 9 of word 1975, and store the result in that position:

S01	*23*	*45*	*6789*
+19	00	99	1975

Contents of 1975 before the operation: −00000 00006. Contents of accumulator 1: +00000 00008. Contents of 1975 after the operation: −00000 00004. The field-overflow indicator is turned on.

Timing: Same as AS#

Comments: This is an unusual operation code, because in some circumstances it goes against the rules of arithmetic. The reason for this is that the value in the storage word is considered plus regardless of its sign, and its sign is never changed, regardless of the result. Figure 32 is a chart of the combinations of two values, 8 and 6, and the results obtained by this code, when one position is field-defined.

Note that the results are the same for each combination of signs, when the values are reversed. If either factor is alpha, its value is considered plus.

The purpose of this operation code is modification of instructions. Because an instruction can have either a plus or minus sign, the value of the result would be affected by the sign, if an AS# or SS# instruction were used.

VALUE IN STORAGE	VALUE IN ACCUMULATOR	RESULT IN STORAGE	
+6	+8	+4	Field Overflow
+6	−8	+2*	
−6	+8	−4*	Field Overflow
−6	−8	−2*	
+8	+6	+4	Field Overflow
+8	−6	+2	
−8	+6	−4*	Field Overflow
−8	−6	−2*	
−4	@3	−7	
@4	+3	@7	

*These results are arithmetically incorrect

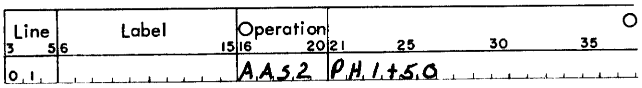
Figure 32. Results – Add to Absolute Storage from Accumulator

Note that neither factor ever has its sign changed by this operation.

Accumulator addresses can be used with this operation. If the instruction uses the same accumulator (AAS3 from accumulator 3 for example), the value in the accumulator is doubled if its sign is plus or alpha, or set to zeros if its sign is minus.

Autocoder Example (Figure 33): PHI has been previously defined as positions 6-9 of word 2500. The assembled instruction is:

S01	*23*	*45*	*6789*
+29	00	69	2550



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		AAS2	PHI+50

Figure 33

Shift Control
+50

Machine Description: The shift-control operations cause the contents of an accumulator to be moved to the right or left within the accumulator a specified number of positions. The digits moved off the end are lost, and the vacated positions at the other end are filled with zeros.

This operation code is augmented; the specific operation to be performed is defined in position 7 of the instruction.

Instruction Format

S01 +50

23 Indexing word. Even though positions 6-9 do not represent an address, they are modified by positions 2-5 of an IW, if one is specified here.

45 Index word, for shift and count operations. The number of places shifted (00-10) replaces positions 4-5 of the index word specified here. (Positions 2-3 of the index word are set to zero, and its sign is set to plus.) These positions are not used in the other shift operations.

6 1-3, designating the accumulator to be shifted

7 0-3, designating the type of shift:

0. Shift right accumulator #SR1, SR2, SR3
1. Shift right and round accumulator # SRR1, SRR2, SRR3
2. Shift left accumulator # SL1, SL2, SL3
3. Shift left and count accumulator # SLC1, SLC2, SLC3

89 00-10, designating the number of positions to be shifted. Not used in shift and count operations.

Shift Right Accumulator #: The entire contents of the accumulator designated in position 6 are shifted to the right the number of positions specified in positions 8-9 of the instruction.

Shift Right and Round Accumulator #: This code is the same as shift right, with the additional feature that the amount shifted is rounded off. A 5 is added to the highest-order digit that is to be dropped off the edge; and if there is a carry, the next digit is increased by 1.

Shift Left Accumulator #: This is the same as shift right, except that the digits are moved to the left. Zeros are inserted into the vacated low-order positions.

Shift Left and Count Accumulator #: The digits in the accumulator designated in position 6 are shifted to the left, until a digit other than a zero is in the high-order position. If the high-order digit is non-zero to start with, no shift takes place.

The number of positions shifted is recorded in an index word, specified in positions 4-5 of the instruction. This number (00-10) is recorded in positions 4-5 of the IW; positions 2-3 of the IW are set to zero, and the remaining positions are unchanged. The sign of the index word is set to plus.

Examples: To shift the contents of accumulator 2 five positions to the right:

S01	*23*	*45*	*6*	*7*	*89*
+50	00	xx	2	0	05

Contents of accumulator 2 before the operation: +12345 67890. Contents of accumulator 2 after the operation: +00000 12345.

To shift the contents of accumulator 2 five positions to the right, and round-off the number:

S01	*23*	*45*	*6*	*7*	*89*
+50	00	xx	2	1	05

Contents of accumulator 2 before the operation: +12345 67890. Contents of accumulator 2 after the operation: +00000 12346.

To shift the contents of accumulator 3 one position to the left:

S01	*23*	*45*	*6*	*7*	*89*
+50	00	xx	3	2	01

Contents of accumulator 3 before the operation: −12000 00000. Contents of accumulator 3 after the operation: −20000 00000.

To shift left and count in accumulator 1:

S01	*23*	*45*	*6*	*7*	*89*
+50	00	17	1	3	yy

Contents of accumulator 1 before the operation: −00002 40390. Although positions 8-9 are not used, they must contain a valid shift number (00-10). Contents of accumulator 1 after the operation: −24039 00000. Contents of IW 17 before the operation: −00 2425 0015. Contents of IW 17 after the operation: +00 0004 0015.

7074 Timing: In microseconds:

SR#	6 + (N−3)*
SRR#	6 + N
SL#	6 + (N−3)*
SLC#	14 + (N−3)*

N = number of positions to be shifted

*Note that if N<3, (N−3) = 0

7070 Timing: See Figure 34.

OPERATION	NOT INDEXED		INDEXED	
	Number of Positions Shifted	Time (μs)	Number of Positions Shifted	Time (μs)
SR# or SL#	0, 1, 2	36	0	60
	3, 4, 5	48	1, 2, 3	72
	6, 7, 8	60	4, 5, 6	84
	9, 10	72	7, 8, 9	96
			10	108
SRR#	0	36	0	60
	1	84	1, 2	120
	2, 3, 4	96	3, 4, 5	132
	5, 6, 7	108	6, 7, 8	144
	8, 9, 10	120	9, 10	156
SLC	0	84	0	120
	1, 2	96	1, 2, 3	132
	3, 4, 5	108	4, 5, 6	144
	6, 7, 8	120	7, 8, 9	156
	9, 10	132	10	168

Figure 34. IBM 7070 Timing – Shift Control

Comments: A +50 shift operation never changes the sign of the accumulator. A shift of ten positions sets the entire accumulator to zeros but does not change the sign. It has the same result as an operation that subtracts the contents of an accumulator from itself.

A shift in excess of ten positions (11-99 in positions 8-9) is signalled as an error, even in shift and count operations. If positions 8-9 are 00, this instruction is the same as NOP except for shift and count.

Autocoder Examples (Figures 35-38): The instruction assembled from Figure 35 is:

S01	*23*	*45*	*6*	*7*	*89*
+50	00	00	2	0	03

The instruction assembled from Figure 36 is:

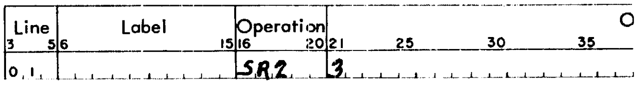
S01	*23*	*45*	*6*	*7*	*89*
+50	00	00	3	1	02

The instruction assembled from Figure 37 is:

S01	*23*	*45*	*6*	*7*	*89*
+50	00	00	1	2	07

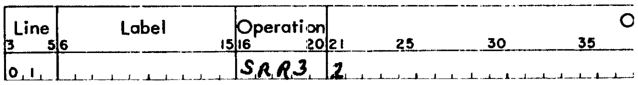
The instruction assembled from Figure 38 is:

S01	*23*	*45*	*6*	*7*	*89*
+50	00	91	2	3	00



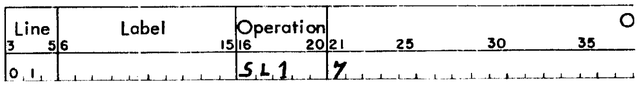
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		SR2	3

Figure 35



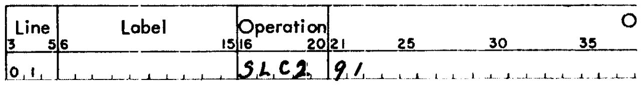
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		SRR3	2

Figure 36



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		SL1	7

Figure 37



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		SLC2	91

Figure 38

Coupled Shift Control
—50

Machine Description: This augmented code is used with accumulators 1 and 2 coupled, with accumulator 1 as the high-order accumulator. The operation is either normal-shift or split-shift. Normal-shift operates the same as the shift operation for a single accumulator (+50), except that it operates a single 20-position accumulator, composed of accumulators 1 and 2. Split-shift operations cause the 20-digit number to be "broken apart" between any two positions, with the positions, direction of shift, and number of positions shifted all specified in the instruction (Figure 39).

For shift right coupled, or shift right and round coupled, the sign of accumulator 2 is replaced by the sign of accumulator 1. For shift left or a shift left and count, the sign of accumulator 1 is replaced by the sign of accumulator 2. This is true even for shifts of zero places.

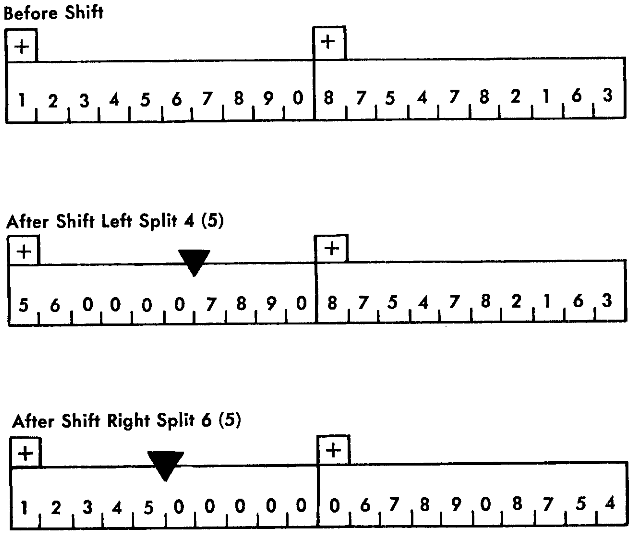


Figure 39. Examples of Split-Shifting

Instruction Format

S01 —50

23 Indexing word. Even though positions 6-9 do not represent an address, they are modified by positions 2-5 of an IW, if one is specified here.

45 Index word, for shift and count. The number of places shifted (00-20) replaces positions 4-5 of the index word specified here (positions 2-3 of the index word are set to zero, and its sign is set to plus). These positions are not used in the other shift operations.

6 0-9, designating the position in accumulator 1 or 2 from which the split-shift operation will take place. For shift-left split, it is the units position of the digits shifted; for shift-right split, it is the high-order position. This position is not used with normal-shift operations.

7 0-7, determining the type of shift:

0—Shift right coupled	SR
1—Shift right and round coupled	SRR
2—Shift left coupled	SL
3—Shift left and count coupled	SLC

4—Shift right from point accumulator 1 SRS
5—Shift left from point accumulator 1 SLS
6—Shift right from point accumulator 2 SRS
7—Shift left from point accumulator 2 SLS

89 00-20, designating the number of positions to be shifted. Not used in shift and count but must be valid.

Examples: To shift right five places in accumulators 1 and 2 coupled:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	xx	x	0	05

Contents of the accumulators before the operation:

Acc 1	Acc 2
+12345 00000	−67890 00000.

Contents of the accumulators after the operation:

Acc 1	Acc 2
+00000 12345	+00000 67890.

To shift the contents of accumulators 1 and 2 coupled, three positions to the right, and round off the value:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	xx	x	1	03

Contents of the accumulators before the operation:

Acc 1	Acc 2
+34780 00022	+00001 78534.

Contents of the accumulators after the operation:

Acc 1	Acc 2
+00034 78000	+02200 00179.

To shift left 16 places in accumulators 1 and 2 coupled:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	xx	x	2	16

Contents of the accumulators before the operation:

Acc 1	Acc 2
+75346 92188	−74310 25505.

Contents of the accumulators after the operation:

Acc 1	Acc 2
−55050 00000	−00000 00000.

To shift the contents of accumulators 1 and 2 coupled, to the left, until a significant digit is in the high-order position; and to store the number of positions shifted, in positions 2-5 of IW 25:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	25	x	3	yy

Positions 8-9 are not used but cannot be greater than 20. Contents of the accumulators before the operation:

Acc 1	Acc 2
−00007 00000	−12345 00222.

Contents of the accumulators after the operation:

Acc 1	Acc 2
−70000 01234	−50022 20000.

Contents of IW 25 before the operation: +66 0155 6420. Contents of IW 25 after the operation: +66 0004 6420.

To split-shift the contents of accumulators 1 and 2 coupled, ten positions to the right, starting with position 5 of accumulator 1:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	xx	5	4	10

Contents of the accumulators before the operation:

Acc 1	Acc 2
+11223 34455	−66778 89900.

Contents of the accumulators after the operation:

Acc 1	Acc 2
+11223 00000	+00000 34455.

To split-shift the contents of accumulators 1 and 2 coupled, 3 positions to the left, starting with position 2 of accumulator 1:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	xx	2	5	03

Contents of the accumulators before the operation:

Acc 1	Acc 2
+53560 00000	−00000 04475.

Contents of the accumulators after the operation:

Acc 1	Acc 2
−00060 00000	−00000 04475.

Note that the sign of accumulator 1 is made the same as that of accumulator 2.

To split-shift the contents of accumulators 1 and 2 coupled, six positions to the right, starting with position 0 of accumulator 2:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	xx	0	6	06

Contents of the accumulators before the operation:

Acc 1	Acc 2
+12345 67890	+12345 67890.

Contents of the accumulators after the operation:

Acc 1	Acc 2
+12345 67890	+00000 01234.

To split-shift the contents of accumulators 1 and 2 coupled, 19 positions to the left, starting with the units position of accumulator 2:

S01	*23*	*45*	*6*	*7*	*89*
−50	23	xx	9	7	19

Contents of the accumulators before the operation:

Acc 1	Acc 2
−99887 76655	+00044 33221.

Contents of the accumulators after the operation:

Acc 1	Acc 2
+10000 00000	+00000 00000.

7074 Timing: In microseconds:

SR	6 + (N−3)*
SRR	8 + N
SL	6 + (N−3)*
SLC	14 + (N−3)*
SRS	11 + N point remains in acc
	8 + N point moves out of acc

SLS 11 + N point remains in acc
8 + N point moves out of acc
N = number of positions to be shifted
*Note that if N<3, (N−3) = 0

7070 Timing: See Figure 40.

OPERATION	NOT INDEXED		INDEXED	
	Number of Positions Shifted	Time (μs)	Number of Positions Shifted	Time (μs)
SR or SL	0, 1, 2, 10, 11, 12	36	0, 10	60
	3, 4, 5, 13, 14, 15	48	1, 2, 3, 11, 12, 13	72
	6, 7, 8, 16, 17, 18	60	4, 5, 6, 14, 15, 16	84
	9, 19, 20	72	7, 8, 9, 17, 18, 19	96
			20	108
SRR	0	36	0	60
	10	120	1, 10, 11	156
	1, 2, 3, 11, 12, 13	132	2, 3, 4, 12, 13, 14	168
	4, 5, 6, 14, 15, 16	144	5, 6, 7, 15, 16, 17	180
	7, 8, 9, 17, 18, 19	156	8, 9, 18, 19, 20	192
	20	168		
SLC	0	84	0	120
	1, 2, 10	96	1, 2, 3, 10	132
	3, 4, 5, 11, 12	108	4, 5, 6, 11, 12, 13	144
	6, 7, 8, 13, 14, 15	120	7, 8, 9, 14, 15, 16	156
	9, 16, 17, 18	132	17, 18, 19	168
	19, 20	144	20	180

SRS from position in:		SLS from position in:		NOT INDEXED		INDEXED	
A1	A2	A1	A2	Number of Positions Shifted	Time (μs)	Number of Positions Shifted	Time (μs)
0	0	9	9	0 -- 20	132	0 — 20	156
1	1	8	8	0 -- 20	120	0 — 20	156
2	2	7	7	0 -- 16	108	0 — 16	144
2			7	17 -- 20	120	17 — 20	156
3	3	6	6	0 -- 16	108	0 — 16	132
3			6	17 -- 20	120	17 — 20	144
4	4	5	5	0 -- 13	96	0 — 13	132
4			5	14 -- 20	108	14 — 20	144
5	5	4	4	0 -- 10	84	0 — 11	120
5			4	11 -- 13	96	12 — 14	132
5			4	14 -- 20	108	15 — 20	144
6	6	3	3	0 -- 10	84	0 — 8	108
6			3	11 -- 13	96	9 — 11	120
6			3	14 -- 20	108	12 — 20	132
7	7	2	2	0 -- 7	72	0 — 8	108
7			2	8 -- 10	84	9 — 11	120
7			2	11 -- 20	96	12 — 20	132
8	8	1	1	0 -- 4	60	0 — 5	69
8			1	5 -- 7	72	6 — 8	108
8			1	8 -- 10	84	9 — 11	120
8			1	11 -- 20	96	12 — 20	132
9	9	0	0	0 -- 4	60	0 — 2	84
9			0	5 -- 7	72	3 — 5	96
9			0	8 -- 10	84	6 — 8	108
9			0	11 —20	96	9 — 20	120

Figure 40. IBM 7070 Timing – Coupled Shift Control

Comments: Because this instruction always uses accumulators 1 and 2 coupled, no accumulator number is required in the Autocoder mnemonics. A normal-shift left of ten positions sets accumulator 2 to zeros and puts its contents and sign into accumulator 1. A normal-shift of ten positions to the right does the reverse. A normal-shift of ten or more positions always sets one accumulator to zeros and gives the other accumulator the sign of the reset accumulator. A normal-shift of 20 positions sets both accumulators to zeros.

In either normal-shift or split-shift instructions, the signs of both accumulators are always the same after the operation is completed. They both have the sign of accumulator 1 after a shift-right, or the sign of accumulator 2 after a shift-left operation. This is true even for shifts of zero places (00 in positions 8-9, or no shift required in shift left and count). The sign of accumulator 2 can be made the same as that of accumulator 1, without changing the contents of either accumulator; for example, by the instruction:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	00	9	4	00

The digit in position 7 could be 4 or 6. Position 6 can contain any digit (whenever positions 8-9 are 00). Code 5, shift left from point in accumulator 1, can never change the contents of accumulator 2. Similarly, code 7 can never change the contents of accumulator 1. Shift left or right from a point can specify a shift of more positions than there are available (e.g., shift right 20 positions from the units position of accumulator 2), as long as the number of positions does not exceed 20. If more than 20, the 7070 stored program stops with the program check light on.

Autocoder Examples (Figures 41-47): The instruction assembled from Figure 41 is:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	00	0	0	12

The instruction assembled from Figure 42 is:

−50 00 00 0 1 05

The instruction assembled from Figure 43 is:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	00	0	2	17

The instruction assembled from Figure 44 is:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	47	0	3	00

Note that the number in the operand column designates the index word in which the shift-and-count position total is to be stored.

The instruction assembled from Figure 45 is:

S01	*23*	*45*	*6*	*7*	*89*
−50	00	00	5	4	02

The instruction assembled from Figure 46 is:

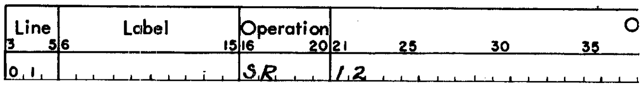
S01	*23*	*45*	*6*	*7*	*89*
−50	00	00	5	6	02

For Autocoder programming, the digits in the coupled accumulators are numbered from 00 through 19, from the high-order position of accumulator 1 to the low-order position of accumulator 2. The 15 here thus means position 5 of accumulator 2. The only difference between this assembled instruction and the previous one is in position 7.

The instruction assembled from Figure 47 is:

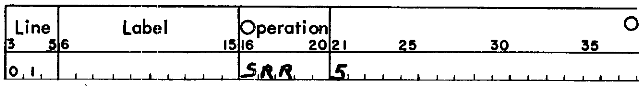
```
S01  23  45  6  7  89
-50  00  00  4  7  15
```

This instruction sets all of accumulator 1 and the five high-order positions of accumulator 2 to zero.



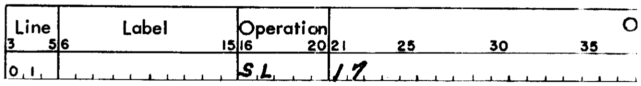
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		SR	12

Figure 41



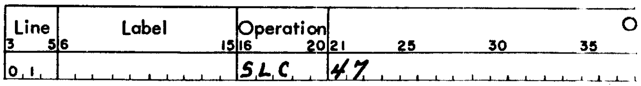
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		SRR	5

Figure 42



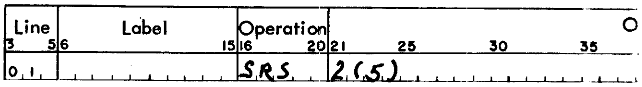
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		SL	17

Figure 43



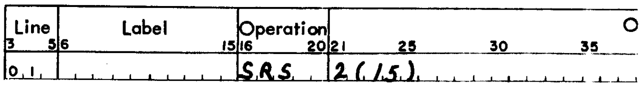
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		SLC	47

Figure 44



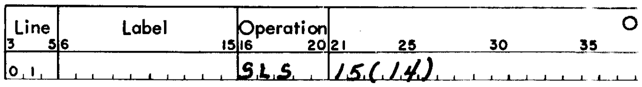
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		SRS	2(5)

Figure 45



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		SRS	2(15)

Figure 46



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		SLS	15(14)

Figure 47

Logic Codes

The computer stored program can make logical decisions based on a considerable number of different situations. The logic operations are classified into two types: *branch* and *compare*. Miscellaneous operations are also included in this section, because they are usually used in connection with logical decisions made by the program. The branch operations are logical decisions; the program either branches or does not, depending on the result of a test. A compare operation is also a test, but the result of the test merely turns on an indicator for later interrogation by a branch instruction. The miscellaneous codes perform no operation but are useful in connection with the logic operation codes, automatic checking features of the system, and program testing.

The branch operation codes in this section are listed in their categories in Figure 48; the compare codes and miscellaneous are listed in Figure 49.

All of the Autocoder symbols that start with B are branch codes. (But not all the branch codes start with B; HB is an example.) All the compare symbols start with C, and no other codes start with C.

Branch if Zero in Accumulator # +10, +20, +30 — BZ1, BZ2, BZ3

Machine Description: The contents of the specified accumulator are tested for zero. If it is zero, the program branches to the location in positions 6-9 (indexable). If it is not zero, the program continues sequentially, taking its next instruction from the location in the instruction counter. The sign of the accumulator is ignored.

Instruction Format

S	Always +.
0	Designates the accumulator: 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3.
1	Always 0.
23	Indexing word.
45	Not used.

CATEGORIES	OP CODES	NAMES	MNEMONICS
Branch			
Accumulators	+10, +20, +30	Branch if zero in accumulator #	BZ1, BZ2, BZ3
	−10, −20, −30	Branch if minus in accumulator #	BM1, BM2, BM3*
Alteration Switch	+51	Branch if alteration switch is on	BAS
Channel busy	+51	Branch if channel is busy	BCB
Electronic switches	+61, +62, +63	Electronic switch control:	
		Branch if electronic switch is on	BES
		Electronic switch set on	ESN
		Electronic switch set off	ESF
		Branch if electronic switch is on, and set on if off	BSN
		Branch if electronic switch is on, and set off if on	BSF
Indicator branch	+40	Branch if low	BL*
	−40	Branch if high	BH*
	−41	Branch if equal	BE*
	+11, +21, +31	Branch if overflow in accumulator #	BV1, BV2, BV3*
	+41	Field overflow control:	
		Branch if field overflow	BFV*
		Sense mode for field overflow	SMFV*
		Halt mode for field overflow	HMFV*

*On the 7070, these require only 24 microseconds for indexing, rather than 36.

This category also includes the branch on sign change instruction, BSC, included in the Compare category under −03, Sign Control. The *floating-decimal overflow* and *underflow* indicator-branch instructions are in the Floating Decimal section.

Other branch instructions are included in the Index Word, Disk Storage, and Priority Codes sections.

Figure 48. Branch Operation Codes

CATEGORIES	OP CODES	NAMES	MNEMONICS
Compare			
Accumulators	+15, +25, +35 −15	Compare accumulator # to storage Compare absolute in accumulator 1 to absolute in storage	C1, C2, C3 CA
Digit	+03	Compare storage to digit	CD
Sign	−03	Sign Control: Compare sign to alpha Compare sign to minus Compare sign to plus Make sign alpha Make sign minus Make sign plus Sense mode for sign change Halt mode for sign change Branch on sign change	 CSA CSM CSP MSA MSM MSP SMSC* HMSC* BSC*
Miscellaneous			
Branch	+01 +02 +00	Branch (unconditional) Branch and load location in index word Halt and branch	B* BLX HB*
Other	−00 −01	Halt and proceed No operation	HP* NOP*

*On the 7070, these require only 24 microseconds for indexing, rather than 36.

Figure 49. Compare Operation Codes

6789 Address of the next instruction, if the accumulator is zero. Not used if the accumulator is non-zero.

Example: To branch to location 1580 if accumulator 2 is zero:

S01	*23*	*45*	*6789*
+20	00	xx	1580

Contents of accumulator 2: −00000 00100. The program does not branch.

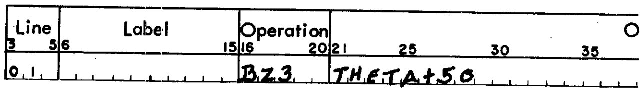
7074 Timing: 6 microseconds.

7070 Timing: 36 microseconds.

Comments: Note that the sign of the accumulator is ignored. The accumulator is considered zero only if all ten digits are zeros; there is no field definition. The accumulator is not changed by this operation.

Autocoder Example (Figure 50): Assume that THETA has been previously defined as word 4100 (or any part of word 4100). The assembled instruction is:

S01	*23*	*45*	*6789*
+30	00	xx	4150



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		BZ3	THETA+50

Figure 50

Branch if Minus in Accumulator #

−10, −20, −30 BM1, BM2, BM3

Machine Description: The sign of the specified accumulator is tested for minus. If it is minus, the program branches to the location in positions 6-9 (indexable). If it is not minus, the program continues sequentially, taking its next instruction from the location in the instruction counter. The contents of the accumulator are ignored.

Instruction Format

S Always −.

0 Designates the accumulator: 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3.

1 Always 0.

23 Indexing word.

45 Not used.

6789 Address of the next instruction, if the accumulator sign is minus. Not used if the sign is plus or alpha.

Example: To branch to location 1670 if accumulator 3 is minus:

S01	*23*	*45*	*6789*
−30	00	xx	1670

Contents of accumulator 3: @ 618264 0000. The program does not branch.

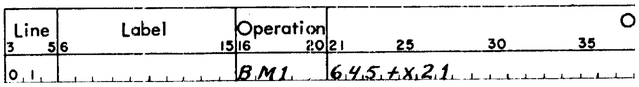
7074 Timing: 4 microseconds.

7070 Timing: 36 microseconds.

Comments: Note that the test is for minus or nonminus; if it is either plus or alpha, there is no branch. The contents of the accumulator are ignored, and neither sign nor contents are changed by the operation.

Autocoder Example (Figure 51): The assembled instruction is:

S01	*23*	*45*	*6789*
−10	21	00	0645



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		BM1	645+X21

Figure 51

Branch on Alteration Switch or Channel Busy +51

Machine Description: This is an augmented code, which performs two functions:

1. Tests any of the four alteration switches on the console and branches to the location in positions 6-9 (indexable) if that switch is on.
2. Tests any one of the tape-disk channels, and branches to the location in positions 6-9 (indexable) if that channel is busy. Several causes can make a channel busy. The most common is the transmission of data between core storage and tape or disk storage, as the result of read or write instruction.

Instruction Format

S01 +51

23 Indexing words.

4 For alteration-switch test, 1-4, specifying the alteration switch.
For 7604 channel-busy test, 1-4, specifying the 7604 channel.
For 7907 channel-busy test, 1-4, specifying the 7907 channel.

5 0-2, determining the operation:
0 Branch if alteration switch is on BAS
1 Branch if 7604 channel is busy BCB
2 Branch if 7907 channel is busy BDCB

6789 Branch address if the specified alteration switch is on, or if the specified channel is busy.

Examples: To branch to location 2244, if alteration switch 3 is on:

S01	*23*	*4*	*5*	*6789*
+51	00	3	0	2244

To branch to location 0888, if channel 2 is busy:

S01	*23*	*4*	*5*	*6789*
+51	00	2	1	0888

7074 Timing: 6 microseconds.

7070 Timing: 36 microseconds.

Comments: The alteration switches afford operator control over the stored program, to the extent that they are used in the program. They are occasionally helpful in program testing. The stored program cannot turn an alteration switch on or off; this is done by console operation only.

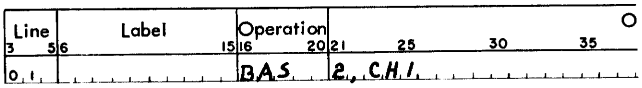
The channel-busy test can be useful in saving time. A tape-read instruction may hold up the stored program; for example, if there is another tape unit on that channel in operation at the time the instruction is given. In many cases, the efficiency of the stored program can be increased if a channel-busy test is given before every tape read or write instruction, particularly if the tape records are long. If the channel is busy, the program can branch and do something else, instead of being interlocked until the channel is free.

A busy channel stays busy until after the final status word has been formed in core storage and, when necessary, a stacking latch has been set. Then the channel becomes free.

This instruction can be used at the completion of a disk-storage seek operation to determine which channel is available for the subsequent disk read or write operation.

Autocoder Examples (Figures 52 and 53): CHI has been defined as word 2360. The instruction assembled from Figure 52 is:

S01	*23*	*4*	*5*	*6789*
+51	00	2	0	2360

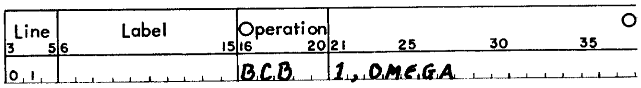


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		BAS	2, CHI

Figure 52

OMEGA is word 3343, previously defined. The instruction assembled from Figure 53 is:

S01	*23*	*4*	*5*	*6789*
+51	00	1	1	3343



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		BCB	1, OMEGA

Figure 53

Electronic Switch Control +61, +62, +63

Machine Description: These augmented codes cover all of the operations involving the 30 electronic switches: testing whether a switch is on or off, turning one on or off, or combining the test with turning a switch on or off. Code +61 provides for testing and/or setting the switches in word 0101; +62 for the switches in word 0102; and +63 for the switches in word 0103.

Instruction Format

S Always +

0 Always 6.

1 1-3, designating the group of ten switches. This position and position 5 determine the specific electronic switch of the 30 available.

23 Indexing word for the branch address if the instruction is a test of an electronic switch. For non-test operations, these positions have no function and can contain any digits.

4 0-4, determining the specific operation:

0—Test	BES
1—Turn on	ESN
2—Turn off	ESF
3—Test and turn on	BSN
4—Test and turn off	BSF

5 0-9, designating the switch number of the ten specified in position 1. These digits actually refer to positions 0-9 of word 0101, 0102, or 0103. The switches are numbered 1-30. Switch 1 has its designation in position 0 of word 0101; switch 2 in position 1, etc. The designation for switch 30 is in position 9 of word 0103. Thus, electronic switch 7, for example, is on if there is a digit 1-9 in position 6 of word 0101, and off if there is a zero in that position.

6789 Branch address (indexable) for the branch instructions (0, 3, or 4 in position 4). For the other two operations, these positions are not used and can contain any digits. If the specified electronic switch is off when a branch instruction is given, these positions are not used.

0—Test: The electronic switch specified in positions 1 and 5 is tested. If it is on, the next instruction is taken from the address in positions 6-9, after indexing if indexing is used. If it is off, the next instruction is taken from the location in the instruction counter.

The test is actually made of the digit, in word 0101, 0102, or 0103, that corresponds to the specified electronic switch. If there is a digit 1-9 in that position, the switch is considered on and the branch is executed. If that position contains a zero, the switch is considered off and no branch is executed.

1—Turn On: The digit position in word 0101, 0102, or 0103, specified by positions 1 and 5 of the instruction, is set to 1. The program does not branch; the next instruction comes from the location in the instruction counter.

2—Turn Off: This is the same as turn on, except that a 0 is inserted into the specified position of word 0101, 0102, or 0103, thus turning the switch off.

3—Test and Turn On: If the specified switch is on, the program branches to the location in positions 6-9 (indexable). Regardless of whether the branch took place, the specified switch is turned on after the test. A 1 is inserted into the specified position of word 0101, 0102, or 0103.

4—Test and Turn Off: Same as test and turn on, except that a 0 is inserted into the specified position of word 0101, 0102, or 0103, after the test is made.

Examples: To branch to word 1352 if electronic switch 14 is on:

S01	*23*	*4*	*5*	*6789*
+62	00	0	3	1352

Contents of word 0102: +01010 11001, unchanged by the operation. The program branches, because of the 1 in position 3.

To turn on electronic switch 25:

S01	*23*	*4*	*5*	*6789*
+63	xx	1	4	xxxx

Contents of word 0103 before the operation: −23408 50332; contents of word 0103 after the operation: −23401 50332.

To turn off electronic switch 29:

S01	*23*	*4*	*5*	*6789*
+63	xx	2	8	xxxx

Contents of word 0103: +11101 00110, the same before and after the operation, because position 9 was already at zero.

7074 Timing: 14 microseconds.

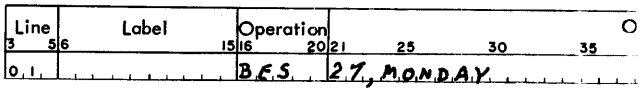
7070 Timing: 48 microseconds.

Comments: The test operations all test the specified electronic switch for an ON condition; the program branches if the switch is on, and does not branch if it is off. The operations to test and turn on, and test and turn off, both make the test for ON, and then turn the switch on or off, as specified.

Words 0101, 0102, and 0103 can be changed by other instructions. They can be treated just like the other core-storage words.

Autocoder Examples (Figures 54-58): MONDAY has been previously defined as word 0427. The instruction assembled from Figure 54 is:

S01	*23*	*4*	*5*	*6789*
+63	00	0	6	0427

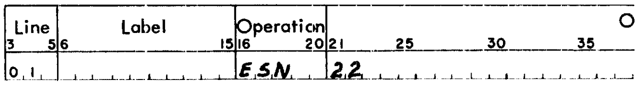


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0,1		BES	27,MONDAY

Figure 54

The instruction assembled from Figure 55 is:

S01	*23*	*4*	*5*	*6789*
+63	00	1	1	0000

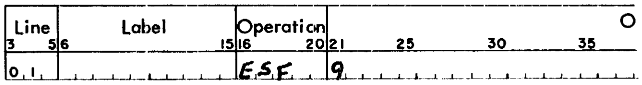


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		ESN	22

Figure 55

The instruction assembled from Figure 56 is:

S01	*23*	*4*	*5*	*6789*
+61	00	2	8	0000

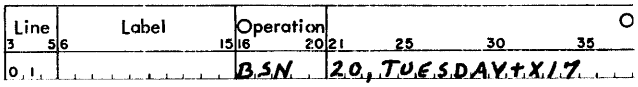


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		ESF	9

Figure 56

Tuesday has been previously defined as word 2704. The instruction assembled from Figure 57 is:

S01	*23*	*4*	*5*	*6789*
+62	17	3	9	2704

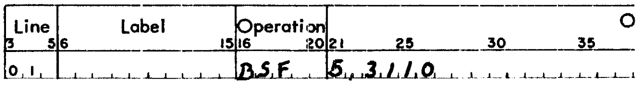


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		BSN	20,TUESDAY+X17

Figure 57

The actual location 3110 is used. The instruction assembled from Figure 58 is:

S01	*23*	*4*	*5*	*6789*
+61	00	4	4	3110



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		BSF	5,3110

Figure 58

Branch if Low

+40 **BL**

Machine Description: This instruction tests the *low* indicator. If it is on, the program branches to the address in positions 6-9 (indexable). If it is not on, the next instruction is taken from the location in the instruction counter. The indicator setting is not changed.

Instruction Format

S01	+40.
23	Indexing word for the branch address. Not used if there is no branch.
45	Not used.
6789	Branch address, used if the low indicator is on (indexable). Otherwise, not used.

Example: To branch to location 2300 if the low indicator is on:

S01	*23*	*45*	*6789*
+40	00	xx	2300

7074 Timing: 4 microseconds.

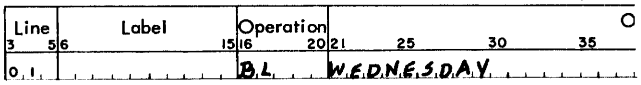
7070 Timing: 36 microseconds.

Comments: This instruction does not affect the setting of the low indicator or the other two indicators; the stored program normally changes these settings only with a compare instruction. (They can be changed by adjusting the priority indicator storage word, 0100, in a priority routine. The adjusted settings are returned to the indicators by a priority-release instruction. See section "Priority Processing" for a complete description of the priority indicator storage word.)

The high, equal, and low indicators are all turned off by pressing computer reset on the console operating keyboard. Pressing reset on the operating keyboard turns them off, only if none, two, or three of them had been on, and the machine has stopped because a compare instruction has discovered this.

Autocoder Example (Figure 59): wednesday has been previously defined as word 1100. The assembled instruction is:

S01	*23*	*45*	*6789*
+40	00	00	1100



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		BL	WEDNESDAY

Figure 59

Branch if High

−40 **BH**

Machine Description: This code is the same as branch if low, except that the high indicator is tested and a branch is executed if it is on.

Instruction Format: Same as branch if low, except for sign.

Example: To branch to location 3450 if the high indicator is on:

S01	*23*	*45*	*6789*
−40	00	xx	3450

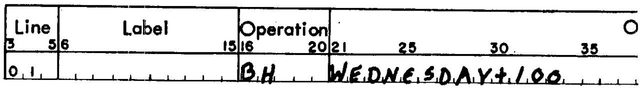
7074 Timing: 4 microseconds.

7070 Timing: 36 microseconds.

Comments: The comments under branch if low also apply to branch if high.

Autocoder Example (Figure 60): wednesday has been previously defined as word 1100. The assembled instruction is:

S01	*23*	*45*	*6789*
−40	00	00	1200



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35	O
01		BH	WEDNESDAY+100	

Figure 60

Branch if Equal
−41 **BE**

Machine Description: This code is the same as branch if low, except that the equal indicator is tested, and a branch is executed if it is on.

Instruction Format: Same as branch if low, except for positions S01, which are −41.

Example: To branch to location 4101 indexed by positions 2-5 of IW 56, if the equal indicator is on:

S01 23 45 6789
−41 56 xx 4101

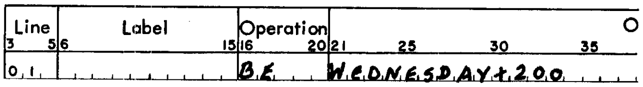
7074 Timing: 4 microseconds.

7070 Timing: 36 microseconds.

Comments: The comments under branch if low also apply to branch if equal. Note that these branch codes do not have to follow the compare instructions immediately; they can come at any time.

Autocoder Example (Figure 61): WEDNESDAY has been previously defined as word 1100. The assembled instruction is:

S01 23 45 6789
−41 00 00 1300



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35	O
01		BE	WEDNESDAY+200	

Figure 61

Branch if Overflow in Accumulator #
+11, +21, +31 **BV1, BV2, BV3**

Machine Description: These codes are similar to the branch if low, equal, and high codes, in that an indicator is tested, and the program branches to the address in positions 6-9 (indexable), if it is on. The indicator may have been set previously in the program; it does not have to be tested immediately. The overflow indicator for an accumulator is turned on when a true-add operation obtains an 11-digit result. The high-order digit of this result is lost (it is a carry 1 in all cases), and the overflow indicator for the accumulator is turned ON.

Unlike the branch if low, equal, and high codes, a branch-overflow instruction turns off the indicator after testing it.

Instruction Format

S Always +.
0 1-3, designating the accumulator; 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3. (Each accumulator has its own overflow indicator.)
1 Always 1.
23 Indexing word for the branch address. Not used if there is no branch.
45 Not used.
6789 Branch address if the specified overflow indicator is on (indexable).

Examples: To branch to location 4501 if the overflow indicator for accumulator 3 is on:

S01 23 45 6789
+31 00 xx 4501

7074 Timing: 4 microseconds.

7070 Timing: 36 microseconds.

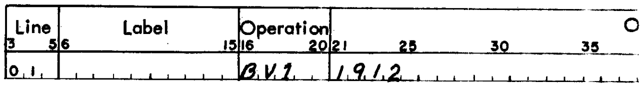
Comments: The only situation that can cause an accumulator to overflow and set the overflow indicator is a true-add to accumulator operation in which at least one of the factors is a ten-digit number.

Overflow occurs if the sum of two numbers is 11 digits in length. When this happens, the ten low-order digits of the correct total are in the accumulator, and the overflow indicator for that accumulator is turned on. The high-order digit of the 11-digit result is a 1, in all cases. If necessary, a program subroutine could be used to keep track of the overflows.

Each accumulator has an overflow light on the console. The light is on whenever the overflow indicator for that accumulator is on. The branch-if-overflow instruction turns off the console light, as well as the indicator. The console also contains an overflow key for each accumulator. If the key is not pressed, an overflow in that accumulator sets the indicator and stops the machine. If the key is pressed, an overflow in that accumulator merely sets the indicator; the program continues, with the indicator on, until a BV instruction tests it and turns it off.

Autocoder Example (Figure 62): The actual location 1912 is used. The assembled instruction is:

S01 23 45 6789
+11 00 00 1912



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35	O
01		BV1	1912	

Figure 62

Field Overflow Control
+41

Machine Description: This augmented code determines the setting of the field-overflow stop/sense switch. It also tests the field-overflow indicator, causing the program to branch to the address in positions 6-9 (indexable) if the indicator is on. This branch code then turns the indicator off if it is on, just as the branch-if-overflow codes (BV#) do.

Instruction Format

S01 +41.

23 Indexing word for the branch address. Not used if the branch instruction is not used, or if it is used but there is no branching.

4 Not used.

5 0-2, designating the operation:

0–Branch if field overflow	BFV
1–Sense mode for field overflow	SMFV
2–Halt mode for field overflow	HMFV

6789 Branch address for the BFV instruction. For the other instructions, these positions are not used and can contain any digits.

0–Branch if Field Overflow: The field-overflow indicator is tested. If it is on, the address in positions 6-9 (indexable) is transmitted to the instruction counter, and the next instruction is taken from that location. The field-overflow indicator is turned off if it was on.

1–Sense Mode for Field Overflow: The field-overflow sense/stop switch is set to sense. If it is already at sense, this instruction has no effect. When the switch is at sense, a field-overflow condition arising in the program turns on the field-overflow indicator but does not stop the machine.

2–Halt Mode for Field Overflow: The field-overflow sense/stop switch is set to stop. If it is already at stop, this instruction has no effect. When the switch is at stop, a field-overflow condition arising in the program turns on the field-overflow indicator, and stops the machine. If the field-overflow indicator is on when this instruction is given, the machine stops.

Examples: To branch to location 2365 if the field overflow indicator is on:

S01	*23*	*4*	*5*	*6789*
+41	00	x	0	2365

To set the field-overflow stop/sense switch to stop, regardless of its previous setting:

S01	*23*	*4*	*5*	*6789*
+41	xx	x	2	xxxx

7074 Timing: 4 microseconds.

7070 Timing: 36 microseconds.

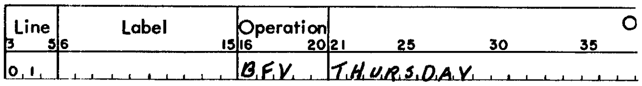
Comments: Two devices are involved with this operation code, and they have no direct relationship with each other. The field-overflow indicator is immediately turned on whenever a field-overflow situation arises in a store or add-to-storage operation. The branch code, BFV, turns it off after making the test. There is no other way to turn the field-overflow indicator off, within the stored program.

The field-overflow stop-sense switch is constantly at either sense or stop, determined by whether a HMFV or SMFV instruction has been given more recently. If it is on stop, a field-overflow situation arising in the program stops the machine, after turning on the field-overflow indicator. To turn off the indicator and start the machine, press RESET and START on the console. If the switch is on sense, the program does not stop. The switch could thus be considered as having stop and no-stop settings.

Pressing COMPUTER RESET, on the console, sets the field-overflow stop/sense switch to stop.

Autocoder Examples (Figure 63 and 64): THURSDAY has been previously defined as word 1775. The instruction assembled from Figure 63 is:

S01	*23*	*4*	*5*	*6789*
+41	00	0	0	1755



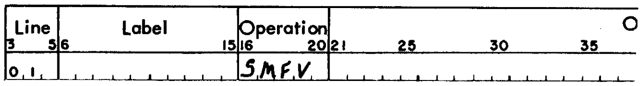
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		BFV	THURSDAY

Figure 63

The instruction assembled from Figure 64 is:

S01	*23*	*4*	*5*	*6789*
+41	00	0	1	0000

Note that no information is needed in the operand section.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		SMFV	

Figure 64

Compare Accumulator # to Storage +15, +25, +35 C1, C2, C3

Machine Description: The entire contents of the specified accumulator, including sign, are compared with the field-defined digits and sign of the storage word addressed by positions 6-9 (indexable). One of the three compare indicators, high, equal, or low, is turned on, and the other two are turned off, depending on the relation of the accumulator value to the storage-word value. If the accumulator is higher, the high indicator is turned on; if the accumulator value is lower, the low indicator is turned on. If they are equal, the equal indicator is turned on.

The following is a chart of the relative values of signs and digits in compare operations:

Highest +99999 99999
to
+00000 00000
−00000 00000
to
−99999 99999
@99999 99999
to
Lowest @00000 00000

Note that plus is higher than minus, which is in turn higher than alpha, regardless of the numeric values. Note also that the greater a minus number is in absolute value, the smaller it is in a compare operation. Thus, a minus 9 is smaller than a minus 8, which is in turn smaller than a minus 7, etc.

The contents of the storage word and the accumulator are unchanged by this operation.

Instruction Format

S Always plus.
0 1-3, designating the accumulator: 1 for accumulator 1, 2 for accumulator 2, 3 for accumulator 3.
1 Always 5.
23 Indexing word.
45 Field definition.
6789 Address (indexable) of the word in core storage, the sign and field-defined digits of which are to be compared with the specified accumulator.

Examples: To compare the entire contents of accumulator 1 with positions 0-4 of word 1806:

S01 23 45 6789
+15 00 04 1806

Contents of accumulator 1: +00001 23456. Contents of word 1806: +23456 78900. The high indicator is turned on; 123,456 is higher than 23,456.

To compare the entire contents of accumulator 3 with all of word 1919:

S01 23 45 6789
+35 00 09 1919

Contents of accumulator 3: −00000 11122. Contents of word 1919: +00000 11122. The low indicator is turned on, because the accumulator value is minus and the storage value is plus.

To compare the entire contents of accumulator 2 with position 9 of word 1750:

S01 23 45 6789
+25 00 99 1750

Contents of accumulator 2: −00000 00005. Contents of word 1750: −18555 02774. The low indicator is turned on, because the accumulator has the minus number that is greater in absolute value.

To compare the entire contents of accumulator 1 with all of word 1515:

S01 23 45 6789
+15 00 09 1515

Contents of accumulator 1: −61777 90091. Contents of word 1750: −18555 02774. The low indicator is turned on, because the accumulator has the minus number that is greater in absolute value.

To compare the entire contents of accumulator 1 with all of word 1515:

S01 23 45 6789
+15 00 09 1515

Contents of accumulator 1: @61777 90091. Contents of word 1515: @61777 90091. The equal indicator is turned on.

7074 Timing: 8 microseconds.

7070 Timing: The duration of a compare instruction is determined by the number of significant digits in the accumulator, or the number of digits specified by field definition, whichever is greater (Figure 65). These timings are the same as complement-add to accumulator.

Number of Digit Positions	1	2	3	4	5	6	7	8	9	10
Microseconds	36	48	48	48	60	60	60	72	72	72

Figure 65. IBM 7070 Timing — Compare Accumulator# to Storage

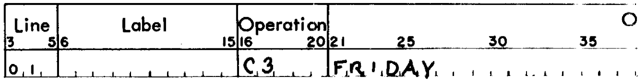
Comments: Note that the sign is more important than the value, in compare operations. If the signs are different, the values have no effect on the result of the comparison. Note, also, that the entire accumulator is always used; field definition is used with the storage word only.

Accumulators can be addressed by these codes. This means that two accumulators can be compared with each other, or an entire accumulator can be compared with a field-defined portion of itself, to determine whether the remaining positions are all zeros.

After a compare operation one of the three indicators, high, low, or equal, is on and the other two are off.

Autocoder Examples (Figures 66 and 67): Assume that FRIDAY has been previously defined as positions 6-9 of word 2120. The instruction assembled from Figure 66 is:

S01 23 45 6789
+35 00 69 2120

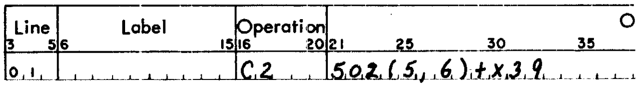


Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
01		C3	FRIDAY

Figure 66

The actual address 502 is used. The instruction assembled from Figure 67 is:

S01 *23* *45* *6789*
+25 39 56 0502



Line	Label	Operation	○
3 5	6 15	16 20	21 25 30 35
0 1		C2	S02(5,6)+X39

Figure 67

Compare Absolute in Accumulator 1 to Absolute in Storage
−15 **CA**

Machine Description: This code is similar to compare accumulator to storage, with two exceptions:

1. The signs of both factors are ignored. The comparison is made of the values only, both considered plus.
2. This code can be used with accumulator 1 only.

Instruction Format

S01 −15.
23 Indexing word.
45 Field definition.
6789 Address (indexable) of the word in storage whose field-defined digits are to be compared, in absolute value, with the entire contents of accumulator 1.

Examples: To compare, in absolute values, the entire contents of accumulator 1 with positions 0-4 of word 2814:

S01 *23* *45* *6789*
−15 00 04 2814

Contents of accumulator 1: +00000 01235. Contents of word 2814: −02235 65222. The low indicator is turned on, even though the storage word is minus.

To compare, in absolute value, the entire contents of accumulator 1 with all of word 2005:

S01 *23* *45* *6789*
−15 00 09 2005

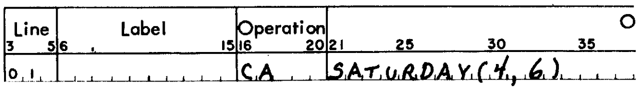
Contents of accumulator 1: +00000 00000. Contents of word 2005: −00000 00000. The equal indicator is turned on.

Timing: Same as COMPARE ACCUMULATOR # TO STORAGE.

Comments: As with all absolute-value instructions (except AAS#) this code can be used with accumulator 1 only, specified in the operation code. Accumulator addresses can be used in positions 6-9, however, for comparison with accumulator 1.

Autocoder Example (Figure 68): Assume that SATURDAY has been defined as the six high-order positions of word 1509. The assembled instruction is:

S01 *23* *45* *6789*
−15 00 46 1509



Line	Label	Operation	○
3 5	6 15	16 20	21 25 30 35
0 1		CA	SATURDAY(4,6)

Figure 68

Compare Storage to Digit
+03 **CD**

Machine Description: The contents of a single specified digit position of the storage word addressed by positions 6-9 (indexable), are compared with a single-digit value contained in the instruction itself. The comparison is of absolute values, and the high, equal, or low indicator is turned on, determined by the value of the digit in storage. If the storage digit is greater than the instruction digit, the high indicator is turned on; if it is less, the low indicator is turned on. If they are equal, the equal indicator is turned on.

Instruction Format

S01 +03.
23 Indexing word.
4 0-9, specifying the digit value that the storage-word digit is to be compared with.
5 0-9, designating the position in the storage word that contains the digit to be compared.
6789 Address (indexable) in storage of the word that contains the digit to be compared.

Example: To compare the units position in word 2932 with a zero:

S01 *23* *4* *5* *6789*
+03 00 0 9 2932

Contents of word 2932: −12345 67890. The equal indicator is turned on.

7074 Timing: 8 microseconds.

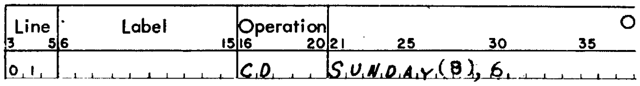
7070 Timing: 36 microseconds.

Comments: This compare instruction does not necessarily involve an accumulator. Accumulators can be addressed, however, in positions 6-9. If there is a 0 in position 4 of the instruction, only the equal or high indicator is turned on. Similarly, a 9 in position 4 can cause only the equal or low indicator to be turned on.

Autocoder Example (Figure 69): Assume SUNDAY to have been defined as word 1948. The assembled instruction is:

S01 *23* *4* *5* *6789*
+03 00 6 8 1948

This instruction compares the digit in position 8 of word 1948 with a 6.



Line	Label	Operation	○
3 5	6 15	16 20	21 25 30 35
0 1		CD	SUNDAY(8),6

Figure 69

Sign Control
−03

Machine Description: This augmented code is used for four different purposes:

1. Comparing the sign of a storage word with the sign value specified in the instruction, and turning on the high, equal, or low indicator as a result.
2. Making the sign of a word plus, minus, or alpha, as specified in the instruction, regardless of its previous sign.
3. Setting the sign-change sense-stop switch to sense or stop.
4. Testing the sign-change indicator, and branching if it is on. This operation automatically turns it off if it is on.

Instruction Format

S01 −03.

23 Indexing word.

4 3, 6, or 9 in the operations for testing and making signs (1 and 2 above):
3—Alpha
6—Minus
9—Plus
Not used with the other operations.

5 0-4, specifying the operation:
0—Compare sign*
1—Make sign**
2—Sense mode for sign change SMSC
3—Halt mode for sign change HMSC
4—Branch if sign change BSC

*The compare-sign instruction combines with each sign designation in position 4 to make up an instruction:

POSITION 4	5		
3	0	Compare Sign to Alpha	CSA
6	0	Compare Sign to Minus	CSM
9	0	Compare Sign to Plus	CSP

**Similarly, the make-sign instruction combines with the three sign designations:

POSITION 4	5		
3	1	Make Sign Alpha	MSA
6	1	Make Sign Minus	MSM
9	1	Make Sign Plus	MSP

6789 For comparing or setting the sign, the location of the word whose sign is to be set or compared with. For testing the sign-change indicator, these positions contain the branch address (indexable in both cases). Not used for the sense-stop switch operations.

Compare Sign: The sign of the storage word addressed by positions 6-9 (indexable) is compared with the sign designation in position 4 of the instruction. An alpha sign has a value of 3, a minus sign 6, and a plus sign 9. If the storage-word sign is greater than the sign in position 4, the high indicator is turned on; if it is less, the low indicator is turned on. If the signs are the same, the equal indicator is turned on. The storage word is not changed by this operation.

Make Sign: The sign of the storage word addressed by positions 6-9 (indexable) is made the same as the sign designated in position 4 of the instruction, regardless of what it had been previously.

Sense Mode for Sign Change: The sign-change sense/stop switch is set to sense. If it is already at sense, this instruction has no effect. When the switch is at sense, a sign-change condition arising in the program turns on the sign-change indicator but does not stop the machine.

Halt Mode for Sign Change: The sign-change sense/stop switch is set to stop. If it is already at stop, this instruction has no effect. When the switch is at stop, a sign-change condition arising in the program turns on the sign-change indicator and stops the machine. If the sign-change indicator is on when this instruction is given, the machine stops.

To restart the program, press the error reset and start keys on the console operating keyboard.

Branch if Sign Change: The sign-change indicator is tested. If it is on, the address in positions 6-9 (indexable) is transmitted to the instruction counter, and the next instruction is taken from that location. The sign-change indicator is turned off, if it is on.

Examples: To compare the sign of word 1690 with plus:

S01	*23*	*4*	*5*	*6789*
−03	00	9	0	1690

Contents of word 1690: −02461 53794. The low indicator is turned on.

To make the sign of word 1690 plus:

S01	*23*	*4*	*5*	*6789*
−03	00	9	1	1690

Contents of 1690 before the operation: −02461 53794.

Contents of 1690 after the operation: +02461 53794.

To set the sign-change sense/stop switch to sense:

S01	*23*	*4*	*5*	*6789*
−03	xx	x	2	xxxx

To branch to location 3000, indexed by positions 2-5 of IW 49, if the sign-change indicator is on:

S01	*23*	*4*	*5*	*6789*
−03	49	x	4	3000

7074 Timing: In microseconds:

CSA	8
CSM	8
CSP	8
MSA	12
MSM	12
MSP	12
SMSC	4
HMSC	4
BSC	4

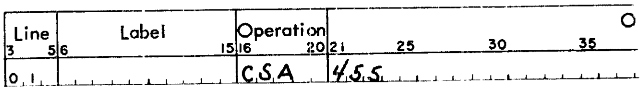
7070 Timing: 36 microseconds for all operations.

Comments: A sign can be fully tested by the COMPARE SIGN-TO-MINUS instruction. If the storage-word sign is plus, the high indicator is turned on; if minus, the equal indicator; and if alpha, the low indicator. The CSA code can never turn on the low indicator, and the CSP code can never turn on the high indicator.

The sign-change indicator and stop/sense switch are two different features. Their relation to each other is the same as described under field overflow for the corresponding features.

Autocoder Examples (Figures 70-73): The actual address 455 is used. The instruction assembled from Figure 70 is:

S01	*23*	*4*	*5*	*6789*
−03	00	3	0	0455

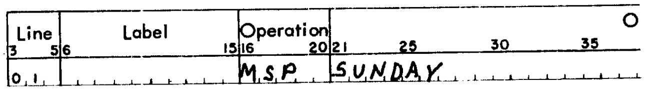


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		CSA	455

Figure 70

SUNDAY has been previously defined as word 1770. The instruction assembled from Figure 71 is:

S01	*23*	*4*	*5*	*6789*
−03	00	9	1	1770

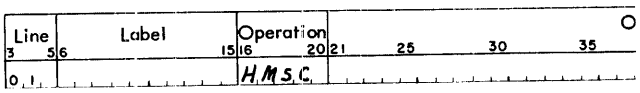


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		MSP	SUNDAY

Figure 71

No operand is required. The instruction assembled from Figure 72 is:

S01	*23*	*4*	*5*	*6789*
−03	00	0	3	0000

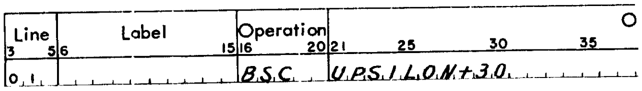


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		HMSC	

Figure 72

UPSILON is word 1690. The instruction assembled from Figure 73 is:

S01	*23*	*4*	*5*	*6789*
−03	00	0	4	1720



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		BSC	UPSILON+30

Figure 73

Branch +01 B

Machine Description: The four-digit address in positions 6-9, indexed if indexing is specified, is transmitted to the instruction counter. The next instruction in the program is taken from that address.

Instruction Format

S01	+01.
23	Indexing word.
45	Not used.
6789	Location of the next instruction.

Examples: To branch unconditionally to location 2000, indexed by positions 2-5 of IW 88:

S01	*23*	*45*	*6789*
+01	88	xx	2000

To branch unconditionally to the address in positions 2-5 of index word 83:

S01	*23*	*45*	*6789*
+01	83	xx	0000

IW 83 must be plus. If it is minus, the program branches to the complement of the address.

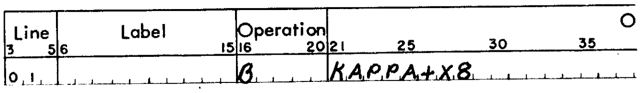
7074 Timing: 4 microseconds.

7070 Timing: 36 microseconds.

Comments: The unconditional branch is usually used for those operations (such as card read, punch, inquiry read and reply), in which an invalid character is detected during transmission of data. The next sequential instruction is taken in these cases, and is an unconditional branch to a subroutine for that particular situation. If no invalid characters are detected, the program automatically bypasses the next sequential instruction location, and proceeds from the location 1 or 2 beyond it. Note that the +01 branch code can be changed to the −01 no-operation code by changing only the sign of the instruction.

Autocoder Example (Figure 74): Assume that KAPPA has been previously defined as word 1080 (or any part of word 1080). The assembled instruction is:

S01	*23*	*45*	*6789*
+01	08	00	1080



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		B	KAPPA+X8

Figure 74

Branch and Load Location in Index Word +02 BLX

Machine Description: This is an unconditional branch and is the same as +01, with this addition: Before the address portion (indexable) of this instruction goes to the instruction counter, the present contents of the instruction counter are stored in positions 2-5 of the index word specified in position 4-5 of this instruction. The remaining positions of the IW are unchanged, and its sign is set to plus.

Instruction Format

S01 +02.
23 Indexing word.
45 Designates the index word that will contain, in positions 2-5, the present contents of the instruction counter.
6789 Location of the next instruction (indexable).

Example: To branch unconditionally to location 3111 and store the contents of the instruction counter into positions 2-5 of IW 50:

S01	*23*	*45*	*6789*
+02	00	50	3111

7074 Timing: 12 microseconds.

7070 Timing: 36 microseconds.

7074 Additional Storage Mode: The five digits from the instruction counter replace the contents of the indexing portion (1-5) of the specified index word.

Comments: The instruction-counter address stored in the index word is the location of the next instruction that would have been taken if it were not for the branch instruction. It is the location of the branch instruction, +1.

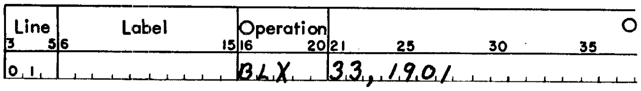
The stored program can be returned to this point by an unconditional branch (+01) instruction with an address of 0000 in positions 6-9, indexed by the index word in which the instruction counter was stored. Observe the following instruction, in location 1611:

S01	*23*	*45*	*6789*
+02	71	71	0000

Assume the contents of IW 71 to be: +00 1234 0000. Indexing takes place first; the 1234 in positions 2-5 of 0071 are added to the 0000 in positions 6-9 of the program register, and the result (1234) goes back to positions 6-9 of the program register. The 1612 in the instruction counter is then brought to positions 2-5 of 0071. The next instruction is taken from 1234. If the program later returns to 1611 without changing 0071, this instruction is just like a no-op; the BLX instruction in 1611 will cause a *branch* to 1612.

Autocoder Example (Figure 75): The assembled instruction is:

S01	*23*	*45*	*6789*
+02	00	33	1901



Line	Label	Operation	○
3 5	6 15	16 20	21 25 30 35
0 1		BLX	33,1901

Figure 75

Halt and Branch

+00 **HB**

Machine Description: This operation causes an unconditional machine stop. The console typewriter automatically types the contents of the instruction counter and program register. Pressing the start key on the operating keyboard restarts the program. The next instruction is taken from the location in positions 6-9 of the instruction (indexable).

Instruction Format

S01 +00.
23 Indexing word.
45 Not used.
6789 Branch address.

Example: To stop the machine and resume the stored program with the instruction in word 1991:

S01	*23*	*45*	*6789*
+00	00	xx	1991

The program cannot continue until the operator presses the start key on the console.

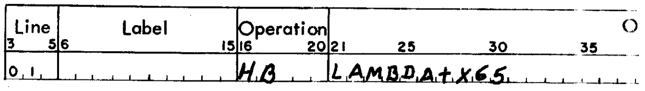
7074 Timing: 4 microseconds.

7070 Timing: 36 microseconds.

Comments: This code can be of considerable value in program testing. The program can be made to run to a predetermined point and stop there, by use of this code. When its contents are typed, the program register contains the halt-and-branch instruction. The instruction counter contains the address of the first instruction to be executed when the program is restarted.

Autocoder Example (Figure 76): Assume that LAMBDA has been previously defined as word 2462. The assembled instruction is:

S01	*23*	*45*	*6789*
+00	65	00	2462



Line	Label	Operation	○
3 5	6 15	16 20	21 25 30 35
0 1		HB	LAMBDA+X65

Figure 76

Halt and Proceed

−00 **HP**

Machine Description: This code is the same as HALT AND BRANCH, except that there is no branch when the program is restarted. The next instruction is taken from the location in the instruction counter.

Instruction Format

S01 −00
23 Indexing word. Even though there is no address with this instruction, positions 6-9 are modified by positions 2-5 of an indexing word if one is specified here.
456789 Not used.

Example: To halt and proceed:

S01 *23* *456789*

−00 00 xxxxxx

The program cannot continue until the operator presses the start key on the console.

7074 Timing: 4 microseconds.

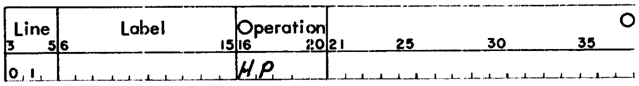
7070 Timing: 36 microseconds.

Comments: Like halt and branch, this code can be used in program testing, to stop the program at a predetermined point. The program register contains the halt and proceed instruction, when its contents are typed. The instruction counter contains the location of the HB instruction, +1.

Autocoder Example (Figure 77): Note that no operand is necessary. The assembled instruction is:

S01 *23* *456789*

−00 00 000000



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
01		HP	

Figure 77

No Operation

−01 **NOP**

Machine Description: No operation is performed. The next sequential instruction is executed.

Instruction Format

S01 −01.

23 Indexing word. Indexing always takes place, if an IW is specified here, even though positions 6-9 are not used.

456789 Not used.

Example: To perform no operation:

S01 *23* *456789*

−01 00 xxxxxx

7074 Timing: 4 microseconds.

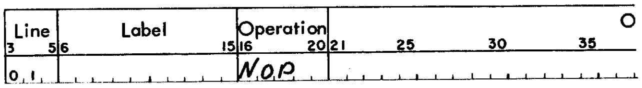
7070 Timing: 36 microseconds.

Comments: This code is used when the stored program is automatically brought to a location, but that instruction is not to be used. An example of this is an inquiry-read instruction, which takes the next sequential location as the next instruction, if an invalid character is detected on transmission of the request data from the synchronizer to core storage. To ignore this validity check, the no-op instruction is used. The next sequential instruction is taken; this is the instruction automatically taken if there is no validity error.

Autocoder Example (Figure 78): No operand is required. The assembled instruction is:

S01 *23* *456789*

−01 00 000000



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
01		NOP	

Figure 78

Index-Word Codes

These operation codes are used directly with the 99 index words in the computer—modifying them, testing them, loading and unloading them, etc. For all of these operations, the index word involved is indicated by positions 4-5 of the instruction. Positions 6-9 can be indexed by an indexing word specified by the IW portion (positions 2-3), just as for any other instructions. There is no field definition in the index-word codes, because positions 4-5 of the instructions are always used to denote the index word being operated on.

The format of an index word is:

S	*01*	*2345*	*6789*
sign	not used	indexing portion	non-indexing portion

The non-indexing portion can be used for decrements, limits, and constants. An index word operated on by one of these codes can be plus, minus, or alpha. The sign of the index word is controlled by the indexing portion; any operation that causes a sign change in the indexing portion changes the sign of the entire index word. Alpha index words are treated as plus in index-word operations that involve arithmetic.. (An alpha index word cannot be used for indexing; however, an index word specified in positions 4-5 of these codes can be alpha, but an alpha index word can never be specified in digits 2-3 of any instruction.)

The index-word operation codes are categorized and presented in Figure 79.

The letter X is used in all Autocoder symbols for the index-word instructions; it is not used in any other instructions. All of the index-word operations are in the 40's; position 0 of the instruction is always 4.

The field-overflow and sign-change indicators are never affected by an index-word instruction. Neither indicator can ever be turned on by an index-word load, unload, or modify instruction.

The 7074 additional storage mode (an optional feature) does not affect the operation of index word codes XSN (+48), BXM (−44), XL (+45), XLIN (−48), and XU (−45).

Index Word Load
+45 **XL**

Machine Description: The entire contents, including sign, of the word addressed by positions 6-9 (indexable) are brought to the index word specified by positions 4-5, replacing what was there.

Instruction Format

S01	+45.
23	Indexing word, for modifying the address in positions 6-9.
45	The index word that is to contain the contents of the storage word.
6789	Address of the word in storage (indexable).

Example: To move the entire contents, including sign, of word 2442 to index word 33:

S01	*23*	*45*	*6789*
+45	00	32	2442

After the operation, the contents of word 2442 and 0033 are identical.

7074 Timing: 12 microseconds.

7070 Timing: 36 microseconds.

7074 Additional Storage Mode: No change.

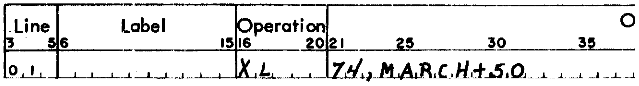
CATEGORY	NUMERIC OPERATION CODE	MNEMONIC OPERATION CODE	NAME
Load-Unload	+45	XL	Index Word Load
	−45	XU	Index Word Unload
	−48	XLIN	Index Word Load With Interchange
Modify	+46	XZA	Index Word Zero And Add To Indexing Portion
	−46	XZS	Index Word Zero And Subtract From Indexing Portion
	+47	XA	Index Word Add To Indexing Portion
	−47	XS	Index Word Subtract From Indexing Portion
	+48	XSN	Index Word Set Non-Indexing Portion
Branch	−44	BXM	Branch If Index Word Is Minus
	+44	BXN	Branch If Index Word Indexing Portion Is Non-Zero
	−43	BCX	Branch Compared Index Word
Modify And Branch	+49	BIX	Branch Incremented Index Word
	−49	BDX	Branch Decremented Index Word

Figure 79. Index Word Operation Codes

Comments: Another index word can be addressed by this instruction. In all cases, this code moves all ten digits and the sign to the index word specified by positions 4-5. Accumulators can also be addressed by this instruction.

Autocoder Example (Figure 80): Assume that MARCH has been defined as word 3250. The assembled instruction is:

S01 *23* *45* *6789*
+45 00 74 3300



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		XL	74, MARCH+50

Figure 80

Index Word Unload
−45 XU

Machine Description: The entire contents, including sign, of the index word specified by positions 4-5 are moved to the storage word addressed by positions 6-9 (indexable).

Instruction Format

S01 −45.
23 Indexing word, for modifying the address in positions 6-9.
45 The index word, the contents of which are to be transmitted to the storage word.
6789 Address of the storage word (indexable).

Example: To move the contents of index word 92 to word 1950; indexed by positions 2-5 of IW 37:

S01 *23* *45* *6789*
−45 37 92 1950

7074 Timing: 12 microseconds.

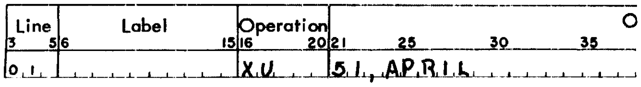
7070 Timing: 36 microseconds.

7074 Additional Storage Mode: No change.

Comments: The index word is not changed by this operation. Accumulator addresses and index-word addresses can be used. If an accumulator is addressed, this code is the same as zero-and-add, with a full word field-defined.

Autocoder Example (Figure 81): Assume that APRIL has been defined as word 1830. The assembled instruction is:

S01 *23* *45* *6789*
−45 00 51 1830



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		XU	51, APRIL

Figure 81

Index Word Load with Interchange
−48 XLIN

Machine Description: The contents of the storage word addressed by positions 6-9 (indexable) are brought to the index word indicated in positions 4-5. Positions 2-5 and 6-9 of the storage word are interchanged in the index word; the contents of positions 6-9 of the storage word go to positions 2-5 of the index word and vice-versa. The sign and positions 0-1 of the storage word go to their corresponding positions in the index word.

Instruction Format

S01 −48.
23 Indexing word, modifying the address in positions 6-9.
45 The index word to be loaded.
6789 Address of the core storage word, the contents of which are to be transmitted to the index word, with positions 2-5 and 6-9 interchanged.

Example: To load the contents of word 2358 into index word 62, with positions 2-5 and 6-9 interchanged:

S01 *23* *45* *6789*
−48 00 62 2358

Contents of 2358: −12 3456 7890. Contents of index word 62 after the operation: −12 7890 3456, regardless of previous contents.

7074 Timing: 12 microseconds.

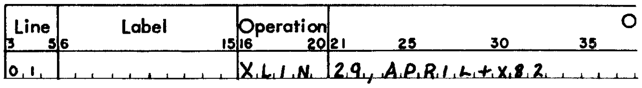
7070 Timing: 48 microseconds.

7074 Additional Storage Mode: No change.

Comments: The storage word is unchanged. Accumulator and index-word addresses can be used with this instruction. The indexing and non-indexing portions of an index word can be reversed by using this instruction with the same index-word specified in the address (contents of positions 4-5 the same as 8-9, with 00 in positions 6-7).

Autocoder Example (Figure 82): Assume APRIL to be defined as word 1830. The assembled instruction is:

S01 *23* *45* *6789*
−48 82 29 1830



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		XLIN	29, APRIL+X82

Figure 82

Index Word Zero and Add to Indexing Portion
+46 XZA

Machine Description: The four-digit number in positions 6-9 of the instruction replaces the previous contents of the indexing portion of the specified index word. The sign of the index word is set to plus. The remaining positions of the index word are unchanged.

Instruction Format

S01 +46.

23 Indexing word. Positions 6-9 of this instruction are indexable, just as if they represented an address.

45 The index word, positions 2-5 of which are to be replaced with positions 6-9 of the instruction.

6789 Not an address, but the actual 4-digit number to be inserted into the index word.

Example: To place the number 0365 into positions 2-5 of index word 78, and set its sign to plus:

S01 *23* *45* *6789*
+46 00 78 0365

7074 Timing: 12 microseconds.

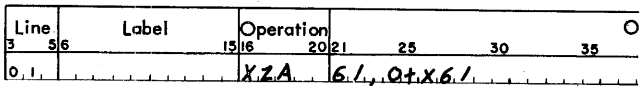
7070 Timing: 48 microseconds.

7074 Additional Storage Mode: The five-digit address in the instruction replaces the contents of the indexing portion (1-5) of the specified index word, and the sign of the index word is set to plus.

Comments: The contents of positions 6-9 are considered plus, and the sign of the index word is automatically set to plus, regardless of the previous sign. This instruction automatically makes index-word 61 plus, regardless of its previous sign:

S01 *23* *45* *6789*
+46 61 61 0000

This instruction in Autocoder is shown in Figure 83.

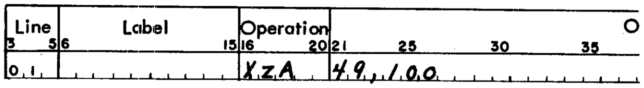


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		XZA	61,0+X61

Figure 83

Autocoder Example (Figure 84): The assembled instruction is:

S01 *23* *45* *6789*
+46 00 49 0100



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		XZA	49,100

Figure 84

Index Word Zero and Subtract from Indexing Portion
−46 XZS

Machine Description: The four-digit number in positions 6-9 of the instruction replaces the previous contents of the indexing portion of the specified index word. The sign of the index word is set to minus. The remaining positions of the index word are unchanged.

Instruction Format: Same as ZERO AND ADD TO INDEXING PORTION, except for the sign.

Example: To place the number 2399 into positions 2-5 of index word 27, and set its sign to minus:

S01 *23* *45* *6789*
−46 00 27 2399

7074 Timing: 12 microseconds.

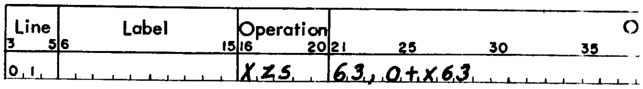
7070 Timing: 48 microseconds.

7074 Additional Storage Mode: The five-digit address in the instruction replaces the contents of the indexing portion (1-5) of the specified index word, and the sign of the index word is set to minus.

Comments: The index-word sign is set to minus, regardless of its previous sign. This is the only difference between this code and +46. This instruction automatically makes index word 63 minus:

S01 *23* *45* *6789*
−46 63 63 0000

This instruction in Autocoder is shown in Figure 85.

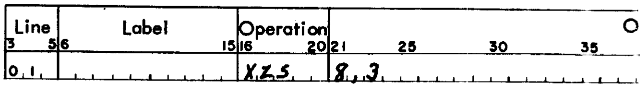


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		XZS	63,0+X63

Figure 85

Autocoder Example (Figure 86): The assembled instruction is:

S01 *23* *45* *6789*
−46 00 08 0003



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		XZS	8,3

Figure 86

Index Word Add to Indexing Portion
+47 XA

Machine Description: The four-digit number in positions 6-9 of the instruction, considered plus, is algebraically added to the indexing portion of the specified index word, the sign of which determines whether the sum or difference of the values will be obtained. The result replaces the previous indexing portion of the index word, and the sign of the result becomes the sign of the index word. The remaining positions of the index word are unchanged.

Instruction Format

S01 +47.

23 Indexing word. Positions 6-9 of this instruction are indexable, just as if they represented an address.

45 The index word that contains, in positions 2-5, one of the factors in the operation, and the result of the operation.

6789 Not an address, but an actual four-digit number, which is a factor in the operation.

Examples: To add the number 25 to the indexing portion of index word 53, and store the result in the indexing portion:

S01 23 45 6789
+47 00 53 0025

Contents of index word 53 before the operation: +00 1234 5678. Contents after the operation: +00 1259 5678.

To add the number 150 to the indexing portion of index word 80 and store the result in the indexing portion:

S01 23 45 6789
+47 00 80 0150

Contents of index word 80 before the operation: −22 0125 0010. Contents after the operation: +22 0025 0010. The sign of index word 80 is changed.

7074 Timing: 14 microseconds.

7070 Timing: 60 microseconds without recomplement, 84 microseconds with recomplement.

7074 Additional Storage Mode: The five-digit address in the instruction is added algebraically to the indexing portion (1-5) of the specified index word.

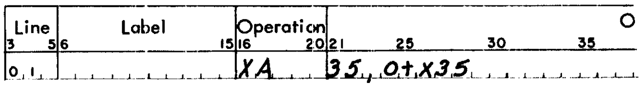
Comments: If a carry to a fifth position results from the addition of the two factors, the high-order 1 is ignored, and the four low-order digits of the result are stored in the index word. There is no field-overflow indication given; it acts as if there had not been a carry.

If the sign of the index word is plus or alpha, this operation obtains the sum of the factors. If it is minus, the difference is obtained. A change in the sign of the index word can occur only if the index-word sign was originally minus and the factor in the instruction is greater than that in positions 2-5 of the index word. The sign change can be from minus to plus only. The sign-change indicator is never turned on by this operation.

This instruction doubles the value in the indexing portion of index word 35:

S01 23 45 6789
+47 35 35 0000

This holds true only if the value does not reach five digits when it is doubled. This instruction in Autocoder is shown in Figure 87.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		XA	35,0+X35

Figure 87

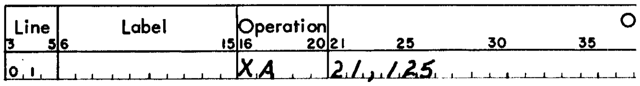
If indexing is used with an XA instruction, and the word used for indexing is minus, a complement number is developed if the indexing value is greater than the number in positions 6-9 of the instruction. Take this instruction as an example:

S01 23 45 6789
+47 14 35 0000

Contents of IW 14: −xx 0007 xxxx. Contents of 0035: −xx 0004 xxxx. Because the IW is minus, the 4-digit factor after indexing becomes +9993 (complement of 0007); it is not recomplemented. The XA instruction adds this factor, considered plus, to the −0004 in the indexing portion of 0035, obtaining a result of +9989. Index word 35 is thus changed to +xx 9989 xxxx by this instruction.

Autocoder Example (Figure 88): The assembled instruction is:

S01 23 45 6789
+47 00 21 0125



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		XA	21,125

Figure 88

Index Word Subtract from Indexing Portion −47 XS

Machine Description: The four-digit number in positions 6-9 of the instruction, considered plus, is algebraically subtracted from the indexing portion of the specified index word, the sign of which determines whether the sum or difference of the values will be obtained. The result replaces the previous indexing portion of the index word, and the sign of the result becomes the sign of the index word. The remaining positions of the index word are unchanged.

Instruction Format: Same as ADD TO INDEXING PORTION, except for the sign.

Examples: To subtract the number 27 from the number in positions 2-5 of index word 59, and store the result in the indexing portion:

S01 23 45 6789
−47 00 59 0027

Contents of index word 59 before the operation: +00 1321 7428. Contents after the operation: +00 1294 7428.

To subtract the number 50 from positions 2-5 of index word 9 and store the result in the indexing portion:

S01 23 45 6789
−47 00 09 0050

Contents of index word 9 before the operation: −00 0250 0000. Contents after the operation: −00 0300 0000.

7074 Timing: 14 microseconds.

7070 Timing: 60 microseconds without recomplement, 84 microseconds with recomplement.

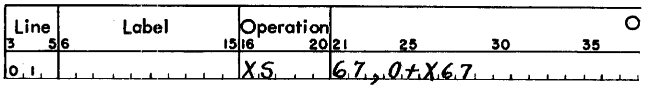
7074 Additional Storage Mode: The five-digit address in the instruction is subtracted algebraically from the indexing portion (1-5) of the specified index word.

Comments: The comments concerning carry, discussed under ADD TO INDEXING PORTION, apply here. If the index-word sign is minus, this operation obtains the sum of the factors. If it is plus or alpha, the difference is obtained. If it is plus, sign change occurs if the instruction value is greater. If it is alpha, it remains alpha even if the instruction value is greater. Sign change can be from plus to minus only. (The sign-change indicator is not turned on, however.) In all cases, the correct arithmetic result is obtained.

This instruction automatically reduces the indexing portion of index word 67 to zero:

S01	*23*	*45*	*6789*
−47	67	67	0000

This instruction in Autocoder is shown in Figure 89.

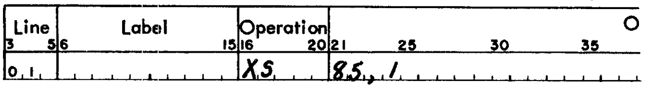


Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		XS	67,0+X67

Figure 89

Autocoder Example (Figure 90): The assembled instruction is:

S01	*23*	*45*	*6789*
−47	00	85	0001



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		XS	85,1

Figure 90

Index Word Set Non-Indexing Portion
+48 XSN

Machine Description: The non-indexing portion of the specified index word is replaced with the four-digit number in positions 6-9 of the instruction. The sign and the remaining positions of the index word are not changed.

Instruction Format

S01 +48.

23 Indexing word. Positions 6-9 of this instruction are indexable, just as if they represented an address.

45 The index word, the non-indexing portion of which is to be set with the four-digit number.

6789 Not an address, but the actual four-digit number to be inserted into positions 6-9 of the index word.

Examples: To insert a 1 into the non-indexing portion of index word 16:

S01	*23*	*45*	*6789*
+48	00	16	0001

Contents of index word 16 before the operation: +00 0025 0124. Contents after the operation: +00 0025 0001.

7074 Timing: 12 microseconds.

7070 Timing: 36 microseconds.

Comments: This instruction has almost the same function for the non-indexing portion of an index word, as the ZERO AND ADD TO INDEXING PORTION code (+46) has for the indexing portion. The only difference is with the signs; this instruction has no effect on the sign of the index word.

This instruction makes the non-indexing portion of index word 47 equal to the indexing portion if its sign is plus:

S01	*23*	*45*	*6789*
+48	47	47	0000

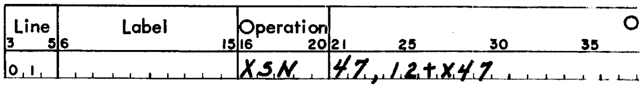
If there is a factor in positions 6-9, the instruction makes the non-indexing portion greater than the indexing portion, by that factor. This instruction makes the non-indexing portion of index word 47, 12 greater than the indexing portion (provided that it is plus):

S01	*23*	*45*	*6789*
+48	47	47	0012

If word 47 is minus, this instruction makes the non-indexing portion equal to the four low-order positions of the sum of the tens complement of the indexing portion, plus 0012. This instruction in Autocoder is shown in Figure 91.

This instruction sets positions 6-9 of index word 28 to 0000:

S01	*23*	*45*	*6789*
+48	00	28	0000

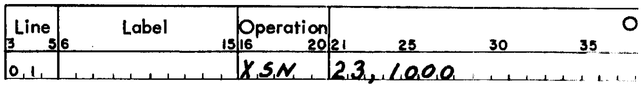


Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		XSN	47,12+X47

Figure 91

Autocoder Example (Figure 92): The assembled instruction is:

S01	*23*	*45*	*6789*
+48	00	23	1000



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		XSN	23,1000

Figure 92

Branch if Index Word Is Minus

−44 **BXM**

Machine Description: The sign of the index word specified by positions 4-5 is tested for minus. If it is minus, the program branches to the location specified in positions 6-9 of the instruction (indexable). If the sign is plus or alpha, the program does not branch. The contents of the index word have no effect on the operation.

Instruction Format

S01	−44
23	Indexing word for modifying the branch address in positions 6-9.
45	The index word, the sign of which is to be tested.
6789	Branch address, if the index-word sign is minus.

Example: To branch to location 2930, if index word 58 is minus:

S01	*23*	*45*	*6789*
−44	00	58	2930

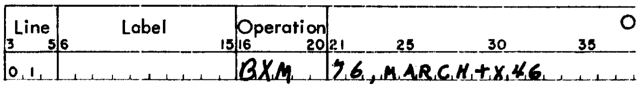
7074 Timing: 8 microseconds.

7070 Timing: 36 microseconds.

Comments: The contents of the index word are ignored. The index word is not changed by this operation.

Autocoder Example (Figure 93): Assume MARCH to be defined as word 2050. The assembled instruction is:

S01	*23*	*45*	*6789*
−44	46	76	2050



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		BXM	76,MARCH+X46

Figure 93

Branch if Index Word Indexing Portion Is Non-Zero

+44 **BXN**

Machine Description: The indexing portion (positions 2-5) of the index word specified in positions 4-5 of the instruction is tested for zero. If any of the four digits is other than zero, the program branches to the location specified in positions 6-9 of the instruction (indexable). If all four digits are zero, the program does not branch.

Instruction Format

S01	+44
23	Indexing word for modifying the branch address in positions 6-9.
45	The index word, the indexing portion of which is to be tested for non-zero.
6789	Branch address if the indexing portion is non-zero (indexable).

Example: To branch to location 3817, if positions 2-5 of index word 56 are not all zeros:

S01	*23*	*45*	*6789*
+44	00	56	3817

Contents of index word 56: −00 0000 1234. The program does not branch.

7074 Timing: 10 microseconds.

7070 Timing: 36 microseconds.

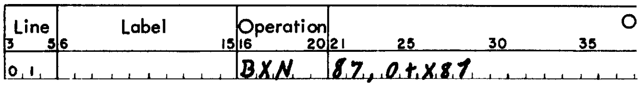
7074 Additional Storage Mode: For the non-zero test, position 1 of the specified index word is ignored.

Comments: The index word is not changed by this instruction. Like BRANCH COMPARED INDEX WORD, this code can be used to control a program loop. Each time, an instruction in the loop subtracts a one from the indexing portion. This instruction branches back to the start of the loop until the indexing portion is reduced to zero. The program then proceeds with the next sequential instruction.

This instruction tests the address in the indexing portion of index word 87 for zero and branches to that address if it is not 0000:

S01	*23*	*45*	*6789*
+44	87	87	0000

In Autocoder, this instruction is shown in Figure 94.

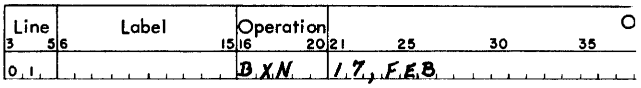


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		BXN	87,0+X87

Figure 94

Autocoder Example (Figure 95): Assume that FEB has been defined as word 1803. The assembled instruction is:

S01	*23*	*45*	*6789*
+44	00	17	1803



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		BXN	17,FEB

Figure 95

Branch Compared Index Word

−43 **BCX**

Machine Description: The value of the indexing portion of the index word specified in positions 4-5 is compared to the value of the non-indexing portion. If the indexing portion is less than, or equal to, the non-indexing portion, the program branches to the location in positions 6-9 (indexable) for its next instruction. If the indexing portion is greater than the non-indexing portion, the program does not branch. The absolute values of the two portions are compared; the sign of the index word is ignored.

Instruction Format

S01 −43

23 Indexing word for modifying the address in positions 6-9.

45 The index word on which the operation is to be performed.

6789 Branch address if the indexing portion is less than, or equal to, the non-indexing portion (indexable).

Example: To branch to location 2930 if the contents of positions 2-5 of index word 77 are not greater than the contents of positions 6-9:

S01	*23*	*45*	*6789*
−43	00	77	2930

Contents of index word 77: +00 1234 1234. The two portions are equal; the program branches to location 2930.

7074 Timing: 10 microseconds.

7070 Timing: 48 microseconds.

7074 Additional Storage Mode: For the comparison, position 1 of the specified index word is ignored.

Comments: The index word itself is not changed by this instruction. Note that the index-word sign is ignored and the absolute values of the two portions are compared.

This instruction can be used when the program is in a loop to keep track of the number of times the loop has been executed and to branch out of it after a specified number of times. The difference between the values in the indexing and non-indexing portions represents the number of times the program should go through the loop. An instruction in the loop decreases the indexing-portion value by one, each time it is performed. When the two portions become equal, the program proceeds out of the loop.

Autocoder Example (Figure 96): Assume that JAN has been previously defined as word 2711. The assembled instruction is:

S01	*23*	*45*	*6789*
−43	13	41	2711

Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
01		BCX	41,JAN+X13

Figure 96

Branch Incremented Index Word

+49 **BIX**

Machine Description: The absolute value of the contents of the indexing portion of the specified index word is increased by 1. This new indexing portion is compared with the non-indexing portion. If the absolute value of this new indexing portion is not greater than the absolute value of the non-indexing portion, the program branches to the location in positions 6-9 (indexable). If the new indexing portion is greater, the program does not branch.

Instruction Format

S01 +49

23 Indexing word for modifying the branch address in positions 6-9.

45 The index word, the indexing portion of which is increased by 1 and compared with the non-indexing portion.

6789 The branch address (indexable) if the new indexing-portion value is not greater than the non-indexing value.

Examples: To increase the indexing portion of index word 75 by 1, and branch to location 2201 if it is not greater than the non-indexing portion:

S01	*23*	*45*	*6789*
+49	00	75	2201

Contents of 0075 before the operation: +00 3329 3330. Contents after the operation: +00 3330 3330. The program branches, because the indexing portion is not greater than the non-indexing portion.

To increase the indexing portion of index word 19 by 1, and branch to location 1651 if it is not greater than the non-indexing portion:

S01	*23*	*45*	*6789*
+49	00	19	1651

Contents of index word 19 before the operation: −00 0012 0012. Contents after the operation: −00 0013 0012. The program does not branch. Note that the absolute value in positions 2-5 is increased; the minus sign is ignored.

7074 Timing: 14 microseconds.

7070 Timing: 84 microseconds.

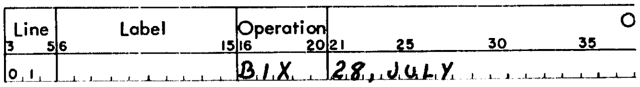
7074 Additional Storage Mode: Positions 1-5 of the specified index word are incremented by +1, but position 1 is ignored in the comparison of indexing portion with non-indexing portion. The no-branch condition occurs only when the difference between the indexing portion (2-5) and non-indexing portion (6-9) is +0001. Any other difference value causes the branch condition.

Comments: This code is an efficient means of controlling a program loop. The number of times the program executes the loop is determined by the difference in the values of the indexing and non-indexing portions of the index word. (The non-indexing portion contains the greater value.) For example, a program loop that uses data in word 0501, then 0502, etc., through 0525 (executing the loop 25 times) can use this instruction to control the addressing, and determine the end of the loop. At the start of the first loop, index word 45 is +00 0001 0025. An instruction referring to the data has an address of 0500, indexed by index word 45. At the end of the first loop, a BRANCH INCREMENTED INDEX WORD instruction changes IW 45 to +00 0002 0025 and branches back to the start of the loop. The loop is executed 24 more times; IW 45 is then +00 0026 0025, and the program does not again branch to start the loop. Each time the loop is executed, the correct data word, from 0501 to 0525, is used, because the address 0500 was indexed each time by IW 45.

The indexing portion is always increased in absolute value; the sign is ignored. If the indexing portion should be 9999, it is changed to 0000 (but the field-overflow indicator is not turned on).

Autocoder Example (Figure 97): Assume JULY to be defined as location 2631. The assembled instruction is:

S01	*23*	*45*	*6789*
+49	00	28	2631



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		BIX	28,JULY

Figure 97

Branch Decremented Index Word

−49 **BDX**

Machine Description: The non-indexing portion of the specified index word is subtracted from the indexing portion. Because these factors are in the same word, they have the same sign, and this operation always obtains the difference of the factors. This result replaces the previous indexing portion. The sign of the result becomes the sign of the index word. The remaining positions of the index word are unchanged.

If the subtraction does not cause a zero result or sign change (if the non-indexing portion was smaller than the indexing portion), the program branches to the location in positions 6-9 (indexable). If the subtraction causes a zero result or sign change (if the non-indexing portion was equal to or greater than the indexing portion), the program does not branch.

Instruction Format

S01	−49.
23	Indexing word, for the branch address.
45	The index word, which contains the two 4-digit factors to be subtracted, and the result.
6789	Branch address (indexable), if the result of the subtraction does not produce a zero result or sign change.

Examples: To subtract the non-indexing portion of index word 52 from the indexing portion, place the result in the indexing portion, and branch to location 3700, unless the result is zero or less:

S01	*23*	*45*	*6789*
−49	00	52	3700

Contents of index word 52 before the operation: +00 0040 0005. Contents after the operation: +00 0035 0005. The program branches.

To subtract the non-indexing portion of index word 40 from the indexing portion, place the result in the indexing portion, and branch to location 1401, unless the result is zero or less:

S01	*23*	*45*	*6789*
−49	00	40	1401

Contents of index word 40 before the operation: −00 0005 0020. Contents after the operation: +00 0015 0020. The program does not branch. Note the sign change.

7074 Timing: 14 microseconds.

7070 Timing: 60 microseconds without recomplement, 84 microseconds with recomplement.

7074 Additional Storage Mode: For this operation, position 1 of the specified index word is not involved in the manipulation.

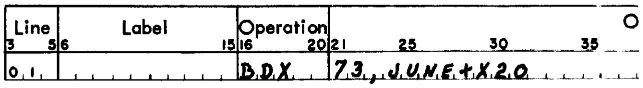
Comments: This instruction is a means of controlling a program loop. The number of times the loop is executed is determined by the number of times the value in the non-indexing portion can be subtracted from the value in the indexing portion before reaching or passing zero. In the same manner as for BRANCH INCREMENTED INDEX WORD, this instruction, BRANCH DECREMENTED INDEX WORD, can be used to modify addresses within the loop, as well as determine the number of times the loop is executed. Take, for example, a program that uses data in word 1727, then 1724, then 1721, etc., through 1700 (the loop is executed ten times). Index word 68 contains +00 0030 0003, at the start of the first loop. Any instruction within the loop that refers to the data has an address of 1697, indexed by IW 68. At the end of the first loop, BRANCH DECREMENTED INDEX WORD changes IW 68 to +00 0027 0003 and returns the program to the first instruction in the loop. Each time through the loop, the data word with an address three less than the previous one is used, because the address 1697 is indexed by IW 68. When IW 68 is reduced to +00 0000 0003, the program does not again branch to the start of the loop, and thus word 1697 is not used. To process the words in ascending order, 1700-1727, make IW 68 minus: −00 0030 0003. Any instruction within the loop that refers to the data has an address of 1730, indexed by IW 68. The indexing portion will still be reduced to 0027, 0024, etc., by the BRANCH DECREMENTED INDEX WORD instruction.

This code differs from BRANCH INCREMENTED INDEX WORD in these respects:

1. The indexing portion is decreased in absolute value instead of increased.
2. The indexing portion is changed by the value in the non-indexing portion instead of by a constant of 1.
3. The sign of the index word can be changed. (Neither instruction can change the contents of positions 0-1 or 6-9, however.)

Autocoder Example (Figure 98:) Assume that JUNE has been previously defined as word 3500. The assembled instruction is:

S01	*23*	*45*	*6789*
−49	20	73	3500



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		BDX	73,JUNE+X20

Figure 98

Block Transmission

One of the outstanding programming features of the system is the ability to move a group or several groups of words in a single instruction. The number of words moved and the number of blocks they are divided into are limited only by the capacity of core storage available.

This feature (called *Block Transmission)* is available in the following types of operations:

Core-to-Core Block Transmission—Blocks of data moved from core storage to other parts of core storage.

Table Lookup—Not a transmission of blocks of data in storage, but the blocks make up the table to be searched.

Magnetic Tape—All reading and writing of magnetic tape.

Disk Storage—All reading and writing in the disk-storage units.

Unit Record—All movement of data from the input synchronizers to the output synchronizers and to the console typewriter.

Inquiry—All movement of data from and to the inquiry-control 1 and 2 synchronizers.

In all of these operations, the locations in core storage are divided into *blocks*. Data in a tape record, disk-storage track, or a synchronizer area are, of course, all in one record. In core-to-core block transmission, either the *from* locations or the *to* locations can be divided into blocks, but not both.

Scatter Read-Write: The scattering of data from a tape record, disk track, or input-output synchronizer area into a number of blocks of core storage, and the gathering of data from the blocks to tape, disk, or synchronizer, are called *scatter read-write.*

Record Definition Words: The locations of the various blocks of storage to be read into or written from are defined by *Record Definition Words* (also referred to in this text as RDW's). In block-transmission instructions, the address portion (positions 6-9) contains the location of the first record definition word to be used. A record definition word contains the first and last addresses of a block of storage words to be read into, or written from.

Positions 2-5 of the RDW contain the address of the first word in the block (called the *starting address*); positions 6-9 contain the address of the last word in the block (called the *stop address*). In a block-transmission operation the RDW is placed for use in a *record-definition register.*

The start address is incremented by one for each word moved and thus becomes the *working address.*

The working address is compared with the stop address on an equal-unequal basis. If the start address of the record-definition word should be higher than the stop address, the block transmission continues until the address capacity of core storage is reached (word 9990), at which time an error is indicated. The start address must therefore be smaller than the stop address in all record-definition words.

When the working address matches the stop address the sign of the RDW is considered. If it is plus, the next sequential word following the RDW location is moved to the record-definition register and used as the RDW for another block transmission. An RDW with a minus sign limits the block-transmission operation to its own stop address.

In this manner, several blocks of data may be transmitted by record definition words chained together by plus signs. The operation is terminated with an RDW having a minus sign.

Additional Storage Mode Record Definition Words: Record definition words (RDW's) must be stored in addresses 00000 through 09999 when the 7074 is operating in additional storage mode (optional feature). The RDW start address occupies positions 1 through 5 of the word, and positions 6-9 are the four low-order digits of the stop address. If the four-digit number in the stop address part of the RDW is smaller in value than the four low-order digits of the start address, the high-order (ten thousands) digit of the stop address is considered to be one greater than the high-order digit of the start address. For example, in the RDW —01 8750 8500, the full five digit stop address is effectively 28500.

It is possible to move an entire block of words from one part of magnetic-core storage to another, in a single instruction. There is no limit to the number of words moved, short of the capacity of core storage available. This operation uses the Process Channel Control. The scatter read/write feature in tape and disk operations is also available in core-to-core block transmissions. It works in much the same way, with the blocks defined by record definition words. A block of words in one part of storage can be scattered into a number of smaller blocks in another part of storage, or can be gathered from a number of blocks. These operations are called *record scatter* and *record gather*, respectively. They are shown schematically in Figure 99.

Core-to-core transmissions differ from other operations that use record definition words, in that two sets of addresses are needed — the core-storage locations that data are moved from, and the locations that data are moved to. Record definition words are used to denote one of these sets of addresses — the *several* blocks that data are scattered *into*, or the *several* blocks that data are gathered *from*.

The *single* block of core-storage locations that data are *scattered from* or *gathered into* is defined in an index word, specified by positions 4-5 of the core-to-core block-transmission instructions. The indexing portion of that word contains the address of the first word of the block.

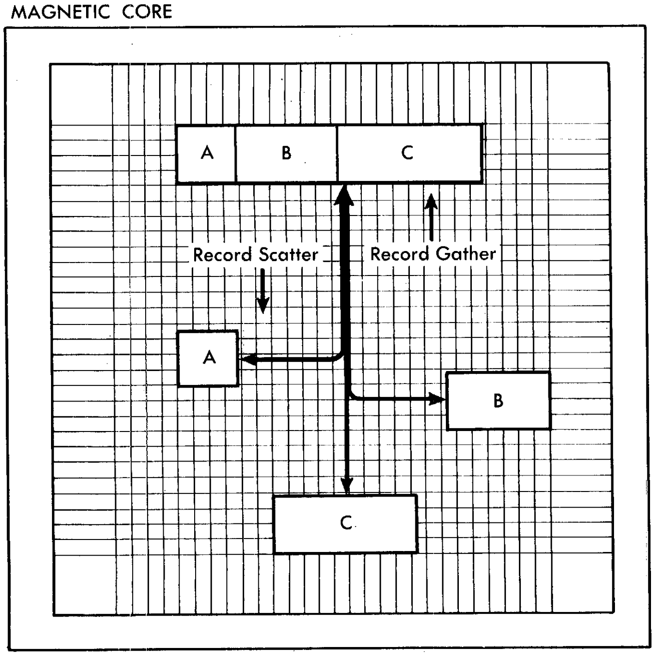


Figure 99. Core-to-Core Block Transmission

Alpha/Numeric Conversion: Numeric words can be converted to alpha and vice versa, in core-to-core block transmission. In numeric-to-alpha conversion, every word becomes two words in alpha coding, with alpha signs. In alpha-to-numeric conversion, every two alpha words are translated into one numeric word by taking the digits in positions 1, 3, 5, 7, and 9 of each word.

There are six operation codes in core-to-core block transmission, presented in the following sequence.

CATEGORIES	OP CODES	NAMES	MNEMONICS
Simple transmission	+65	Record scatter	RS
	−65	Record gather	RG
Alpha/numeric conversion	+56	Edit numeric to alphameric	ENA
	−56	Edit numeric to alphameric with sign control	ENS
	+57	Edit numeric to alphameric with blank insertion	ENB
	−57	Edit alphameric to numeric	EAN

The RG and RS symbols are the only Autocoder symbols that start with R.

Record Scatter
+65 RS

Machine Description: The contents of a single block of core-storage words, the starting address of which is defined by positions 2-5 of the index word specified by positions 4-5 of the instruction, are moved to one storage block or a series of blocks, under control of record-definition word(s), the first of which is located by positions 6-9 (indexable) of the instruction. Movement of data is under control of the process channel control.

Instruction Format

S01 +65.

23 Indexing word for modifying the RDW address in positions 6-9.

45 An index word, positions 2-5 of which contain the location of the first storage word, the contents of which are to be moved.

6789 Location of the first (or only) record-definition word, used to define the area to receive the transmitted data.

Example: To transmit the contents of a block of storage words, the starting address of which is in positions

2-5 of index word 86, to two blocks of storage, the first of which is defined by the RDW in location 2841:

S01 *23* *45* *6789*
+65 00 86 2841

Contents of index word 86: +00 1721 0000. Contents of word 2841: +00 3511 3520. Contents of word 2842: −00 3101 3105. This instruction moves the contents of words 1721-1730 to words 3511-3520, and the contents of words 1731-1735 to words 3101-3105.

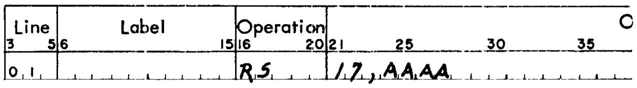
7074 Timing: 8 microseconds setup time, plus 4 microseconds for each RDW, plus 8 microseconds for each word moved.

7070 Timing: 36 microseconds setup time, plus 36 microseconds for each RDW, plus 24 microseconds for each word moved.

Comments: If only one record-definition word is used, this is not strictly a record-scatter operation. It is merely the movement of data from one block of core-storage words to another. This instruction can be used to move a storage block a few locations *up* in storage, replacing all but the last few words in the original block. For example, a storage block in locations 1217-1226 can be moved *up* two locations, to 1215-1224. (After the operation the contents of 1224 equal those of 1226, and 1223 = 1225.) This cannot be done in the opposite direction, however; 1217-1226 could not be moved to 1219-1228 by a single RS instruction, for example. The same two words of data would be in 1217-1218, 1219-1220, 1221-1222, etc.

Autocoder Example (Figure 100): Assume that AAAA has been previously defined as word 2618. The assembled instruction is:

S01 *23* *45* *6789*
+65 00 17 2618



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		RS	17,AAAA

Figure 100

Record Gather

−65 **RG**

Machine Description: The address in positions 6-9 (indexable) specifies the first (or only) record-definition word that defines the *from* area. The starting address of the *to* area is specified in positions 2-5 of the index word designated in positions 4-5 of the instruction. Movement of data is under control of the process channel control.

Instruction Format

S01 −65.
23 Indexing word, for modifying the RDW address in positions 6-9.
45 An index word, positions 2-5 of which contain the location of the first storage word to be read into.
6789 Location of the first (or only) record-definition word, used to define the transmitting area.

Example: To gather the contents of three blocks of storage words, the first of which is defined by the RDW in location 3441, into a single block, the starting address of which is specified in positions 2-5 of index word 77:

S01 *23* *45* *6789*
−65 00 77 3441

Contents of index word 77: +00 2001 1234. Contents of word 3441: +00 1401 1405. Contents of word 3442: +00 4110 4115. Contents of word 3443: −00 2710 2713. This instruction moves the contents of 1401-1405 to 2001-2005, the contents of 4110-4115 to 2006-2011, and the contents of 2710-2713 to 2012-2015.

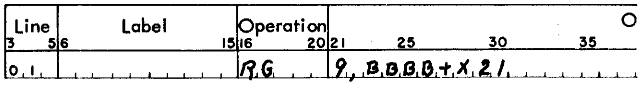
7074 Timing: 8 microseconds setup time, plus 4 microseconds for each RDW, plus 8 microseconds for each word moved.

7070 Timing: 36 microseconds setup time, plus 36 microseconds for each RDW, plus 25 microseconds for each word moved.

Comments: Like record-scatter, record-gather using just one RDW is merely the movement of data from one block of core storage to another. A block can be moved *up* a few locations, as described under "Record Scatter."

Autocoder Example (Figure 101): Assume that BBBB has been previously defined as word 4550. The assembled instruction is:

S01 *23* *45* *6789*
−65 21 09 4550



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		RG	9,BBBB+X21

Figure 101

Edit Numeric to Alphameric

+56 **ENA**

Machine Description: This is a record-scatter type of operation. The single block of storage words, located by the starting address in positions 2-5 of the index word specified in positions 4-5 of the instruction, is transmitted to the locations defined by record-definition words, the first of which (or only one) is addressed by positions 6-9 (indexable) of the instruction. The receiving and transmitting addresses cannot be the same.

The first word of numeric data (ten digits) is converted to two words (20 digits) of two-digit alpha representation. The signs of the numeric words are ignored. These two words, with alpha signs attached, are stored in the locations specified by the *start* and *start +1* addresses of the first record-definition word. The process is continued, converting words located sequentially in the numeric area and storing in consecutive locations, under the control of the process channel control. The operation terminates after the stop address of the last RDW is used.

Instruction Format

S01 +56.

23 Indexing word, for modifying the RDW address in positions 6-9.

45 An index word, positions 2-5 of which contain the location of the first numeric storage word to be read from.

6789 Location of the first (or only) record-definition word, used to define the receiving (alpha) area.

Example: To scatter the contents of a block of storage words, the starting address of which is in positions 2-5 of index word 68, and convert each word into two alpha words, the locations of which are defined by RDW's starting in location 548:

S01 *23* *45* *6789*
+56 00 68 0548

Contents of index word 68: +00 1875 0000. Contents of word 548: +00 1201 1212. Contents of word 549: +00 2151 2176. Contents of word 550: −00 1700 1705. The six words in locations 1875-1880 are converted to alpha and stored in the 12 locations 1201-1212. The next 13 words, in 1881-1893, are converted to alpha and stored in the 26 locations 2151-2176. The next three words, in 1894-1896, are converted to alpha and stored in the six locations 1700-1705.

7074 Timing: 8 microseconds setup time, plus 4 microseconds for each RDW, plus 12 microseconds for each word moved.

7070 Timing: 36 microseconds setup time plus 36 microseconds for each RDW, plus 120 microseconds for each *numeric* word moved.

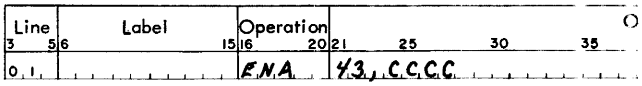
Comments: In numeric-to-alpha conversion, the lower of the two sequential alpha words converted from a numeric word contains alpha coding for the five high-order digits, and the second alpha word contains the alpha coding for the five low-order digits of the numeric word.

The record-definition words define the *alpha* area—the *to* locations in numeric-to-alpha conversion. The total area defined by RDW's is twice as big as the numeric area used. Each RDW must define an even number of storage words. Because the start and stop addresses are inclusive (i.e., the RDW defines the start *through* stop addresses), the difference of the two addresses must be an odd number.

This operation does not distinguish between plus and minus signs in the numeric words; the numeric-word signs are ignored.

Autocoder Example (Figure 102): Assume that CCCC has been previously defined as word 1905. The assembled instruction is:

S01 *23* *45* *6789*
+56 00 43 1905



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		ENA	43, CCCC

Figure 102

Edit Numeric to Alphameric with Sign Control −56 ENS

Machine Description: The code operates in the same way as EDIT NUMERIC TO ALPHAMERIC (ENA, +56) except: for all but the units digit of each numeric word, a 9 is inserted into the position to its left in the alpha word, thus producing the alpha representation (90-99) for that digit. The units digit of the low-order alpha word has a 6 or 7 inserted next to it; 6 if the sign of the numeric word is plus, and 7 if it is minus. For example, the word −12345 67890 becomes

@9192939495
@9697989970

If alphameric words are included in the "numeric" area to be converted, the tens position of each resultant low-order alpha word gets a digit 9. For example, the word @6162636465 becomes @9691969296
@9396949695

Instruction Format: Same as EDIT NUMERIC TO ALPHAMERIC, except for the sign.

Example: To scatter the contents of a block of storage words, the starting address of which is in positions 2-5 of index word 34, convert each word into two alpha words, the locations of which are defined by RDW's starting in location 407; and convert the signs to 6's, 7's or 9's in the tens digit of every second alpha word, 6 for plus, 7 for minus, and 9 for alpha:

S01 *23* *45* *6789*
−56 00 34 0407

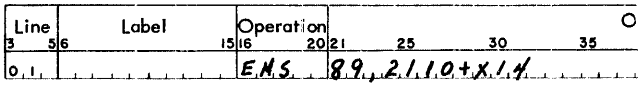
Contents of index word 34: +00 2417 0000. Contents of word 407: −00 0901 0920. The ten words in locations 2417-2426 are converted to alpha and stored in the 20 locations 901-920. The tens digits of words 902, 904, 906, etc. are all either 6, 7, or 9 depending on the signs of the words from which they were converted.

Timing: Same as EDIT NUMERIC TO ALPHAMERIC.

Comments: Use of this code enables the program to record the signs of the numeric records when they are converted to alpha. Note that the tens digit of each second alpha word is 6, 7 or 9 if this code is used, it is always 9 if ENA is used. In all cases, the words are given alpha signs.

Autocoder Example (Figure 103): Actual address 2110 is used. The assembled instruction is:

S01	*23*	*45*	*6789*
−56	14	89	2110



Line 3 5	Label 6 15	Operation 16 20	○ 21 25 30 35
01		ENS	89,2110+X14

Figure 103

Edit Numeric to Alphameric with Blank Insertion +57 ENB

Machine Description: This code operates in the same way as EDIT NUMERIC TO ALPHAMERIC (ENA, +56) except: zeros, instead of nines, are inserted in even digit positions (0, 2, 4, 6, 8) to the left of each leading zero from the numeric word. Thus, blank alphameric characters (00) appear to the left of the most significant digits in the resulting alphameric words. For example, the word +00002 57841 becomes @0000000092
@9597989491

Instruction Format: Same as EDIT NUMERIC TO ALPHAMERIC, except for the operation code.

Example: To scatter the contents of a block of storage words, the starting address of which is in positions 2-5 of index word 52, convert each word into two alpha words, the locations of which are defined by RDW's starting in location 1222; and convert each insignificant zero in the original numeric words to 00 instead of 90:

S01	*23*	*45*	*6789*
+57	00	52	1222

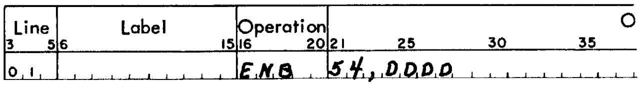
Contents of index word 52: −45 2300 0000. Contents of word 1222: +00 1001 1016. Contents of word 1223: −00 2713 2718. The eight words in locations 2300-2307 are converted to alpha and stored in the 16 locations 1001-1016. The next three words, in locations 2308-2310, are converted to alpha and stored in the six locations 2713-2718. High-order insignificant zeros in the numeric words are each converted to 00 instead of 90.

Timing: Same as EDIT NUMERIC TO ALPHAMERIC.

Comments: A numeric-to-alpha operation can use blank insertion (+57, ENB), sign control (−56, ENS), or neither (+56, ENA). Blank insertion and sign control combined cannot be used by a single instruction. If a numeric word is all zeros and the ENB instruction is used, two full alpha words of zeros are created.

Autocoder Example (Figure 104): Assume that DDDD has been previously defined as word 1719. The assembled instruction is:

S01	*23*	*45*	*6789*
+57	00	54	1719



Line 3 5	Label 6 15	Operation 16 20	○ 21 25 30 35
01		ENB	54,DDDD

Figure 104

Edit Alphameric to Numeric −57 EAN

Machine Description: This is a record-gather type of operation. Record-definition words, addressed by positions 6-9 (indexable) of the instruction, define blocks of alpha words, which are converted to numeric words and stored in a single storage block, the starting address of which is in positions 2-5 of the index word specified in positions 4-5 of the instruction.

Each pair of two sequential alpha words is converted into a single numeric word. Positions 1, 3, 5, 7, and 9 of the first alpha word become positions 0-4 of the numeric word; those same positions of the second alpha word become positions 5-9 of the numeric word. Position 8 of the second alphameric word is interrogated. If it is anything but a 7, a plus sign is assigned to the resultant numeric word; if 7, a minus sign is assigned. The operation is under control of the process channel control.

Instruction Format

S01 −57.

23 Indexing word, for modifying the RDW address in positions 6-9.

45 An index word, positions 2-5 of which contain the first storage location to be filled with the numeric data.

6789 Location of the first (or only) record-definition word, used to define the transmitting area.

Example: To gather the contents of one or several blocks of storage, the locations of which are defined by RDW's starting in location 2691, into a single block, the starting address of which is in positions 2-5 of index word 96; converting each two words into a single word:

S01	*23*	*45*	*6789*
−57	00	96	2691

Contents of index word 96: +00 4651 0000. Contents of word 2691: +00 0746 0755. Contents of word 2692: −00 1301 1302. The ten words in locations 746-755 are converted to numeric and stored into the five locations 4651-4655. The two words in locations 1301 and 1302 are converted to numeric and stored into word 4656. The tens digit of each second word, 747, 749, 751, etc. and 1302, is tested for 7 or non-7. For each 7, the sign of the corresponding numeric word is made minus; for each non-7, it is made plus.

7074 Timing: In microseconds:
8 for setup time
+4 for each RDW
+12 for each word moved

7070 Timing: In microseconds:
36 for setup time
+36 for each RDW
+72 for each *numeric* word moved

Comments: The only kind of data that should be converted by this code is numeric information in the two-digit alpha coding (codes 90-99), information that may have been previously converted to two-digit coding by one of the numeric-to-alpha instructions. Note that positions 0, 2, 4, 6, and 8 of each word are lost, regardless of what they contained.

Data from the first of each pair of alpha words go to positions 0-4 of the numeric word, and data from the next sequential alpha word go to positions 5-9. As with the numeric-to-alpha codes, each RDW must designate an even number of addresses; the RDW's define the alpha area in alpha-to-numeric conversion, as well as in numeric-to-alpha conversion.

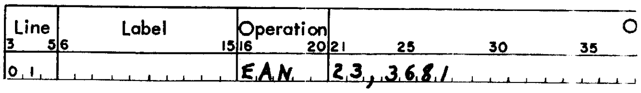
Sign control is automatic with the EAN code. Position 8 of each second alpha word is tested for 7 or non-7.

This instruction can place the resultant numeric data back into the first half of the areas occupied by the original alpha data. These are two reasons why it can do this, when numeric-to-alpha operations cannot:

1. The resultant numeric data take only half as much storage as the original alpha data.
2. The EAN instruction obtains each word from storage only once, whereas the numeric-to-alpha codes obtain each word twice.

Autocoder Example (Figure 105): The actual address 3681 is used. The assembled instruction is:

S01 23 45 6789
−57 00 23 3681



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 ○
0 1		EAN	23, 3681

Figure 105

Table Lookup

The program is able to search tables of information stored in blocks of magnetic-core storage. The search starts at the first word of the table and continues until it finds the word it is looking for, or until it has searched the entire table. Moreover, the table does not have to be completely in one storage block. It can be broken up into a number of blocks, and all blocks are searched in a single operation.

The blocks are defined by record definition words.

A table lookup operation consists of comparing a search value called the *search argument* with the table values, each of which is called a *table argument.* Accumulator 3 contains the search argument for the operation. The comparison is for a table value that is equal to the search argument, equal or higher, or lower, depending on the instruction.

Increment: On all table lookup operations, the search address (working address part of the record definition register) is incremented by the value in the non-indexing portion of index word 98 (non-indexing portion considered plus no matter what sign the index word has). This means that, instead of every word on the table being searched, every second, third, etc., word is searched, depending on the increment value specified. If, for example, the increment value is 0004 and the start address is 0700, words 0700, 0704, 0708, etc., are searched. If the stop address should be 0783, word 0780 is searched, but 0784 is not. The first word in each block is always used, regardless of the increment. There must be an increment; 0000 in the non-indexing portion of 0098 causes an error stop.

Results: When the operation is ended, the location of the found table argument is placed in the indexing portion of index word 98. The remaining positions of index word 98 are unchanged, and its sign is set to plus. If a table location is not found on table lookup operations before the stop address of the last record definition word is exceeded, the next sequential instruction is taken and the indexing portion and sign of index word 98 are not changed. If a table location is found, the next instruction will be taken from the location of the table lookup instruction, +2.

There are three table lookup instructions as shown in Figure 106.

These are the only codes whose Autocoder symbols start with the letter L.

Lookup Lowest
+66 **LL**

Machine Description: The search argument in accumulator 3 is compared with the field-defined portions of the storage words addressed by the RDW, the location of which is designated in positions 6-9 (indexable). Signs are included in the comparisons. The entire accumulator is used; a search argument of less than ten digits must be in the low-order portion of the accumulator. The search continues through sequential or incremented locations, determined by the value in positions 6-9 of index word 98. The table can be in a single block of storage or several blocks, depending on the RDW signs.

The entire table is searched, and the lowest value and its address obtained, provided the table value is lower than that of the original search argument. At completion of the operation, the lowest found value is in accumulator 3, if one was found; or the search argument is in accumulator 3, if a lower value was not found. The location of the lowest found value is in positions 2-5 of index word 98, and the next instruction is taken from the location two greater than that of the LOOKUP LOWEST instruction. Index word 98 is given a plus sign. If at the end of the search a lower value was not found, the next instruction is taken from the next sequential location, and index word 98 is unchanged. Accumulator 2 is used as a working register during this operation.

Instruction Format

S01 +66.

23 Indexing word for modifying the RDW address in positions 6-9.

45 Field definition. Defines the portion of each table word to be compared with the value in accumulator 3.

6789 Address in storage of the first or only record-definition word, which defines the location(s) of the table to be searched.

CATEGORY	OP CODES	NAMES	MNEMONICS
Table look-up	+66	Lookup lowest	LL
	+67	Lookup equal only	LE
	+68	Lookup equal or High	LEH

Figure 106. Table Lookup Operation Codes

Examples: To locate the lowest value in a table defined by the record-definition word in location 2740, searching each table word in positions 5-9 only:

S01 23 45 6789
+66 00 59 2740

Contents of RDW 2740: −00 1100 1110; the lowest value on this table is +xxxxx 00206, in word 1108. Contents of accumulator 3 before the operation: +00000 50000. Contents after the operation: +00000 00206. Contents of index word 98 before the operation: +00 0048 0001. After the operation: +00 1108 0001.

To obtain the lowest value on a table, defined by the record-definition word in 1732, searching each table word entirely:

S01 23 45 6789
+66 00 09 1732

Contents of the RDW in 1732: −00 0900 0948; the lowest value on this table is +00001 23456 in word 0918. Contents of accumulator 3 before the operation: +99999 99999. Contents after the operation: +00001 23456. Contents of index word 98 before the operation: +00 0944 0002. After the operation: +00 0918 0002. Note that the increment value is 2. This means that only the even-numbered words 0900-0948 are searched, 25 words in total. The odd-numbered table words are ignored.

7074 Timing: In microseconds:
16 + 4R + 6T + 4F
R = number of RDW's
T = number of table words
F = number of found conditions

7070 Timing: In microseconds:
36
+ 36 per record-definition word
+ 108 per table word
+ 36 if a lowest value is found
+ 60 for each lower value found

Comments: The table data do not have to be in low-to-high sequence for this operation; the values can be completely random. This instruction is a means of sorting the values in a table and putting them in lowest-to-highest sequence in another section of core storage. The programming procedure is as follows: the beginning search argument is all 9's. After the lowest value in the table is found, it is filled with 9's. (IW 98 can be used to index an address of 0000 for this.) The lookup lowest operation is repeated, again starting with all 9's, this time obtaining the second-lowest value in the original table, etc.

If the signs of the table values are not significant in the operation, this procedure can be used: Make all of the table signs alpha, and the search argument +00000 00000. After the LL operation obtains each table value, change the sign of that value to plus, thus making it higher than the search argument.

If there are two lowest values on a table (equal to each other), the first one found is used.

The comparison between the search argument and the table arguments includes the signs of both factors. The same relation of values is used for these operations as for compare operations:

Highest +99999 99999
to
+00000 00000

−00000 00000
to
−99999 99999

@99999 99999
to
Lowest @00000 00000

Plus is higher than minus, which is in turn higher than alpha. Note that a minus 9 is lower than minus 8, which is lower than minus 7, etc. This means that, if the table values are minus, the highest in absolute value is obtained, regardless of the sign (+ or −) of the original search argument. (A plus search argument always obtains a *lowest* value in this case; a minus search argument obtains one only if it finds one greater in absolute value.) A minus search argument can never obtain a plus table value with this code; an alpha search argument can never obtain a plus or minus table value.

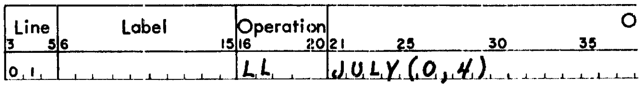
Note that the field-defined portion of each table word is compared with the entire contents of accumulator 3. If the number of significant digits in accumulator 3 exceeds the number of digits specified by field definition, all table values have the same relation to the search argument, as determined by sign (Figure 107).

Search Argument containing more significant digits:	than	Table Argument digits field-defined:
+		All + values are lower
+		All − values are lower
+		All Alpha values are lower
−		All + values are higher
−		All − values are higher
−		All Alpha values are lower
Alpha		All + values are higher
Alpha		All − values are higher
Alpha		All Alpha values are lower

Figure 107. Relative Field Lengths — Lookup Lowest

Autocoder Example (Figure 108): Assume that JULY has been previously defined as location 3023. The assembled instruction is:

S01	*23*	*45*	*6789*
+66	00	04	3023



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		LL	JULY(0,4)

Figure 108

Lookup Equal Only
+67 **LE**

Machine Description: The search argument in accumulator 3 is compared with the field-defined portions of the storage words addressed by the RDW, the location of which is designated in positions 6-9 (indexable). The entire accumulator is used; a search argument of less than ten digits must be in the low-order portion of the accumulator. The search continues through sequential or incremented locations, determined by the value in positions 6-9 of index word 98. The table can be in a single block of storage or several blocks, depending on the RDW sign(s).

The table is searched until a word is found that is equal to the search argument, in sign and value, or until the entire table is searched. If an equal value is found, its location is in positions 2-5 of index word 98 at completion of the operation. The next instruction is taken from the location of the LOOKUP EQUAL instruction, +2. The sign of index word 98 is set to plus. If no equal value is found, index word 98 is unchanged, and the next instruction is taken from the next sequential location.

Instruction Format

S01 +67.

23 Indexing word for modifying the RDW address in positions 6-9.

45 Field definition. Defines the portion of each table word to be compared with the value in accumulator 3.

6789 Address in storage of the first or only record-definition word, which defines the location of the table to be searched.

Example: To search for equal, a table defined by the record-definition word in location 4315, searching each table word in the two high-order positions only:

S01	*23*	*45*	*6789*
+67	00	01	4315

Contents of RDW 4315: −00 2201 2225. Contents of accumulator 3: +00000 00043, unchanged by the operation. Assume that word 2209 contains this value also. Contents of index word 98 before the operation: −00 2221 0001; after the operation: +00 2209 0001. Note that its sign is set to plus.

7074 Timing: In microseconds:
16 + 4R + 6T
R = number of RDW's
T = number of table words

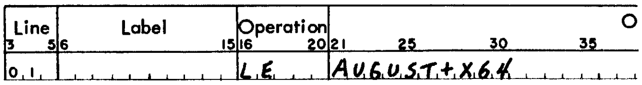
7070 Timing: In microseconds:
36
+36 per record definition word
+108 per table word searched

Comments: Unlike LOOKUP LOWEST, this operation searches a table only until it finds an equal, rather than the entire table every time. The table data does not have to be in any particular sequence; the values can be completely random. If there are two equal values on a table, and both are equal to the search argument, the first value is used (the program does not "know" about the second one).

Signs are included in this operation. If they are unequal, the values are not considered equal.

Autocoder Example (Figure 109): Assume that AUGUST is defined as location 1050. The assembled instruction is:

S01	*23*	*45*	*6789*
+67	64	09	1050



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		LE	AUGUST+X64

Figure 109

Lookup Equal or High
+68 **LEH**

Machine Description: The search argument in accumulator 3 is compared with the field-defined portions of the storage words addressed by the RDW, the location of which is designated in positions 6-9 (indexable). The entire accumulator is used; a search argument of less than ten digits must be in the low-order portion of the accumulator. The search continues through sequential or incremented locations, determined by the value in positions 6-9 of index word 98. The table can be in a single block of storage or several blocks, depending on the RDW sign(s).

The table is searched until a word is found that is equal to, or greater than, the search argument, or until the entire table is searched. If an equal or higher value is found, its location is in positions 2-5 of index word 98 at completion of the operation. The next instruction is taken from the location of the LEH, +2. The sign of index word 98 is set to plus. If no equal or higher value is found, index word 98 is unchanged, and the next instruction is taken from the next sequential location.

Instruction Format

S01 +68.

23 Indexing word for modifying the RDW address in positions 6-9.

45 Field definition. Defines the portion of each table word to be compared with the value in accumulator 3.

6789 Address in storage of the first or only record-definition word, which defines the location of the table to be searched.

Example: To search, for equal or higher, the table defined by the record-definition word in location 731, searching for four high-order positions only:

S01	*23*	*45*	*6789*
+68	00	03	0731

Contents of RDW 731: −00 2800 2899. Contents of accumulator 3: +00000 05250, unchanged by the operation. The table values are in ascending sequence; word 2856 is +5240 743529, and word 2857 is +5260 542123. Contents of index word 98 before the operation: +00 2888 0001. After the operation: +00 2857 0001.

7074 Timing: In microseconds:

16 + 4R + 6T
R = number of RDW's
T = number of table words

7070 Timing: In microseconds:

36
+ 36 per record definition word
+108 per table word searched

Comments: The table values should be in ascending sequence, if this operation is used. The signs of both factors are considered, and the relative values are the same as for compare operations:

Highest	+99999 99999 to +00000 00000
	−00000 00000 to −99999 99999
	@99999 99999 to
Lowest	@00000 00000

If the search argument and the table values are minus, the table values must be stored in *descending* order by absolute value for this code to be used. A plus search argument can never obtain a minus or alpha table value, and a minus search argument can never obtain an alpha table value with this code.

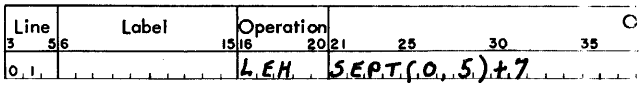
If the number of significant digits in accumulator 3 exceeds the number of digits specified by field definition, either the first table word is obtained by the LEH operation or none of them are, depending on the signs (Figure 110).

Search Argument containing more significant digits:	than	Table Argument digits field-defined:
+		No + values are obtained
+		No − values are obtained
+		No Alpha values are obtained
−		The first + value is obtained
−		The first − value is obtained
−		No Alpha value is obtained
Alpha		The first + value is obtained
Alpha		The first − value is obtained
Alpha		No Alpha value is obtained

Figure 110. Relative Field Lengths — Lookup Equal or High

Autocoder Example (Figure 111): Assume that SEPT has been defined as location 1600. The assembled instruction is:

S01	*23*	*45*	*6789*
+68	00	05	1607



Line 3 5	Label 6 15	Operation 16 20	C 21 25 30 35
0 1		LEH	SEPT(0,5)+7

Figure 111

IBM 729 Magnetic Tape Units

A full-scale 7070-7074 system has four 7604 data transmission channels that connect the 729 Magnetic Tape Units (Figure 112) with core storage. As many as six 729 tape units in any combination of Models II and IV can be connected through each 7604 Model 1 or 2 data transmission channel. One 7604 Model 3 may be used; its two channels can each accommodate any combination of as many as six 729 II, IV, V, and VI tape units. If a channel is equipped with the additional tape attachment optional feature, it can handle as many as ten tape units. Thus, a maximum of forty 729 tape units in a combination of Models II, IV, V, and VI can be used. Another optional feature is the switch control console with 729 tape switching; this permits remote control of 729 tape units.

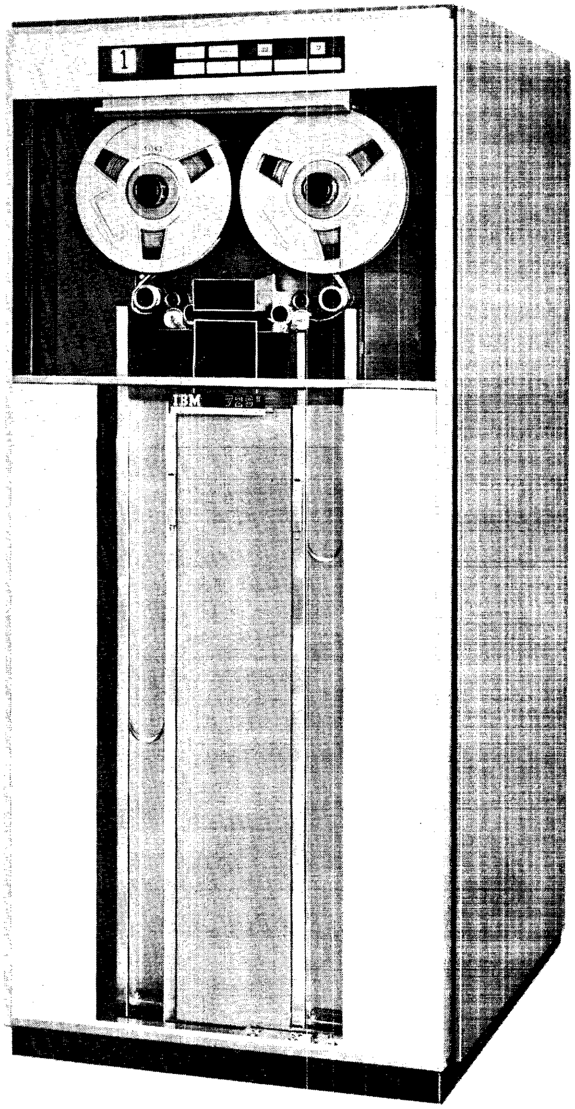


Figure 112. IBM 729 Magnetic Tape Unit

The 729 II and 729 V move tape at a ratė of 75 inches per second; the 729 IV and VI operate at 112.5 inches per second. The 729 II and IV read or write tape at either of two densities, 200 or 556 characters per inch. The 729 V and VI read or write tape at three densities: 200, 556, or 800 characters per inch.

A manual tape densities option switch on the magnetic tape panel of the Customer Engineering console conditions 729 V and/or 729 VI tape units to operate in two of their three densities:

Position 1	200 and 556
Position 2	200 and 800
Position 3	556 and 800

Position 1 permits 729 II, 729 IV, 729 V, and 729 VI tape units to be used on a 7604 Model 3 channel in either high or low density. Position 2 limits 729 II and 729 IV tape units on the controlled channel to low density operations at 200 characters per inch. Position 3 limits 729 II and 729 IV tape units on the controlled channel to low density operations that will actually be at 556 characters per inch.

Attempts to operate 729 II or 729 IV tape units in high density when they are controlled by a tape densities option switch set to positions 2 or 3 will set the channel busy latch on. Subsequent attempts to use the channel will cause the program to halt.

Details of tape unit operation may be found in the *IBM Magnetic Tape Units Reference Manual,* Form A22-6589.

Operating Principles

IBM magnetic tape is a special plastic tape, coated on one side with a layer of magnetic material. Data are recorded on the tape in the form of magnetized spots or bits. These bits are written by the 7070-7074 in the binary coded decimal (BCD) even-parity arrangement. Each tape write operation writes one record. Successive records on tape are separated from each other by ¾ inch of unrecorded tape, called the inter-record gap (IRG).

Information written on tape remains there until the tape is used in a new write operation. This time period may extend indefinitely. When the recorded information is no longer needed, the tape may be used to record new data. Only when a tape is written is the previous information destroyed. The write operation automatically erases old information. Reflective markers on tape provide photo-sensed indication of the beginning (or load point) and the physical end of the useful portion of the tape. The markers are available in convenient roll form in a dispenser.

Tape for 729 tape units is wound on plastic reels 10.5 inches in diameter. A full reel contains about 2,400 feet of usable tape, but lengths as short as 50 feet can be used. The reels are available in red, yellow, blue, and grey. Pressure-sensitive identification labels for reels are available in matching colors.

Tape end retainers are used to prevent reels of tape from unwinding. The pressure-sensitive feature of the tape end retainer permits them to be affixed to the inside of the tape reel container for convenient access and re-use by the operator.

IBM Heavy Duty magnetic tape is recommended for use on IBM 729 II, IV, V, VI Magnetic Tape Units. Heavy Duty tape is available in lengths of 1,200 and 2,400 feet.

BCD Code

The code structure used to record data on 729 tape is 7-bit alphameric, called binary coded decimal (BCD). Seven longitudinal tracks on the tape provide the seven bit positions necessary to record a BCD character. A character is represented by one lateral column of bits across the width of the tape. Figure 113 shows the BCD coding for each alpha, numeric, and special character that is acceptable to the system. There is a correlation between BCD and the IBM punched card code: The AB bits are equivalent to 12-zones; the B bits, 11-zones; the A bits, 0-zones; and combinations of 8421 bits make up the digit values 0-9. (The 8 and 2 bits represent zero.)

Validity Checking

The C track on 729 tape contains a check bit for BCD characters that would otherwise have an odd number of bits. In all reading and writing operations, each character is examined to establish that it has an even number of bits.

Also, a longitudinal check is made of each record on tape. When a tape record is written by a 729 tape unit, the total number of bits in each track (C, B, A, 8, 4, 2, 1) are tested for being odd or even and a bit is written at the end of the record in each track that has an odd number of bits. The check character thus created must itself have an even number of bits to be even (Figure 114). Each tape-read operation tests the check character itself for an even number of bits.

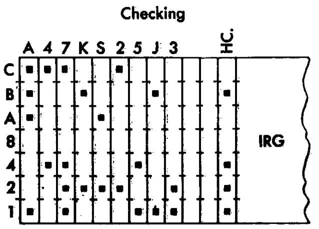


Figure 114. Horizontal Check

Organization of Data on Tape

Data may be arranged on 729 tape in the following categories:

WORD

A word on tape consists of ten or less characters, five characters if it is alpha, 5-10 if it is numeric.

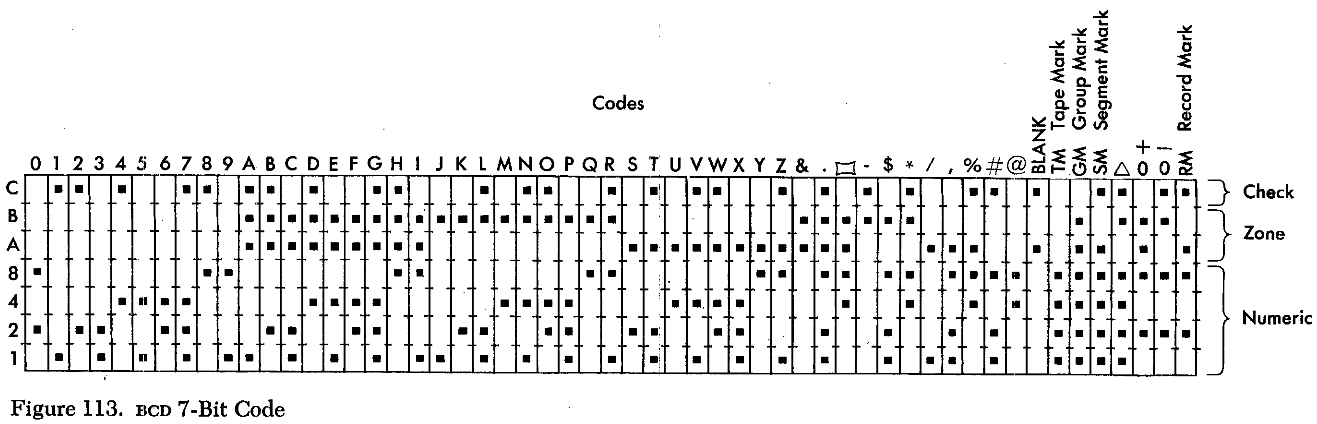


Figure 113. BCD 7-Bit Code

RECORD

A tape record consists of any number of words. Adjacent records on tape are separated from each other by the inter-record gap. Every read or write operation reads or writes one tape record.

SEGMENT

A tape segment usually contains a number of records. Segments are defined as the records contained between segment marks. A segment mark is written only by the tape segment mark write (TSM) instruction. Each segment mark is a one-character record comprised of bits CA8421.

FILE

A file of tape data is all of the records in a group, or category — a customer list for accounts receivable, for example, or an employee list for payroll. A file may be part of a tape reel, or several reels in length.

Further information on tape records and their organization is contained in the *IBM Magnetic Tape Units Reference Manual,* Form A22-6589.

Features of IBM 7070-7074 Tape Operations

A number of features of tape operations in the IBM 7070-7074 add considerably to the efficiency of the system. One feature, the high-speed tape unit, has already been discussed. Other features add further to the effective speed of reading and writing tape records, the efficiency of arranging data read from tape into core storage, and the ease of programming tape operations.

Simultaneous Operations

The 7070-7074 can perform four tape operations simultaneously: reading four tapes, writing four tapes, or any combination of these. Also, the stored program can continue while these operations are taking place. Tape units operating simultaneously must be connected to the system via different channels. The simultaneous-operation feature is made possible by the ability of the four tape channels to work independently of the programming unit and of each other.

An example of the timesaving value of simultaneous operations is a file maintenance routine, which requires the reading of a tape record updating it, and writing it on another tape. Figure 115 shows the overlap of the operations made possible by the simultaneous operations feature. The timing of a sequential operation is shown for comparison.

Numeric and Alphameric Modes

Because all of the words in core storage are numeric or alphameric, sign designations must be recorded when information is written on tape. Each word is written and read in either alphameric or numeric mode. An alphameric word occupies five character positions on tape. A numeric word occupies from five to ten character positions on tape.

The last digit of each numeric word on tape carries sign information: if the sign is plus, bits are written in the A and B track in combination with the low-order

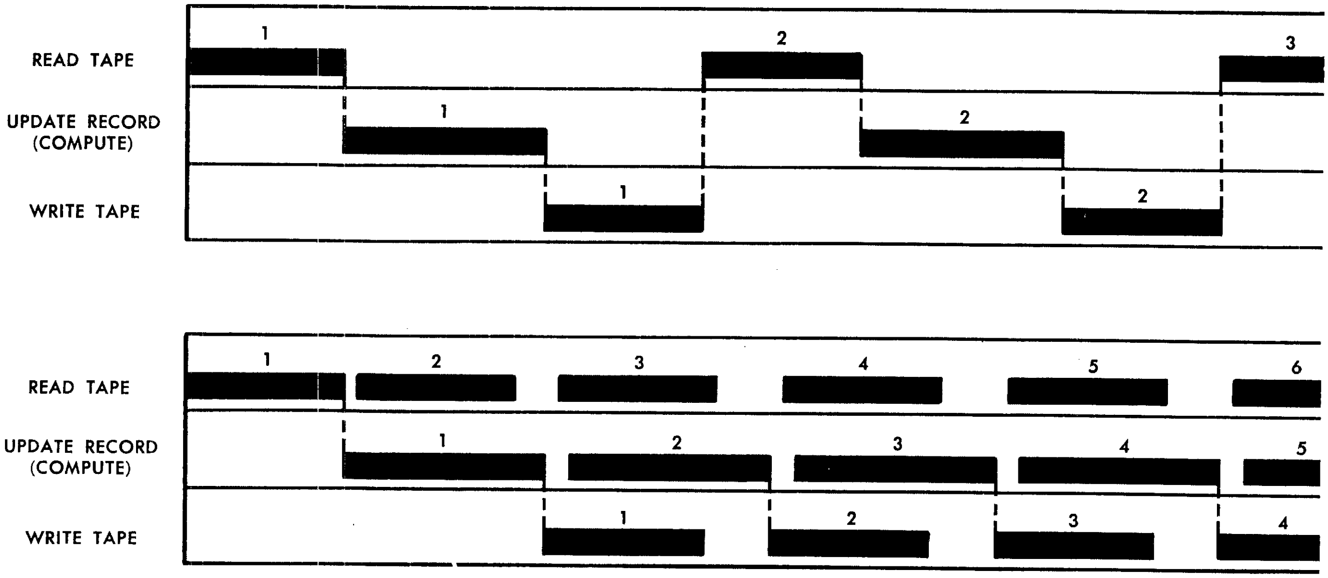


Figure 115. Simultaneous Operations

digit bits; if the sign is minus, a B bit is combined with the low-order digit bit. This sign-over-units arrangement is similar to IBM punched card coding (the 12 zone equivalent to plus and the 11 zone equivalent to minus.

Tape reading in the alphameric mode automatically translates every five characters into a ten-digit word in core storage with an alphameric sign. Numeric words are moved without change, except for the units position character, the zone coding of which is interpreted to give plus or minus designation to the word in core storage. Figure 116 shows examples of numeric and alphameric modes.

Tape reading automatically starts in the alpha mode; the mode is changed to numeric by detection of the mode-change character, sometimes called the *delta* (Δ), on the tape. Detection of this character changes the mode from alpha to numeric or vice versa; the character signifies *change the mode.*

In tape writing, the sign of each core-storage word determines the mode. Every time the mode is changed from alpha to numeric or vice versa, the delta is automatically recorded on tape. (Tape writing automatically starts in the alpha mode.) The delta appears only on tape; it never enters core storage from tape. In BCD code, a delta is CB8421.

Here is an example of a three-word record written on tape. The three words are:

+0123301234
−5678856789
@7461799298

They are written on tape in this sequence:

Δ012330123D567885678RΔMAR28

Because the first word is numeric, the first character on tape is the delta. It denotes a change in mode from alpha to numeric. The units-digit 4 in the first word combines with the plus sign indication to form the letter D. The units-digit 9 in the second word com-

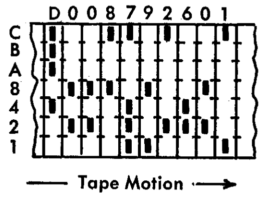
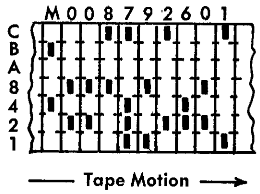
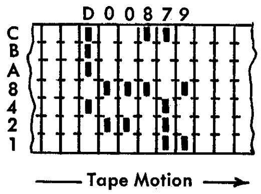
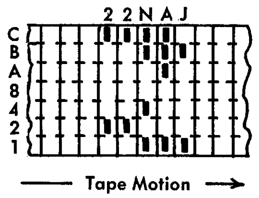
MODE		WORD IN STORAGE	WORD ON TAPE
NUMERIC	NORMAL	+1062978004	 D 0 0 8 7 9 2 6 0 1 C B A 8 4 2 1 Tape Motion →
		− 1062978004	 M 0 0 8 7 9 2 6 0 1 C B A 8 4 2 1 Tape Motion →
	COMPRESSED	+0000978004	 D 0 0 8 7 9 C B A 8 4 2 1 Tape Motion →
ALPHAMERIC		@7161759292	 2 2 N A J C B A 8 4 2 1 Tape Motion →

Figure 116. Numeric and Alphameric Modes

bines with the minus sign indication to form the letter R. The second delta on the tape is caused by the alpha sign of the third word.

Scatter Read-Write

The scatter read-write feature of the system is a powerful programming tool. A single record read from tape can be divided into as many parts as desired by the programmer, and distributed into different blocks of core storage. This is done in only one program step, the same instruction that initiates the tape-read operation. The blocks of storage are defined by record-definition words.

This feature can be used in writing tape as well as reading. The program can gather data from various core-storage blocks and automatically assemble it into one tape record. Figure 117 shows an inventory record separated by field and category as it is read from tape, or conversely, gathered from blocks of core storage and written as a single tape record.

A series of records in storage may be written on the tape as a single record. This is called a *grouped record.* The scatter read-write feature can be used to read and write grouped records, while each individual record has its own core-storage section.

The scatter read-write feature can also be used for bringing data to and from disk storage, the unit-record input-output synchronizers, the inquiry control synchronizers, to the console typewriter, and for transmission of data from core storage to different parts of core storage.

Record-Mark Control

Record-mark words give additional flexibility to the scatter read-write feature. In many cases, the data read from tape into a core-storage block may be variable in

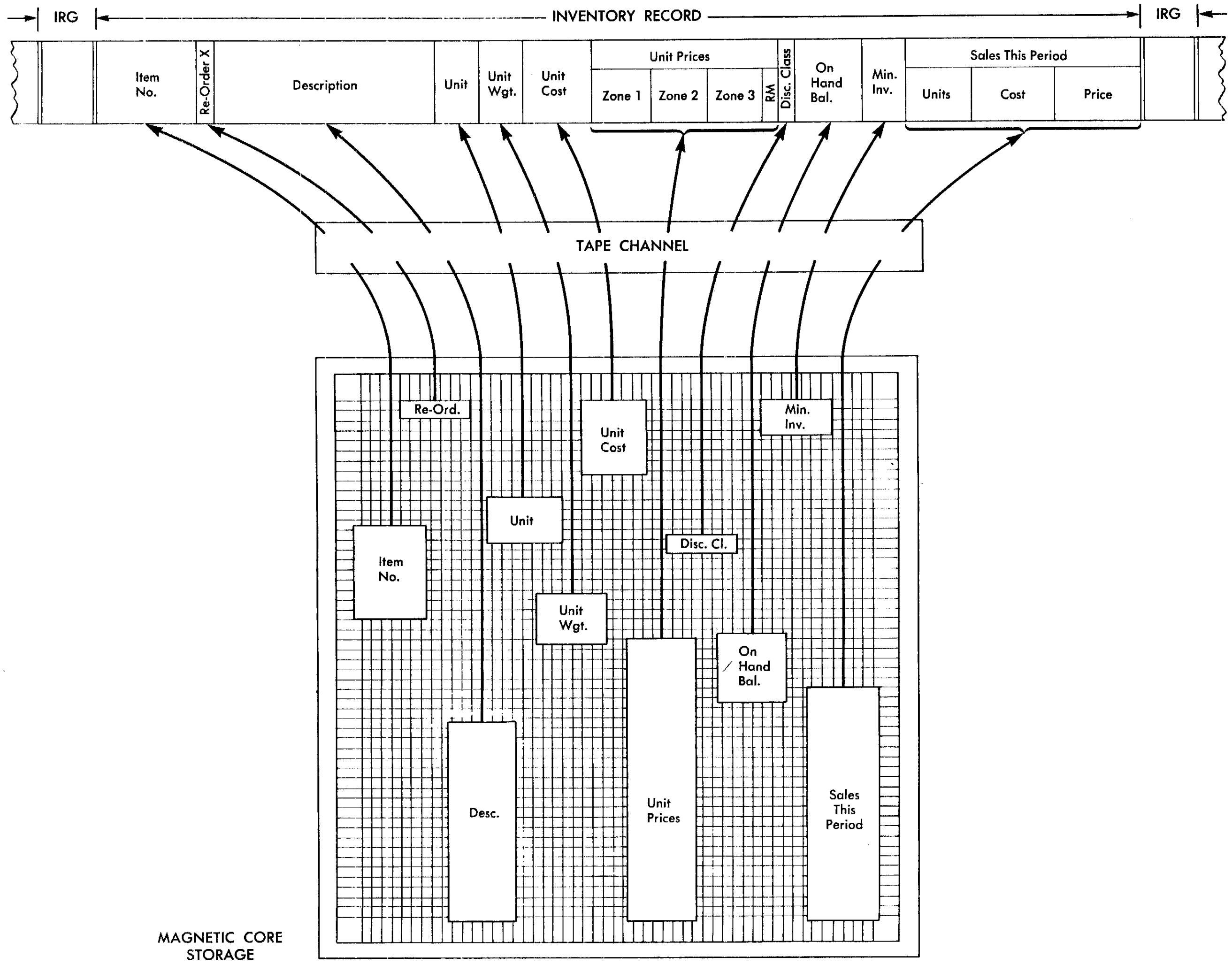


Figure 117. Scatter Read-Write

length; a name and address field, for example. The block of storage reserved for a variable-length field must, of course, be large enough to accommodate the largest size that the field on tape can be – in this instance, the longest name and address. But to make all of the name and address fields in the tape the same size as the largest (by adding blanks to fill up to the allotted size) would be impractical, and would waste tape capacity and processing time. Record marks make this unnecessary.

A record-mark word on tape is an alpha word consisting of a special character (‡, CA82) in the low-order position, and any other four characters. Detection of this word automatically denotes the end of a block in a scatter read-write operation. In scatter read, a tape record being read into a block of storage does not continue to fill that block after a record-mark word has been read. Instead, the next block receives the tape characters that immediately follow the record-mark word. Detection of a record mark has the same effect as the working and stop addresses in the record-definition register being equal. (See "Block Transmission.")

In scatter tape writing, a record-mark word in core storage terminates the movement of data from the block and initiates transmission from the next block. The tape read or write instruction determines whether record-mark words will be operative. If not, a record-mark word is treated as a normal alphameric word.

Zero Elimination

Numeric core storage words with high-order zeros may be written on tape with up to five of the high-order zeros eliminated. When these shortened numeric words are read from tape, the missing high-order zeros are automatically provided in the core storage location receiving the word. Zero elimination instructions do not alter words when reading or writing in the alphameric mode.

Use of Core Storage Locations 9990 to 9999

These locations are separate from register and counter addresses and may be used as a permanent ten-word storage area that is not affected by computer reset, zero storage routines, or dropping power. The date or console data to be written on tape header and trailer labels may be stored in these locations.

Storage locations 9990 to 9999 are accessible only to and from tape or disk via 7604 channels or 7907 data channels; they cannot be addressed from the console, by instructions, or by unit record RDW's.

The information to be stored in locations 9990 to 9999 may be written on tape and then read from tape into storage in the manner suggested in the following program example:

```
0000   −8100100002   TRW
0001   −8100130020   TW
0002   +5100110002   BCB
0003   −8100100002   TRW
0004   −8100110021   TR
0005   +5100110005   BCB
```

0020 contains the RDW that identifies the location in storage of the information to be transferred to core storage locations 9990 to 9999 by first writing and then reading tape.

0021 contains −0099909999.

Tape Operation Codes

These codes govern all of the operations of the 729 tape units: reading, writing, positioning, rewinding, etc. In all cases, the instruction initiates the operation, and the stored program can continue while the tape unit is operating. When the tape unit completes the operation, the main program can be signalled for priority.

Machine Description: All of the operations that have anything to do with the 729 tape units are incorporated in four augmented codes ±81, ±82, ±83 and ±84. With each code, the sign determines whether there will be a normal-condition priority signal when the tape operation is completed; plus means that there will, minus means that there will not. (Unusual-condition priority occurs regardless of the sign. See "Priority Processing.") The only difference between the operation codes is the 7604 channel involved: 81 is used for tape units connected to the system via channel control 1; 82 is used for channel control 2; 83 for channel control 3; and 84 for channel control 4. The operation itself is defined in position 5 of the instruction.

Instruction Format

S Plus means that there will be a priority signal at the end of the tape operation, if it is a read, write or tape-spacing operation. Minus means that there will not be a priority signal, unless there is an unusual condition. (See "Priority Processing" for full descriptions of these conditions.) For those tape instructions that do not involve reading, writing, or spacing, the sign of the instruction has no effect.

0 Always 8, identifying the instruction as an operation involving magnetic tape.

1 Always 1, 2, 3, or 4 identifying the 7604 tape channel. This position and position 4 determine which of the 40 tape units is involved.

23 Indexing word. Positions 6-9 can *in all cases* be modified by positions 2-5 of the indexing word specified here.

4 Specifies the 729 tape unit on the 7604 channel designated in position 1. For a standard channel, the six (maximum) tape units are numbered 0 through 5. For a channel equipped with the additional tape attachment special feature, the ten (maximum) tape units are numbered 0 through 9. The use of tape unit numbers 6 through 9 in connection with a *standard* channel will cause the channel busy latch to be set on and remain on, and no tape operation will occur.

5 Digits 1-9 in this position specify the tape operation to be performed. Digit 0 indicates that the operation is defined by the digit in position 9 of the instruction. The digit in position 5 defines the 729 operation as follows:

1	Tape Read	(P)TR
2	Tape Read per Record Mark Control	(P)TRR
3	Tape Write	(P)TW
4	Tape Write per Record Mark Control	(P)TWR
5	Tape Write with Zero Elimination	(P)TWZ
6	Tape Write per Record Mark Control and with Zero Elimination Combined	(P)TWC
7	Tape Segment Forward Space per Count	(P)TSF
8	Tape Segment Backspace per Count	(P)TSB
9	Tape Read All Alpha	(P)TRA
0	Look at position 9 to further define the operation.	

If the letter P is used in the mnemonic, the Autocoder assembly program gives the instruction a plus sign; if it is not used, the instruction gets a minus sign.

6789 The function of these positions depends on the digit in position 5.

If position 5 contains a digit 1-6, positions 6-9 contain the address of the first record-definition word, denoting the first group of core-storage locations to be read into or written from.

If there is a 7 or 8 in position 5, positions 6-9 contain the address of a storage word that defines the number of segments to be spaced over. The difference between the values of positions 2-5 and positions 6-9 of the specified word denotes the number of segments to be spaced over. (Normally, positions 2-5 are 0000, and positions 6-9 represent the number of segments.)

If there is a 0 in position 5, positions 6-9 do not represent an address; instead, position 9 contains the operation code, and positions 6-8 are not used. These operations do not require record-definition words. The digit in position 9 defines the 729 operation as follows:

0	Tape Select	TSEL
1	Tape Mark Write	(P)TM
2	Tape Rewind	TRW
3	Tape Rewind Unload	TRU
4	Tape Backspace	TRB
5	Tape Write Segment Mark	(P)TSM
6	Tape Skip	TSK
7	Tape End of File Off	TEF
8	Tape Set Low Density	TSLD
9	Tape Set High Density	TSHD

The digits in 6-8 can be anything; they are ignored if there is a 0 in position 5.

Execution times for 729 tape instructions are as follows:

7074 Timing: In microseconds:

TR	19
TRR	19
TW	19
TWR	19
TWZ	19
TWC	19
TSF	19
TSB	19 (not at load point) 28 (at load point)
TSEL	28
TM	19
TWR	28
TRU	28
TRB	19 (not at load point) 28 (at load point)
TSM	19
TSK	28
TEF	28
TSLD	28
TSHD	28

7070 Timing: 130 microseconds for each of these operations.

FOLLOWING are detailed descriptions of the tape instructions denoted by a digit 1-9 in position 5 of the instruction. All of these codes utilize positions 6-9 as the address of a core-storage word (indexable as per positions 2-3). All of these instructions can be used with priority (plus sign in the operation code).

Tape Read

1. (P)TR

This code specifies that the 729 tape unit addressed by the digits in positions 1 and 4 of the instruction is to read a tape record. The record-definition word addressed by positions 6-9 (indexable) defines the core-storage area to be read into.

If the first character in the tape record is not a delta, it is translated into the two-digit alphameric code, and placed in the two high-order positions of the first word in the RDW-defined area. Each succeeding tape character is similarly translated, with each group of five filling a storage word in high-order to low-order sequence. The storage-word addresses are serially advanced under control of the record-definition register.

When the first delta on the tape is detected (it could be the first character in the tape record), the operation changes to the numeric mode. Each digit on the tape is translated to a single digit in the core-storage word. Each time a delta is detected, the mode changes.

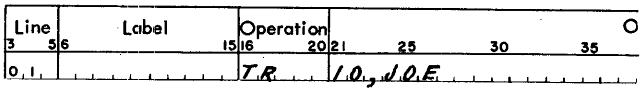
In the numeric mode, an alpha character on the tape automatically denotes the units position of the core-storage word to be read into. The bit coding determines the sign of the word: the A and B bits combined denote plus, the B bit alone denotes minus. The number of numeric characters to go into a storage word cannot be less than five nor, of course, more than ten (not less than four digits and the sign-over-units alpha character, nor more than nine digits and the alpha character). In the case of less than ten digits, the high-order positions of the storage word are filled with zeros.

Example: To read a record from 729 tape unit 3 on 7604 channel 2 into the core-storage area for which word 1715 is the first RDW, and to initiate a normal-priority signal at completion:

S01	*23*	*4*	*5*	*6789*
+82	00	3	1	1715

Autocoder Example (Figure 118): Assume that JOE has been defined as word 2310. The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
−81	00	0	1	2310



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		TR	10,JOE

Figure 118

Tape Read per Record Mark Control

2. (P)TRR

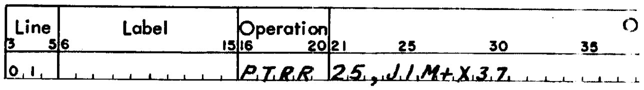
This code is the same as tape read, with this addition: The detection of a record mark (CA82) in the low-order position of an alpha word on tape causes a new RDW to be read into the record-definition register, if the present one is plus; or terminates the movement of data from tape, if the present RDW is minus. It has the same effect as equal working and stop addresses in the record-definition register.

Example: To read, per record-mark control, a record from 729 tape unit 4 on 7604 channel 3 to the storage area for which word 1234 is the first RDW, with no normal-priority signal at completion:

S01	*23*	*4*	*5*	*6789*
−83	00	4	2	1234

Autocoder Example (Figure 119): JIM is word 2341. The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+82	37	5	2	2341



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		PTRR	25,JIM+X37

Figure 119

Tape Write

3. (P)TW

The 729 tape unit addressed by the digits in positions 1 and 4 is instructed to write a tape record. The record-definition word addressed by positions 6-9 (indexable) defines the core-storage area to be read from. If the sign of the first core-storage word is alpha, its contents are translated from two-digit alpha coding to the 7-bit BCD coding, and the five characters are written on tape, in high-order to low-order sequence.

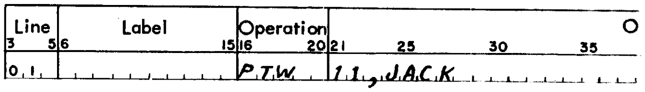
If the sign of the first core-storage word is plus or minus, a delta is written as the first character in the tape record. The sign of each succeeding word is tested; each time it is alpha when the preceding one was numeric (+ or −), or vice-versa, the delta is automatically recorded on tape. The storage-word addresses are serially advanced under control of the record-definition register.

Example: To write a record on 729 tape unit 0 on 7604 channel 1 from the core-storage area for which 3321 is the first RDW, and to signal for normal-condition priority at the completion of the operation:

S01	*23*	*4*	*5*	*6789*
+81	00	0	3	3321

Autocoder Example (Figure 120): JACK's address is 4341. The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+81	00	1	3	4341



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		PTW	11,JACK

Figure 120

Tape Write per Record Mark Control

4. (P)TWR

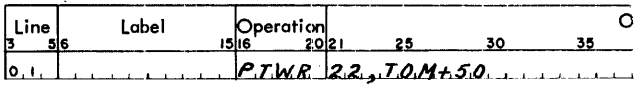
This code is the same as TAPE WRITE, with this addition: The detection of a record mark (80) in the two low-order positions of an alpha storage word causes a new RDW to be read into the record-definition register, if the present one is plus, or terminates the tape-writing operation, if the present RDW is minus. It has the same effect as equal working and stop addresses in the record-definition register.

Example: To write, per record-mark control, a record on 729 tape unit 5 on 7604 channel 2, from the core-storage area for which 3635 is the first RDW, with no normal-condition priority signal:

S01	*23*	*4*	*5*	*6789*
−82	00	5	4	3635

Autocoder Example (Figure 121): TOM is word 1517. The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+82	00	2	4	1567



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 ○
01		PTWR	22,TOM+50

Figure 121

Tape Write with Zero Elimination

5. (P)TWZ

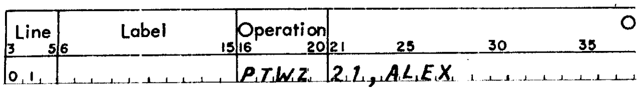
This code is the same as TAPE WRITE, with this addition: High-order insignificant zeros in numeric words (+ or − sign), up to a maximum of five per word, are not written on the tape. NOTE: Tape *reading* automatically fills in high-order zeros for numeric words on tape; tape *writing* eliminates high-order zeros only if the instruction calls for it.

Example: To write, with zero elimination, a record on 729 tape unit 4 on 7604 channel 1, from the core-storage area defined by the RDW in: 1400 + positions 2-5 of IW 31. (No normal-condition priority signal is to be given.)

S01	*23*	*4*	*5*	*6789*
−81	31	4	5	1400

Autocoder Example (Figure 122): ALEX is 0440. The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+82	00	1	5	0440



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 ○
01		PTWZ	21,ALEX

Figure 122

Tape Write with Zero Elimination and per Record Mark Control Combined

6. (P)TWC

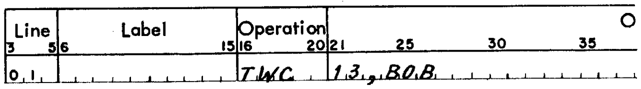
This code is the same as tape write, with the addition of both the record-mark control feature described above under code 4, and the zero elimination described above under code 5. Thus, maximum efficiency of tape-writing time and output-tape capacity is achieved by this code.

Example: To write, with zero elimination, per record-mark control, a record on 729 tape unit 1 on 7604 channel 4, from the core-storage area for which 1311 is the first RDW, and to signal for normal-condition priority at completion:

S01	*23*	*4*	*5*	*6789*
+84	00	1	6	1311

Autocoder Example (Figure 123): BOB is word 1066. The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
−81	00	3	6	1066



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 ○
01		TWC	13,BOB

Figure 123

Timing of Tape Read and Write Operations

Because tape reading and writing involve physical motion of the tape and mechanical motion of tape units, variations in timing may be large in comparison with internal processing speeds. Therefore, it is recommended that the programmer use priority tape instructions together with the branch-channel-busy instruction, rather than calculated times, to synchronize tape input-output operations with the stored program.

The timing variations in tape-write operations are related only to the duration of the recording pulses; the tape passing speed affects only the distance between characters. Timing variations in tape-read operations, however, are related to tape passing speed. The speed is a factor of the unit reading the tape, as well as the unit upon which the tape was written.

	DENSITY	CHARACTER RATE	MILLISECONDS PER CHARACTER
729 II	High	41,667 per second	.024
	Low	15,000 per second	.067
729 IV	High	62,500 per second	.016
	Low	22,500 per second	.044
729 V	Highest	60,000 per second	.017
729 VI	Highest	90,000 per second	.011

An important factor in deriving operation times for tape operations is the time it takes for the tape unit to get to full speed at the start of the operation, and to complete the movement of tape after the operation is completed. In milliseconds, these times are:

729 II and V	10.8
729 IV and VI	7.3

The length of time it takes to read or write on tape is obtained by adding this figure to the duration of reading or writing a tape record:

	DURATION	
729 II and V	10.8 + CN	milliseconds
729 IV and VI	7.3 + CN	milliseconds

C = Milliseconds per character
N = Number of characters

Regardless of the 729 model used, the tape read or write instruction itself takes about 108 microseconds to be executed. This is the length of time taken before the next stored-program instruction can be executed.

Some customer programmers have counted the number of program instruction cycles that are executed during a particular 729 tape operation used in their program. This information has been used to set up a loop of instructions to time out the operation before beginning interrogation of tape status words. When such a timing loop in a program created for a 7070 is executed on the faster 7074, it will time out too soon to permit the desired information in the tape status word to be available for interrogation. This difficulty can be eliminated by modifying the 7070 timing loop for use on the 7074.

Tape Segment Forward Space per Count

7. **(P)TSF**

The 729 tape unit denoted by positions 1 and 4 of the instruction is instructed to move the tape forward until the specified number of tape segments has been spaced over. The number of segments to be spaced over is indicated by the difference in the start and stop addresses of the record-definition word addressed by positions 6-9 (indexable).

As the tape moves forward, the first character of each record is checked for validity. An invalid character causes condition code 1 to be placed in the tape final status word. (See "Priority Processing.") Because the invalid character may be a segment mark, a programmed check should be made to verify that the tape is at the desired location.

The working and stop address portions of the record-definition register, although they do not represent core-storage addresses to be read into or written from, work in much the same way. As each segment mark (CA8421) is spaced over, the *working address* is compared with the *stop address*. If the addresses are equal, the operation stops and the tape is in position to read the first record in the next segment. If the addresses are not equal, the *working address* is increased by one and the tape spaces forward to the next segment mark.

Only one record-definition word is used in this operation, regardless of whether its sign is plus or minus.

Example: To move tape on 729 tape unit 5 on 7604 channel 1 forward the number of segments specified by the difference plus one of the values in positions 2-5 and 6-9 of word 1645 (no normal-condition priority signal is to be given):

S01	*23*	*4*	*5*	*6789*
−81	00	5	7	1645

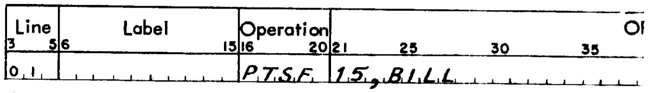
Tape Unit Timing: The time it takes to space over several tape records is dependent on the number of records spaced over and the size of each record:

C = Milliseconds per character
N = Number of characters
R = Number of multi-character records
S = Number of single-character records

	MILLISECONDS
729 II and V	CN + 10.8 (R+S)
729 IV and VI	CN + 7.3 (R+S)

Autocoder Example (Figure 124): BILL is 1730. The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+81	00	5	7	1730



Line 3 5	Label 6 15	Operation 16 20	Ol 21 25 30 35
0 1		PTSF	15, BILL

Figure 124

Tape Segment Backspace per Count

8. **(P)TSB**

This code works in the same manner as TAPE SEGMENT FORWARD SPACE PER COUNT, with this difference:

The tape is moved backward, instead of forward, until the specified number of segment marks has been sensed. As the tape moves backward, the first character in each record of less than five characters in length is checked for validity. An invalid character causes condition code 1 to be placed in the tape final status word. (See "Priority Processing".) Because the invalid character may be a segment mark, a programmed check should be made to verify that the tape is at the desired location. This instruction can be used with an output tape, one that is being written, whereas TAPE SEGMENT FORWARD SPACE should be used only with input tapes.

Example: To move tape on 729 tape unit 2 on 7604 channel 1 backward the number of segments specified by the difference plus one of the values in positions 2-5 and 6-9 of word 1414, and to signal for normal-condition priority at completion:

S01	*23*	*4*	*5*	*6789*
+81	00	2	8	1414

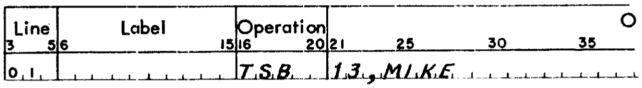
Tape Unit Timing: Tape characters cannot be read when the tape is being moved backward. A single-character record may be a segment mark, or not; thus, when a single-character record is detected during a (P) TSB operation, the tape must be stopped and moved forward, to read the character. The following formulas apply:

C = Milliseconds per character
N = Number of characters
R = Number of multi-character records
S = Number of single-character records

	MILLISECONDS AFTER READ
729 II and V	CN + 10.3 (R−1) + 68.6S
729 IV and VI	CN + 6.8 (R−1) + 47.5S
	AFTER WRITE,
729 II and V	Add 7.5 milliseconds to above
729 IV and VI	Add 5.0 milliseconds to above

Autocoder Example (Figure 125): MIKE is 3841. The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
−81	00	3	8	3841



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		TSB	13,MIKE

Figure 125

Tape Read All Alpha

9. (P)TRA

This code is the same as tape read, except:

All 729 tape record characters, whether numeric or alphameric, are read in the alpha mode and become two characters in storage. For invalid characters, the A and B bits from tape are placed in positions 6 and 3, respectively, of the first storage character. (Positions 2, 1, and 0 are not used.) The 8, 4, 2, 1 and C bits from tape are placed in positions 6, 3, 2, 1 and 0, respectively, of the second storage character. A delta character in the record does not affect the mode of reading; it is read into storage as two no-bit characters.

TAPE READ ALL ALPHA is used to read a record into storage after a tape word error has been detected. (See "729 Tape Errors" section of this manual). In using the TRA instruction, the programmer must remember to provide twice as much storage space for numeric tape words as is normally required.

If storage space is limited, half the record may be read with the tape read all alpha (TRA) instruction and displayed on the console typewriter for analysis. Then the last half of the record may be read into storage by using two RDW's with identical addresses, the first with a plus sign and the second with a minus sign. Additional space must be provided, however, for the mode change characters (deltas) that enter storage as two blank digit positions in the TRA mode.

FOLLOWING are detailed descriptions of the 729 tape instructions denoted by a digit 0 in position 5 and digits 0-9 in position 9 of the tape-control instruction. Positions 6-8 can be of any value; they are ignored. In all cases, however, positions 6-9 are indexable, just as if they represented an address. These operations do not require record-definition words.

With the exception of TAPE MARK WRITE and TAPE SEGMENT MARK WRITE, these operations cannot be used to signal priority. Thus, the sign of the instruction can be either plus or minus with no effect on the operation.

Tape No-Op Select

xxx0 TSEL

This code is the means of testing for the availability of a specific 729 tape unit. When this instruction is executed, the 7604 channel becomes busy if the specified tape unit is not in ready status. The actual test is performed by a BRANCH CHANNEL BUSY operation (BCB +51), which normally follows it.

The purpose of this instruction is to check whether the operator has loaded a tape reel on the unit and made it ready to be read or written.

Example: To set up 729 tape unit 4 on 7604 channel 1, so that a BCB test will determine whether it is available to the stored program:

S01	*23*	*4*	*5*	*6789*
±81	00	4	0	xxx0

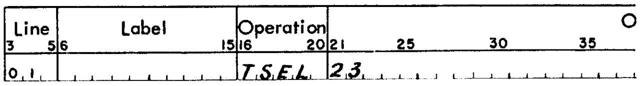
Sign can be either plus or minus. Positions 6-8 can contain any digits.

7074 Timing: 28 microseconds.

7070 Timing: 130 microseconds.

Autocoder Example (Figure 126): The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+82	00	3	0	0000



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		TSEL	23

Figure 126

Tape Mark Write

xxx1 (P)TM

This code writes a tape mark (8421) on the 729 tape unit addressed by the digits in positions 1 and 4. The tape mark is a single-character record, preceded and followed by an IRG. It is not in core storage; this instruction automatically produces the tape mark.

This instruction can be used to initiate a normal-priority signal (plus sign in the operation code).

Example: To write a tape mark on 729 tape unit 4 on 7604 channel 2, with no signal for normal-condition priority:

S01	*23*	*4*	*5*	*6789*
−82	00	4	0	xxx1

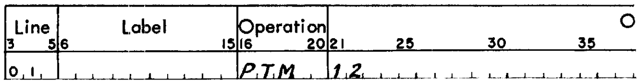
Positions 6-8 can contain any digits.

Tape Unit Timing: The equivalent of writing a single-character record:

729 II or V, High Density	10.8 + .024 milliseconds
729 II or V, Low Density	10.8 + .067 milliseconds
729 IV, or VI, High Density	7.3 + .016 milliseconds
729 IV or VI, Low Density	7.3 + .044 milliseconds

Autocoder Example (Figure 127): The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+81	00	2	0	0001



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		PTM	1 2

Figure 127

Tape Rewind
xxx2 **TRW**

The 729 tape unit designated by the digits in positions 1 and 4 is instructed to rewind its tape. The unit is effectively disconnected from the system until the rewind operation is completed, at which time it again becomes available to the stored program. The ready light on the unit turns off while the tape is rewinding, and turns on again when the unit is again ready for use.

Example: To rewind 729 tape unit 0 on 7604 channel 4 and make it again available to the stored program after the rewind operation is completed:

S01	*23*	*4*	*5*	*6789*
±84	00	0	0	xxx2

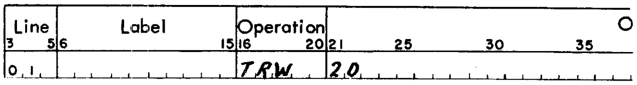
Sign can be either plus or minus. Positions 6-8 can contain any digits.

Tape Unit Timing: An entire 2400-foot reel of tape takes about 1.2 minutes to rewind on a 729 II or V, or .9 minute on a 729 IV or VI.

The 7604 tape channel is released and made available to the stored program after 35 milliseconds.

Autocoder Example (Figure 128): The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+84	00	0	0	0002



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		TRW	2 0

Figure 128

Tape Rewind Unload
xxx3 **TRU**

This code is the same as TAPE REWIND except that tape is unloaded from the vacuum columns, the upper section of the head assembly raises, and the tape unit is *not* available to the system after the rewind operation is completed. It cannot be used again by the stored program until the operator makes it available.

Example: To rewind 729 tape unit 5 on 7604 channel 1 and make it unavailable to the stored program at the completion of the rewind operation:

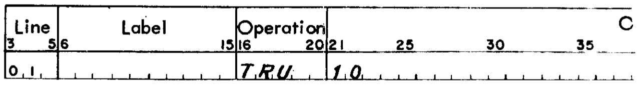
S01	*23*	*4*	*5*	*6789*
±81	00	5	0	xxx3

The sign can be either plus or minus. Positions 6-8 can contain any digits.

Tape Unit Timing: Same as TAPE REWIND.

Autocoder Example (Figure 129): The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+81	00	0	0	0003



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		TRU	1 0

Figure 129

Tape Record Backspace
xxx4 **TRB**

The specified 729 tape unit backspaces the tape one record. (Tape records are defined as separated by inter-record gaps.) If the tape backspaces over a segment mark or tape mark, this is considered as one record. If the tape is at load point at the time this command is given, no tape motion takes place; the instruction is the equivalent of NOP.

Example: To backspace 729 tape unit 2 on 7604 channel 2 one record:

S01	*23*	*4*	*5*	*6789*
±82	00	2	0	xxx4

The sign can be either plus or minus. Positions 6-8 can contain any digits.

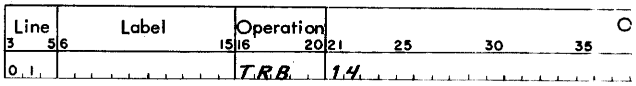
Timing:

	MILLISECONDS, AFTER A READ OPERATION
729 II and V	46.8 + CN
729 IV and VI	31.2 + CN
	AFTER A WRITE OPERATION
729 II and V	Add 7.5 milliseconds to above
729 IV and VI	Add 5.0 milliseconds to above

C = Milliseconds per character
N = Number of characters in the tape record

Autocoder Example (Figure 130): The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+81	00	4	0	0004



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		TRB	1 4

Figure 130

Tape Segment Mark Write

xxx5 (P)TSM

This instruction causes a tape segment mark (CA8421) to be written as a one-character record on the tape. The segment mark is automatically provided for this operation; it does not need to be in storage.

This instruction can be used to initiate a normal-priority signal (plus sign in the operation code).

Example: To write a segment mark on 729 tape unit 3 on 7604 channel 2 and to signal for normal-condition priority:

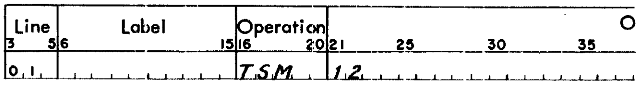
S01	*23*	*4*	*5*	*6789*
+82	00	3	0	xxx5

Positions 6-8 can contain any digits.

Tape Unit Timing: Same as TM.

Autocoder Example (Figure 131): The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
−81	00	2	0	0005



Line	Label	Operation	O
3 5	6 15	16 20	21 25 30 35
0 1		TSM	1,2

Figure 131

Tape Skip

xxx6 TSK

The specified 7604 channel is conditioned so that the 729 tape unit will space forward about four inches before writing in the next write instruction. The tape passed in the write operation is erased. The unit on which the spacing takes place is the first unit, on a channel, which is instructed to write following the skip instruction for any unit on that channel.

This command is used after rewriting a record several times has failed to correct an error detected by the two-gap head, usually because of a defect on the surface of the tape. It is usually followed immediately by a write operation. If the next instruction for that channel is a read command, no skipping takes place, and the skip instruction loses its effect on any subsequent write command. Any other type of operation that intervenes before a write command does not affect the function of skip on the write command.

Example: To condition a tape unit on channel 3 to space forward about 4 inches prior to writing, on the next write instruction:

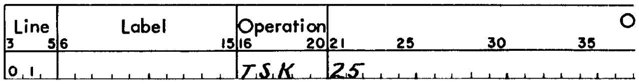
S01	*23*	*4*	*5*	*6789*
±83	00	Y	0	xxx6

Position four can designate any tape unit, on that channel, that is available to the stored program. Normally, the unit is the same one designated by the subsequent write instruction; it is the write instruction that spaces the specific tape unit. The sign can be either plus or minus. Positions 6-8 can contain any digits.

Tape Unit Timing: The time required for the tape unit to skip forward about four inches occurs during the next tape write operation; it is 40.5 milliseconds for a 729 II or V and 27 milliseconds for a 729 IV or VI.

Autocoder Example (Figure 132): The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+82	00	5	0	0006



Line	Label	Operation	O
3 5	6 15	16 20	21 25 30 35
0 1		TSK	25

Figure 132

Tape End of File Off

xxx7 TEF

Any time that a 729 tape unit senses a foil strip while writing, or a tape mark while reading, the tape indicator light on that unit turns on, and an EOF condition code results. This instruction turns it off. Any time that the EOF indicator is on, a read or write operation involving that tape unit causes an EOF priority condition. (See "Priority Processing.")

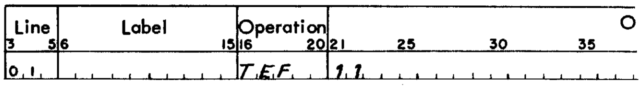
Example: To turn off the EOF indicator for 729 tape unit 5 on 7604 channel 1:

S01	*23*	*4*	*5*	*6789*
±81	00	5	0	xxx7

The sign can be plus or minus. Positions 6-8 can contain any digits.

Autocoder Example (Figure 133): The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+81	00	1	0	0007



Line	Label	Operation	O
3 5	6 15	16 20	21 25 30 35
0 1		TEF	1,1

Figure 133

Tape Set Low Density

xxx8 TSLD

The 729 II or IV tape unit specified by positions 1 and 4 is set to operate at a density of 200 characters per inch. If the unit is a 729 II, its passing speed will be 15,000 characters per second on subsequent read or write operations; if the unit is a 729 IV, its speed will be 22,500 characters per second.

The 729 V and VI tape units read or write at *three* densities: 200, 556, and 800 characters per inch. A manual switch on the magnetic tape panel of the customer engineering console provides for selection of operation in any two of the densities: 200 and 800, or 200 and 556, or 556 and 800. The TSLD instruction conditions the 729 V or VI tape unit to operate at the low density specified by the setting of the manual density selection switch. If the tape is to be read, its characters must have the same density. If not, an error (priority condition code 1) is indicated.

Example: To set tape unit 3 (a 729 II or IV) on 7604 channel 2 to a density of 200 characters per inch:

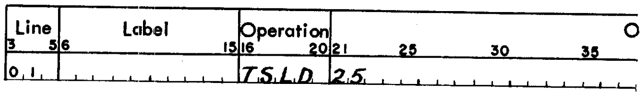
```
 S01  23  4  5  6789
±82   00  3  0  xxx8
```

The sign can be either plus or minus. Positions 6-8 can contain any digits.

Timing: 729 II or V 19.2 milliseconds
729 IV or VI 12.8 milliseconds

Autocoder Example (Figure 134): The assembled instruction is:

```
 S01  23  4  5  6789
+82   00  5  0  0008
```



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		TSLD	25

Figure 134

Tape Set High Density
xxx9 **TSHD**

The 729 II or IV tape unit specified by positions 1 and 4 is set to operate at a density of 556 characters per inch. If the unit is a 729 II, its character rate will be 41,667 characters per second on subsequent read or write operations; if the unit is a 729 IV, its rate will be 62,500 characters per second.

The 729 V and VI tape units read or write at *three* densities: 200, 556, or 800 characters per inch. A manual switch on the magnetic tape panel of the customer engineering console provides for selection of operation in any two of the densities: 200 and 800, or 200 and 556, or 556 and 800. The TSHD instruction conditions the 729 V or VI tape unit to operate at the high density specified by the setting of the manual density selection switch. If the tape is to be read, its characters must have this same density. If not, an error (priority condition code 1) is indicated.

Example: To set tape unit 5 (a 729 II or IV) on channel 1 to a density of 556 characters per inch:

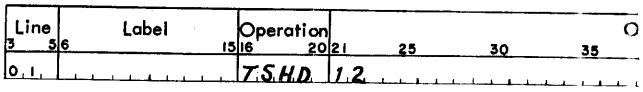
```
 S01  23  4  5  6789
±81   00  5  0  xxx9
```

The sign can be either plus or minus. Positions 6-8 can contain any digits.

Timing: 729 II or V 19.2 milliseconds
729 IV or VI 12.8 milliseconds

Autocoder Example (Figure 135): The assembled instruction is:

```
 S01  23  4  5  6789
+81   00  2  0  0009
```



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		TSHD	12

Figure 135

IBM 729 Tape Errors

This section provides information for programmers and other 7070-7074 systems personnel concerned with the analysis of 729 tape error conditions.

Tape Final Status Word Condition Codes 0 and 1

One or more of several types of errors are indicated when condition code 0 or 1 appears in position one of a tape final status word. These error conditions take precedence over any other priority conditions that may exist at the same time. Condition 0 has precedence over condition 1.

CONDITION 0 — TAPE WORD ERROR (TWE)

This condition occurs during tape reading if the record has:

1. Less than five digits or more than ten digits before the sign character is detected (numeric mode).
2. A mode change character (delta) within a word.

When the TWE condition occurs, some of the tape information may not reach storage and that which does reach storage may be mislocated or invalid. Accurate analysis of the error condition requires re-reading the record with the tape read all alpha (TRA) instruction. This instruction causes all characters on tape to be read in the alpha mode; each tape character occupies two positions in core storage. In using the TRA instruction, the programmer must remember to provide twice as much storage space for numeric tape words as is normally required.

If storage space is limited, half the record may be read with the tape read all alpha (TRA) instruction and displayed on the console typewriter for analysis. The last half of the record may then be read into storage by using two RDW's with identical addresses, the first with a plus sign and the second with a minus sign. Additional space must be provided, however, for the mode change characters (deltas) that enter storage as two blank digit positions in the TRA mode.

Error records read in TRA mode may be displayed on the 7150 Console typewriter for analysis. The operator can cause the contents of a core-storage word to type out by keying the address of the word on the 7150 Console. Numeric words type the sign of the word followed by the ten digits. Alphameric words type an "A" for sign, followed by the five alphameric characters.

After an alphameric word has been typed, its ten-digit core-storage notation may be typed by pressing the display key on the 7150 Console. The word is preceded by the core-storage sign-bit value "3" for alphameric sign indication.

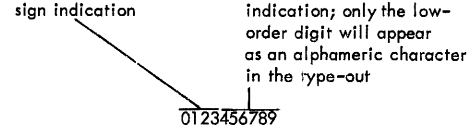
The unit-record type (TYP) instruction initiates a block transmission of words from core storage to the typewriter. Numeric words type ten digits with sign. Alphameric words type five alphameric characters without sign.

ANALYSIS OF TYPE-OUT OF TRA WORDS

Because of the zero-elimination feature, valid numeric tape words can have as few as five digits or as many as ten digits. Each numeric word is defined on tape by having the sign of the word appear with the units position digit. Plus sign indication is made for a BCD digit on tape by writing a B bit and an A bit along with the bit configuration for the digit; minus is indicated by adding only a B bit. Such digits with sign indication, when read with TRA, appear in a type-out as alpha characters. The sign value and numeric value of such characters may be determined by use of the coding system shown in Figure 3.

For example, the digit 7 is represented in BCD code with the C421 bit combination. Adding the A and B bits for plus sign indication produces the CBA421 character representing the letter G. Adding only the B bit for minus sign indication forms the B421 character for the letter P. (C bits are added only to BCD characters that would otherwise have an odd number of bits.) $\overset{+}{0}$ and $\bar{0}$ digits type out as + and #, respectively.

Sign indication appearing with the first, second, third, or fourth digit of a numeric word causes a TWE. The tape read all alpha (TRA) instruction makes no provision for zero insertion for numeric words less than ten digits in length. Therefore, the varying length of numeric tape words must be kept in mind when analyzing the type-out of a numeric word read with TRA:


Invalid position for sign indication
Valid position for sign indication; only the low-order digit will appear as an alphameric character in the type-out
0123456789

For example, the number +11111 appears on tape as $1111\overset{+}{1}$ in the following BCD bit representation:

```
BIT POSITIONS
      C          CCCCC
      B              B
      A              A
      8
      4
      2
      1          11111
```

When read in TRA mode, a type-out of the word stored would look like this:

1111A

The A is the alphameric translation of the zoned tape character that carries the plus sign for the word. The type-out shows the word to be valid, and the next word in the record may be examined:

111A1

The sign indication character A is in an invalid position in the word. The BCD bit configuration for the word that caused this TWE would look like this:

```
CCCCC
   B
   A
11111
```

This arrangement of BCD bits on tape (representing $111111111\bar{1}$) is valid:

```
CCCCCCCCC
         B
1111111111
```

When read in TRA mode, a type-out of the stored words would appear as follows:

111111111J

The following BCD bits on tape (representing 1111111111) are not a valid word and would cause a TWE when read in numeric mode:

```
CCCCCCCCCC
1111111111
```

A type-out of the stored words would appear as follows:

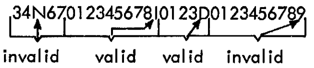
1111111111

The absence of sign indication establishes the invalidity of this word.

As a further example, if this series of zoned and unzoned digits were written on tape:

$$34\bar{5}6701234567 8\overset{+}{9}0123\overset{+}{4}0123456789$$

and were read by the 7070-7074 in TRA mode, a type-out of the stored words would appear, translated from BCD code, as follows:



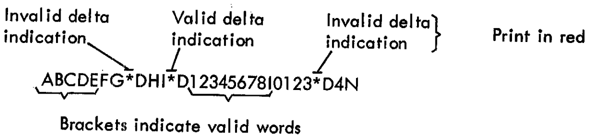
The arrow associated with each bracket points out the character that indicates validity or invalidity for the bracketed numeric word.

INVALID DELTAS

A mode change character (delta) anywhere in a record except between words is invalid and causes a TWE. When read in TRA mode, a delta enters storage as a blank character (two blank digit positions) and appears on a type-out in red as *D. Invalid deltas are identified by their position within a word. For example, the tape characters:

```
                +    -
ABCDEFG△HI△1234567890123△45
```

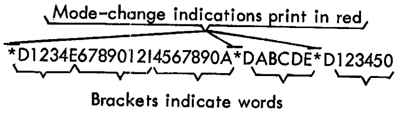
when read with TRA, would type out:



The varying length of numeric tape words must be considered when analyzing the type-out of TRA words. For example, this series of tape characters:

```
  +     -     +            -
△1234567890123456789O1△ABCDE△123456
```

are shown to be valid words in the type-out:


Mode-change indications print in red
*D1234E678901214567890A*DABCDE*D123450
Brackets indicate words

CONDITION 1 ERROR

Condition 1 indicates one or more of the following errors:

Read
- a. Vertical check on tape character.
- b. Longitudinal check on a tape record.
- c. Two-of-five validity check on translated data.
- d. A tape character not translated to two-of-five coding.
- e. Tape unit density setting mismatch.

Write
- f. Insufficient signal strength.
- g. Address failure.
- h. Translator failure.
- i. Invalid data in initial or final status words.
- j. Record definition word error.
- k. Timing error.
- l. Loss of data--overload.
- m. Validity error detected by two-gap head.

Condition 1 errors due to item e are caused by attempts to read tape recorded in a density that does not match the density setting of the tape unit used.

If condition 1 is caused by a tape data error, the channel busy latch releases and efforts to correct the error are possible. If condition 1 is caused by machine errors listed as items g, h, i, j, or k, however, the flow of data between tape and storage stops and the channel busy latch does not release. This condition of the channel busy latch prevents execution of subsequent corrective tape operations in the program; if efforts to resume operation do not turn off the channel busy latch, customer engineering attention is indicated.

On a system equipped with Engineering Change 249841, a timing error (item k) will not halt the flow of data between tape and core storage, nor will the channel busy latch remain set. This permits operations to proceed after a non-recurrent timing error.

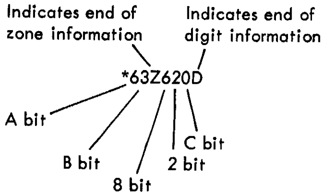
Condition 1 tape data errors that resist correction by error routines may yield to procedures based upon the following characteristics of the tape read all alpha (TRA) instruction.

When read in TRA mode, a tape character with an odd number of bits (invalid), enters core storage as a translation of the bits present. The A and B bits enter even digit positions in core storage; the 8-4-2-1 bits and the C bit enter odd digit positions in core storage. The bits translate in this manner:

BCD BITS	TRANSLATION IN EVEN STORAGE POSITION	TRANSLATION IN ODD STORAGE POSITION
A bit	6	
B bit	3	
8 bit		6
4 bit		3
2 bit		2
1 bit		1
C bit		0

The type-out of this invalid core storage information is in the format specified in the "Type" portion of the "Unit Record" section of this manual; an asterisk precedes the translation of the zone bits, the letter Z indicates the end of the zone bits and is followed by the translation of the other bits, and a letter D indicates the end of the digit bits. All of the characters—* through D—print out in red.

For example, the BCD bits CBA82 (invalid) would print in red:


Indicates end of zone information
Indicates end of digit information
*63Z620D
A bit
B bit
8 bit
2 bit
C bit

As another example, assume that the tape word ABCDE has an extra 4 bit in the C character. When read in TRA mode, the word would type as follows:

$$\text{AB}\underbrace{\text{*63Z321D}}_{\text{red}}\text{DE}$$

The absence of both an A bit and a B bit in the odd-parity tape character will cause a letter D to type out in their place. For example, the translation of BCD bits 8-4-2 (invalid) would type out in red:

*DZ632D

In many cases, analysis of the type-out of the translation of odd-parity tape characters in relationship to the magnetic tape BCD code shown in Figure 3 will provide information sufficient for error corrections.

If a condition code 1 error consists of a single-bit dropout or pickup in one character, an immediate correction may be made by comparison of the LRC register and TAU trap register, both located on the magnetic tape panel of the customer engineering console. The track in error is indicated by a light in the LRC register. (Each of the seven bit positions in a BCD character is written in one of seven "tracks" on tape.) The character in error is displayed in seven-bit BCD code in the TAU trap register. If the sense of the bit in the error track is reversed, the true character may be recognized and manually corrected in storage through console procedures.

VALIDITY CHECKING ON SEGMENT MARK COUNT OPERATIONS

During a tape segment forward space per count (TSF) operation, the first character of each record passed during the search is checked for validity. During the tape segment backward space per count (TSB) operation, the first character in each record of less than five characters in length is checked for validity. An invalid character causes condition code 1 to be placed in the tape final status word. Because the invalid character may have been a segment mark, a programmed check should be made to determine if the tape is at the desired location.

Inter-Record Gap Noise

It is recommended that the tape rewind unload (TRU) instruction be used whenever it is desirable to dismount a reel of tape for inspection, for a later restart, or other reasons. This procedure will eliminate inter-record gap noise records caused by manually unloading a tape unit that is in write status. This recommendation is based on the following considerations:

A tape unit in write status that receives a TRU instruction will automatically erase tape before rewinding. This insures a noise-free inter-record gap (IRG) past the last record written.

When a tape unit that is in write status is manually unloaded, the tape unit does not automatically erase forward prior to rewinding. An attempt to reload the reel of tape to the position where writing was discontinued will involve reading forward to the last good record. When writing is resumed, the IRG preceding the new record may contain some of the old information left on the tape from previous usage of the reel. The reason for this is that tape moves farther past the write head after reading a record than it does after writing a record. This means that writing will be resumed past the point on tape where the end of the last IRG was written. The difference in stopping points is approximately 1/10 to 1/8 inch.

Use of the TRU instruction before dismounting a reel of tape will preclude possible IRG noise records due to manually unloading a reel of tape from a tape unit in write status.

IBM 729 V and VI Magnetic Tape Units

The new triple-density IBM 729 V (75 inches per second) and IBM 729 VI (112.5 inches per second) Magnetic Tape Units can be used on 7070-7074 systems. One IBM 7604 Model 3 Tape Control with two channels is used with these units. Each IBM 7604 Model 3 channel has a manual tape densities option switch on the magnetic tape panel of the customer engineering console. This switch conditions 729 V and/or 729 VI tape units to operate in two of the three densities:

Position 1 — 556 and 800 characters per inch.
Position 2 — 200 and 800 characters per inch.
Position 3 — 200 and 556 characters per inch.

Position 1 limits 729 II and 729 IV tape units on the controlled channel to low density operations that will actually be at 556 characters per inch. Position 2 limits 729 II and 729 IV tape units on the controlled channel to low density operations at 200 characters per inch. Position 3 permits 729 II, 729 IV, 729 V, and 729 VI tape units to be used on a 7604 Model 3 channel in either high or low density.

Attempts to operate 729 II or 729 IV tape units in high density when they are controlled by a tape densities option switch set to position 1 or 2 will set the channel busy latch on. Subsequent attempts to use the channel will cause the program to halt.

Tape Compatibility

An incomplete understanding of the requirements for tape compatibility between the 7070-7074 and 1401-1410 Data Processing Systems can cause unnecessary tape error conditions. Proper procedures enhance efficiency of inter-system tape operations.

OUTPUT

The following characteristics of the 1401-1410 should be noted when writing tape that will eventually be handled by that system.

1. The 1401 and 1410 are variable word length systems.
2. The 1401-1410 stores each character in 7-bit, odd-parity BCD coding.
3. All characters acceptable to the 1401-1410 can be created by the 7070-7074 in alphameric code.
4. The mode change character (delta) will enter 1401-1410 storage as a character and must be handled there by programming.
5. Numeric words – the signed units position results in an alphameric BCD character in 1401-1410 usage.
6. Zero elimination – tape written with zero elimination cannot be expanded automatically to fixed field or word lengths in the 1401 unless the advanced programming feature and the compressed tape feature are installed.
7. The 1401 reads into storage whatever is written on tape. There is no provision for RDW control of blocked records. The advanced programming feature for the 1401 permits deblocking with record mark control. This ability is standard on the 1410.

Tape records written in conformance with the above considerations can eliminate unnecessary programming inconvenience for the 1401-1410 systems.

INPUT

The standard 1401-1410 systems can write tape in a format that the 7070-7074 systems will accept readily. The mode change character (delta) can be programmed by the 1401-1410 to reduce programming and storage requirements in the 7070-7074.

Tape characters such as record mark, segment mark, mode change, and tape mark can be entered with punched cards into the 1401-1410 and written on tape.

Double sign in a word, although valid in the 1401-1410, is invalid for the 7070-7074 systems.

Unit Record

Unit Record Operation Codes

All of the operations that involve unit record equipment (7500 Card Reader, 7501 Console Card Reader, 7550 Card Punch, 7400 Printer), as well as the type operation for the console typewriter, are incorporated in the +69 unit record control operation. Positions 4 and 5 of the instruction further define the operation. Movement of data to and from core storage is under control of record definition words.

Instruction Format

S01 These three positions of the instruction are always +69.

23 Indexing word. Positions 6-9 of the instruction can in all cases be modified by the value in positions 2-5 of the specified index word.

4 The digit in this position either specifies the synchronizer with which the unit record device is connected (1, 2, or 3) or serves to augment position 5 of the instruction.

5 The digit in this position specifies the operation to be performed, as follows:

0 – Unit Record Signal (US). Position 4 of the instruction must specify the synchronizer used (1, 2, or 3).

1 – Read (UR). If the 7500 Card Reader is to be used, position 4 of the instruction must specify the synchronizer used (1, 2, or 3). If the 7501 Console Card Reader is to be used, position 4 must contain the digit 4.

2 – Write (UW) *or* Punch (UP). Position 4 of the instruction must specify the synchronizer used (1, 2, or 3).

3 – Write Invalid (UWIV) *or* Punch Invalid (UPIV). Position 4 of the instruction must specify the synchronizer used (1, 2, or 3).

4 – Type (TYP). Position 4 of the instruction must contain the digit 0.

6789 Address of the initial record definition word.

Timing

Transfers to and from the input-output synchronizers interrupt the execution of the computer program for a period of one to six milliseconds. The duration of this interruption (interlock time) is the same for all three units, but the compute time available to the program depends upon the specific unit being used.

In the table below, cycle time is the actual time it takes the corresponding unit to complete one operation, and compute time is the cycle time less the interlock time.

	MAXIMUM	MINIMUM
7500 Card Reader		
Cycle time — 120 ms		
Interlock time	6 ms	1 ms
Compute time	119 ms	114 ms
7550 Punch		
Cycle time — 240 ms		
Interlock time	6 ms	1 ms
Compute time	239 ms	234 ms
7400 Printer		
Cycle time — 400 ms		
Interlock time	6 ms	1 ms
Compute time	399 ms	394 ms

Following are detailed descriptions of the unit record instructions.

Unit Record Read (7500 Card Reader)

1 **UR**

This instruction initiates a block transmission of from one to 16 words from the designated input synchronizer to storage locations specified by the process channel control system. Data to be entered are arranged by control-panel wiring. The address portion of the instruction specifies the location of the first record definition word (RDW).

Word 1 of the synchronizer is transmitted to the storage location specified by the first RDW start address. If the final RDW stop address is encountered before 16 words are transmitted, the last word transmitted is at that stop address location. If the start address of the RDW is greater than the stop address, the operation is terminated immediately. In no case are more than 16 words transmitted. There is no transmission of data if an end-of-file (EOF) condition is recognized.

After the transmission, an immediate restart signal is created to allow the program to continue. A card read cycle is then performed to refill the synchronizer unless end of file has been sensed, or an error has been detected and RVI is wired to AUTO STOP. After execution of the unit record read instruction, the instruction counter contains:

L+1, if an error was detected
L+2, if the end-of-file condition exists
L+3, if no error was detected and the end-of-file condition does not exist

where L is the location in storage of the unit record read instruction.

Example: Read eight words from card reader synchronizer 1 to location 0600-0607 (Figure 136).

	LOCATION	CONTENTS
Assembled instruction	1000	+6900112000
RDW	2000	–0006000607
Next instruction is:		
1. if error	1001	Begin error routine
2. if EOF	1002	Begin EOF routine
3. if correct	1003	Continue processing

Line	Label	Operation	O
3 5	6 15	16 20	21 25 30 35
0 1	INSTR	UR	1,RECORD

Figure 136

Unit Record Read (7501 Console Card Reader)

1 **UR**

This instruction initiates a record transmission of from one to eight words from IBM card input to core storage. Word 1 of the card goes to the storage location specified by the start address in the RDW (record definition word). If the stop address in the RDW is encountered before eight words are transmitted, the last word transmitted is at the stop address location.

If, at the beginning of the read operation, the start address of the first RDW is greater than the stop address, the 7501 passes the card to the prestack position, transmits only the first word of information from the card to the start address location, and halts with the program check light (7150 operating panel) on. Any succeeding RDW's having start addresses greater than stop addresses cause the same procedure (the storing of only the first word of the record specified in that RDW). The same procedure occurs when an RDW has an alpha sign.

After the unit record read instruction has been executed, the instruction counter contains:

L+1, if an error was detected
L+2, if the end-of-file condition exists
L+3, if no error was detected and the end-of-file condition does not exist

where L is the location in storage of the unit record read instruction.

Example: Read eight words from the 7501 to locations 0600 through 0607 in core storage (Figure 137).

Line	Label	Operation	OPERAN
3 5	6 15	16 20	21 25 30 35 40
0 1	INSTR	UR	4,RDWCARD
0 2		B	ERRORROUT
0 3		B	EOFROUTINE
0 4		ZA1	A RESUME

Figure 137

	LOCATION	CONTENTS
Assembled instruction	1000	+6900412000
Record definition word	2000	–0006000607
Next instruction is:		
1. if error	1001	Begin error routine
2. if end of file	1002	Begin EOF routine
3. if correct	1003	Continue processing

Unit Record Signal

0 **US**

This instruction does not cause any data transmission on the synchronizer specified. The only function executed is to turn on the program switch exit associated with the addressed input synchronizer. This control switch causes an early-timed impulse (PSE) to be emitted on the control panel of the associated input unit the next time synchronizer is called for. If an end-of-file signal is given, no control panel impulse is available. This impulse is used to pick up selectors, impulse the offset stacker, etc. The instruction counter is always advanced by one.

Example: Set up a signal for the next read cycle on card reader 2 (Figure 138).

	LOCATION	CONTENTS
Assembled instruction:	1000	+6900200000
Next instruction is:	1001	Continue processing

Line	Label	Operation	O
3 5	6 15	16 20	21 25 30 35
0 1	INSTR	US	2

Figure 138

Unit Record Punch, Write

2 **UP, UW**

NOTE: Either punch or write is accepted by Autocoder for any card output unit. Both are assembled as a 2 in digit five.

This instruction initiates a block transmission of up to 16 words from the storage area specified by the RDW addresses. If the RDW is set for less than 16 words, the remaining synchronizer words are filled with zeros and assigned alpha signs. If the transmission of data from storage to synchronizer is valid, a punch-write cycle is then performed to empty the synchronizer while processing continues. Output format is determined by control-panel wiring.

After execution of the UP or UW instruction, the instruction counter contains:

L+1, if the transmission of information from storage to synchronizer is invalid, and no printing or punching takes place

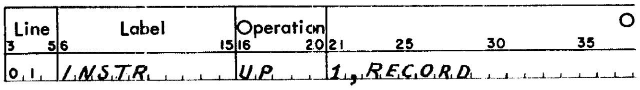
L+2, if the channel 9 latch is on

L+3, if the transmission of information is valid

where L is the location in storage of the UP or UW instruction.

Example: Punch the eight words contained in locations 0600-0607 on punch 1 (Figure 139).

	LOCATION	CONTENTS
Assembled instruction	1000	+6900122000
Next instruction is:		
1. if error	1001	Begin error routine
2. if channel 9	1002	Begin channel 9 routine
3. if correct	1003	Continue processing



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1	INSTR	UP	1, RECORD

Figure 139

Printer Channel 9 Test

The printer test for a punch in channel 9 of the carriage control tape is a function of the +69 unit record control instruction. This test is made at the time a print instruction is executed.

If a hole in channel 9 is sensed on a print cycle, the channel 9 latch is set on at the beginning of the next print cycle. If a write invalid instruction is interpreted, the test of the channel 9 latch is not executed.

If the channel 9 latch is on, the next printer write instruction is interpreted as a NOP instruction, and the program advances to L +2. If the channel 9 latch is off, then the write instruction is executed and the program advances to L +1 if invalid data were transmitted, or L +3 if valid data were transmitted.

The channel 9 latch is reset at the beginning of the next printer operation even if the instruction is write invalid.

The channel 9 test signals the program when a specific line on a report has been printed and is used for printing page totals, page numbering, overflow sheet identifications, etc.

The channel 9 punch must be punched in the carriage control tape one print line ahead of the line to be detected. The channel 9 punch will not be sensed if it passes during the following carriage operations:

1. Short skip
2. Long skip
3. Last line of a skip
4. First line after a skip

Unit Record Punch Invalid, Write Invalid
3 **UPIV, UWIV**

NOTE: Either punch or write is acceptable for any card output unit. Both are assembled as a 3 in digit five.

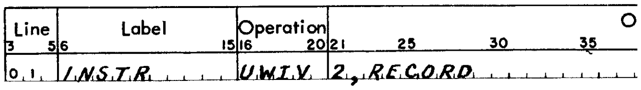
This instruction has the function of the normal punch-write instruction except that validity and ring checks are suppressed. It is a method to punch-write records containing invalid characters.

When the instruction is executed the corresponding validity and ring checks are suppressed for one cycle only and a punch-write cycle is initiated. The normal storage-to-synchronizer transmission is performed under control of the processing channel control system. The check circuits, in their suppressed condition, cause an early-timed error punch-write impulse (EWI) to be emitted on the control panel of the output unit. After execution of the UPIV or UWIV instruction, the instruction counter contains L+1, where L is the location in storage of the UPIV or UWIV instruction. The channel 9 test does not occur with these instructions.

Output format is still under control of panel wiring.

Example: Print the eight words contained in locations 0600-0607 on printer 2 (Figure 140).

	LOCATION	CONTENTS
Assembled instruction	1000	+6900232000
Next instruction	1001	Continue processing



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1	INSTR	UWIV	2, RECORD

Figure 140

Type
4 **TYP**

This instruction initiates a block transmission of words (specified by one or more RDW's) from core storage to the typewriter by way of the synchronizer register. The typing operation is preceded by a carriage return, and continues until the start and stop address in the record definition register are equal.

All numeric words type with sign and ten digits. All alphameric words type with no sign and five alphameric characters. Invalid characters type in bit representation (see below) and even though they represent errors, do not stop execution of the stored program or halt the typing operation. A carriage return occurs upon completion of typing.

After the type instruction has been executed, the IC (instruction counter) contains:

L+1, if an error was detected

L+2, if no error was detected

where L is the location in storage of the type instruction.

If invalid characters are present during the type operation, the IC may contain either L+1 or L+2 at completion of the operation, depending upon where in the system the invalid transmission occurs. If the invalid transmission occurs between core storage and the synchronizer register, the IC will contain L+1; if between the synchronizer register and the console typewriter, the IC will contain L+2. In either case, the invalid characters type in bit representation and the stored program continues. If a non-validity error is detected during the type operation, the IC will contain L+1 and the stored program will halt.

Bit typing is the typing of the bit representations (01236) of the invalid digit. It is preceded by an asterisk and followed by the letter D, all automatically typed in red. Assume, for example, that +0123456789 in storage is to be typed, but the digit 6, which should be made up of the 0 and 6 bits, has an extraneous 3 bit. The typed word appears as follows:

+012345*630D789
red

If there are no bits, only the * and the D are typed.

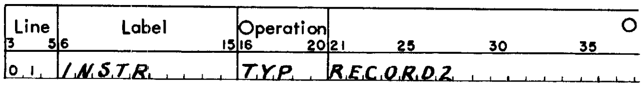
The purpose of the D is to define it as a digit. In an alpha word, positions 0, 2, 4, 6, and 8 are identified by Z for zones, and positions 1, 3, 5, 7, and 9 are identified by a D for digits. If either digit of an alpha character is invalid, both the zone and digit are bit-typed. Assume an alpha word containing ABCDE, @6162636465 in core storage, in which the 3 digit of the C contains an extraneous 2 bit added to its 0 and 3 bits. The word types as follows:

AB*60Z320DDE
red

The bits of both the 6 and the 3 of the 63 are typed, even though only the 3 was invalid.

Example: Type out the five words in locations 0810-0814 (Figure 141).

	LOCATION	CONTENTS
Assembled instruction	1000	+6900042051
Next instruction:		
1. if error	1001	Begin error routine
2. if correct	1002	Continue processing



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
01	INSTR	TYP	RECORD2

Figure 141

IBM 7500 Card Reader

One to three IBM 7500 Card Readers (Figure 142), each operating at a maximum speed of 500 cards per minute, can be attached to an IBM 7070 or 7074 Data Processing System. Thus, maximum input speed can be up to 1500 cards per minute (120,000 characters per minute).

Control-panel signals can be initiated by the stored program to control format of subsequent cards or offset stacking.

An end-of-file key allows automatic transfer to programmed end-of-file procedures.

Reading with internally wired format control is accomplished by use of the load hub on the control panel.

SPOOL (Simultaneous Peripheral Operation On Line) routines can be performed by the card reader when priority signals for automatic program control are assigned by console setting.

Each character of input data is checked for validity. If an invalid character is detected, transmission of the input data from the synchronizer to storage is automatically repeated. If all the characters prove to be correct on this second try, the program continues normally; otherwise, an error indicator is set and the program branches.

Further information and control panel wiring diagrams are contained in the *IBM Reference Manual, IBM Unit Record Input-Output Equipment for IBM 700-7000 Series Data Processing Systems,* Form A22-6660.

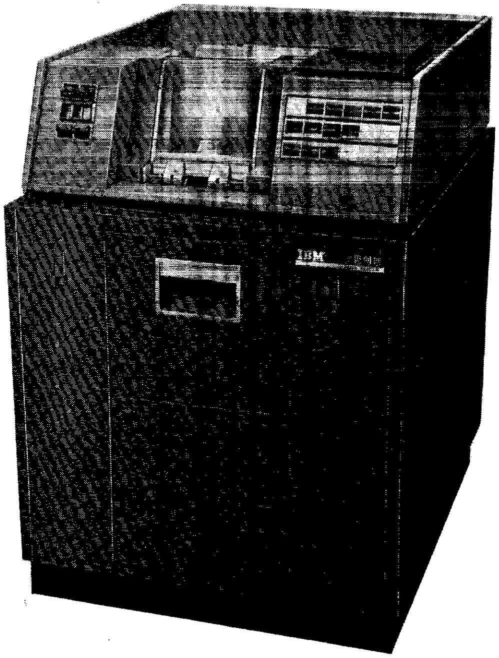


Figure 142. IBM 7500 Card Reader

IBM 7501 Console Card Reader

The IBM 7501 Console Card Reader (Figure 143) permits low-cost direct card input to tape-oriented systems on an exception basis. The over-all reading process (including card feeding) takes place at 60 cards per minute. The 7501 has no control panel. It reads column-by-column, sensing the punched holes photoelectrically.

Card Reading

Pressing the feed-reset key on the 7501 puts the first card in preregister position. A read instruction from the computer causes the 7501 to carry on the following cycle:

1. Move the card from preregister position, scanning from column 1 through column 80.
2. Release the card.
3. Feed another card into preregister position.

Input Card Format

Input cards may contain as many as eight words in 7070-7074 standard word format of ten digits plus sign. The sign is represented as a zone punch over the numeric punch in the units (rightmost) column of each word. The zone punches are: a 12 punch for a plus sign, an 11 punch for a minus sign, and both 12 and 11 punches for an alpha sign. (NOTE: The 7500 Card Reader can not read a 12-11 alpha punch.)

The recommended programming and punching procedures are:

1. Preferably, punch all 80 columns of a card and program for eight words to be read. Since it is possible that these cards may be read by a 7500 Card Reader or a 1402 Card Read Punch, using all 80 columns insures compatibility.
2. If fewer than eight words is the maximum number that will be used throughout an operation, program for a record definition word to indicate the correct number of word positions needed, and then punch that number of columns multiplied by ten (the number of columns per word). See Figure 144.
3. If an input record happens to contain fewer words than specified by the record definition word, substitute 0 punches (with necessary zone punches) for the missing words (Figure 145).
4. Punch zone coding in the tenth punched column of every 7070-7074 input word. Failure to do this produces an error indication in the computer when the card is read.

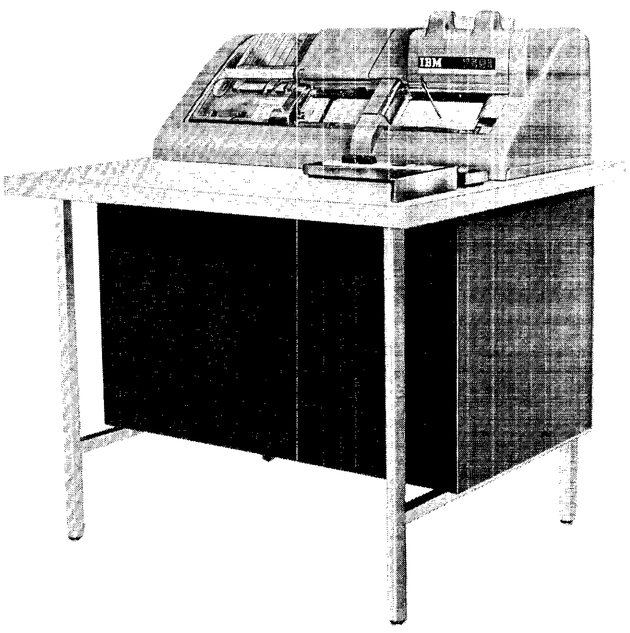


Figure 143. IBM 7501 Console Card Reader

Error Checks

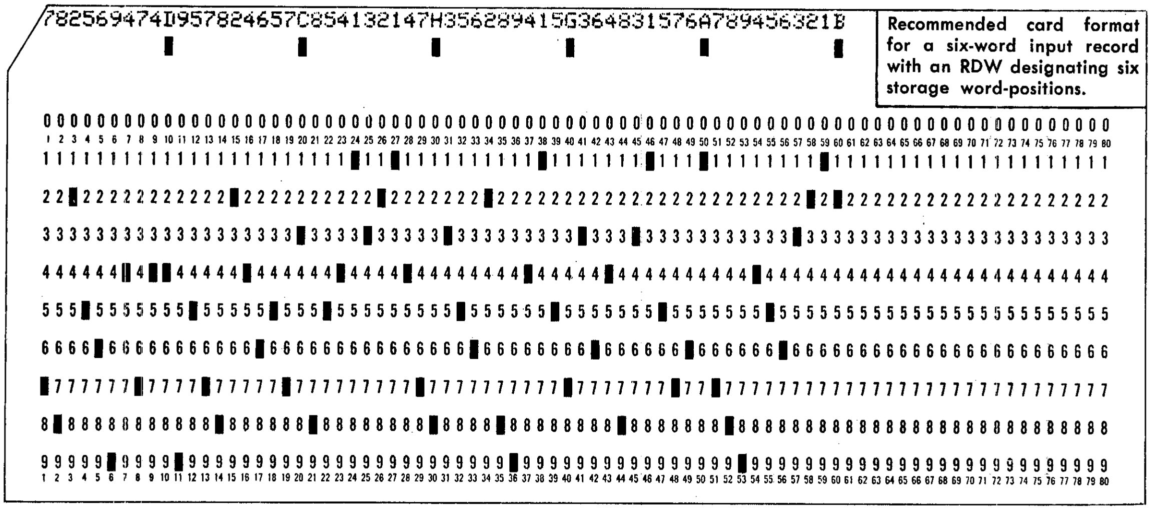
Circuits within the computer check the information received from the 7501 to detect: (1) double numeric punches in any columns of input words, (2) the lack of zone punching in the units column of input words, and (3) the presence of an incompleted input word. When the computer senses any of these error conditions, it automatically steps the instruction counter one position, causing a branch to a programmed error routine. (If there were no error, the instruction counter would have advanced three positions.) The computer accepts as valid, however, the sensing of fewer complete words than are specified in the record definition word. It also accepts as valid a record definition word of more than eight core storage positions. Although each of these conditions fails to use some reserved positions of storage, it does not constitute a data error. The stored program itself must contain a check of the exact number of words to be received from a card, if such a check is desired.

Blank columns within a ten-column word create an error condition. For example:

$$3\ 1\ 2\ b\ 6\ 4\ 9\ b\ 8\ \overset{12}{2}$$

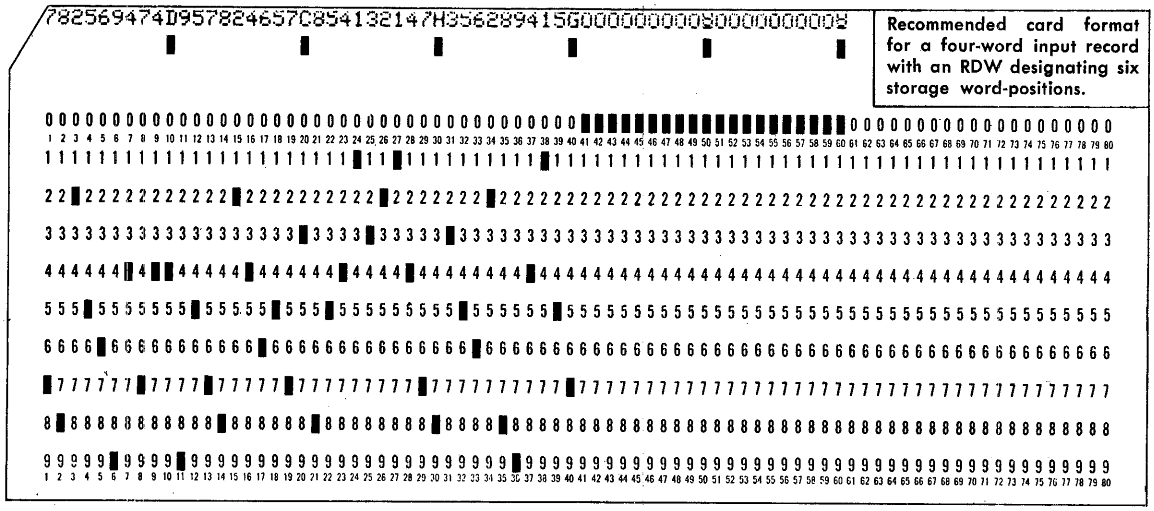
All columns within a word must have a numeric punch. For further information on programming, see the *IBM Programming Systems Reference Manual, IBM 7070-7074 Utility Programs.* Form C-28-6110.

In addition to system error checks, the 7501 contains an error light and a buzzer to signal any of the following conditions: (1) a full stacker, (2) a feeding failure (even in end-of-file mode), (3) an empty hopper prior to a read command (when not in end-of-file mode), and (4) the failure of a card to pass to the prestacker.


782569474D957824657C854132147H356289415G364831576A789456321B

Recommended card format for a six-word input record with an RDW designating six storage word-positions.

Figure 144


782569474D957824657C854132147H356289415G000000000800000000008

Recommended card format for a four-word input record with an RDW designating six storage word-positions.

Figure 145

Keys, Lights, and Switches

The operator may signal to, or be signaled by, the 7501 through the following devices (listed from left to right as they appear on the operator's panel).

Feed Reset Key: This key has three uses: (1) to feed the first card into the preregister position, (2) to release a card held by the reader after a machine halt caused by an error condition (and to turn off the error light), and (3) to take the reader out of end-of-file mode (and turn off the end-of-file light).

Independent Operation Switch: This switch must remain in the OFF position, signifying that the reader is connected with the system. It is turned on only for machine maintenance.

Motor Switch: This switch must remain in the ON position at all times except during maintenance procedures. The master power control for the system turns the motor of the 7501 on and off.

End-of-File Key: Pressing this key puts the machine in an end-of-file mode so that, when the hopper is empty, the reader recognizes the condition as valid and does not sound the error buzzer or light the error light. If the end-of-file key was pressed previously, information from the last card reaches core storage when the computer executes the next unit record read instruction. If additional cards are run into the 7501 reader before the computer executes another read instruction, *no* end-of-file condition will occur and the program will advance to L+1 or L+3 (depending upon the validity of the information transferred). If no additional cards are run into the reader, the end-of-file condition *will* exist when the computer attempts to execute another read instruction. The end-of-file condition causes (1) the computer to interpret the read instruction as a NOP instruction and (2) the program to advance to L+2.

Error Light: This red light comes on whenever the 7501 senses one or more of the conditions described in the last paragraph of "Error Checks." It goes off as soon as the feed-reset key is depressed, provided the error condition has been corrected.

Ready Light: This white light is on whenever a card is at the preregister position, the power is on, and the independent operation switch is set to OFF.

End-of-File Light: This ivory light comes on immediately after the end-of-file key is pressed. It indicates that the 7501 is in end-of-file mode. The light remains on until the operator presses the feed-reset key (either to cancel the end-of-file mode or to feed the first card of a new file to the preregister position).

Power On Light: This green light indicates that the system is receiving current.

IBM 7550 Card Punch

One to three IBM 7550 Card Punch units (Figure 146) can be attached as output devices for the IBM 7070 and 7074 Data Processing Systems. Information from the output synchronizer is directed by the control panel to punch cards in any desired format.

During continuous operation, a maximum of 250 cards a minute can be punched.

It is possible to perform gang-punching, double-punch and blank-column detection, column splitting, etc. SPOOL (Simultaneous Peripheral Operation On Line) routines can be accomplished when priority signals for automatic program control are assigned.

Validity checking can be suspended to allow punching after an invalid character has been detected.

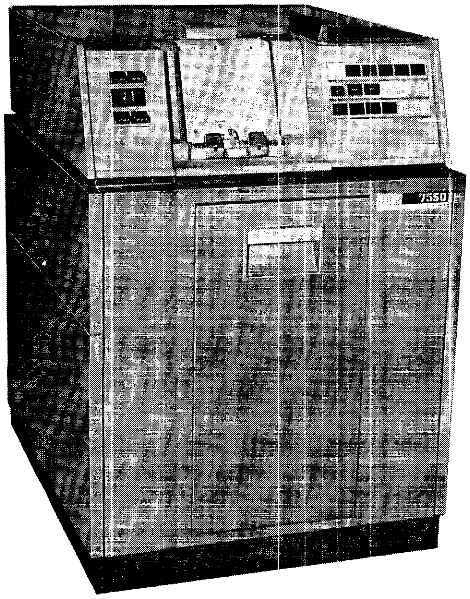


Figure 146. IBM 7550 Card Punch

Punching with internally wired format control is possible by use of the unload hubs on the control panel.

Punching and offset stacking can be controlled by control-panel signals initiated by the stored program instructions.

Further information and control panel wiring diagrams are contained in the *IBM Reference Manual, IBM Unit Record Input-Output Equipment for IBM 700-7000 Series Data Processing Systems,* Form A22-6660.

IBM 7400 Printer

One to three IBM 7400 Printers (Figure 147) can be attached to an IBM 7070 or 7074 Data Processing System. The 7400 Printer prepares printed reports from data transmitted from the system through the output synchronizer. The machine prints information from 120 print wheels that form a solid bank 12 inches wide. Each print wheel has 47 different characters, all the letters of the alphabet, all the numbers and 11 special characters.

The 7400 can print a maximum of 150 lines a minute, or 9000 lines an hour.

Printing with internally wired format control is possible by use of the unload hubs on the control panel. On the 7400, validity checking can be suspended to allow printing after an invalid character has been detected.

SPOOL (Simultaneous Peripheral Operation On Line) operations can be performed when priority signals for automatic program control are assigned by console settings.

Forms can be positioned in the machine automatically by the use of the carriage, which is set up for

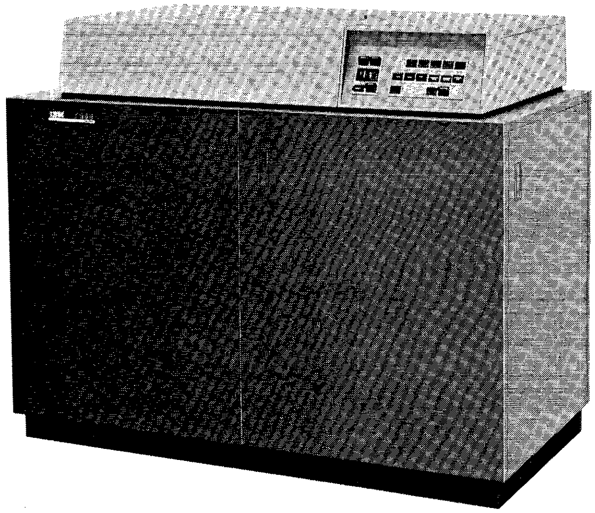


Figure 147. IBM 7400 Printer

operation by inserting a prepunched paper tape in the tape-control mechanism.

It is possible to signal the program unit from the printer. A hole in channel 9 in the carriage tape allows a signal, which sets up a control in the computer, to be tested by the program. This enables the stored program to know when a certain line on the printed form has been reached and to branch to a subroutine for obtaining page totals, controlling forms skipping, etc. The control is reset on the next print command.

Further information and control panel wiring diagrams are contained in the *IBM Reference Manual, IBM Unit Record Input-Output Equipment for IBM 700-7000 Series Data Processing Systems,* Form A22-6660.

Inquiry

The IBM 7900 Inquiry Station (Figure 148) is a means of sending a message to the 7070 or 7074, having a stored-program routine process it, and obtaining a typed reply when it comes back. As many as ten inquiry stations can be used. They are organized into two groups of five stations each, called inquiry control 1 and inquiry control 2. Each group is completely independent of the other. The five (or fewer) stations in each group are interdependent; when one is being used, the others are inoperative until the reply has been typed.

Each group has a ten-word synchronizer, which is used for the inquiry input and output of the reply. Data are moved between main core storage and the synchronizers through the CPU circuitry, in a manner similar to the movement of data between the unit-record equipment and core storage. The movement of data to and from the synchronizers is controlled by record-definition words. Formats for the inquiry and reply between the station and the synchronizer are controlled by a perforated control tape at the inquiry station. Three different formats can be made available for each station. The maximum cable length connection of a station from the central processing unit is 2500 feet. Each inquiry typewriter can be used for manual typing. Single-case typewriters are used.

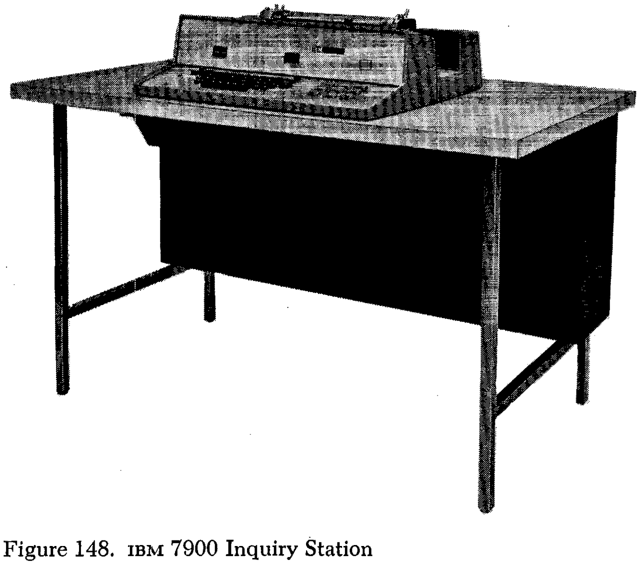


Figure 148. IBM 7900 Inquiry Station

Inquiry is automatically on a priority basis. When an inquiry is made, the program receives a priority signal and branches to a subroutine to process the inquiry data. The program branches to location 0106 on a priority signal from an inquiry control 1 station, and to 0107 on a signal from an inquiry control 2 station.

After the computer subroutine has processed the inquiry data, an inquiry-write instruction in the subroutine transfers the reply information back to the synchronizer. As soon as the synchronizer becomes filled with the reply, two independent events occur: (1) the inquiry station starts typing the reply information that is on the synchronizer, and (2) the computer proceeds to the next instruction of the subroutine. Normally, the next instruction is the priority release instruction that causes the computer to resume its main program at the place where it was interrupted by the inquiry.

Because several inquiry stations are connected with each of the two inquiry control synchronizers, a reply (output) operation requires the presence of a control word on the synchronizer to determine which of the five stations is to type the reply and which one of the three possible formats is to be followed. The control word always occupies the first word on the synchronizer. The reply control word must contain two elements of information: format number (1, 2, or 3) in position 4 of the word, and station number (1, 2, 3, 4, or 5) in position 5. The remaining positions of the word may be filled with any combination of digits. The reply control word usually is placed on the synchronizer before the reply operation (during a request operation) in either of two ways: (1) manually, by means of the inquiry station keyboard, or (2) automatically, by means of the perforated control tape at the station. During the request operation, the control word transfers to core storage along with the other request words. Because the control word information is not used by the machine during the request operation, the word may contain any combination of digits during request. Thus, if the control word brought to core storage during request does not contain the proper format and station numbers needed for reply, this information must be entered by the computer subroutine before the inquiry-write instruction transfers the control word and other reply words back to the synchronizer.

Operation

The manual operating features of an IBM 7900 Inquiry Station are the typewriter keyboard and the control keys and lights. The typewriter keyboard is on the left side of the panel (Figure 149). It has all of the features of an IBM electric typewriter, except that it types in single-case only.

The power key-light is located above the control keys and lights. If the inquiry station is in power-off status, pressing the power key-light turns on power and illuminates the key-light. If the station is in power-on status, pressing the power key-light turns off power and the key-light.

The nine lights and key-lights on the right side of the panel (Figure 149) are the means of signalling that an inquiry is to be made, selecting the format, indicating and cancelling errors, and signalling that the inquiry has been completed. REQUEST, READY, and PROCEED are lights, and the other six are key-lights. A key-light is an illuminated pushbutton switch. When the lamp beneath a key-light is on, it shows that the switch is on or that conditions are right for the switch to be turned on.

If the main 7070 or 7074 system is not on when the power key-light is pressed, the inquiry station ready light does not turn on. If the main system is on when the station power key-light is pressed, the ready light turns on, indicating that an inquiry can now be made. If the inquiry station is on (power key-light on) but the system is not (station ready light off), then turning on the main system causes the station ready light to turn on.

The operation of the control keys and lights is shown in Figure 150, which lists the manual operations and what happens as a result of them. The manual operations (four of them if no errors) are indicated by asterisks. Machine operations that take place at the same time are included in brackets.

Checking

Several automatic checking features in inquiry operations assure the correct operation of the keyboard, the transmission of correct data from and to inquiry station, and the proper station and format for the reply.

READY light on — Station is available for use.

*Press one of the three FORMAT key-lights. This selects one of the three formats on the control tape.

The control tape positions at the proper format position.

The FORMAT key-light comes on.

The REQUEST key-light (bottom row) comes on.

*Press the REQUEST key-light.

The synchronizer is signalled that an inquiry is coming.

The REQUEST key-light turns off.

The REQUEST light (top row) comes on.

If no other inquiry station in the control group is being used (if the synchronizer is available):

The PROCEED light comes on.

The REQUEST light turns off.

The control word is placed on the synchronizer either automatically by the control tape or manually by the operator at the keyboard.

*Type the inquiry message. If the message is the correct length, the RELEASE key-light comes on.

*Press the RELEASE key-light.

The PROCEED light turns off.

The FORMAT key-light turns off.

The Inquiry Control (1 or 2) stacking latch is set on, thus signalling the 7070 for priority. (See section on Priority Processing.)

The RELEASE key-light turns off after the information transfers from the synchronizer to core storage.

Figure 150. Inquiry Procedure

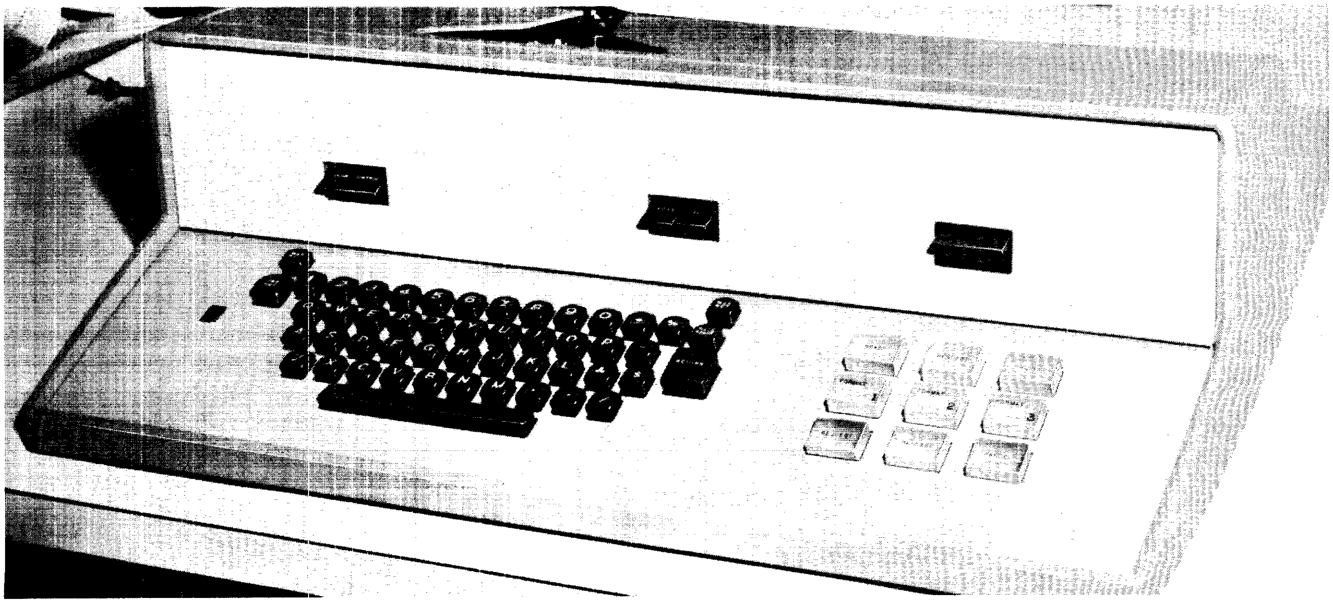


Figure 149. Inquiry Station, Typewriter Keyboard, and Operating Keys and Lights

REQUEST

Four automatic checks occur for each request operation at an inquiry station. They are described here in sequence of occurrence:

1. Keyboard Validity Check: As each key is pressed at the inquiry station keyboard, it is translated into the 2-out-of-5 coding for that character and recorded on the inquiry-control synchronizer at the position designated by the control tape. (Numeric digits are translated into one digit each; alphabetic characters are translated into two digits each.) An automatic validity check on each character, as it is translated, assures that it was translated into a valid 2-of-5 bit combination. If an invalid character is detected, the cancel key-light comes on.

2. Compare Check: After each typed character is recorded on the synchronizer, it is automatically compared with the relays set by pressing the keyboard key. This assures that the character on the synchronizer is the same as the one keyed in. If the compare operation detects an inequality, the cancel key-light comes on.

3. Keyboard Operation Check: An error in typing may be detected by the machine or by the operator. It is detected by the machine if the typed message is too short or too long; it is too short if the release key-light is pressed before its light comes on (before the end-of-message indication on the tape); it is too long if the operator attempts to continue typing after the release key light has come on. (End of message is indicated on the tape by detection of the start position of the next format.) Other typing errors are those caused by pressing an alphabetic character key when a numeric field is being typed and by pressing any key other than dash, space bar, CR, or TAB when the sign position of a word is being typed. A typing error causes the cancel key-light to turn on.

4. Synchronizer Validity Check: A validity check is made on the data moved from the synchronizer to core storage. If an invalid character is detected, the stored program advances to the *next* instruction in sequence after the inquiry-read instruction. If there are no invalid characters, the program advances to the *second* instruction in sequence.

REPLY

There are two automatic check operations for each reply operation at an inquiry station:

1. Synchronizer Validity Check: The information that moves from core storage to the synchronizer at the beginning of a reply operation is checked for validity during the transfer. If there is an invalid character, an error latch is set in the computer and the machine then proceeds to check for legitimate format and station numbers in the control word. If all characters are valid, the computer proceeds directly to the format and station numbers check.

2. Format and Station Numbers Check: The format number and station number in the control word on the synchronizer are checked to determine if they are legitimate. The format number in position 4 of the word must be a 1, 2, or 3, and station number in position 5 must be a 1, 2, 3, 4, or 5. If both numbers are legitimate, the stored program advances immediately to either (1) the *second* instruction in sequence after the inquiry-write instruction if there were no invalid characters, or (2) the *next* instruction in sequence after the inquiry-write instruction if an invalid character was detected. Then the control tape at the inquiry station is searched for the corresponding format number before typing begins. If either format or station number is incorrect, the stored program advances immediately to the *next* instruction in sequence after the inquiry-write instruction, the ready lights at all inquiry stations turn off, the control tape at the station is not searched, and no typing occurs.

During the search for format number on the control tape, the corresponding format key-light is on. As soon as the format number is found, the format key-light turns off and typing of the synchronizer contents begins. If there are invalid characters on the synchronizer, they will be typed as red asterisks.

An incorrect format or station number in the control word causes the station ready lights to turn off and prevents typing. The lights can be turned on again only if one of the following corrective procedures is followed: (1) pressing the computer reset key on the 7150 console, (2) pressing the power-off key and then the power-on key-light on the 7150 console, or (3) executing another inquiry-write instruction in the stored program with valid format and station numbers.

CANCEL KEY-LIGHTS

During a request operation at an inquiry station, failure of any one of the four automatic checks (keyboard validity, compare, keyboard operation, and synchronizer validity) causes the cancel key-light to turn on. The operator must then restart the request operation by, first, pressing the cancel key-light to turn it off and place the station in ready status again and, second, pressing the proper format key-light.

The cancel key-lights turn on at all inquiry stations, regardless of their status, if either the 7150 computer reset key or the 7150 power-on key (when power is off) is pressed. When the key-light turns on, the keyboards of all stations lock. If a request operation is in progress at a station and its proceed light is on, the proceed light will turn off. The operator must restart the request in the same way as for an automatic check failure. If a reply operation is in progress at a station, typing will stop. If a reply operation begins *after* the cancel key-lights turn on, the operation will proceed in the normal manner.

Control Tape

The control tape at an inquiry station is a 16-channel perforated tape and provides for the following:

1. Arrangement of data for entry to the synchronizer from the typewriter during a request operation, and to the typewriter from the synchronizer during a reply.
2. Control and identification of the inquiry station and format being used (control word).
3. Forms control through carriage tabulation, line spacing, etc.
4. Zero suppression on reply.

The tape is continuous, and a complete revolution can be from 12 to 48 inches in length. A sample section of tape is shown in Figure 151.

The tape moves one position for each character being typed in both inquiry and reply, selecting the word-digit positions of the synchronizer to be read into or out of.

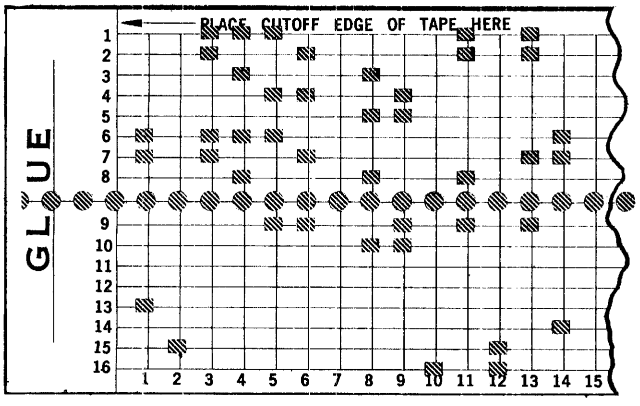


Figure 151. Inquiry Station Control Tape

The synchronizer positions not read into are filled automatically with either zeros (digit positions) or plus signs (sign positions). (An exception to this is the first word of the synchronizer in a request operation. The word must be completely filled with data in the digit positions from either the control tape or the keyboard. If no sign is placed in the sign position by the operator or by the control tape, the machine automatically furnishes a plus sign.)

CHANNEL FUNCTIONS

Channels 1-5 specify the synchronizer word that the typed digit is to go to (request) or come from (reply). The ten words are represented by the digits 1-9 for the first 9, and 0 for word 10, in 2-out-of-5 code, with code values 0, 1, 2, 3, and 6 represented by channels 1, 2, 3, 4, and 5 respectively.

SYNCHRONIZER WORD	2-OUT-OF-5 CODE	TAPE CHANNELS
1	0-1	1-2
2	0-2	1-3
3	0-3	1-4
4	1-3	2-4
5	2-3	3-4
6	0-6	1-5
7	1-6	2-5
8	2-6	3-5
9	3-6	4-5
10	1-2 (=0)	2-3

Channels 6-10 specify the digit position of the synchronizer word, in the same manner as channels 1-5 specify the word; 0, 1, 2, 3, and 6 are represented by channels 6, 7, 8, 9, and 10, respectively. The digit positions are 0-9 from left to right, or high-order to low-order:

DIGIT POSITION	2-OUT-OF-5 CODE	TAPE CHANNELS
0	1-2	7-8
1	0-1	6-7
2	0-2	6-8
3	0-3	6-9
4	1-3	7-9
5	2-3	8-9
6	0-6	6-10
7	1-6	7-10
8	2-6	8-10
9	3-6	9-10

Channel 11 has two functions: (1) It designates the low-order position for an alphabetic character if, in the same column, there are punches in channels 6-10 that represent digit positions 1, 3, 5, 7, or 9 of the synchronizer word. Detection of an "alphabetic character" column in the tape during a request operation causes the machine to place an alpha sign on the synchronizer word. Channels 1-5 of the column must contain punches that represent the appropriate synchronizer word. (2) It designates the sign of a numeric word if, in the same column, there are punches in channels 6-10 that represent digit position 0. Detection of a "sign" column in the tape during a request operation provides the operator with sign control for numeric words. Pressing the space bar on the keyboard places a plus sign on the synchronizer word; pressing the dash key places a minus sign on the word. If there is no "sign" column in the tape, the machine automatically places a plus sign on the word. During a reply operation, detection of a "sign" column in the tape causes a minus sign to type for a minus synchronizer word, and a space to occur for a plus word. The machine does not type alpha signs during a reply operation. Channels 1-5 of the column must contain punches that represent the appropriate synchronizer word.

Channel 12 has two main functions: (1) During a request operation, it specifies that the typed character is to enter the control word (first word on synchronizer), in the positions designated by channels 6-10. In this case, channels 1-5 are not needed to designate which synchronizer word, so they are used instead to specify the actual digit that is to enter the control word position. Channel 12 can be thought of as making the control tape a digit emitter. When the channel 12 punch is sensed, the tape advances automatically, and no typing takes place. (2) During a reply operation, a punch in channel 12 causes zero suppression. That is, no zeros will type until a significant digit is reached.

Channel 13 has two functions: (1) During a request operation, it indicates the starting position of tape format. The column containing the channel 13 punch must also contain punches in channels 7-9 that represent the format number (1, 2, or 3). When a format key-light is pressed, the control tape rotates until the corresponding channel 13 format number column is in position. (2) During a reply operation, a punch in channel 13 indicates the last position of tape format. The column containing the channel 13 punch must also contain either (a) punches in channel 7-9 that represent the format number (1, 2, or 3), or (b) a punch in channel 12. The combination of channel 12 and channel 13 punches must be used if a blank column (or columns) follows the reply format. A blank column would be found in that part of the tape where the ends were joined.

Channel 14 has two functions: (1) During a request operation it indicates the last position of tape format. The column containing the channel 14 punch must also contain punches in channels 7-9 that represent the format number. The column may also contain a channel 11 punch. If the channel 11 punch is present, it specifies that there must be a reply to the request before another request operation can begin. (2) During a reply operation, it indicates the starting position of tape format. The column containing the channel 14 punch must also contain punches in channels 7-9 that represent the format number.

A single "channel 14" column with the format number will suffice to indicate both last position of request format and starting position of reply format, if the format numbers are the same. A channel 13 punch must never be in the same column with a channel 14 punch.

A punch in channel 15 causes an automatic carriage return. The tape suspends data transmission within the computer until the carriage return is complete. Variable-field-length applications may require the operator to press the carriage-return key before channel 15 is reached. In this case, the control tape advances automatically to the channel 15 column.

The function of channel 16 is to cause the typewriter to tabulate automatically to the next tab stop. The advance of the tape and manual operation of the tab key for variable-length fields are identical to the previously described carriage-return operation by channel 15. Tabulations should not be programmed for fewer than three spaces.

When both channels 15 and 16 are punched, the typewriter is caused to space one column. The spacing function suspends any transmitting or receiving of data. Spacing by the control tape functions for both request and reply operations.

Neither channel 15 nor 16 may be punched in columns with word or digit punching.

Operation Code

An inquiry routine is usually started by an inquiry-read instruction. This brings to magnetic-core storage the information that had been typed at the inquiry station. When the inquiry subroutine has been completed, an inquiry-write instruction is given, which transmits the reply data to the inquiry station and automatically initiates typing the reply data. The stored program does not have to wait for the request data to be typed by the inquiry-station operator, nor for the reply to be typed by the reply instruction.

Inquiry Control
+54

Machine Description: All of the instructions for request and reply from and to the inquiry stations are included in the augmented code +54, INQUIRY CONTROL. Positions 4 and 5 of the instruction define the operation, and which of the two inquiry-control groups is involved. (As described, the specific inquiry station within the control group is identified in word 1 of the request and reply data.)

Instruction Format

S01 +54
23 Indexing word
4 Inquiry control group; 1 or 2
5 Operation:
0–Inquiry Read QR
1–Inquiry Write QW
6789 Address of first record-definition word (indexable). This word defines the core-storage area to be read into on a read operation, or read from on a write operation.

Inquiry Read: This instruction initiates the transmission of data from the synchronizer specified in position 4, to the core-storage words defined by the RDW(s) addressed by positions 6-9. Transmission is stopped by the RDW(s), or when ten words have been moved, whichever comes first. The stored program proceeds when this transmission is completed.

Inquiry Write: This code works in the same way as inquiry read, except that the flow of data is from core storage to the synchronizer, and from there to the inquiry station. The stored program can continue after the data have been transmitted to the synchronizer.

Examples: To read request information from a station in control group 2, into the core storage area for which word 4400 is the first RDW:

S01	*23*	*4*	*5*	*6789*
+54	00	2	0	4400

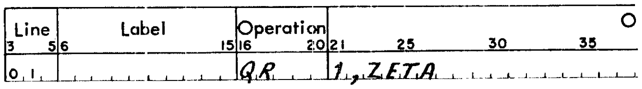
Assume that at the completion of the inquiry subroutine, the reply data has been put into the same storage area as the request data. To send this reply to the inquiry station:

S01	*23*	*4*	*5*	*6789*
+54	00	2	1	4400

Timing: An inquiry-read (QR) instruction requires an average of 3,000 microseconds. An inquiry-write (QW) instruction takes longer because of the necessity of clearing the synchronizer before the data are moved, and because of the automatic station and format check. The average time is 17,000 microseconds.

Autocoder Example (Figure 152): Zeta has been previously defined as word 2330. The assembled instruction is:

S01	*23*	*4*	*5*	*6789*
+54	00	1	0	2330



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
01		QR	1, ZETA

Figure 152

Programming Comments

Because request/reply operations on an inquiry control must share the same synchronizer, two rules of priority govern the operations:

1. A station in proceed status (typing in a request) has sole access to the synchronizer until after the information transfers from synchronizer to core storage.
2. A request operation in which the control tape is punched to indicate "reply expected" makes the synchronizer unavailable for another request operation until after the expected reply operation takes place.

The programmer should make certain that he places the inquiry-read instruction in the inquiry subroutine, not in the main program, so that it will be executed only when the computer is in interrupt mode. Otherwise, the computer will "hang up" and will require resetting by use of either the computer reset key or program reset key.

An inquiry-write instruction may be placed either in the main routine or inquiry subroutine. If a request or reply operation is in progress on an inquiry control when the computer attempts to execute a main routine inquiry-write instruction relating to the same channel control, the program will interlock until the request or reply operation is completed. Then the main routine inquiry-write instruction will be executed. Should the inquiry control stacking latch be on but masked, the computer will "hang up" and will require resetting. If the computer is in the interrupt mode because of some non-inquiry (tape, disk, card reader, etc.) interruption, execution of the inquiry-write instruction can cause the program to "hang up" if the inquiry control stacking latch is on or the control is in proceed status.

Priority Processing

One of the most important considerations in planning and programming an application for a data processing system is that of deriving the most efficient use of all components of the system. In some programs this may be card reading or card punching, in others it may be tape reading or writing, and so on. Programs are said to be input-bound, printer-bound, seek-bound, compute-bound, and so on, referring to the phase of the application that causes delay to other parts of the program. Sometimes it is desirable to combine two or more independent programs, so that one is functioning while the other is completing an operation.

An outstanding feature of the IBM 7070-7074 systems is *priority processing*, which makes it possible to combine programs and virtually eliminate any lost time waiting for an operation to be completed. There need not be a delay; one program or the other is constantly functioning.

For example, one program, called the main program, may have a comparatively large number of program steps. Another called the priority routine has relatively few instructions but involves almost continuous use of a card reader, card punch, printer, tape unit, or disk unit.

Priority processing eliminates many timing considerations by allowing unit-record, magnetic-tape, and disk-storage units to signal the stored program as each operation is completed. On the basis of these *priority signals*, a stored program can be established to determine which steps to perform next, to best use all components, and to establish automatic error routines.

When the main program is interrupted by a priority routine, the location of the next main program instruction is stored in positions 2-5 of index word 0097 and the settings of the indicators (sign change, high, low, equal, etc.) are automatically stored in location 0100. This preserves all information necessary to resume the main program and permits the priority routine to use the indicators in the meantime. If the accumulators are to be used in a priority routine, their contents should first be stored by the priority routine. The priority routine should restore the contents of the accumulators before releasing priority.

The main program then takes up exactly where it left off, with the address of its next instruction brought from index word 97 to the instruction counter and the main program information restored to the indicators.

The priority routine can be a subroutine that serves the main program, or it may be a complete program in itself, totally independent of the main program. The latter case is described by the term SPOOL, for Simultaneous Peripheral Operations On-Line. The purpose of this feature is to permit tape-to-card, card-to-tape, and tape-to-printer operations to be performed concurrently with any other program that leaves a tape unit *and* the printer, reader, or punch free. Disk storage can be used in place of a tape unit in any of these operations. Figure 153 shows schematically the time relationships of a typical main program operation with a SPOOL operation. Notice that the main routine is interrupted each time the input card reader completes the reading of a card.

More than one tape, disk storage, or card input-output may operate on a priority basis; only the main routine can be signalled for priority, however; it is not possible to interrupt a priority routine. If a second priority is ready while a first one is in progress, it will wait until the first is completed. The main routine is resumed only when no priority routines are waiting.

Several devices used in priority processing are described in this section.

Stacking Latches

A stacking latch is the means by which an input-output or storage unit signals the main routine for priority. When the unit has finished its operation, it sets a stacking latch. The main routine is constantly testing to see whether any stacking latches have been set, with no loss of time.

A full-capacity system has 64 stacking latches assigned as follows:

Two for any two of the six unit-record machines	2
One for each inquiry control group of five stations	2
One for each tape unit	40
One for each of the three access arms in each of the four 7300 disk files	12*
One each for *normal* and *attention* on each 7907 Channel	8
	—
	64

*IBM 7300 Disk Storage units are no longer in production.

Sequencing Scanner

The stacking latches are continuously scanned by the *sequencing scanner.* Figure 154 is a representation of

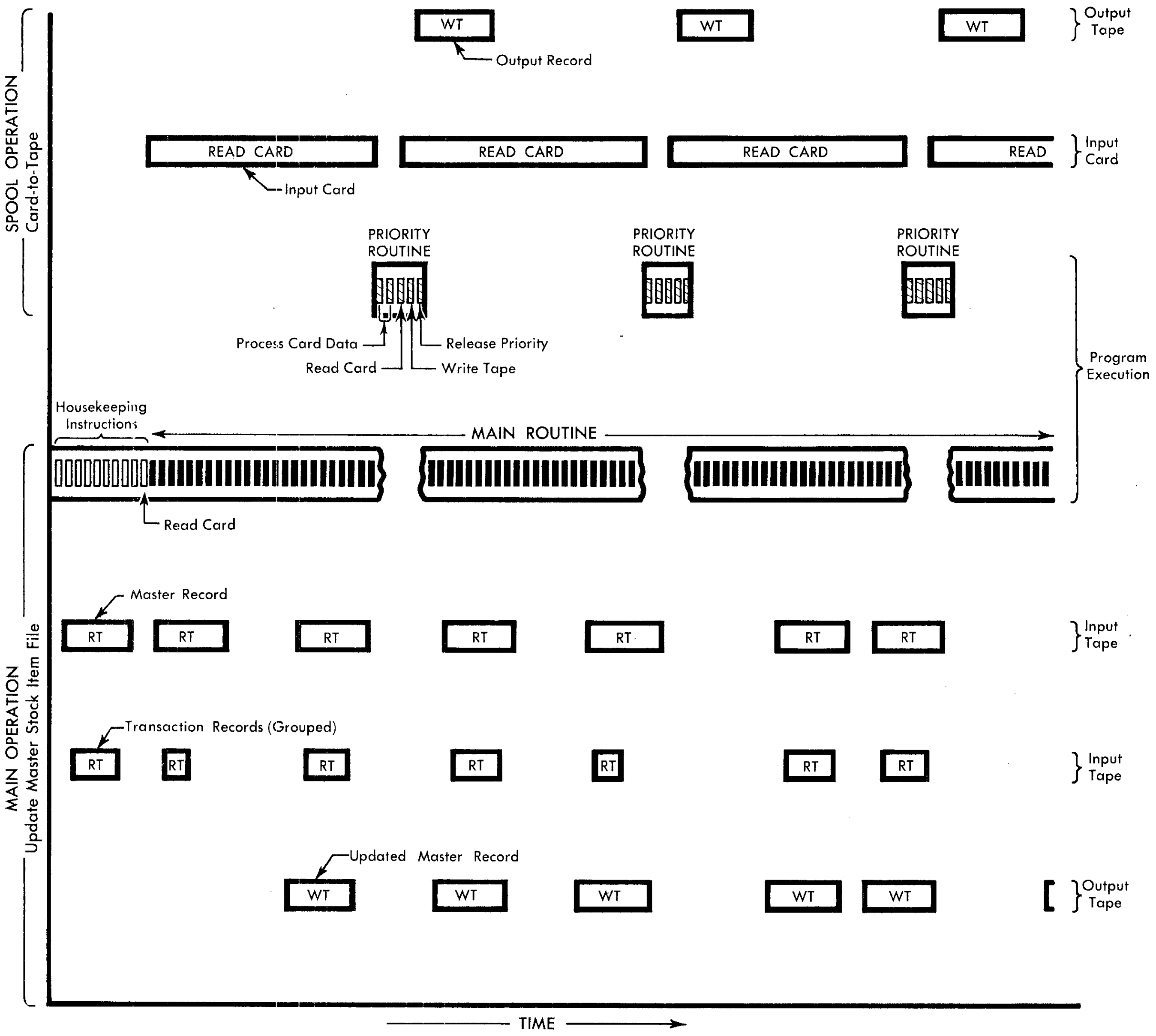


Figure 153. Relationship of Main and SPOOL Operations

how this is done. The scanner is constantly scanning, testing each of the stacking latches in turn. If two latches should be turned on at the same time, the one detected first depends on the position of the sequencing scanner at that time. Figure 154 shows the sequencing scanner stopped by the stacking latch for tape unit 0, channel 1.

If a stacking latch is turned on and it has not been masked, the sequencing scanner stops at that latch the next time it gets to it. When the priority routine is started, the stacking latch is turned off and the sequencing scanner proceeds again.

A scan of all 64 stacking latches on the 7070 takes about 1.2 milliseconds. Stacking latches on the 7074 are not searched until an interrupt occurs; about 20 microseconds or less is required.

Priority Waiting Latch

The setting of this latch causes a priority signal to be generated and a priority routine to be initiated. It is set only when all three of these things have occurred:

1. A stacking latch has been set, provided that
2. it is not masked, and
3. the sequencing scanner has reached it.

Priority Control Masks

A stacking latch can be masked so that it will not cause a priority signal when it is set. The mask prevents a

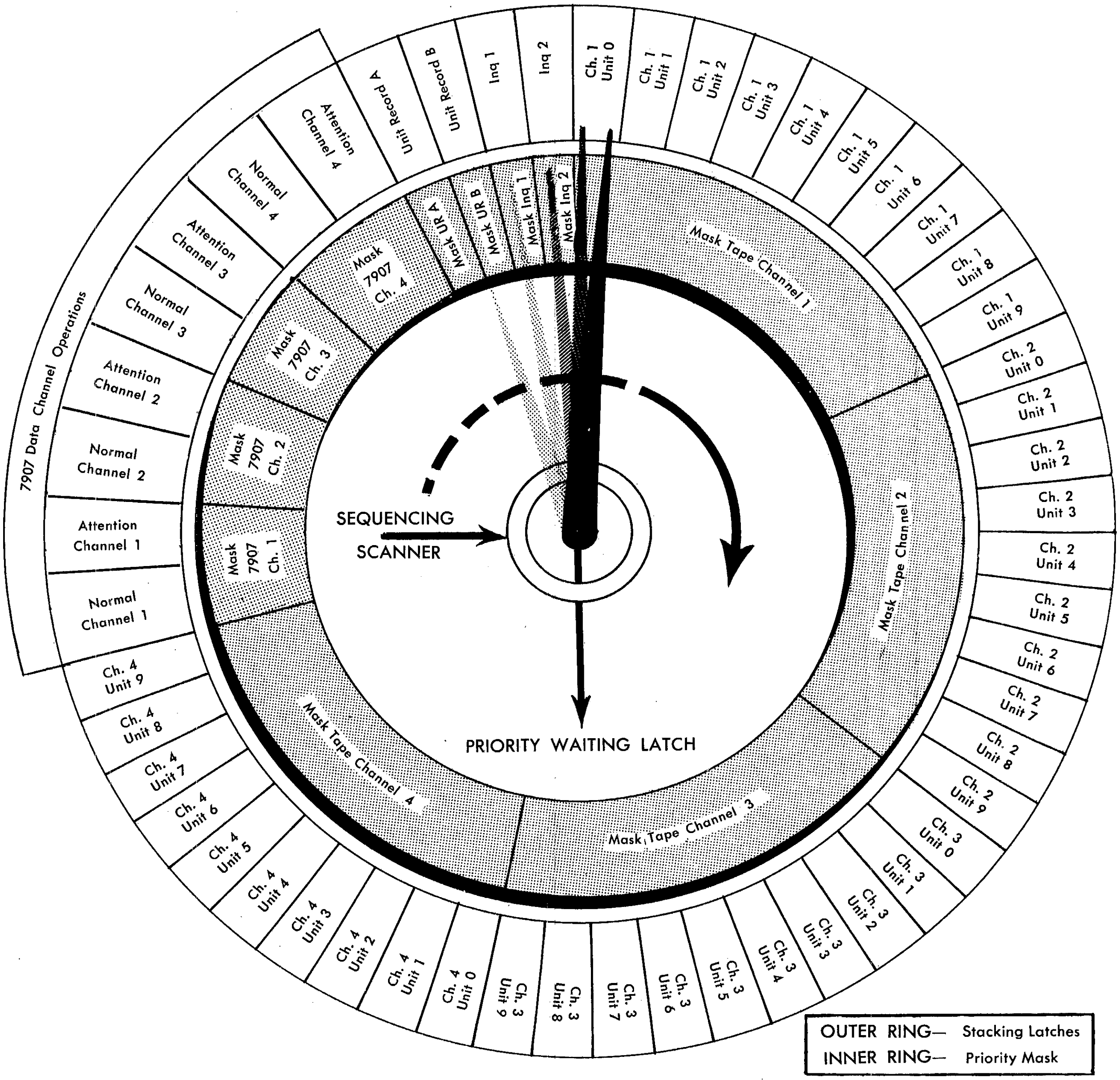


Figure 154. Stacking Latches, Priority Masks, and Sequencing Scanner

stacking latch from stopping the sequencing scanner, and thus setting the priority waiting latch. Figure 154 shows how the latches are masked. It should be noted that an entire tape channel is masked, thus prohibiting any of the ten tape units connected to it from setting the priority waiting latch. Similarly, channels 1, 2, 3 or 4 of the 7907 Data Channel associated with 1301 Disk Storage or 7340 Hypertape operations may be masked. The priority masks are set by PRIORITY CONTROL, +55.

Index Word 97

When the main routine is signalled for priority, the contents of the instruction counter are automatically sent to positions 2-5 of index word 97. The instruction counter contains the address of the next program step to be executed in the main routine. Thus, when the priority routine is completed, the PRIORITY RELEASE command returns the main routine to the point where it left off. The sign of index word 97 is set to plus.

Priority Indicator Storage Word, 0100

Ten indicators and two sense-stop switches are set during a stored program:

Accumulator overflow (one indicator for each accumulator)

High-low-equal compare

Floating-decimal overflow and underflow

Field overflow, and setting of field-overflow sense-stop switch

Sign change, and setting of sign change sense-stop switch

At the beginning of a priority routine, the settings of these ten indicators and the two sense-stop switches are automatically stored in word 0100. After their settings have been stored, all indicators are turned off and the two sense-stop switches are set to stop. The settings are restored to the indicators and switches by the PRIORITY RELEASE command at the end of the priority routine. Thus, any changes in the indicators by the priority routine have no effect on the main routine.

The values in word 0100 depend on the indicator settings, as follows:

POSITION:

9. Accumulator 3 overflow
   0. No overflow
   1. Overflow
8. Accumulator 2 overflow
   0. No overflow
   1. Overflow
7. Accumulator 1 overflow
   0. No overflow
   1. Overflow
6. Low
   0. Not low
   1. Low
5. Equal
   0. Not equal
   1. Equal
4. High
   0. Not high
   1. High
3. Floating point overflow
   0. No overflow
   1. Overflow
2. Floating point underflow
   0. No underflow
   1. Underflow
1. Field Overflow
   0. No overflow, and field-overflow sense-stop switch is at stop.
   5. No overflow, and field-overflow sense-stop switch is at sense.
   9. Overflow, and field-overflow sense-stop switch is at sense.
0. Sign change
   0. No sign change, and sign-change sense-stop switch is at stop.
   5. No sign change, and sign-change sense-stop switch is at sense.
   9. Sign change, and sign-change sense-stop switch is at sense.

If any of these digits are changed in the priority routine, the corresponding indicator setting will be affected when priority is released.

Priority Mode Latch

A 7070-7074 program is always in priority mode or non-priority mode. The priority mode latch is set on when a priority routine is started and is reset off by the priority release instruction. It prevents a priority routine from being interrupted by another priority signal.

Status Words

Every tape-read, tape-write, disk-read or disk-write instruction generates two status words, initial-status and final-status.

The initial-status word is a referenced record of the desired action. The final-status word is a reference record of the actual result of the action. These status words are important in indicating the reason for the priority signal, when such a signal is generated.

Priority signals are generated for one of two reasons:

1. A condition other than a correct-length record (CLR) has been generated.
2. The instruction completed is a priority instruction.

Failure to produce a CLR condition identifies a special condition that requires alternative action. The final-status word describes the special condition. In some of these conditions, it is necessary to refer to the initial-status word for the desired action.

The interpretation of a tape-read or write instruction by the program controls causes an initial-status word to be generated and stored at the location assigned that function, for the specific tape unit involved. The stored program then continues while the tape record is read or written and the information is placed in core storage, under control of the record-definition words.

When the last word has been read or written and stored, the final-status word is generated and placed in the location assigned to it.

Priority Operation

During each program step in the main routine a test is made to determine whether any stacking latches have been set, by testing the priority waiting latch. If it is not on, the program continues normally. If the priority waiting latch has been set, by a tape or disk operation, the status words have been formed and stored. See Figure 155.

The following takes place if the test finds the priority waiting latch set:

1. The operation currently taking place in the main routine is completed—recomplementing, etc.
2. The contents of the instruction counter (location of the next main-routine instruction), are stored in positions 2-5 of index word 97 and its sign is set to plus.
3. The indicator and sense-stop settings are stored in word 0100. Then the indicators are turned off and the sense-stop switches are set to stop.
4. The priority mode latch is set.
5. The stacking latch and priority waiting latch are turned off and the sequencing scanner starts again.
6. If the interrupt was caused by tape or disk, the address of the final status word is placed in index word 99.
7. The program branches to the location of the first instruction in the priority routine. These locations depend on what caused the priority signal, and they are covered in the next sections.

Unit-Record Priority

A priority routine caused by the completion of a read, punch, or print operation is called unit-record priority.

Two unit-record operations can be available for priority at any one time. Console switches determine whether or not there will be priority, and which unit or units will be involved. The two priority controls are called A and B, and each can be assigned to any of the six input-output units. Assignment is made by the console dials. The right-hand dial in each set determines the unit that will set that stacking latch each time an operation is completed (Figure 190).

In operations involving magnetic tape, such as card-to-tape, tape-to-printer, etc., priority signalling of the main program can be automatically delayed until both the card unit and the tape channel are free. This is done by setting the left-hand dial to channel control 1, 2, 3 or 4. When the unit record priority A dial is set to channel control 2, for example, the sequencing scanner, when stopped by the unit record A stacking latch, will not set the priority waiting latch; however, the main program will not be interrupted until channel 2 is free.

The first instruction in a priority routine initiated by priority control A is located in word 0104; for priority control B, it is in word 0105. The instruction in each of these locations is normally an unconditional branch to the location of the first instruction in the priority routine.

Inquiry Priority

Inquiry priority is automatic. A manual inquiry to storage from any of the five stations in each inquiry control group automatically starts a priority routine (unless, of course, the stacking latch is masked). A stacking latch for inquiry control 1 or 2 is set by pressing the release key at the inquiry station, after the inquiry request has been typed.

A priority routine resulting from an inquiry control 1 station starts in location 0106; resulting from an inquiry control 2 station, in 0107. As with unit-record priority, the instructions in these locations are usually unconditional branches.

Tape Priority

The priority signalling of the main program by the completion of a tape operation is called tape priority. There are two causes of tape priority: normal and unusual-condition. Normal tape priority means that the main program will be automatically signalled at the conclusion of the tape operation, just as in card priority. Tape priority, however, is under control of the stored program rather than the console. A plus sign in the tape control instruction determines that there will be a priority signal at the conclusion of the operation; a minus sign determines that there will not be a signal for normal condition. In Autocoder symbols, the letter P is used if the operation is to signal normal priority. Unusual-condition priority occurs regardless of the sign of the instruction.

The computer cannot execute an instruction involving a 729 tape unit when the stacking latch for that unit is on. The computer interlocks if it attempts to execute such an instruction between the time the latch turns on and the time the sequencing scanner reaches the latch. After the scanner reaches the latch, the computer will proceed if it is in non-priority mode and the latch is not masked. Otherwise, the computer will stop. If it proceeds, the address of the instruction involving the tape unit will be stored in positions 2-5 of indexing word 97 as the computer enters priority mode.

729 TAPE STATUS WORDS

Status words are automatically provided, with no loss in time, for tape priority operations to help the programmer determine what operations caused the priority signal. Two status words are formed: *initial* and *final*.

729 Tape Initial Status Words: Every tape instruction that might cause a priority signal when it is completed automatically creates an initial status word at the beginning of the operation. The format of the initial-status word is as follows:

S The sign of the .729 tape instruction: plus, priority; minus, no priority.

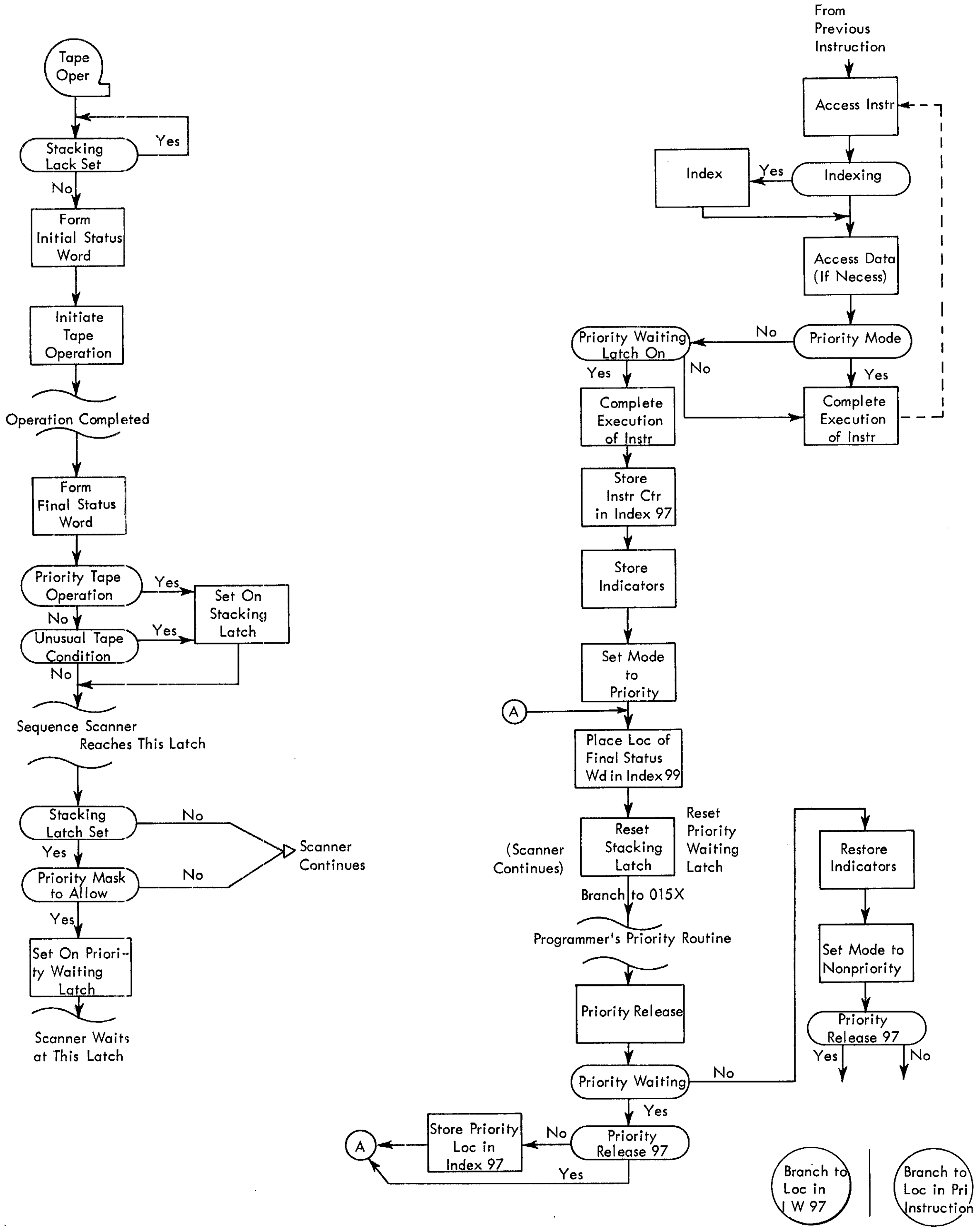


Figure 155. Priority Processing Flow Chart

0 Always 8, from position 0 of the instruction, identifying it as a 729 tape operation.

1 Defines the specific operation—contains the digit in position 5 of the 729 tape instruction.

2345 Contents of the instruction counter when the 729 tape instruction was given—the location of the next sequential instruction after the 729 tape instruction.

6789 Contents of the corresponding positions of the 729 tape instruction. If position 5 of the instruction (position 1 of this word) contains a digit 1-9, it is the location of the first RDW. If position 5 contains 0, position 9 contains the operation code, and positions 6-8 can contain any digits.

Each 729 tape unit has a location for an initial status word: 0160-0169 for the ten units for 7604 channel 1, words 0170-0179 for the tens units for 7604 channel 2, 0180-0189 for 7604 channel 3, and 0190-0199 for 7604 channel 4. Each time an initial status word is created by a tape control instruction, it is stored in the location corresponding to the 729 tape unit and 7604 channel specified.

729 Tape Final Status Words: A final status word is automatically created at the conclusion of each 729 tape read or write operation. The final status word for each tape unit is stored in a location whose address is 50 *less* than that of the initial status word: 0110-0119 for 7604 channel 1, 0120-0129 for 7604 channel 2, 0130-0139 for 7604 channel 3 and 0140-0149 for channel 4. The tens position specifies the 7604 channel, 1-4; and the units position specifies the 729 tape unit, 0-9. The format of the final status words is as follows:

S Always plus.

0 This position contains a digit put in by the program itself and is not disturbed by formation of the rest of the word. It is used to determine the exact address that the program will branch to in starting the priority routine.

1 Condition code, indicating the cause of the priority signal:

0. TWE (729 tape word error)
1. ERROR
2. CLR (correct length record)
3. SLR (short length record)
4. LLR (long length record)
5. EOF (end of file)
6. EOS (end of segment)
7. SCLR (short character length record)

2345 Working address of the record definition register at completion of the operation—the last core-storage location used by the last RDW in the operation (except for LLR).

6789 Contents of the RDW address register—the location of the last record-definition word used, +1.

The location of this word (0110-0149) is automatically stored in the indexing portion (positions 2-5) of index word 99 when the priority routine is started, the sign of 0099 is set to plus, and the remaining positions of the word are set to zeros.

The final status word developed as a result of (P)TM and (P)TSM is a special case. Only the condition code (position 1) resulting from the operation is inserted. The remaining positions and the sign are left unchanged. This is because RDW's are not used by these instructions.

Tape Priority Routine Start Address: To start a 729 tape-priority routine, the program branches to one of the ten locations 0150-0159, determined by the digit in position 0 of the final status word.

CONDITION CODES

The condition code in position 1 of the final status word indicates whether the priority signal is due to a normal or an unusual condition, and specifically what the condition is. All but CLR (2) are unusual-condition codes (they may be errors, or not). The digits in this position mean:

0. TWE—tape word error: Occurs during 729 tape reading if the record has: (a) Less than five digits or more than ten digits before the sign character is detected (numeric mode); (b) A mode change character (delta) anywhere in the record except between words. When the TWE condition occurs, some of the 729 tape information may not reach storage and that which does reach storage may be mislocated or invalid. Accurate analysis of the error condition requires re-reading the record with the tape read all alpha (TRA) instruction. (See the "IBM 729 Tape Errors" section of this manual.)

1. Error

a. Vertical check on tape character.
b. Longitudinal check on a tape record.
Read c. Two-of-five validity check on translated data.
d. A tape character not translated to two-of-five coding.
e. Tape unit density setting mismatch.
f. Insufficient signal strength.
g. Address failure.
h. Translator failure.
Write i. Invalid data in initial or final status words.
j. Record definition word error.
k. Timing error.
l. Loss of data--overload.
m. Validity error detected by two-gap head.

Condition 1 errors due to item e are caused by attempts to read tape recorded in a density that does not match the density setting of the tape unit used.

If condition 1 is caused by a tape data error, the channel busy latch releases and efforts to correct the error are possible. (See "IBM 729 Tape Errors" section of this manual.) If condition 1 is caused by machine errors listed as items g, h, i, j, or k, however, the flow of data between tape and storage stops and the channel busy latch does not release. This condition of the channel busy latch prevents execution of subsequent corrective tape operations in the program; if efforts to resume operation do not turn off the channel busy latch, customer engineering attention is indicated.

On a system equipped with Engineering Change 249841, a timing error (item k) will not halt the flow of data between tape and core storage, nor will the channel busy latch remain set. This permits operations to proceed after a non-recurrent timing error.

The occurrence of a 0 or 1 error condition takes precedence over any other priority conditions that may exist at the same time.

2. *CLR—Correct Length Record:* This code occurs if none of the unusual conditions has occurred. A priority signal results only if the tape instruction had a plus sign.

3. *SLR—Short Length Record:* Occurs if the working and stop addresses in the record definition register are not the same when the end of the 729 tape record (the IRG) is reached, in tape reading. The address in positions 2-5 of the final status word is that of the last core-storage word filled. Short-length record occurs in TAPE-SEGMENT BACKSPACE if the load point (beginning of the tape) is reached before the specified number of segments has been spaced over.

4. *LLR—Long Length Record:* Occurs if the working and stop addresses in the record definition register coincide, but the end of the tape record (the IRG) has not yet been reached, in 729 tape reading. The contents of positions 2-5 of the final status word is the sum of the address of the last core-storage word filled plus the number of excess words in the tape record. This will indicate to the programmer the number of words by which the record is greater than the storage area provided.

5. *EOF—End of File:* Occurs when reading a tape mark or writing into the foil strip, indicating that the end of the 729 tape reel has been reached. Either of these operations turns on the end-of-file indicator. Until the indicator is turned off, every subsequent read or write instruction will result in the EOF priority condition. It is possible to "coast" into the reflective spot after a write operation has been completed. If this happens, an EOF condition is indicated at the completion of the following write instruction. End-of-file occurs in TAPE-SEGMENT FORWARD SPACE and TAPE-SEGMENT BACKWARD SPACE if the tape mark is read before the specified number of segments has been passed over.

6. *EOS—End of Segment:* Occurs on reading a tape segment mark during a 729 read instruction.

7. *SCLR—Short Character Length Record:* Occurs only in reading alpha data from 729 tape if the number of characters does not fill an exact number of core-storage ten-digit words; that is, the number of alphameric characters is not a multiple of five. When this condition occurs, the remainder of the word (low-order positions) is filled with zeros (alpha coding for blanks). The address in positions 2-5 of the final status word is that of the last core-storage word filled.

PRECEDENCE OF 729 TAPE PRIORITY CONDITIONS

It is possible for more than one 729 tape priority condition to occur at any time. For example, in writing a tape record, correct-length-record and end-of-file may both occur.

Figure 155 shows the sequence of precedence of the condition codes and which codes apply to each 729 tape operation. Note that status words are formed for every operation that might initiate a priority signal.

Priority Codes

These instructions are used specifically with the automatic priority processing feature of the 7070-7074. (Other operation codes, such as +81, +91, etc. instruct an operation to signal priority at completion, but they do not affect priority directly.)

These priority codes are presented in the sequence shown in Figure 157.

Priority Control

+55 **PC, PC1**

Machine Description: This operation determines which stacking latches will be masked and which will not. The mask is contained in the word addressed by positions 6-9 (indexable). This word determines which stacking latches can stop the sequencing scanner and thus initiate a priority routine. A mask is effective until another priority-control instruction resets it.

Instruction Format

S01 +55

23 Indexing word for modifying the address of the mask word.

4 Not used

5 Must be a 0 for systems lacking 7907 Data Channel (s).

The instruction mnemonic is PC.

May be either a 0 or a 1 for systems having 7907 Data Channel(s). The instruction mnemonic is PC1. A 0 indicates that the mask word applies to 7907 I-O devices.

					CONDITION CODES, IN ORDER OF PRECEDENCE							
					HIGHEST ←							LOWEST →
			PRIORITY	STATUS	UNUSUAL CONDITION							NORMAL
MNEMONIC	NAME	OPER.	POSSIBLE?	WORDS?	0. TWE	1. ERROR	5. EOF	6. EOS	4. LLR	7. SCLR	3. SLR	2. CLR
		Position 5:										
TR	Tape Read	1	Yes	Yes	TWE	Error	EOF	EOS	LLR	SCLR	SLR	CLR*
PTR	Priority Tape Read	1	Yes	Yes	TWE	Error	EOF	EOS	LLR	SCLR	SLR	CLR
TRR	Tape Read per Record Mark Control	2	Yes	Yes	TWE	Error	EOF	EOS	LLR	SCLR	SLR	CLR*
PTRR	Priority Tape Read per Record-Mark Control	2	Yes	Yes	TWE	Error	EOF	EOS	LLR	SCLR	SLR	CLR
TW	Tape Write	3	Yes	Yes		Error	EOF					CLR*
PTW	Priority Tape Write	3	Yes	Yes		Error	EOF					CLR
TWR	Tape Write per Record-Mark Control	4	Yes	Yes		Error	EOF					CLR*
PTWR	Priority Tape Write per Record-Mark Control	4	Yes	Yes		Error	EOF					CLR
TWZ	Tape Write with Zero Elimination	5	Yes	Yes		Error	EOF					CLR*
PTWZ	Priority Tape Write with Zèro Elimination	5	Yes	Yes		Error	EOF					CLR
TWC	Tape Write with Zero Elimination and per Record Mark Control Combined	6	Yes	Yes		Error	EOF					CLR*
PTWC	Priority Tape Write with Zero Elimination and per Record Mark Control Combined	6	Yes	Yes		Error	EOF					CLR
TSF	Tape Segment Forward Space per Count	7	Yes	Yes		Error	EOF					CLR*
PTSF	Priority Tape Segment Forward Space per Count	7	Yes	Yes		Error	EOF					CLR
TSB	Tape Segment Backward Space per Count	8	Yes	Yes		Error	EOF				SLR	CLR*
PTSB	Priority Tape Segment Backward Space per Count	8	Yes	Yes		Error	EOF				SLR	CLR
TRA	Tape Read All Alpha	9	Yes	Yes		Error	EOF	EOS	LLR	SCLR	SLR	CLR*
PTRA	Priority Tape Read All Alpha	9	Yes	Yes		Error	EOF	EOS	LLR	SCLR	SLR	CLR
		Position 9:										
TSEL	Tape No-op Select	0	No	No								
TM	Tape Mark Write	1	Yes	Yes		Error	EOF					CLR*
PTM	Priority Tape Mark Write	1	Yes	Yes		Error	EOF					CLR
TRW	Tape Rewind	2	No	No								
TRU	Tape Rewind Unload	3	No	No								
TRB	Tape Record Backspace	4	No	No								
TSM	Tape Write Segment Mark	5	Yes	Yes		Error	EOF					CLR*
PTSM	Priority Tape Write Segment Mark	5	Yes	Yes		Error	EOF					CLR
TSK	Tape Skip	6	No	No								
TEF	Tape End of File Off	7	No	No								

*CLR does not initiate a priority condition.

Figure 156. Precedence of IBM 729 Tape Priority Conditions

CATEGORIES	OP CODES	NAMES	MNEMONICS
Priority mask control	+55	Priority Control, 7604 Priority Control, 7907	PC PC 1
Stacking latches	+60	Stacking latch test: Any stacking latch Unit-record Inquiry controls 729 tape channels 7907 channels	 BAL BUL BQL BTL BDCL,BDCA
	-61	Stacking latch set: Unit-record Inquiry controls 729 tape channels 7907 channels	 ULN QLN TLN DCLN,DCAN
	-62	Stacking latch reset: Unit-record Inquiry controls 729 tape channels 7907 channels	 ULF QLF TLF DCLF,DCAF
Release	+64	Priority release	PR

Figure 157. Priority Operation Codes

The mask word addressed is checked for a 0 or a 1 in each position. A 0 allows priority and a 1 masks priority for the stacking latches.

7907 I-O Devices:

DIGIT POSITION OF MASK WORD	STACKING LATCHES
S	Always plus
0	Normal latch, channel 1
1	Attention latch, channel 1
2	Normal latch, channel 2
3	Attention latch, channel 2
4	Normal latch, channel 3
5	Attention latch, channel 3
6	Normal latch, channel 4
7	Attention latch, channel 4
8-9	Not used

Non-7907 I-O Devices:

DIGIT POSITION OF MASK WORD	STACKING LATCHES
S	Not used
0	Tape channel 4
1	Tape channel 3
2	Inquiry control group 2
3	Inquiry control group 1
4	7300 disk read and write
5	7300 disk seek
6	Tape channel 2
7	Tape channel 1
8	Unit record B
9	Unit record A

6789 Address of the mask word.

Examples: To set the priority masks according to the 1's and 0's in word 1590:

S01	*23*	*45*	*6789*
+55	00	00	1590

Contents of word 1590: +00111 10011. After this instruction, only the tape units can cause priority signals.

To set the priority masks according to the 1's and 0's in word 1500, indexed by positions 2-5 of IW 73:

S01	*23*	*45*	*6789*
+55	73	xx	1500

Contents of the storage word: +00000 00000. After this instruction, all of the stacking latches can cause priority signals.

7074 Timing: 12 microseconds.

7070 Timing: 36 microseconds.

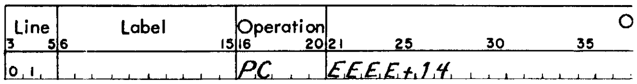
Comments: If there is a digit other than zero or one in the mask word, an error is signalled and the machine stops. The sign of the word is ignored.

A mask does not prevent a stacking latch from being set. It does, however, prevent a stacking latch from setting a priority waiting latch, and thus sending a priority signal and initiating a priority routine. If a stacking latch is masked and is set, and later a PC instruction is given which unmasks that stacking latch, the sequencing scanner will stop at the stacking latch and signal priority.

Note that there are 14 masks for the 64 stacking latches. The inquiry-control and the unit-record masks, in digit positions 2-3 and 8-9 of the mask word respectively, each control one stacking latch. Each tape-channel mask controls the ten stacking latches for the ten tape units on that channel. The 7300 disk read-write mask covers all 12 disk-storage stacking latches, one for each access arm, and the disk-seek mask covers the same 12 stacking latches for seek operations. The 7907 channels 1, 2, 3, and 4 are each masked by a priority mask word. The masks have no effect on the operations to test, set, or reset the stacking latches.

Autocoder Example (Figure 158): Assume that EEEE has been previously defined as word 2300. The assembled instruction is:

S01	*23*	*45*	*6789*
+55	00	00	2314



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		PC	EEEE+14

Figure 158

Stacking Latch Test
+60

Machine Description: This augmented instruction tests a specified stacking latch and branches to the address in positions 6-9 (indexable) if it is on. The stacking latch is designated in positions 4-5, and its setting is not changed by the operation.

Instruction Format

S01 +60

23 Indexing word for modifying the branch address

45 Designates the particular stacking latch to be tested:

00	Any stacking latch	BAL
01, 02	Unit-record latch A, B	BUL
03, 04	Inquiry controls 1, 2	BQL
10-19	729 tape units 0-9 on 7604 channel 1	BTL
20-29	729 tape units 0-9 on 7604 channel 2	BTL
30-39	729 tape units 0-9 on 7604 channel 3	BTL
40-49	729 tape units 0-9 on 7604 channel 4	BTL
81-84	Normal latches, 7907 channel 1-4	BDCL
91-94	Attention latches, 7907 channel 1-4	BDCA

6789 Branch address if the specified stacking latch is on.

Examples: To branch to location 3200 indexed by positions 2-5 of IW 63, if any of the 56 stacking latches is on:

S01	*23*	*45*	*6789*
+60	63	00	3200

To branch to location 2111, if the stacking latch for 729 tape unit 0 on 7604 channel 2 is on:

S01	*23*	*45*	*6789*
+60	00	20	2111

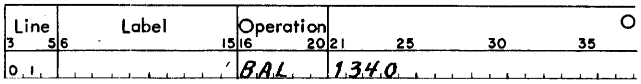
7074 Timing: 7 microseconds.

7070 Timing: 36 microseconds.

Comments: This instruction can be used as a means of branching to a subroutine at a point determined by the main program, instead of when the sequencing scanner detects the stacking latch. Note that the setting of the stacking latch is not changed by this operation.

Autocoder Examples (Figures 159-163): Actual address 1340 is used. The instruction assembled from Figure 159 is:

S01	*23*	*45*	*6789*
+60	00	00	1340

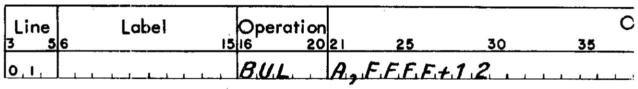


Line	Label	Operation	
3 5	6 15	16 20	21 25 30 35
01		BAL	1340

Figure 159

Assume that FFFF has been defined as word 2650. The instruction assembled from Figure 160 is:

S01	*23*	*45*	*6789*
+60	00	01	2662

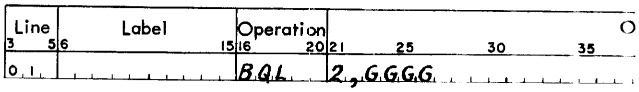


Line	Label	Operation	
3 5	6 15	16 20	21 25 30 35
01		BUL	A,FFFF+12

Figure 160

Assume that GGGG has been defined as word 2946. The instruction assembled from Figure 161 is:

S01	*23*	*45*	*6789*
+60	00	04	2946

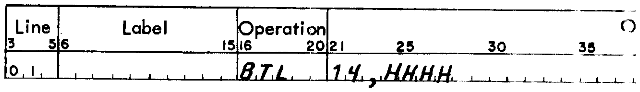


Line	Label	Operation	
3 5	6 15	16 20	21 25 30 35
01		BQL	2,GGGG

Figure 161

Assume that HHHH has been defined as word 1505. The instruction assembled from Figure 162 is:

S01	*23*	*45*	*6789*
+60	00	14	1505

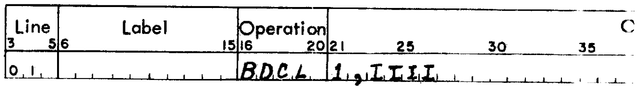


Line	Label	Operation	
3 5	6 15	16 20	21 25 30 35
01		BTL	14,HHHH

Figure 162

Assume that IIII has been defined as location 4154. The instruction assembled from Figure 163 is:

S01	*23*	*45*	*6789*
+60	00	81	4154



Line	Label	Operation	
3 5	6 15	16 20	21 25 30 35
01		BDCL	1,IIII

Figure 163

Stacking Latch Set On
−61

Machine Description: This augmented code turns on a specified stacking latch. Positions 4-5 designate the latch to be set on, in the same manner as the stacking-latch test instruction designates the latch to be tested.

Instruction Format

S01 −61

23 Indexing word. Even though positions 6-9 are not used, they are modified by positions 2-5 of an indexing word, if one is specified.

45 Designates the particular stacking latch to be turned on:

01, 02	Unit-record latch A, B	ULN
03, 04	Inquiry controls 1, 2	QLN
10-19	729 tape units 0-9 on channel 1	TLN
20-29	729 tape units 0-9 on channel 2	TLN
30-39	729 tape units 0-9 on channel 3	TLN
40-49	729 tape units 0-9 on channel 4	TLN
81-84	Normal latch, 7907 channel 1-4	DCLN
91-94	Attention latch, 7907 channel 1-4	DCAN

6789 Not used.

Example: To turn on the stacking latch for inquiry control group 1:

S01	*23*	*45*	*6789*
−61	00	03	xxxx

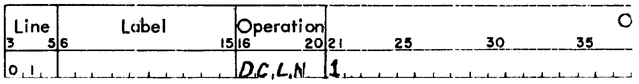
7074 Timing: 7 microseconds.

7070 Timing: 36 microseconds.

Comments: If the specified stacking latch is not masked, priority will be initiated when it is detected by the sequencing scanner. If a stacking latch is on, it causes a branch on a stacking-latch test instruction, whether it is masked or not.

Autocoder Example (Figure 164): The assembled instruction is:

```
    S01  23  45  6789
    −61  00  81  0000
```



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		DCLN	1

Figure 164

Stacking Latch Reset Off
−62

Machine Description: This code is the same as stacking latch on, except that the stacking latch specified in positions 4-5 is turned off.

Instruction Format

S01 −62

23 Indexing word. Even though positions 6-9 are not used, they are modified by positions 2-5 of an indexing word, if one is specified.

45 Designates the particular stacking latch to be turned off:

01, 02	Unit-record latch A, B,	ULF
03, 04	Inquiry controls 1, 2	QLF
10-19	729 tape units 0-9 on 7604 channel 1	TLF
20-29	729 tape units 0-9 on 7604 channel 2	TLF
30-39	729 tape units 0-9 on 7604 channel 3	TLF
40-49	729 tape units 0-9 on 7604 channel 4	TLF
81-84	Normal latch, 7907 channel 1-4	DCLF
91-94	Attention latch, 7907 channel 1-4	DCAF

6789 Not used.

Example: To turn off the stacking latch for 729 tape unit 5 on 7604 channel 2:

```
    S01  23  45  6789
    −62  00  25  xxxx
```

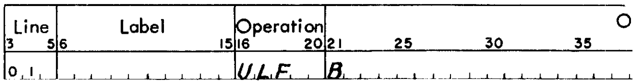
7074 Timing: 7 microseconds.

7070 Timing: 36 microseconds.

Comments: The only difference between the Autocoder symbols of this code and stacking-latch set on is the letter F instead of N. Regardless of whether the stacking latch is masked, this instruction is effective.

Autocoder Example (Figure 165): The assembled instruction is:

```
    S01  23  45  6789
    −62  00  02  0000
```



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01		ULF	B

Figure 165

Priority Release
+64 PR

Machine Description: This code is usually the last instruction in a priority routine. It releases priority, so that the main routine can continue from where it left off when interrupted by the priority signal. Index word 97, called the *priority word,* contains in positions 2-5 the address of the next instruction to be executed in the main program. (It was put there when the priority routine was started.) The location of the next instruction after a priority-release operation is determined by two factors:

1. Is there another priority routine waiting?
2. Is the address of the PRIORITY RELEASE instruction 0097?

There are four courses of action that can be taken, depending on combinations of the two factors:

1. No priority waiting, address is 0097: The address in positions 2-5 of index word 97 is sent to the instruction counter, and the next instruction is taken from that address.
2. No priority waiting, address is not 0097 (1234, for example): the instruction counter is set to 1234, and the main routine starts with that instruction. Index word 97 is unchanged.
3. Another priority routine waiting, address is 0097. The main routine is not resumed. The waiting priority routine is started instead. Index word 97 is unchanged.
4. Another priority waiting, address is not 0097 (1234, for example): the waiting priority routine is started. The address in positions 2-5 of index word 97 is replaced by the address 1234. This means that the return address to the main routine is lost, unless it was stored elsewhere during the first priority routine.

Instruction Format

S01 +64

23 Indexing word. The address in positions 6-9, whether it be positions 2-5 of 0097 or some other address, is modified by positions 2-5 of an IW, if one is specified.

45 Not used.

6789 Contains the location of the next instruction, depending on whether this address is 0097 or not, and on whether there is another priority routine waiting, as described above.

Examples: To release priority, and return to the main program at the point where it was interrupted by the priority routine:

S01	*23*	*45*	*6789*
+64	00	xx	0097

If there is no other priority routine waiting, the program takes its next instruction from the address in positions 2-5 of index word 97. If there is another priority routine waiting, it is started, and index word 97 is unchanged.

To release priority, and branch to location 1144:

S01	*23*	*45*	*6789*
+64	00	xx	1144

If there is no other priority routine waiting, the program takes its next instruction from 1144, and index word 97 is unchanged. If there is another priority routine waiting, it is started, and the address 1144 is transmitted to positions 2-5 of index word 97.

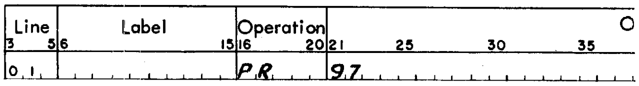
7074 Timing: 16 microseconds.

7070 Timing: 36 microseconds.

Comments: A distinction is made between an address of 0097 and an address of any other number. If the address is other than 0097, this instruction acts much like a branch code; the next instruction is taken from that location. If the address, after indexing if so specified, is 0097, the next instruction is not taken from 0097 but from the location in positions 2-5 of 0097. If a PR instruction is given in non-priority mode, it is considered a NO-OP; the next sequential instruction is given.

Autocoder Examples (Figures 166, 167): The instruction assembled from Figure 166 is:

S01	*23*	*45*	*6789*
+64	00	00	0097

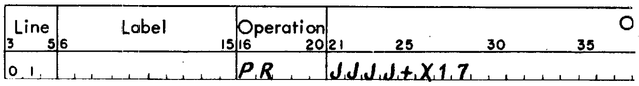


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		P R	9 7

Figure 166

Assume that JJJJ has been defined as word 2272. The instruction assembled from Figure 167 is:

S01	*23*	*45*	*6789*
+64	17	00	2272



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		P R	J J J J + X 1 7

Figure 167

IBM 7907 Data Channel Operations

Status Words

Every time the 7070-7074 executes a channel select instruction, it forms an initial status word and stores the word in a core storage location corresponding to the 7907 channel being used. The locations reserved for 7907 initial status words are:

0351 – channel 1
0352 – channel 2
0353 – channel 3
0354 – channel 4

The format of a 7907 initial status word is:

S	Sign. Same as sign of channel select instruction.
0	Same as position 0 of channel select instruction.
1	Same as position 1 of channel select instruction.
2345	Address plus one of the channel select instruction.
6789	Same as positions 6-9 of channel select instruction.

Whenever a 7907 Data Channel operation terminates, the computer forms a final status word and stores the word in a core location corresponding to the input-output channel. The locations reserved for final status words are:

0301 – channel 1
0302 – channel 2
0303 – channel 3
0304 – channel 4

The format of a 7907 final status word is:

S	Sign. Minus – attention from channel device; plus – no attention.
0	Condition code. All of these condition codes except 2 cause an immediate interrupt. The priority is 1, 5, 4, 3, 2. 1 – channel error 2 – correct length record 3 – short length record 4 – long length record (read only) 5 – unusual end
1	Always the digit "0" in normal storage mode; combines with positions 2-5 in additional storage mode to form a five-digit address in additional storage mode.
2345	Working address of RDW address register at completion of a read or write operation.
6789	Address plus one of the last channel command.

Condition codes are reset by:

1. A data channel select instruction.
2. Computer reset.

CONDITION CODE 1

Channel errors can be categorized as control errors or data validity errors. A control error is caused by improper machine operation or by the detection of illegal bit combinations that can occur if the selected channel format does not correspond with the format of the actual incoming data. A control error causes immediate termination of data transmission. The FSW will contain the address plus one of the last word transferred to or from core storage.

Data validity errors may be detected at several validity check points along the data path from the device to core storage. A data validity error permits the I-O operation to continue to the end of the RDW that is being processed.

Because the checked data path includes the device control unit, an unusual end condition originating from the device can coexist with the higher priority condition code 1. Therefore, a sense operation should always be performed to determine whether the channel device has error indications that would have caused an unusual end.

CONDITION CODE 2

A correct length record indication occurs in the absence of condition code 1, 3, 4, or 5.

CONDITION CODE 3 ON READ OR READ BACKWARD OPERATIONS

Condition code 3 (short length record) occurs when the record read is shorter, by words or characters, than the record specified by the channel command.

A condition code 3 FSW on a read or read backward operation specifies in positions 2-5 an address one greater than that of the last full word placed in core storage. The word specified in positions 2-5 carries the sign of the last full word handled and its contents indicate whether the short length record terminated with a full word or a partial word. An all-zero content indicates that the short length record terminated with a full word. Any character(s) other than zero indicate that the short length record terminated with a partial word.

A partial word indication can be caused by a 1301 format track error, a programming error (such as 6-bit mode reading of information recorded in 8-bit mode), or by lost character(s).

CONDITION CODE 3 ON WRITE OPERATIONS

Condition code 3 (short length record) occurs when the record to be written is longer, by words or characters, than can be accepted by the I-O device. This can occur as a result of an attempt to exceed the available area in disk storage, 7750 storage, or 1414-6 buffers.

A condition code 3 FSW on a write operation specifies in positions 2-5 an address two greater than the address of the last full word transferred to the I-O device. In no case, however, is this address greater than the stop address in the RDW.

CONDITION CODE 4 ON READ OR READ BACKWARD OPERATIONS

Condition code 4 (long length record) occurs only with read operations. It does not occur on write operations. A condition code 4 FSW specifies in positions 2-5 the address of the last word placed in core storage. This address is the stop address specified in the RDWR command. No indication is made of the number of characters or words that followed the last word stored.

CONDITION CODE 5

An unusual end is caused by an error condition recognized by the device control unit. A sense operation is required to determine the cause of the unusual end.

SIGN CHANGES FOR SENSE WORDS AND 7907 FINAL STATUS WORDS

The sign of the 7907 final status word and all sense words associated with the 7907 Data Channel now carries information regarding attention status for the I-O control unit.

7907 final status word sign — Minus, attention; plus, no attention

Sign of sense words — Plus, attention; minus, no attention

This information is particularly useful in connection with the IBM 7907 Data Channel Switch Optional Feature. This device divides the channel into two parts and permits the physical connection of a channel to two different adapters, such as the 1414-6 Input/Output Synchronizer and the 7631 File Control.

Program control of this feature is as follows: Switch position one is specified by a 1 in digit position 4 of the data channel select instruction; a 2 specifies switch position 2.

The sign position of sense words and the 7907 final status word is used to keep track of the attention status of the devices associated with the two switch positions.

A final status word with a plus sign indicates an absence of attention for both switch positions; a minus sign indicates an attention for one or both of the switch positions.

The sign of the sense word or words for the device on each switch position can be tested to determine the origin of the attention. A sense word with a minus sign indicates an absence of attention; a plus sign indicates an attention condition on the selected switch position.

For operations that turn on the 7907 attention stacking latch for a switch-controlled 7907 channel, a priority routine can make use of the sign of the 7907 FSW in a manner suggested in the sequence of instructions and commands shown in Table 1.

The sequence tests the signs of the sense words and the 7907 FSW and arranges for the use of a routine appropriate to the device (1301 or 1414-6) causing the attention.

Testing the sign of the FSW for a plus sign before issuing the priority release instruction insures that all attention signals from both switch positions of the channel have been accommodated.

INSTRUCTIONS			CHANNEL COMMANDS			REMARKS
LOC	OP CODE	OPERAND	LOC	OP CODE	OPERAND	WORD 0900 CONTAINS +0100004000 WORD 0901 CONTAINS +0100000325
1900	xx	x - - - - - x				Attention interrupt for ch 1 causes branch to 0321
0321	PDCU +93	00167000				Select 7907 ch 1, sw pos 1
0322	PR +64	00000097	7000	RDWS −5	000500051	Form 1301 sense words. Completion causes normal interrupt.
0311	B +01	00000325				Branch
0325	BXM −44	00502398				Branch if 1301 sense word is minus
0326 ↓ 0374	xx ↓ xx	x - - - - - x ↓ x - - - - - x				Routine to interrogate and handle 1301 sense word significance
0375	CSM −03	00600301				Compare 7907 ch 1 FSW to minus
0376	BE −41	00000321				Branch on minus FSW
0377	PR +64	00000097				Release priority; return to main program
2398 2399	ZA1 +13 ZST1−11	00090900 00090311				Change branch address for 7907 ch 1 normal attention (1414-6)
2400	PDCU +93	00268000				Select 7907 ch 1, sw pos 2
2401	PR +64	00000097	8000	RDWS −5	050005000	Form 1414-6 sense word. Completion causes normal interrupt.
0311	B +01	00004000				Branch
4000 ↓ 4030	xx ↓ xx	x - - - - - x ↓ x - - - - - x				Routine to interrogate and handle 1414-6 sense word significance
4031 4032	ZA1 +13 ZST1−11	00090901 00090311				Change branch address for 7907 ch 1 normal attention (1301)
4033	CSM −03	00600301				Compare 7907 ch 1 FSW to minus
4034	BE −41	00000321				Branch on minus FSW
4035	PR +64	00000097				Release priority; return to main program

Table 1. Programming Example for Use of Sign Information in 7907 Final Status Word and Sense Words

Sense Words

Sense words are special purpose data words. The data channel sends the sense command to the I-O control unit. The I-O control unit encodes the condition of certain latches and triggers that exists in the I-O control unit at the time of the sense command. The encoded data is sent to the data channel as packed numeric bytes. The sense bytes are decoded by the data channel. The results of the decoding are sent to core storage as complete ten-digit numeric words for use by the programmer as he desires.

Different I-O control units send different amounts of sense data. The 7640 Hypertape Control unit sends enough bytes to decode into three numeric core storage words (Figure 167.1); the 7740 Communication Control System, for one numeric word (Figure 167.2); the 7750 Programmed Transmission Control unit, for one numeric word (Figure 167.2); and the 7631 File Control unit, for two numeric words (Figure 167.3).

Stacking Latches

The 7907 Data Channel has two stacking latches associated with each of its input-output channels. The normal stacking latch for a channel is used to signal completion of an operation involving data transmission. The latch turns on if the channel select instruction which began the operation has a plus sign, or if an unusual condition (condition codes 1, 3, 4, or 5 in the final status word) occurs during the operation.

The attention stacking latch for a channel is used to signal a change in the status of the attached device when the change does not involve data transmission. The seek operation, for example, would generate a signal to turn on the attention stacking latch when the seek was completed. If the signal occurs while the normal stacking latch is on or while the channel is busy, it will wait until the normal latch turns off or until the channel becomes free before turning on the attention stacking latch.

Digit Code	Condition	Sense Data If On	
		Position	Word
5	Operator Required	0	1
6	Program Check		
6	Data Check	1	1
5	Exceptional Cond		
5	Select Register 8 Bit	2	1
6	Select Register 4 Bit		
6	Select Register 2 Bit	3	1
5	Select Register 1 Bit		
5	Sel TU Not Ready	4	1
6	Sel TU Not Loaded		
6	Sel TU File Protect	5	1
5	Reserved		
5	Inv Cont Code	6	1
6	Sel TU Busy		
6	Sel TU at BOT	7	1
5	Sel TU at EOT		
5	Corr Occurred	8	1
6	Chan Par Ck		
6	Code Check	9	1
5	Envelope Check		
5	Over-run Check	0	2
6	Ex Skew Check		
6	Trk Start Check	1	2
5	Reserved		
5	Sel TU Read TM	2	2
6	Sel TU in EWA		
6	No Data Trans	3	2
5	Reserved		
5	Read Sect Busy	4	2
6	Write Sect Busy		
6	Backward Mode	5	2
5	Reserved		
5	Log TU 0 Called Att	6	2
6	Log TU 1 Called Att		
6	Log TU 2 Called Att	7	2
5	Log TU 3 Called Att		
5	Log TU 4 Called Att	8	2
6	Log TU 5 Called Att		
6	Log TU 6 Called Att	9	2
5	Log TU 7 Called Att		
5	Log TU 8 Called Att	0	3
6	Log TU 9 Called Att		
6	Reserved	1	3
5	Reserved		
5	Reserved	2	3
6	Diag Mode Sw On		
6	Int On Corr Err	3	3
5	Err Corr Off		
5	Loop Write to Read	4	3
6	Write Clock Fast		
6	Write Clock Slow	5	3
5	Fix Check Bits		
5	Reserved	6	3
6	Reserved		
6	Reserved	7	3
5	Reserved		
5	Reserved	8	3
6	Reserved		
6	Reserved	9	3
5	Reserved		

Digit code 4 indicates no condition
Digit code 7 indicates both conditions

Figure 167.1. Sense Data for 7640 Hypertape Control

Digit Position	Digit Code	Condition	Comments
0	5	Reserved	
	6	Program Check	Invalid control information received.
1	5	Exceptional Condition	Defined in positions 2 and 3.
	6	Data Check	Defined in positions 2 and 3.
2	5	Reserved	
	6	Message Length Check	
3	5	Channel Held	
	6	Channel Queue Full	
4	5	Reserved	
	6	Reserved	
5	5	Reserved	
	6	Interface Timeout	The operation in progress in the 7740/50 was terminated because the channel failed to respond to a 7740 or 7750 service request.
6	5	Reserved	
	6	Data Message Ready	One or more data messages in 7740/50 storage are ready for transfer to 7070/74 storage.
7	5	7740/50 INBUF Available	Space is available in the 7740/50 input buffer to accept data messages from the 7070/74.
	6	Service Message Ready	One or more service messages in 7740/50 storage are ready for transfer to 7070/74 storage.

Positions 8 and 9 are not used. They always contain the digit 4.

Digit code 4 indicates no condition.
Digit code 7 indicates both conditions.

Figure 167.2. Sense Data for 7740 Communication Control System and 7750 Programmed Transmission Control

The priority branch addresses associated with normal and attention stacking latches are:

0311 – normal branch address, channel 1
0321 – attention branch address, channel 1
0312 – normal branch address, channel 2
0322 – attention branch address, channel 2
0313 – normal branch address, channel 3
0323 – attention branch address, channel 3
0314 – normal branch address, channel 4
0324 – attention branch address, channel 4

STORE COMMAND STATUS WORD OPERATIONS

The store command status word commands provide a means of keeping track of the progress of the 7907 channel command program by establishing check points. Their use is not mandatory; they may be used in a series of commands whenever desired.

First Sense Word Format

S	Plus, attention; minus, no attention	
	Digit Code	
0	5	Not applicable
	6	Program check
1	5	Exceptional condition check
	6	Data check
2	5	Invalid sequence check
	6	Invalid code check
3	5	No record found check
	6	Format check
4	5	Invalid address check
	6	Response check
5	5	Parity or check character code check
	6	Data compare check
6	5	Access inoperative check
	6	Access not ready check
7	5	File control circuit check
	6	Disk storage circuit check
8	5	Reserved
	6	Six-bit mode
9	Always 4	

Second Sense Word Format

	S	Plus, attention; minus, no attention	
		Digit Code	
Access 0	0	5	Module 0 attention
		6	Module 1 attention
	1	5	Module 3 attention
		6	Module 2 attention
	2	5	Module 4 attention
		6	Module 5 attention
	3	5	Module 7 attention
		6	Module 6 attention
	4	5	Module 8 attention
		6	Module 9 attention
Access 1	5	5	Module 1 attention
		6	Module 0 attention
	6	5	Module 2 attention
		6	Module 3 attention
	7	5	Module 5 attention
		6	Module 4 attention
	8	5	Module 6 attention
		6	Module 7 attention
	9	5	Module 9 attention
		6	Module 8 attention

Digit code 4 indicates no condition
Digit code 7 indicates both conditions

Figure 167.3. Sense Data for 7631 File Control

Command Status Word Format:

S Always plus.

0 Always 2.

1 Always the digit zero in normal storage mode; combines with positions 2-5 in additional storage mode to form a five-digit address.

2345 Address where the command status word is to be stored.

6789 Address plus one of the store command status word command that formed this command status word.

If an error occurs in a sequence of commands preceded by a store command status word command, the command status word (in a location specified by the programmer) contains the address of the first command in the sequence of commands that encountered the error situation. This information enhances the efficiency of error routine operations. A programming example is shown in Table 2.

7070-7074 STORAGE LOCATION	CONTENTS
1000	Contains +20 1000 2001 when the command sequence is between 2000 and 2004. Contains +20 1000 2005 when the command sequence is between 2004 and 2008. Contains +20 1000 2009 when the command sequence is between 2008 and 2011.
Assume that a channel-select instruction gives an initial address of 2000.	
2000	Store command-status-word in location 1000 (+)
2001	Control command (+)
2002	Read command (+)
2003	Read command (+)
2004	Store command-status-word in location 1000 (+)
2005	Control (+)
2006	Write* (+)
2007	Write (+)
2008	Store command-status-word in location 1000 (+)
2009	Control (+)
2010	Write (+)
2011	Write (−)

*Assume that an error is detected at this point. The 7907 would form and store a final status word. This would cause a normal interrupt with a branch to a priority routine to determine the nature of the error. Assume that the program desires to rewrite the record in error. The priority routine makes a new channel-select instruction using the address contained in position 1000. Because a store-command-status-word command was placed in location 2004, the address at location 1000 is now 2005 (2004 plus 1), which brings the program back to the control command to prepare the 7907 I-O device for writing the desired record.

Table 2. Programming Example of Store Command-Status-Word Operation

Scheduling Simultaneous I-O Operations on the 7070

The procedure described below permits determination of the number of I-O devices that can operate simultaneously without loss of data because of failure to gain access to core storage. The procedure assumes the following worst-case conditions:

1. Each I-O device operates at its maximum data rate.

2. Data transfers under control of an RDW for each data word (column C of Table 3), or one RDW for the entire operation (column D of Table 3).

3. I-O devices may be attached to channels in any desired combination within hardware limitations. If a 7907 Data Channel has a channel switch feature, only the higher speed device should be considered in the calculations for that channel.

The procedure has five steps:

1. Select the desired devices and their respective data rates from columns A and B of Table 3.

2. Assign a channel to each device.

3. Ascertain which channel has the device with the highest data rate and the lowest priority. Priority is in this order -- 7604 channels 1-4, or 7604 channels 1-2 and then 7907 channels 1-2.

4a. For operations using an RDW for each data word, determine the weight factor for the device selected in step 3 from column C. From this number subtract 2 for each channel with a higher priority sequence and subtract 1 for each channel with lower priority.

4b. For operations using an RDW defining more than one data word, use the weight factor from column D for the device selected in step 3 and subtract 1 for each channel with higher priority sequence.

5. If the result of 4a or 4b is greater than or equal to 0, the selected devices will operate at maximum transmission rate without loss of data. If the result is less than 0, the combination of devices may result in a loss of data when operating at maximum transmission rate.

The following examples are for operations using an RDW for each data word; column C of Table 3 is used.

7070 Example 1:

			POSITIVE	NEGATIVE
			WEIGHT	WEIGHT
CHANNEL	DEVICE	DATA RATE	FACTOR	FACTOR
7604 1	729 VI	90 KC		−2
7604 2	729 VI	90 KC		−2
7907 1	1301	90 KC	4	
			4	−4=0

This configuration will run without loss of data.

7070 Example 2:

			POSITIVE	NEGATIVE
			WEIGHT	WEIGHT
CHANNEL	DEVICE	DATA RATE	FACTOR	FACTOR
7604 1	729 VI	90 KC		−2
7604 2	729 VI	90 KC		−2
7907 1	1301	90 KC		−2
7907 2	1301	90 KC	4	
			4	−6=−2

This configuration *may* result in loss of data. Where program routines can detect and correct a loss of data, the allowable delay may be exceeded. If loss of data does occur, condition code 1 is set in the final status word.

COLUMN A	COLUMN B	COLUMN	COLUMN
CHANNEL/DEVICE	DATA RATE (KC)	C	D
7604			
729 II, V (200)	15	44	47
729 IV, VI (200)	22.5	28	31
729 II, V (556)	41.7	12	15
729 V (800)	60	6	9
729 IV, VI (556)	62.5	6	9
729 VI (800)	90	3	6
7907			
7740	48	8	11
7750	71.5	6	9
1414-6	90	4	7
1301	90	4	7

Table 3. Weight Factors for 7070

Scheduling Simultaneous I-O Operations on the 7074

The procedure described below permits determination of the number of I-O devices that can operate simultaneously without loss of data because of failure to gain access to core storage. The procedure has three steps:

1. Select the desired devices and their respective data rates from columns A and B of Table 4.

2. Assign the highest speed device to the high priority channel. Rules governing the assignment of priority are as follow:

Rule 1. The order of priority is 7907 channels 1 through 4 and then 7604 channels 1 through 4.

Rule 2. When used, an IBM 7640 Hypertape Control is attached to 7907 channels 1 and 2.

Rule 3. When devices other than 7340 Model 1 Hypertape Drives are used on 7907 channels 1 and 2 simultaneously with 729 tape units, both 7907 channels receive the weight factor of the 7907 channel device with the greater weight factor.

Rule 4. If a 7907 Data Channel has a channel switch feature, the weights for the higher speed device are used in this procedure.

Rule 5. When both 7604 channels are used, they each receive the weight factor of the 729 tape unit with the greater weight factor.

Rule 6. Under worst-case conditions, the following combinations cannot run error free:

> Exception 1 — Two 7907 channel devices and one 729 VI tape unit running at 800 characters per inch.

Exception 2 — One 7907 channel device and two 729 VI tape units running at 800 characters per inch.

Rule 7. Card equipment is interlocked and need not be considered.

3. Add the weight factors for all the devices from columns C, D, or E of Table 4, depending on the configuration being calculated. If the sum of the weights is less than or equal to 100, the combination will run without loss of data. This total weight of 100 permits error-free operation for worst-case situations where each I-O device operates in an overlapped manner at maximum speed with alpha words, and one RDW for each word.

The worst-case situation is unlikely and, where program routines can detect and correct a loss of data, the total allowable weight may be extended. For example, when each RDW defines ten or more data words and a few recoverable errors having little effect on total job time can be tolerated, the maximum total weight factor permissible is 125. A further extension to a total weight factor of 200 is permissible if no record scattering is involved.

If the highest speed device is not on the highest priority channel, the maximum total weight factor permissible is 70. If loss of data does occur, condition code 1 is set in the final status word.

COLUMN A CHANNEL/DEVICE	COLUMN B DATA RATE (KC)	COLUMN C NO 729 IN USE	COLUMN D ONE 729 IN USE	COLUMN E TWO 729's IN USE
7604				
729 II, V (200)	15	–	4	5
729 IV, VI (200)	22.5	–	5	6
729 II, V (556)	41.7	–	12	14
729 V (800)	60	–	15	19
729 IV, VI (556)	62.5	–	15	19
729 VI (800)	90	–	41	41
7907				
7740	48	10	12	14
7750	71.5	13	15	17
1414-6	90	16	18	20
1301	90	16	18	20
1302	184	39	44	49
7340 (WR)	170	38	43	48
7340 (RD)	194	40	45	50

EXAMPLE: Simultaneous operation of two channels with 729 IV Magnetic Tape units (556 cpi) and two channels of IBM 1301 Model 1 or 2 Disk Storage involves the following factors from column E: 19+19+20+20=78; such a combination will operate without errors.

Table 4. Weight Factors for 7074

Floating-Decimal-Point Arithmetic

The computing capabilities of the IBM 7070 and 7074 are greatly increased by use of floating-decimal-point operations, available as an option. In floating-decimal-point notation, a ten-digit word of data is divided into the *characteristic,* in positions 0-1, and the number itself, in positions 2-9. The eight-digit number, called the fraction, has a value between +1 and −1. The characteristic is 50 plus a power of ten, by which the fraction would be multiplied to equal its actual value. A characteristic of 50 in position 0-1 equals 0 in actual value (10^0), 51 equals +1, 49 equals −1, etc. The floating-decimal-point range of characteristic 00-99 thus equals actual-value exponents of 10 from −50 through +49.

This notation affords a very wide range in values that can be represented by a ten-digit word. The smallest absolute value (other than zero) of a floating-decimal-point number is:

CHARACTERISTIC	FRACTION
00	00000001

This represents a 1 in the 58th position to the right of the decimal. The largest value of a floating-decimal-point number is:

CHARACTERISTIC	FRACTION
99	99999999

This equals a whole number of 99,999,999 followed by 41 zeros. A number in floating-decimal-point notation may be plus or minus. The sign of a word is the sign of the fraction.

Figure 168 shows some examples of numbers in actual notation (left column) and 7070-7074 floating-decimal-point notation (right column). The other columns show how the floating-decimal-point values were obtained from the actual values.

All arithmetic operations can be performed by the floating-decimal-point codes. In all cases except the absolute-value codes, the signs of the factors are considered in the operations.

All numbers referred to in descriptions of floating-decimal-point codes are floating-decimal-point values; that is, each ten-digit word consists of an eight-digit fraction and a two-digit characteristic. There is no field definition with these codes; positions 4 and 5 are not used and can contain any digits.

Accumulators 1 and 2 are coupled for some floating-decimal-point operations. In these operations, the sign of accumulator 2 is considered the same as that of accumulator 1, and the characteristic of accumulator 2 is considered to be eight less than that of accumulator 1. The actual sign and characteristic of accumulator 2 are ignored. Because each accumulator retains its characteristic, the fraction is 16 digits in size. The coupled accumulators can be used for addition and subtraction and contain the multiplication product and dividend in division. In multiplication, two eight-digit fractions can be multiplied to give a 16-digit product. In division, an eight-digit divisor can be divided into a 16-digit dividend to obtain an eight-digit quotient and an eight-digit remainder. This is called *modified double-precision floating-decimal-point arithmetic.*

One starting factor must always be placed in accumulator 1 on a previous program step (except for Floating Zero and Add) and the result of an operation is automatically placed in accumulator 1. If so specified, the result is extended into accumulator 2. Accumulator 3 is used as a working register in all floating-decimal-point operations except Floating Zero and Add.

In floating-decimal-point arithmetic, alpha signs cannot be used. If either factor is alpha, an error stop results. Also, floating-decimal-point operations using accumulator addresses 9991-9993 will cause an error stop with the program check light on.

ABSOLUTE VALUE	SCIENTIFIC NOTATION	FRACTION	EXPONENT OF 10	CHARACTERISTIC	FLOATING-DECIMAL POINT NOTATION
+46,230	$+.4623 \times 10^{+5}$	+.46230000	+5	55	+5546230000
+3.1416	$+.31416 \times 10^{+1}$	+.31416000	+1	51	+5131416000
+0.5	$+.5 \times 10^{0}$	+.50000000	0	50	+5050000000
+0.00006	$+.6 \times 10^{-4}$	+.60000000	−4	46	+4660000000
−1234.0	$-.1234 \times 10^{4}$	−.12340000	+ 4	54	−5412340000
−0.4239	$-.4239 \times 10^{0}$	−.42390000	0	50	−5042390000
−0.0005862	$-.5862 \times 10^{-3}$	−.58620000	−3	47	−4758620000

Figure 168. Floating-Decimal-Point Examples

It is recommended that normalized factors be used in all floating-decimal-point operations; in the descriptions of operation codes that follow, normalized factors are assumed. Special applications, however, may require the use of unnormalized factors. Figure 183 shows results that occur when unnormalized factors are used.

When an operation tries to develop a characteristic greater than 99, the floating-decimal overflow indicator is turned on, and the machine either stops or continues, depending on the setting of the floating-decimal overflow stop-sense switch on the console. The term floating-decimal underflow is used to define an attempt to develop a characteristic less than 00 (minus 50 in actual value). The floating-decimal underflow and overflow indicators can be interrogated by the program. Only the overflow indicator can cause a program stop.

Two kinds of *zero* numbers can arise during floating-decimal-point operations:

Normal Zero—characteristic and fraction all zeros. Example for the floating add operation:

Factor A +52 12345678
Factor B −52 12345678
00 00000000

Order-Of-Magnitude Zero—non-zero characteristic and zero fraction. Example for the floating multiply operation:

Multiplicand +52 62340000
Multiplier +56 59630000
ACC. 1 ACC. 2
Product +58 37173342 +50 00000000

An order-of-magnitude zero is retained in accumulator 2 if it contains the low-order fraction digits of a double-precision result in which some higher-order fraction digit is non-zero. If the characteristic of accumulator 1 is greater than 00, but that of accumulator 2 is less than 00, an underflow condition is not indicated but accumulator 2 is set to zero.

Figure 169 shows the floating-decimal-point codes. The Autocoder mnemonics for all of the floating-decimal-point codes, and only those codes, start with F.

Floating Zero and Add
+75 FZA

Machine Description: The contents of the storage word addressed by positions 6-9 (indexable) are brought to accumulator 1, replacing its previous contents, and then normalized, if necessary. Accumulator 2 is set to zero.

Instruction Format

S01 +75.

23 Indexing word, for the address in positions 6-9.

45 Not used.

6789 Address of the floating-decimal-point number to be brought to accumulator 1, and normalized if necessary.

CATEGORIES	OP CODES	NAMES	MNEMONICS
Add-subtract	+75	Floating zero and add	FZA
	+74	Floating add	FA
	−74	Floating subtract	FS
	+76	Floating add double precision	FAD
	−76	Floating add double precision and suppress normalization	FADS
	+77	Floating add absolute	FAA
	−77	Floating subtract absolute	FSA
	+71	Floating round	FR
Multiply-divide	+73	Floating multiply	FM
	−73	Floating divide	FD
	−75	Floating divide double precision	FDD
Branch	+70	Floating branch overflow	FBV
	−70	Floating branch underflow	FBU

Figure 169. Floating-Decimal-Point Codes

Examples: To bring the contents of word 2662 to accumulator 1, and normalize it if necessary:

S01 23 45 6789
+75 00 xx 2662

Contents of word 2662: −24 00000001. Contents of accumulator 1, after the operation: −17 10000000, regardless of previous contents.

To bring the contents of word 622 to accumulator 1, and normalize it if necessary:

S01 23 45 6789
+75 00 xx 0622

Contents of word 622: +80 21426384. Contents of accumulator 1, after the operation: +80 21426384, regardless of previous contents. The numbers are the same because no normalizing was required.

7074 Timing: In microseconds:

$11+N^2$

N^2=number of positions of post-normalization required

7070 Timing: 108-156 microseconds, depending on the number of positions to be normalized.

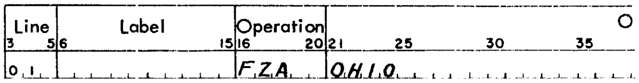
Comments: A storage word can be normalized by using this instruction and then storing accumulator 1 into the same location. A STORE ACCUMULATOR 1 instruction (ST 1) is used. To bring a word to accumulator 1 but not normalize it, use ZERO ACCUMULATOR 1

AND ADD. If the high-order digit of the fraction is nonzero, this instruction has the same result as ZA 1.

Whenever a floating-decimal-point number is normalized, its characteristic is reduced by one for each high-order zero in the original fraction. For example, the number 27 00750121 is normalized to 25 75012100. This means that the floating-decimal underflow indicator can be turned on by any operation, such as FZA, that normalizes a floating-decimal-point value. A fraction of zero carries a characteristic of 00.

Autocoder Example (Figure 192): Assume that OHIO has been defined as word 1735. The assembled instruction is:

S01	*23*	*45*	*6789*
+75	00	00	1735



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
01		FZA	OHIO

Figure 170

Floating Add

+74 FA

Machine Description: 8-digit number + 8-digit number = 16-digit result. The contents of the location addressed by positions 6-9 (indexable) are added to the amount already in accumulator 1, and the result is normalized and developed in accumulators 1 and 2. Accumulator 2 is reset before the operation, and there is a difference of 8 in the resultant characteristics of the two accumulators.

Instruction Format

S01 +74.

23 Indexing word for modifying the address in positions 6-9.

45 Not used.

6789 Address of the floating-decimal-point number to be added to the number in accumulator 1.

Examples: To add, in floating-decimal-point notation, the contents of word 2359 to the amount already in accumulator 1:

S01	*23*	*45*	*6789*
+74	00	xx	2359

Contents of accumulator 1 before the operation: +57 20415855. Contents of word 2359: +51 72885343. Contents of accumulators 1 and 2 after operation:

ACC 1	ACC 2
+57 20415927	+49 88534300

To add, in floating-decimal-point notation, the contents of word 987 to the amount already in accumulator 1:

S01	*23*	*45*	*6789*
+74	00	xx	0987

Contents of accumulator 1 before the operation: −33 21467522. Contents of word 987: +36 23013333. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
+36 22991865	+28 47800000

To add, in floating-decimal-point notation, the contents of word 1164 to the amount already in accumulator 1:

S01	*23*	*45*	*6789*
+74	00	xx	1164

Contents of accumulator 1 before the operation: −69 67714400. Contents of word 1164: −69 84002217. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
−70 15171661	−62 70000000

7074 Timing: In microseconds:

$16+(N^1-1)^*+(N^2-1)^*$, when no recomplementing required.

$19+(N^1-1)^*+N^2$, when recomplementing required.

N^1 = difference between characteristics of the numbers.

N^2 = number of positions of post-normalization required.

*Note that if $N^1 < 1$, $(N^1-1)=0$ and if $N^2 < 1$, $(N^2-1)=0$.

7070 Timing: 168-336 microseconds. The variance of as many as 168 microseconds is due to the shifting caused by the difference in characteristics, and to normalizing the result.

Comments: This instruction, like all floating-decimal-point codes, uses accumulator 1 as the *base* accumulator; the more important part of the result, the high-order portion of the 16-digit fraction, is in accumulator 1.

At the conclusion of the operation, positions 6-9 of the program register contain the characteristic of accumulator 2 (00xx), and positions 4-5 contain the digits 01, from the insertion of that characteristic.

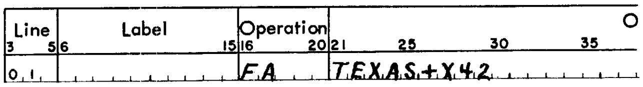
Note that accumulator 3 is used in this operation. Note too, that the contents of accumulator 2 have no effect on the result; it is reset prior to the operation.

The floating-decimal overflow and underflow indicators are set by the characteristic of the result in accumulator 1. If the operation attempts to develop a characteristic greater than 99 in accumulator 1, the *overflow* indicator is turned on and can be tested later by a floating-branch-overflow instruction. The characteristic in positions 0-1 of accumulator 1 is the units and tens digits of the resultant characteristic. (This is also true of the characteristic in accumulator 2, if it also exceeds 99.)

If the operation attempts to develop a characteristic less than 00, the underflow indicator is turned on. Both accumulators are reset to zero. If the characteristic of accumulator 1 is greater than 00, but that of accumulator 2 is less than 00, an underflow condition is not indicated but accumulator 2 is set to zero. Figure 183 shows results that occur when unnormalized factors are used.

Autocoder Example (Figure 193): Assume that TEXAS has been defined as word 2580. The assembled instruction is:

S01	*23*	*45*	*6789*
+74	42	00	2580



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		FA	TEXAS+X42

Figure 171

Floating Subtract
−74 **FS**

Machine Description: 8-digit number − 8-digit number = 16-digit result. The contents of the location addressed by positions 6-9 (indexable) are subtracted from the amount already in accumulator 1, and the result is normalized and developed in accumulators 1 and 2. Accumulator 2 is reset before the operation, and there is a difference of 8 in the resulting characteristics in the accumulators.

Instruction Format

S01 −74.

23 Indexing word for modifying the address is positions 6-9.

45 Not used.

6789 Address of the floating-decimal-point number to be subtracted from the number in accumulator 1.

Examples: To subtract, in floating-decimal-point notation, the contents of word 4714 from the amount already in accumulator 1:

S01	*23*	*45*	*6789*
−74	00	xx	4714

Contents of accumulator 1 before the operation: +91 10214889. Contents of word 4714: +88 71450024. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
+91 10143438	+83 97600000

To subtract, in floating-decimal-point notation, the contents of word 1300 indexed by positions 2-5 of IW 48, from the amount already in accumulator 1:

S01	*23*	*45*	*6789*
−74	48	xx	1300

Contents of accumulator 1 before the operation: +75 44162810. Contents of the data word: −75 79100026. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
+76 12326283	+68 60000000

To subtract, in floating-decimal-point notation, the contents of word 1907 from the amount already in accumulator 1:

S01	*23*	*45*	*6789*
−74	00	xx	1907

Contents of accumulator 1 before the operation: −32 28184000. Contents of word 1907: −39 10120000. Contents of accumulators 1 and 2 after the operation:

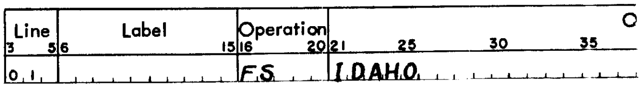
ACC 1	ACC 2
+39 10119997	+31 18160000

Timing: Same as floating-add.

Comments: The rules of algebra apply to floating add and subtract, just as for fixed-decimal add and subtract. Alpha factors cannot be used, however, with the floating-decimal operations. Figure 183 shows results that occur when unnormalized factors are used.

Autocoder Example (Figure 172): Assume that IDAHO has been defined as word 785. The assembled instruction is:

S01	*23*	*45*	*6789*
−74	00	00	0785



Line 3 5	Label 6 15	Operation 16 20	O 21 25 30 35
0 1		FS	IDAHO

Figure 172

Floating Add Double Precision
+76 **FAD**

Machine Description: 16-digit number + 8-digit number = 16-digit result. This code is the same as FLOATING ADD except that accumulator 2 is not set to zero before the operation starts. The program must have placed the values in the accumulators prior to this instruction.

The characteristic of the number from storage must be equal to, or greater than, the characteristic of the number in accumulator 1. The characteristic in accumulator 2 should be eight less than that of accumulator 1.

Instruction Format

S01 +76.

23 Indexing word for the address in positions 6-9.

45 Not used.

6789 Address of the floating-decimal-point number to be added to the number in accumulators 1 and 2 coupled.

Examples: To add, in floating-decimal-point notation, the contents of word 645 to the amount already in accumulators 1 and 2:

S01	*23*	*45*	*6789*
+76	00	xx	0645

Contents of accumulators 1 and 2 before the operation:

ACC 1	ACC 2
+51 23000740	+43 02232306

Contents of word 645: +55 71774321. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
+55 71776621	+47 07400223

To add, in floating-decimal-point notation, the contents of word 1707 to the amount already in accumulators 1 and 2:

S01	*23*	*45*	*6789*
+76	00	xx	1707

Contents of accumulators 1 and 2 before the operation:

ACC 1	ACC 2
+21 76714350	+13 14414388

Contents of word 1707: −22 11220000. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
−21 35485649	−13 85585620

Note that the accumulator signs are changed.

7074 Timing: Same as floating-add.

7070 Timing: 168-360 microseconds.

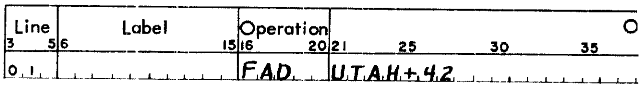
Comments: The characteristic of the word from storage must not be smaller than that of accumulator 1. It is advisable to compare the two factors that are to be added by a FAD instruction, in order to put the smaller one into the accumulators first if they are unequal.

The characteristic of accumulator 2 before the operation is ignored, as is the sign. Its sign is considered to be that of accumulator 1. The fraction in accumulator 1 before the operation should be normalized, but the fraction in accumulator 2 need not be. Note that if the modified characteristics are different, some of the low-order digits in accumulator 2 are lost to the operation. In the first example shown above, the four low-order digits of accumulator 2 (2306) are lost. The number of digits lost is determined by the difference in characteristic values of the storage word and accumulator 1.

Figure 183 shows results that occur when unnormalized factors are used.

Autocoder Example (Figure 173): Assume that UTAH has been defined as word 1700. The assembled instruction is:

S01	*23*	*45*	*6789*
+76	00	00	1742



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		FAD	UTAH+42

Figure 173

Floating Add Double Precision and Suppress Normalization

−76 **FADS**

Machine Description: 16-digit number + 8-digit number = 16-digit-or-less result. This code is the same as FLOATING ADD DOUBLE PRECISION except that the result in accumulators 1 and 2 is not normalized after the operation is completed.

Instruction Format: Same as FLOATING ADD DOUBLE PRECISION, except for the sign.

Example: To add, in floating-decimal-point notation, the contents of word 4582 to the amount already in accumulators 1 and 2, but not normalize the result:

S01	*23*	*45*	*6789*
−76	00	xx	4582

Contents of accumulators 1 and 2 before the operation:

ACC 1	ACC 2
+37 95791886	+29 00507500

Contents of word 4582: −38 12720011. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
−38 03140822	−30 39949250

7074 Timing: Same as floating-add.

7070 Timing: 144-300 microseconds.

Comments: The characteristic of the result (in accumulator 1) can never be less than that of the value from storage. If there is a carry 1, it is greater than the storage-word characteristic by one; otherwise, it is the same. The main purpose of this instruction is to align decimal positions of several numeric values, by using this code with a storage value consisting of a characteristic based on the desired decimal location, and a fraction of eight zeros. Take, for example, the two values +53 12345012 and +51 78754614. If brought to accumulator 1, and if accumulator 2 is set to zero, these values are:

ACC 1	ACC 2
+53 12345012	+00 00000000
+51 78754614	+00 00000000

Perform a FADS operation on each, using a storage value of +55 00000000, and the results are:

+55 00123450	+47 12000000
+55 00007875	+47 46140000

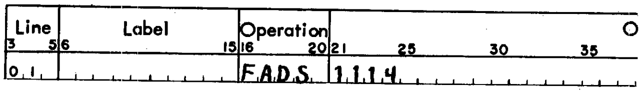
The two fractions are now decimally aligned. They can be stored and printed as:

1234501200
  78754614

The accumulators are not set to zero if the fraction in accumulator 1 is set to zero by this operation (using complement-add).

Autocoder Example (Figure 174): The actual address 1114 is used. The assembled instruction is:

S01	*23*	*45*	*6789*
−76	00	00	1114



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35	O
01		FADS	1114	

Figure 174

Floating Add Absolute

+77 **FAA**

Machine Description: 8-digit number + 8-digit number, considered plus = 16-digit result. This instruction is the same as FLOATING ADD, except that the sign of the storage word is considered plus. Thus, the sign of accumulator 1 determines whether the operation is true-add or complement-add.

Instruction Format

S01 +77.
23 Indexing word, for the address in positions 6-9.
45 Not used.
6789 Address of the floating-decimal-point number the contents of which, considered plus, are to be added to the number in accumulator 1.

Examples: To add, in floating-decimal-point notation, the absolute value of the contents of word 1585 to the amount already in accumulator 1 and develop the result in accumulators 1 and 2:

S01	*23*	*45*	*6789*
+77	00	xx	1585

Contents of accumulator 1 before the operation: +25 74026849. Contents of word 1585: −26 54000600. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
+26 61403284	+18 90000000

To add, in floating-decimal-point notation, the absolute value of the contents of word 4830 to the amount already in accumulator 1 and develop the result in accumulators 1 and 2:

S01	*23*	*45*	*6789*
+77	00	xx	4830

Contents of accumulator 1 before the operation: −43 56490066. Contents of word 4830: −47 29955000. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
+47 29949350	+39 99340000

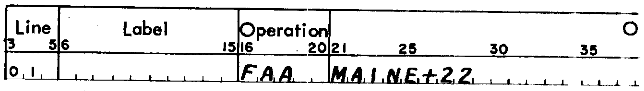
Timing: Same as FLOATING ADD.

Comments: This code has the same relation to FLOATING ADD as ADD ABSOLUTE TO ACCUMULATOR 1 has to the ADD TO ACCUMULATOR 1 instruction. Since one factor is considered plus and the operation is add, a plus sign in accumulator 1 causes a true-add operation, and a minus sign causes a complement-add operation.

The comments that apply to FLOATING ADD also apply to this code. Figure 183 shows results that occur when unnormalized factors are used.

Autocoder Example (Figure 175): Assume that MAINE has been defined as word 1650. The assembled instruction is:

S01	*23*	*45*	*6789*
+77	00	00	1672



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35	O
01		FAA	MAINE+22	

Figure 175

Floating Subtract Absolute

−77 **FSA**

Machine Description: 8-digit number − 8-digit number, considered plus, = 16-digit result. This instruction is the same as FLOATING ADD ABSOLUTE except that the value from storage is always subtracted.

Instruction Format

S01 −77.
23 Indexing word for modifying the address in positions 6-9.
45 Not used.
6789 Address of the floating-decimal-point number the contents of which, considered plus, are to be subtracted from the number in accumulator 1.

Example: To subtract, in floating-decimal-point notation, the absolute value of the contents of word 841 from the amount in accumulator 1 and develop the result in accumulators 1 and 2:

S01	*23*	*45*	*6789*
−77	00	xx	0841

Contents of accumulator 1 before the operation: +22 52507225. Contents of word 841: +24 70502575. Contents of accumulators 1 and 2 after the operation:

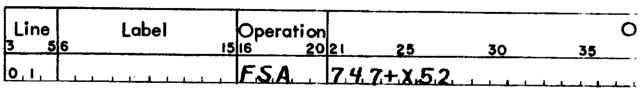
ACC 1	ACC 2
−24 69977502	−16 75000000

Timing: Same as FLOATING ADD.

Comments: Figure 183 shows results that occur when unnormalized factors are used.

Autocoder Example (Figure 176): Actual address 747 is used. The assembled instruction is:

S01	*23*	*45*	*6789*
−77	52	00	0747



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35	O
01		FSA	747+X52	

Figure 176

Floating Round
+71 **FR**

Machine Description: The fraction in accumulator 1 is rounded off, by adding a 5 to the high-order position of the fraction in accumulator 2, and carrying the result into accumulator 1 if that position contained a 5 or higher. If it carries all through accumulator 1 (if accumulator 1 stood at all 9's), the fraction of accumulator 1 becomes 10000000, and its characteristic is increased by one. Accumulator 2 is set to zero and made plus. The operation is always true-add; the signs of the accumulators are ignored.

Instruction Format

S01 +71.

23456789 Not used.

Example: To round off the floating-decimal-point value in accumulators 1 and 2:

S01 *23456789*
+71 xxxxxxxx

Contents of accumulators 1 and 2 before the operation:

ACC 1	ACC 2
+33 01234567	+25 82124719

After the operation:

+33 01234568	+00 0000000

Contents of accumulators 1 and 2 before the operation:

ACC 1	ACC 2
−91 75502300	−83 49990000

After the operation:

−91 75502300	+00 00000000

Contents of accumulators 1 and 2 before the operation:

ACC 1	ACC 2
−54 99999999	−46 50000000

After the operation:

−55 10000000	+00 00000000

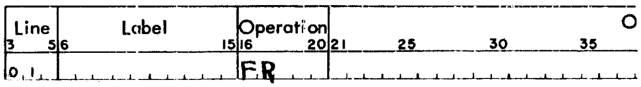
7074 Timing: 4 microseconds.

7070 Timing: 72 microseconds.

Comment: The characteristic of accumulator 2 is ignored, as are the sign and all but the high-order fraction digit. The characteristic of accumulator 1 is increased by one in this operation only if its fraction was all 9's and the high-order digit of the fraction in accumulator 2 was 5 or greater, before the operation. Although positions 2-9 are not used, positions 2-3 cannot designate an indexing word that is alpha. If it does, an error stop results. In all cases, indexing takes place if positions 2-3 are not 00.

Autocoder Example (Figure 177): Note that no operand is needed. The assembled instruction is:

S01 *23456789*
+71 00000000



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35 O
0 1		FR	

Figure 177

Floating Multiply
+73 **FM**

Machine Description: 8-digit multiplier × 8-digit multiplicand = 16-digit product. Prior to this instruction, the multiplier must be in accumulator 1. The location of the multiplicand is specified by positions 6-9 (indexable). The 16-digit product is developed in accumulators 1 and 2 and is normalized one position if the high-order digit is zero. The product is not affected by what was in accumulator 2 prior to the operation.

The characteristic of the result is determined by adding the characteristics of the two factors and subtracting either 50 or 51. If the product of the two fractions has 16 significant digits, subtract 50; if 15, subtract 51. (Since both original factors had to be normalized to start with, the product must be either 15 or 16 digits in size.)

The product of the original fractions becomes the fraction of the product in accumulators 1 and 2.

Instruction Format

S01 +73.

23 Indexing word for modifying the address in positions 6-9.

45 Not used.

6789 Address of the multiplicand.

Examples: To multiply, in floating-decimal-point notation, the contents of word 2680 by the number in accumulator 1:

S01 *23* *45* *6789*
+73 00 xx 2680

Contents of accumulator 1 before the operation: +39 21427685. Contents of word 2680: −45 10000000. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
−33 21427685	−25 00000000

Using the same instruction, with different factors: Contents of accumulator 1 before the operation: +55 22200000. Contents of word 2680: +55 50000000. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
+60 11100000	+52 00000000

Contents of accumulator 1 before the operation: +35 22200000. Contents of word 2680: +46 50000000. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
+31 11100000	+23 00000000

Contents of accumulator 1 before the operation: −24 12340000. Contents of word 2680: +72 20000000. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
−45 24680000	−37 00000000

7074 Timing: In microseconds:

Exact: $28 + 2N^1 + 4N^2 + 6N^3 + N^4$

Average: $28 + 3.4N + N^4$

N^1 = number of 1's, 2's, and 4's in multiplier

N^2 = number of 3's, 5's, 6's, and 8's in multiplier

N^3 = number of 7's and 9's in multiplier

N^4 = number of positions of post-normalization required

N = number of significant digits in multiplier

Note: The multiplier is the 8-digit fraction.

7070 Timing: In microseconds:

$48\ (N^1 + 2N^2 + 3N^3) + 4N^4 + 4N^5 + 408$

N^1 = number of 1, 2, 4 digits

N^2 = number of 3, 5, 6, 8 digits

N^3 = number of 7, 9 digits

N^4 = number of zeros

N^5 = number of zero groups (in multiplier)

If the total is not a multiple of 12, use the next higher multiple of 12. Consider multiplier-digit positions 0-1 to be a zero group. A zero group is a zero or two or more adjacent zeros.

Comments: Note that if the characteristics of both original factors are below 50, the characteristic of the result in accumulator 1 will be farther below 50. Similarly, if they are both above 50, the resulting characteristic in accumulator 1 will be farther above 50. Observe the first three operations under "Examples":

	CHAR. OF ONE FACTOR	CHAR. OF OTHER FACTOR	CHAR. OF RESULT IN ACC. 1
1st ex.	39(50−11)	45(50−5)	33(50−11−5,−1)
2d ex.	55(50+5)	55(50+5)	60(50+5+5)
3d ex.	35(50−15)	46(50−4)	31(50−15−4)

The differences from 50 are added to each other, and the resulting difference is the sum of these differences. If a product of 15 digits is developed, the characteristic of the result is one less, as in the first example.

Now observe the last example:

	CHAR. OF ONE FACTOR	CHAR. OF OTHER FACTOR	CHAR. OF RESULT IN ACC. 1
4th example	24	72	45(50 − 26 + 22, − 1)

The same principle works here. The resulting difference from 50 of −26 and +22 is −4. Because the product is 15 digits instead of 16, the characteristic of the result becomes 45 instead of 46.

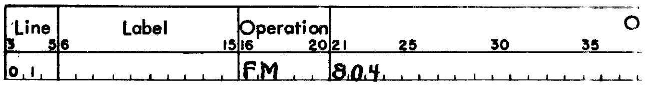
Note that the signs of the original factors do not affect the value of the characteristic of the result. They affect the sign of the result, however, according to the rules of algebra: plus × plus = plus, plus × minus = minus, and minus × minus = plus. (Alpha values cannot be used.)

The rules for floating-decimal-point overflow and underflow are the same as for FLOATING ADD. Note, however, that FLOATING MULTIPLY can more easily cause an overflow or underflow condition. If the sum of the characteristics of the two original factors is greater than 150, an overflow will result. Similarly, if their sum is less than 50, an underflow will result.

At completion of the operation, the program register has 01 in positions 4-5 and the characteristic of the product in accumulator 2 in positions 6-9 (00xx). The fraction of the product in accumulator 1 will always be normalized, even when unnormalized factors are used.

Autocoder Example (Figure 178): Actual address 804 is used. The assembled instruction is:

S01 23 45 6789
+73 00 00 0804



Line	Label	Operation	O
3 5	6 15	16 20	21 25 30 35
01		FM	804

Figure 178

Floating Divide

−73 **FD**

Machine Description: 8-digit dividend ÷ 8-digit divisor = 8-digit quotient with 8-digit remainder. Accumulator 2 is reset to zero and given the same sign as accumulator 1, which by a prior instruction contains the dividend. The fraction portion of the dividend is divided by the fraction portion of the divisor in the same manner as in fixed-point divide.

At the conclusion of the operation, the quotient is in accumulator 1 and the remainder is in accumulator 2, each with its corrected characteristic.

In floating-decimal division operations, the absolute value of the fraction of the divisor does not have to be smaller than the absolute value of the fraction of the dividend. In all cases involving normalized factors, regardless of the relative absolute values of divisor and dividend, a correct eight-digit quotient with correct characteristic is developed. Machine stops may occur when unnormalized factors are involved. (See Figure 183.)

The relative absolute values of the divisor and dividend fractions do, however, help to determine the characteristics of the quotient and remainder. If the dividend fraction is greater than, or equal to, the divisor fraction, the characteristic of the quotient is equal to the dividend characteristic minus the divisor characteristic, plus 51. The characteristic of the remainder is seven less than the characteristic of the dividend.

If the dividend fraction is smaller than the divisor fraction, the characteristic of the quotient is equal to the dividend characteristic minus the divisor characteristic, plus 50. The characteristic of the remainder is eight less than the characteristic of the dividend.

Instruction Format

S01 −73.
23 Indexing word for modifying the address in positions 6-9.
45 Not used.
6789 Address of the divisor.

Examples: To divide, in floating-decimal-point notation, the number in accumulator 1 by the number in word 1565:

S01	*23*	*45*	*6789*
−73	00	xx	1565

Contents of accumulator 1 (dividend) before the operation: +49 70000000. Contents of word 1565 (divisor): +28 30000000. Contents of accumulator 1 (quotient) after the operation: +72 23333333. Contents of accumulator 2 (remainder) after the operation: +42 10000000.

Using the same instruction, with different factors: Contents of accumulator 1 (dividend) before the operation: −41 22000000. Contents of word 1565 (divisor): −45 70000000. Contents of accumulator 1 (quotient) after the operation: +46 31428571. Contents of accumulator 2 (remainder) after the operation: −33 30000000.

7074 Timing: In microseconds:

Exact: $29 + 6N^1 + 8N^2$
Average: $26 + 6.8N$
N^1 = number of 1's, 2's, 3's, 8's, 9's, and 0's (except high order zeros) in quotient
N^2 = number of 4's, 5's, 6's and 7's in quotient
N = number of significant digits in quotient
Note: The quotient is the 8-digit fraction.

7070 Timing: The sum of the quotient digits is used in determining division timing. For example, the sum of the digits in a quotient of 51326 is: 5 + 1 + 3 + 2 + 6 = 17. The formula is: Total number of microseconds = 360 + 48 [8 + (sum of the quotient digits)]. The quotient is eight digits in size.

Comments: The characteristic of the quotient can be determined by the same means described under "Floating Multiply"; keep in mind that the dividend is equal to the divisor times the quotient. Observe the factors given under "Examples":

CHAR. OF DIVIDEND	CHAR. OF DIVISOR	CHAR. OF QUOTIENT	CHAR. OF REMAINDER
1st example			
49(50−1)	28(50−22)	72(50+21+1)	42(49−7)
2d example			
41(50−9)	45(50−5)	46(50−4)	33(41−8)

The rules of algebra concerning signs is the same for FLOATING-DIVIDE as for fixed-point divide. If the signs of the dividend and divisor are the same, the quotient is plus; if they are different, the quotient is minus. The remainder always has the sign of the dividend. Alpha factors cannot be used.

The rules for FLOATING-DECIMAL OVERFLOW AND UNDERFLOW in this operation are the same as for FLOATING ADD. A good rule-of-thumb is that the characteristics of the dividend and divisor should not be more than 50 apart (approximately). Specifically, they should meet these qualifications:

	DIVIDEND CH. GREATER	DIVISOR CH. GREATER
Dividend fraction equal or greater than divisor fraction	Should not be more than 48 greater.	Should not be more than 51 greater.
Dividend fraction smaller than divisor fraction	Should not be more than 49 greater.	Should not be more than 50 greater.

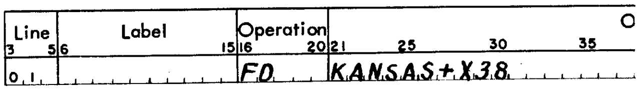
If it is necessary to normalize the quotient, these figures are reduced accordingly.

At completion of the operation, the program register has 01 in positions 4-5, and the characteristic of the remainder in positions 6-9 (00xx).

An unnormalized quotient will be generated if the dividend is unnormalized and if the absolute fraction of the divisor is greater than ten times the absolute fraction of the dividend. Figure 183 shows results that occur when unnormalized factors are used.

Autocoder Example (Figure 179): Assume that KANSAS has been previously defined as word 750. The assembled instruction is:

S01	*23*	*45*	*6789*
−73	38	00	0750



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		FD	KANSAS+X38

Figure 179

Floating Divide Double Precision —75 FDD

Machine Description: 16-digit dividend ÷ 8-digit divisor = 8-digit quotient with 8-digit remainder. Same as FLOATING DIVIDE except that accumulator 2 is not reset to zero initially and its fraction digits become the eight low-order digits of the dividend, making a 16-digit dividend.

Instruction Format: Same as FLOATING DIVIDE, except for the sign.

Examples: To divide, in floating-decimal-point notation, the number in accumulator 1 by the number in word 1635:

S01	*23*	*45*	*6789*
−75	00	xx	1635

Contents of accumulators 1 and 2 before the operation:

ACC 1	ACC 2
+75 12151822	+67 24273032

Contents of word 1635: +56 30000000. Contents of accumulators 1 and 2 after the operation:

ACC 1	ACC 2
+69 40506074	+67 04273032

Using the same instruction, with different factors:
Contents of accumulators 1 and 2 before the operation:

ACC 1	ACC 2
−27 66664446	−19 55550000

Contents of word 1635: +45 22220000. Contents of accumulators 1 and 2 after the operation:

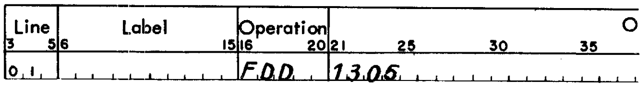
ACC 1	ACC 2
−33 30002001	−20 03335000

Timing: Same as FLOATING DIVIDE.

Comments: The comments for FLOATING DIVIDE also apply to this code. For this operation, accumulator 2 contains the eight low-order digits of a 16-digit fraction, whose characteristic is specified in positions 0-1 of accumulator 1. The sign and characteristic of accumulator 2 are ignored by the operation. If the sign is not the same, or if the characteristic is eight less than that of accumulator 1, the result is obtained as if they were.

Autocoder Example (Figure 180): Actual address 1305 is used. The assembled instruction is:

S01	*23*	*45*	*6789*
−75	00	00	1305



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35	O
01		FDD	1305	

Figure 180

Floating Branch Overflow

+70 **FBV**

Machine Description: The floating-decimal overflow indicator is tested. If it is on, the program branches to the location in positions 6-9 (indexable), and the indicator is turned off. If the indicator is not on, the next sequential instruction is taken.

Instruction Format

S01 +70.
23 Indexing word for modifying the branch address in positions 6-9.
45 Not used.
6789 Branch address, if the floating-decimal overflow indicator is on.

Example: To branch to location 2701 if the floating-decimal overflow indicator is on, and turn it off:

S01	*23*	*45*	*6789*
+70	00	xx	2701

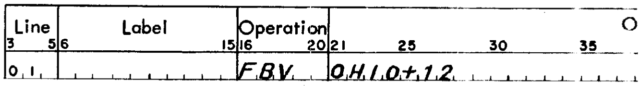
7074 Timing: 4 microseconds.

7070 Timing: 36 microseconds.

Comments: After this instruction, the floating-decimal overflow indicator is always off. Whenever this indicator is on, the exponent overflow light on the console is on.

Autocoder Example (Figure 181): Assume that OHIO has been defined as word 3800. The assembled instruction is:

S01	*23*	*45*	*6789*
+10	00	00	3812



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35	O
01		FBV	OHIO+12	

Figure 181

Floating Branch Underflow

−70 **FBU**

Machine Description: The floating-decimal underflow indicator is tested. If it is on, the program branches to the location in positions 6-9 (indexable), and the indicator is turned off. If the indicator is not on, the next sequential instruction is taken.

Instruction Format

S01 −70.
23 Indexing word for modifying the branch address in positions 6-9.
45 Not used.
6789 Branch address, if the floating-decimal underflow indicator is on.

Example: To branch to location 1842 if the floating-decimal underflow indicator is on, and turn it off:

S01	*23*	*45*	*6789*
−70	00	xx	1842

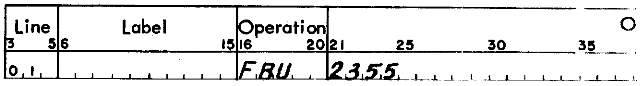
7074 Timing: 4 microseconds.

7070 Timing: 36 microseconds.

Comments: After this instruction, the floating-decimal underflow indicator is always off. Unlike the overflow indicator, this indicator has no light on the console to signal when it is on.

Autocoder Example (Figure 182): Actual address 2355 is used. The assembled instruction is:

S01	*23*	*45*	*6789*
−70	00	00	2355



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35	O
01		FBU	2355	

Figure 182

Table 1. Normalized Results - Floating Add (+74, FA)
A normalized result will be generated in accumulator 1 with the Floating Add instruction (also with FAA, FAD, FS, and FSA).

Smaller Factor	Larger Factor	Difference in Characteristics	Accumulator 1	Accumulator 2
38 87654321	55 00000002	17	48 20000000	00 00000000
38 87654321	55 00000001	17	48 10000000	00 00000000
38 87654321	55 00000000	17	00 00000000	00 00000000
38 87654321	54 00000002	16	47 20000000	00 00000000
38 87654321	54 00000001	16	47 10000000	00 00000000
38 87654321	54 00000000	16	00 00000000	00 00000000
38 87654321	53 00000002	15	46 20000000	38 80000000
38 87654321	53 00000001	15	46 10000000	38 80000000
38 87654321	53 00000000	15	38 80000000	30 00000000
38 87654321	52 00000002	14	45 20000008	37 70000000
38 87654321	52 00000001	14	45 10000008	37 70000000
38 87654321	52 00000000	14	38 87000000	30 00000000
38 87654321	51 00000002	13	44 20000087	36 60000000
38 87654321	51 00000001	13	44 10000087	36 60000000
38 87654321	51 00000000	13	38 87600000	30 00000000
38 87654321	50 00000002	12	43 20000876	35 50000000
38 87654321	50 00000001	12	43 10000876	35 50000000
38 87654321	50 00000000	12	38 87650000	30 00000000

Table 2. Unnormalized Quotient - Floating Divide (-73, FD)
An unnormalized quotient will be generated with the Floating Divide instruction (also with FDD) if the dividend is unnormalized and if the absolute fraction of the divisor is greater than ten times the absolute fraction of the dividend.

Dividend Divisor	Quotient	Comparison of Divisor with Ten Times the Dividend		
		Divisor	Condition	10(Dividend)
.00000002 .12345000	.00000016 (Unnormalized)	.12345000	greater than	.00000020
.00012345 .12345000	.001xxxxx (Unnormalized)	.12345000	greater than	.00123450
.00012345 .01234500	.01xxxxxx (Unnormalized)	.01234500	greater than	.00123450
.00012345 .00123460	.09999190 (Unnormalized)	.00123460	greater than	.00123450
.00012345 .00123450	.10000000 (Normalized)	.00123450	equal to	.00123450
.00012345 .00123440	.10000810 (Normalized)	.00123440	less than	.00123450
.00012345 .00012345	1.xxxxxxx (Normalized)	.00012345	less than	.00123450

Table 3. Machine Stop - Floating Divide (-73, FD)
There will be a machine stop with the Floating Divide instruction (also with FDD) if ten times the absolute fraction of the divisor is less than or equal to the absolute fraction of the dividend.

Dividend Divisor	Quotient	Comparison of Ten Times the Divisor with the Dividend		
		10 (Divisor)	Condition	Dividend
.12345670 .99999999	.123xxxxx	9.9999999	greater than	.12345670
.12345670 .12345670	1.0000000	1.2345670	greater than	.12345670
.12345670 .09999999	1.2345xxx	.99999999	greater than	.12345670
.12345670 .01234568	9.9xxxxxx	.12345680	greater than	.12345670
.12345670 .01234567	Machine Stop	.12345670	equal to	.12345670
.12345670 .00123456	Machine Stop	.01234560	less than	.12345670

Figure 183. Results of Floating-Decimal-Point Operations When Unnormalized Factors Are Used

Programming Summaries

This section contains the following charts and lists:

1. Functional Chart of 7070-7074 Operation Codes*
2. List of Operation Codes by Category
3. Core Storage and Register Addresses
4. Operation Codes That Allow Accumulator Addresses
5. Operation Codes That Use Field Definition
6. Store and Add-to-Storage Codes
7. Index of 7070-7074 Operation Codes by Autocoder Mnemonics*
8. Clearing a Specified Portion of Core Storage to Zeros
9. List of 7070 Instruction Execution Times
10. List of 7074 Instruction Execution Times
11. Compatibility of the 7070-7074.

*These include the page number containing the description of each code.

Functional Chart of 7070-7074 Operation Codes

This chart presents program instructions in operation-code order. The ten-digit instruction is divided into its four parts: Operation Code, Indexing Word, Control, and Address. Where necessary, the control or address portion is further divided into individual digit positions. Included in the chart are the Autocoder mnemonics and names of the operations, and a column for comments. The letter S and digits 0-9 at the top refer to the sign and digit positions of each instruction.

Pages containing descriptions of these operations are referenced, and this chart thus serves as an index by operation code.

S01			23	45		6789		
±0P	ABBR	NAME	IW	CL		ADDRESS	PAGE	COMMENTS
+00	HB	Halt and Branch	IW	Not used		Branch address	49	Unconditional branch
−00	HP	Halt and Proceed	IW	Not used		Not used	49	Next instruction from IC
+01	B	Branch	IW	Not used		Branch address	48	Next instruction from location in positions 6-9
−01	NOP	No Operation	IW	Not used		Not used	50	Next instruction from IC
+02	BLX	Branch and Load Location in Index Word	IW	Operand Index Word		Branch address	48	Contents of IC are stored in index word, positions 2-5
+03	CD	Compare Storage to Digit	IW	Value compared with	Position to be compared	Address of data	46	Turns on high, low, or equal indicator
−03		Sign Control	IW	Sign value 3, 6 or 9 Six operations:	Operation 0. Compare 1. Make	Address of data for operations 0-1 Branch address for operation 4	47	Compare operations turn on high, low, or equal indicator
	CSA			Compare sign to alpha				
	CSM			Compare sign to minus				
	CSP			Compare sign to plus				
	MSA			Make sign alpha				
	MSM			Make sign minus				
	MSP			Make sign plus				
	SMSC				2. Sign change sense			
	HMSC				3. Sign change stop			
	BSC				4. Sign change test			
+04	BASS	Branch if Additional Storage Switch is On	IW	0-Not used		Part of branch address	16	Additional Storage is an optional feature for the 7074
	ASSN	Additional Storage Switch On		1-Not used		Not used		
	ASSF	Additional Storage Switch Off		2-Not used		Not used		
+10	BZ1	Branch if Zero in Accumulator 1	IW	Not used		Branch address	38	Sign is ignored
−10	BM1	Branch if Minus in Accumulator 1	IW	Not used		Branch address	39	Contents are ignored
+11	BV1	Branch if Overflow in Accumulator 1	IW	Not used		Branch address	43	If overflow indicator is on, branch and turn it off
−11	ZST1	Zero Storage and Store Accumulator 1	IW	Field Definition		Address of data	28	
+12	ST1	Store Accumulator 1	IW	Field Definition		Address of data	29	Field overflow and sign change indicators might be set
−12	STD1	Store Digits from Accumulator 1 and Ignore Sign	IW	Field Definition		Address of data	29	Field overflow indicator might be set
+13	ZA1	Zero Accumulator 1 and Add	IW	Field Definition		Address of data	20	
−13	ZS1	Zero Accumulator 1 and Subtract	IW	Field Definition		Address of data	21	
+14	A1	Add to Accumulator 1	IW	Field Definition		Address of data	22	
−14	S1	Subtract from Accumulator 1	IW	Field Definition		Address of data	23	
+15	C1	Compare Accumulator 1 to Storage	IW	Field Definition		Address of data	44	Sets high, low, or equal indicator
−15	CA	Compare Absolute in Accumulator 1 to Absolute in Storage	IW	Field Definition		Address of data	46	Sets high, low, or equal indicator
+16	ZAA	Zero Accumulator 1 and Add Absolute	IW	Field Definition		Address of data	26	
−16	ZSA	Zero Accumulator 1 and Subtract Absolute	IW	Field Definition		Address of data	27	
+17	AA	Add Absolute to Accumulator 1	IW	Field Definition		Address of data	27	
−17	SA	Subtract Absolute from Accumulator 1	IW	Field Definition		Address of data	27	
+18	AS1	Add to Storage from Accumulator 1	IW	Field Definition		Address of data	30	Field overflow and/or sign change indicators might be set
−18	SS1	Subtract Accumulator 1 from Storage	IW	Field Definition		Address of data	31	Field overflow and/or sign change indicators might be set

S01 ±0P	ABBR	NAME	23 IW	45 CL	6789 ADDRESS	PAGE	COMMENTS
+19	AAS1	Add to Absolute Storage from Accumulator 1	IW	Field Definition	Address of data	32	Field overflow indicator might be set
+20	BZ2	Branch if Zero in Accumulator 2	IW	Not used	Branch address	38	Sign is ignored
−20	BM2	Branch if Minus in Accumulator 2	IW	Not used	Branch address	39	Contents are ignored
+21	BV2	Branch if Overflow in Accumulator 2	IW	Not used	Branch address	43	If overflow indicator is on, branch and turn it off
−21	ZST2	Zero Storage and Store Accumulator 2	IW	Field Definition	Address of data	28	
+22	ST2	Store Accumulator 2	IW	Field Definition	Address of data	29	Field overflow and/or sign change indicators might be set
−22	STD2	Store Digits from Accumulator 2 and Ignore Sign	IW	Field Definition	Address of data	29	Field overflow indicator might be set
+23	ZA2	Zero Accumulator 2 and Add	IW	Field Definition	Address of data	20	
−23	ZS2	Zero Accumulator 2 and Subtract	IW	Field Definition	Address of data	21	
+24	A2	Add to Accumulator 2	IW	Field Definition	Address of data	22	
−24	S2	Subtract from Accumulator 2	IW	Field Definition	Address of data	23	
+25	C2	Compare Accumulator 2 to Storage	IW	Field Definition	Address of data	44	Sets high, low, or equal indicator
+28	AS2	Add to Storage from Accumulator 2	IW	Field Definition	Address of data	30	Field overflow and/or sign change indicators might be set
−28	SS2	Subtract Accumulator 2 from Storage	IW	Field Definition	Address of data	31	Field overflow and/or sign change indicators might be set
+29	AAS2	Add to Absolute Storage from Accumulator 2	IW	Field Definition	Address of data	32	Field overflow indicator might be set
+30	BZ3	Branch if Zero in Accumulator 3	IW	Not used	Branch address	38	Sign is ignored
−30	BM3	Branch if Minus in Accumulator 3	IW	Not used	Branch address	39	Contents are ignored
+31	BV3	Branch if Overflow in Accumulator 3	IW	Not used	Branch address	43	If overflow indicator is on, branch and turn it off
−31	ZST3	Zero Storage and Store Accumulator 3	IW	Field Definition	Address of data	28	
+32	ST3	Store Accumulator 3	IW	Field Definition	Address of data	29	Field overflow and/or sign change indicator might be set
−32	STD3	Store Digits from Accumulator 3 and Ignore Sign	IW	Field Definition	Address of data	29	Field overflow indicator might be set
+33	ZA3	Zero Accumulator 3 and Add	IW	Field Definition	Address of data	20	
−33	ZS3	Zero Accumulator 3 and Subtract	IW	Field Definition	Address of data	21	
+34	A3	Add to Accumulator 3	IW	Field Definition	Address of data	22	
−34	S3	Subtract from Accumulator 3	IW	Field Definition	Address of data	23	
+35	C3	Compare Accumulator 3 to Storage	IW	Field Definition	Address of data	44	Set high, low, or equal indicator
+38	AS3	Add to Storage from Accumulator 3	IW	Field Definition	Address of data	30	Field overflow and/or sign change indicator might be set
−38	SS3	Subtract Accumulator 3 from Storage	IW	Field Definition	Address of data	31	Field overflow and/or sign change indicator might be set
+39	AAS3	Add to Absolute Storage from Accumulator 3	IW	Field Definition	Address of data	32	Field overflow indicator might be set
+40	BL	Branch if Low	IW	Not used	Branch address	42	Tests the low indicator
−40	BH	Branch if High	IW	Not used	Branch address	42	Tests the high indicator
+41	BFV SMFV HMFV	Field Overflow Control	IW	Not used Operation: 0. Test 1. Sense 2. Stop	Branch address in Operation 0 Not used for operations 1-2	43	Test operation tests field overflow indicator
−41	BE	Branch if Equal	IW	Not used	Branch address	43	Tests the equal indicator
−43	BCX	Branch Compared Index Word	IW	Operand index word	Branch address	56	
+44	BXN	Branch if Index Word Indexing Portion is Non-zero	IW	Operand index word	Branch address	56	Sign and the other six digit positions are ignored
−44	BXM	Branch if Index Word is Minus	IW	Operand index word	Branch address	56	Contents are ignored
+45	XL	Index Word Load	IW	Operand index word	Address of data	51	
−45	XU	Index Word Unload	IW	Operand index word	Address of data	52	

S01 ±0P	ABBR	NAME	23 IW	45 CL	6789 ADDRESS	PAGE	COMMENTS
+46	XZA	Index Word Zero and Add to Indexing Portion	IW	Operand index word	Four-digit factor	52	Sign is set to plus and the other six digit positions are unchanged
−46	XZS	Index Word Zero and Subtract from Indexing Portion	IW	Operand index word	Four-digit factor	53	Sign is set to minus and the other six digit positions are unchanged
+47	XA	Index Word Add to Indexing Portion	IW	Operand index word	Four-digit factor	53	Sign may change. Other positions are unchanged
−47	XS	Index Word Subtract from Indexing Portion	IW	Operand index word	Four-digit factor	54	Sign may change. Other positions are unchanged
+48	XSN	Index Word Set Non-indexing Portion	IW	Operand index word	Four-digit factor	55	Sign and other six digit positions are unchanged
−48	XLIN	Index Word Load and Interchange	IW	Operand index word	Address of data	52	Positions 2-5 and 6-9 are interchanged
+49	BIX	Branch Incremented Index Word	IW	Operand index word	Branch address	57	Branch if incremented indexing portion is not greater than non-indexing portion.
−49	BDX	Branch Decremented Index Word	IW	Operand index word	Branch address	58	Branch if decremented indexing portion is not brought to zero or beyond zero by non-indexing value
+50	SR# SRR# SL# SLC#	Shift Control	IW	Index word for shift and count	Accu-mulator (1-3) Opn * Length of shift (00-10)	32	*Operation: 0—Shift right 1—Shift right and round 2—Shift left 3—Shift left and count
−50	SR SRR SL SLC SRS SLS SRS SLS	Coupled Shift Control	IW	Index word for shift and count	Digit Position for split, 0 for normal Opn * Length of shift (00-20)	34	*Operation, normal: 0—Shift right 1—Shift right and round 2—Shift left 3—Shift left and count Operation, split: 4—Shift right from point Acc 1 5—Shift left from point Acc 1 6—Shift right from point Acc 2 7—Shift left from point Acc 2
+51	BAS BCB BDCB	Branch on Alteration Switch or Channel Busy	IW	Operation: Switch 1-4 7604 Chan 1-4 7907 Chan 1-4	Branch address 0. Alt sw 1. 7604 Ch bsy 2. 7907 Ch bsy	40	
+53	M	Multiply	IW	Field definition of multiplier	Address of multiplier	24	
−53	D	Divide	IW	Field definition of divisor	Address of divisor	25	
+54	QR QW	Inquiry Control	IW	Inquiry ctrl group (1 or 2) Operation: 0. Request 1. Reply	Address of first record definition word	99	
+55	PC PC1	729 Priority Control 7907 Priority Control	IW	00 01	Address of priority mask	108	0—allow; 1—mask
+56	ENA	Edit Numeric to Alphameric	IW	Index word—locates first numeric word	Address of first record definition word	62	RDW defines alpha area
−56	ENS	Edit Numeric to Alphameric with Sign Control	IW	Index word—locates first numeric word	Address of first record definition word	63	RDW defines alpha area
+57	ENB	Edit Numeric to Alphameric with Blank Insertion	IW	Index word—locates first numeric word	Address of first record definition word	64	RDW defines alpha area
−57	EAN	Edit Alphameric to Numeric	IW	Index word—locates first word to be filled with numeric data	Address of first record definition word	64	RDW defines alpha area
+60	BAL BUL BUL BQL	Stacking Latch Test	IW	Stacking latch: 00 Any stacking latch 01 Unit Record A 02 Unit Record B 03 Inquiry Control 1	Branch address	110	

S01 ±0P	ABBR	NAME	23 IW	45 CL	6789 ADDRESS	PAGE	COMMENTS
	BQL			04 Inquiry Control 2			
	BTL			10-19 729 Tape units 0-9 on 7604 Channel 1			
	BTL			20-29 729 Tape units 0-9 on 7604 Channel 2			
	BTL			30-39 729 Tape units 0-9 on 7604 Channel 3			
	BTL			40-49 729 Tape units 0-9 on 7604 Channel 4			
	BDCL			81-84 7907 Channel 1-4			
	BDCA			91-94 7907 Channel 1-4			
+61		Electronic Switch Control	IW	Operation: Switch Number	Branch address	40	Controls the ten switches in word 0101
	BES			0-Test			
	ESN			1-Turn on (0-9)			
	ESF			2-Turn off			
	BSN			3-Test and turn on			
	BSF			4-Test and turn off			
−61		Stacking Latch Set On	IW	Stacking latch:	Not used	111	
	ULN			01 Unit Record A			
	ULN			02 Unit Record B			
	QLN			03 Inquiry Control 1			
	QLN			04 Inquiry Control 2			
	TLN			10-19 729 Tape units 0-9 on 7604 Channel 1			
	TLN			20-29 729 Tape units 0-9 on 7604 Channel 2			
	TLN			30-39 729 Tape units 0-9 on 7604 Channel 3			
	TLN			40-49 729 Tape units 0-9 on 7604 Channel 4			
	DCLN			81-84 7907 Channel 1-4			
	DCAN			91-94 7907 Channel 1-4			
+62		Electronic Switch Control		Same as +61, except for the ten switches that are controlled		40	Controls the ten switches in word 0102
−62		Stacking Latch Reset Off	IW	Stacking latch:	Not used	112	
	ULF			01 Unit Record A			
	ULF			02 Unit Record B			
	QLF			03 Inquiry Control 1			
	QLF			04 Inquiry Control 2			
	TLF			10-19 729 Tape units 0-9 on 7604 Channel 1			
	TLF			20-29 729 Tape units 0-9 on 7604 Channel 2			
	TLF			30-39 729 Tape units 0-9 on 7604 Channel 3			
	TLF			40-49 729 Tape units 0-9 on 7604 Channel 4			
	DCLF			81-84 7907 Channel 1-4			
	DCAF			91-94 7907 Channel 1-4			
+63		Electronic Switch Control		Same as +61, except for the ten switches that are controlled		40	Controls the ten switches in word 0103
+64	PR	Priority Release	IW	Not used	Branch address	112	Branch address is usally 0097

S01 ±0P	ABBR	NAME	23 IW	45 CL	6789 ADDRESS	PAGE	COMMENTS
+65	RS	Record Scatter	IW	Index word—locates first transmitting word	Address of first record definition word	61	RDW defines receiving area
−65	RG	Record Gather	IW	Index word—locates first receiving word	Address of first record definition word	62	RDW defines transmitting area
+66	LL	Lookup Lowest	IW	Field Definition	Address of first record definition word	66	
+67	LE	Lookup Equal Only	IW	Field Definition	Address of first record definition word	68	Index Word 98: positions 2-5, found address
+68	LEH	Lookup Equal or High	IW	Field Definition	Address of first record definition word	68	positions 6-9, increment
+69		Unit Record Control	IW	Syn. Oper:	Address of first record definition word		
	US			(1-3) 0—Set PES		89	
	UR			(1-3) 1—Read (7500)		88	
	UR			(4) 1—Read (7501)		88	
	UW/UP			(1-3) 2—Write or Punch		89	
	UWIV/ UPIV			(1-3) 3—Write or Punch invalid		90	
	TYP			(0) 4—Type		90	
	ITS			Always digit "9" 1—Interval Timer Store		19	
	ITZ			Always digit "9" 0—Interval Timer Zero	Not used	19	
+70	FBV	Floating Branch Overflow	IW	Not used	Branch address	129	Tests the floating decimal overflow indicator
−70	FBU	Floating Branch Underflow	IW	Not used	Branch address	129	Tests the floating decimal underflow indicator
+71	FR	Floating Round	IW	Not used	Not used	126	
+73	FM	Floating Multiply	IW	Not used	Address of multiplicand	126	
−73	FD	Floating Divide	IW	Not used	Address of divisor	127	
+74	FA	Floating Add	IW	Not used	Address of data	122	Accumulator 2 cleared at start
−74	FS	Floating Subtract	IW	Not used	Address of data	123	Accumulator 2 cleared at start
+75	FZA	Floating Zero and Add	IW	Not used	Address of data	121	
−75	FDD	Floating Divide Double Precision	IW	Not used	Address of divisor	128	
+76	FAD	Floating Add Double Precision	IW	Not used	Address of data	123	Accumulator 2 not cleared at start
−76	FADS	Floating Add Double Precision Suppress Normalization	IW	Not used	Address of data	124	
+77	FAA	Floating Add Absolute	IW	Not used	Address of data	125	
−77	FSA	Floating Subtract Absolute	IW	Not used	Address of data	125	
±81		Tape Control 7604 Ch 1	IW	729 Tape Unit (0-9) Oper:	If position 5 is not zero, address of first record definition word		
	(P)TR			1—Read		76	+Priority
	(P)TRR			2—Read per RM		77	−No priority
	(P)TW			3—Write		77	Code 81 means Channel 1
	(P)TWR			4—Write per RM		78	
	(P)TWZ			5—Write with zero elimination		78	
	(P)TWC			6—Write per RM and zero elimination		78	
	(P)TSF			7—Segment forward space per count		79	
	(P)TSB			8—Segment backspace per count		79	

S01 ±0P	ABBR	NAME	23 IW	45 CL	6789 ADDRESS	PAGE	COMMENTS
	(P)TRA			9—Read all alpha		80	
				0—Operation in position 9	If pos. 5 is zero, operation is in pos. 9:		
	TSEL				0—no-op select	80	
	(P)TM				1—write tape mark	80	
	TRW				2—rewind	81	
	TRU				3—rewind unload	81	
	TRB				4—backspace	81	
	(P)TSM				5—write tape segment mark	82	
	TSK				6—skip	82	
	TEF				7—turn off end of file	82	
	TSLD				8—set lower density	82	
	TSHD				9—set high density	83	
±82,±83,±84		Tape Control		Same as ± 81, except other Tape Channels are used			
±88	TRN	Read Binary 729 Tape	IW	01-91 729 Tape Unit	Address of first (or only) RDW	18	7074 Optional feature
±93		Select 7907 Data Ch 1	IW	1 Packing Format:	Command address	140	
	(P)DCP			1-Packed decimal digits			
	(P)DCUA			2-Unpacked alpha start			
	(P)DCUR			3-Unpacked alpha start with RMC			
	(P)DCPR			4-Packed with RMC			
±94,±96,±97	(P)DCU	Same as ±93 except for 7907 Channels 2, 3, and 4		6-Unpacked numeric start			

List of Operation Codes by Category

1. Arithmetic: All instructions that involve any type of arithmetic operation—adding, subtracting, multiplying, or dividing, including reset-add and subtract.

2. Shift: All instructions that include, or may include, the shifting of the number in an accumulator.

3. Branch: Instructions that may cause the program to take its next instruction from the address in positions 6-9 (indexable).

4. Compare: All instructions that turn on either the *high, equal,* or *low* indicator as the result of a test.

5. Data to Core Storage--One Word or Less: All operations that bring information to a single storage word or part of a word, as specified by field definition (positions 4 and 5). These operations all cause previous data in a storage word to be replaced by new data. (Codes to load index words are not included.)

6. Data to Core Storage—One Word or More: All operations that bring information to one or more storage words, under control of record-definition words (addressed by positions 6-9). These operations cause data in core storage to be replaced by new data.

7. Index Words: All operations that use index words for other than indexing, specify the operand index word in positions 4 and 5.

8. Magnetic Tape: Instructions that involve the use of a tape channel or unit.

9. Data Channel: Instructions that involve the use of the 7907 Data Channel.

10. Unit Record: All instructions that involve the use of a card reader, punch, printer, the console typewriter, or the input-output synchronizers.

11. Core-to-Core Block Transmission: The codes that move one or more words of core storage to other locations in core storage, under control of record-definition words.

12. Table Lookup: The three table lookup codes.

13. Inquiry: The codes that involve the processing of inquiry requests and formulation of replies.

14. Priority: All of the instructions that have anything to do with Priority Processing.

15. Floating Decimal-Point: The instructions made available by the optional floating-decimal-point arithmetic feature.

16. Miscellaneous: All instructions that do not fully come under the other categories.

Each list is given in sequence by Autocoder mnemonics.

Arithmetic

+14,+24,+34	A#	Add to Accumulator #
+17	AA	Add Absolute to Accumulator 1
+18,+28,+38	AS#	Add to Storage from Accumulator #
+19,+29,+39	AAS#	Add to Absolute Storage from Accumulator #
+49	BIX	Branch Incremented Index Word
−49	BDX	Branch Decremented Index Word
−53	D	Divide
+74	FA	Floating Add
−74	FS	Floating Subtract
+73	FM	Floating Multiply
−73	FD	Floating Divide
+77	FAA	Floating Add Absolute
−77	FSA	Floating Subtract Absolute
+75	FZA	Floating Zero and Add
+76	FAD	Floating Add Double Precision
−75	FDD	Floating Divide Double Precision
−76	FADS	Floating Add Double Precision and Suppress Normalization
+53	M	Multiply
−14,−24,−34	S#	Subtract from Accumulator #
−17	SA	Subtract Absolute from Accumulator 1
−18,−28,−38	SS#	Subtract Accumulator # from Storage
+50	SRR#	Shift Right and Round Accumulator #
+50	SLC#	Shift Left and Count Accumulator #
−50	SR	Shift Right and Round Coupled
−50	SLC	Shift Left and Count Coupled
+47	XA	Index Word Add to Indexing Portion
−47	XS	Index Word Subtract from Indexing Portion
+46	XZA	Index Word Zero and Add to Indexing Portion
−46	XZS	Index Word Zero and Subtract from Indexing Portion
+13,+23,+33	ZA#	Zero Accumulator # and Add
−13,−23,−33	ZS#	Zero Accumulator # and Subtract
+16	ZAA	Zero Accumulator 1 and Add Absolute
−16	ZSA	Zero Accumulator 1 and Subtract Absolute

Shift

+71	FR	Floating Round
+75	FZA	Floating Zero and Add
+50	SR#	Shift Right Accumulator #
+50	SRR#	Shift Right and Round Accumulator #
+50	SL#	Shift Left Accumulator #
+50	SLC#	Shift Left and Count Accumulator #
−50	SR	Shift Right Coupled
−50	SRR	Shift Right and Round Coupled
−50	SL	Shift Left Coupled
−50	SLC	Shift Left and Count Coupled
−50	SRS	Shift Right Split
−50	SLS	Shift Left Split

Branch

+01	B	Branch
+02	BLX	Branch and Load Location in Index Word
−10, −20, −30	BM#	Branch if Minus in Accumulator #
+10, +20, +30	BZ#	Branch if Zero in Accumulator #
+11, +21, +31	BV#	Branch if Overflow in Accumulator #
+41	BFV	Branch if Field Overflow
−03	BSC	Branch if Sign Change
−40	BH	Branch if High
−41	BE	Branch if Equal
+40	BL	Branch if Low
+51	BAS	Branch if Alteration Switch is On
+61, +62, +63	BES	Branch if Electronic Switch is On
+61, +62, +63	BSF	Branch if Electronic Switch is On and Set Off if On
+61, +62, +63	BSN	Branch if Electronic Switch is On and Set On if Off
+51	BCB	Branch if 7604 Channel is Busy
+60	BAL	Branch if Any Stacking Latch is On
+60	BQL	Branch if Inquiry Control Latch is On
+60	BTL	Branch if 729 Tape Latch is On
+60	BUL	Branch if Unit Record Latch is On
−44	BXM	Branch if Index Word is Minus
+44	BXN	Branch if Index Word is Nonzero
−43	BCX	Branch Compared Index Word
+49	BIX	Branch Incremented Index Word
−49	BDX	Branch Decremented Index Word
+70	FBV	Floating Branch Overflow
−70	FBU	Floating Branch Underflow
+00	HB	Halt and Branch
+04	BASS	Branch if Additional Storage Switch is On
+51	BDCB	Branch if 7907 Data Channel is Busy
+60	BDCA	Branch if 7907 Data Channel Attention Latch is On
+60	BDCL	Branch if 7907 Data Channel Normal Latch is On

Compare

+15, +25, +35	C#	Compare Accumulator # to Storage
−15	CA	Compare Absolute in Accumulator 1 to Absolute in Storage
+03	CD	Compare Storage to Digit
−03	CSA	Compare Sign to Alpha
−03	CSP	Compare Sign to Plus
−03	CSM	Compare Sign to Minus

Data to Core Storage—One Word or Less (Field Definition)

+18, +28, +38	AS#	Add to Storage from Accumulator #
+19, +29, +39	AAS#	Add to Absolute Storage from Accumulator #
−18, −28, −38	SS#	Subtract Accumulator # from Storage
+12, +22, +32	ST#	Store Accumulator #
−12, −22, −32	STD#	Store Digits from Accumulator # and Ignore Sign
−45	XU	Index Word Unload
−11, −21, −31	ZST#	Zero Storage and Store Accumulator #

Data to Core Storage—One Word or More (Block Transmission)

−57	EAN	Edit Alphameric to Numeric
+56	ENA	Edit Numeric to Alphameric
+57	ENB	Edit Numeric to Alphameric with Blank Insertion
−56	ENS	Edit Numeric to Alphameric with Sign Control
+81, +82, +83, +84	PTR	Priority 729 Tape Read
+81, +82, +83, +84	PTRR	Priority 729 Tape Read Per Record Mark Control
+54	QR	Inquiry Read
−65	RG	Record Gather
+65	RS	Record Scatter
−81, −82, −83, −84	TR	729 Tape Read
−81, −82, −83, −84	TRR	729 Tape Read Per Record Mark Control
+69	UR	Unit Record Read

Index Words

+02	BLX	Branch and Load Location in Index Word
−44	BXM	Branch if Index Word is Minus
+44	BXN	Branch if Index Word is Nonzero
−43	BCX	Branch Compared Index Word
+49	BIX	Branch Incremented Index Word
−49	BDX	Branch Decremented Index Word
−57	EAN	Edit Alphameric to Numeric
+56	ENA	Edit Numeric to Alphameric
+57	ENB	Edit Numeric to Alphameric with Blank Insertion
−56	ENS	Edit Numeric to Alphameric with Sign Control
−65	RG	Record Gather
+65	RS	Record Scatter
+50	SLC#	Shift Left and Count Accumulator #
−50	SLC	Shift Left and Count Coupled
+45	XL	Index Word Load
−48	XLIN	Index Word Load and Interchange
−45	XU	Index Word Unload
+47	XA	Index Word Add to Indexing Portion
−47	XS	Index Word Subtract from Indexing Portion
+46	XZA	Index Word Zero and Add to Indexing Portion
−46	XZS	Index Word Zero and Subtract from Indexing Portion
+48	XSN	Index Word Set Non-indexing Portion

729 Magnetic Tape

+51	BCB	Branch if 7604 Channel is Busy
+60	BTL	Branch if 729 Tape Latch is On
+55	PC	Priority Control
+81,+82,+83,+84	PTR	Priority 729 Tape Read
+81,+82,+83,+84	PTRA	Priority 729 Tape Read All Alpha
+81,+82,+83,+84	PTRR	Priority 729 Tape Read per Record Mark Control
+81,+82,+83,+84	PTW	Priority 729 Tape Write
+81,+82,+83,+84	PTWZ	Priority 729 Tape Write with Zero Elimination
+81,+82,+83,+84	PTWR	Priority 729 Tape Write per Record Mark Control
+81,+82,+83,+84	PTWC	Priority 729 Tape Write with Zero Elimination and per Record Mark Control
+81,+82,+83,+84	PTSB	Priority 729 Tape Segment Backward Space
+81,+82,+83,+84	PTSF	Priority 729 Tape Segment Forward Space
±81,±82,±83,±84	PTM	Priority 729 Tape Mark Write
±81,±82,±83,±84	PTSM	Priority 729 Tape Segment Mark Write
−81,−82,−83,−84	TR	729 Tape Read
−81,−82,−83,−84	TRA	729 Tape Read All Alpha
−81,−82,−83,−84	TRR	729 Tape Read per Record Mark Control
−81,−82,−83,−84	TW	729 Tape Write
−81,−82,−83,−84	TWR	729 Tape Write per Record Mark Control
−81,−82,−83,−84	TWZ	729 Tape Write with Zero Elimination
−81,−82,−83,−84	TWC	729 Tape Write with Zero Elimination and per Record Mark Control Combined
±81,±82,±83, ±84	TRB	729 Tape Record Backspace
−81,−82,−83,−84	TSB	729 Tape Segment Backward Space
−81,−82,−83,−84	TSF	729 Tape Segment Forward Space
±81,±82,±83, ±84	TRW	729 Tape Rewind
±81,±82,±83, ±84	TM	729 Tape Mark Write
±81,±82,±83, ±84	TSM	729 Tape Segment Mark Write
±81,±82,±83, ±84	TEF	729 Tape End of File Off
±81,±82,±83, ±84	TSK	729 Tape Skip
±81,±82,±83, ±84	TSEL	729 Tape No-op Select
±81,±82,±83, ±84	TRU	729 Tape Rewind Unload
±81,±82,±83, ±84	TSHD	729 Tape Set High Density
±81,±82,±83, ±84	TSLD	729 Tape Set Low Density
−61	TLN	729 Tape Latch Set On
−62	TLF	729 Tape Latch Set Off

7907 Data Channel Instructions

+51	BDCB	Branch if Data Channel Busy
+55	PC1	Priority Control Data Channel
+60	BDCA	Branch if Data Channel Attention Latch is On
+60	BDCL	Branch if Data Channel Normal Latch is On
−61	DCAN	Data Channel Attention Latch On
−61	DCLN	Data Channel Normal Latch On
−62	DCAF	Data Channel Attention Latch Off
−62	DCLF	Data Channel Normal Latch Off
±93,±94,±96,±97	PDCP	Data Channel Select Packed
	PDCUA	Data Channel Select Unpacked Alpha Start
	PDCUR	Data Channel Select Unpacked Alpha Start with RMC
	PDCPR	Data Channel Select Packed with RMC
	PDCU	Data Channel Select Unpacked Numeric Start

Unit Record

+60	BUL	Branch if Unit Record Latch is On
+69	TYP	Type
+55	PC	Priority Control
+69	UR	Unit Record Read I/O
+69	UP	Unit Record Punch I/O
+69	UW	Unit Record Write I/O
+69	UPIV	Unit Record Punch Invalid I/O
+69	UWIV	Unit Record Write Invalid I/O
−61	ULN	Unit Record Latch Set On
−62	ULF	Unit Record Latch Set Off
+69	US	Unit Record Signal I/O

Core-to-Core Block Transmission

−57	EAN	Edit Alphameric to Numeric
+56	ENA	Edit Numeric to Alphameric
+57	ENB	Edit Numeric to Alphameric with Blank Insertion
−56	ENS	Edit Numeric to Alphameric with Sign Control
−65	RG	Record Gather
+65	RS	Record Scatter

Table Lookup

+67	LE	Lookup Equal Only
+66	LL	Lookup Lowest
+68	LEH	Lookup Equal or High

Inquiry

+60	BQL	Branch if Inquiry Control Latch is On
+55	PC	Priority Control
+54	QR	Inquiry Read
+54	QW	Inquiry Write
−61	QLN	Inquiry Control Latch Set On
−62	QLF	Inquiry Control Latch Set Off

Priority

+60	BAL	Branch if Any Stacking Latch is On
+60	BQL	Branch if Inquiry Control Latch is On
+60	BTL	Branch if 729 Tape Latch is On
+60	BUL	Branch if Unit Record Latch is On
+55	PC	Priority Control
+64	PR	Priority Release

+81,+82,+83,+84	PTR	Priority 729 Tape Read
+81,+82,+83,+84	PTRA	Priority 729 Tape Read All Alpha
+81,+82,+83,+84	PTRR	Priority 729 Tape Read Per Record Mark Control
+81,+82,+83,+84	PTW	Priority 729 Tape Write
+81,+82,+83,+84	PTWZ	Priority 729 Tape Write with Zero Elimination
+81,+82,+83,+84	PTWR	Priority 729 Tape Write Per Record Mark Control
+81,+82,+83,+84	PTWC	Priority 729 Tape Write with Zero Elimination and Per Record Mark Control Combined
+81,+82,+83,+84	PTSB	Priority 729 Tape Segment Backward Space
+81,+82,+83,+84	PTSF	Priority 729 Tape Segment Forward Space
+81,+82	PTM	Priority 729 Tape Mark Write
+81,+82	PTSM	Priority 729 Tape Segment Mark Write
−61	QLN	Inquiry Control Latch Set On
−62	QLF	Inquiry Control Latch Set Off
−61	TLN	729 Tape Latch Set On
−62	TLF	729 Tape Latch Set Off
−61	ULN	Unit Record Latch A Set On
−62	ULF	Unit Record Latch A Set Off
−61	ULN	Unit Record Latch B Set On
−62	ULF	Unit Record Latch B Set Off
+60	BDCL	7907 Stacking Latch Test
−61	DCLN	7907 Normal Latch Set On
−61	DCAN	7907 Attention Latch Set On
−62	DCLF	7907 Normal Latch Set Off
−62	DCAF	7907 Attention Latch Set Off

Floating-Decimal-Point

+74	FA	Floating Add
−74	FS	Floating Subtract
+73	FM	Floating Multiply
−73	FD	Floating Divide
+71	FR	Floating Round
+77	FAA	Floating Add Absolute
−77	FSA	Floating Subtract Absolute
+75	FZA	Floating Zero and Add
+76	FAD	Floating Add Double Precision
−75	FDD	Floating Divide Double Precision
−76	FADS	Floating Add Double Precision and Suppress Normalization
+70	FBV	Floating Branch Overflow
−70	FBU	Floating Branch Underflow

Miscellaneous

+51	BAS	Branch if Alteration Switch is On
+61,+62,+63	BES	Branch if Electronic Switch is On
+61,+62,+63	BSF	Branch if Electronic Switch is On and Set Off if On
+61,+62,+63	BSN	Branch if Electronic Switch is On and Set On if Off
+61,+62,+63	ESN	Electronic Switch On
+61,+62,+63	ESF	Electronic Switch Off
+00	HB	Halt and Branch
−00	HP	Halt and Proceed
+41	HMFV	Halt Mode for Field Overflow
+69	ITZ	Interval Timer Zero
+69	ITS	Interval Timer Store
−03	HMSC	Halt Mode for Sign Change
−03	MSA	Make Sign Alpha
−03	MSP	Make Sign Plus
−03	MSM	Make Sign Minus
−01	NOP	No Operation
+41	SMFV	Sense Mode for Field Overflow
−03	SMSC	Sense Mode for Sign Change
+04	BASS	Branch if Additional Storage Switch is On
+04	ASSN	Additional Storage Switch On
+04	ASSF	Additional Storage Switch Off
±88	TRN	Read Binary 729 Tape

Core Storage and Register Addresses

Core Storage Addresses

0000-4999	(5000-word capacity)
0000-9989	(9990-word capacity)
0000-29999	(7074 with Additional Core Storage optional Feature — 30,000 word capacity)

Core Storage Locations With Special Functions

0001-0099	Indexing words 01-99
0097	Priority address word
0098	Table lookup indexing value and found address
0099	Address of priority final status word
0100	Address at which indicator settings are stored prior to priority routine
0101-0103	Electronic switches
0104	Unit-record priority A branch address
0105	Unit-record priority B branch address
0106	Inquiry-control 1 priority branch address
0107	Inquiry-control 2 priority branch address
0110-0119	Final-status words, 729 tape units 0-9, channel 1
0120-0129	Final-status words, 729 tape units 0-9, channel 2
0130-0139	Final-status words, 729 tape units 0-9, channel 3
0140-0149	Final-status words, 729 tape units 0-9, channel 4
0150-0159	729 Tape priority branch addresses
0160-0169	Initial-status words, 729 tape units 0-9, channel 1
0170-0179	Initial-status words, 729 tape units 0-9, channel 2
0180-0189	Initial-status words, 729 tape units 0-9, channel 3
0190-0199	Initial-status words, 729 tape units 0-9, channel 4
0301-0304	Final-status words, 7907 channels 1-4
0311-0314	Normal stacking latch branch addresses, 7907 channels 1-4
0321-0324	Attention stacking latch branch addresses, 7907 channels 1-4
0351-0354	Initial status words, 7907 channels 1-4

Register Addresses

9991	Accumulator 1
9992	Accumulator 2
9993	Accumulator 3
9995	Program register—addressable from console only
9999	Instruction counter—addressable from console only

Operation Codes That Allow Accumulator Addresses

+03	Compare Storage to Digit	CD
−03	Sign Control:	
	Compare Sign to Alpha	CSA
	Compare Sign to Minus	CSM
	Compare Sign to Plus	CSP
	Make Sign Alpha	MSA
	Make Sign Minus	MSM
	Make Sign Plus	MSP
−11, −21, −31	Zero Storage and Store Accumulator #	ZST1, ZST2, ZST3
+12, +22, +32	Store Accumulator #	ST1, ST2, ST3
−12, −22, −32	Store Digits from Accumulator # and Ignore Sign	STD1, STD2, STD3
+13, +23, +33	Zero Accumulator # and Add	ZA1, ZA2, ZA3
−13, −23, −33	Zero Accumulator # and Subtract	ZS1, ZS2, ZS3
+14, +24, +34	Add to Accumulator #	A1, A2, A3
−14, −24, −34	Subtract from Accumulator #	S1, S2, S3
+15, +25, +35	Compare Accumulator # to Storage	C1, C2, C3
−15	Compare Absolute Accumulator 1 to Absolute in Storage	CA
+16	Zero Accumulator 1 and Add Absolute	ZAA
−16	Zero Accumulator 1 and Subtract Absolute	ZSA
+17	Add Absolute to Accumulator 1	AA
−17	Subtract Absolute from Accumulator 1	SA
+18, +28, +38	Add to Storage from Accumulator #	AS1, AS2, AS3
−18, −28, −38	Subtract Accumulator # from Storage	SS1, SS2, SS3
+19, +29, +39	Add to Absolute Storage from Accumulator #	AAS1, AAS2, AAS3
+45	Index Word Load	XL
−45	Index Word Unload	XU
−48	Index Word Load and Interchange	XLIN
+53	Multiply	M
−53	Divide	D

Operation Codes That Use Field Definition

−11, −21, −31	Zero Storage and Store Accumulator #	ZST1, ZST2, ZST3
+12, +22, +32	Store Accumulator #	ST1, ST2, ST3
−12, −22, −32	Store Digits from Accumulator # and Ignore Sign	STD1, STD2, STD3
+13, +23, +33	Zero Accumulator # and Add	ZA1, ZA2, ZA3
−13, −23, −33	Zero Accumulator # and Subtract	ZS1, ZS2, ZS3
+14, +24, +34	Add to Accumulator #	A1, A2, A3
−14, −24, −34	Subtract from Accumulator #	S1, S2, S3
+15, +25, +35	Compare Accumulator # to Storage	C1, C2, C3
−15	Compare Absolute in Accumulator 1 to Absolute in Storage	CA
+16	Zero Accumulator 1 and Add Absolute	ZAA
−16	Zero Accumulator 1 and Subtract Absolute	ZSA
+17	Add Absolute to Accumulator 1	AA
−17	Subtract Absolute from Accumulator 1	SA
+18, +28, +38	Add to Storage from Accumulator #	AS1, AS2, AS3
−18, −28, −38	Subtract Accumulator # from Storage	SS1, SS2, SS3
+19, +29, +39	Add to Absolute Storage from Accumulator #	AAS1, AAS2, AAS3
+53	Multiply	M
−53	Divide	D
+66	Lookup Lowest	LL
+67	Lookup Equal Only	LE
+68	Lookup Equal or High	LEH

Store and Add-to-Storage Codes

All store and add-to-storage codes can turn on the field-overflow indicator:

−11, −21, −31	Zero Storage and Store Accumulator #	ZST1, ZST2, ZST3
+12, +22, +32	Store Accumulator #	ST1, ST2, ST3
−12, −22, −32	Store Digits from Accumulator # and Ignore Sign	STD1, STD2, STD3
+18, +28, +38	Add to Storage from Accumulator #	AS1, AS2, AS3
−18, −28, −38	Subtract Accumulator # from Storage	SS1, SS2, SS3
+19, +29, +39	Add to Absolute Storage from Accumulator #	AAS1, AAS2, AAS3

These codes *can* turn on the sign-change indicator:

+12, +22, +32	Store Accumulator #	ST1, ST2, ST3
+18, +28, +38*	Add to Storage from Accumulator #	AS1, AS2, AS3
−18, −28, −38*	Subtract Accumulator # from Storage	SS1, SS2, SS3

*Only if less than a full word is field-defined

These codes *cannot* turn on the sign-change indicator:

−11, −21, −31	Zero Storage and Store Accumulator #	ZST1, ZST2, ZST3
−12, −22, −32	Store Digits from Accumulator # and Ignore Sign	STD1, STD2, STD3
+19, +29, +39	Add to Absolute Storage from Accumulator #	AAS1, AAS2, AAS3

Index of 7070-7074 Operation Codes by Autocoder Mnemonics

OP CODE MNEMONIC	OP CODE NUMERIC		PAGE
C1, 2, 3	+15,+25,+35	Compare Accumulator # to Storage	44
CA	−15	Compare Absolute in Accumulator 1 to Absolute in Storage	46
CD	+03	Compare Storage to Digit	46
CSA	−03	Compare Sign to Alpha	47
CSM	−03	Compare Sign to Minus	47
CSP	−03	Compare Sign to Plus	47
D	−53	Divide	25
DCAF	−62	Data Channel Attention Latch Off	112
DCAN	−61	Data Channel Attention Latch On	111
DCLF	−62	Data Channel Normal Latch Off	112
DCLN	−61	Data Channel Normal Latch On	111
EAN	−57	Edit Alphameric to Numeric	64
ENA	+56	Edit Numeric to Alphameric	62
ENB	+57	Edit Numeric to Alphameric with Blank Insertion	64
ENS	−56	Edit Numeric to Alphameric with Sign Control	63
ESF	+61,+62,+63	Electronic Switch Off	40
ESN	+61,+62,+63	Electronic Switch On	40
FA	+74	Floating Add	122
FAA	+77	Floating Add Absolute	125
FAD	+76	Floating Add Double Precision	124
FADS	−76	Floating Add Double Precision and Suppress Normalization	124
FBU	−70	Floating Branch Underflow	129
FBV	+70	Floating Branch Overflow	129
FD	−73	Floating Divide	127
FDD	−75	Floating Divide Double Precision	128
FM	+73	Floating Multiply	126
FR	+71	Floating Round	126
FS	−74	Floating Subtract	123
FSA	−77	Floating Subtract Absolute	125
FZA	+75	Floating Zero and Add	121
HB	+00	Halt and Branch	49
HMFV	+41	Halt Mode for Field Overflow	43
HMSC	−03	Halt Mode for Sign Change	47
HP	−00	Halt and Proceed	49
ITS	+69	Interval Timer Store	19
ITZ	+69	Interval Timer Zero	19
LE	+67	Lookup Equal Only	68
LEH	+68	Lookup Equal or High	68
LL	+66	Lookup Lowest	66
M	+53	Multiply	24
MSA	−03	Make Sign Alpha	47
MSM	−03	Make Sign Minus	47
MSP	−03	Make Sign Plus	47

MNEMONIC OP CODE	NUMERIC OP CODE		PAGE
NOP	−01	No Operation	50
PC	+55	Priority Control	108
PCI	+55	Data Channel Priority Control	108
PDCP	±93,94,96,97	Data Channel Select Packed	140
PDCU	±93,94,96,97	Data Channel Select Unpacked Numeric Start	140
PDCPR	±93,94,96,97	Data Channel Select Packed with RMC	140
PDCUA	±93,94,96,97	Data Channel Select Unpacked Alpha Start	140
PDCUR	±93,94,96,97	Data Channel Select Unpacked Alpha Start with RMC	140
PR	+64	Priority Release	112
PTM	+81,+82,+83,+84	Priority 729 Tape Mark Write	80
PTR	+81,+82,+83,+84	Priority 729 Tape Read	76
PTRA	+81,+82,+83,+84	Priority 729 Tape Read All Alpha	80
PTRN	±88	Read Binary 729 Tape	18
PTRR	+81,+82,+83,+84	Priority 729 Tape Read per Record Mark Control	77
PTSB	+81,+82,+83,+84	Priority 729 Tape Segment Backward Space	79
PTSF	+81,+82,+83,+84	Priority 729 Tape Segment Forward Space	79
PTSM	+81,+82,+83,+84	Priority 729 Tape Segment Mark Write	82
PTW	+81,+82,+83,+84	Priority 729 Tape Write	77
PTWC	+81,+82,+83,+84	Priority 729 Tape Write with Zero Elimination and per Record Mark Control Combined	78
PTWR	+81,+82,+83,+84	Priority 729 Tape Write per Record Mark Control	78
PTWZ	+81,+82,+83,+84	Priority 729 Tape Write with Zero Elimination	78
QLF	−62	Inquiry Latch Set Off	112
QLN	−61	Inquiry Latch Set On	111
QR	+54	Inquiry Read	99
QW	+54	Inquiry Write	99
RG	−65	Record Gather	62
RS	+65	Record Scatter	62
S1, 2, 3	−14,−24,−34	Subtract from Accumulator #	23
SA	−17	Subtract Absolute from Accumulator 1	27
SL	−50	Shift Left Coupled	34
SL1, 2, 3	+50	Shift Left Accumulator #	32
SLC	−50	Shift Left and Count Coupled	34
SLC1, 2,3	+50	Shift Left and Count Accumulator #	32
SLS	−50	Shift Left Split	34
SMFV	+41	Sense Mode for Field Overflow	43
SMSC	−03	Sense Mode for Sign Change	47
SR	−50	Shift Right Coupled	34
SR1, 2, 3	+50	Shift Right Accumulator #	32
SRR	−50	Shift Right and Round Coupled	34
SRR1, 2, 3	+50	Shift Right and Round Accumulator #	32
SRS	−50	Shift Right Split	34
SS1, 2, 3	−18,−28,−38	Subtract Accumulator # from Storage	31
ST1, 2, 3	+12,+22,+32	Store Accumulator #	29
STD1, 2, 3	−12,−22,−32	Store Digits from Accumulator # and Ignore Sign	29
TEF	±81,±82,±83,±84	729 Tape End of File Off	82
TLF	−62	729 Tape Latch Set Off	112
TLN	−61	729 Tape Latch Set On	111
TM	−81,−82,−83,−84	729 Tape Mark Write	80

Clearing a Specified Portion of Core Storage to Zeros

It is often desirable to clear all of magnetic core storage, or a specified portion of it, to zeros. (The zeros can be either all plus or all minus.) This technique is often used during program testing; it can also be used to clear core storage after one program has been used, before loading a second program. Areas used for storage of data such as tables can also be cleared by this method.

Instructions and Addresses

Two instructions are loaded from the console. They are a read-card instruction, and the record-definition word defining the storage words to be read into. The instruction is stored in location 0000, and the RDW is stored in location 0001, as follows:

LOCATION	CONTENTS	
0000	+69 00 1 1 0001	UR
0001	−00 0002 0006	(RDW)

The card to be read contains the instructions for the operation and the addresses that specify the portion of core storage to be cleared. Assume, for example, that storage locations 1700-1799 are to be cleared. Columns 1-50 are brought to words 0002-0006, and contain the following:

CARD COLUMNS	LOCATION	CONTENTS	
1-10	0002	±0000000000	zeros to be used
11-20	0003	−4505020000	instructions
21-30	0004	+6500050006	
31-40	0005	+0017000000	addresses
41-50	0006	−0017011799	

The card can be completely prepunched except for the addresses in columns 33-36 and 43-50 and the zone punch in column 10 for plus or minus zeros. Figure 184 is an example of the prepunched card. (Note the prepunched X in column 50.)

Operation

Place the card in the 7500 feed, and start the program from the console at 0000. The card is read, bringing columns 1-50 to words 0002-0006. If there is no validity error or end of file, the next instruction is taken from 0003. (End-of-file does not occur unless there are no other cards to be read; the end-of-file key on the 7500 must be used, however, if there are no cards in back of this one.)

The instruction in 0003 is as follows:

Index-word	*S01*	*23*	*45*	*6789*	
unload	−45	05	02	0000	XU

This instruction puts the zeros in word 0002 into location 0000, indexed by positions 2-5 of word 0005. These positions contain 1700; thus, word 1700 is loaded with zeros.

The instruction in word 0004 is as follows:

Record	*S01*	*23*	*45*	*6789*	
scatter	+65	00	05	0006	RS

Word 0006 is an RDW, defining the remainder of the area to be cleared (1701-1799). This is the receiving area. Positions 2-5 of index word 05 define the first word of the transmitting area (1700). Thus, this instruction "scatters" the contents of 1700 to 1701, then the contents of 1701 to 1702, etc., until the contents of 1798 are brought to 1799, which is the *stop* address of the RDW.

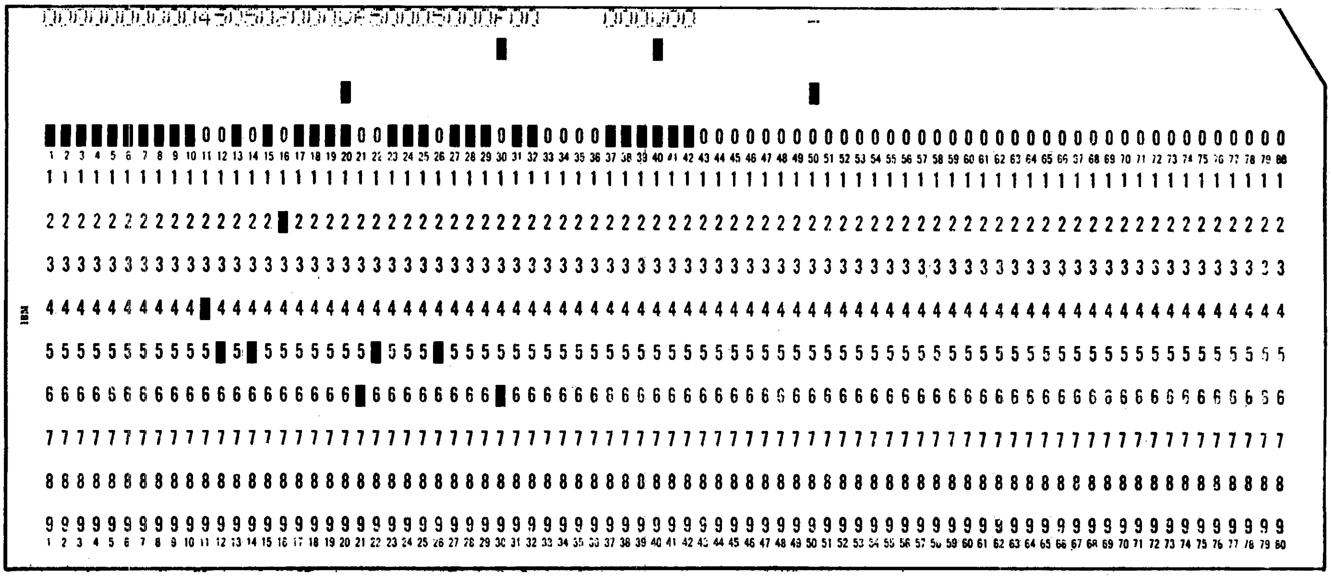


Figure 184. Input Card for Clearing a Portion of Storage

The operation code in word 0005 is +00, which is halt and proceed (HP), thus ending the routine. In this example, the contents of the instruction counter and program register are typed as follows:

IC
0006
PR
+0017000000

In an operation where 0005 is cleared to zeros, the program register would be typed as ±0000000000.

Timing: All of core storage, locations 0000-9989, can be cleared to zeros by this method, in about 2.5 seconds, including card-read and typing time.

List of 7070 Instruction Execution Times

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
+00	HB	36
−00	HP	36
+01	B	36
−01	NOP	36
+02	BLX	36
+03	CD	36
−03	Sign Control:	
	CSA	36
	CSM	36
	CSP	36
	MSA	36
	MSM	36
	MSP	36
	SMSC	36
	HMSC	36
	BSC	36
+10	BZ1	36
−10	BM1	36
+11	BV1	36
−11	ZST1	
	For following number of digit positions:	
	1	48
	2	48
	3	60
	4	60
	5	60
	6	72
	7	72
	8	72
	9	84
	10	36
+12	ST1	Same as ZST1
−12	STD1	Same as ZST1
+13	ZA1	
	For following number of digit positions:	
	1	36
	2	48
	3	48
	4	48
	5	60
	6	60
	7	60
	8	72
	9	72
	10	36
−13	ZS1	Same as ZA1
+14	A1	
	For following number of digit positions:	
	True add to accumulator:	
	1	48
	2	48
	3	48
	4	60
	5	60
	6	60
	7	72
	8	72
	9	72
	10	84
	Complement add to accumulator:	
	1	36
	2	48
	3	48
	4	48
	5	60
	6	60
	7	60
	8	72
	9	72
	10	72
	If recomplement:	
	1	60
	2	60
	3	72
	4	84
	5	84
	6	96
	7	108
	8	108
	9	120
	10	132

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
−14	S1	Same as A1
+15	C1	
	For following number of digit positions:	
	1	36
	2	48
	3	48
	4	48
	5	60
	6	60
	7	60
	8	72
	9	72
	10	72
−15	CA	Same as C1
+16	ZAA	Same as ZA1
−16	ZSA	Same as ZA1
+17	AA	Same as A1
−17	SA	Same as A1
+18	AS1	
	For following number of digit positions:	
	True add to storage:	
	1	48
	2	48
	3	60
	4	60
	5	60
	6	72
	7	72
	8	72
	9	84
	10	84
	Complement add to storage:	
	1	48
	2	48
	3	60
	4	60
	5	60
	6	72
	7	72
	8	72
	9	84
	10	84
	If recomplement:	
	1	60
	2	72
	3	84
	4	84

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
	5	96
	6	108
	7	108
	8	120
	9	132
	10	132
−18	SS1	Same as AS1
+19	AAS1	Same as AS1
+20	BZ2	36
−20	BM2	36
+21	BV2	36
−21	ZST2	Same as ZST1
+22	ST2	Same as ZST1
−22	STD2	Same as ZST1
+23	ZA2	Same as ZA1
−23	ZS2	Same as ZA1
+24	A2	Same as A1
−24	S2	Same as A1
+25	C2	Same as C1
+28	AS2	Same as AS1
−28	SS2	Same as AS1
+29	AAS2	Same as AS1
+30	BZ3	36
−30	BM3	36
+31	BV3	36
−31	ZST3	Same as ZST1
+32	ST3	Same as ZST1
−32	STD3	Same as ZST1
+33	ZA3	Same as ZA1
−33	ZS3	Same as ZA1
+34	A3	Same as A1
−34	S3	Same as A1
+35	C3	Same as C1
+38	AS3	Same as AS1
−38	SS3	Same as AS1
+39	AAS3	Same as AS1
+40	BL	36
−40	BH	36
+41	Field Overflow:	
	BFV	36
	SMFV	36
	HMFV	36
−41	BE	36
−43	BCX	48
+44	BXN	36
−44	BXM	36
+45	XL	36
−45	XU	36
+46	XZA	48
−46	XZS	48

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
+47	XA	60 without recomplement 84 with recomplement
−47	XS	Same as XA
+48	XSN	36
−48	XLIN	48
+49	BIX	84
−49	BDX	60 without recomplement 84 with recomplement
+50	Shift Control:	

	NOT INDEXED		INDEXED	
OPERATION	NUMBER OF POSITIONS SHIFTED	TIME (μs)	NUMBER OF POSITIONS SHIFTED	TIME (μs)
SR# or SL#	0, 1, 2	36	0	60
	3, 4, 5	48	1, 2, 3	72
	6, 7, 8	60	4, 5, 6	84
	9, 10	72	7, 8, 9	96
			10	108
SRR#	0	36	0	60
	1	84	1, 2	120
	2, 3, 4	96	3, 4, 5	132
	5, 6, 7	108	6, 7, 8	144
	8, 9, 10	120	9, 10	156
SLC	0	84	0	120
	1, 2	96	1, 2, 3	132
	3, 4, 5	108	4, 5, 6	144
	6, 7, 8	120	7, 8, 9	156
	9, 10	132	10	168

−50 Coupled Shift Control:

SR or SL	0, 1, 2, 10, 11, 12	36	0, 10	60
	3, 4, 5, 13, 14, 15	48	1, 2, 3, 11, 12, 13	72
	6, 7, 8, 16, 17, 18	60	4, 5, 6, 14, 15, 16	84
	9, 19, 20	72	7, 8, 9, 17, 18, 19	96
			20	108
SRR	0	36	0	60
	10	120	1, 10, 11	156
	1, 2, 3, 11, 12, 13	132	2, 3, 4, 12, 13, 14	168
	4, 5, 6, 14, 15, 16	144	5, 6, 7, 15, 16, 17	180
	7, 8, 9, 17, 18, 19	156	8, 9, 18, 19, 20	192
	20	168		
SLC	0	84	0	120
	1, 2, 10	96	1, 2, 3, 10	132
	3, 4, 5, 11, 12	108	4, 5, 6, 11, 12, 13	144
	6, 7, 8, 13, 14, 15	120	7, 8, 9, 14, 15, 16	156
	9, 16, 17, 18	132	17, 18, 19	168
	19, 20	144	20	180

SRS FROM POSITION IN:		SLS FROM POSITION IN:		NOT INDEXED		INDEXED	
A1	A2	A1	A2	NUMBER OF POSITIONS SHIFTED	TIME (μs)	NUMBER OF POSITIONS SHIFTED	TIME (μs)
0	0	9	9	0 – 20	132	0 – 20	156
1	1	8	8	0 – 20	120	0 – 20	156
2	2	7	7	0 – 16	108	0 – 16	144
2			7	17 – 20	120	17 – 20	156
3	3	6	6	0 – 16	108	0 – 16	132
3			6	17 – 20	120	17 – 20	144
4	4	5	5	0 – 13	96	0 – 13	132
4			5	14 – 20	108	14 – 20	144
5	5	4	4	0 – 10	84	0 – 11	120
5			4	11 – 13	96	12 – 14	132
5			4	14 – 20	108	15 – 20	144
6	6	3	3	0 – 10	84	0 – 8	108
6			3	11 – 13	96	9 – 11	120
6			3	14 – 20	108	12 – 20	132
7	7	2	2	0 – 7	72	0 – 8	108
7			2	8 – 10	84	9 – 11	120
7			2	11 – 20	96	12 – 20	132
8	8	1	1	0 – 4	60	0 – 5	69
8			1	5 – 7	72	6 – 8	108
8			1	8 – 10	84	9 – 11	120
8			1	11 – 20	96	12 – 20	132
9	9	0	0	0 – 4	60	0 – 2	84
9			0	5 – 7	72	3 – 5	96
9			0	8 – 10	84	6 – 8	108
9			0	11 – 20	96	9 – 20	120

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
+51	Branch on Alteration Switch or Channel Busy:	
	BAS	36
	BCB	36
	BDCB	36
+53	M	

Formula (timing in μs)

$48\ (N^1+2N^2+3N^3)+4N^4+4N^5+180$

where N^1 = number of 1, 2, 4 digits
N^2 = number of 3, 5, 6, 8 digits
N^3 = number of 7, 9 digits
N^4 = number of zeros
N^5 = number of zero groups (in multiplier)

If the total is not a multiple of 12, use the next higher multiple of 12. A zero group is a zero or two or more adjacent zeros.

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
−53	D Formula (timing in μs) 192 + 48 (10 + sum of quotient digits) where R = number of RDW's W = number of numeric words moved	
+54	Inquiry Control:	
	QR	3,000 (avg)
	QW	17,000 (avg)
+55	PC	36
	PC1	36
+56	ENA Formula (timing in μs) 36 + 36R + 120W where R = number of RDW's W = number of numeric words moved	
−56	ENS	Same as ENA
+57	ENB	Same as ENA
−57	EAN Formula (timing in μs) 36 + 36R + 72W where R = number of RDW's W = number of numeric words moved	
+60	Stacking Latch Test:	
	BAL	36
	BUL	36
	BQL	36
	BTL	36
	BDCA	36
	BDCL	36
+61	Electronic Switch Control:	
+62	BES	48
+63	ESN	48
	ESF	48
	BSN	48
	BSF	48
−61	Stacking Latch Set On:	
	ULN	36
	QLN	36
	TLN	36
	DCAN	36
	DCLN	36
−62	Stacking Latch Reset Off:	
	ULF	36
	QLF	36
	TLF	36
	DCAF	36
	DCLF	36
+64	PR	36

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
+65	RS Formula (timing in μs): 36 + 36R + 24W where R = number of RDW's W = number of words moved	
−65	RG	Same as RS
+66	LL Formula (timing in μs): 36+ 36 per RDW +108 per table word + 36 if a lowest value is found + 60 for each lower value found	
+67	LE Formula (timing in μs): 36+ 36 per RDW +108 per table word searched	
+68	LEH	Same as LE
+69	Unit Record Control:	
	US	2,800 (avg)
	UR	2,800 (avg)
	UW/UP	2,800 (avg)
	UWIV/UPIV	2,800 (avg)
	ITS	1,300 – but when timer register is incrementing, up to 9 milliseconds additional time may be required
	ITZ	60 (avg)
+70	FBV	36
−70	FBU	36
+71	FR	72
+73	FM Formula (timing in μs): $48\,(N^1+2N^2+3N^3)+4N^4+4N^5+408$ where N^1 = number of 1, 2, 4 digits N^2 = number of 3, 5, 6, 8, digits N^3 = number of 7, 9 digits N^4 = number of zeros N^5 = number of zero groups (in multiplier) If the total is not a multiple of 12, use the next higher multiple of 12. Consider multiple-digit positions 0-1 to be a zero group. A zero group is a zero or two or more adjacent zeros.	
−73	FD Formula (timing in μs): 360 + 48 [8 + (sum of quotient digits)]. The quotient is 8 digits in size.	
+74	FA	168-336

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
−74	FS	168-336
+75	FZA	108-156
−75	FDD	Same as FD
+76	FAD	168-360
−76	FADS	144-300
+77	FAA	168-336
−77	FSA	168-336
±81,	729 Tape Control:	
±82,	TR	130
±83,	TRR	130
±84	TW	130
	TWR	130
	TWZ	130
	TWC	130
	TSF	130
	TSB	130
	TRA	130
	TSEL	130
	TM	130
	TRW	130
	TRU	130
	TRB	130
	TSM	130
	TSK	130
	TEF	130
	TSLD	130
	TSHD	130
±93,	7907 Data Channel Select:	
±94,	DCP	108
±96,	DCUA	108
±97	DCU	108
	DCPR	108
	DCUR	108

List of 7074 Instruction Execution Times

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
+00	HB	4
−00	HP	4
+01	B	4
−01	NOP	4
+02	BLX	12
+03	CD	8
−03	Sign Control:	
	CSA	8
	CSM	8
	CSP	8
	MSA	12
	MSM	12
	MSP	12
	SMSC	4
	HMSC	4
	BSC	4
+04	Additional Storage Control:	
	ASSN	6
	ASSF	6
	BASS	6
+10	BZ1	6
−10	BM1	4
+11	BV1	4
−11	ZST1	8 full field 12 not full field
+12	ST1	12
−12	STD1	12
+13	ZA1	8
−13	ZS1	8
+14	A1	8 true 10 complement
−14	S1	8 true 10 complement
+15	C1	8
−15	CA	8
+16	ZAA	8
−16	ZSA	8
+17	AA	8 true 10 complement
−17	SA	8 true 10 complement
+18	AS1	12 true 14 complement
−18	SS1	12 true 14 complement
+19	AAS1	12 true 14 complement
+20	BZ2	6
−20	BM2	4
+21	BV2	4
−21	ZST2	8 full field 12 not full field
+22	ST2	12
−22	STD2	12
+23	ZA2	8
−23	ZS2	8
+24	A2	8 true 10 complement
−24	S2	8 true 10 complement
+25	C2	8
+28	AS2	12 true 14 complement
−28	SS2	12 true 14 complement

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
+29	AAS2	12 true 14 complement
+30	BZ3	6
−30	BM3	4
+31	BV3	4
−31	ZST3	8 full field 12 not full field
+32	ST3	12
−32	STD3	12
+33	ZA3	8
−33	ZS3	8
+34	A3	8 true 10 complement
−34	S3	8 true 10 complement
+35	C3	8
+38	AS3	12 true 14 complement
−38	SS3	12 true 14 complement
+39	AAS3	12 true 14 complement
+40	BL	4
−40	BH	4
+41	Field Overflow:	
	BFV	4
	SMFV	4
	HMFV	4
−41	BE	4
−43	BCX	10
+44	BXN	10
−44	BXM	8
+45	XL	12
−45	XU	12
+46	XZA	12
−46	XZS	12
+47	XA	14
−47	XS	14
+48	XSN	12
−48	XLIN	12
+49	BIX	14
−49	BDX	14
+50	Shift Control:	
	SR#	6 + (N−3)*
	SRR#	6 + N
	SL#	6 + (N−3)*
	SLC#	14 + (N−3)*
	where N = number of positions to be shifted	
	*Note that if $N<3$, $(N-3)=0$	

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
−50	Coupled Shift Control:	
	SR	6 + (N−3)*
	SRR	8 + N
	SL	6 + (N−3)*
	SLC	14 + (N−3)*
	SRS	11 + N point remains in acc 8 + N point moves out of acc
	SLS	11 + N point remains in acc 8 + N point moves out of acc
	where N = number of positions to be shifted	
	*Note that if $N<3$, $(N-3)=0$	
+51	Branch on Alteration Switch or Channel Busy:	
	BAS	6
	BCB	6
	BDCB	6
+53	M	
	Formula (timing in μs) Exact: $26 + 2N^1 + 4N^2 + 6N^3$ Average: $26 + 3.4N$ where N^1 = number of 1's, 2's, and 4's in multiplier N^2 = number of 3's, 5's, 6's, and 8's in multiplier N^3 = number of 7's and 9's in multiplier N = number of significant digits in multiplier	
−53	D	
	Formula (timing in μs) Exact: $26 + 2N^0 + 6N^1 + 8N^2$ Average: $26 + 2N^0 + 6.8N$ where N^0 = number of high order zeros in quotient N^1 = number of 1's, 2's, 3's, 8's, 9's, and 0's (except high order zeros) in quotient N^2 = number of 4's, 5's, 6's, and 7's in quotient N = number of significant digits in quotient	

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
+54	Inquiry Control:	
	QR	3,000 (avg)
	QW	17,000 (avg)
+55	PC	12
	PC1	12
+56	ENA	
	Formula (timing in μs) 8 + 4R + 12W where R = number of RDW's W = number of words moved	
−56	ENS	Same as ENA
+57	ENB	Same as ENA
−57	EAN	Same as ENA
+60	Stacking Latch Test:	
	BAL	7
	BUL	7
	BQL	7
	BTL	7
	BDCA	7
	BDCL	7
+61,	Electronic Switch Control:	
+62,	BES	14
+63	ESN	14
	ESF	14
	BSN	14
	BSF	14
−61	Stacking Latch Set On:	
	ULN	7
	QLN	7
	TLN	7
	DCAN	7
	DCLN	7
−62	Stacking Latch Reset Off:	
	ULF	7
	QLF	7
	TLF	7
	DCAF	7
	DCLF	7
+64	PR	16
+65	RS	
	Formula (timing in μs) 8 + 4R + 8W where R = number of RDW's W = number of words moved	
−65	RG	Same as RS
+66	LL	
	Formula (timing in μs) 16 + 4R + 6T + 4F where R = number of RDW's T = number of table words F = number of found conditions	

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
+67	LE	
	Formula (timing in μs) 16 + 4R + 6T where R = number of RDW's T = number of table words	
+68	LEH	
	Formula (timing in μs) 16 + 4R + 6T where R = number of RDW's T = number of table words	
+69	Unit Record Control:	
	US	2,800 (avg)
	UR	2,800 (avg)
	UW/UP	2,800 (avg)
	UWIV/UPIV	2,800 (avg)
	ITS	1,300 — but when timer register is incrementing, up to 9 milliseconds additional time may be required
	ITZ	60 (avg)
+70	FBV	4
−70	FBU	4
+71	FR	4
+73	FM	
	Formula (timing in μs) Exact: $28 + 2N^1 + 4N^2 + 6N^3 + N^4$ Average: $28 + 3.4N + N^4$ where N^1 = number of 1's, 2's, and 4's in multiplier N^2 = number of 3's, 5's, 6's, and 8's in multiplier N^3 = number of 7's and 9's in multiplier N^4 = number of positions of post-normalization required N = number of significant digits in multiplier	
	Note: In the formulas above, the multiplier is the 8-digit fraction.	
−73	FD	
	Formula (timing in μs) Exact: $29 + 6N^1 + 8N^2$ Average: $26 + 6.8N$ where N^1 = number of 1's, 2's, 3's, 8's, 9's, and 0's (except high order zeros) in quotient N^2 = number of 4's, 5's, 6's, and 7's in quotient	

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
	N = number of significant digits in quotient	
	Note: In the formulas above, the quotient is the 8-digit fraction.	
+74	FA	
	Formula (timing in μs)	
	$16 + (N^1-1)* + (N^2-1)*$, when no recomplementing required	
	$19 + (N^1-1)* + N^2$, when recomplementing required	
	where N^1 = difference between characteristics of the numbers	
	N^2 = number of positions of postnormalization required	
	*Note that if $N^1<1$, $(N^1-1) = 0$	
	and if $N^2<1$, $(N^2-1) = 0$	
−74	FS	Same as FA
+75	FZA	
	Formula (timing in μs)	
	$11 + N^2$	
	where N^2 = number of positions of postnormalization required	
−75	FDD	Same as FD
+76	FAD	Same as FA
−76	FADS	Same as FA
+77	FAA	Same as FA
−77	FSA	Same as FA
±81,	729 Tape Control:	
±82,	TR	19

NUM OP CODE	MNEMONIC OP CODE	TIME (μs)
±83,	TRR	19
±84	TW	19
	TWR	19
	TWZ	19
	TWC	19
	TSF	19
	TSB	19 not at load point 28 at load point
	TSEL	28
	TM	19
	TWR	28
	TRU	28
	TRB	19 not at load point 28 at load point
	TSM	19
	TSK	28
	TEF	28
	TSLD	28
	TSHD	28
±88	TRN	19
±93,	7907 Data Channel Control:	
±94,	DCP	19
±96,	DCU	19
±97	DCPR	19
	DCUA	19
	DCUR	19

Note: Indexing an instruction adds 5 microseconds to the time.

Compatibility of 7070-7074

The 7074 has been designed to provide greatly increased internal processing speed while retaining program compatibility with the 7070. To attain this objective, each operation code is decoded to perform the same functions in as short a time as possible. As a result, the 7074 is essentially program compatible with the 7070. However, because the 7074 accomplishes internal operations in much less time than the 7070, some operational differences result during certain sequences of instructions that follow I-O operations. Also, to gain speed, working registers are used differently, resulting in differences in content at the completion of certain instructions. This section explains operational differences that exist between the 7070 and 7074.

1. ±81 xxx Any tape operation resulting in other than condition 2 (Correct Length Record).
   +51 BCB Branch channel busy to same location.
   −62 TLF Reset stacking latch set by ±81 code.

The 7070 will not go into priority mode since the latch is reset before the priority controls can be made operative. In the 7074, the priority controls are activated immediately upon the setting of the latch, and the interrupt will result.

2. ±81 xxx Any tape operation using RDW addresses.
   +46 XZA Where the index word is used as the RDW in the previous instruction.

The 7070 will use the RDW in the tape operation before it can be modified. The 7074 will modify the RDW before it can be used in the tape operation. The +46 code is an example of several such instructions which can also modify the RDW. In general, it is not recommended programming to modify any RDW being used by a tape operation until that operation has been completed.

3. +66 LL Lookup Lowest
Accumulator 2, in this operation, is a working register, acting as temporary storage for the location of the lowest value found at any point in the operation. In both 7070 and 7074, this address is transmitted to index word 0098 at the completion of the operation. However, the final contents of the accumulator are different.

4. When an interrupt occurs, the contents of the instruction counter are transmitted to index word 0097, positions 2-5, for reference at the end of the priority routine. The 7070 resets 0097 before this address is stored. The 7074 resets only positions 2-5; the remainder of the index word is unaltered. The contents of other positions in this word do not affect the priority release instruction that restores the instruction counter to its former value using positions 2-5 as a reference.

5. Accumulator 3 is used as a working register during all floating-point operations, except Floating Zero and Add. Results are always in accumulators 1 and 2. Since the 7070 and 7074 use working registers differently, the contents of accumulator 3 during and after floating-point operations are not necessarily the same.

6. On the 7070, digit position 5 of the +55 PC or PC1 instruction is not checked for being a 1 or a 0. On a 7074 lacking 7907 Data Channel(s), a digit other than 0 in position 5 of a +55 PC or PC1 instruction causes an IX WD 00 error; on a 7074 with 7907 Data Channel(s), a digit other than 0 to 1 in position 5 of a +55 PC or PC1 instruction causes an IX WD 00 error.

IBM 7150 Console

The IBM 7150 Console contains operating keys, lights, and switches for the central processing unit of the 7070-7074. As shown in Figure 185, it has three operating components: the operating panel, the operating keyboard, and the typewriter keyboard. The two keyboards are beside each other in the center portion of the console, and the operating panel is vertically mounted at the right. The console typewriter is in the same unit as the keyboards.

There are no lights on the console for the purpose of displaying the contents of a storage word. This is done by the console typewriter, providing a permanent record of all storage words that have been displayed.

Another feature of the IBM 7150 Console is the illuminated key, which is both a key and a light. When the key is pressed, it lights, indicating that it is on. The light goes off when its feature is turned off.

Automatic Typing on Machine Stop

Whenever execution of the stored program stops for routine causes, the console typewriter automatically types the contents of the IC (instruction counter) and PR (program register). The four digits of the IC type first, the carriage tabs, and then the ten digits with sign of the PR type. A carriage return occurs before and after typing. Routine causes of program execution stoppage are programmed halt (HB or HP), address stop, and depression of the stop key on the operating keyboard.

Error conditions that stop execution of the stored program may or may not result in automatic typing of the IC and PR contents. Typing occurs only if the instruction in the PR has been completely executed. Partial execution is signalled by the blinking on and off of the program-advance light on the operating panel. Some of the error conditions that do result in automatic typing of IC and PR contents are those caused by:

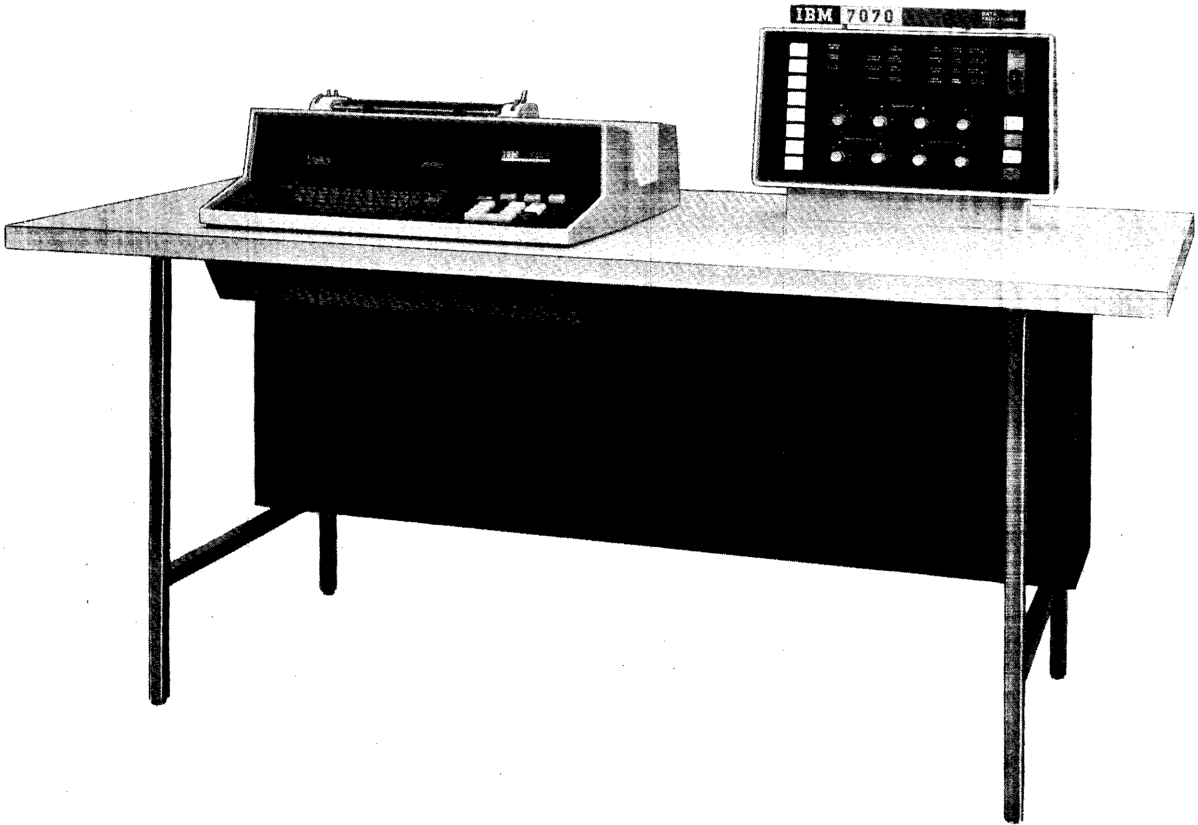


Figure 185. IBM 7150 Console

1. A zero divisor when executing the divide instructions (−53, D; −73, FD; and −75, FDD).
2. The divisor's being lower than the contents of accumulator 1 when executing the divide (−53, D) instruction.
3. The condition in floating-decimal-point arithmetic that ten times the absolute fraction of the divisor is less than or equal to the absolute fraction of the dividend. See Figure 183 for examples of results when unnormalized factors are used.
4. An alpha signed RDW (record definition word) when executing the lookup lowest (+66, LL) instruction.
5. An alpha-signed RDW when executing the edit alphameric to numeric (−57, EAN) instruction.
6. A zero increment when executing the lookup equal only (+67, LE) instruction.

When automatic typing occurs, the instruction counter contains the location of the next instruction to be executed, not the location of the instruction then in the program register.

When a stop occurs with a branch instruction in the program register, the instruction counter contains the location of the next sequential instruction, regardless of whether branching is to take place.

Operating Panel

Figure 186 is a picture of the operating panel. The surface of the panel is black, enabling a light to stand out clearly when it is on; when an indicator light comes on, the printing is illuminated against the black background. On the right are the main power keys and lights. At the left are key-lights for the alteration switches, and accumulator and exponent overflow. In the center are the control lights and the dials for address stop and unit-record priority control.

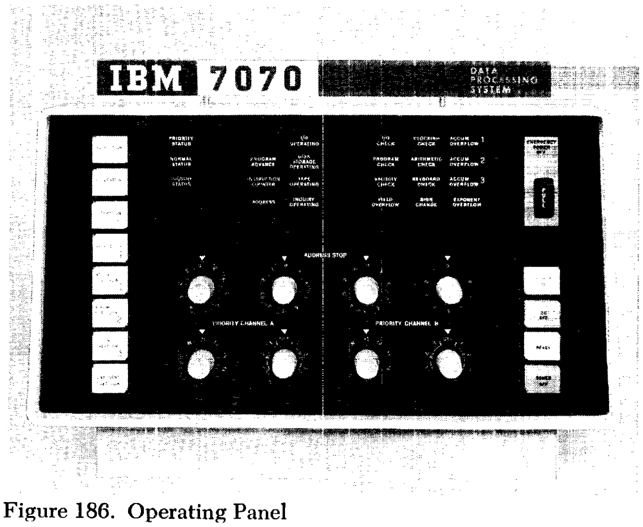


Figure 186. Operating Panel

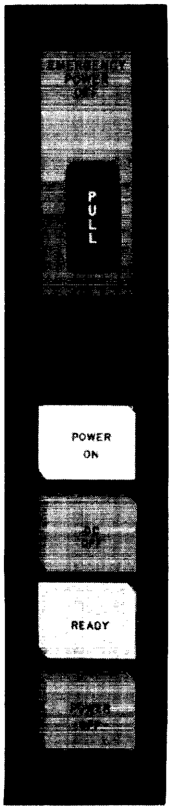


Figure 187. Power Keys and Lights

Power Keys and Lights

The power keys and lights (Figure 187) are located at the right of the operating panel. They are the main ON and OFF operating features in the system.

Emergency Pull: This should be used *only* in case of emergency, when all power must be shut off immediately to prevent injury to an individual or damage to the machine. If this device is used to turn off power, only a customer engineer should turn on the power again. Pulling this switch removes all power from all units except the disk files and unit-record equipment. Direct current only is removed from the disk files. The unit-record equipment is not affected by this switch.

Power On: This is a key-light. Pressing it provides full operating power to the system, either from a power-off condition or DC-off condition. When the key is pressed, its light comes on. It indicates that power has been supplied to the system and is turned off only by pressing the power-off key.

Ready: When the system is ready for operation, this light comes on. It takes a short time for the machine to be ready after the power has been turned on, because power must be supplied to various components of the system in a specified sequence. The ready light comes on immediately if POWER ON is pressed from a DC-off condition.

DC Off: Pressing this key turns off the DC power only. It is used when the machine is to be idle for a short time. When this key is pressed, the ready light turns off, but the power-on key-light stays on. Full power is restored immediately by pressing the power-on key.

Power Off: This key is the means of turning off the machine. When it is pressed, the ready light and the power-on key-light turn off. The computer is turned on again by pressing the power-on key.

Alteration Switches and Overflow Keys

These features (Figure 188) are on the left of the operating panel. They afford control by the console operator over certain features in the stored program.

Alteration Switches: The top four keys on the left side of the operating panel are the alteration switches. They are key-lights. Pressing one of these keys turns on the corresponding alteration switch, which can be interrogated by the stored program instruction +51, BAS. Each key latches in its pressed position. To turn off an alteration switch, press the key. It will unlatch from its pressed position and return to its normal position, and the light will turn off.

Accumulator Overflow: Like the alteration switches, these keys latch when they are pressed and are turned off by being pressed again. Each key is lighted while it is on. The accumulator overflow keys are the means of determining whether an overflow in an accumulator will set the internal overflow indicator for that accumulator for later interrogation by the program, or will stop the machine as well, when it occurs. If the key for an accumulator is not pressed on, an overflow in that accumulator stops the machine. If the key is on (lighted), an overflow does not stop the machine, but turns on the overflow indicator for that accumulator. This indicator can be tested later in the program by a BV# instruction: +11 (BV1), +21 (BV2) *or* +31 (BV3).

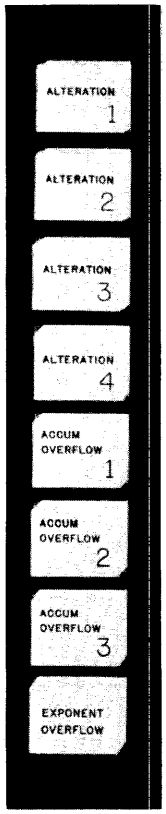


Figure 188. Alteration Switches and Overflow Keys

The accumulator overflow *lights* on the right of the panel are associated directly with the overflow *indicators* for the accumulators. Each light indicates that the corresponding overflow indicator is on. The light goes out when the indicator is turned off. Thus, an accumulator overflow light indicates either why the machine has stopped, if its accumulator overflow key-light is off, or that the program is continuing after it has caused an accumulator to overflow, if its accumulator overflow key-light is on.

Exponent Overflow: This key-light has the same function for exponent overflow in floating-decimal-point arithmetic operations that the accumulator overflow keys have for normal overflow. Exponent overflow occurs when an operation attempts to develop a characteristic greater than 99.

Control Lights

The lights on the operating panel are called control lights, rather than display lights, because they do not display the contents of storage words or registers. They indicate the status that the system is in, the components that are in operation, and the type of error detected by the checking features of the system.

Figure 189 shows all of the control lights on; normally, only a few of them are on.

PRIORITY STATUS, NORMAL STATUS, INQUIRY STATUS

The system is always in the normal or priority status. In normal status, the main program is operating. In priority status, a priority routine is functioning. Inquiry is

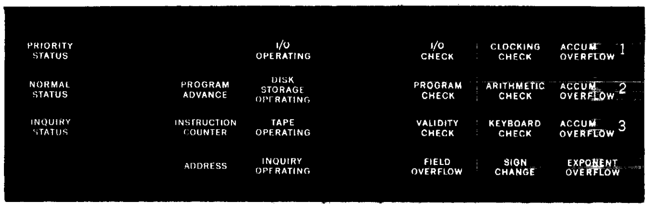


Figure 189. Control Lights

the status that is set by pressing the INQUIRY ONLY key on the operating keyboard. In normal status, an inquiry from any one of the inquiry stations sends a priority signal, and the program branches to location 0106 or 0107, depending on the inquiry control group. In inquiry-only status, the program is not operating. An inquiry starts its inquiry routine at word 0106 or 0107.

OPERATING

The operating lights indicate which types of components are put into operational status. There are five component types: the programming unit, the unit-record machines, the disk units, the tape units, and the inquiry stations. The appropriate light turns on briefly when an instruction involving the operation of one of the units of that type is sent to the program register and executed.

These lights serve their main purpose when an operation involving a particular type of unit cannot be executed. This may be due to machine error, programming error, or normal operations (the card reader having run out of cards, for example).

Program Advance: This light serves to inform the console operator of the status of stored program execution. The light may be on, off, or blinking. When on, the light indicates that program execution is occurring normally; when off, it shows that execution has stopped. The blinking condition shows either (1) that program execution has stopped with an instruction only partially executed, or (2) that the operation called for by an instruction has taken more than two seconds to complete (e.g., a long typing operation). When the program advance light is blinking, pressing the stop key on the operating keyboard turns off the light. If the blinking occurs because an operation takes more than two seconds to complete, pressing the stop key turns off the light and, when the operation is finally completed, causes the console typewriter to type out the contents of the instruction counter and program register.

Instruction Counter, Address: Whenever the stored program stops, one of these lights is on. If the next instruction in the stored program is to come from the instruction counter, the instruction counter light is on. This light is on even if the next instruction is one or two locations beyond the next sequential location, as in a UR operation not involving validity-check or end-of-file. The address light is on if the next instruction is to come from the address in the instruction, in a branch operation. This could be an unconditional branch, or the result of a test, such as BM3 when accumulator 3 is minus.

I-O Operating: This light comes on whenever an operation involving a unit-record machine is sent to the program register. The codes are: UR, UW, UWIV, UP, UPIV, US, and TYP.

Disk Storage Operating: This light flashes whenever a disk seek, read or write instruction is sent to the program register (codes ±91, ±92, and +95). The code for release (−95, DAR) has no effect.

Tape Operating: Any instruction that involves the operation of a tape unit turns on this light when it is in the program register. The light remains on until the tape unit has completed the operation.

Inquiry Operating: Whenever a +54 inquiry-control instruction (QR or QW) is in the program register, this light comes on. The light also turns on when the inquiry keyboard is in use. The light turns off at completion of a request or reply operation.

CHECK

A checking light comes on whenever the system stops processing, because an automatic checking feature has discovered an error. The error may be caused by incorrect programming, operation, control-panel wiring, or by the machine itself.

I-O Check: An error in operation of any of the machines that transmit data to and from the synchronizer drum causes this light to come on, and the computer to stop. Human errors that cause this include failing to fill the input synchronizer because of improper control-panel wiring; or improper alphabetic-input wiring, detected by the automatic alphabetic check. Machine checks include validity check on transmission of data between the synchronizer drum and the synchronizer register, or any check on the operation of the input-output units.

Program Check: Detection of a program control error causes this light to turn on. A program control error may be the result of:

1. An edit-numeric-to-alpha instruction whose RDW (record definition word) specifies an *odd* number of alphameric words.
2. An instruction (other than tape or disk storage) whose RDW contains a start address greater than the stop address.
3. Any instruction whose RDW has an alpha sign.
4. An indexed instruction whose associated index word has an alpha sign.
5. Division overflow.
6. A tape control instruction referring to a tape unit that has experienced an error condition when either (a) the tape unit stacking latch is on because of a previous operation with the unit but the latch is masked, or (b) the computer is in the interrupt mode.
7. An instruction using field control when position 4 of field control is greater than position 5.
8. An improper or illegal data address.

9. The sign control instruction (−03) when position 4 of the instruction contains a digit other than 3, 6, or 9.
10. A machine error when executing a table lookup instruction.
11. An index-word instruction that contains the digits 00 in positions 4 and 5 of the instruction.
12. An out-of-range shift operation.

Validity Check: The detection of an invalid character, on any transfer of data that the stored program must complete before being able to continue, turns on this light. This includes flow of data to and from magnetic-core storage, to and from the registers and accumulators, and to and from the synchronizers on the drum. (Validity checks on data flow to and from the magnetic-tape units and the disk files are indicated by the final-status words. Validity checks between the synchronizer drum and the card unit are indicated by the I-O check light.)

Field Overflow: Whenever a field overflow occurs on a store or add-to-storage type of operation, the field-overflow indicator is turned on, and the machine either continues or stops, depending on the setting of the internal field-overflow stop-sense switch. Whenever the field-overflow indicator is on, regardless of the setting of the stop-sense switch, the field overflow console light is on. Thus, it indicates either why the machine has stopped (switch at stop) or that the program is continuing after it has caused a field overflow (switch at sense). The instruction BRANCH IF FIELD OVERFLOW (BFV), turns off the indicator if it is on, and turns off the field overflow light as well.

Clocking Check: This light indicates a failure in the internal timing circuits of the computer. It denotes a machine error only, not a programming or operating error. The correction procedure after a clocking check is important, however. In all cases, the program should be restarted back at some previous point. A clocking error may affect totals that are being updated or processed, and the operator must restart the program at a point that again starts development of those totals.

Arithmetic Check: This light indicates an error in the arithmetic circuits. It may be a machine or programming error. As with the clocking check, restart after detection of this type of error should return the program to a point that starts the development of the totals involved when the arithmetic error was detected.

Keyboard Check: This light is used to indicate an error in operation of the console typewriter keyboard. When the console typewriter is being used to enter data manually into the computer, only the numeric keys can be used. (Alpha data must be entered in the two-digit numeric code.) If an alpha key is pressed while the console is in manual-entry status, the keyboard check light comes on.

Sign Change: In the same manner as field overflow, this light indicates that the sign-change indicator is on; a sign change has resulted from a store or add-to-storage type of operation. The BRANCH IF SIGN CHANGE instruction (BSC), turns off the indicator and the sign change light.

Accumulator Overflow and Exponent Overflow: The function of these lights was explained in the section "Alteration Switches and Overflow Keys."

ADDRESS STOP

Four dials for address stop are located in the center of the operating panel. Each dial has a digit setting (0-9) and can be turned to any digit (Figure 190).

The operator can have the program stop at a desired address by setting the address stop dials at that address and pressing the address stop key on the operating key-

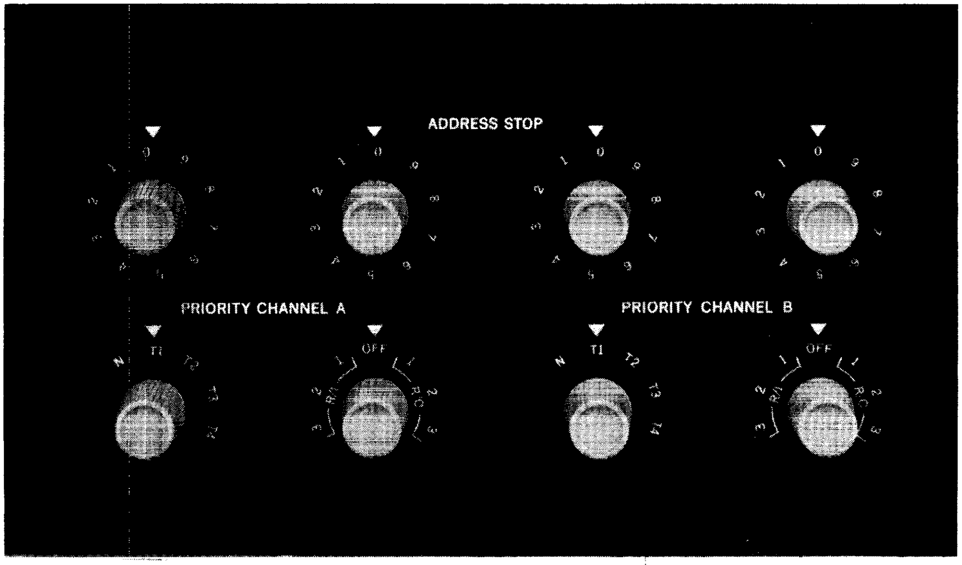


Figure 190. Address Stop and Unit-Record Priority Controls

board. The stored program continues normally until it comes to that address, for either instruction or data. If it stops because of an instruction address, the stop occurs before that instruction takes place. If the stop results from a data address (positions 6-9), it comes immediately after the instruction involving the address has been executed. In all cases of address stop, the address is the one after indexing, if indexing is used.

With the index-word codes, the operand index word (specified by positions 4-5 of the instruction), causes an address stop if its address is set in the dials (0001-0099). For a code whose address represents the first of a series of record-definition words, address stop is effective for all generated addresses. Address stop also is effective in operations that control the electronic switches in locations 0101, 0102, and 0103.

If positions 6-9 of an instruction are the same as the address in the address stop dials, the program stops only if positions 6-9 actually represent a usable address. For example, a halt and proceed (−00, HP) instruction cannot cause the program to stop on address stop, because positions 6-9 are not used for an address. Similarly, the index-word codes that use positions 6-9 as a 4-digit factor cannot cause an address stop.

PRIORITY CHANNEL A, PRIORITY CHANNEL B

These dials are the means of assigning unit-record priority controls A and B to the desired card readers, punches, or printers. The right-hand dial under each control (A and B) has seven positions, one designating each of the six (maximum) unit-record machines, and one for OFF. When a unit-record machine is thus designated, the completion of a cycle (read, punch, or print) automatically sets the priority stacking latch for unit record A or B. If the stacking latch has not been masked, the program branches to a subroutine when the unit-record cycle has been completed. (See section "Priority Processing.")

The left-hand dial under each control (A and B) has six positions. Positions T1, T2, T3, and T4 represent the four tape channels; position N represents no (none) tape channel. If the dial is set at one of the four tape channel positions, the unit-record stacking latch (A or B) is turned on when the unit record device indicated by the right-hand dial has completed a cycle and will be honored only when the specified tape channel is not busy. This arrangement increases the efficiency of card-to-tape and tape-to-printer priority operations. The SET position is used to turn on the unit record stacking latch at the beginning of the priority operation. After the operation has begun, the dial is turned back to position N, T1, T2, T3, or T4. It is not necessary to stop the computer in order to start a unit-record priority operation.

Console Typewriter

The typewriter on the IBM 7150 Console is used for five different purposes:

1. Typing output data under control of the stored program.
2. Automatic display of the contents of the instruction counter and program register when the stored program stops.
3. Displaying the contents of a core-storage word or addressable register.
4. Changing the contents of a core-storage word or addressable register.
5. Manual typing, independent of the rest of the system.

The flow of data between console typewriter and core storage is as follows: typed information is recorded in the synchronizer register as each digit is typed (Figure 3). A word is moved from the synchronizer register to the arithmetic register and thence to core storage. Data from core storage to the console typewriter are transmitted to the arithmetic register, thence to the synchronizer register, and from there to the typewriter.

A typewriter interlock is provided to prevent a data movement to the synchronizer if the typewriter has not completed printing on a previous type operation.

Forty-four characters are available for typing: 26 letters, ten digits, and eight special characters: plus sign (+), minus sign (−), comma (,), period (.), slash (/), asterisk (*), dollar sign ($), and pound sign (#). Any character other than these 44 is typed as a pound sign (#). A maximum of 75 characters can be typed on a line.

As shown in Figure 191, the keyboard is the same as that of other IBM electric typewriters. The alphabetic keys are used only when typing independently of the rest of the system (except the A for an alpha sign, as described under "Display"). The pin-feed platen enables multi-carbon paper forms to feed accurately.

TYPEWRITER OPERATIONS UNDER PROGRAM CONTROL

Data are transmitted from core storage to the console typewriter under control of record-definition words, as described under "Block Transmission." The stored program continues with the next instruction after the typing operation is completed. (See TYPE instruction.)

An automatic carriage return is performed before starting printing when a TYP command is executed. This assures printing starting in position 1.

Typing is in sequence, starting from the high-order position of the first word defined by the first RDW and continuing to the units position of the last word defined by the last RDW. Carriage return at the end of each line is automatic.

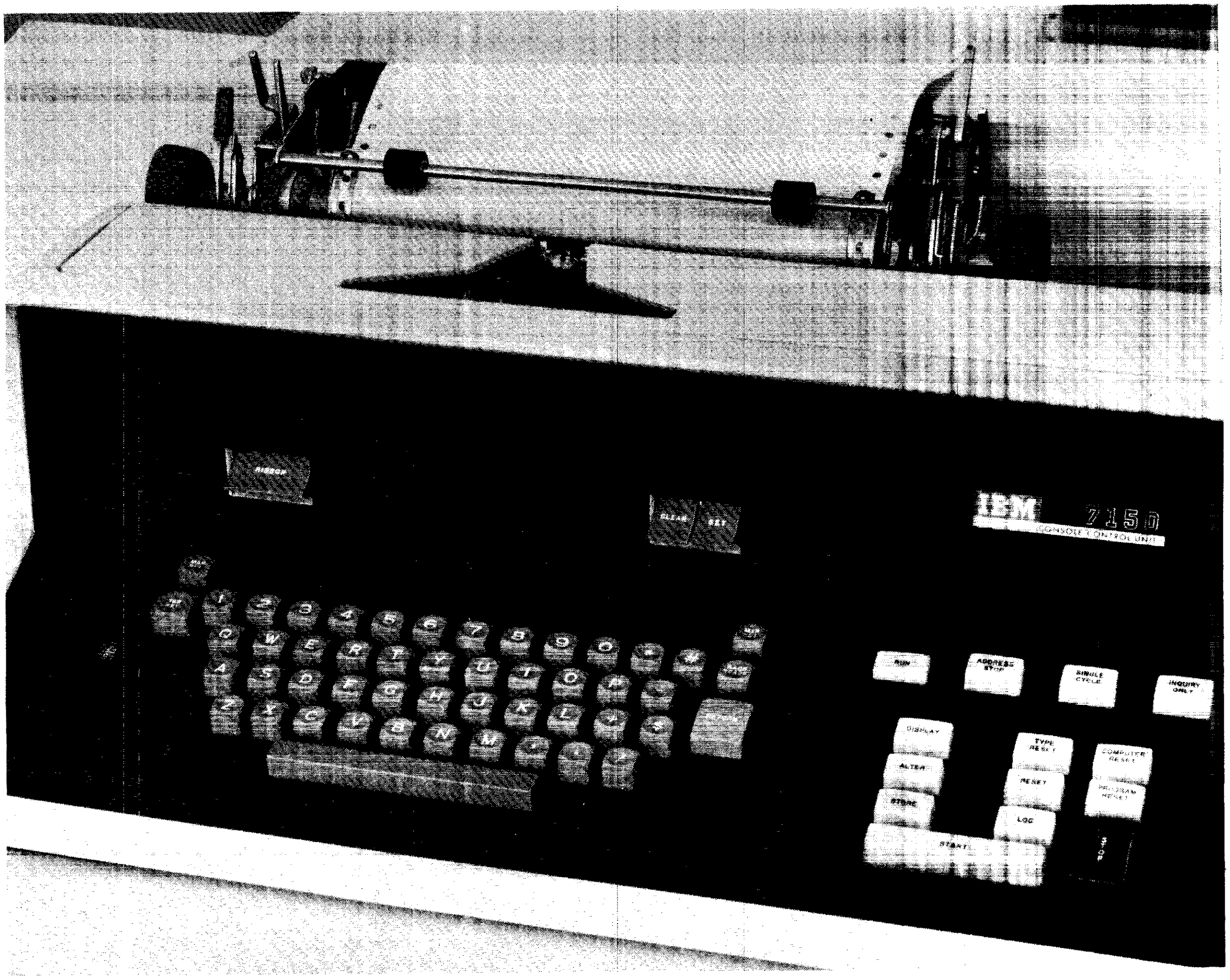


Figure 191. Console Typewriter and Operating Keyboard

The sign position of a word is scanned first, and on numeric words, a dash is printed for negative words and a plus sign for positive words. An alpha sign is neither printed nor spaced but sets up circuits for conversion of two-out-of-five code to alpha.

If the word is numeric, the sign and digit positions 0-9 are typed in that order. A space follows position 9. Thus, a series of numeric words are printed in 11-character groups (sign and ten digits), separated by single spaces.

On alpha words the converted alpha characters are printed in sequence beginning with positions 0-1. The contents of a second alpha word are printed directly following the previous one, with no spaces for sign. Thus, alpha data words are printed out in a continuous series. Characters coded 00 in alpha words cause a space.

Operating Keyboard

The operating keyboard (Figure 191) is located to the right of the console-typewriter keyboard. It contains the console operational features that are used for program testing or for operator control of data and instructions. For the purpose of explanation, the keys are grouped in three categories:

1. *Status Keys:* Run, Address Stop, Single Cycle and Inquiry Only.
2. *Control Keys:* Start, Stop, Computer Reset, Reset and Program Reset.
3. *Operation Keys:* Display, Alter, Store, Log and Type Reset.

Status Keys

The status keys are located at the top of the operating keyboard. They are the latch type—at all times, one is in its pressed position and the other three are not. When one of these keys is pressed, the key that had been pressed previously automatically resets to its OFF position. Thus, the computer is always in *run, address-stop, single-cycle,* or *inquiry-only status.*

Run: When this key is pressed, the stored program runs continuously in a normal manner. This can be considered "normal" status, the one used almost exclusively once the program has been tested and is performing on a regular basis.

Address Stop: In this status, the stored program runs continuously until it uses the address set on the address stop dials in the operating panel. This address can be for either instruction or data; the stop occurs when the address enters the program register, from either the instruction counter or storage (after indexing). If it is an instruction address, the program step is the next one to be executed in the program. If it is the address of data, the stop occurs immediately *after* that instruction has been executed. When an address stop occurs, the contents of the instruction counter and program register are automatically typed.

Single Cycle: Each pressing of the program start key causes the stored program to execute one instruction if the computer is in this status. After the instruction is completed, the contents of the instruction counter and program register are automatically typed. Note: It is not possible to enter the priority mode when the machine is in single-cycle status. Attempting to do so will force the stored program into a "loop," for which there is no error indication. To single-cycle through the priority routine, the operator should use the address-stop technique to enter the priority routine from the main routine by address-stopping the machine at the first instruction of the priority routine.

Inquiry Only: When this key is pressed, the stored program does not function, but an inquiry can be processed. A priority signal from an inquiry station starts an inquiry subroutine at either 0106 or 0107, depending on the inquiry-control group involved. The following is the sequence of operations for using the inquiry-only feature.

1. Stop the stored program by pressing the stop key (contents of the instruction counter and program register are typed.)
2. Press the inquiry-only key. The inquiry-only status light on the operating panel turns on.
3. Press the start key. The computer goes into an idle condition, the program-advance light flashes on and off, and the inquiry mask latches are set initially to allow interruption.
4. Release of an inquiry causes the contents of the instruction counter to be stored in positions 2-5 of index word 97, the inquiry routine to start in 0106 or 0107, and the priority status light on the operating panel to come on. (The instruction counter contains the location of the next stored program instruction, as typed in paragraph 1.)
5. If the priority-release instruction at completion of the inquiry subroutine has an address of 0097, the address in positions 2-5 of that word is returned to the instruction counter, the computer returns to the idle condition, and the priority status light on the operating panel turns off. If the address of the PR instruction is other than 0097, the address itself goes to the instruction counter.

When the machine is taken out of the inquiry-only status, and the program is started again, it starts at the point where it had left off (assuming that the priority routine did not change positions 2-5 of index word 97), whether or not any inquiry subroutine had been processed.

Control Keys

The control keys are used to start and stop the program and reset the various registers and error-detection circuits prior to starting. The start and stop keys are at the lower portion of the keyboard. Computer reset and program reset are above the stop key, and reset is beside the program reset key.

Start: Pressing this key starts the stored program. The first instruction is taken from the location in the instruction counter at that time. In single-cycle status, pressing this key causes one program instruction to be executed.

Stop: Pressing this key stops the stored program. The instruction being executed at the time is completed, and the contents of the instruction counter and program register are automatically typed. The stop key is also used to stop the program advance light from blinking.

Reset: The function of the reset key is to reset a detecting circuit that has stopped the computer. This may be due to a machine error, or to a situation that has arisen in the stored program which caused the program to stop. Any time that one of the checking lights on the operating panel comes on, the computer stops. (The functions of these lights are explained in the section "Operating Panel.") Pressing the reset key turns off the light and resets the error-detecting circuits except when program execution has stopped but the computer remains in the run mode. This condition is signalled by the blinking on and off of the program advance light, and requires that the stop key be pressed before pressing the reset key. The program can be continued from that point by pressing the start key.

As described in the "Operating Panel" section, use of the accumulator overflow key determines whether an accumulator overflow will set an indicator and stop the machine, or merely set the indicator and allow the program to continue. If an overflow stops the machine, pressing RESET turns off the indicator and allows the program to continue when the start key is pressed. The exponent overflow indicator, used in floating-decimal-point operations, also has a stop-sense key on the operating panel, and RESET has the same function for this indicator as for accumulator overflow.

The sign-change and field-overflow indicators are made to stop the machine or not when they are turned on by stored program instructions, rather than by keys on the operating panel. The function of the reset key in the case of these two indicators, however, is the same. If the sign-change indicator, for example, is in the halt mode and a sign change occurs, the machine stops, and pressing RESET turns off the indicator. RESET does not change the indicator to the sense mode; only the SMSC instruction can do that. Field overflow works in the same way as sign change. Note that the sequence in which an indicator is put in the halt mode (HMSC, HMFV) and set, could be reversed. An indicator can be in the sense mode, and on; if it is then put into the halt mode, the machine stops, the program check light comes on, and RESET must be used.

The high, low, and equal indicators are reset by pressing the reset key, if none of them, two of them, or all three are on. For the normal condition, one of them on, RESET has no function. (A compare operation automatically resets all these indicators and then turns on one of them as a result of the comparison.)

Computer Reset: The computer reset key resets the program controls and all registers. After this key has been pressed, the program should be started over again, because virtually all of the results of the program are destroyed. Figure 192 lists in tabular form the effects on the machine devices (latches, switches, registers, counters, etc.) of pressing the computer reset, program reset, and reset keys.

Program Reset: The program reset key resets the machine in much the same way as the computer reset key, but to a lesser degree. Refer to Figure 192 for a comparison of the effects on machine devices between program reset and computer reset.

Reset, Computer Reset, and Program Reset Effect on the Branch Latch: A branch or no-branch condition is determined during the execution of each instruction. A branch condition causes the branch latch to be set on.

This branch or no-branch condition is indicated on an error stop by the control lights on the 7150 console. An instruction counter light indicates a no-branch condition; an address light indicates a branch condition.

If the address light indicates a branch condition on an error stop, use of the computer reset, program reset, or reset key turns the branch latch off.

The instruction counter must be reset to obtain the desired branch. The address is provided by the error stop print-out in positions 6-9 of the program register.

Operation Keys

These keys are the means by which an operator can examine the contents of any core-storage location, by having it typed on the console typewriter. He can also change a storage word by using these keys.

The display, alter, and store keys are operative only when the stored program is *not* running. These keys are located on the left of the operating keyboard. The type reset key is above the reset key. The log key, which can be used when the program is running or not, enables the operator to use the console typewriter without affecting the rest of the system. It is located below the reset key.

To type out the contents of a storage word, the operator merely types the four-digit address. If the operator did not hit tab or carriage return first, this address is typed directly to the right of the PR contents that were typed when the program was halted. There is an automatic tab, and the contents of that address are typed (directly below the contents of the program register, if tab or carriage return had not been used). If the word is numeric, it is typed with sign (+ or −) followed by ten digits. If the word is alpha, it typed with an "A" for sign, followed by the five alphameric characters.

LOCATION (typed by operator)		CONTENTS
1234	TAB	±1234512345 (if numeric) CR
2345	TAB	AABCDE (if alphameric) CR

There is an automatic carriage return after the word is typed.

Display: The display key is used to display an alphameric word in its ten-digit core-storage notation, after it has been displayed as five characters. Assume, for example, that the five characters JAN 15 in word 8245 have just been displayed:

8245 TAB AJAN15 CR

The operator presses the display key. There is an automatic tab, and the ten digits of word 8245 are typed, preceded by a "3" for alpha:

TAB 37161759195 CR

Alter: When a word has been displayed, it can be changed by the operator if he first presses the alter key. This causes an automatic tab and a ribbon shift to red, and the operator types the sign and ten digits that are to go into that word. The complete word must be typed in numeric digits. An alpha word need not be displayed numerically but can be altered numerically only. If the operator wishes to alter the contents of the IC (instruction counter) when the machine has stopped with the address light (operating panel) on, pressing the alter key turns off the light so that the next instruction will be taken from the IC.

Store: After visually checking what he has typed, the operator presses the store key, and the new data are

DEVICE	CONDITION OF DEVICE AFTER PRESSING KEY		
	Computer Reset	Program Reset	Reset
Instruction Counter	Filled with zeros	Filled with zeros	NC (No Change)
Program Register	Filled with zeros	Positions 6-9 filled with zeros	"
Accumulator 1	" " "	NC (No Change)	"
Accumulator 2	" " "	"	"
Accumulator 3	" " "	"	"
Input/Output Error Latch	Set OFF	"	"
Program Error Latch	" "	Set OFF	Set OFF
Validity Error Latch	" "	" "	" "
Clocking Error Latch	" "	NC	NC
Arithmetic Error Latch	" "	Set OFF	Set OFF
Keyboard Error Latch	NC (No Change) ④	NC ④	NC ④
Field Overflow Indicator Latch	Set OFF	Set OFF if STOP/ SENSE sw was at STOP	Set OFF if STOP/ SENSE sw was at STOP
Field Overflow Stop/Sense Switch	Set to STOP	NC	NC
Sign Change Indicator Latch	Set OFF	Set OFF if STOP/ SENSE sw was at STOP	Set OFF if STOP/ SENSE sw was at STOP
Sign Change Stop/Sense Switch	Set to STOP	NC	NC
Accumulator 1 Overflow Indicator Latch	Set OFF	Set OFF ⑤	Set OFF ⑤
Accumulator 2 Overflow Indicator Latch	" "	" " "	" " "
Accumulator 3 Overflow Indicator Latch	" "	" " "	" " "
Low Indicator Latch	" "	" " ①	" " ①
Equal Indicator Latch	" "	" " ②	" " ②
High Indicator Latch	" "	" " ③	" " ③
Floating Point Overflow Indicator Latch	" "	" " ⑤	" " ⑤
Floating Point Underflow Indicator Latch	" "	NC	NC
Input/Output Check Light	Turned OFF	"	"
Program Check Light	" "	Turned OFF	Turned OFF
Validity Check Light	" "	" "	" "
Field Overflow Light	" "	Turned OFF if STOP/ SENSE sw was at STOP	Turned OFF if STOP/ SENSE sw was at STOP
Clocking Check Light	" "	NC	NC
Arithmetic Check Light	" "	Turned OFF	Turned OFF
Keyboard Check Light	NC ④	NC ④	NC ④
Sign Change Light	Turned OFF	Turned OFF if STOP/ SENSE sw was at STOP	Turned OFF if STOP/ SENSE sw was at STOP
Accumulator 1 Overflow Light	" "	Turned OFF ⑤	Turned OFF ⑤
Accumulator 2 Overflow Light	" "	" " "	" " "
Accumulator 3 Overflow Light	" "	" " "	" " "
Exponent Overflow Light	" "	" " "	" " "
(All) Stacking Latches	Set OFF	NC	NC
Interrupt Mode Latch	" "	"	"
Priority Mask Register	Filled with 1's	"	"
Magnetic Tape Units	Reset	Reset	Reset
Disk Storage Units	"	NC	NC
Inquiry Stations	"	"	"
Input/Output Synchronizers	Erased	"	"
Branch Latch	Set OFF	Set OFF	Set OFF

NOTES:

① if Equal and/or High latches were also ON

② if Low and/or High latches were also ON

③ if Low and/or Equal latches were also ON

④ use STOP key to reset

⑤ if associated console switch is ON (set to STOP)

Figure 192

stored in the core-storage word. If there were no errors such as validity errors, a red pound sign (#) is typed at the right of the ten digits, and a carriage-return is executed. If invalid characters are present in the information that is displayed, bit typing occurs and the validity check light on the operating panel turns on. It is not necessary to turn off the validity check light with the reset key if the operator intends to alter the contents of the location. Storing valid information in the location automatically turns off the light. The bit typing that occurs when invalid information is displayed is the same bit typing as that described for the TYPE operation.

Log: When the stored program is not operating, and the typewriter is to be used for normal typing, press this key. This prevents the automatic display of the storage word the address of which is the first four digits typed, or keyboard error if an alpha key is pressed. Typing can continue until the stop key is pressed, resetting the typewriter to normal condition.

When the stored program is operating, manual typing can be performed without pressing the log key. If a stored-program TYP instruction is given while manual typing is taking place, the program instruction takes precedence, and the typewriter is removed from manual control. After the programmed type operation is completed, the operator can again resume manual typing.

The alphabetic keys can be used manually only after pressing the log key. Any other use of the alpha keys results in a keyboard error (except the A for an alpha sign in an alter operation).

Type Reset: Any time that the typewriter is typing from a TYP instruction, pressing this key stops the typing operation at the end of the word being typed. A red asterisk is typed, the carriage returns, and any interlocks set up by the typing operation are released. The contents of the instruction counter and program register are then typed.

Checking

The checking features of the IBM 7150 Console are in two main categories: checks on machine operation (validity checks) and checks on manual operation of the typewriter keyboard. An error in keyboard operation may be detected by the machine or by the operator himself. The transmission of data from the computer to the console does not, of course, involve operator errors but is checked for validity. All of the checking features are discussed here in the sequence of operations to display, alter, and store a word of data. Validity checking of all movement of data from the computer to the console typewriter is then discussed in a section by itself.

In all cases of an error being made when the operating keyboard and the typewriter keyboard are used to display and/or alter a storage word, the operation must be started over again. This is done by first pressing the stop key, to reset the error-detection circuits, and again displaying the storage word by typing its address. In keying in an address, or altering a word, only the numeric keys on the typewriter can be used. If an alphabetic key is pressed in either of these operations, the keyboard locks, and the stop key must be pressed.

If the operator discovers that he has made an error in typing the address of a word to be displayed, he presses the stop key. There is an automatic carriage return, and a red asterisk types. He then can attempt again to type the correct address. The error must be detected before the last digit of the address has been typed; once four digits have been keyed in, the contents of that word are immediately and automatically typed.

The operator may discover that he has made an error in typing the sign and ten digits of a word in an alter operation. He presses the stop key, the typewriter automatically types a red asterisk, and the carriage returns. The operator must then start over by keying in the four-digit address of the word to be altered.

A typing error undetected by the operator may occur if less than or more than ten digits and sign are typed in an alter operation. In either case, pressing the store key does not store the typed data in the storage location. Pressing the stop key then causes typing of a red asterisk and a carriage return.

The movement of data from the arithmetic register to core storage when the store key is pressed is checked for validity. If an invalid character is detected, the validity check light on the operating panel turns on, and the reset key must be pressed.

Executing an Instruction from the Console

It is often desirable, in program testing, to create an instruction from the console and then have the stored program execute it. This is done as shown in Figure 193. If the instruction counter contains an address of 9995, it does not increase by +1 with the execution of each instruction; it stops at 9995. This situation applies only to those instructions which, after execution, cause +1 to add to the instruction counter. Thus, if the program is in *run* status, it continuously performs a one-instruction loop. If the status is *single-cycle,* the instruction is executed once for each pressing of the start key.

xxxx TAB	±xxxxxxxxxx	Contents of instruction counter and program register automatically typed on machine stop.
9995	±xxxxxxxxxx	Display the program register, repeating the type-out of its contents.
	±yyyyyyyyyy#	Alter the program register, by pressing Alter, typing the sign and ten digits, and pressing Store. #Indicates it was stored correctly.
9999	xxxx	Display the instruction counter. (Note only 4 digits, with no sign, are typed.)
	9995#	Alter the instruction counter, giving it the address of the program register. #Indicates it was stored correctly. Press the Start key.

Figure 193. Executing an Instruction from the Console

Console Operation — 7074 System

Some manual console operations are different on the 7074 than on the 7070. This section presents correct 7074 procedures.

Memory Test

1. Depress computer reset key (NOTE: if any other reset key is used, stacking latches must first be masked to avoid interrupts during the memory test).
2. Key operation code into program register (location 9995).
3. Depress memory test key on the CE console.
4. Depress start key (in either single cycle or run mode). The instruction counter (IC) will be incremented by one as each op code is executed and will specify the location operated upon.

Instruction Counter Operations

An op code may be executed from the program register (PR) in the following manner:

1. Depress reset key.
2. Key op code into 9995.
3. Key 9995 into instruction counter (location 9999). After this step, the reset key should not be depressed until the IC operation is completed.
4. Depress start key (in either single cycle or run mode). The op code keyed into 9995 will be executed once in single cycle and continuously in run mode.

To return to the program after an IC 9995 operation is completed:

1. Depress reset key (to reset IC 9995 controls).
2. Key desired start address in IC (location 9999).
3. Depress start key.

Execution of RDW codes that use IC 9995 are as follows:

1. Depress reset key.
2. Key op code into 9995.
3. Depress single cycle key (an RDW code is valid for only one execution from the program register for each time it is keyed in. This is due to the program register being changed during the execution of the op code).

Type-out on Stop

A type-out on stop for RDW codes will vary from the type-out obtained on the 7070. Positions 2-5 and/or 6-9 of the program register will be changed. The positions changed will be dependent on the RDW code printed out. The storage location should be displayed for the correct op code.

On error type-outs for card and inquiry operations, the instruction counter will be incremented by one or three, depending on when the error occurs in relationship to the RDW cycles of the instruction.

Stop on Error

Following a stop on error, no op code, either manually inserted or program-contained, can be executed with a HSP error condition showing. If an error condition exists, the reset key or check disable key must be depressed prior to start key depression.

Check Disable

The check disable key on the CE console can be used to bypass information errors only. An attempt to bypass a program check error or MAB error will result in erroneous indications.

Address Stop Mode

In address stop mode, the 7074 will stop at the completion of the op code if the address switch setting is the same as *any* address used during the op code execution with the exception of the index word address in positions 2-3 in the instruction.

Timing Loops

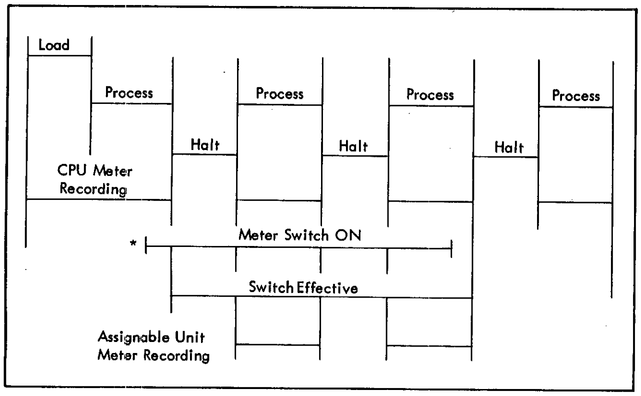
A timing loop of specific duration on the 7070 may have been used to time an I-O operation. Upon completion of the timing loop, a previously initiated tape operation would be completed and ready for interrogation of tape status words. If the same routine is used on the 7074, the faster execution of the timing loop will not allow sufficient time for the operation; tape status words will not be ready for interrogation.

System Reaction and Recovery Techniques—Metered Machines

An IBM data processing system consists of a CPU cluster, assignable units, and I/O units. The customer usage meter on assignable units (control units, channels, files, and drums) is controlled by an enable switch. This switch is effective only during a CPU halt condition (see Figure 193A) and should be on when the program is started. If the enable (meter) switch is off the assignable unit and any attached I/O units are disabled, and the CPU hangs up when the program selects a disabled unit. The following procedure may be used to recover from such a condition:

SYSTEM REACTION	RECOVERY TECHNIQUE
7907 Data Channel: Data channel will flag an error, terminate the operation, and force an interrupt. A condition code one (1) is stored in the final status word which may be interrogated by the program. *Note:* The first instruction through the 7907 to an assignable unit (7631, 7640) on that channel will hang-up the channel only. With the channel hung up in a busy status, the CPU can: (1) Hang in a busy loop. (2) Hang up if the program selected the channel.	Stop the system, turn on the meter switch, and restart the system. *Note:* "Restart the system" means that the same precautions should be taken as in recovering from a CPU hang-up due to any other cause.
1301 (7070/74), 1302 (7074): An access inoperative condition is indicated in the sense status word.	Stop the CPU program, turn on the meter switch, and restart at the point in the program where the stop occurred.
*7603 *I/O Synchronizer:* If the 7603 is addressed in the off-line condition, the system hangs-up.	Stop the system, change the switch setting to on-line, and restart the system. *Note:* The on-line/off-line switch also functions as the meter switch.
*7604 *Tape Control:* Channel busy is indicated and prevents execution of a tape instruction. If the unit is addressed, the system will hang-up. (A channel can be interrogated under program control for the busy condition, and a branch executed.)	Stop the system, turn on the channel enable switch (meter switch), and restart the system.
*7605 *Disk Storage Control:* If the unit is addressed in the off-line condition, the system hangs-up.	Stop the system, change switch setting to on-line, and restart the system. *Note:* The on-line/off-line switch also functions as the meter switch.

*Any attached I/O unit which is not ready when addressed by the program causes the system to hang-up until the I/O unit is made ready.



* The chart indicates that it is not mandatory for the meter switch to be on before starting the program. If a program halt precedes the program selection of the assignable unit, the meter switch may be turned on any time prior to or during the halt.

Figure 193A. Meter (Enable) Switch Operation

Customer Engineering Console

The Customer Engineering (CE) Console is used by IBM customer engineers in diagnostic testing and preventive maintenance. With it, the customer engineer is able to monitor and analyze the performance of the entire system. Certain features of the console reveal in minute detail the conditions that occur during the execution of a stored program. For example, if the system stops with the program check light on (7150 Console), indicator lights on the CE console reveal what condition has caused the stop. By using the CE console to supplement the 7150 Console, an operator can determine the cause of error conditions more quickly and with more certainty than otherwise. During normal operation all key-type switches should be set to the NORMAL position (top half of key in the depressed position). Also, the 729 magnetic tape and 7300 disk storage panels have control dials which must be set for on-line operation.

Figure 194 shows the four separate panels that make up the 7070 CE console:

1. Arithmetic-and-Programming and Storage Panel
2. Magnetic Tape Panel
3. Unit Record Panel
4. Disk Storage and Inquiry Panel

On the 7074 system a High-Speed Processor and Storage Panel is used in place of the Arithmetic-and-Programming and Storage Panel.

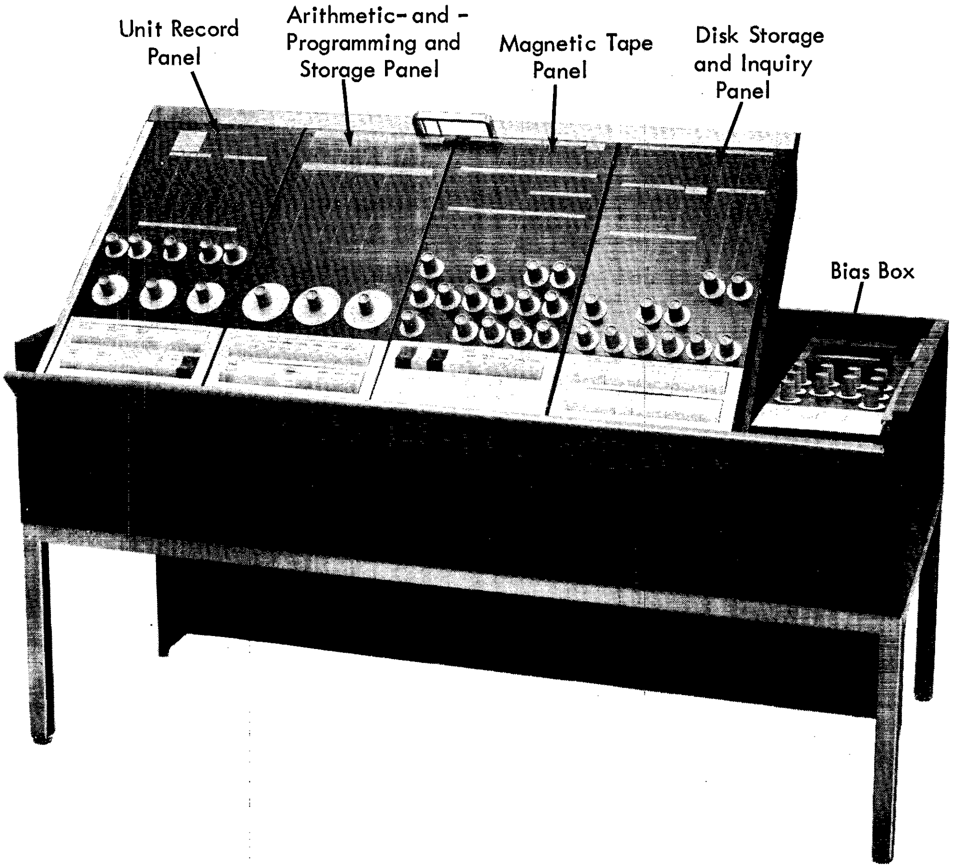

Unit Record Panel
Arithmetic-and-Programming and Storage Panel
Magnetic Tape Panel
Disk Storage and Inquiry Panel
Bias Box

Figure 194. IBM 7070 Customer Engineering Console Panel

Arithmetic-and-Programming and Storage Panel for 7070

Figure 195 shows the 7070 Arithmetic-and-Programming (A-P) and Storage Panel of the CE console. For descriptive purposes, the features of the panel are grouped in three main sections: indicator lights, dial-type selector switches, and key-type switches.

Address and Data Lights

These lights (Figure 195-1) display in 2-of-5 bit code the state of core storage address and data triggers. The information to be displayed is selected by the dial switches below the indicator lights.

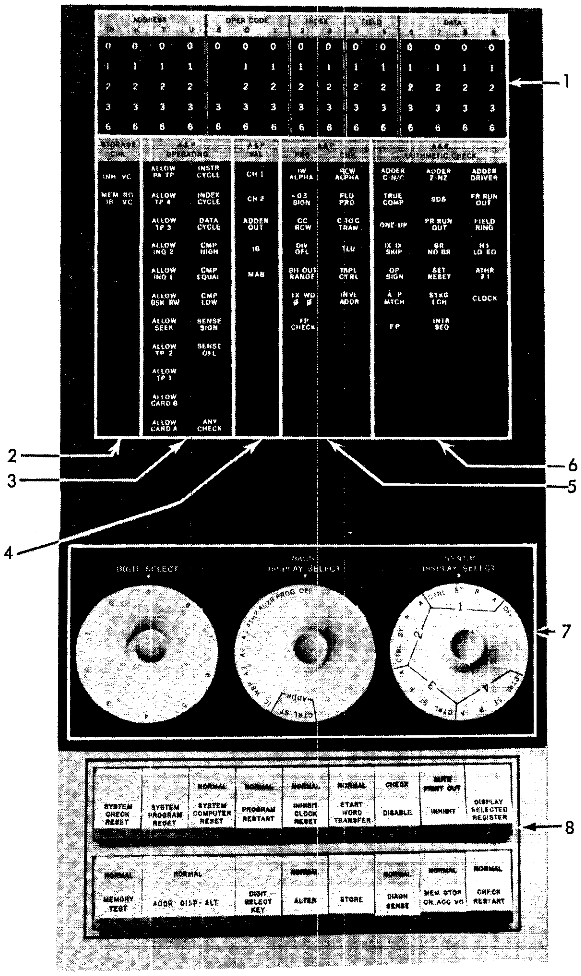


Figure 195. IBM 7070 Arithemetic-and-Programming and Storage

Check and Operating Lights

Storage Check Lights (Figure 195-2)

INH VC (Inhibit Validity Check) indicates a 2-of-5 validity check on information read into storage. It can be on only during a storage cycle. The computer stops when this light turns on.

MEM RO IB VC (Storage Read-Out Information Bus Validity Check) indicates an invalid bit combination in information read out of storage at the request of the arithmetic and program control unit.

ADDR CK (Address Check) indicates an invalid storage address. The computer stops when this light turns on, and no further requests of storage can be made.

A-P Operating Lights (Figure 195-3)

ALLOW PA TP (Allow Paper Tape)

ALLOW TP 4 (Allow Tape Channel 4)

ALLOW TP 3 (Allow Tape Channel 3)

ALLOW INQ 2 (Allow Inquiry Control 2)

ALLOW INQ 1 (Allow Inquiry Control 1)

ALLOW DISK R/W (Allow Disk Storage Read/Write)

ALLOW SEEK (Allow Disk Storage Seek)

ALLOW TP 2 (Allow Tape Channel 2)

ALLOW TP 1 (Allow Tape Channel 1)

ALLOW CARD B (Allow Unit Record Priority Control B)

ALLOW CARD A (Allow Unit Record Priority Control A)

The above 11 lights display the settings of the priority mask register.

INSTR CYCLE (Instruction Cycle) indicates that the program controls are performing the instruction cycle.

INDEX CYCLE indicates that the program controls are performing the indexing cycle.

DATA CYCLE indicates that the program controls are performing the data cycle.

CMP HIGH (Compare High)

CMP EQUAL (Compare Equal)

CMP LOW (Compare Low)

The three COMPARE INDICATOR lights display the settings of the low, equal, and high indicator latches.

SENSE SIGN indicates the setting of the sign change sense-stop switch. When the light is on, the switch is set at SENSE.

SENSE OFL (Sense Overflow) indicates the setting of the field overflow sense-stop switch. When the light is on, the switch is set at SENSE.

ANY CHECK indicates that an error condition exists in the arithmetic and programming area of the machine.

A-P Validity Lights (Figure 195-4)

CH 1 (Channel 1) indicates an invalid combination of bits on adder entry channel 1. The computer stops when this light turns on.

CH 2 (Channel 2) indicates an invalid combination of bits on adder entry channel 2. The computer stops when this light turns on.

ADDER OUT indicates an invalid combination of bits on the adder output channel. The computer stops when this light turns on.

IB (Information Bus) indicates an invalid combination of bits on the information bus at the time of a request from the arithmetic and programming control unit for use of the bus. The computer stops when this light turns on. This check also occurs when a +55 PC or PC1 instruction encounters a digit other than 0 or 1 in a mask word.

MAB (Storage Address Bus) indicates an invalid combination of bits on the storage address bus at the time of a request from the arithmetic and programming control unit for use of the bus. The computer stops when this light turns on.

A-P Program Check Lights (Figure 195-5)

IW ALPHA (Index Word Alpha) indicates that an index word with alpha sign has entered the auxiliary register. The computer stops when this light turns on.

−03 SIGN indicates an illegal sign digit (other than a 3, 6, or 9) in the fourth position of the field register during execution of the sign control instruction. The computer stops when this light turns on.

CC RCW (Unit Record RDW) indicates that the start address of a record definition word is larger than the stop address in an operation involving unit record equipment. The computer stops when this light turns on.

DIV OFL (Division Overflow) indicates that the machine has attempted to develop a quotient of more than ten digits during a divide operation. The computer stops when this light turns on.

SH OUT RANGE (Shift out of Range) indicates that the machine has attempted to shift more than ten or twenty (double shift) positions during a shift control operation. The computer stops when this light turns on.

IX WRD 00 (Index Word 00) indicates that positions 4 and 5 of an instruction dealing with an index word (e.g., index word load, shift control) contain the digits 00. The computer stops when this light turns on.

FP CHECK (Floating Point Check) indicates that one of the following error conditions has occurred:

1. During a FLOATING ADD DOUBLE PRECISION operation, the storage word to be added is smaller than the 20-digit number in the accumulators.
2. During any floating-decimal-point operation, any one of the factors has an alpha sign.
3. During any floating-decimal-point operation, an accumulator address (9991-9993) is used.

The computer stops when this light turns on.

RCW ALPHA (Record Definition Word Alpha) indicates that the machine has attempted to use a record definition word which has an alpha sign. The computer stops when this light turns on.

FLD PRG (Field Program) indicates that the stop digit is smaller than the start digit in an operation using programmed field control. The computer stops when this light turns on.

C TO C TRAN (Core-to-Core Transfer) indicates that one of the following error conditions has occurred:

1. During an "edit" operation (EAN, ENA, ENB, ENS), the RDW (record definition word) start address is greater than or equal to the stop address or the start and stop addresses are not an even number of words apart.
2. During block transmission or table lookup operations, the RDW start address is greater than the stop address.
3. During a table lookup operation, the increment value in index word 98 is zero.

The computer stops when this light turns on.

TLU (Table Lookup) indicates failure of the "found table argument" latch to remain set after it has been set on.

TAPE CTRL (Tape Control) indicates that one of the following error conditions has occurred:

1. The 7070 is in the interrupt mode and the stored program calls on a tape unit whose stacking latch is set.
2. The stored program calls on a tape unit whose stacking latch is set and which is attached to a tape channel that is masked.

The computer stops when this light turns on.

INVL ADDR (Invalid Address) indicates that the machine has attempted to use an illegal address in executing an instruction. The computer stops when this light turns on.

A-P Arithmetic Check Lights (Figure 195-6)

ADDER C-N/C (Adder Carry-No Carry) indicates that both the carry and no-carry latches are set on during an operation involving the adder. The computer stops when this light turns on.

TRUE COMP (True Complement) indicates that both the true-add and complement-add latches are set on at the same time. The computer stops when this light turns on.

ONE UP indicates that both the zero-up and one-up latches are set on at the same time. The computer stops when this light turns on.

IX-IX SKIP (Index-Index Skip) indicates that both the index and index-skip program control latches are set on at the same time. The computer stops when this light turns on.

OP SIGN (Operation Sign) indicates that the operation register contains other than a plus or minus sign. The computer stops when this light turns on.

A-P MTCH (A-P Match) indicates that both the match and mismatch latches are set on at the same time. The computer stops when this light turns on.

ADDER Z-NZ (Adder Zero-Non-Zero) indicates that both the adder zero and adder non-zero latches are set either on or off at the same time. The computer stops when this light turns on.

SDS (Significant Digit Scanner) indicates that significant digits remain in the output from the auxiliary register during operations that involve field definition of an accumulator and the auxiliary register. The computer stops when this light turns on.

PR RUN OUT (Program Run Out) indicates that the program ring has not turned off in time. The computer stops when this light turns on.

BR-NO BR (Branch-No Branch) indicates that neither the branch nor no-branch latch is set on at the beginning of an instruction cycle. The computer stops when this light turns on.

SET RESET indicates that one of the following error conditions has occurred:

1. During a branch on accumulator overflow operation, the overflow latch is not set off.
2. During a store accumulator operation, the field overflow latch is not set on when the accumulator field is larger than that designated by programmed field control.
3. During a store accumulator operation, a carry is sensed as a result of the last digit of true add.

The computer stops when this light turns on.

STKG LCH (Stacking Latch) indicates that a stacking latch has not been set properly after execution of a STACKING LATCH SET or STACKING LATCH RESET instruction. The computer stops when this light turns on.

INTR SEQ (Interrupt Sequence) indicates that an error indication latch is set on at the time of initiation of interrupt sequence cycles. The computer stops when this light turns on.

ADDER DRIVER indicates that more than one adder driver is on at the same time for either adder entry channel 1 or adder entry channel 2. The computer stops when this light turns on.

FR RUN OUT (Field Ring Run Out) indicates that the field ring has not stopped by the end of the 12th position. The computer stops when this light turns on.

FIELD RING indicates that the field cycle ring has failed to turn on because of a field ring match condition. The computer stops when this light turns on.

HI LO EQ (High Low Equal) indicates that more than one of the high, low, or equal indicator latches is set on at the same time. The computer stops when this light turns on.

ARTH ZI (Arithmetic Zero Insert) indicates that the arithmetic zero insert circuits have failed to turn off in the correct sequence. The computer stops when this light turns on.

CLOCK indicates that one or more of the system clocks has stopped or is out of step with the others. The computer stops when this light turns on.

Dial-Type Selector Switches (Figure 195-7)

DIGIT SELECT: This switch is used to select the digit that is to be entered in a position of the address start register or storage word.

BASIC DISPLAY SELECT: This switch is used to select the machine component whose contents will be displayed in the address and data lights at the top of the panel. The switch has 11 positions:

OFF
PROG (Program Register)
AUXR (Auxiliary Register)
ATHR (Arithmetic Register)
A1 (Accumulator 1)
A2 (Accumulator 2)
A3 (Accumulator 3)
WBR (Synchronizer Register)
IC (Instruction Counter)
ADDR ST (Address Start Register)
ADDR CTRL (Address Control A Register)

SYNCR DISPLAY SELECT (Tape Channel Display Select): This switch is used to select the tape channel component whose contents will be displayed in the address and data lights at the top of the panel. The switch has 17 positions. One of these is the OFF position. The other 16 are grouped into four sections of four positions each. Each of the four sections represents a separate tape channel (1, 2, 3, or 4). The four positions within each section are:

A (Buffer A Register) (Transmission Register A)
B (Buffer B Register) (Transmission Register B)
ST (Start Address Register)
CTRL (Control Address Register)

Key-Type Switches (Figure 195-8)

Note that when the panel is not being used, all switches must be in their normal or off position.

SYSTEM CHECK RESET performs the same function as the reset key on the 7150 console.

SYSTEM PROGRAM RESET performs the same function as the program reset key on the 7150 console.

SYSTEM COMPUTER RESET performs the same function as the computer reset key on the 7150 console.

PROGRAM RESTART: Pressing this key and then the log key (7150 console) forces a system computer reset to take place every drum revolution. Following the A-P computer reset time, a program start is forced. This feature is for use only with the A-P unit. The maximum usable program time is approximately three milliseconds. The IB, MAB, and BR-NO BR error indicator lights are on until the log key is pressed.

INHIBIT CLOCK RESET: Pressing this key prevents all system clocks from resetting when any reset key is pressed. This feature is used primarily when magnetic tape units and unit record devices are being operated from the CE console.

START WORD TRANSFER: Pressing this key forces start word transfer so that the unit record controls can be checked when the program restart key is used.

CHECK DISABLE: When this key is in the DISABLE position, the machine will not stop when an error occurs.

AUTO PRINT OUT INHIBIT: When this key is in the INHIBIT position, the machine will not type out the contents of the instruction counter and program register when a stop occurs. This feature prevents the contents of the auxiliary and synchronizer registers from being disturbed when the stop occurs. The system remains in the run mode, so the stop key (7150 console) must be pressed before proceeding.

DISPLAY SELECTED REGISTER: Pressing this key causes information to be displayed in the address and data lights at the top of the panel. The information displayed is determined by the setting of the BASIC DISPLAY SELECT or SYNCHRONIZER DISPLAY SELECT dial-type switch.

MEMORY TEST (Storage Test): Pressing this key sets up conditions in the machine so that one can single-cycle through storage locations in much the same way as one single-cycles through a stored program. In the latter operation, each depression of the start key (7150 console) causes the contents of the IC (instruction counter) to be incremented by +1. In the storage test operation, the contents of the IC remain constant but the contents of program register D (positions 6-9 of a word) are incremented by +1 with each depression of the start key. Before depressing the start key one must:

1. Store the address of the program register (9995) in the IC (9999).
2. Load the program register with either an ST# (store accumulator number) or ZA# (zero accumulator number and add) instruction. If the ST# instruction is to be used in the test, then the designated accumulator must first be loaded with the desired value.

ADDR DISP-ALT (Address Display Alter): Pressing this key sets up the conditions necessary for displaying and altering the contents of core storage and addressable machine components in the address and data lights at the top of the panel. Once displayed, the contents can be altered by use of the digit select switch, digit select key, alter key, and store key. The procedure for displaying and altering the contents of a core storage location is as follows:

1. Press the ADDRESS DISPLAY ALTER key.
2. Select the high-order (thousands) digit of the storage address with the DIGIT SELECT switch, and then press the DIGIT SELECT key to enter the digit in the high-order position of the address start register. Repeat this procedure three times to enter the hundreds, tens, and units digits in the register. After the units digit has entered the register, the contents of the address will be displayed automatically in the address and data lights at the top of the panel.
3. Press the ALTER key.
4. With the DIGIT SELECT switch, select the digit that is to replace (or duplicate) the high-order digit displayed in the lights, and then press the DIGIT SELECT key. Working from high-order to low-order position, repeat this procedure as many times as necessary in order to replace or duplicate the digits.
5. Press the STORE key to enter the new contents in the address that was placed in the address start register in step 2 (above).
6. Restore the ADDRESS DISPLAY ALTER and ALTER keys to their normal positions.

DIGIT SELECT KEY: See step 2, above.

ALTER: See step 3, above.

STORE: See step 5, above.

DIAG SENSE (Diagnostic Sense): Pressing this key forces the stored program to continue with the next instruction in sequence after an error has occurred.

MEM STOP ON ACC VC (Storage Stop on Access Validity Check): When this key is pressed, sensing of an address or information validity check causes storage to lock up so that the address and data triggers contain the address and information in error.

CHECK RESTART: Pressing this key causes execution of stored program to stop. Any time an error occurs, a system computer reset occurs, and the stored program restarts automatically at location 0000. This feature is used with any operation except INQUIRY. (Note that a "hang-up" condition also causes program restart at location 0000.)

High-Speed Processor and Storage Panel for 7074

Figure 196 shows the 7104 High-Speed Processor (HSP) and Storage panel of the CE console. Indicator lights, dial-type selector switches, and key-type switches facilitate diagnostic testing and preventive maintenance.

Address and Data Lights

These lights (Figure 196-1) display the state of 7301 Core Storage address and data triggers in 2-of-5 code. The information to be displayed is selected by the dial switches located below the indicator lights.

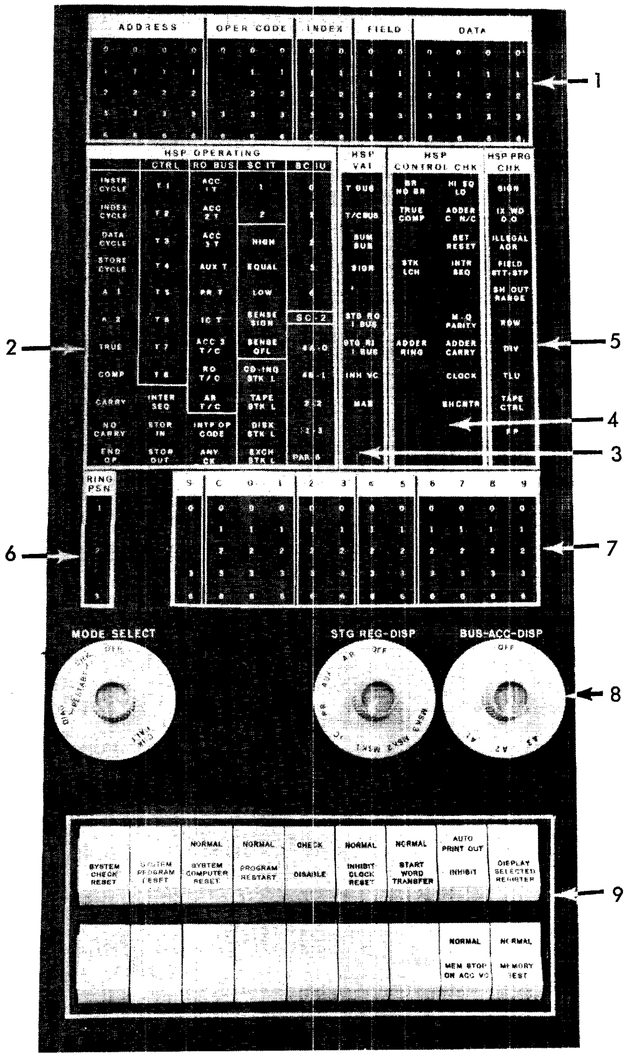


Figure 196. IBM 7074 High-Speed Processor and Storage Panel

HSP Operating Lights (Figure 196-2)

These lights indicate operating conditions of key control and storage triggers in the HSP.

INSTR CYCLE (Instruction Cycle) indicates that the program controls are performing the instruction cycle.

INDEX CYCLE indicates that the program controls are performing the indexing cycle.

DATA CYCLE indicates that the program controls are performing the data cycle.

STORE CYCLE indicates that the program controls are performing the store cycle.

A1 (Adder First) indicates that the adder ring is in its first position.

A2 (Adder Second) indicates that the adder ring is in its second position.

TRUE indicates that the true-add trigger is on.

COMP (Complement) indicates that the comp-add trigger is on.

CARRY indicates that the carry trigger is on.

NO CARRY indicates that the no-carry trigger is on.

END OP (End Operation) indicates that the end-operation trigger is on.

INTER SEQ (Interrupt Sequence) indicates that the program controls are sequencing into a "Priority Status."

STORAGE IN indicates a memory read-in request. It stays on until the request is answered.

STORAGE OUT indicates a memory read-out request. It stays on until the request is answered.

INTP OP CODE (Interpret Operation Code) comes on during the time the program controls are performing the operation. It stays on until the end of the operation.

ANY CHECK comes on whenever any validity, program, or arithmetic check occurs.

CTRL; T 1-8 (Control Triggers 1 through 8) indicate that the corresponding trigger is on.

RO BUS; ACC 1 T, 2 T, 3 T, AUX T, PR T, IC T (Read Out to True Bus – Accumulator 1, 2, 3, or Auxiliary, Program, or Instruction Counter Register) indicate that the corresponding register is reading out to the true bus.

RO BUS; ACC 3 T-C, RO T-C, AR T-C (Read Out to True Complement Bus – Accumulator 3, Auxiliary Register, or Arithmetic Register) indicate that the corresponding register is reading out to the TC bus.

CD-INQ STK L (Card or Inquiry Stacking Latch) indicates that a card I-O or manual inquiry stacking latch is on.

TAPE STK L (Tape Stacking Latch) indicates that a 729 tape synchronizer stacking latch is on.

DISK STK L (Disk Stacking Latch) indicates that a 7300 disk storage stacking latch is on.

EXCH STK L (Exchange Stacking Latch) indicates that an exchange channel control stacking latch is on.

HIGH (Compare High)
EQUAL (Compare Equal)
LOW (Compare Low)

The three compare indicator lights display the settings of the high, equal, and low indicator triggers.

SENSE SIGN indicates the setting of the sign change sense-stop switch. When the light is on, the switch is set at SENSE.

SENSE OFL (Sense Overflow) indicates the setting of the field overflow sense-stop switch. When the light is on, the switch is set at SENSE.

SC IT – 1, 2 (Shift Counter 1, tens position: 1 or 2) indicate the value in shift counter 1, tens position. A one or a two is indicated by the corresponding light. Zero is indicated by neither light being on.

SC IU – 0, 1, 2, 3, 6 (Shift Counter 1, units position value) indicate the value in shift counter 1, units position. A 0, 1, 2, 3, or 6 is indicated by the corresponding light in 2-of-5 code.

SC-2, 4A-0 comes on during the time that shift counter 2 contains a zero. For multiply-divide operations, it signifies a times-four multiple.

SC-2, 4B-1 comes on during the time that shift counter 2 contains a one. It also signifies a times-four multiple for multiply-divide operations.

SC-2, 2-2 comes on during the time that shift counter 2 contains a two. It also signifies a times-two multiple for multiply-divide operations.

SC-2, 1-3 comes on during the time that shift counter 2 contains a three. It also signifies a times-one multiple for multiply-divide operations.

SC-2, PAR-6 comes on during the time that shift counter 2 contains a six. It also signifies parity for multiply-divide operations.

HSP Validity Lights (Figure 196-3)

These lights come on during HSP operations only.

T BUS (True Bus) indicates an invalid combination of bits on the true bus during a read out to the true bus from a register. The computer stops when this light turns on.

T-C BUS (True Complement Bus) indicates an invalid combination of bits on the T-C bus during a read out to the T-C bus from a register. The computer stops when this light turns on.

SUM BUS indicates an invalid combination of bits on the sum bus during an add cycle. The computer stops when this light turns on.

SIGN indicates an invalid combination of bits in the sign position of a register. The computer stops when this light turns on.

STG RO I BUS (Storage Read Out Information Bus) indicates an invalid combination of bits on the memory information bus during a memory read out cycle. The computer stops when this light turns on. This check also occurs when a +55 PC or PC1 instruction encounters a digit other than zero or one in a mask word.

STG RI I BUS (Storage Read In Information Bus) indicates an invalid combination of bits on the memory information bus during a memory read in cycle. The computer stops when this light turns on.

INH VC (Inhibit Validity Check) indicates invalid 2-of-5 characters going through the inhibit drivers. The computer stops when this light turns on.

MAB (Memory Address Bus) indicates an invalid combination of bits on the memory address bus. The computer stops when this light turns on.

HSP Control Check Lights (Figure 196-4)

BR NO BR (Branch, No Branch) indicates that either none or both of branch or no branch triggers are on at the beginning of an instruction cycle. The computer stops when this light turns on.

TRUE COMP (True Complement) indicates that true add and complement add triggers are both on at the same time or that neither is on. The computer stops when this light turns on.

STK LCH (Stacking Latch) indicates that a stacking latch failed to obey a set-on instruction or failed to obey a reset-off instruction. The computer stops when this light turns on.

ADDER RING indicates that adder ring position 1 is also on in position 2 at the same time, or that neither is on. The computer stops when this light turns on.

HI EQ LO indicates that one and only one of the high, low, or equal triggers is not set. The computer stops when this light turns on.

ADDER C N-C (Adder Carry – No Carry) indicates that the zero position of the adder has set either none or both of the carry or no-carry triggers. The computer stops when this light turns on.

SET RESET indicates a check on the following:

1. Set reset of sign sense trigger.
2. Reset of sign change trigger.
3. Reset of field overflow trigger.
4. Reset of accumulator overflow triggers.

The computer stops when this light turns on.

INTR SEQ (Interrupt Sequence) indicates any check during sequencing into priority status. The computer stops when this light turns on.

M-Q PARITY (Multiplier-Quotient Parity) indicates an even number of bits in the MQ register during multiply-divide operations. The computer stops when this light turns on.

ADDER CARRY indicates an invalid output of the look-ahead carry feature of the adder. The computer stops when this light turns on.

CLOCK indicates that drum timing rings are not synchronized or that interrupt timing rings are operating incorrectly. The computer stops when this light turns on.

SH CNTR (Shift Counter) indicates an invalid 2-of-5 character in the shift register or both the 1 and 2 in the tens position are ON. The computer stops when this light turns on.

HSP Program Check Lights (Figure 196-5)

SIGN indicates one of the following:

1. Program register sign is alpha.
2. For sign control code −03, the PR (Program Register) fourth position contains other than a digit 3, 6, or 9.
3. The auxiliary register has read in an index word or an RDW word with an alpha sign.

The computer stops when this light turns on.

IX WD 00 (Index Word Zero) indicates that address 00 has been sensed in positions 4-5 of the PR. The computer stops when this light turns on.

ILLEGAL ADR (Illegal Address) indicates that an improper address has been programmed. The computer stops when this light turns on.

FIELD STT-STP (Field Start-Stop) indicates that the start digit is larger than the stop digit for codes that operate under field control. Also indicates invalid mask bits for read-in or read-out of the arithmetic register. The computer stops when this light turns on.

SH OUT RANGE (Shift Out of Range) indicates improperly programmed shift lengths. The computer stops when this light turns on.

RDW (Record Definition Word) indicates one of the following:

1. Start address larger than stop address for RDW codes.
2. On alpha-numeric conversion codes, the RDW specifies an uneven number of alpha words.
3. An RDW with an alpha sign.

The computer stops when this light turns on.

DIV (Divide) indicates that the machine has attempted to develop a quotient of more than ten digits during a divide operation. The computer stops when this light turns on.

TLU (Table Lookup) indicates that the increment for TLU in index word 0098 has been sensed as zero. The computer stops when this light turns on.

TAPE CTRL (Tape Control) indicates one of the following:

1. The program is in priority status and a 729 tape unit instruction has been given for a tape unit for which the stacking latch is already set.
2. The stored program calls on a tape unit whose stacking latch is set and that is attached to a tape channel that is masked.

The computer stops when this light turns on.

FP (Floating Point) indicates one of the following:

1. An alpha sign being used in one of the registers on a FP operation (except accumulator 1 on reset add-subtract).
2. On a floating add double, the exponent of the memory word is smaller than the exponent of accumulator 1.
3. During a floating-decimal-point operation, an accumulator address (9991-9993) is being used.

The computer stops when this light turns on.

Ring Position 1, 2, 3, Lights (Figure 196-6)

These lights indicate that the corresponding position of the instruction, data, index, or store ring is on.

Bus Display Lights (Figure 196-7)

These 58 lights display the contents of the true bus.

Dial Switches (Figure 196-8)

The following are mode select switch settings:

CHECK RESTART forces a computer reset and a start pulse when a validity, program, or arithmetic check is sensed.

DIAG RESTART (Diagnostic Restart): Upon sensing a validity, program, or arithmetic check – or a "hang" condition – this switch setting forces a computer reset without resetting the check triggers and the instruction counter. A start pulse is generated to enable the diagnostic program to continue.

CHECK HALT: This setting causes any check sensed to immediately halt the program and maintain all indicators and triggers in the state at which the check occurred.

STG REG DISP (Storage Register Display) is used in conjunction with the display selected register switch to display the corresponding register in the memory data triggers.

BUS ACC DISP (Bus Accumulator Display) is used in conjunction with the display selected register switch to display the corresponding accumulator in the true bus indicator.

CE Console Key Switches (Figure 196-9)

SYSTEM CHECK RESET performs the same function as the reset key on the operator's console.

SYSTEM PROGRAM RESET performs the same function as the prog reset key on the operator's console.

SYSTEM COMPUTER RESET performs the same function as the comp reset key on the operator's console.

PROGRAM RESTART causes a computer reset and forces a start pulse for each drum revolution. The log key on the operator's console must be depressed to enable this switch to function.

CHECK DISABLE permits error stops to be bypassed. Its primary use is to bypass data errors. Some program or arithmetic checks cannot be bypassed by the use of this switch.

INHIBIT CLOCK RESET prevents reset of the system clocks when a computer reset is generated.

START WORD TRANSFER forces a start word transfer pulse for +69 codes. It should be used in conjunction with the program restart switch.

INHIBIT AUTO PRINT-OUT blocks the automatic print-out after any stop.

DISPLAY SELECTED REGISTER is used in conjunction with STG REG DISP and BUS ACC DISP.

1. Causes the selected register to read out to the memory data triggers for display purposes.
2. Causes accumulators to read out to the true bus for display on the true bus indicators.

MEM STOP ON ACC VC (Memory Stop on Access Validity Check) allows memory to lock up on sensing a validity check so that the address and data triggers retain the information in which the validity check occurred.

MEMORY TEST permits read-in or read-out operations to be cycled continuously through each address in memory. This procedure is entirely independent of the stored program. Prior to depressing the start key with this memory test switch on, address 9995 must be loaded with a store to memory code and the specified accumulator set to any desired value, or address 9995 may be loaded with a reset add code. Depressing the start key under these conditions forces the instruction counter to be incremented by one after each operation and used as a data address instead of an instruction address.

Magnetic Tape Panel

The 729 magnetic tape panel (Figure 197) was designed for a twofold purpose: as a neon display for programmer's use and as a customer engineering tool.

When used for neon display, the magnetic tape panel is located in the CE console, next to the 7150 Console.

When it is used as a customer engineering tool, the magnetic tape panel is removed from the CE console and plugged into the core control unit (7602). Because of the test control logic built into the 7602 and 7604 units, the CE is able to perform all basic tape operations. This provides customer engineering personnel with the following advantages:

1. Enables the 729 tape system to be removed from the rest of the system for servicing.
2. Performs the various 729 tape operations without requiring use of the logic in A-P unit.
3. Confines testing procedures and servicing techniques to a small machine area.

The CE operations are designed to use the normal control circuity within the tape channel. All tape operations are performed in much the same manner as the normal instructions. The differences are in the use of the CE address and information gates, and limiting the storage requests to the 7604 unit.

The following sections explain the various groups of indicators, selection switches, and control switches.

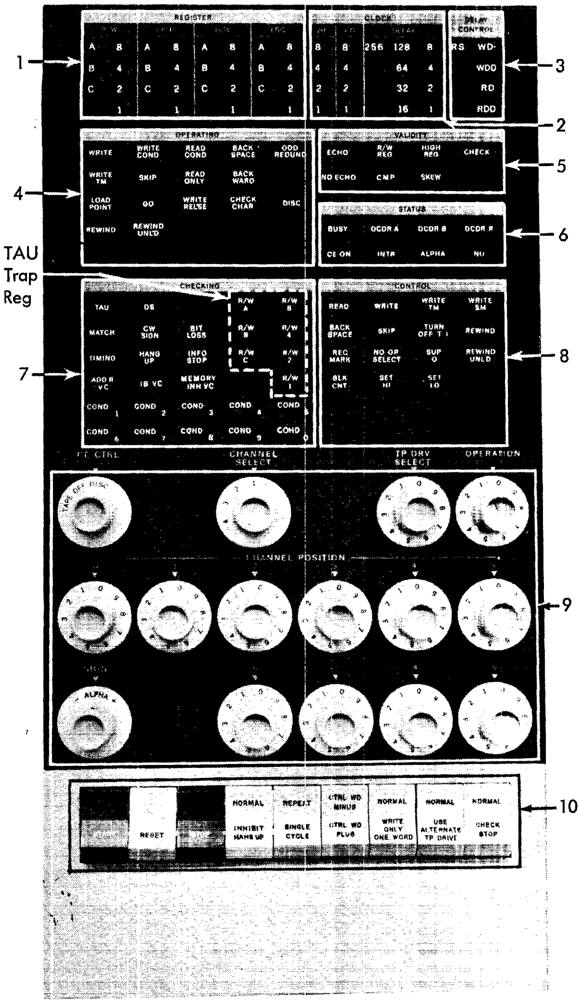


Figure 197. IBM 729 Magnetic Tape Panel

Register Lights (Figure 197-1)

R-W. (Read-Write): The R-W register lights display the bit combinations that have been read into the register from either the high or low registers (skew registers).

HIGH: The high-register lights (skew register A) display the bit combinations that are available during a read or write operation.

LOW: The low-register lights (skew register B) display the bit combinations that are available during the read or write operations.

LRC (Longitudinal Redundancy Check): The longitudinal redundancy check lights display the write triggers that are not reset if an LRC error has been detected.

Clock Lights (Figure 197-2)

WR, RD: The WR (write) and RD (read) lights display how far the write or read clock has advanced in the event of an error.

DELAY: The delay lights display how far the various tape delays have progressed in the event of a delay error.

Delay Control (Figure 197-3)

The DELAY CONTROL lights, along with the CLOCK DELAY lights, are used to determine what type of delay was being performed and how far the delay progressed.

Operating Lights (Figure 197-4)

The OPERATING lights are used to display the various TAU (Tape Adapter Unit) triggers that are conditioned to perform the specific tape operations.

Validity Lights (Figure 197-5)

The VALIDITY lights display the various TAU errors that can be detected during a read or write operation.

Status Lights (Figure 197-6)

The STATUS lights are used to display the status of the particular 7604 channel that is being used.

Checking Lights (Figure 197-7)

The CHECKING lights display the various 7604 channel errors that can occur during a 729 tape read or write operation The checking lights also display the condition latch that has been set and stored in the final status word at the completion of a tape operation. The lights enclosed with dotted lines represent the TAU trap register, which works in conjunction with the longitudinal redundancy check register to "trap" invalid 729 tape characters.

Control Lights (Figure 197-8)

The CONTROL lights display the various control latches that are set to perform a specific 729 tape operation.

Dial-Type Selector Switches (Figure 197-9)

CE CTRL (Customer Engineering Control): The CE control switch is used to initiate the CE ON operation and develop the TAPE OP or DISK OP *call.* The switch turned to either position interlocks the particular tape channel.

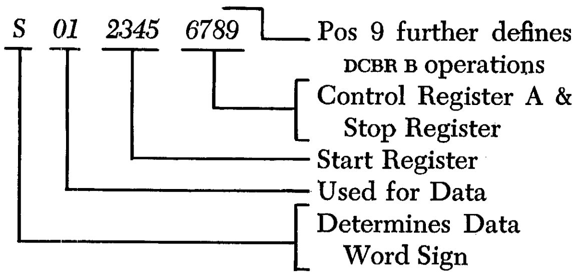
CHANNEL SELECT: This switch selects the individual 7604 tape channel for indicator display. When used during CE operation, the channel select switch should be turned to the selected channel *before* placing the CE control switch to the TAPE or DISK position.

TP DRV SELECT (Tape Drive Select): The 729 tape drive select switch specifies the tape unit that is to be used when reading or writing from the CE console. This switch simulates position 4 of the tape instruction.

OPERATION: The operation switch simulates position 5 of a normal 729 tape instruction. This switch specifies the tape operation that is to be performed and also determines whether decoder A or B is to be used. With the selection switch setting on zero, the decoder B latch is turned on. This signifies that position 9 of the tape instruction further defines the tape operation. The selection switch setting on any other position develops decoder A, along with the tape operation that is to be performed.

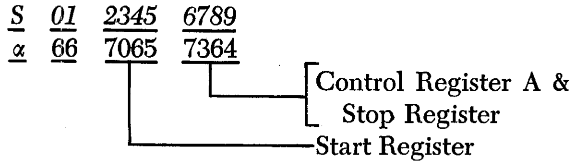
CHANNEL POSITION: Channel position switches 0-9 are used to simulate a ten-digit data word on the information bus. Positions 2-9 are also used for determining the control word (record definition word) and the control register A setting. Under this condition, both the control register A and the stop register have the same value.

1. CE WORD FORMAT (Channel Position Switches)


S 01 2345 6789
Pos 9 further defines DCBR B operations
Control Register A & Stop Register
Start Register
Used for Data
Determines Data Word Sign

If an alpha word is to be written, positions 2-9 are also used to develop the control word and the control register A address.

2. ALPHA WORD FORMAT (Channel Position Switches)



If the operation switch is set on zero, position 9 of the alpha word designates a backspace operation because of the 4 in position 9.

SIGN: The sign switch is used to insert the sign value into the sign position of the buffer B register. This sign is used to determine the sign of the data word. The normal sign analysis for determining the control word sign and the interrupt mode is inhibited during a CE operation.

Key-Type Switches (Figure 197-10)

START: Pressing the start key sets the CE start latch. The start latch is used to develop a CE A-P address and information gate to start the CE operation.

RESET: Pressing the reset key causes a channel reset to be developed which resets the selected 7604 tape channel.

STOP: Pressing the stop key sets the CE stop latch. With the CE stop signal available, the control word plus latch is turned off and the control word minus latch turned on. The 729 tape operation continues until the start and stop registers develop a match signal. The stop latch can also be set automatically if an error condition is sensed or a select and TI ON signal is developed.

INHIBIT HANG-UP: With the key set to the NORMAL position, all synchronizer control type errors will cause the channel to hang-up at completion of the operation. With the key set to the INHIBIT HANG-UP position, channel busy is allowed to be reset at completion of the operation, regardless of the type of error.

REPEAT-SINGLE CYCLE: With the key in the REPEAT position, the 729 tape operation occurs repeatedly until the CE stop latch is set on. With the key in the SINGLE CYCLE position, the tape operation occurs only once.

CTRL WD MINUS, CTRL WD PLUS (Record Definition Word Minus, Record Definition Word Plus)

1. PLUS POSITION: With a control word plus sign available, the 729 tape read or write operation continues until the CE stop latch is turned on. If the start and stop registers develop a match, a control word cycle is initiated to place the contents of the channel selection switches into the start and stop registers. This allows a continuous read or write operation to be performed.
2. MINUS POSITION: With a control word minus sign available, the 729 tape read or write operation continues until start and stop registers develop a match signal. With a match signal and the minus sign, the end of the operation is signaled and a final status word cycle is initiated. Whether the CE operation is to be automatically repeated is dependent upon the setting of the repeat-single cycle switch.

WRITE ONLY ONE WORD: When this key is pressed, a 729 tape unit writes only one word on tape regardless of the start and stop values selected by channel position switches 2 through 9.

USE ALTERNATE TP DRIVE: When this key is pressed, tape operations alternately occur between whatever tape unit is indicated by the tape drive select switch and tape unit 0.

CHECK STOP: With the key set to the CHECK STOP position, all repetitive CE 729 tape operations are prevented if other than error condition 2 is sensed.

Unit Record Panel

Figure 198 shows the Unit Record Panel of the CE console. For descriptive purposes, the features of the panel are grouped into three main sections: indicator lights, dial-type selector switches, and key-type switches.

Indicator Lights

RBS′, RBS, and PBS (Figure 198-1): These lights display the information on the drum at the positions selected by the SECTOR, DISPLAY, WORD, and DIGIT dial-type switches.

WORD CHECK (Figure 198-2): These lights specify in binary code the particular word in error when the machine stops because of invalid information. Note that the machine stops for an error condition only if the designated STOP-ON-CHECK switch (Figure 198-9) is on.

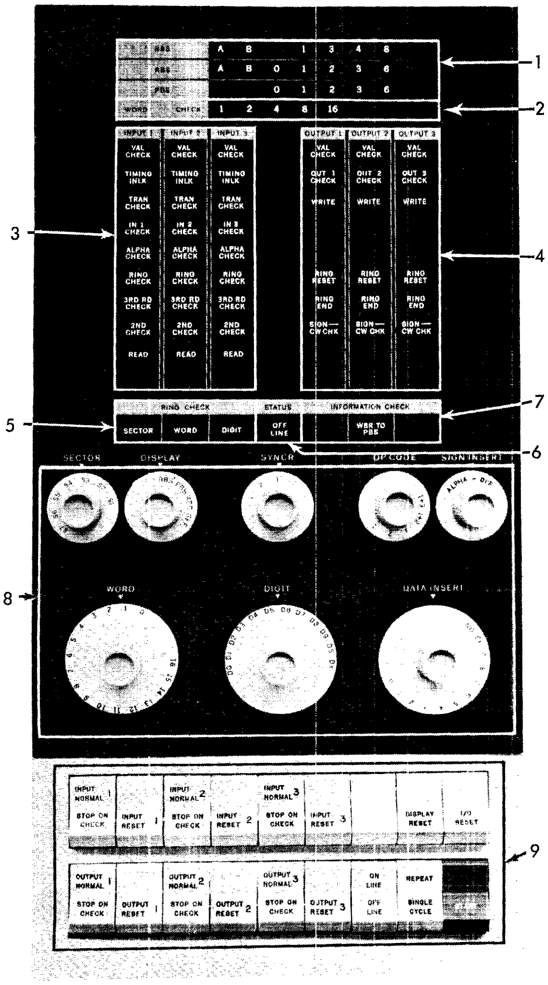


Figure 198. Unit Record Panel

Input 1, Input 2, Input 3

The following indicator lights (Figure 198-3) display the status of the three input synchronizers:

VAL CHECK (Validity Check) indicates that the number of bits transferred from the "7-to-2-of-5" translator to the word buffer register (synchronizer register) was not two. This light must be reset manually if the INPUT # STOP ON CHECK key-type switch is on.

TIMING INLK (Timing Interlock) indicates that the transfer and erasing of information from one card was not completed before information from the next card began entering the same area of the drum. This light must be reset manually.

TRAN CHECK (Transfer Check) indicates that one of the following conditions has occurred:

1. A "next read-in error" was on at the start of a transfer.
2. An RBS vs RBS′ error, validity error, or machine error when the INPUT # STOP ON CHECK key-type switch is on.

This light must be reset manually.

IN 1 CHECK, IN 2 CHECK, IN 3 CHECK (Input Number Check): Each light is equivalent to the clocking light of the card reader that is attached to the corresponding synchronizer. The light indicates the occurrence of a third-reading ring check, timing interlock error, or ring check during a start transfer.

ALPHA CHECK indicates that the alpha information read from second reading brushes into RBS′ and transferred to RBS did not match the information read directly from third reading brushes into RBS.

RING CHECK indicates that the second or third reading rings were not reset to zero-zero at the beginning of a scan or that the second reading ring carry latch was not turned on when the ring failed to advance the required 80 times.

3RD RD CHECK (Third Read Check): This light is equivalent to the ring check light of the card reader. It indicates that either (1) the third reading ring on the reader control panel is wired improperly or (2) the ring has functioned incorrectly with respect to the number of digits to be read.

2ND CHECK indicates a transfer error, alpha check, validity check, or input number check when the INPUT number STOP ON CHECK key-type switch is on.

READ indicates that the corresponding card reader is in the process of reading or that the input synchronizer has received an instruction to read but cannot respond.

Output 1, Output 2, Output 3

The following indicator lights (Figure 198-4) display the status of the three output synchronizers:

VAL CHECK (Validity Check) indicates an error in the information from PBS (Punch Buffer Storage) or from the data latches when printing or punching alpha information.

OUT 1 CHECK, OUT 2 CHECK, and OUT 3 CHECK (Output Number Check): When its corresponding OUTPUT STOP ON CHECK key-type switch is on, each light indicates for the corresponding output synchronizer an error condition associated with WBR to PBS, ring reset, sign CW check, ring check, or validity check.

WRITE indicates that the corresponding output device is in the process of printing or punching or that the output synchronizer has received an instruction to print or punch but cannot respond.

RING RESET indicates that the output rings were not reset to zero-zero or that the rings at latch 97 were reset before a read-out scan.

RING END indicates that the output rings were not driven to zero-zero and that the rings at latch 97 were turned on during a read-out scan.

SIGN CW CHECK indicates that the output rings were not driven to 37 during sign analysis or to 07 during control time.

Ring Check

The following lights (Figure 198-5) specify the particular ring in the drum timing circuits that has malfunctioned:

SECTOR, WORD, DIGIT: These lights can be reset only with the computer reset key on the 7150 Console.

Status (Figure 198-6)

OFF LINE indicates that all critical lines between the input-output devices and CPU are tied down so that, in effect, the devices are isolated from CPU control.

Information Check (Figure 198-7)

WBR TO PBS indicates the transfer of invalid information from WBR to PBS.

Dial-Type Selector Switches (Figure 198-8)

SECTOR: This dial is used to select the sector of the drum that is to be displayed by the indicator lights. The dial has seven positions, one for each sector of the drum.

DISPLAY: This dial is used to select the track of the drum that is to be displayed by the indicator lights. The dial has four positions: RBS′, PBS, RBS, and OFF.

SYNCR (Synchronizer): This dial is used to select the input-output synchronizer for "off-line" operation. The dial has three positions, one for each input-output synchronizer.

OP CODE (Operation Code): This dial is used to select the desired operation code for reading or writing when the input-output devices are to be operated in the off-line status. The five positions of the dial are: 1, unit record read: 2, unit record write or punch: 3, unit record write or punch invalid; 1 + 2, unit record read-write-punch with automatic stop on error; and 1 + 3, unit record read-write-punch invalid.

SIGN INSERT: This dial is used to select the sign that is to be inserted on an output synchronizer. The dial has three positions: −, ALPHA, and +.

WORD: This dial is used to select the word that is to be displayed by the indicator lights. The dial has 16 positions, one for each word.

DIGIT: This dial is used to select the digit within a word that is to be displayed. The dial has twelve positions: D0-D9, DS, and DX.

DATA INSERT: This dial is used to select a character that is to be inserted on a synchronizer. The dial has twelve positions: 0-9, EX (extra bits), and NO (no bits).

Key-Type Switches (Figure 198-9)

The following switches have two settings, NORMAL (off) and ON.

INPUT 1 STOP ON CHECK, INPUT 2 STOP ON CHECK, INPUT 3 STOP ON CHECK: When one of these keys is set to STOP ON CHECK, the error check indicator lights (Figure 198-3 and 5) for the corresponding input synchronizer are effective, and the machine will stop whenever an error occurs. The stop-on-check condition prevents the RBS track from being erased when an error occurs, so that the information can be displayed with the dial-type display switches.

INPUT 1 RESET, INPUT 2 RESET, INPUT 3 RESET: When an error condition occurs on an input synchronizer, pressing one of these keys resets the error latches for the corresponding synchronizer.

DISPLAY RESET: Pressing this key turns off the following display lights if they are on: RBS′, RBS, and PBS.

I-O RESET (Input-Output Reset): Pressing this key resets all unit record input-output controls.

OUTPUT 1 STOP ON CHECK, OUTPUT 2 STOP ON CHECK, OUTPUT 3 STOP ON CHECK: When one of these keys is set to STOP ON CHECK (on), the error check indicator lights (Figure 198-4 and 7) for the corresponding output synchronizer are effective, and the machine will stop whenever an error occurs. The stop-on-check condition prevents the PBS track from being erased when an error occurs, so that the information can be displayed with the dial-type display switches.

OUTPUT 1 RESET, OUTPUT 2 RESET, OUTPUT 3 RESET: When an error condition occurs on an output synchronizer, pressing one of these keys resets the error latches for the corresponding synchronizer.

ON LINE, OFF LINE: When the key is set to ON LINE, the unit record device is under control of the 7070. When the key is set to OFF LINE, the device is under control of the unit record CE panel. Note that the IBM 7900 Inquiry Stations are inoperative when the key is set to OFF LINE.

REPEAT, SINGLE-CYCLE: When this key is set to REPEAT during off-line operations, a single depression of the start key (below) causes the operation to be executed repetitively. When the key is set to SINGLE CYCLE, the start key must be held in the depressed condition for continuous operation.

START: Pressing this key initiates a unit record operation (read, punch, print).

Disk Storage and Inquiry Panel

Figure 199 shows the Disk Storage and Inquiry panel of the CE console. For descriptive purposes, the features of the panel are grouped into two main sections: IBM 7300 Disk Storage (no longer in production) and IBM 7900 Inquiry.

7300 Disk Storage

Selected Address Lights (Figure 199-1)

These lights indicate the 7300 disk storage address (access arm, unit/module, disk, and track) selected by the dial-type selector switches (Figure 199-6). The lights are operative only during reading and writing — not during seeking. The ACCESS and MODULE lights display the corresponding access arm and module numbers in digital form; disk and track numbers are in 2-of-5 bit code.

Indicate Free Access Lights (Figure 199-2)

These lights indicate which access arm in each disk storage unit is in the process of seeking or has been reserved for seeking.

Operating Lights (Figure 199-3)

These lights indicate for each of the two disk channels (synchronizers) the status of reading and writing. Writing status is specified by a combination of two write routine lights as follows:

WR ROUT 1 and WR ROUT 3: write burst status.
WR ROUT 1 and WR ROUT 4: write record status.
WR ROUT 2 and WR ROUT 3: OFF status.
WR ROUT 2 and WR ROUT 4: write check status.

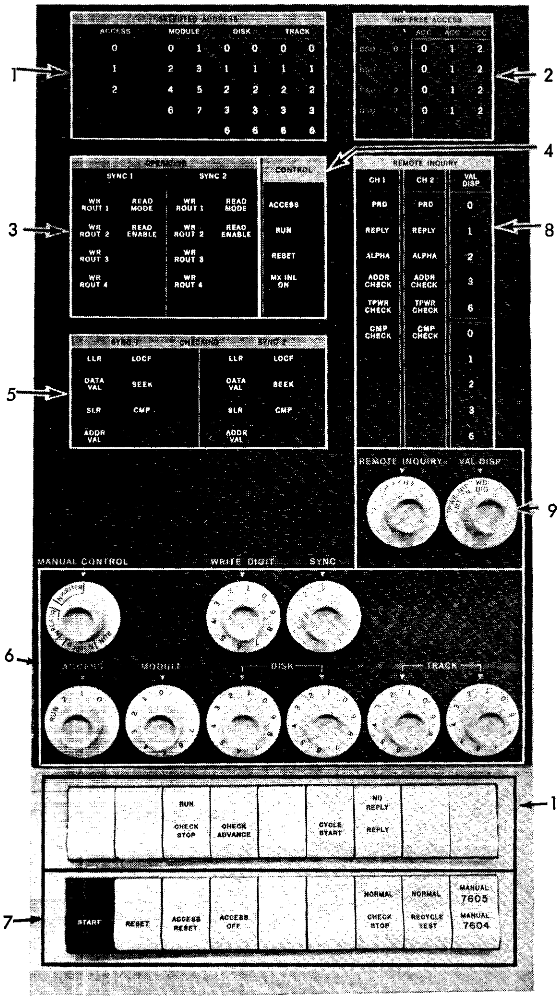


Figure 199. IBM 7300 Disk Storage and Inquiry Panel

The READ MODE light is on whenever the read mode latch is on, and the READ ENABLE light is on whenever the read enable latch is on. Reading occurs either during a read operation or write check operation.

Control Lights (Figure 199-4)

ACCESS indicates that an access arm is moving to a disk-track location. If the light remains on, it indicates that the location has not been found.

RUN indicates that the access dial switch is set to the RUN position. When the manual control dial switch is also set to the RUN position, the RUN light indicates that the disk storage units are awaiting instructions.

RESET indicates that the reset key on the panel or the auxiliary reset key on the 7300 Disk Storage Unit has been pressed.

MX INL ON (Matrix Interlock On) indicates that the matrix interlock latch is on.

Checking Lights (Figure 199-5)

These lights indicate error conditions for each of the two 7604 disk channels (synchronizers). The lights operate in conjunction with the check stop key (Figure 199-7). When the key is set to NORMAL, an error condition causes the corresponding light to flash on and off. When the key is set to CHECK STOP, an error condition stops the 7300 disk storage operation in progress and turns on the corresponding check light. The light remains on until the reset key (Figure 199-7) is pressed. The checking lights are as follows:

LLR (Long Length Record): The program specifies less than 60 words during a read or write operation. A condition 4 will appear in the final status word.

DATA VAL (Data Validity): Failure to have 2-of-5 bits for every digit.

SLR (Short Length Record): The program specifies more than 60 words during a read or write operation. A condition 3 will appear in the final status word.

ADDR VAL (Address Validity): Read or write operation attempted at a disk address different from location of access arm.

LOCF (Locafier): Locafier check failure.

SEEK: Last seek operation not completed correctly. Note that this light turns on at the beginning of the next read or write operation.

CMP (Compare): Compare check failure during a write operation.

Dial-Type Selector Switches (Figure 199-6)

MANUAL CONTROL: This dial is used to select either one of the six possible manual 7300 disk operations or on-line operation under program control. The seven dial positions are:

WRITE R: The selected access arm writes repetitively a 60-word record at disk-track address 0000. The information in the record is selected by the write digit dial (below).

WRITE N: The selected access arm writes one 60-word record at disk-track address 0000. The information in the record is selected by the WRITE DIGIT dial (below).

READ R: The selected access arm reads repetitively the record at a selected disk-track address.

READ N: The selected access arm reads one record at a selected disk-track address.

SEEK R: The selected access arm seeks repetitively in sequence from disk-track address 0000 to the selected address.

SEEK N: The selected access arm seeks a selected disk-track address.

RUN: Disk storage operations are under control of the stored program.

The R (repetitive) operations described above continue until the reset key (Figure 199-7) is pressed.

WRITE DIGIT: This dial is used to select the digit value to be written in all 600 positions of the record during a manual write operation.

SYNC (Synchronizer): This dial is used to select the 7604 disk channel (1 or 2) for manual operations. When disk storage operations are under control of the stored program, the position of the SYNC dial determines which set of operating and checking lights (SYNC 1 or SYNC 2) will be active.

ACCESS: This dial is used to select an access arm (0, 1, or 2) for manual seek, read, and write operations or to select an access arm for resetting or turning off. When 7300 disk storage operations are under control of the stored program or when all access arms are reset or turned off simultaneously, the dial must be set at the RUN position.

MODULE: This dial is used to select the module number (0-7) part of the disk address when the 7300 disk storage units connected to the system are all Model 2's or a combinatoin of Model 2's and Model 1's. If the units are all Model 1's, then positions 0, 2, 4, and 6 of the dial represent units 0, 1, 2, and 3, respectively.

DISK: These two dials are used to select the tens and units digits of the disk-face part (00-99) of the disk storage address.

TRACK: These two dials are used to select the tens and units digits of the track part (00-99) of the disk storage address.

Key-Type Switches (Figure 199-7)

START: Pressing this key starts the manual operation selected by the manual control dial.

RESET: Pressing this key resets all circuits in the 7605 Disk Storage Control unit. In addition, the first depression of this key after a change from direct current off to direct current on causes all access arms to move to address 0000 and the read-write heads to be lifted from the disk. The arms are then ready for manual or program-controlled operation. Pressing the reset key is a part of the normal power-on operating procedure.

ACCESS RESET: Pressing this key causes the arm selected by the access and module dials to move to address 0000 and the read-write heads to be lifted from the disk. If the access dial is set at RUN, all access arms are similarly affected. When 7300 disk operations will not be performed for an extended period of time, it is recommended that this key be used to cause the read-write heads to be lifted from the disk.

ACCESS OFF: Pressing this key causes power to be removed from the arm which is selected by the access and module dials, making the arm unavailable for operation. If the access dial is set at RUN, all access arms are made unavailable. Pressing the access reset key will make the arms available again for operation.

CHECK STOP: The setting of this key determines whether or not detection of an error condition will cause a 7300 disk storage operation to stop immediately or recycle the operation in an attempt to correct the error. If the key is at NORMAL, recycling occurs; if the key is at CHECK STOP, the operation will stop immediately.

RECYCLE TEST: If the key is set at RECYCLE TEST, the 3-bit and 6-bit positions of the locafier track units digit will be transposed before the four locafier digits reach the locafier check circuitry. The transposition will cause an error condition on a 7300 read or write operation if the locafier disable switch is off. Then the access arm will recycle through ten disks and return to the selected address.

MANUAL 7605, MANUAL 7604: If the key is set at MANUAL 7605, all manual operation may be performed from the 7300 disk storage and inquiry panel. If the key is set at MANUAL 7604, manual read and write operations are performed from the 729 magnetic tape panel; however, the disk storage and inquiry panel also must be used to:

1. Specify the access arm to be used.
2. Specify the disk channel (synchronizer) to be used.
3. Specify either READ N OR WRITE N. Read or write operation is under control of the 7604.

Inquiry

Remote Inquiry Lights (Figure 199-8)

These lights indicate the status of an inquiry operation, type of error (if present), location on the synchronizer, and contents of the synchronizer location. The information displayed in the lights depends upon the settings of the REMOTE INQUIRY and VALIDITY DISPLAY dial-type selector switches (Figure 199-9).

CH 1 and CH 2 (Inquiry Control 1 and Inquiry Control 2): These lights indicate for the corresponding inquiry controls the status of an operation and type of error condition (if present). The setting of the REMOTE INQUIRY dial (Figure 199-9) determines which column of lights will be active. The status lights are PRD (proceed), REPLY, and ALPHA. The error lights are ADDR (address) CHECK, TPWR (typewriter) CHECK, and CMP (compare) CHECK.

VAL DISP (Validity Display): These lights indicate in 2-of-5 bit code a synchronizer location, contents of a synchronizer location, or typewriter-format numbers, depending upon the setting of the VALIDITY DISPLAY dial (Figure 199-9). The assignment of lights is shown below.

INFORMATION	UPPER SET OF FIVE LIGHTS	LOWER SET OF FIVE LIGHTS
Synchronizer Location	Word	Digit
Contents of Location	Numeric	Zone
Typewriter-Format No.	Typewriter	Format

Dial-Type Selector Switches (Figure 199-9)

REMOTE INQUIRY: This dial is used to select the inquiry control. The setting of the dial determines which set of indicator lights, CH 1 or CH 2, will be active.

VAL DISP (Validity Display): This dial is used to select what will be displayed in the validity display indicator lights. The dial has three settings: WD-DIG (word-digit), NU-ZN (numeric-zone), and TPWR-FMT (typewriter-format).

Key-Type Switches (Figure 199-10)

RUN CHECK STOP: When this key is in the CHECK STOP position, detection of an error condition causes the inquiry operation to stop; in the RUN position, detection of an error condition causes the typewriter to type a red asterisk but does not stop the operation.

CHECK ADVANCE: Pressing this key when the RUN CHECK STOP key is in the CHECK STOP position causes a momentary bypass of the check-stop circuitry.

CYCLE START: Pressing this momentary contact key sets on the cycle start latch. The key is used to check the circuits that initiate the proceed status. When the station is in proceed status, the key may be used instead of the character keys on the typewriter keyboard to initiate a drum write operation.

REPLY, NO REPLY: When this key is in the REPLY position, the reply operation (inquiry write) occurs repetitively. The key must be in the NO REPLY position for normal operation.

Diagnostic Test (7070-7074)
+09

Machine Description: The diagnostic sense switch is turned on or off, or the 4-digit address (indexable) in positions 6-9 is transmitted to the instruction counter (IC) if the error latch specified by positions 4-5 is set on. If the latch is set off, the contents of the IC remain unchanged.

Instruction Format

S01	+09.	
23	Index word.	
45	Number of error latch, or:	
	98	Turns diagnostic sense switch on.
	99	Turns diagnostic sense switch off.
6789	Branch address if specified error latch is set on.	

Comments: The diagnostic sense switch can also be turned on and off manually. Once the switch has been turned on by either method, it must be turned off by the same method. An exception to this rule is that a programmed diagnostic switch ON condition can be reset off at the console by the computer reset key if the system is in stop status.

Normal tape I-O operations are not possible in diagnostic mode on the 7070, but are possible on the 7074.

Diagnostic Reset (7070-7074)
−09

Machine Description: This instruction resets error latches as specified in positions 4-5. The location of the next instruction is taken from the instruction counter.

Instruction Format

S01	−09.	
23	00	Index word.
45	00	Resets CPU error latches.
	01	Resets error latches *and* all I-O channel controls.
6789	0000	Not used.

Diagnostic Bit Control (7074)
+08

Machine Description: This instruction provides the ability to store a bit configuration defined by positions 7-9 into an accumulator, the arithmetic register (AR), or the auxiliary register (AUX). The defined bits are stored in all positions of the register specified in position 6, except that field definition is permitted for AR operations.

Or, this instruction can test the bit pattern of defined position(s) of the AR, or position 9 of accumulator 2. If any of the defined bits match the bits specified by positions 7-9, the equal indicator is turned on. If none of the bits match, the low indicator is turned on.

Positions 7-9 specify the bit pattern to be stored or tested. Any digit that contains a 2-of-5 code 6 bit (digit 6, 7, 8, or 9) may be placed in position 7 to define a 0 bit.

In position 8, any digit that contains a 3 bit but not a 6 bit (digit 3, 4, or 5) defines a 1 bit; any digit that contains a 6 bit but not a 3 bit (digit 6, 7, or 8) defines a 2 bit. If both a 1 bit and a 2 bit are to be defined by position 8, the digit 9 (containing both a 3 bit and a 6 bit) is used.

In position 9, any digit that contains a 3 bit but not a 6 bit (digit 3, 4, or 5) defines a 3 bit; any digit that contains a 6 bit but not a 3 bit (digit 6, 7, or 8) defines a 6 bit. If both a 3 bit and a 6 bit are to be defined by position 9, the digit 9 (containing both a 3 bit and a 6 bit) is used in position 9.

A 0, 1, 2, 3, 4 or 5 in position 7 causes no bits to be set. A 0, 1, or 2 in the 8 or 9 position causes no bits to be set.

Example: 699 in pos 7-9 defines all bits (0, 1, 2, 3, and 6). Similarly, 003 defines a 3 bit, 060 defines a 2 bit, 600 defines a 0 bit, and 669 defines a 0, 2, 3, and 6 bit.

Instruction Format

S01	+08.		
23	00 (This code cannot be indexed).		
45	Field definition when 4 or 6 is specified in position 6.		
6	Augments Code:		
	1	Set Acc 1	Sets bits in all ten positions as specified in positions 7-9.
	2	Set Acc 2	
	3	Set Acc 3	
	4	Set AR	
	5	Set AUX	
	6	Test AR*	
	7	Test position 9 of Acc 2*	

*If any of the bits defined by positions 7-9 match the tested register (position 9 of Acc 2; any position of AR), the equal indicator is turned on. If they do not match, the low indicator is turned on.

7 6, 7, 8, or 9 defines a 0 bit
0, 1, 2, 3, 4, or 5 defines no bits
8 3, 4, or 5 defines a 1 bit
6, 7, or 8 defines a 2 bit
9 defines a 1 bit and a 2 bit
0, 1, or 2 defines no bits
9 3, 4, or 5 defines a 3 bit
6, 7, or 8 defines a 6 bit
9 defines both a 3 bit and a 6 bit
0, 1, or 2 defines no bits

Comments: Any arithmetic instruction — for example, Zero and Add — can be used to move information from core storage into the AR for subsequent testing. The diagnostic sense switch must be on if invalid information is to be handled.

Diagnostic Store Register (7074) —08

Machine Description: This instruction provides the ability to store the contents of the arithmetic or auxiliary registers into an address specified in positions 6789.

Instruction Format
S01 —08.
23 00 (This code cannot be indexed).
4 0 Store arithmetic register.
1 Store auxiliary register.
5 Not used.
6789 Address of core storage location to receive contents of specified register.

7070 Error Latches

01	Carry No-Carry
02	Zero Non-Zero
03	Adder Driver
04	Adder Output
05	Channel 1 Validity Check
06	Channel 2 Validity Check
07	A-P Match
08	Operation Sign
09	—03 P.R. 4 Interpretation
10	Branch No-Branch
11	Index Skip-Index
12	Hi-Lo-Equal Failure
13	Set Reset
14	One-up Error
15	Core-to-Core Transfer
16	Table Look-up
17	Out of Range Shift
18	Record Definition Word Sign Alpha
19	Significant Digit Scanner
20	Invalid Address
21	Index Operation Sign Alpha
22	Memory Sign Reverse
23	True Complement
24	Division Overflow
25	Arithmetic Zero Insert
26	Field Ring Error
27	Field Program
28	Field Ring Run-out
29	Program Ring Run-out
30	CC RCW (Unit Record RDW)
31	Tape Control
32	Information Bus Validity Check
33	Address Validity A-P
34	Floating Point Error
35	Index Word Error
36	Stacking Latch Error
00	Diag. Any Error

7074 Error Latches

00	Any Error
01	T Bus
02	T-C Bus
03	Sum Bus
04	Sign Validity Check
06	Storage Read-out I Bus
07	Storage Read-in I Bus
08	Inhibit Validity Check
09	Memory Address Bus
20	Out of Range Shift
21	Index Word 00
22	Tape Control
23	Fld Start-Stop
24	Sign
25	RDW
26	Divide Overflow
27	TLU
28	Floating Point
29	Illegal Address
32	IB Validity Check
40	Branch No-Branch
41	Interrupt Sequence
42	Stacking Latch Error
43	Adder Ring
44	Adder Carry
45	Adder Carry-No Carry
46	True-Complement
47	Hi-Lo-Equal
52	M-Q Parity
54	Set-Reset
56	Shift Counter
57	Hang

Index

READER'S COMMENT FORM

IBM 7070-7074 Principles of Operation GA22-7003-6

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. Comments and suggestions become the property of IBM.

	Yes	No
• Does this publication meet your needs?	□	□
• Did you find the material:		
Easy to read and understand?	□	□
Organized for convenient use?	□	□
Complete?	□	□
Well illustrated?	□	□
Written for your technical level?	□	□

- What is your occupation? ____________________
- How do you use this publication?

As an introduction to the subject?	□	As an instructor in a class?	□
For advanced knowledge of the subject?	□	As a student in a class?	□
For information about operating procedures?	□	As a reference manual?	□

Other ____________________

- Please give specific page and line references with your comments when appropriate. If you wish a reply, be sure to include your name and address.

COMMENTS:

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

Your comments, please . . .

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold   Fold

First Class
Permit 40
Armonk
New York

Business Reply Mail

No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department 813
1133 Westchester Avenue
White Plains, New York 10604

Fold   Fold

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)