File No. 1130-01
Form A26-5881-4

IBM Systems Reference Library

IBM 1130 Functional Characteristics

This manual provides basic programming and operating information for the 1130 Computing System. The functional aspects of the System are explained in detail, and the operational characteristics are described in terms of program instructions, input/output operations, and Central Processing Unit console displays and functions. Intended as a reference manual, the material presented assumes some prior knowledge of stored program computers.

Intended primarily as a reference tool, this manual presents information on a level that requires a minimum of prior knowledge of stored-program computers. Some of the terms used in the following pages, however, may be unfamiliar to the inexperienced. To avoid lessening the value of the book as a reference tool, explanations of terms are confined to the context of their use.

Machine operators and programmers should use this manual in conjunction with the 1130 Operating Procedures, Form A26-5717.

Fifth Edition

This is a major revision of the previous edition (A26-5881-3) and makes it obsolete. Technical Newsletter N26-0199 is included in this edition and is therefore obsolete.

This publication was prepared for production using the IBM System/360 Conversational Terminal System (CTS). Page impressions for photo-offset printing were obtained from a typewriter terminal and then reduced 20 percent.

The illustrations in this manual have a code number in the lower corner. This is a publishing control number and is not related to the subject matter.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.

A form is provided at the back of this publication for your comments.

This manual was prepared by the IBM Systems Development Division, Product Publications, Dept. 455, Bldg. 064, San Jose, California 95114. Send comments concerning the contents of this manual to this address

INTRODUCTION 1

CENTRAL PROCESSING UNIT 2
CORE STORAGE 2
Addressing 2
Reserved Core Storage Locations 2
DATA FORMAT 3
INSTRUCTION FORMATS 3
Short Instruction Format 3
Long Instruction Format 3
REGISTERS 4
Index Registers 4
Machine Registers 4
Arithmetic Indicators 5
EFFECTIVE ADDRESS GENERATION .. 6
Short Instruction 6
Long Instruction 6

CPU INSTRUCTIONS 8
LOAD AND STORE INSTRUCTIONS ... 8
Load ACC 8
Load Double 9
Store Accumulator 9
Store Double 9
Load Index 9
Store Index 10
Store Status 10
Load Status 10
ARITHMETIC INSTRUCTIONS 10
Add 10
Add Double 10
Subtract 11
Subtract Double 11
Multiply 11
Divide 11
Logical AND 12
Logical OR 12
Logical Exclusive OR 12
SHIFT INSTRUCTIONS 13
Shift Left ACC 13
Shift Left ACC and EXT 13
Shift Left and Count ACC 13
Shift Left and Count ACC and EXT 14
Shift Right ACC 14
Shift Right ACC and EXT 14
Rotate Right ACC and EXT 15
BRANCH INSTRUCTIONS 15
Branch or Skip on Condition ... 15
Branch and Store IAR 16
Modify Index and Skip 16
Wait 17
EXECUTE I/O 18
Input/Output Control Command . 18
INTERRUPT 19
Interrupt Philosophy 19
Program Operation 20
Interrupt Identification 21
Device Status Word 21
CYCLE-STEAL 21

CONSOLE 23
CONSOLE PRINTER FUNCTIONAL DESCRIPTION 23
Data Coding 23
PRINTER PROGRAMMING 23
I/O Control Commands 23
DSW Indicators 24
KEYBOARD FUNCTIONAL DESCRIPTION 24
KEYBOARD PROGRAMMING 24
I/O Control Commands 24
DSW Indicators 25
CONSOLE ENTRY SWITCHES 25
1131 CPU Usage Meter 26

DISK STORAGE DRIVES 27
Storage Capacity 27
1131 Single Disk Storage 27
Disk Cartridge 27
Access Mechanism 27
Data Organization 27
Timing 28
Data Checking 29
IBM 2310 Disk Storage 29
PROGRAMMING SINGLE DISK STORAGE 30
I/O Control Commands 30
DSW Indicators 31
Programming Considerations ... 32
2310 Usage Meter 32

PUNCHED CARD INPUT/OUTPUT DEVICES. 33
IBM 1442 Card Read Punch 33
Data Coding 33
Card Feeding 34
Card Reading 34
Card Punching 34
Program Load 35
Last Card Sequence 35
PROGRAMMING 35
I/O Control Commands 35
DSW Indicators 36
1442 Usage Meter 37
IBM 2501 Card Reader 37
FUNCTIONAL DESCRIPTION 37
Data Coding 37
Card Feeding 37
Program Load 37
Card Reading 37
Last Card Sequence 38
PROGRAMMING 38
I/O Control Commands 38
DSW Indicators 38
Reader and System Timing 39
2501 Usage Meter 39

PAPER TAPE INPUT/OUTPUT DEVICES .. 40
Tape Specifications 40
Character Code 40
Program Load from 1134 41
PROGRAMMING 41

I/O Control Commands 41
DSW Indicators 41

PRINTERS 43
IBM 1132 Printer 43
FUNCTIONAL DESCRIPTION 43
Forms Control 43
Data Format 43
PROGRAMMING 43
Printer I/O Control Commands .. 43
DSW Indicators 44
1132 Usage Meter 45
IBM 1403 Printer 45
FUNCTIONAL DESCRIPTION 46
Printing 46
Spacing and Skipping 46
Control Tape 46
PROGRAMMING 46
DSW Indicators 47
1403 Usage Meter 48
IBM 1627 PLOTTER 49
FUNCTIONAL DESCRIPTION 49
PROGRAMMING 50
I/O Control Command 50
DSW Indicators 50
IBM 1231 OPTICAL MARK PAGE READER . 51
DATA SHEET 51
Data Sheet Terminology 51
Marking the Data Sheet 51
FUNCTIONAL DESCRIPTION 52
Document Path 52
Message Format 52
Mark Recognition and Discrimination 52
Data Flow 53
Field Checking 54
Alphabetic Coding 55
PROGRAMMING 55
System Programming 56
DSW Indicators 56
1231 Usage Meter 58

IBM 2250 DISPLAY UNIT 59
FUNCTIONAL DESCRIPTION 59
DISPLAYS 60
Graphic Mode 60
Character Mode 60
CHANNEL INTERFACE SECTION 60
PROGRAMMING 61
Input/Output Control Commands 61
DSW Indicators 62
1133 Usage Meter 64

STORAGE ACCESS CHANNEL 65
FUNCTIONAL DESCRIPTION 65
PROGRAMMING 65
I/O Control Commands 66
Special Power Sequencing Considerations 67

SYNCHRONOUS COMMUNICATIONS ADAPTER 68
Binary Synchronous Communications 68
Synchronous Transmit-Receive . 68
LINE ATTACHMENT 68
Half-Duplex Operation 69
FUNCTIONAL DESCRIPTION 69
Synchronous Transmit-Receive Operation 71
Binary Synchronous Communications Operation 74
Data Transmission - Binary Synchronous 76
PROGRAMMING 77
I/O CONTROL COMMANDS 78
Write 78
Read 78
Control 78
Initiate Write 79
Initiate Read 79
Sense Device 79
TIMING FOR SCA PROGRAMMING ... 79

OVERLAPPING INPUT/OUTPUT OPERATIONS AND THROUGHPUT CONSIDERATIONS 80
Cycle-Stealing Concept 80
Direct Program Control 80
Exposure to Loss of Data 80
Device Priority 81
Service Request Limitations .. 81

APPENDIX A. CHARACTER CODES 85
INDEX 87

The IBM 1130 Computing System provides the capability, capacity, and versatility to accomplish the computations that any business, large or small, is confronted with in everyday operations. The 1130 fulfills the engineering and scientific needs with computing power well above that of previous systems in the same cost range. Modularity throughout the system permits a natural and orderly growth path and also permits customized configurations for special applications.

The 1130 system design is oriented to the operator and is particularly suited for operation by the individual requiring the problem solution. A minimum of training and experience with computing systems is sufficient for engineering and research personnel to use the power and speed of the 1130 system to solve problems in individual projects.

IBM provides two powerful programming systems and many application programs to relieve the 1130 user of much detailed programming. The FORTRAN language provided for the 1130 is ideally suited for engineering and scientific personnel; for the user requiring a more intimate knowledge of the system, the powerful assembler language is provided. In addition, a disk-resident monitor system is provided.

The heart of the 1130 system is the 1131 Central Processing Unit. The desk-like CPU has a console printer and keyboard. In two of the three models, a single disk storage drive is provided. The CPU has various core storage capacities and speeds.

The 1131 model 1 has a core storage capacity of 4,096 or 8,192 sixteen-bit words; the core storage cycle time is 3.6 microseconds. The 1131 model 2 has a core storage capacity of 4,096, 8,192, 16,384, or 32,768 words and also has a 3.6 microsecond cycle time. An additional 512,000 words of storage are available on-line with the single disk storage feature which is located within the CPU enclosure in the model 2. The 1131 model 3 has core storage capacities of 8,192, 16,384, or 32,768 words with a cycle time of 2.2 microseconds. The model 3 also has the single disk storage feature.

The CPU console includes data displays and switches for operator convenience and control, a keyboard for data entry, and a console printer.

The many input/output devices and features available with the 1130 system makes the 1130 an ideal growth system. The synchronous communications adapter special feature extends the power of the system by providing the capability to communicate via telephone lines with the IBM System/360 and many other devices. The storage access channel provides the ability to easily tailor non-IBM devices to directly access the 1130 core storage. In addition to these features, input and output may be in the form of punched cards, punched paper tape, special hand-marked documents, pen plotting, line printing, graphic display, and of course disk storage.

A variety of applications is suited for processing by the 1130, and several application programs are provided by IBM. Applications include some aspect of every industry, as well as financial, and governmental operations. In the aerospace, construction, engineering, fabrication and assembly industries, the 1130 can be used for complex mathematical problems, operating analysis and scheduling, estimating, equipment and machine design, simulation, and job cost analysis.

In the processing industries, blending formulas, material balance, material evaluation, forecasting, and unit operations are a few of the applications suitable for the 1130.

Also, in many areas of the transportation, marketing, financial, insurance, utilities, and distribution fields, the 1130 system provides capabilities not previously available in a system of its size.

CENTRAL PROCESSING UNIT

The ability of the IBM 1131 Central Processing Unit (Figure 1 and 2) to ask for and accept input data, perform the calculations required, and produce the output results desired is due to the many functional elements of the machine. Each of these elements is explained in this section, and from these descriptions the CPU emerges as the sum of its parts -- the nerve center of the computing system.

The descriptions that follow concern the storage of data and program instructions, the formats in which data and instructions are stored and used, the functions of CPU registers, the fundamental arithmetic operations and how they are performed, and the aspects of addressing core storage and attached input/output (I/O) units.

CORE STORAGE

The 1131 main storage in the CPU uses magnetic cores for data and program instruction storage. Core storage capacity is 4,096 (4k) to 32,768 (32k) 16-bit words, depending upon the model used:

	A	B	C	D
Model 1	4k	8k	-	-
Model 2	4k	8k	16k	32k
Model 3	-	8k	16k	32k

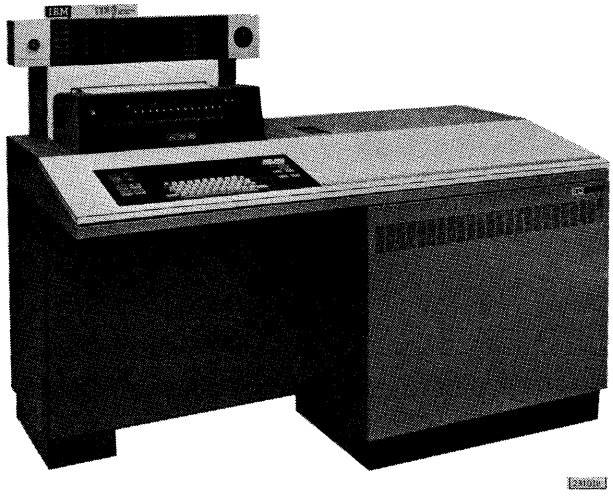


Figure 1. IBM 1131 Central Processing Unit (Model 1A, 1B, 2A, or 2B)

A 16-bit word can be placed into core storage or retrieved from core storage in one storage cycle: 3.6 microseconds for models 1 and 2, and 2.2 microseconds for model 3.

Addressing

Each 16-bit word in core storage is locatable through an address that specifies the position of the word. Addresses range from 00000 to 32767. The high-order address is contiguous with the low-order address, thus providing for "wraparound" addressing. This means that in sequential processing of addresses the highest position of core storage (4095, or 8191, or 16383, or 32767) is followed by 00000 without further specification by the CPU.

The programmer should note that storage is addressed, even if the address word contains an address above actual core storage size.

Reserved Core Storage Locations

The following core storage decimal addresses are reserved for the specific use of the CPU:

Core Storage Address	Description
00001	Index register 1
00002	Index register 2
00003	Index register 3
00008-00013	Interrupt vectors
00032-00039	1132 Printer scan field

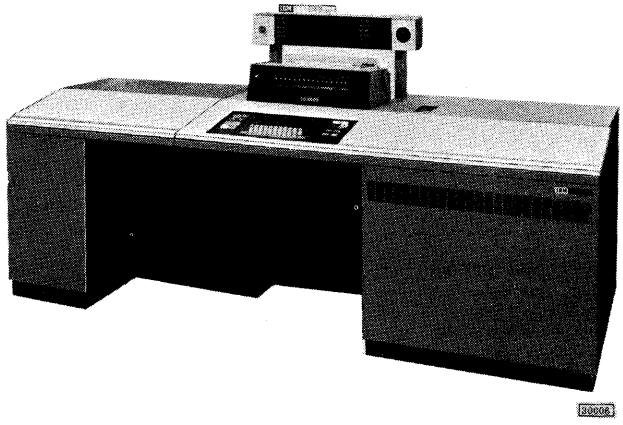


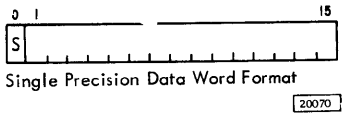
Figure 2. IBM 1131 Central Processing Unit (Model 2C, 2D, 3B, 3C, or 3D)

DATA FORMAT

Data in the 1131 CPU is in fixed-point binary form. Each number is treated as a signed integer: positive numbers are in true binary form with a sign of 0, and negative numbers must be stored and operated upon in 2's complement form with a sign bit of 1. Complementing is done by inverting each bit of the number (including the sign bit) and adding 1 to the low order bit. The following example illustrates this procedure.

Positive number	0001101001001100
Inverted	1110010110110011
Add 1	1
Resulting negative number	1110010110110100

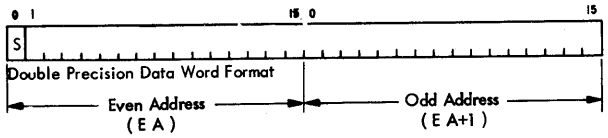
Data is stored as either a single precision word or a double precision word. A single precision data word comprises 16 bits; bit positions are numbered 0 to 15 from left to right. The high-order bit (0) is the sign position.



Single Precision Data Word Format

20070

The largest base-10 (decimal) values of single precision words are +32,767 and -32,768. A double precision data word contains 32 bits, and is composed of two sequential single precision words. The high-order bit (0) is the sign position.



Double Precision Data Word Format

20071A

A double precision data word is addressed by the leftmost word, which must have an even address.

The highest base-10 values of double precision data words are +2,147,483,647, and -2,147,483,648. The largest positive number 2 (to the 31st power)-1, is one less than the largest negative number, -2 (to the 31st power), because the sign (0 for plus, 1 for minus) is, arithmetically, part of the number.

All CPU storage is in binary form, and internal addressing and console displays are in 16-bit binary notation. Because of the ease of operation with 16-bit words in the hexadecimal number system (base 16), all programming systems for the IBM 1130 Computing System use this notation.

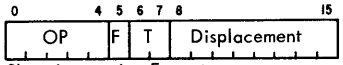
The binary and hexadecimal number systems are described in the IBM publication Number Systems (Form C20-1618).

INSTRUCTION FORMATS

Program instructions in the 1131 occupy either one or two sequential locations; that is, instructions are either short format or long format.

Short Instruction Format

The short instruction consists of one 16-bit word.



Short Instruction Format

20074

OP (Operation) Code: These five bits specify the operation to be performed. Specific operations are described in the CPU Instructions section.

F (Format): The F bit controls the instruction format. It is always 0 for short format and 1 for long format.

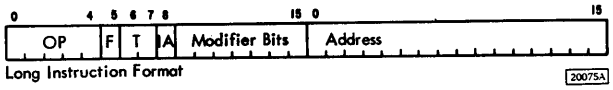
T (Tag): These two bits specify the register to be used in effective address generation; 00 indicates the instruction address register; 01 indicates index register 1 (XR1); 10 indicates XR2; and 11 indicates XR3.

Displacement: The data in these eight bits is added to the data in the instruction address register or index register specified by the tag bits to form the effective address. (The Effective Address Generation section describes this and other aspects or address modification). If the displacement amount is negative, it is in 2's complement. For address modification purposes, the sign in bit position 8 is internally expanded to high-order positions, 0-7, to obtain a 16-bit negative number.

> **Note: Displacement bits have other uses; for example, bits 8 and 9 are used as shift modifiers.**

Long Instruction Format

The long instruction consists of two consecutive 16-bit words. The first eight bits of the first word are the same as the short format.



Long Instruction Format

The last eight bits of the first word and the last word are used as follows:

IA (Indirect Address): A 0 indicates a direct address (contained in the second word). A 1-bit designates an indirect address, which is described in the Effective Address Generation section.

Modifier Bits: Bit positions 9 through 15 have various uses as modifiers. They are described under the applicable instruction.

Address: The second word of a two word instruction contains the address which may be used in its current form or modified by indirect addressing and/or EA modification. Only bits 4-15 are used in a 4k system, bits 3-15 in an 8k system, bits 2-15 in a 16k system, and bits 1-15 in a 32k system.

REGISTERS

The CPU has auxiliary storage areas, called registers, that are used to store data during the performance of operations directed by the stored program. Each register has a distinct purpose and is concerned with a specific type of data. Closely interrelated, they provide the CPU with the necessary functions to provide the results required.

Index Registers

Index registers are located in core storage and are used to contain data added to an instruction address to provide an effective address. In a short instruction, the amount in the displacement field of the instruction is added to the amount in the index register specified by the tag bits (6 and 7). The result becomes the effective address used by the instruction in the operation specified by the operation code. In a long instruction the amount in the address word is added to the amount in the index register specified to obtain the true address or the indirect address.

The index register may contain either a positive or a negative amount.

Register Number	Bits 6 and 7 of Instruction	Core Storage Location
1	01	00001
2	10	00002
3	11	00003

Machine Registers

The registers in the CPU function as necessary to enable the CPU to provide the results specified by the program. The abbreviation for each register name is the designation by which it is usually identified. (See Figure 3.)

ACC (Accumulator): This 16-bit register contains the result of an arithmetic operation. It can be loaded from or stored in core storage, shifted right or left, and otherwise manipulated by specific arithmetic and logical instructions. The ACC is also used with sense interrupt and sense device commands. (The ACC may also be referred to as the A-register.)

EXT (Accumulator Extension): This 16-bit register is the low-order extension of the ACC. It is used during multiply and divide operations, shifting of the ACC and EXT, and double-word arithmetic. (The EXT may also be referred to as the Q-register.)

TAR (Temporary Accumulator): This 16-bit register is the image of the ACC and is used to store the contents of the ACC during effective address computation. (The TAR may also be referred to as the U-register.)

AFR (Arithmetic Factor Register): This 16-bit register holds one operand during arithmetic and logical operations. (The other operand is provided by the ACC.) (The AFR may also be referred to as the D-register.)

SBR (Storage Buffer Register): This 16-bit register is the buffer between the CPU and core storage, and every word of data transferred into or out of core storage passes through the SBR. (The SBR may also be referred to as the B-register.)

SAR (Storage Address Register): This register contains the address pertaining to each reference to a core storage word except I/O transfers to and from cycle-

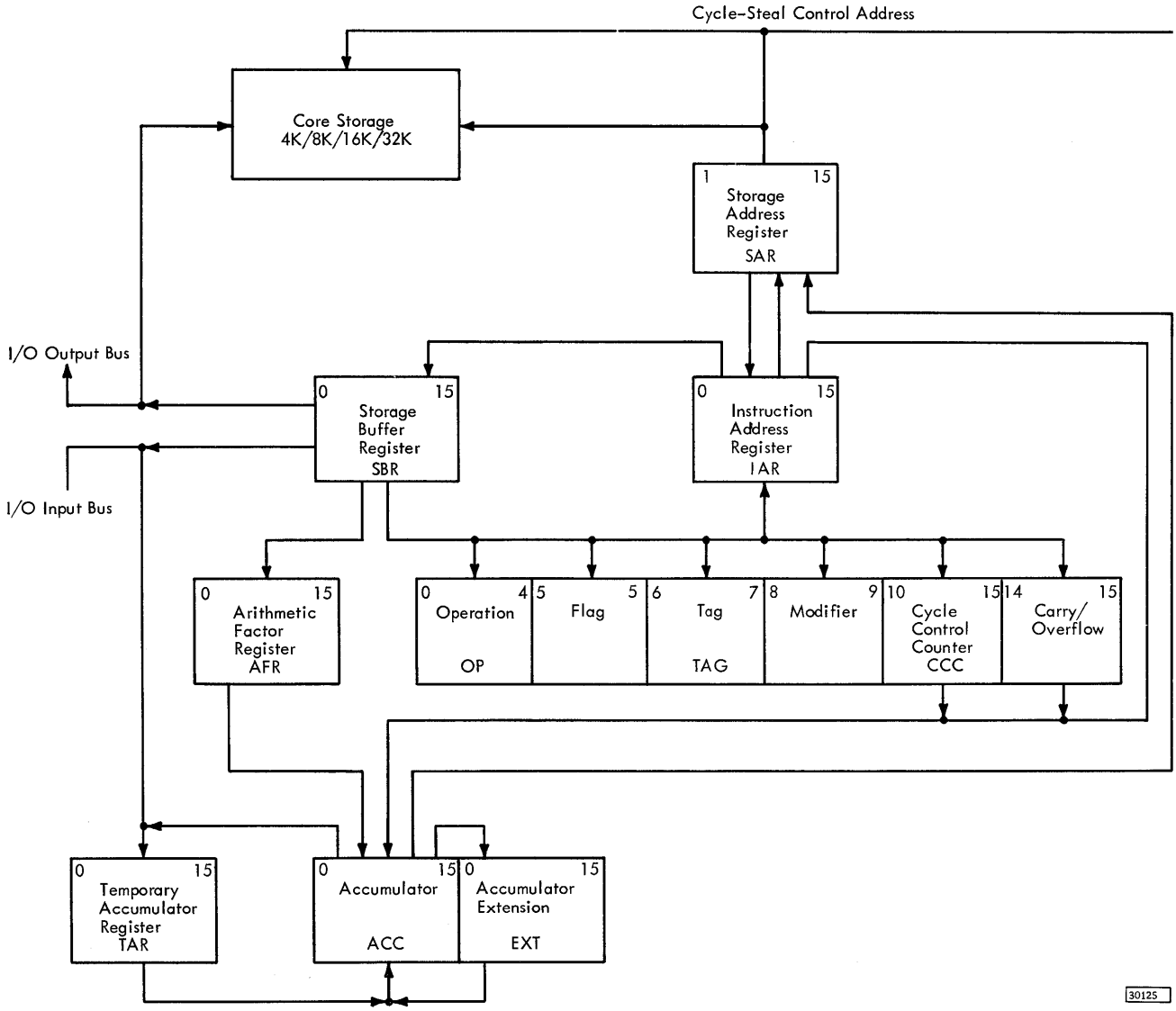


Figure 3. CPU Controls and Data Flow

steal devices, which provide their own storage addressing. The SAR is 13, 14, or 15 bits, depending on core storage size. (The SAR may also be referred to as the M-register.)

IAR (Instruction Address Register): This register is 13, 14, or 15 bits, depending on core storage size. It holds the address of the next sequential instruction. (The IAR may also be referred to as the I-register.)

OP (Operation Register): This five-bit register holds the operation code of the instruction being performed.

TAG (Operation Tag Register): This three-bit register contains the F- and T-bits of the instruction. It controls the instruction length and selects the index register.

CCC (Cycle Control Counter): This six-bit register is used primarily to count CPU cycles and control shift operations.

Arithmetic Indicators

Two indicators are associated with the ACC for arithmetic and shift operation. Each can be turned on irrespective of the other.

Carry Indicator: The carry indicator turns on if, in a shift operation, the last position shifted out of the high-order position of the ACC was a 1-bit. In add operations, the carry indicator

turns on if a carry out of the high-order position of the ACC occurs. In subtract operations, the carry indicator turns on if a borrow beyond the high-order position of the ACC occurs.

The carry indicator is automatically reset prior to execution of each add, subtract, and shift left operations.

The carry indicator may be stored (and reset) by a store status instruction; the status is reflected by a 1-bit for on and a 0-bit for off in position 14 of the word at the effective address. The carry indicator may be set to the status indicated by position 14 of a load status instruction.

Overflow Indicator: This indicator is turned on by an add, subtract, or divide operation when the result exceeds the capacity of the ACC. The overflow indicator may be reset only by program test, a store status instruction, or a load status instruction in which the modifier has a 0 in bit position 15. The status of the overflow indicator may be stored by a store status instruction; the status is reflected by storing a 0 for off and a 1 for on in position 15 of the word at the effective address.

EFFECTIVE ADDRESS GENERATION

As has been noted previously, the location of a 16-bit single precision word or a 32-bit double precision word is denoted by a binary address. The range of addresses, expressed in decimal numbers, is 00000 (0000 hexadecimal) through 32767 (7FFF hexadecimal). Most of the program instructions, which are explained in the CPU Instructions section, instruct the CPU to obtain the data at a specified location and perform a certain operation on it. For example, an add instruction could say, in effect, add the amount stored at location 1904 (0770 hexadecimal) to the amount in the accumulator.

The location 1904 is the effective address of the data referred to by the instruction.

It is part of the versatility of the 1131 CPU that the address in the instruction being executed can be modified as a specific occasion requires. As the result of a particular computation, for example, one of several courses may be indicated. Computation of the effective address of the location of the next instruction or of the next data worked on allows the CPU to proceed according to the predetermined course of action. The factors involved in computing the effective address are described in the following paragraphs.

Short Instruction

The short instruction displacement field contains the amount that is added to either the instruction address register or one of the three index registers to obtain the effective address; the tag bits of the instruction determine which register is to be used. See Figure 4. Note that F=0 is a short instruction:

Tag Bits	F=0 (Direct Addressing)	F=1, IA=0 (Direct Addressing)	F=1, IA=1 (Indirect Addressing)
T=00	EA=Disp+IAR	EA=Add	EA=C/Add
T=01	EA=Disp+XR1	EA=Add+XR1	EA=C/(Add+XR1)
T=10	EA=Disp+XR2	EA=Add+XR2	EA=C/(Add+XR2)
T=11	EA=Disp+XR3	EA=Add+XR3	EA=C/(Add+XR3)

Disp=Contents of Displacement field of instruction.

Add=Contents of Address field of instruction.

C=Contents of Location specified by Add or Add+XR.

Note: This table does not apply to the MDX, LDX, STX, LDS, Shift or Wait instructions.

20111D

Figure 4. Effective Address Determination

IAR: Tag bits of 00 indicate that the sum of the displacement and IAR forms the effective address. (The IAR contains the address of the next sequential instruction.)

Index Registers: Three index registers can also be used to modify the displacement to form the effective address. Tag bits 01, 10, or 11 designate registers 1, 2, or 3. Again, the contents of the specified register, added to the displacement, form the effective address.

Long Instruction

Long instructions are modified in much the same way as short instructions with the added versatility of indirect addressing. The effective address is developed as shown in Figure 4.

Direct Addressing: In the long instruction, a direct address is indicated by a 0 in the IA field. The effective address is governed by the contents of the tag field. Tag bits of 00 indicate that the address field of the instruction contains the effective address, which requires no modification. Tag bits of 01, 10, or 11 specify that the contents of the address field are added to index register 1, 2, or 3, respectively, to form the EA.

Indirect Addressing: A 1-bit in the IA field of the instruction signifies that addressing is indirect; i.e., the address field of the instruction contains the address of the location in core storage that contains the effective address. If no index register is used (T=00), the (indirect) address is specified by the contents of the core storage location specified in the address field. If an index register is used (T=01, 10, or 11), the (indirect) address is specified by the contents of a core storage location which is designated by adding the contents of the address field to the specified index register. As an example (Figure 5), the indirect address is 914 (0392 hexadecimal). The CPU goes to that address and finds the contents of the location to be 2719 (0A9F hexadecimal). The EA, then, is 2719. If index register 1 (T=01) were used, and if the contents of that index register were 3, the CPU would go to address (914 + 3) 917 to find the EA.

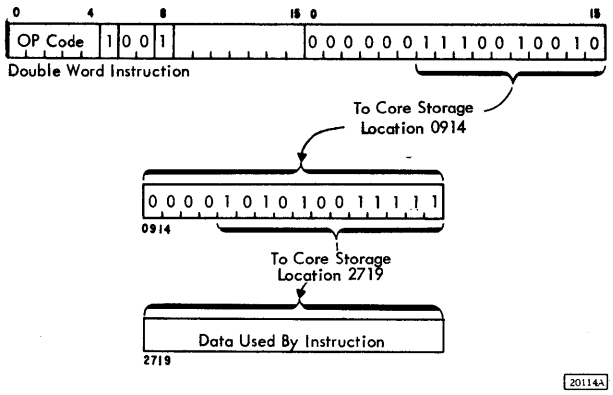


Figure 5. Indirect Addressing

Indirect addressing provides one more level of modification of a given address and provides more programming versatility.

CPU INSTRUCTIONS

The IBM 1130 instruction set (Figure 6) comprises individual instructions divided into five classes. Modifications of these instructions enable additional operations. In the descriptions that follow, the name of the instruction is followed by the mnemonic symbols and the binary representation of the operation code.

LOAD AND STORE INSTRUCTIONS

Load ACC (LD-11000)

The contents of the core storage location specified by the EA replace the contents of the ACC. The contents of core storage are unchanged.

Instruction	Mnemonic	Binary OP Code	Execution Times (in microseconds) for 3.6 µsec Core Storage								Execution Times (in microseconds) for 2.2 µsec Core Storage							
			Single Word (F = 0)				Double Word (F = 1)				Single Word (F = 0)				Double Word (F = 1)			
			T = 00		T = 01, 10, or 11		T = 00		T = 01, 10, or 11		T = 00		T = 01, 10, or 11		T = 00		T = 01, 10, or 11	
			Avg.	Max.	Avg.	Max.	Avg. ①	Max. ①	Avg. ①	Max. ①	Avg.	Max.	Avg.	Max.	Avg. ①	Max. ①	Avg. ①	Max. ①
Load and Store																		
Load ACC	LD	11000	7.6	-	11.2	-	10.8	-	14.8	-	4.6	-	6.8	-	6.6	-	9.0	-
Load Double	LDD	11001	11.2	-	14.9	-	14.4	-	18.0	-	6.8	-	9.1	-	8.8	-	11.0	-
Store ACC	STO	11010	7.6	-	11.2	-	10.8	-	14.8	-	4.6	-	6.8	-	6.6	-	9.0	-
Store Double	STD	11011	11.2	-	14.9	-	14.4	-	18.0	-	6.8	-	9.1	-	8.8	-	11.0	-
Load Index	LDX	01100	4.5	-	7.2	-	7.2	-	11.8	-	2.7	-	4.4	-	4.4	-	7.2	-
Store Index	STX	01101	7.6	-	11.2	-	11.8	-	15.4	-	4.6	-	6.8	-	7.2	-	9.4	-
Load Status*	LDS ⑦	00100	3.6	-	3.6	-	-	-	-	-	2.2	-	2.2	-	-	-	-	-
Store Status	STS	00101	7.6	-	11.2	-	10.8	-	14.8	-	4.6	-	6.8	-	6.6	-	9.0	-
Arithmetic																		
Add	A	10000	8.0	13.0	11.7	16.6	11.2	16.2	15.3	20.3	4.9	7.9	7.1	10.1	6.8	9.9	9.4	12.4
Add Double	AD	10001	12.2	22.0	15.8	25.6	15.3	25.2	19.3	29.5	7.5	13.4	9.6	15.6	9.4	15.4	11.8	18.0
Subtract	S	10010	8.0	13.0	11.7	16.6	11.2	16.2	15.3	20.3	4.9	7.9	7.1	10.1	6.8	9.9	9.4	12.4
Subtract Double	SD	10011	12.2	22.0	15.8	25.6	15.3	25.2	19.3	29.5	7.5	13.4	9.6	15.6	9.4	15.4	20.1	18.0
Multiply	M	10100	25.7	40.0	29.3	43.6	29.3	43.6	32.9	47.2	15.7	24.4	17.9	26.6	17.9	26.6	11.8	28.8
Divide	D	10101	76.0	150.8	79.6	154.4	79.6	154.4	83.2	150.0	46.4	92.1	48.6	94.4	48.6	94.4	50.8	91.6
AND	AND	11100	7.6	-	11.2	-	10.8	-	14.8	-	4.6	-	6.8	-	6.6	-	9.0	-
OR	OR	11101	7.6	-	11.2	-	10.8	-	14.8	-	4.6	-	6.8	-	6.6	-	9.0	-
Exclusive OR	EOR	11110	7.6	-	11.2	-	10.8	-	14.8	-	4.6	-	6.8	-	6.6	-	9.0	-
Shift Left* Modifier Bits 8 & 9:																		
Shift Left ACC 00	SLA ⑦	00010																
Shift Left ACC and EXT 10	SLT ⑦	00010																
Shift Left and Count ACC 01	⑧ SLCA ⑦	00010																
Shift Left and Count ACC and EXT 11	⑧ SLC ⑦	00010	③	-	④	-	-	-	-	-	③	-	④	-	-	-	-	-
Shift Right* Modifier Bits 8 & 9:																		
Shift Right ACC 00 or 01	SRA ⑦	00011																
Shift Right ACC and EXT 10	SRT ⑦	00011																
Rotate Right 11	RTE ⑦	00011	⑤		⑥						⑤		⑥					
Branch																		
Branch and Store IAR	BSI	01000	7.6	-	11.2	-	10.8②	-	14.8	-	4.6	-	6.8	-	6.6②	-	9.0	-
Branch or Skip on Condition	BSC	01001	3.6	-	3.6	-	7.2②	-	11.2	-	2.2	-	2.2	-	4.4②	-	6.8	-
Modify Index and Skip	MDX	01110	4.5	9.9	11.2	16.2	18.5	23.4	18.5	23.4	2.7	6.0	6.8	9.9	11.3	14.3	11.3	14.3
Wait*	WAIT ⑦	00110 ⑨	3.6	-	3.6	-	-	-	-	-	2.2	-	2.2	-	-	-	-	-
Input/Output																		
Execute I/O	XIO ⑩	00001	11.2	-	14.8	-	14.4	-	18.4	-	6.8	-	9.0	-	8.8	-	11.2	-

* Valid in short format only

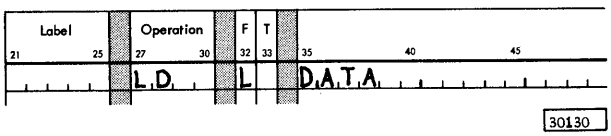
Notes:

1. Indirect addressing, where applicable, adds one storage cycle (2.2 or 3.6 µsec) to execution time
2. If branch is taken
3. One storage cycle + .45(N-4)
4. Two storage cycles + .45 (N-4)
5. N > 16: One storage cycle + .45(N-19)
   N < 16: One storage cycle + .45(N-4)
6. N > 16: Two storage cycles + .45(N-19)
   N < 16: Two storage cycles + .45(N-4)
   where N = number of positions shifted
7. Indirect addressing not allowed
8. If T = 00, functions as SLA or SLT
9. All unassigned OP codes are defined as Wait operations
10. If XIO Read or Write, add one storage cycle

20243D

Figure 6. 1130 Instruction Set and Execution Times

The carry and overflow indicators are not affected.

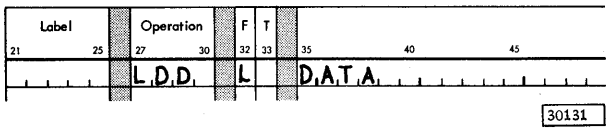


In the preceding example the ACC is loaded with the contents of DATA.

Load Double (LDD-11001)

The contents of the core storage locations specified by the EA and EA + 1 are loaded into the ACC and EXT, respectively. This instruction provides a double-word load for use with double-word arithmetic. The EA must be even for correct operation. If the EA is odd, the contents of that location are loaded into both the ACC and EXT. The contents of core storage are not changed.

Carry and overflow indicators are not affected.

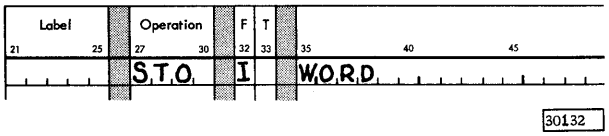


In the preceding example the ACC and EXT are loaded with the contents of DATA and DATA + 1. To achieve the desired results DATA must be an even address.

Store Accumulator (STO-11010)

The contents of the ACC replace the contents of the core storage location specified by the EA. The contents of the ACC are not changed.

The carry and overflow indicators are not affected.

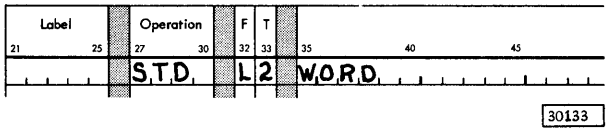


In the preceding example the ACC is stored at the location whose address is stored at WORD.

Store Double (STD-11011)

The contents of the ACC and EXT replace the contents of the core storage locations specified by the EA and EA + 1. This instruction provides a double-word store for use with double-word arithmetic. The EA must be even for correct operation. If the EA is odd, the contents of the ACC are stored in the EA and the contents of the EXT are not stored. The contents of the ACC and EXT are not changed.

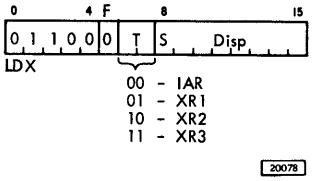
The carry and overflow indicators are not affected.



In the preceding example the ACC and EXT are stored at EA and EA + 1. The EA is determined by adding the contents of WORD to index register 2. The EA must be an even address.

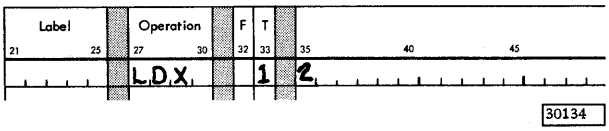
Load Index (LDX-01100)

The contents of the instruction address register or index register specified by the tag bits of the instruction are replaced by the data specified. In the short instruction (F=0), the register is loaded with the displacement. The eight high-order positions of the register are filled with the value of the sign bit (bit position 8 of the instruction) to complete the 16-bit word. Unlike other instructions in the long format (F = 1), an LDX with IA = 0 loads the register with the address rather than the contents of the storage location. (Note that this procedure is different from that of other load instructions.) When the IA = 1, the contents of the storage location specified by the address are loaded into the register.



Loading the IAR results in an unconditional branch to the address loaded.

The carry and overflow indicators are not affected.

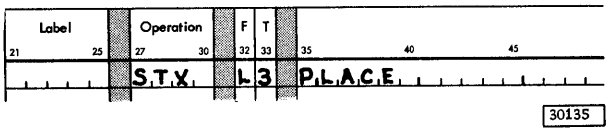


In the preceding example index register 1 is loaded with the value 2.

Store Index (STX-01101)

The contents of the register specified by the tag bits are stored in the core storage location specified by the EA. (See the table under Load Index for tag bit codes.) The contents of the register are not changed.

The carry and overflow indicators are not affected.



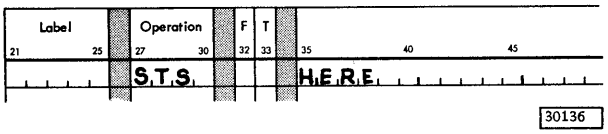
In the preceding example the contents of index register 3 are stored at location PLACE.

Store Status (STS-00101)

The status of the carry and overflow indicators are stored in bits 14 and 15, respectively, of the word at the EA. Bits 0-7 of the storage word remain unchanged; bits 8-13 are reset to zeros. The status of each indicator is reflected by storing a 1-bit if the indicator is on and a 0 if the indicator is off.

The carry and overflow indicators are reset as a result of the operation.

Note: The word in core storage in which the status of the indicators is stored is normally the next load status instruction, the description of which follows.

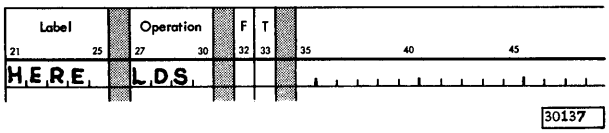


In the preceding example the status of the carry and overflow indicators are stored in position 14 and 15, respectively, of the word located at address HERE. A 1-bit is stored if the indicator is on; a 0-bit is stored if the indicator is off.

Load Status (LDS-00100)

This instruction is always in the short format (F=0). The carry and overflow indicators are set to the conditions indicated by bits 14 and 15, respectively, of the instruction. A 1 sets the indicator to the on condition; a 0 sets it to the off condition.

The carry and overflow indicators are set according to the bits in position 14 and 15.



In the preceding example the status of the carry and overflow indicators are set according to bits 14 and 15 of the instruction. These bits are usually determined by a previous store status.

Note: The load status instruction is the word in core storage in which the status of the indicators is stored by the previous (store status) instruction.

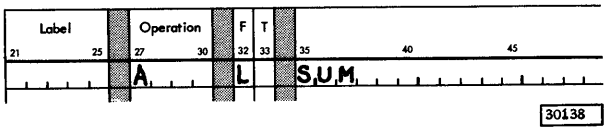
ARITHMETIC INSTRUCTIONS

Add (A-10000)

The contents of the core storage location specified by the instruction are added algebraically to the contents of the ACC. Negative data is in 2's complement form. The sum replaces the contents of the ACC. The contents of the core storage location remain unchanged.

The overflow indicator is turned on if the sum is greater than the capacity of the ACC 2 (to the 15th power)-1 or -2 (to the 15th power). If the indicator is on when the overflow occurs, it is not changed.

The carry indicator is set by a carry out of the high-order bit position of the ACC. The carry indicator is dynamic and is conditioned for each add instruction.



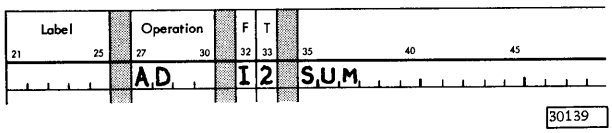
In the preceding example the contents of SUM are added to the ACC.

Add Double (AD-10001)

The contents of the core storage locations at EA and EA + 1 are added algebraically to the contents of the ACC and EXT. Negative data is in 2's complement form. This instruction provides double-word addition in which the ACC and EXT are considered as one 32-bit accumulator. The sum replaces the contents of the ACC and EXT; the contents of the core storage locations are not changed. The EA must be even for correct operation. If the EA is odd, the contents of the location are added to both the ACC and EXT.

The carry and overflow indicators are affected as in the add instruction where

the ACC is considered as one 32-bit register.

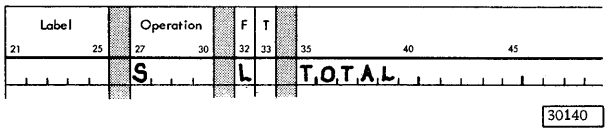


In the preceding example the contents of SUM are added to the value in index register 2 to obtain the effective address. The contents of EA (EA must be an even address) and EA + 1 are treated as a single 32-bit word and are added to the ACC and EXT.

Subtract (S-10010)

The contents of the core storage location specified by the instruction are directly subtracted from the contents of the ACC. The result replaces the contents of the ACC. The contents of the core storage location are not changed.

The carry and overflow indicators are affected as in the add instruction. The carry indicator, if on, reflects a borrow condition.

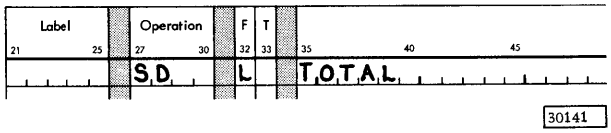


In the preceding example the contents of TOTAL are subtracted from the ACC.

Subtract Double (SD-10011)

The contents of the core storage locations at EA and EA + 1 are subtracted from the contents of the ACC and EXT. This instruction provides double-word subtraction in which the ACC and EXT are considered one 32-bit accumulator. The difference replaces the contents of the ACC and EXT; the contents of the core storage location are not changed. The EA must be even for correct operation. If the EA is odd, the contents of that location are subtracted from both the ACC and EXT.

The carry and overflow indicators are affected in the same way as in the subtract instruction.

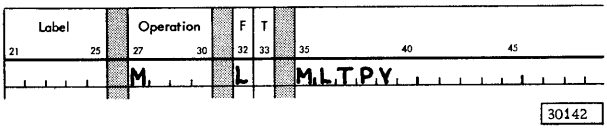


In the preceding example TOTAL and TOTAL +1 are subtracted from the ACC and EXT. TOTAL must be an even address.

Multiply (M-10100)

The contents of the core storage location specified by the instruction (the multiplicand) is multiplied algebraically by the contents of the ACC (multiplier). The 32-bit product replaces the contents of the ACC and EXT. Bit 15 of the EXT is the low-order bit, and bit 0 of the ACC is the high-order bit. Contents of core storage are unchanged.

The carry and overflow indicators are not affected.



In the preceding example the contents of MLTPY are multiplied by the contents of the ACC. The resultant product is 32 bits with its low-order bit in position 15 of the EXT.

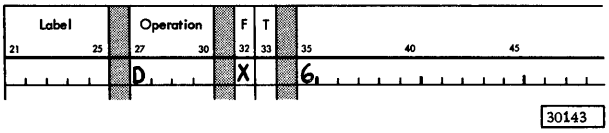
Note: The largest product that can be developed is 2 (to the 30th power), which results if both multiplier and multiplicand are -2 (to the 15th power) The multiply instruction does not provide for double-precision arithmetic.

Divide (D-10101)

The contents of the ACC and EXT are considered as a single 32-bit word. The dividend must be in the ACC and EXT with the units position in bit 15 of the EXT. The dividend is divided by the contents of the core storage location specified

by the instruction. The quotient replaces the contents of the ACC; the remainder, which carries the sign of the dividend, is placed in the EXT.

The overflow indicator is turned on by an attempt to divide by zero or by a quotient overflow, which occurs when the quotient exceeds the range of -2 (to the 15th power) to 2 (to the 15th power) -1. An overflow causes the ACC and EXT to be left in an undefined state.



The contents of the ACC and EXT are considered as a 32-bit dividend; therefore, it is generally necessary to shift the ACC right 16 places before dividing. In the preceding example the ACC and EXT are divided by the contents of the EA which is determined by adding the displacement (6) to the IAR. The quotient replaces the ACC and the remainder is in the EXT.

Note: The x format is required in the 1130 assembler language to indicate that six represents a displacement from from IAR and not an absolute address.

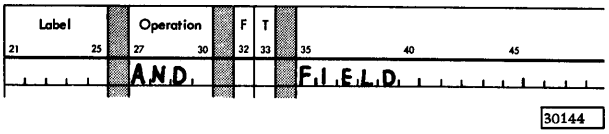
Logical AND (AND-11100)

The contents of the core storage location specified by the instruction are ANDed, bit by bit, with the contents of the ACC; the results replace the contents of the ACC. The contents of core storage remain unchanged.

The AND operation compares each bit position of two words (fields) and places a 1-bit in the result field (ACC) position if both fields contain a 1-bit in that position. The table that follows illustrates the four possible bit combinations in the same bit position of two ANDed words.

Core Storage	ACC	Result
0	0	0
0	1	0
1	0	0
1	1	1

The carry and overflow indicators are not affected.



In the preceding example the contents of FIELD are ANDed with the ACC. If a 1-bit appears in both words in corresponding positions, a 1-bit is placed in that position of the ACC; otherwise, a 0 is placed there.

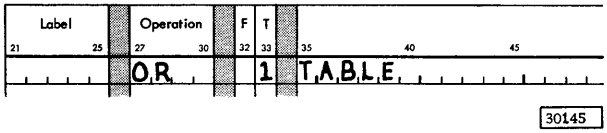
Logical OR (OR-11101)

The contents of the core storage location specified by the instruction are ORed, bit by bit, with the contents of the ACC. The results replace the contents of the ACC; the contents of core storage are unchanged.

The OR operation compares each bit position of two words (fields) and places a 1-bit in that position of the result field (ACC) if either field contains a 1-bit in that position. The table that follows illustrates the four possible bit combinations in the same bit position of the two ORed fields.

Core Storage	ACC	Result
0	0	0
0	1	1
1	0	1
1	1	1

The carry and overflow indicators are not affected



The contents of the effective address (determined by modifying the address of TABLE by index register 1) are ORed into the ACC. A 1-bit appearing in either the ACC or the word at EA causes a 1-bit to be placed in the corresponding position of the ACC.

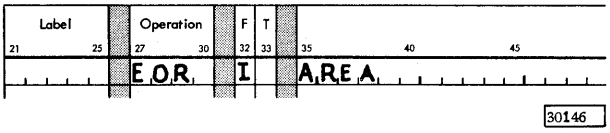
Logical Exclusive OR (EOR-11110)

The contents of the core storage location specified by the instruction are exclusive-ORed, bit by bit, with the contents of the ACC. The result replaces the contents of the ACC; the contents of core storage are unchanged.

The exclusive-OR operation compares each bit position of two words (fields) and places a 1-bit in that position of the result field (ACC) if either field, but not both, contains a 1-bit in that position. The table that follows illustrates the four possible bit combinations in the same bit position of the two exclusive ORed fields.

Core Storage	ACC	Result
0	0	0
0	1	1
1	0	1
1	1	0

The carry and overflow indicators are not affected.



In the preceding example the EA is determined by the contents of AREA. The contents of the EA are exclusively ORed into the ACC. That is, ACC bits which are equal to their corresponding bits at EA are set to 0, otherwise to 1.

SHIFT INSTRUCTIONS

All shift operations are in the short format (F=0) only. Each of the three shift right and four shift left instructions is defined by bits 8 and 9 of the basic shift right and shift left instructions. Except for the shift left and count instructions, the number of positions shifted is controlled by the contents of the six low-order bits of the register specified by the tag bits, as shown by the table below. The number of positions shifted is called the shift count. (XR is the abbreviation for index register.)

Tag Bits	Shift Controlled By Low-Order Six Bits
00	Displacement
01	XR1
10	XR2
11	XR3

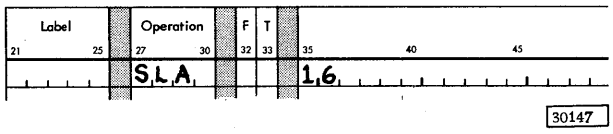
If the shift count is 0 in the control field addressed, the instruction performs as a no-op, and the carry indicator is not affected.

Shift Left ACC (SLA-00010)

Bits 8 and 9 = 00

The ACC is shifted left the number of positions specified by the shift count, and vacated (low-order) bit positions are set to 0. The EXT is not affected.

The condition of the carry indicator is determined by the contents of the last bit position shifted out of the ACC. The Carry indicator is turned on if the last bit shifted out of the high-order position of the ACC was a 1; it is turned off if this bit was 0. The overflow indicator is not affected.



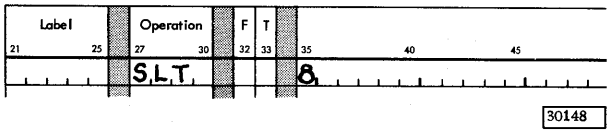
In the preceding example the ACC is shifted left 16 places. The vacated positions are set to 0. In this particular case the ACC is cleared.

Shift Left ACC and EXT (SLT-00010)

Bits 8 and 9 = 10

The ACC and EXT are shifted left (as a 32-bit register) the number of positions specified by the shift count, and vacated bit positions are set to 0.

The carry and overflow indicators are affected as in the shift left ACC instruction.



In the preceding example the ACC and EXT are shifted left eight places as one 32-bit register. Vacated positions are set to 0.

Shift Left and Count ACC (SLCA-00010)

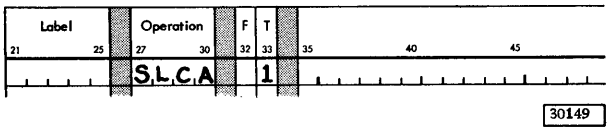
Bits 8 and 9 = 01

Tag bits of 00 cause this instruction to be executed the same as a shift left ACC instruction. Tag bits of 01, 10, or 11 cause the six low-order bits of the designated register to be transferred to the CCC (cycle control counter) as a shift

count. The count is decremented by one for each position the ACC is shifted to the left. The shift is terminated by a 1-bit being shifted to the high-order position of the ACC or the CCC being decremented to zero. The decremented count is then loaded into the six low-order positions of the index register. Bit positions 0-7 of the index register are not affected.

The carry indicator is turned on if the shift is terminated by a 1-bit in the high-order of the ACC. It is turned off if the shift is terminated by the CCC being decremented to zero. If a 1-bit in the high-order position of the ACC coincides with the CCC being decremented to zero, the carry indicator is turned off.

The overflow indicator is not affected.



In the preceding example index register 1 should be set to the maximum number of bit positions that could contain a 1-bit. As the ACC is shifted to the left, index register 1 is decremented by one each time a zero is shifted out. The first 1-bit in ACC position zero terminates the operation. By branching to a table indexed by register 1, a unique subroutine is provided for each position of the ACC that could contain a 1-bit. This selection capability is quite helpful in determining device status and interrupt conditions.

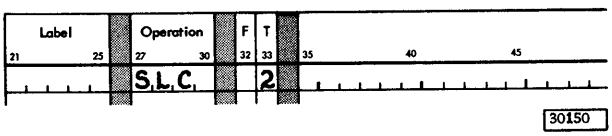
Note: If the count (n) is decremented to zero, a shift left n position has occurred. If the count is initially zero or if the sign bit is initially a 1-bit, the instruction performs as a no-op.

Shift Left and Count ACC and EXT (SLC-00010)

Bits 8 and 9 = 11

This instruction is the same as the shift left and count ACC instruction, except that both the ACC and EXT are shifted. The high-order bits of the EXT are shifted into the low-order positions of the ACC, and the vacated low-order positions of the EXT are set to 0.

The carry and overflow indicators are the same as for the shift left and count ACC instruction.



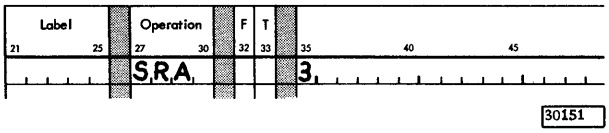
In the preceding example both the ACC and EXT are shifted left until a 1-bit appears in the high-order position of the ACC (or index register 2 is decremented to zero). Index register 2 is decremented by one each time a 0 is shifted out of the ACC.

Shift Right ACC (SRA-00011)

Bits 8 and 9 = 00 or 01

The ACC is shifted right logically (vacated high-order positions set to 0's) the number of positions specified by the shift count. Low-order bits shifted out are lost. The EXT is not affected.

The carry and overflow indicators are not affected.



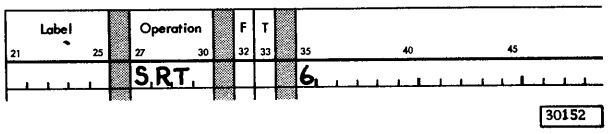
In the preceding example the contents of the ACC are shifted right 3 places. The vacated positions of the ACC are set to 0, and the positions shifted out are lost.

Shift Right ACC and EXT (SRT-00011)

Bits 8 & 9 = 10

The ACC and EXT are shifted right arithmetically (the value of the sign bit is entered into all vacated positions), as a 32-bit word, the number of positions specified by the shift count. Low-order bits of the EXT are shifted out and lost.

The Carry and Overflow indicators are not affected.



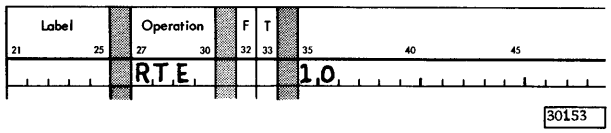
In the preceding example the ACC and EXT are shifted to the right 6 places. The value of the sign bit is placed in all vacated positions of the ACC. The bits shifted out of the EXT are lost.

Rotate Right ACC and EXT (RTE-00011)

Bits 8 and 9 = 11

The ACC and EXT are shifted right (as a 32-bit word) the number of positions specified by the shift count. In effect, a continuous loop is formed, so that the high-order positions of the ACC pick up the bits shifted out of the low-order positions of the EXT. For example, if the shift count is three, all positions of the ACC and EXT shift three positions to the right, and the values of EXT bit positions 13, 14, and 15 are put in ACC bit positions 0, 1, and 2.

The carry and overflow indicators are not affected.



In the preceding example the ACC and EXT are rotated to the right ten places; that is, the bits shifted out of the low-order of the EXT are shifted into the high-order of the ACC.

BRANCH INSTRUCTIONS

Branch instructions provide the means for departing from a sequential series of instructions, by testing to determine if a stated condition or combination of conditions exists, and returning to the point from which the departure was made. Note the unique difference in short and long format instructions in the following descriptions.

Branch or Skip on Condition (BSC or BOSC-01001)

Six separate conditions of the ACC can be tested by placing a 1-bit in the appropriate bit position of the instruction. The bit positions and corresponding conditions tested for are contained in the table that follows.

Bit Position	Condition
15	Overflow indicator off
14	Carry indicator off
13	ACC contents even
12	ACC positive, not zero
11	ACC negative
10	ACC zero

The contents of the ACC are not changed by testing.

Short Instruction Format (F = 0)

If any one of the conditions specified by the instruction is true, the program skips over the next word in core storage and goes to the second word in sequence. This means that a BSC instruction in the short format must always be followed by a short-format instruction. If an instruction in the long format were to follow, a skip would send the program to the second word of the instruction, and a programming error would result.

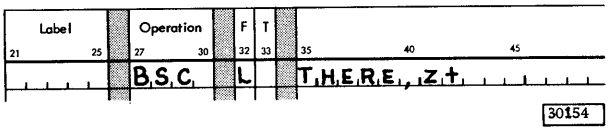
If none of the conditions is true, the next sequential instruction is executed.

If bit positions 10 through 15 contain 0's (no condition tested), the instruction performs as a no-op.

Long Instruction Format (F = 1)

When none of the conditions specified is true, the program branches to the EA. If any one of the conditions is true, the next sequential instruction is executed.

If no condition is specified, the program branches to the EA. This allows the long format of the BSC instruction to be used as an unconditional branch. An explanation of the computation of the EA is contained in the section Effective Address Generation. When this instruction specifies an indirect address (IA = 1), it enables a return to the program routine from which the CPU departed to execute a subroutine. This is accomplished by making the indirect address in this instruction identical to the EA of the branch and store IAR instruction that effected the branch.



In the preceding example, the program branches to THERE if and only if the ACC is negative. Otherwise, the next sequential instruction is executed.

Programming Note: When an interrupt request has been detected by the hardware, the CPU is directed to service the request by interrupting the program. The CPU program is interrupted at the end of the instruction in progress, and the hardware forces a branch and store IAR indirectly to the address stored in the interrupt vector. Core storage location 8 contains the indirect address for interrupt level 0; locations 9, 10, 11, 12, and 13 contain the indirect addresses for levels 1, 2, 3, 4, and 5, respectively.

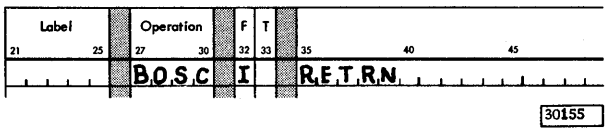
During the servicing of an interrupt, all interrupt requests of equal or lower

priority are effectively constrained from interrupting. However, if a request is detected for a higher priority level than is presently in progress, the program is immediately interrupted again. This is frequently called nesting of interrupts.

At the completion of servicing any level of interrupt, it is necessary to signal the priority hardware to reset the priority status. This reset permits lower priority requests, including those that may have been temporarily constrained, but recorded, to be accepted once again by the CPU. This is effected by making the last instruction of the interrupt service subroutine a BOSC (BSC with bit 9 = 1). This instruction resets the priority level that was just serviced and enables a lower priority interrupt that was pending to be serviced. If no such interrupts were recorded, the CPU program branches to the effective address.

Programmed recognition of waiting interrupts should not be confused with a normal subroutine linkage back to a mainline program, in which case bit 9 may be set to 0.

The BOSC is a conditional instruction; the interrupt is reset only when the branch or skip occurs.



In the preceding example the program branches to the address stored at RETRN and resets the interrupt priority level being serviced.

Indicators

The overflow indicator is reset when tested by the BSC or BOSC instruction; the carry indicator is not reset by testing.

Branch and Store IAR (BSI-01000)

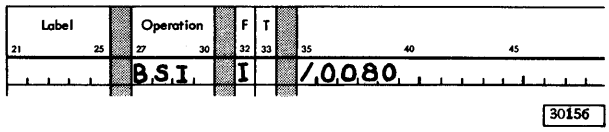
The BSI instruction can be used in either the short or long format.

Short Instruction Format (F = 0)

The contents of the IAR (the location of the next sequential instruction) are stored at the EA of the BSI instruction. The IAR is then set to the EA + 1, which becomes the location of the next instruction executed. For example, assume that the BSI instruction is at storage location 500 and that the EA generated is 600. Execution of the BSI instruction stores 501 at storage location 600 and branches to 601, which is the location of the next instruction.

Long Instruction Format (F = 1)

In the long format, the BSI instruction branches conditionally under the same circumstances as the BSC instruction. The conditions to be tested are designated by bit placement in bits 10-15, as shown by the table in the description of the BSC instruction. If none of the conditions is true, the contents of the IAR are stored at the EA and execution of the instruction proceeds as described for the short format. If one or more of the conditions is true, the next sequential instruction is executed and the IAR is not stored.



In the preceding example the instruction address register is stored at the effective address which is specified by the contents of storage location 80. (The slash denotes hex.) The program then branches to EA + 1.

Indicators

In the short format, the carry and overflow indicators are not affected; in the long format, the overflow indicator is reset when tested. The carry indicator is not reset.

Modify Index and Skip (MDX-01110)

This instruction can be used to modify an index register, the IAR, or the contents of a word in core storage. Except as noted, a skip occurs if the index register or storage word being modified changes sign or is zero after the operation.

A skip causes the program to skip over the next word in storage and go to the second word in sequence. This means that an MDX instruction which could cause a skip should be followed by an instruction in the short format. If a long-format instruction were to follow, a skip would send the program to the second word in

the instruction, and a programming error would result.

Short Instruction Format (F = 0)

The expanded displacement is added to the register specified by the tag bits of the instruction, according to the table that follows. The displacement is expanded to 16 bits by duplicating the sign bit eight positions to the left in the resulting high-order position.

Tag Bits	Operation
00	Displacement added to IAR
01	Displacement added to XR1
10	Displacement added to XR2
11	Displacement added to XR3

When the tag bits of the instruction are 00, the MDX instruction becomes a no-op, a skip, or a branch depending on the value of the displacement. Since the IAR contains the address of the next instruction, a displacement value of zero merely sends the CPU to the next instruction; a positive value of one results in a skip; and any other value results in a branch to the modified address in the IAR. (The displacement can also be negative.)

Long Instruction Format (F = 1)

Modification is accomplished according to the tag and IA fields of the instruction. If the tag is 00, the expanded displacement (bits 8 through 15 of the first word of the instruction) is added to the contents of the storage location specified by the address field of the instruction. The displacement is expanded to 16 bits by duplicating the sign bit eight positions to the left in the resulting high-order position. If the tag bits are not 00, the IA bit becomes the controlling factor, as shown below.

IA Bit = 0: The contents of the address field of the instruction are added to the index register (XR) specified by the tag bits:

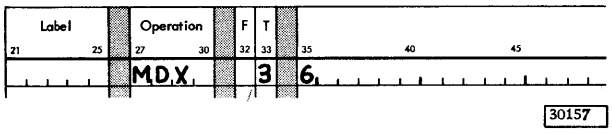
T = 01 XR1
T = 10 XR2
T = 11 XR3

IA Bit = 1: The contents of the core storage location specified by the address are added to the designated index register, according to the tag bit values noted above.

Note: If IA bit = 1 (indirect addressing) and no index register is specified (T = 00), erroneous results occur. The expanded displacement will be 128 less than indicated because of bit 8 being turned on.

Indicators

The carry and overflow indicators are not affected.

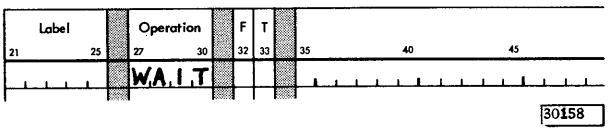


In the preceding example index register three is incremented by six. If index register three changes sign the IAR is incremented by one.

Wait (WAIT-00110 and Undefined Op Codes)

This instruction is in the short format only. The operation of the CPU stops in a wait condition and can be restarted manually or by the detection of an interrupt. A manual restart causes resumption of the program with the next sequential instruction; an interrupt causes resumption at a point determined by the interrupt branch operation. Cycle stealing operations continue in the wait condition.

The carry and overflow indicators are not affected.



In the preceding example the CPU stops. The CPU is automatically restarted if an interrupt request is detected. It may also be restarted by pressing PROGRAM START.

EXECUTE I/O (XIO-00001)

The XIO instuction is the only CPU instruction for servicing external devices. The instruction can be in either the short or long format, and operation is the same, except for the inherent differences in the manner in which the EA is generated, and the fact that the long format can have either a direct or indirect address.

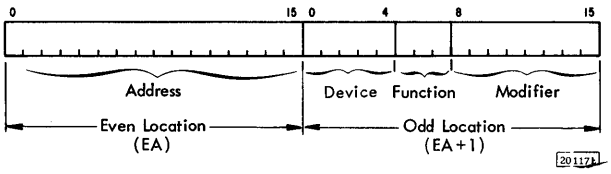
The effective address is the core storage location of the first word of the input/output control command (IOCC); EA + 1 is the location of the second word of the IOCC.

The EA must be even for proper execution.

The contents of the ACC, if significant must be stored prior to execution of the XIO instruction because the ACC is used in the analysis of the IOCC.

Input/Output Control Command

This command allows one instruction, the XIO, to control all input/output devices in the 1130 system. The format of the IOCC follows:



Address

The use of this 16-bit field depends on the function and the device specified.

Device

This five-bit field identifies the I/O device. The device code assignment follows:

Device Code (Binary)	Device
00001	Console Keyboard and Printer
00010	4242 Card Read Punch
00011	1134 Paper Tape Reader and Paper Tape Punch
00100	CPU Single Disk Storage
00101	1627 Plotter
00110	1132 Printer
00111	Console Entry Switches
01000	1231 Optical Mark Page Reader
01001	2501 Card Reader
01010	Synchronous Communications Adapter
10001	2310 Disk Storage Drive 1
10010	2310 Disk Storage Drive 2
10011	2310 Disk Storage Drive 3
10100	2310 Disk Storage Drive 4
10101	1403 Printer
11001	2250 Display Unit

Function

The primary I/O functions are specified by the three-bit function code:

000- Not used

001- Write
This code is used to transfer a single word from core storage to an I/O unit. The address of the storage location is provided by the address field of the IOCC.

010- Read
This code is used to transfer a single word from an I/O unit to core storage. The address of the storage location is provided by the address field of the IOCC.

011- Sense Interrupt
This code is used to load the ACC, with the interrupt level status word (ILSW) for the level being serviced at the time it is issued. This command is common to all I/O devices; therefore, no device code is needed.

100- Control
This code causes the selected device to interpret the modifier and/or address fields as a specific control action.

101- Initiate Write
This code provides the ability to initiate a write operation on a device or unit that will subsequently make data transfers from core store via a data channel.

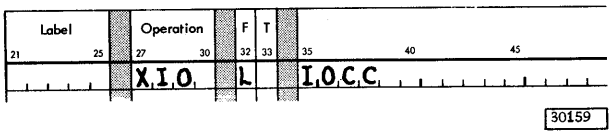
110- Initiate Read
This code provides the ability to initiate a read operation on a device or unit that will subsequently make data transfers to core storage via a data channel.

111- Sense Device

This code loads the ACC with the Device Status Word (DSW) for the device specified by the IOCC. The status indicators that are capable of causing interrupts are reset by specifying modifier bits as follows: bit 15 for the highest level to which the device is connected, bit 14 for the next highest level, and so on.

Modifier

This portion of the IOCC provides additional information for the device and function specified. Where modifier bits affect a command, they are defined with the particular device and command in the following sections.



In the preceding example the I/O control command located at IOCC is executed. IOCC must be an even address.

INTERRUPT

To visualize an interrupt-type system in operation, you may consider an intersection of two streets where the traffic is controlled by lights (stop light). The stop light switches the priority from one street to another in an alternating fashion. An interrupt system in which some priorities are higher than others may be visualized as a railroad crossing a street or highway. The train always has the higher priority and interrupts the flow of automobile traffic each time it approaches the crossing.

To allow for coordination of overlapped I/O operations and provide for a smooth flow of productive processing, it is necessary to provide a means of switching from one program routine to another. In the 1130, an interrupt system is provided for this purpose.

An interrupt causes a program to be suspended and allows some alternate operation to be started. The interrupt facility provides an automatic branch from the normal program sequence to a core storage location identified by an indirect address (interrupt vector) provided in core storage location 8-13, depending on the level of interrupt. A maximum of six interrupt levels are available; the number provided with each system depends on the I/O configuration.

The I/O devices available for attachment to the 1130 are listed below with the interrupt level to which they are assigned and the interrupt vector used to obtain the indirect address of the subroutine to service each interrupt. Interrupt priority depends on the level; the lower the level, the higher the priority.

Level	Interrupt Vector Location	Device
0	8	1442 Card Read Punch (column read, punch)
1	9	1132 Printer, synchronous communications adapter
2	10	Disk storage, storage access channel (SAC)
3	11	1627 Plotter, 2250 Display Unit, SAC
4	12	1442 (operation complete,) keyboard console printer, 1134 Paper Tape Reader, 1055 Paper Tape Punch, 2501 Card Reader, 1403 Printer, 1231 Optical Mark Page Reader, SAC
5	13	Console (program stop switch and interrupt run), SAC

Interrupt Philosophy

Because of the number of types of interrupt requests, it is not always possible to cause a branch to a unique address for each interrupt condition. For the same reason, it is frequently not desirable to cause one branch for all interrupt requests and require the program to determine the individual request(s) requiring service. Therefore, it is expedient to group the many individual request lines into a lesser number of priority levels. This grouping accomplishes two very important functions: First, it allows all interrupt requests common to a specific device to have the privilege of interrupting immediately if

the only requests waiting or being serviced are of a lower priority level. Service is returned to the initial request only after all higher level requests have been serviced. Second, since a unique branch can be defined for each interrupt priority level, it is possible to combine many interrupt requests on a common priority level and therefore use a common interrupt subroutine to service many requests.

There are two important operating characteristics of the 1130 interrupt system: (1) when more than one request line is connected to any priority level, it is necessary, by programming means, to identify the individual request(s) causing the priority level to be energized; (2) the first request that causes an interrupt prevents future requests on the same or lower priority levels from interrupting until the completion of servicing the first interrupt is signaled by a branch out operation. (See Branch or Skip on Condition -- BSC.)

Interrupts that occur on the same level for which an interrupt is being serviced can be detected and acknowledged before the branch out operation is executed.

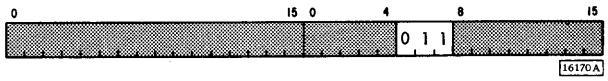
Program Operation

An interrupt may be recognized by the CPU at the completion of any program instruction. It is initiated by the basic interrupt control, which forces execution of a CPU-generated branch and store IAR (BSI) instruction. The indirect address (interrupt transfer vector) of the generated BSI instruction is in one of the words in locations 8-13, corresponding to the level of interrupt. This location should contain the address of the location in the interrupt routine where the IAR is to be stored.

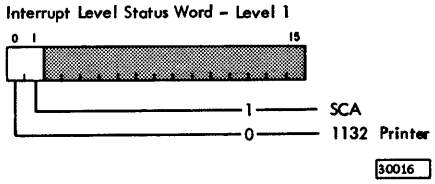
As defined by the BSI instruction description, the IAR is stored at the EA (effective address) and program execution is resumed with the branch to the EA + 1. It is the responsibility of the interrupt subroutine to store all data and/or index registers that are used by the routine, and to restore the same registers prior to departing from the subroutine. (See the description of BSC.)

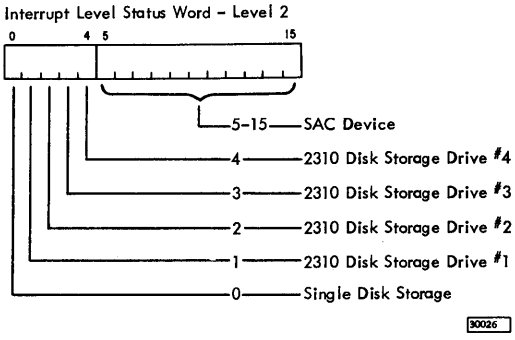
Several devices can request an interrupt on most levels. It thus becomes necessary for the program to determine the requesting device. This is accomplished by issuing an XIO instruction with a function of sense interrupt.

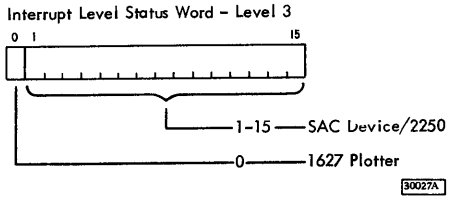
The sense interrupt function is decoded and sent to all I/O devices, along with the current interrupt level being serviced. Each device requesting service on the current level will have a bit appear in the interrupt level status word (ILSW) that is loaded into the accumulator.

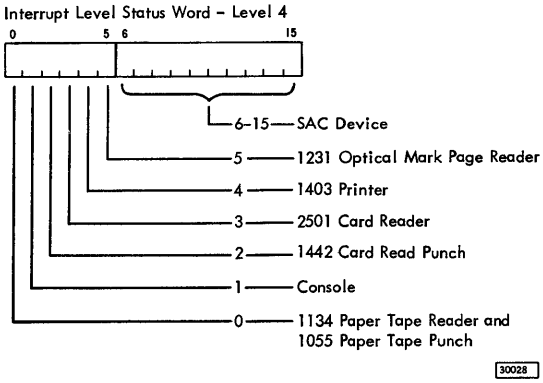


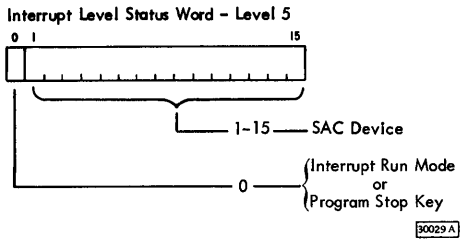
Each device is given a particular bit position in its ILSW to indicate its interrupt request status, a 1-bit if on and 0 if off. The status indicator(s) in the device(s) is not affected by the sensing of the ILSW. It is possible for a device to contain several conditions which may cause an interrupt on the same interrupt level. When this condition exists, the interrupt conditions are logically ORed to become a single interrupt. The identification of the interrupting condition within the device is accomplished by sensing the device status word (DSW) as discussed in subsequent paragraphs.











Interrupt Identification

Following loading of the ILSW in the ACC (accomplished by an XIO sense interrupt instruction), the shift left and count instruction is used to facilitate examination of the ILSW. First, an index register is loaded with a quantity which corresponds to the number of request signals connected to a particular interrupt level, followed by the shift left and count instruction (SLCA). The resulting count in the index register is unique and corresponds to the first nonzero bit of the ILSW in the accumulator. (It is also possible to execute a shift left and count of both the ACC and EXT. Refer to the SLCA Instruction Description.) The SLCA is followed by a branch or skip on condition instruction (BSC) utilizing the F = 1 format with IA = 1, indexed with the result of the SLCA. This combination of instructions provides, in conjunction with a branch table, a unique branch for each nonzero bit of the ILSW.

After the device causing an interrupt has been identified from data in the ILSW, it is necessary to determine the indicator(s) within the particular device causing the interrupt. This determination is accomplished by issuing a subsequent XIO sense device instruction with an area assignment corresponding to that of the device being interrogated. The status indicators are reset after the information has been loaded in the ACC if a bit is present in position 15 of the modifier. If a device can initiate interrupts on more than one interrupt level, the indicators are reset by specifying modifier bits as follows: bit 15 for the highest level to which the device is connected, bit 14 for the next highest level, and so on.

The data in the ACC is now referred to as the DSW (device status word).

> Note: The DSW interrupt indicators must be reset before leaving a service subroutine; otherwise, another interrupt request will be generated for this device.

Device Status Word

The DSW contains one bit of information for each indicator within the device. These usually fall into three categories, (1) error or exception interrupt conditions, (2) normal data or service-required interrupts, and (3) routine status conditions. Figure 7 is a program example using interrupt.

CYCLE-STEAL

The cycle-stealing concept of the 1130 permits the CPU program to start an operation on an I/O device and then continue the mainline program while the I/O device is performing its operation. Each I/O device that operates in this manner takes (steals) a cycle from the CPU when it is needed.

Each cycle-stealing device within the system has its own registers, counters, etc., necessary to address core storage. Therefore, the controls in the CPU are not affected and generally only one CPU cycle is needed to transfer a word. By operating in this fashion, I/O devices place relatively little time demand upon the CPU.

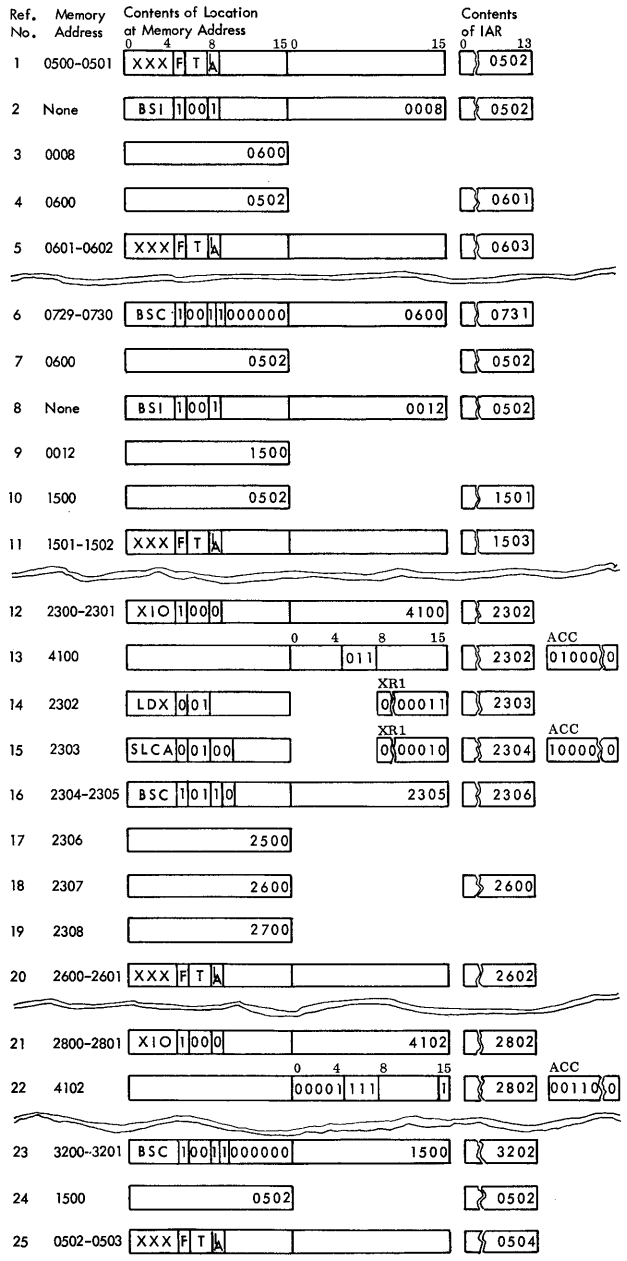
Since the CPU is much faster than any I/O device on the system, several I/O devices may be operating in what appears to be a simultaneous manner. In actuality, the data transfers are alternated; that is, overlapped. See the section entitled "Overlapping Input/Output Operations and Throughput Considerations."

Sample Interrupt Recognition Procedure

The notes that follow are numbered to correspond to the reference numbers in the procedure. Each reference number cited in text is circled, e.g., ①, to avoid confusion with numbers necessary to the procedure, such as memory addresses.

In the registers, instructions, and data words, only the necessary 0-bits and 1-bits are shown. Op codes are shown in alphabetic symbols, and decimal numbers are used to identify memory locations. Binary notation, where used, is obvious. IAR, ACC, and XR1 are shown only where needed for understanding of the operation.

1. Mainline program instruction. During execution of this instruction, the 1442 initiates a card read interrupt.



2. At the conclusion of ① instruction, the CPU blocks the next program instruction and interposes a CPU-generated BSI to start the Level 0 interrupt procedure.

3. IA of Level 0 BSI is 0008; EA at 0008 is 0600.

4. IAR (0502) is stored at EA (0600); IAR is then loaded with EA+1 (0601).

5. First instruction of Level 0 interrupt subroutine. Subroutine must store status of each register, all data, etc., that could be altered by execution of subroutine. Before leaving subroutine, program must restore all registers, data, etc., to condition that existed when interrupt occurred.

6. Last instruction of interrupt subroutine is a BSC instruction with Bit-9 value of 1, which resets the priority status so that interrupts of equal or lower priority can be recognized. If no interrupt is waiting, return to interrupted program is effected by the IA(0600) of the BSC being equal to the EA of the CPU-generated BSI that initiated the interrupt routine, ② . BSC is shown as an unconditional branch (Bits 10-15 = 0); branch could have been conditional, i.e., branch executed only if no conditions specified by Bits 10-15 were true.

   NOTE: The term interrupted program is used to designate either the main program being executed or an interrupt subroutine of lower priority. For example, the ① instruction could be in a routine to service a console printer-keyboard, Level 4 interrupt. Thus, the mainline program can be thought of as a routine with no priority, to which the CPU returns when no interrupts are waiting.

7. EA (0502) is the location of the next mainline program instruction and is loaded into the IAR.

8. To illustrate an interrupt with low priority occurring while a higher priority interrupt is being serviced, we assume the console printer-keyboard initiated a Level 4 interrupt while the card read interrupt was being serviced. We assume no Level 0, 1, 2, or 3 interrupts are waiting and the Level 4, CPU-generated BSI can be interposed, as in ② for Level 0.

9. IA (0012) of the BSI is the memory location assigned to Level 4 interrupts and contains the EA (1500).

10. The IAR is stored at the EA (1500) and then loaded with EA+1 (1501).

11. First instruction of Level 4 subroutine. See ⑤ . Last housekeeping instruction takes subroutine to ⑫ .

12. XIO instruction EA (4100) is the memory location of the IOCC. IAR contents remain at 2302 because the IOCC controls an I/O device and is not a sequential program instruction.

13. IOCC function code of 011 (Sense Interrupt) causes the ILSW for Level 4 to be loaded into the ACC.

14. XR1 is loaded with a quantity equal to the number of response signals connected to the ILSW.

15. SLCA instruction is terminated when the 1-bit associated with the console printer-keyboard interrupt is shifted into the high order position of the ACC. XR1 is reduced by one.

16. BSC address is modified by XR1 (+2) to form the IA (2307). A bit in the 0-position of the ILSW (paper tape reader and paper tape punch) results in an XR1 of 3 and an IA of 2308. A bit in the 2-bit position (card read-punch) results in an XR1 of 1 and an IA of 2306.

17. An IA of 2306 has the EA 2500, which is the memory location of the first insturction of the card read-punch interrupt subroutine.

18. An IA of 2307 has the EA 2600, which is the memory location of the first instruction of the console printer-keyboard interrupt subroutine.

19. An IA of 2308 has the EA 2700, which is the memory location of the first instruction of the paper tape reader and paper tape punch interrupt subroutine.

20. First instruction of housekeeping sequence for console printer-keyboard subroutine.

21. XIO instruction EA (4102) is the memory location of the console printer-keyboard IOCC.

22. The IOCC Sense Device function code (111) causes the DSW of the console printer-keyboard (00001) to be loaded into the ACC. The 1-bit in position 15 causes the response to be reset. The example shows 1-bits in positions 2 and 3 of the ACC (DSW), which indicate that the operator initiated an interrupt request on the keyboard and that the console entry switches are to be read. A programmed subroutine determines the cause of the interrupt (Bit 2) and the interrupting device (Bit 3). A routine then follows that reads the data into memory, accomplishes any housekeeping required, and releases the CPU, as shown in ㉓ .

23. Procedure is the same as in ⑥ and ⑦ . IA (1500) of the BSC instruction is equal to EA of the CPU-generated BSI instruction that initiated the interrupt; see ⑧ , ⑨ , and ⑩ .

24. EA (0502), located at 1500, is loaded into IAR.

25. The instruction at 0502 is the next one to be executed in the mainline program. See ① for previous mainline instruction.

20250

Figure 7. Sample Interrupt Recognition Program

The console (Figure 8) is an integral part of the IBM 1131 Central Processing Unit and consists of the input keyboard, console printer, display panel, function switches and lights and console entry switches.

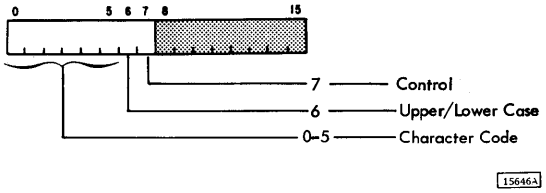
While the keyboard and console printer are usually considered as one unit, control of each of them by the operator and by the stored program is discrete. For this reason, the functional description and programmed operation of each unit is considered separately in the sections that follow.

CONSOLE PRINTER FUNCTIONAL DESCRIPTION

The console printer provides output at a maximum rate of 15.5 characters per second. Data to be printed is transferred from core storage to the console printer by direct program control.

Data characters and control characters (space, tabulate, etc.) are sent to the console printer by means of the write command. Because control characters and data characters are sent in the same manner, the message to be printed contains a mixture of data characters and control characters in the sequence necessary to give the desired formatted output.

The character format within a core storage word to be transmitted to the console printer is:



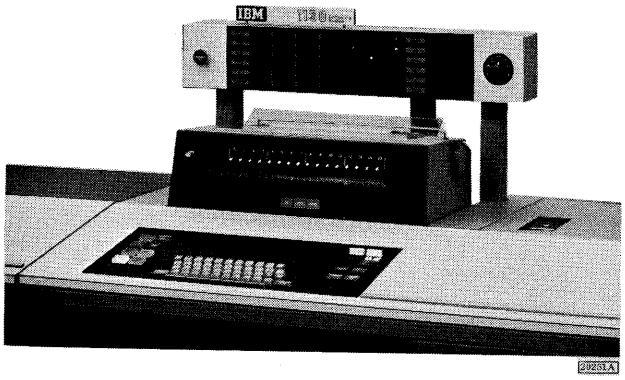


Figure 8. 1131 Console

Each word transmitted to the console printer contains one data character or one control character.

Data Coding

Data to be printed is coded by the program into the console printer code. Appendix A shows the character code.

The data-character code also contains (in bit 6) the information as to whether the character is an upper case (UC) shift or lower case (LC) shift character. The printer shifts automatically as required for each data character.

A printer write command is modified by bit 7 of the output character word. If bit 7 equals one, the write command to the printer is interpreted as a control function. If bit 7 equals zero, the write command is interpreted as a print function.

The codes for console printer control functions are shown in Appendix A.

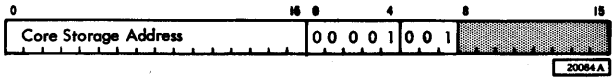
PRINTER PROGRAMMING

The console printer operates in the 1130 system under direct program control.

I/O Control Commands (IOCC)

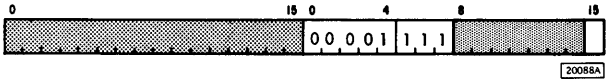
The console printer is addressed by a five-bit device code in the IOCC, 00001.

Write (001)



This command causes bits 0-7 of the word at the core storage location specified by the address to be sent to the printer for printing or control.

Sense Device (111)



This command causes the keyboard/console printer device status word to be placed in the ACC. Figure 9 shows the DSW. Modifier bit 15 on specifies that all keyboard/console printer responses are to be reset.

DSW Indicators

The following indicators are entered into the CPU by a Sense Device command.

Service Response (Interrupt): This indicates an interrupt which occurs each time the console printer has completed printing the data and/or the control operation required by the last word transmitted by the write command.

Not Ready: When off, this indicates that the printer is properly loaded with forms, has dc power, and is not busy. It is necessary that the program always determine that the not ready indicator is off before a write command is given. If a write command is given while not ready is on, loss of information will probably occur. No indication is given of this loss.

Busy: When on, this indicates that the console printer is in the process of typing a character or executing a control and therefore should not be given a write command. The busy line is active from the time data is sent to the printer until the printer has completed the action required.

KEYBOARD FUNCTIONAL DESCRIPTION

The input speed of the keyboard is limited only by the speed of the operator. Keyboard entries are not automatically printed unless the CPU is programmed to provide an output of the entry on the printer. The keyboard emits a coded character for each key struck by the operator. These characters are related to IBM card coding. (See Appendix A Character Code Chart.) Striking an A character key places bits in positions 0 and 3 of the CPU word; striking the I character key places bits in positions 0 and 11 of the word; striking a 9 character places a bit in position 11 of the word; etc.

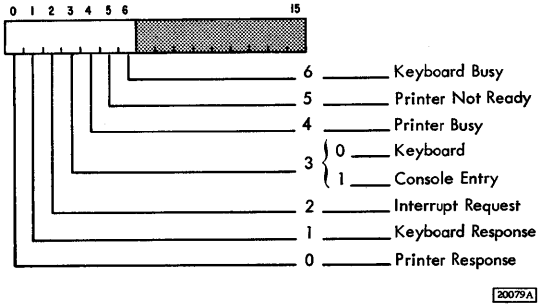
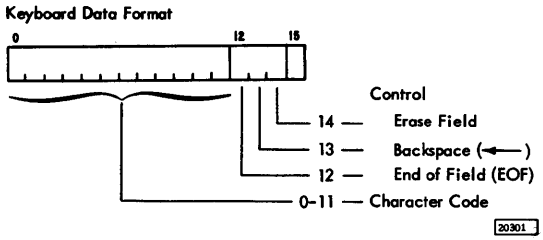


Figure 9. Keyboard/Console Printer Device Status Word



The two-position console/keyboard switch indicates to the program the desired source of the console input data, either the keyboard or the console entry switches.

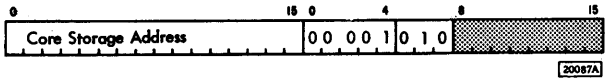
KEYBOARD PROGRAMMING

The keyboard operates under direct program control of the 1130 Computing System.

I/O Control Commands (IOCC)

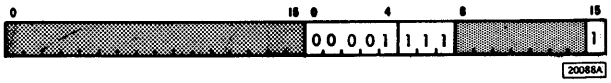
The keyboard is addressed by the same device code used by the console printer, 00001.

Read (010)



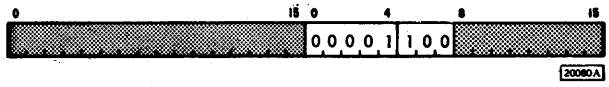
This command enters a single input character from the keyboard into the core storage location specified by the address of the IOCC. A control command must have been previously executed to place the keyboard in a select status.

Sense Device (111)



This command reads the keyboard/console printer device status word into the ACC. Modifier bit 15 on specifies that all keyboard/console printer responses are to be reset.

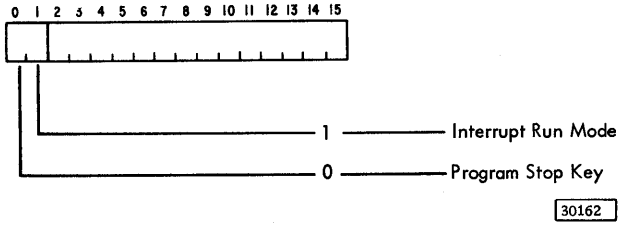
Control (100)



This command places the keyboard in a select status so that a character can be

entered. The read command resets the keyboard select status.

Special Keyboard Console Programming: Either the program stop key or the interrupt run mode causes an interrupt on level 5. The 0-bit in the ILSW is set on. After determining the interrupt is on level 5, the program must issue a sense device with area code 7. The following DSW is presented to determine the cause of the interrupt.



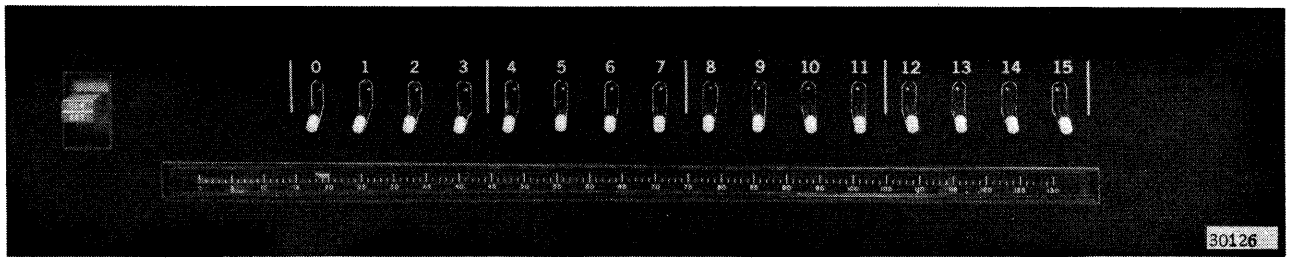
DSW Indicators

Interrupt Request (Interrupt): This indicates an interrupt which is initiated by the request key located on the keyboard.

Keyboard Response (Interrupt): This indicates an interrupt which signals that a character key has been pressed and that a character is ready to be entered into core storage.

CONSOLE ENTRY SWITCHES

These 16 toggle switches are used to set up data or instructions to be entered into core storage. Each switch represents a bit position in a 16-bit word. The procedures that follow provide for entering the information from the console entry switches (CES) by means of manual control, keyboard interrupt, or XIO instruction.



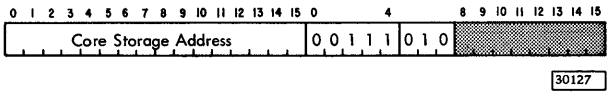
Manual Entry: This procedure causes the bits set in the console entry switches to be loaded into the word at the core storage address in the instruction address register (IAR).

1. Set the mode switch to load
2. Set the CES to the binary core storage address where the data is to be stored.
3. Press the load IAR switch. (Note: The parity indicators P1 and P2 may not show correct parity.)
4. Set the data word in the CES.
5. Press the start key.

Keyboard Interrupt: This procedure requires an interrupt subroutine to service a level 4 interrupt.

1. Set the console/keyboard switch to CONSOLE.
2. Press the keyboard interrupt request key.
3. A level 4 interrupt occurs. The subroutine must analyze the ILSW to determine the interrupting device (keyboard). The keyboard/console printer DSW is loaded into the ACC by a sense device command.
4. The DSW must be analyzed by the subroutine to determine the cause of the interrupt. DSW bit 2=1 indicates an operator interrupt request was initiated by the interrupt request key on the keyboard. DSW bit 3=1 indicates to the interrupt subroutine that the CES should be read by a read command.
5. Return to the mainline program is by the regular method of a BOSC instruction.

Read (010)



This command reads the settings of the CES. Each CES that is on causes a 1-bit to be placed in the corresponding location of the core storage word located at the address specified by the address field of the IOCC. The area code is 7 (00111).

1131 CPU Usage Meter

This meter runs when either of the following conditions is present:

1. When the 1131 clock is running or
2. When any of the attached I/O equipment is operating to finish an instruction given by the 1131 program.

For the second condition the meter will run regardless of the state of the clock, and will stop when the I/O equipment has finished the particular operation the instruction specified.

A customer engineering meter and key lock switch are also provided in the 1131. When this key switch is turned on it activates the C E meter and deactivates the 1131 usage meter and all usage meters on the attached I/O equipment This C E meter then records time in the same manner as specified for the usage meter. The purpose of the C E meter is to record system time during maintenance.

The 1131 usage meter provides the master time measurement of useful work done by the 1130 computing system. All I/O unit meters are interlocked by this meter running and cannot record time if it is stopped. No I/O meter will ever record more time than the 1131 usage meter.

Disk storage provides the 1130 system with low-cost random or sequential access data storage. Disk storage for the 1130 is divided into two separate entities, each dependent upon the other. The disk storage drive provides the access mechanism, the magnetic read/write heads, and the mechanical and electronic components necessary to record and retrieve data. The disk cartridge provides a storage medium. Both the disk storage drive and the disk cartridge will be described in the sections which follow.

Storage Capacity

The storage capacity provided to the 1130 system by the disk storage is 512,000 words per storage drive. A single drive, located within the CPU enclosure, is provided with the model 2 and model 3. The IBM 2310 Disk Storage Drive Model B is also available for attachment to a system that has an 1131 model 2 or model 3 and an IBM 1133 Multiplex Control Enclosure. Two 2310's may be attached to the 1133; each 2310 provides one (2310 model B1) or two (2310 model B2) drives each. Therefore, a total on-line data capacity up to 2,560,000 words is provided. Off-line capacity is virtually unlimited because the interchangeable disk cartridge is easily removed and replaced with another.

Procedure for changing cartridges and other related operating information is provided in 1130 Operating Procedures, Form A26-5717.

1131 SINGLE DISK STORAGE

One single disk storage drive is contained in the CPU cabinet (1131 model 2 or model 3) and is connected to the CPU by a high-speed data channel. It is composed of two components: the disk cartridge, and the drive assembly and access mechanism.

Disk Cartridge

The disk cartridge (Figure 10) is a single disk completely enclosed in a protective housing. The recording medium is an oxide-coated disk that provides two surfaces for the magnetic recording of data. When mounted on a storage drive, the disk rotates at 1,500 revolutions per minute.

Access Mechanism

The disk storage access mechanism has two horizontal arms. Each arm has a magnetic read/write head, and each head is positioned to read or write on the corresponding disk surface as the access arms straddle the disk in the manner of a large tuning fork. The entire assembly moves horizontally forward and backward, so that the heads have access to the entire recording area.

The access mechanism is positioned automatically at the home position (outside cylinder) when the disk cartridge is inserted.

Data Organization

The disk access mechanism is moved back and forth by programmed commands and can be placed in any one of 203 positions, from a point near the periphery of the disk to a point near the center of the disk. At each position, the heads can read or write in a circular pattern on both surfaces of the disk, as it revolves. The circular patterns of data are called tracks. The track on the upper surface of the disk and the corresponding track on the lower surface, both of which can be read or written while the access mechanism is in the same position, are called a cylinder. Figure 11 shows the innermost and outermost cylinders of two tracks each. To complete the picture,

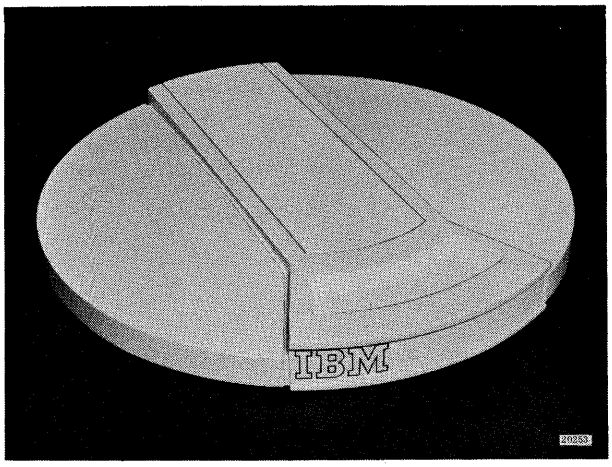


Figure 10. 2315 Disk Cartridge Assembly

the 201 intermediate cylinders, or pairs of tracks, should be visualized; they were omitted for the sake of clarity of the diagram.

For convenience in transferring data between the CPU core storage and disk storage, each track is divided into four equal segments called sectors. Sectors are numbered by the cylinder, from 0 through 7, as shown in Figure 12. Sectors 0-3 divide the upper surface track, and sectors 4-7, the lower. A sector contains 321 data words and is the largest segment of data that can be read or written with a single instruction.

In the programs and programming systems provided by IBM, e.g., the monitor system and its programs, the first word of a 321-word sector is used for the cylinder sector number. Therefore, the first word of the sector cannot be used by the programmer if the assembler program or other components of the monitor system are to be used.

A disk storage word comprises 16 data bits and four check and space bits.

Figure 13 shows the organizational components of disk storage. Note that capacities are based on the 320-word sector; also, the number of cylinders is 200 rather than 203. Three cylinders (24 sectors) are provided as alternates to be used if a surface is defective.

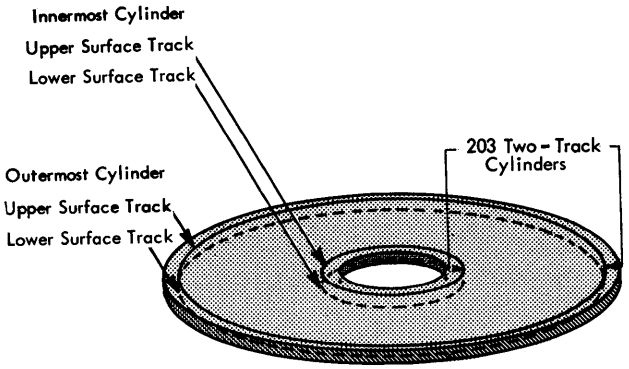
Timing

Timing considerations of disk storage operation involve three elements: access time, reading and writing data, and the time during which the CPU is tied up. (See the section entitled Overlapping Input/Output Operations and Throughput Considerations.)

Access: The access mechanism moves in increments of two cylinders at the rate of 15 ms per increment. Thus, in the formula that follows, the number of cylinders (N) must be even. (The next higher even number is used if an odd number of cylinders is specified.) During the stabilization period (22.5 ms) that follows the last incremental movement, a read or write command can be given and will be started at the end of the stabilization period.

Access time (ms) = 7.5N+T
Where T = 22.5(±2.5)ms

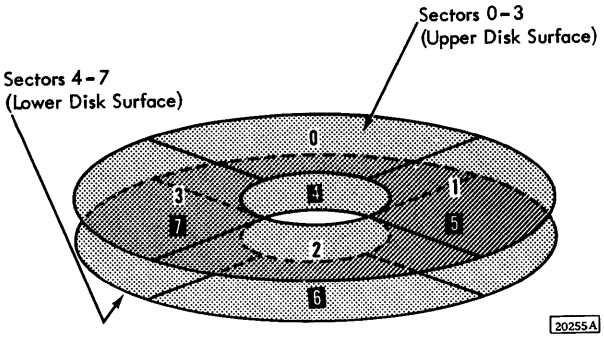
Read/Write: Reading or writing of data in disk storage is at the rate of 27.8 us. per word. Average rotational delay is 20 ms, based on 1,500 rpm, or 40 ms



NOTE: The thickness of the disk has been greatly exaggerated in order to show the relative positions of the upper and lower surface tracks.

20254A

Figure 11. Disk Storage Cylinder Schematic



20255A

Figure 12. Disk Storage Sector Numbers

No. of / Per	Word	Sector	Track	Cylinder	Disk
Bits	16	5,120	20,480	40,960	8,192,000
Data Words		320	1,280	2,560	512,000
Sectors			4	8	1,600
Tracks				2	400
Cylinders					200

30128

Figure 13. Disk Storage Data Organization

per revolution. Thus, a sector can be read or written in an average of 30 ms. Although there are no timing considerations for head switching, there are programming considerations in consecutive sector operations because there is an interval of over 235 us. between sectors; the interval is increased by 27.8 us. for each word less than 321 read or written.

A full cylinder of eight 321-word sectors can be read or written in 100 ms because the rotational delay is required for only the first sector.

CPU Time: An interrupt in a disk storage operation occurs only at the end of the seek, read, or write operation. This means that once the instruction is initiated, disk storage operation is virtually independent of the CPU. As data is being read or written, a cycle is literally "stolen" from the CPU operation in progress every 27.8 us. for the transmission of the next word.

Data Checking

Data is checked on each data transfer between core storage and disk storage. When writing on disk storage, the number of 1-bits in each word is effectively divided by four, as the word is shifted out of the file data register in the disk storage attachment, by incrementing a two-position counter. The number of bits necessary to make the division even (modulo 4) is added to the end of the word. See Figure 14.

The modulo 4 check is performed as each word is read from disk storage. A word that is not modulo 4 causes the data error bit to be set in the disk storage DSW.

The data checking provided in write operations only ensures that data was transferred correctly to the disk drive; however, several factors -- chipped or dirty disk, etc. -- could keep data from actually being recorded correctly on the disk surface. For this reason, the programmer is encouraged to always perform a read-check operation immediately after writing and while source data is still available.

Number Of Data Bits Written 0-15	0	1	2	3
	4	5	6	7
	8	9	10	11
	12	13	14	15
Modulo 4 Counter	00	01	10	11
Check Bits				
16	0	1	1	1
17	0	1	1	0
18	0	1	0	0
19	0	0	0	0

22218

Figure 14. Check Bit Chart

IBM 2310 DISK STORAGE

The IBM 2310 Disk Storage, Model B, (Figure 15) provides additional random-access storage capabilities for the 1130. The 2310 Model B1 contains one single disk storage drive, whereas the 2310 Model B2 contains two single disk storage drives. A maximum of two 2310's may be attached to the 1130 via a channel multiplexer in the 1133. If two 2310's are attached, at least one must be a model B2.

The functional description -- that is, capacity, data organization, data checking, access mechanism, timing, etc. -- is the same for all single disk storage drives, whether located within the CPU or the 2310. Therefore, the descriptions provided under the 1131 Single Disk Storage Drive are not repeated here.

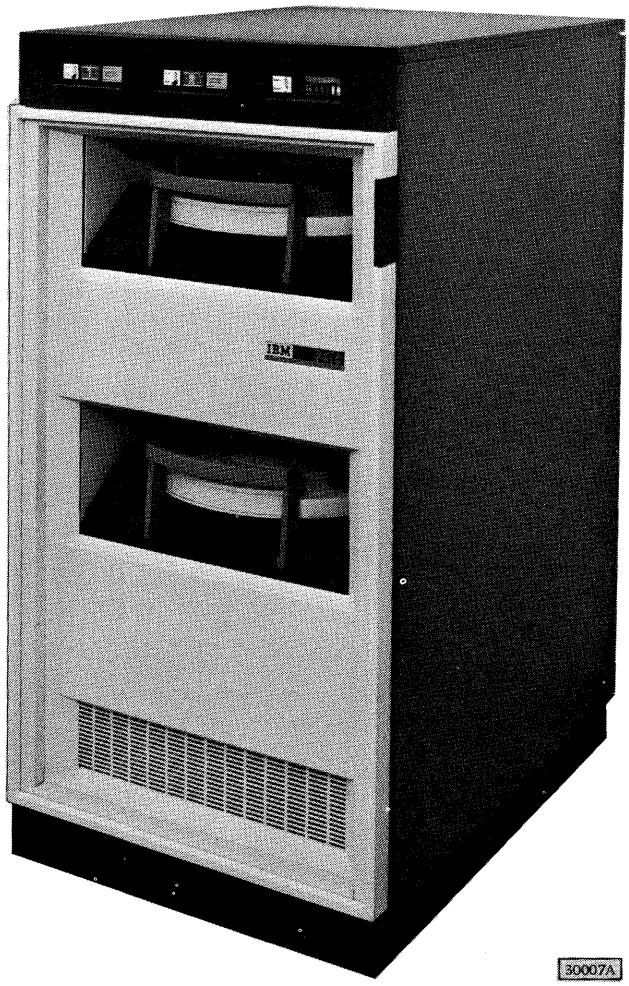


Figure 15. IBM 2310 Disk Storage Model B2

PROGRAMMING SINGLE DISK STORAGE

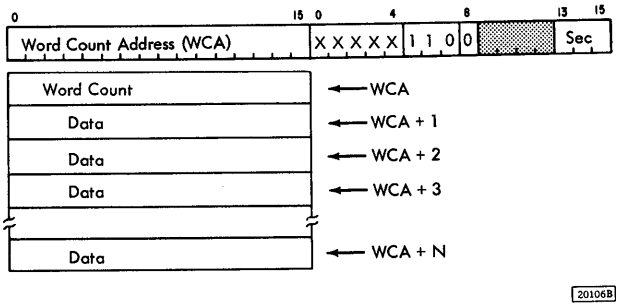
The single disk storage drives in the 1130 system are controlled by I/O control commands (IOCC) provided by the program in CPU core storage. Each of the five drives available responds to its assigned device code. The five-bit device field in the IOCC identifies the specific disk storage drive as follows:

Drive No.	Device Code	Device Location
0	00100	CPU
1	10001	2310 1st Drive
2	10010	2310 2nd Drive
3	10011	2310 3rd Drive
4	10100	2310 4th Drive

I/O Control Commands

Initiate Read (110)

This command causes the number of words specified by the word count to be read from the disk storage drive identified by the device code. The sector to be read is identified by modifier bits 13-15.



The address word of the command contains the word count address (WCA), and modifier bit 8 determines whether the command is a read command (0) or a read-check command (1).

A full sector, 321 words, is the maximum transmission with one command. Succeeding sectors, or parts of sectors, require the initiate read command for each one.

An operation-complete interrupt occurs when the number of words in the word count has been transmitted.

Read (Bit 8 = 0): Beginning with the first word of the indicated sector, data is read into core storage location WCA + 1 and ascending addresses. The word count, stored at the location specified by the WCA, controls the number of words transmitted and, consequently, the number of core storage locations occupied by the disk storage data. For example, assume that a word count of 152 is stored at WCA 1000. The 152 words read from disk storage would be stored at addresses 1001 through 1152.

The programmer must be aware of the core storage locations required for incomming disk storage data so that useful data is not written over and lost.

Read-Check (Bit 8 = 1): Data is read from disk storage, as in the read command, and the number of bits of each word is checked for modulo 4. Unlike the read command, data is not transferred into core storage. Therefore, a storage area does not need to be provided, and no time demand is placed on the CPU. Once the read-check command has been started, the modulo 4 checking is independent of the CPU. If the number of bits in a word, including check bits, is not even when divided by four, the data error indicator is set in the disk storage DSW. Neither disk storage nor core storage is affected by the read-check command.

To achieve the maximum level of performance that the disk storage is capable of providing, the program should provide error recovery procedures. Errors are often due to temporary conditions which can be sucessfully recovered by re-executing the read or write command.

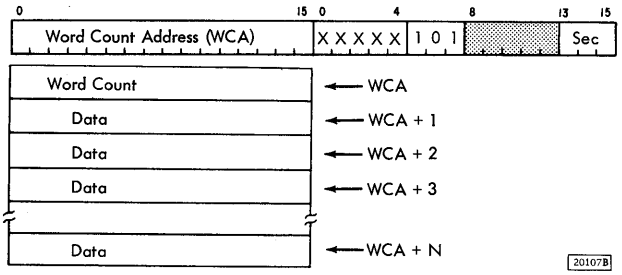
A write command which does not write correctly because of temporary or intermittent conditions can be detected by immediately verifying the data just written. In this way, any such "soft" write error can be corrected while the data is still available in main storage. If this write checking procedure is not followed, the "soft" write error becomes a "hard" error, which can be corrected only by reconstruction or adjustment. In almost all cases, permanent data files should be verified as soon as written, while for transient or work files, verification may not always be required. **The programmer should weigh the possible reconstruction time versus the time consumed in write verification before deciding not to verify write data.**

An initiate read with a word count of zero should not be used.

Initiate Write (101)

This command causes the number of words specified by the word count to be written in disk storage, beginning at the first word of the sector indicated by modifier

bits 13-15. The disk storage drive to be used is designated by the device code, bits 0-4.



The address word of the command contains the address of the word count. The data is transmitted from core storage location WCA + 1 and ascending addresses until the number of words specified by the word count has been written. If the word count is less than 321 words, the remainder is written with all 0's. Succeeding sectors, or parts of sectors, require an initiate write command for each one.

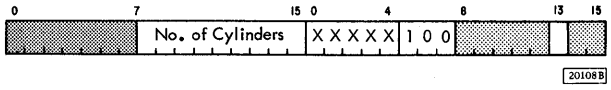
An operation-complete interrupt occurs when the number of words in the word count has been transferred.

An initiate write command should be followed immediately by a read-check command to verify that data can be read correctly.

An initiate write command with a word count of zero should not be used.

Control (100)

This command causes the access mechanism of the drive designated by the device code to move in increments of two cylinders for the number of cylinders specified by the address word of the command. If the number of cylinders is odd, the first increment consists of one cylinder.



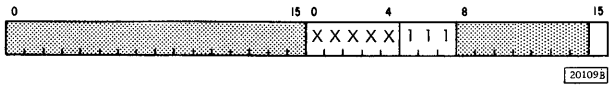
Modifier bit 13 controls the direction of movement: a 0 moves the access mechanism forward (toward the center of the disk); a 1 moves it backward.

When the access mechanism has moved the number of cylinders specified, an operation-complete interrupt occurs.

Note: Cylinders do not carry an identifying number. It is the responsibility of the program, therefore, to maintain the necessary information relative to the position of the access mechanism. A control command which specifies an access motion of zero cylinders is treated as a no-operation and does not result in an operation-complete interrupt.

Sense Device (111)

This command causes the device status word (Figure 16) of the disk storage identified by the device code to be read into the ACC.



Operation complete and data error (except select and unsafe) indicators are reset if modifier bit 15 is a 1.

DSW Indicators

Operation Complete (Interrupt): This is the only interrupt associated with disk storage, and is turned on at the end of a read, read-check, write, or control (access) operation. It also occurs if the disk storage is in a read, read-check, or write operation at the leading edge of a sector pulse; this occurs if the word count specified is greater than 321.

Data Error: This indicator is turned on when:

1. A modulo 4 error is detected during a read, read-check, or write operation,
2. The disk storage is in a read or write mode at the leading edge of a sector pulse.
3. A write select error has occurred in the disk storage drive.
4. The power unsafe latch is set in the attachment.

Conditions 1 and 2 are turned off by a sense device Command with modifier bit 15 set to 1. Conditions 3 and 4 are reset by turning off the disk storage drive, allowing for the cartridge unlock indicator

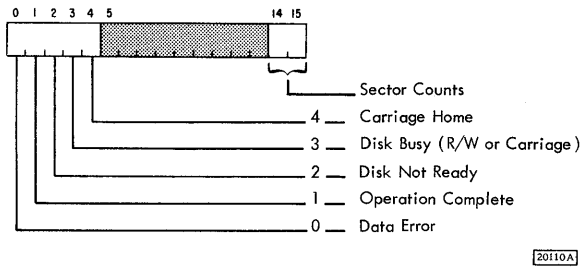


Figure 16. Disk Storage Device Status Word

to light, turning on the drive and waiting until the disk ready indicator (heads loaded for the 2310) to light. After this sequence of events disk operation can resume.

Operation Complete: This indicator is turned on at the end of:

1. A read, read-check, or write when the word count is reduced to zero.
2. A read, read-check, or write operation at the leading edge of the following sector pulse if the word count is not reduced to zero by that time; data error is also turned on.
3. A carriage seek operation in which there was access movement.

Disk Not Ready: This indicator is turned on with disk not ready or busy or disabled or off-line or power unsafe latch set. Also included in the disk not ready is the write select error, which can be a result of power unsafe or write select. (Bit 0 and bit 2 will be turned on.)

Disk Busy (R/W or Carriage): This indicator is on during execution of a disk storage command. It turns off when the operation is completed.

Carriage Home: This indicator is on when the access mechanism is at the home position (cylinder 000).

Sector Count: These bits represent the sector number of the next available sector to be used for reading or writing.

Programming Considerations

Disk Organization

It is important in planning a routine for loading disk storage that the cylinder concept be taken into consideration. Related data should be grouped in the same cylinder, when possible, to eliminate unnecessary seek operations. Therefore, when disk addresses are assigned to a group of related data, the disk locations made available should be limited to the number required, plus an expansion factor. The most frequently used data should be stored in the low-numbered cylinders to minimize seek time.

Customer Error-Correction Routines

If an error is detected by the CPU circuitry, the following procedure should be executed:

1. Re-seek the cylinder upon which the error was detected.
2. Re-execute the operation in which the error occurred.

This procedure should be executed from three to ten times prior to establishing the occurrence of a disk error.

Note: IBM-supplied disk subroutines perform standard error recovery procedures.

2310 Usage Meter

This meter will run when the following conditions are present:

1. The 1133 is in an enabled status.
2. The 2310 enable/disable switch is in the enable position. The CPU must be stopped when the switch is changed from one position to another to affect the enable/disable status.
3. The 1131 usage meter is running. The 2310 meter will run simultaneously with the 1131 meter until the status is disabled or the 1133 status is disabled.

Eighty-column punched card input and output is provided to the 1130 system by the IBM 1442 Card Read Punch, Model 5, 6, or 7, and/or the IBM 2501 Card Reader.

IBM 1442 CARD READ PUNCH

The IBM 1442 Card Read Punch (Figure 17), Model 6 or Model 7, provides both card input and card output for the 1130. The 1442 Model 5 is a card punch only and is considered the companion unit to the 2501 to provide a separate card path for card input and output. However, a model 6 or 7 may be installed with the 2501 in place of a model 5; in this case the 2501 should be considered the primary input unit. Functionally, the 1442 model 5 has the same punching characteristics as the 1442 model 7.

The 1442 is a single unit that processes cards serially, column by column, from a single supply hopper. All cards first pass the read station (model 6 and 7), then the punch station. This permits each card to be read, punched, or read and punched. Reading and punching cannot occur simultaneously -- that is, one card cannot be punched while the following card is being read -because the reading and punching rates are different.

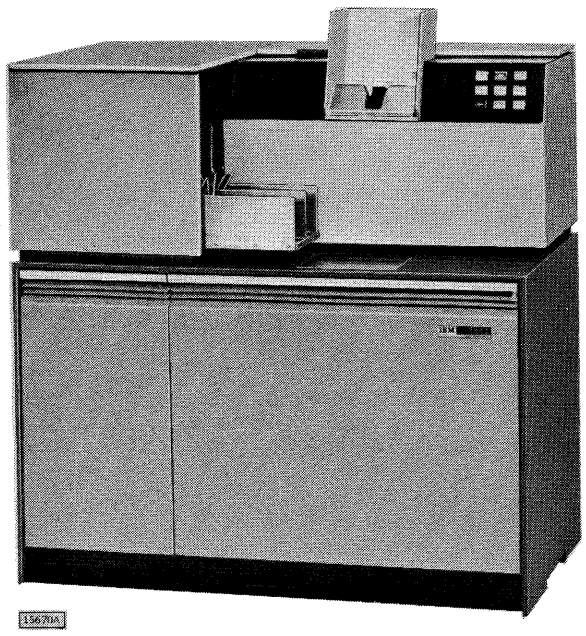


Figure 17. IBM 1442 Card Read Punch

Maximum machine speeds are:

Card Reading: Model 6, 300 cards per minute
Model 7, 400 cards per minute.

Card Punching: Model 5, 160 columns per second
Model 6, 80 columns per second
Model 7, 160 columns per second

Maximum reading rates are attained only when successive start read commands arrive early enough to re-energize the read clutch before the clutch latch point is reached. To accomplish this, successive start read commands must arrive within 35 milliseconds, model 6, or 25 milliseconds, model 7, after the operation is complete interrupt is given by the card read punch. If a start read command does not arrive within this time, the maximum reading rate becomes 285 cards per minute for model 6 and 375 cards per minute for model 7.

Punching rates depend on the position of the card when the last column has been spaced or punched. The punching speed ranges are:

Model 6 - 49 cpm to 262 cpm
Model 5 and 7 - 91 cpm to 355 cpm

The approximate time required to process a single card is:

Model 6 - 216 ms + 12.5 ms per card column spaced or punched
Model 5 and 7 - 163 ms + 6.25 ms per card column spaced or punched

The following table shows the approximate punch cycle times and cards-per-minute rates based upon the last column punched.

Last Column Punched	Punch Time (ms)		Total Punch Cycle Time (ms)		Cards per Minute	
	Model 6	Model 5&7	Model 6	Model 5&7	Model 6	Model 5&7
1	13	6	229	169	262	355
10	125	63	341	226	176	265
20	250	125	466	288	127	208
30	375	188	591	351	102	171
40	500	250	716	413	84	145
50	625	313	841	476	71	126
60	750	375	966	538	62	112
70	875	438	1091	601	55	100
80	1000	500	1216	663	49	91

30129

Data Coding

The card read punch reads and punches IBM card image only. Code translation must be done by the stored program. As shown in Figure 18 the twelve rows (12-9) in a card column correspond to the 0-11 bits, respectively or a core storage word. A 1-bit represents a punched hole; a 0-bit

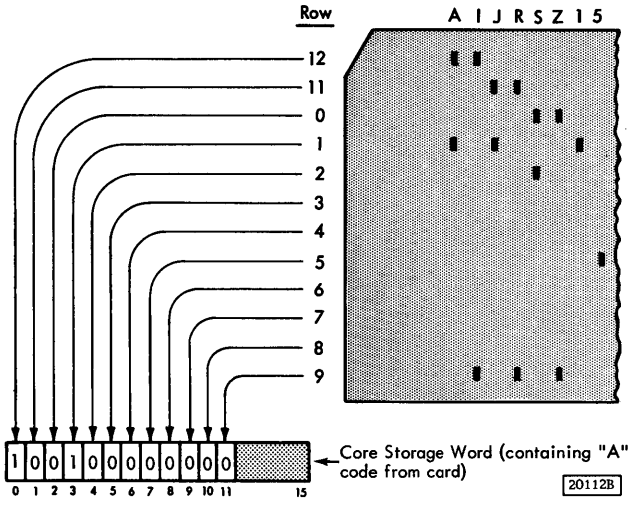


Figure 18. Normal Mode Read

represents a card position not punched. Thus, the word in Figure 18 contains 1-bits in bit positions 0 and 3 to represent the "A" read from the card. For output, a 1-bit results in a hole punched in the related position of the card read column.

A special load mode is initiated by pressing the program load key on the 1130 console. In the load mode data is split (Figure 19) as it enters core storage to form the load program.

Card Feeding

An initial feed cycle results when the 1442 start key is pressed; this feeds the first card into position at the read station (sense station - 1442 Model 5).

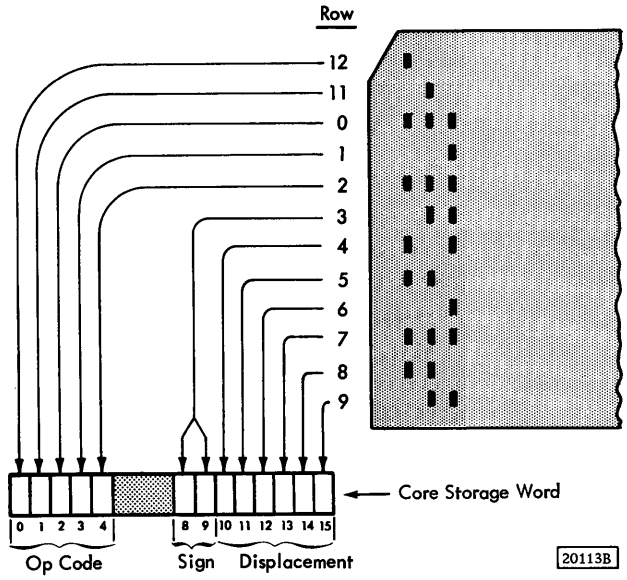


Figure 19. Load Mode Read

The initial feed cycle places the 1442 in a ready condition, which is necessary before reading or punching may begin.

A constant-speed drive moves the cards through the serial path during a feed cycle. A feed cycle is initiated by a Control command with modifier bits designating feed cycle, start read, or start punch. The feed cycle does three things:

1. It moves a card from the punch station to the stacker.
2. It moves a card through the read station and places it in the punch station with column 1 under the punches.
3. It moves a card from the hopper to the read station.

An incremental drive moves the card through the punch station for punching.

When the hopper is emptied, the operator can either reload the hopper and continue operations or he can initiate a last-card sequence.

Card Reading

A control (start read) command initiates card reading. This command causes columns 1-80 of the card to be read in one continuous motion of the card. Each column of data is read, checked, and placed in a buffer register. A read response interrupt is given for each column read. Checking is accomplished automatically by reading each column twice and comparing the results bit by bit. This read-check-interrupt process continues until all 80 columns have been read. The last card indicator in the DSW is turned on if the card read is the last card in the deck.

Card Punching

A control (start punch) command initiates card punching. As each column passes the punch station a punch response interrupt is given.

Automatic checking is accomplished by comparing the punch check echo data with the single-character punch buffer, which contains the character from the CPU. Each column punched is checked at the same time that the punch response interrupt is given for the data of the next column tc be punched.

The card motion and punching process continues until the punch data word contains a one in the 12-bit position (punch data is in bits 0-11). When this end-punch bit is detected, that column is punched and the card is moved to the next column. An operation complete interrupt is given. No more punch response

interrupts are given. No further punching can take place on the card.

Failure to have an end-punch bit 12 results in more than 80 columns being punched. No indication of this programming error is presented in the DSW.

A feed cycle is necessary to eject a punched card to the stacker and can be initiated by either of the three control commands: feed cycle, start read, or start punch.

A control command specifying start punch results in a feed cycle if it has not been preceded by a control command specifying feed cycle or start read.

Program Load

Program load can be initiated by pressing the program load key on the 1130 console after a system reset and the "run in" cycle of a load card. This load mode causes the load-card data to be placed in 80 consecutive storage positions beginning at 00000, then causes the CPU to go to position 00000 for its next instruction.

Last Card Sequence

The last-card indicator is turned on in the 1442 DSW when the last card passes the read station. The program determines when to enter the last-card sequence by testing the indicator.

When the start key is pressed without cards in the hopper, the 1442 is placed in the ready condition and allows two more feed cycles to process the last card.

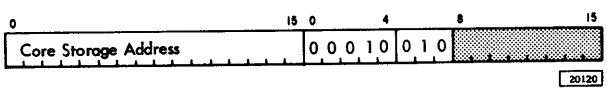
PROGRAMMING

The 1442 operates under direct program control of the CPU.

I/O Control Commands

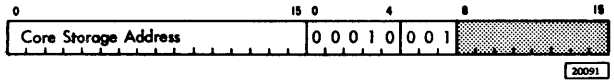
The card read punch is addressed by the 5-bit device code, 00010.

Read (010)



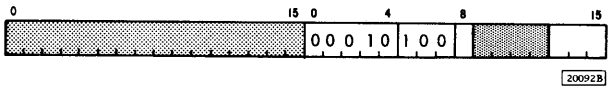
This command causes a card column image to be entered from the card reader into the core storage location specified by the address.

Write (001)



This command causes the data in the core storage location specified by the address of the IOCC to be punched as a column of the card.

Control (100)



This command causes the 1442 to perform the function specified by the modifier.

Modifier bits that have significance are:

Bit 14 — Feed Cycle - causes all cards in the feed path to advance one station. Cards pass through the read and punch stations without being processed.

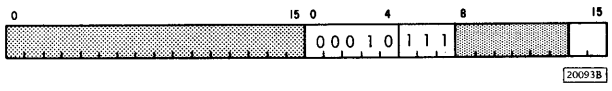
Bit 13 — Start Read - causes the card to move through the read station. As each column is read and checked, the card read punch initiates a read column response interrupt.

Bit 15 — Start Punch - starts the punching operation and initiates a punch response interrupt. If a card is not at the punch station, a card will feed past the read station without being read.

Bit 8 — Stacker Select - (model 6 or 7 only) causes the card leaving the punch area to enter the alternate stacker. This control applies only to the next card leaving the punch station after this command has been issued.

Modifiers bits 13, 14, and 15 of this control command should not be used in combination with each other.

Sense Device (111)



This command directs the 1442 to place its device status word (Figure 20) into the ACC. Modifier bit 15 on resets responses for level 0; modifier bit 14 on resets responses for level 4.

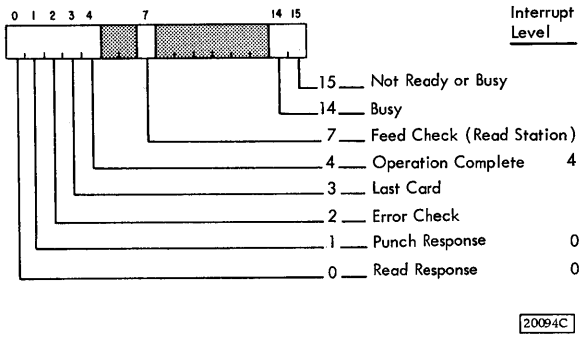


Figure 20. 1442 Device Status Word

DSW Indicators

The three interrupts associated with the 1442 are divided into two groups. The sense interrupt (011) command causes the active ILSW to be loaded into the ACC.

Level 0 Interrupt Indicator

Read Response (Interrupt): This indicates an interrupt which signals that a column of data is ready to be entered into core storage. This interrupt request must be serviced within 800 us. for the 1442 model 6 and 700 us. for the 1442 model 7. Time from the start read to the first read column request interrupt is 28.4 ms for the model 6 and 23.8 ms for the model 7.

Punch Response (Interrupt): This indicates an interrupt which signals that a column of data must be transferred from the CPU within 300 us. Time from the start punch command to the first punch column response interrupt varies from 1.22 ms to 12.5 ms on the model 6 and 1.56 ms to 6.25 ms on the model 5 and 7.

Level 4 Interrupt Indicator

Operation Complete (Interrupt): This indicates an interrupt which occurs after a card has been processed. For reading, it indicates that column 80 of the card has passed the read station. This interrupt occurs 20.6 ms after column 80 for the model 6 and 15.4 ms after column 80 for the model 7.

For punching, this interrupt occurs after the last column to be punched has been punched and checked and the punch drive has stopped. This occurs 12.5 ms after the terminating end-punch has been detected for the model 6 and 6.25 ms after the terminating end-punch for the model 5 and 7.

The operation complete interrupt is forced if a hopper check, feed check, transport error, or feed clutch error occurs while the 1442 is busy. This interrupt is also forced by a read registration check or punch check. No subsequent reading or punching can be done in the card that caused the error. In most cases, intervention by the operator is necessary to clear the error condition before card processing may resume.

There is no time limit on the request for service of the operation complete interrupt. However, to maintain rated speed, the model 5 and 7 must be serviced within 25 ms; the model 6 must be serviced within 35 ms if reading and 25 ms if punching.

Non-Interrupt Indicators

Not Ready: This indicator shows that the 1442 is either busy or not ready. When the 1442 is not ready, manual intervention is required. The following conditions must be met to place the 1442 in a ready condition.

1. Power on.
2. Card registered at the read (sense for Model 5) station (initially).
3. Cards in hopper or last-card sequence in progress.
4. Stacker not full.
5. Feed-check light off (no card jam or feed failure).
6. If the stop key has been pressed, the start key must have been subsequently pressed.
7. Chip box not full or removed.

Busy: This indicator shows that a command cannot be started because an operation is already in progress.

Last Card: This indicator shows that column 80 of the last card has passed the read station and the hopper is empty. This indicator will be on when the operation complete interrupt occurs.

Error Check: Indicates that any of several error conditions exist on the 1442. Error conditions such as card feed failure are indicated by lamps on the 1442 console.

Programming Note: The error indicator does not turn on until after the operation complete interrupt is given. An exception to this is an XIO start punch operation requiring an automatic feed cycle. If another operation is initiated before the error indicator is turned on, the error forces an operation complete interrupt although no reading or writing has taken place. A start punch requiring an automatic feed cycle is treated as two

operations: (1) feed cycle, and (2) punch operation.

1442 Usage Meter

This meter will run when both of the following conditions are present:

1. The unit is selected for operation by program control.
2. The 1131 usage meter is running.

The meter will run simultaneously with the 1131 meter until either a program controlled stop or manual nonprocess runout is performed on the machine.

IBM 2501 CARD READER

The IBM 2501 Card Reader (Figure 21), Model A1 and Model A2, provides card input for the IBM 1130 Computing System. Card reading is under direct program control.

FUNCTIONAL DESCRIPTION

The IBM 2501 model A1 reads cards at a maximum rate of 600 cards per minute (cpm); the model A2 reads at a maximum rate of 1000 cpm.

Cards are read serially -- that is, column by column -- beginning with column 1. Each column is read twice and the two readings are compared to check reading accuracy. Thus, off-punched and mispositioned cards are detected.

Data Coding

The 2501 reads punched cards in card image only. Any code translation required must be done by the stored program in the CPU. As shown in Figure 18, the twelve rows (12-9) in a card column correspond to the 0-11 bits, respectively, of a core storage word.

A special load mode is initiated by pressing the program load key on the 1130 console. In the load mode, data is split as it enters core storage to form the load program. Refer to Figure 19.

Card Feeding

After the initial feed cycle (run in), card reading may begin. Card feeding is initiated by an initiate read command. This command causes the card to begin moving. If the data is to be ignored (as in card feeding), the word count must be zero.

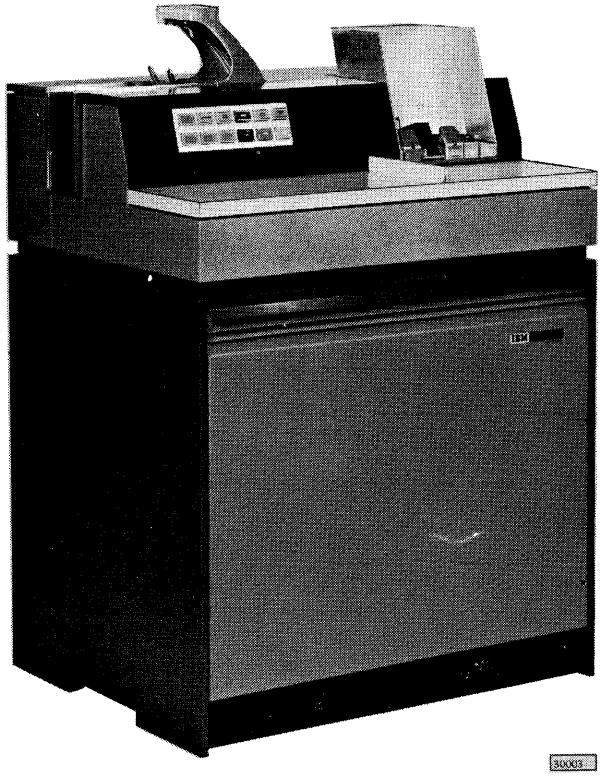


Figure 21. IBM 2501 Card Reader

Card movement is as follows:

1. The card at the read station moves through the read station to the stacker.
2. A card moves from the hopper to the read station.

When the hopper is emptied, the 2501 leaves the ready condition. The operator may reload the hopper and press the start key to continue processing or the operator may press the start key without reloading the hopper to initiate the last card sequence.

Program Load

Program load may be initiated by pressing the program load key on the 1131 console. This causes the load card data to be loaded into the first 80 core storage locations. After the card has been loaded the instruction address register is reset to 00000 and the CPU goes to this address for its next instruction.

Card Reading

An initiate read (110) command causes the card to be read in one continuous motion.

The number of columns actually transferred to core storage depends upon the word count in the first word of the data table.

The data is read into a buffer register where it is checked. Then a cycle steal request is given for each column to be transferred. The checking is accomplished by reading the data a second time and comparing it to the data previously read into the buffer. After the last column (column 80) has been read, an operation complete interrupt is requested.

Last Card Sequence

When the hopper becomes empty during a feed cycle, the 2501 is taken out of the ready status. Intervention by the operator is required to continue processing cards. The operator may reload the hopper and press the 2501 start key to continue processing or the operator may initiate the last card sequence by pressing the 2501 start key with the hopper empty.

The last card sequence places the 2501 in the ready condition for one more feed cycle and turns on the last card indicator. An operation complete interrupt is given at this time. The last card indicator remains on until a sense device command (111) is issued with bit 15 on.

PROGRAMMING

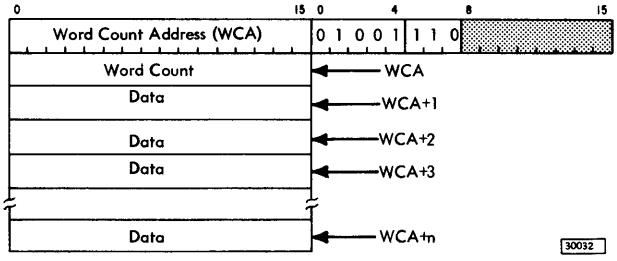
The IBM 2501 Card Reader operates under the control of the stored program in the CPU.

I/O Control Commands

The 2501 is addressed by the five-bit device code 01001.

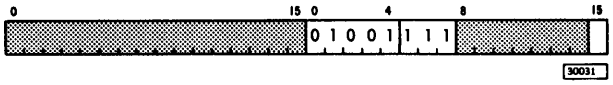
The address portion of the IOCC specifies the location in core storage of the data table. The first word of the data table designates the number of words to be read. This word is called the word count. The word count is located in bit positions 9 through 15 and should never exceed 80 (50 hexadecimal). If the word count exceeds 80, information in succeeding core locations is destroyed.

Initiate Read (110)



This code provides the ability to start a read operation, which subsequently makes data transfers to core storage via a data channel by means of cycle stealing.

Sense Device (111)



This command sets the accumulator with the device status word (DSW) of the 2501 The DSW bits 3 and 4 (last card and operation complete) are reset if bit 15 is on when this command is executed.

DSW Indicators

Not Ready or Busy: This indicates that the 2501 is not in a ready condition, or that it has received an instruction and is in the process of executing it.

Busy: This indicates that a card read is in progress and therefore another read card cannot be initiated. This indicator turns off when the operation complete interrupt occurs.

Operation Complete (Interrupt): This is the only interrupt associated with the 2501. This interrupt occurs after column 80 has passed the read station and feed checking has been completed. The operation complete interrupt is independent of the word count and terminates further cycle steal requests. The number of characters actually transferred to the CPU depends upon the word count. The 2501 is assigned to interrupt level 4. Bit 3 in ILSW 4 is turned on if the 2501 caused the interrupt. The sense interrupt (011) command causes ILSW 4 to be loaded into the accumulator if level 4 is being serviced.

Last Card: This indicates that the last card has been fed from the hopper, and the operator initiated a last-card sequence. The indicator may be turned off by a sense device command with reset (bit 15) on.

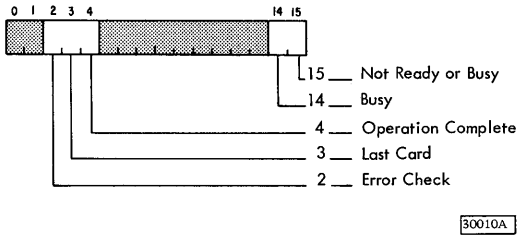


Figure 22. 2501 Device Status Word

Error Check: This indicates a feed check or a read check.

Reader and System Timing

There are two basic timing considerations of importance to the user of a 2501 Card Reader attached to an IBM 1130 Computing System:

1. Card throughput in cards per minute (cpm).
2. Time available for other system operation.

System Operations

The 1131 is capable of performing operations (such as reading, processing, and punching) simultaneously. After an operation is initiated, the CPU is busy for only 288 microseconds. The remainder of the card read cycle is available for other use.

Card Throughput

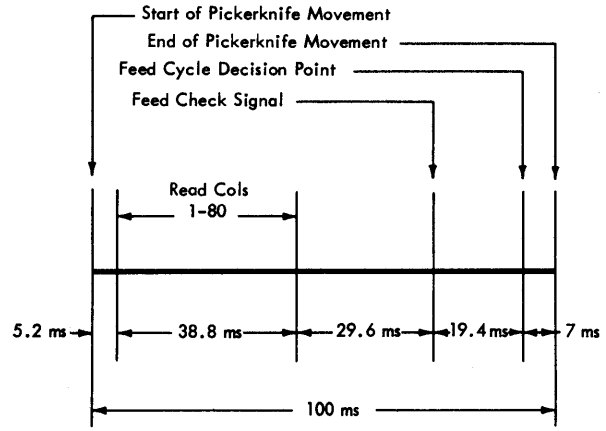
The 2501 model A1 has a 100-ms card feed cycle; the model A2 has a 60-ms card feed cycle. To maintain the rated speed, an initiate read command must occur every 100 ms for the model A1 and every 60 ms for the model A2. A basic timing consideration of importance to the user of the 2501 Card Reader is the time between the feed check signal and the feed cycle decision point. This timing is shown in Figure 23. In order to maintain rated throughput, the read instruction must be received within 18.3 ms (A1) and 3.0 ms (A2) following an interrupt (feed check signal).

If an initiate read command misses the feed cycle decision point, the card reader waits until the feed cycle decision point of the next cycle before starting to execute the command. The result in this case is the throughput is about one-half of the maximum.

2501 Usage Meter

This meter runs when both of the following conditions are present:

Model A1 - 600 CPM (All times shown are nominal at rated thruput.)



Model A2 - 1,000 CPM (All times shown are nominal at rated thruput.)

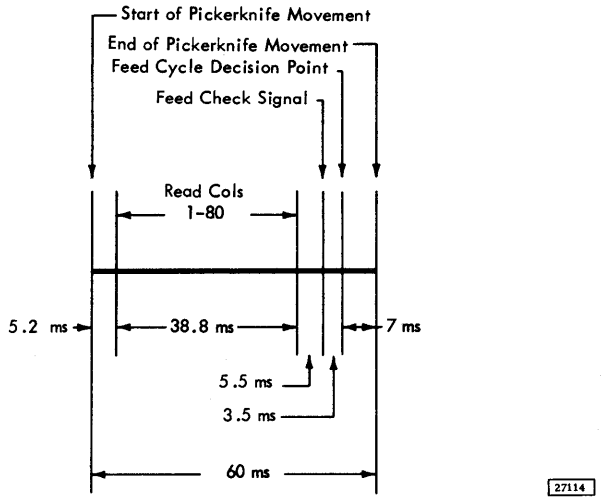


Figure 23. 2501 Timing Schematic

1. The unit is selected for operation by program control.
2. The 1131 usage meter is running.

The meter will continue to run simultaneously with the 1131 meter until either a last card routine is initiated by program control or a manual nonprocess runout is performed.

PAPER TAPE INPUT/OUTPUT DEVICES

The IBM 1055 Paper Tape Punch (Figure 24) and the IBM 1134 Paper Tape Reader (Figure 25) provide paper tape I/O for the 1130.

The 1134 and 1055 operate under direct program control.

The 1134 reads one-inch, eight-channel paper tape at a maximum rate of 60 columns per second.

The 1055 is capable of punching eight-channel chad paper tape or edge-punched documents that have prepunched feed holes. The 1055 punches at the rate of 14.8 characters per second.

Tape Specifications

Both the 1055 and the 1134 are capable of using paper tape, Mylar* laminated paper tape, and Mylar coated aluminum tape that meet the specifications in Figure 26.

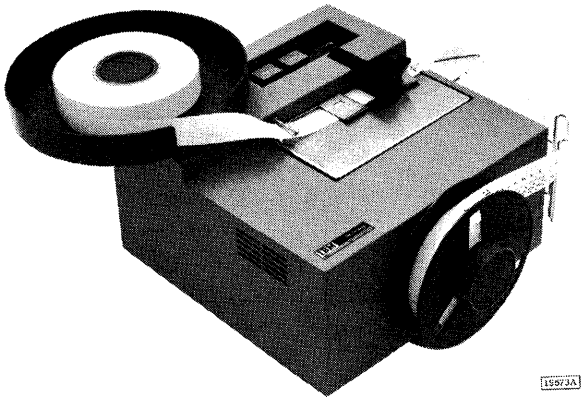


Figure 24. IBM 1055 Paper Tape Punch

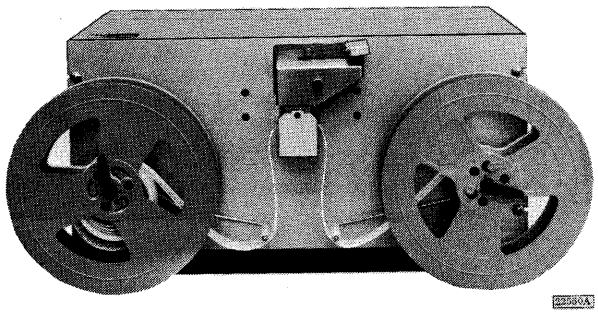


Figure 25. IBM 1134 Paper Tape Reader

Character Code

The 1134 reads input data into the core storage as an image of the holes in the tape. One paper tape character is read into each addressed core storage location. Any code translation must be made by programming.

Figure 27 indicates which bits of the word correspond to the respective holes in the paper tape read by the 1134.

The 1055 punches data as an image of the data contained in positions 0-8 of the core storage word as shown in Figure 27.

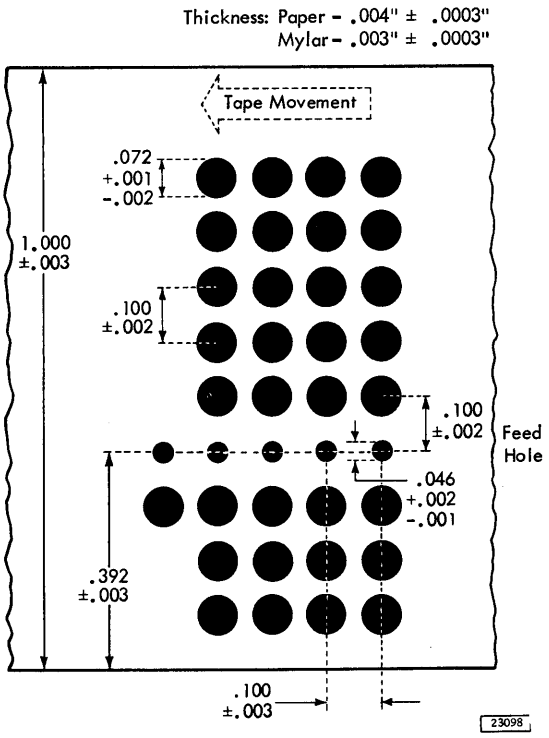


Figure 26. Tape Specifications

*Trademark of E.I. duPont de Nemours Company

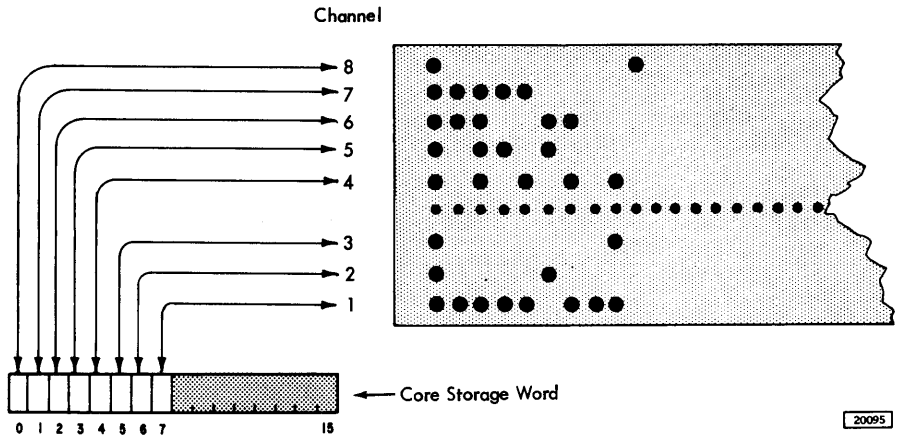


Figure 27. Paper Tape/Core Storage Format

Program Load from 1134

An 1130 system that does not have card I/O will have the program load feature added to the 1134. This feature operates by means of design logic rather than program control. Four-bit paper tape characters are automatically assembled into four-character groups to form 16-bit data words. The program load feature then loads these words into core storage beginning at location 00000.

Only the first four (1-4) tape channels are used. When a channel 5 punch is encountered, program loading stops; the IAR is reset to zero; and program control begins at 00000. Delete characters are permitted at the beginning, but once the program begins to load, the channel 5 in a delete character will end the load.

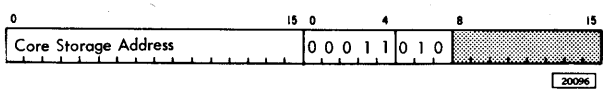
PROGRAMMING

The IBM 1134 Paper Tape Reader and the IBM 1055 Paper Tape Punch operate under direct program control with the exception of the paper tape program load feature.

I/O Control Commands (IOCC)

The 1134 and 1055 are addressed by the same five-bit device code, 00011.

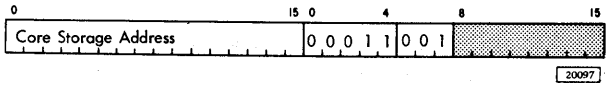
Read (010)



This command reads one character from paper tape into core storage.

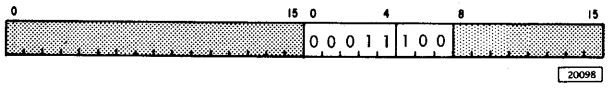
The address word specifies the location in core storage where the tape character is to be stored.

Write (001)



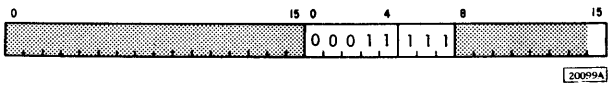
This command writes one character from core storage to the paper tape punch. The address word specifies the location in core storage where the tape character is stored.

Control (100)



This command must be given prior to each character to be read from the 1134. Execution of this command causes: (1) one character to enter the paper tape reader buffer, and (2) the tape to be advanced one column. A reader service response interrupt is initiated to indicate that a character from paper tape can be read into the core storage location specified by a subsequent read (paper tape) command.

Sense Device (111)



This command is used to enter the device status word (Figure 28) into the ACC. Modifier bit 15 on indicates that the responses are to be reset.

DSW Indicators

Reader Response (Interrupt): This indicates an interrupt which occurs on

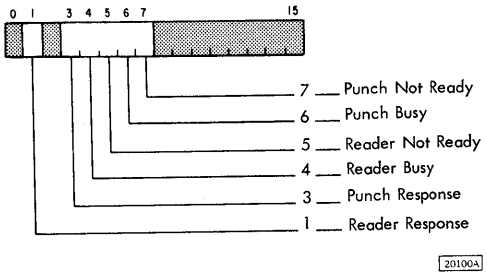


Figure 28. Paper Tape Device Status Word

level 4 when the reader has completed the execution of a control command. This interrupt indicates to the CPU that a character is available to be entered into core storage.

Punch Response (Interrupt): This indicates an interrupt which occurs on level 4 when the punch has completed punching as directed by the execution of a write command. It indicates that the punch can accept the next command.

Punch Not Ready: This indicator is on when the tape is not feeding freely from the tape spool, when the tape pressure roll holder is not down and holding the tape against the feed wheel, or when tape is not present. Manual intervention is required to clear these conditions. The indicator is also on if the punch is busy. (See punch busy indicator.)

This indicator should always be tested by the program before a write command is given. If a write command is given while this indicator is on, loss of information will probably occur. No indication is given of this loss.

Reader Not Ready: This indicator is on when the tape tension switch is open. This condition exists when the paper tape is broken or not feeding freely. Manual intervention is required to clear these conditions. This indicator is also on if the reader is busy. (See reader busy indicator.)

The program should test this indicator before a read command is given. If a read command is given while this indicator is on, erroneous data can be read into core storage. No valid indication can be given as to whether the data read is correct or incorrect.

Punch Busy: This indicator is on for the total time the punch is mechanically engaged and punching a character (68 ms). During this time the punch should not be sent another write command.

Reader Busy: This indicator is on from the time a control command (start paper tape reader) is given until data is available. A reader response interrupt signals that data is available.

Two on-line printers are available for attachment to the 1130 system. A system may include an IBM 1132 Printer and/or an IBM 1403 Printer, Model 6 or Model 7. The 1403 attachment also requires the attachment of the IBM 1133 Multiplex Control Enclosure.

IBM 1132 PRINTER

The 1132 Printer (Figure 29) provides printed output for the 1130 system at maximum rates of 80 lines per minute (lpm) for alphamerical printing and 110 lpm for all-numerical printing. The print line is 120 print positions long; horizontal spacing is ten characters per inch. Vertical spacing, which is preselected by the operator, is six or eight lines per inch.

FUNCTIONAL DESCRIPTION

The 1132 contains a printwheel with 48 alphabetic, numeric, and special characters for each of the 120 printing positions. Special (FORTRAN) characters are as follows:

```
¢-/.$,*()'+=
```

All printwheels rotate continuously and in synchronization with each other. Each wheel moves forward to print when the data in the output record specifies that the character to be printed is in position. Thus, all similar characters for the entire line are printed on the same cycle. Forty-eight cycles (one for each character possible) are required to print a complete line.

The 1132 uses interrupt circuitry and responds on level 1.

Figure 29. IBM 1132 Printer

Forms Control

Forms control is provided through a tape-controlled carriage that uses the standard IBM carriage tape. Channels 1 through 6,9, and 12 are available to the stored program.

Spacing is always performed one line at a time under control of the stored program in the CPU.

Carriage skipping is initiated by the stored program and stopped by the program when the predetermined line is reached. Skipping speed is 10 inches per second.

Data Format

The 1132 character code is shown in the appendix. Each character occupies the first eight bits of a core storage word. The data to be printed is assembled in core storage in the same order, including spaces, as the line that is to be printed. During each of the 48 cycles necessary to print all 48 characters, the character next in position to print is read from the character emitter and is compared with each character of the output record, all by the CPU program. For each equal comparison, the program places a 1-bit in the printer scan field in the position corresponding to the printwheel to be fired. The printer scans the field in a cycle-steal mode and fires each printwheel whose position contains a 1-bit. The printer scan field is located in core storage locations 32 through 39. The 16 bits of each of the first seven words and bits 0 through 7 of the eighth word represent the 120 printwheels.

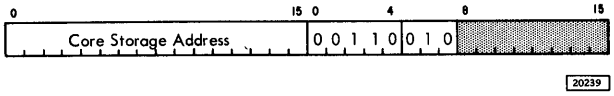
PROGRAMMING

The IBM 1132 Printer operates under direct program control of the CPU.

Printer I/O Control Commands

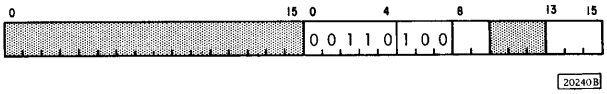
The 1132 is addressed by the binary device code of 00110.

Read Emitter (010)



This command causes the eight-bit code of the next character emitted by the printer to be read into the core storage location specified.

Control (100)



This command causes the execution of the function specified by the modifier bit. A 1 bit in the position indicated in parentheses after each command causes the operation described.

Start Printer (Bit 8): This causes the printer to start taking the printer scan field information. The printer continues to take print scan cycles at 11.2-ms intervals until it receives a stop printer command. Each position that contains a 1-bit causes the corresponding printwheel to print the character in position on that cycle. After the field of eight words has been scanned, a 1-bit is placed in bit position 0 of the 1132 device status word. (See Figure 30.) This causes an interrupt when level 1 is the highest level waiting.

Stop Printer (Bit 9): This instruction causes the printer to be put in a ready (not busy) state and inhibits subsequent printer interrupts. The stop printer instruction should not be given until all of the following conditions are met:

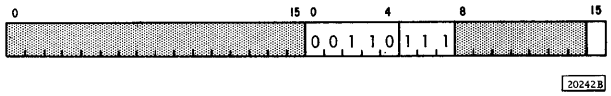
- Eighteen scan cycles have been completed after the command to print the last character.
- The carriage has stopped after a skip operation.
- The interrupt response from the last space command has occurred.

Start Carriage (Bit 13): This command initiates a skip operation, which is halted by a stop carriage instruction.

Stop Carriage (Bit 14): This command stops the carriage at the end of a skip operation. A punch in carriage control tape channel 1, 2, 3, 4, 5, 6, 9, or 12 initiates an interrupt request, identified by bit 1 of the DSW. When the desired tape channel bit in the DSW is on, a stop carriage command should be given.

Space (Bit 15): This command is given to space the carriage one line. After the space operation, an interrupt is initiated and a 1-bit is put in bit position 2 of the DSW to indicate spacing is completed. Another space can now be initiated.

Sense Device (111)



This instruction causes the DSW of the 1132 Printer to be placed in the ACC. The functions of the bit positions of the DSW are shown in Figure 30.

If bit 15 contains a 1, the interrupt responses in the DSW are reset.

DSW Indicators

Three interrupts are associated with the 1132, each on level 1. The associated indicators are turned on in the DSW.

Read Emitter Response (Interrupt): After a start printer command has been executed, the 1132 will interrupt the CPU program each time the printwheels are aligned to print another character. The read emitter command must then be executed to determine the character to be printed.

Skip Response (Interrupt): This indicates an interrupt which is initiated by the 1132 each time the carriage brushes detect a punch in the carriage tape while a skip operation is in progress. The CPU program must test the DSW bits to determine if the carriage is at the proper channel.

Space Response (Interrupt): This indicates an interrupt which turns on at the completion of a space operation to signal the CPU program.

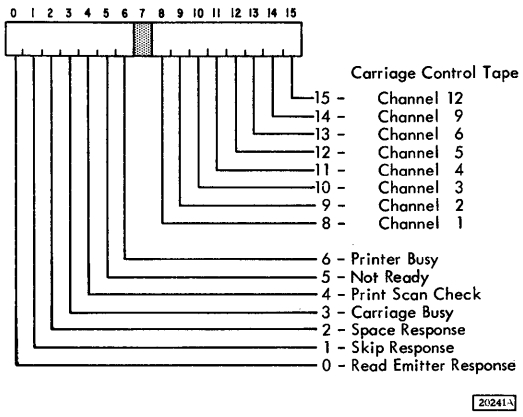


Figure 30. 1132 Device Status Word

Note: After an interrupt has been serviced, the level must be reset by a BOSC instruction.

Carriage Busy: This indicator turns on when the 1132 begins carriage movement. It turns off when movement stops.

Print Scan Check: This indicator is turned on when the printer attachment addresses word 39 and there is not a 1-bit in position 15. A 0 in bit 15 indicates that the printer subroutine did not finish setting up the print scan field.

Not Ready: This indicator is turned on by an out-of-forms condition, motor power off, or at the end of the operation in progress if the stop key is pressed.

Carriage Control Channels: As each hole in the carriage tape is read after a start carriage control command, the associated indicator in the DSW is turned on.

Programming Notes

The status of the 1132 indicators should be checked before a line is printed. This is accomplished by transferring the printer DSW into the ACC with a sense device command. The modifier bits of the sense device command should be set to 0's to prevent reset of the DSW responses and indicators. Bits 3, 5, and 6 of the DSW are tested and if all three positions are 0, the printer is ready to print the next line. A start printer control command is then given to start the sequence. A scan field transfer, using cycle steal cycles, takes place under control of the printer. Therefore, the scan field must be clear and have a 1-bit in position 15 of core storage word 39 before the start print command is given.

After the code of the next character has been emitted by the printer, a level 1 interrupt is given and the character is read into core storage by a read emitter command. There are 11.2 ms available to test each position of the output record with the character read and set up the 1 bits in the printer scan field. At the end of the 11.2 ms, the 1132 attachment begins its scan and fires each printwheel with a corresponding 1-bit in the printer scan field. If the program has been interrupted for a considerable period by higher levels, the scan may not have been completed. To insure that the program detects this condition, the first steps of the printer subroutine for each character should clear the printer scan field to 0's and, upon completion of the programmed scan, place a 1 bit in position 15 of the eighth word (39). When the printer attachment scans the field it checks this position. If it is 0, the print scan check indicator (bit 4 of the DSW) is turned on. The program can test this indicator and branch to an error routine that provides 47 idle scan cycles and resumes programmed scanning at the point where the scanning was interrupted. This results in overprinting of the characters that were printed unless the error routine keeps track of the positions that were printed and does not set them up again on this scan.

After the final scan cycle for a line of printing, 16 idle scan cycles must be taken before spacing or skipping is started to allow time for completion of the mechanical operation of printing the last character. If the operation is a single or double space, the next scan cycle can be started two scan cycles after the last space command is given.

During an idle scan cycle the printer scan field should be set to 0's, except for bit 15 of the eighth word (39), to prevent the print scan check indicator from being turned on.

1132 Usage Meter

This meter will run when both of the following conditions are present:

1. The unit is selected for operation by program control.
2. The 1131 usage meter is running.

The meter will continue to run simultaneously with the 1131 meter until it is stopped by the operator.

IBM 1403 PRINTER

The IBM 1403 Printer (Figure 31) greatly increases the output capabilities of the IBM 1130 Computing System while reducing the time that the CPU is required to print, thus leaving more time for other functions. The 1403 is available in two models for attachment to the 1130:

1. The model 6 has a maximum printing speed of 340 lines per minute (lpm).
2. The model 7 has a maximum printing speed of 600 lpm.

Each printer can print 48 different characters in 120 positions. There are 26 alphabetic, 10 numeric, and 12 special characters.

Vertical spacing and skipping are initiated by the stored program. Horizontal spacing is ten characters per inch. Standard vertical spacing is six and eight lines per inch, controlled manually by the operator. Skipping is about 33 inches per second.

Figure 31 IBM 1403 Printer

FUNCTIONAL DESCRIPTION

Printing

The alphabetic, numeric, and special characters are assembled in a chain. As the chain travels in a horizontal plane, each character is printed as it is positioned opposite a magnet-driven hammer that presses the form against the chain.

Data to be printed must first be edited, translated to the 1403 binary code (see Appendix), and arranged in core storage in exactly the form that it is to be printed. The data format in core storage is two seven-bit characters per word (Figure 32).

Spacing and Skipping

Spacing is always performed one line at a time under control of the stored program in the CPU.

Carriage skipping is controlled by prepunched holes in a paper or plastic tape that corresponds in length to the length of one or more forms. Holes punched in the tape stop the form when it reaches any predetermined position.

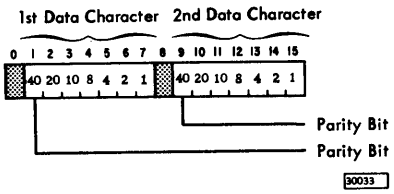


Figure 32. 1403 Data Format-Hexadecimal Bits

Control Tape

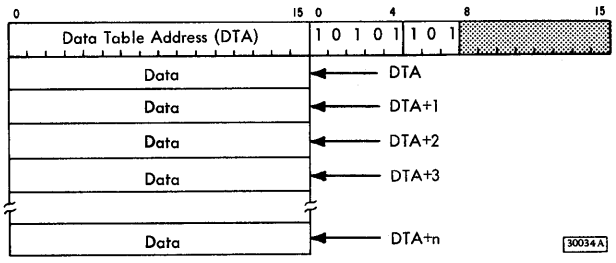
The control tape has 12 columns, indicated by vertical lines. These positions are called channels. Holes can be punched in each channel throughout the length of the tape. A maximum of 132 lines can be used to control forms although, for convenience, the tape blanks are slightly longer. Horizontal lines are spaced six to the inch for the entire length of tape. Round holes in the center of the tape are prepunched for the pin-feed drive that advances the tape in synchronization with the movement of a printed form through the carriage. The effect is exactly the same as though the control holes were punched along the edge of each form.

PROGRAMMING

All operations of the 1403 are under control of the stored program in the CPU.

Data to be printed must be edited, translated to the 1403 binary code, and arranged in core storage in exactly the form that it is to be printed.

Initiate Write (101)



An initiate write command transfers data from core storage to the print buffer using the cycle steal method.

During data transfer each core location to be printed is addressed twice. During the first cycle, bits 1-7 are transferred to an even address of the print buffer.

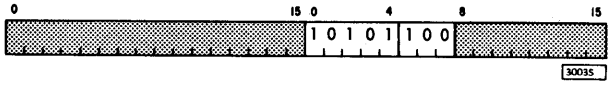
During the second cycle, bits 9-15 are transferred to the next higher odd address of the print buffer.

The total time demand on the processor is dependent on the core storage cycle time. Approximately 432 us. is required for the 3.6-us. core storage, and approximately 264 us. is required for the 2.2-us. core storage.

The printer does not interrupt the CPU until after the 120-position buffer is filled. It then initiates a transfer complete interrupt on level 4.

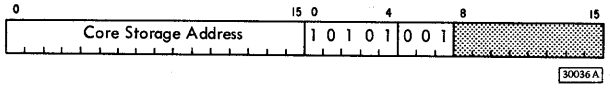
After completion of printing the line a level 4 interrupt is initiated to signal print complete.

Control (100)



An XIO control command initiates a single line space.

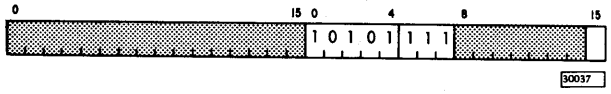
Write (001)



An XIO write command controls carriage skipping. This command causes the carriage to skip even if it is at the specified channel. It skips until that channel is detected again. The carriage may be controlled to skip to any channel 1-12 by placing a 1-bit in positions 4-15 of the core location specified by the address.

A carriage control command given prior to loading of the print buffer causes immediate execution of the command. If the command is given during loading of the buffer, the command is not executed until after the line is printed. The programmer must check to insure that the carriage is not busy when the command is given.

Sense Device (111)



An XIO sense device command causes the 1403 DSW (Figure 33) to be placed into the accumulator. If bit 15 is on when the command is executed, the DSW interrupt indicators and channel 9 and 12 indications are reset.

DSW Indicators

Transfer Complete Interrupt: The 1403 requests this interrupt when the 1403 buffer is full.

Print Complete Interrupt: This interrupt indicates the 1403 has completed printing a line.

Carriage Interrupt: This interrupt indicates the 1403 has completed a skip or space operation.

In addition to the preceding interrupts, the following status conditions are also indicated in the 1403 DSW.

Parity Check: This indicates an even bit count of the seven-bit print buffer data word.

Print Check: This indicates an error occurred in modification of the buffer address register.

Sync Check: This indicates that the print chain is not synchronized with the compare counter.

Carriage Channel 9: This indicates the carriage passed a channel 9 punch in control tape.

Carriage Channel 12: This indicates the carriage passed a channel 12 punch (normally used for the last printing line on a form) in the control tape.

Carriage Busy: This indicates that the carriage is performing a space or skip operation. This bit goes off when bit 3 comes on to signify completion.

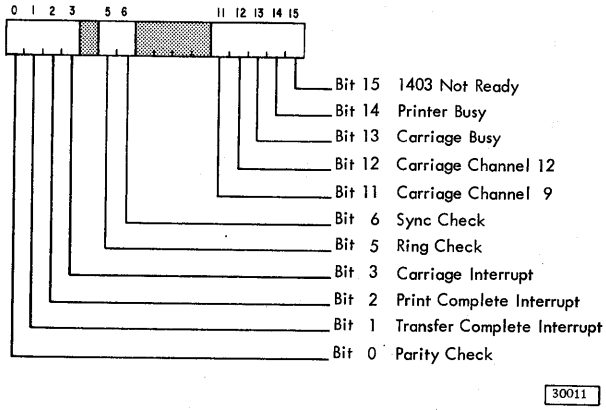


Figure 33. 1403 Device Status Word

Printer Busy: This indicates that the 1403 buffer is being loaded or a line is being printed.

Not Ready: This indicates the 1403 is not ready. Printing, spacing, or skipping under program control cannot occur until the 1403 is ready.

1403 Usage Meter

This meter will run when the following conditions are present:

1. The 1133 is in an enable status.
2. The 1403 is selected for operation by program control.
3. The 1131 usage meter is running.

This meter will run simultaneously with the 1131 meter until either stopped by the operator or the 1133 is placed in a disable status.

The IBM 1627 Plotter (Figure 34) provides an exceptionally versatile, reliable, and easy-to operate plotting system for the 1130 system. The plotter converts tabulated digital information into graphic form. Bar charts, flow charts, organization charts, engineering drawings, and maps are among the many graphic forms of data that can be plotted on the 1627.

Two models of the 1627 are available and the major characteristics are as follows:

Model 1 — Plotting area: 11 inches by 120 feet, 1/100 inch incremental-step size, 18,000 steps/minute.

Model 2 — Plotting area: 29-1/2 inches by 120 feet, 1/100 inch incremental-step size, 12,000 steps/ minute.

See Figure 35 for more information. The 1627 is equipped with a ball point pen, which last for about five or six hours of continous plotting. A liquid-flow ink pen is optional.

FUNCTIONAL DESCRIPTION

Data from core storage is transferred serially to the 1627 under direct program control, where it is translated into 1627 actuating signals. These signals are then converted into drawing movements by the 1627.

The actual recording is produced by incremental movement of the pen on the paper surface (y-axis) and/or the movement of the paper under the pen (x-axis). The pen is mounted in a carriage that travels horizontally across the paper. The vertical plotting motion is achieved by rotation of the pin feed drum, which also acts as a platen (Figure 36).

The drum and the pen carriage are bidirectional; that is, the paper moves up or down, and the pen moves left or right. Control is also provided to raise or lower the pen from or to the paper surface. The pen remains in the raised or lowered position until directed to change to the opposite status.

The drum and the pen-carriage movements and the pen status are controlled by bits transferred to the 1627. Each output word is decoded into a directional signal that causes a 1/100 inch incremental movement of the pen carriage (Figure 37) and/or paper, or a raise-pen or lower-pen movement. The motion or action resulting from each word in the output record is shown in Figure 38.

		Model 1	Model 2
Speed	X, Y Increments Pen Status Change	18,000 Steps/Min 600 Operations/Min	12,000 Steps/Min 600 Operations/Min
Increment Size		1/100 Inch	1/100 Inch
Chart Paper	Width Plotting Width Length Sprocket Hole Dimensions	12 Inches 11 Inches 120 Feet .130 Inch Dia on 3/8 Inch Centers	31 Inches 29 1/2 Inches 120 Feet .188 Inch Dia on 1 Inch Centers

15667

Figure 35. 1627 Operating Characteristics

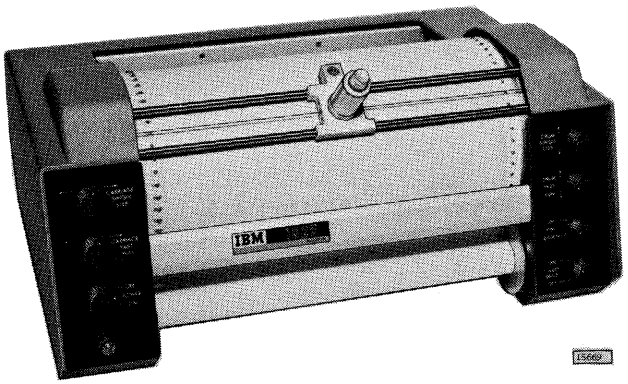


Figure 34. IBM 1627 Plotter

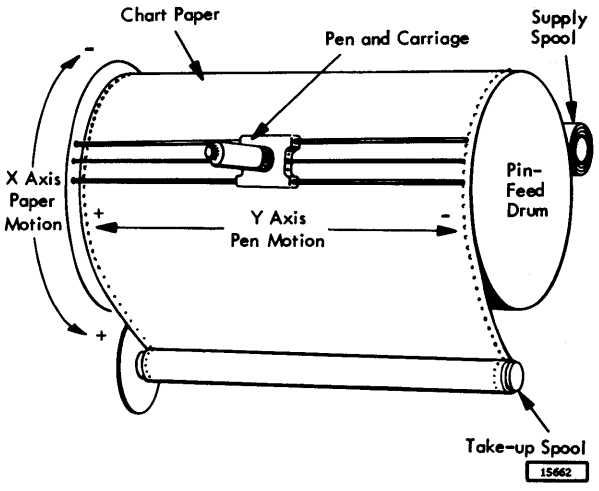


Figure 36. Plotter Paper and Pen Movement

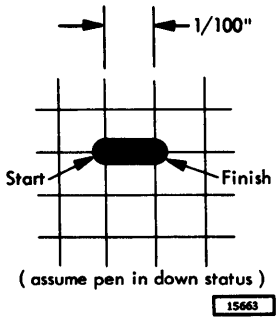


Figure 37. Result of One Horizontal (y-axis) Movement

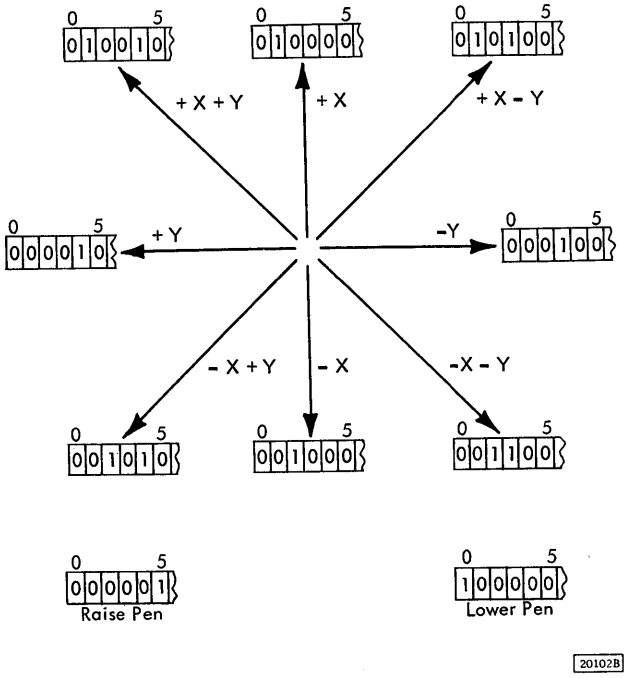


Figure 38. Plotter Command Codes

The time required for execution of raise-pen and lower-pen commands is 100 ms. The time to plot a point is approximately 3.3 ms model 1, or 5 ms model 2.

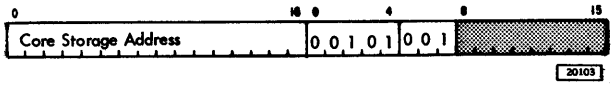
PROGRAMMING

The 1627 Plotter operates under direct program control of the 1130 and responds on interrupt level 3.

I/O Control Command (IOCC)

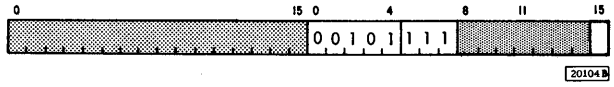
The 1627 is addressed by the five-bit device code 00101.

Write (001)



This command causes bit positions 0 through 5 of the word in the core storage location specified by the address to be sent to the 1627 to control the movement of the pen or drum. (See Figure 38.)

Sense Device (111)



This command causes the 1627 device status word (Figure 39) to be placed into the accumulator. Modifier bit 15 on specifies reset for the plotter response.

DSW Indicators

Plotter Response (Interrupt): This is the only interrupt associated with the 1627. This interrupt occurs when the 1627 has completed the action specified by the last write command. The 1627 is on interrupt level 3.

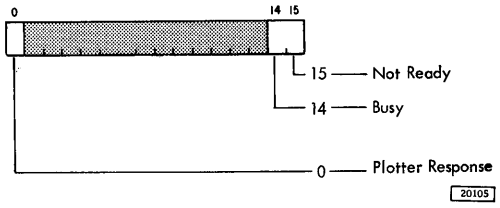


Figure 39. 1627 Device Status Word

Not Ready: This indicates the 1627 is not ready to execute commands.

Busy: This indicates that the 1627 is in a busy status and cannot accept a character. After the first write command, the program should wait for succeeding plotter interrupts to issue additional write commands. If a write command is given while busy is on, loss of information will probably occur. No indication is given of this loss.

The IBM 1231 Optical Mark Page Reader (OMPR) represents a breakthrough in source recording and data entry. The OMPR provides a facility for recording the data at its source, in a form that can be read directly into the IBM 1130 Computing System.

The 1231 (Figure 40) reads positional marks made by an ordinary lead pencil on paper documents. The positional marks are read directly into the 1130 core storage.

Documents are read at a maximum rate of 2,000 sheets per hour.

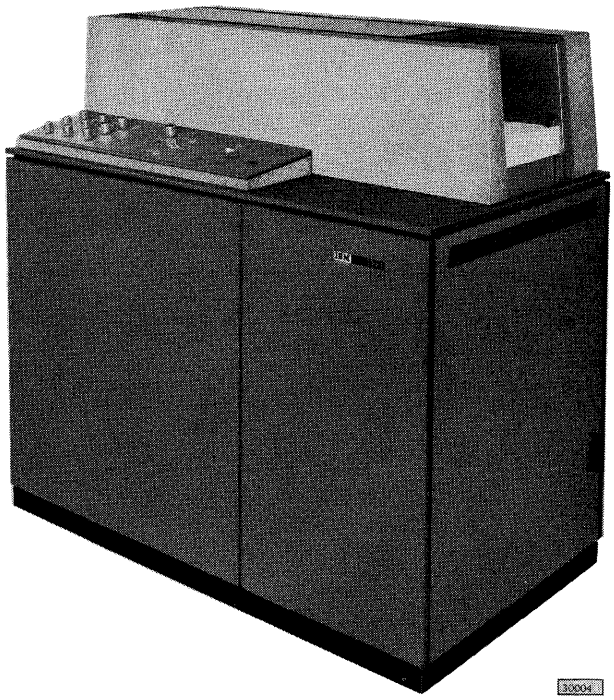


Figure 40. IBM 1231 Optical Mark Page Reader

DATA SHEET

The document used as input to the Optical Mark Page Reader is an 8-1/2" x 11" sheet of paper called a "data sheet." The data sheet contains a maximum of 1,000 mark positions. The mark positions are arranged in as many as 50 rows, each row containing a maximum of 20 mark positions.

Each row is divided into two groups of ten mark positions each. The ten mark positions are called "words." Each word is divided into two groups of five mark positions called "segments." Consequently, each data sheet can have a maximum of 50 rows, 100 words, and 200 segments. A data sheet normally contains five rows per inch but may have less.

Timing marks are printed along the right-hand edge of each data sheet. These marks are used to synchronize the motion of the document with the sensing unit of the reader. Each word on the data sheet has an associated timing mark. For forms design information see the System Reference Library publication IBM 1231, 1232 Optical Mark Page Readers (Form A21-9012).

Data Sheet Terminology

Timing Mark: A rectangular mark preprinted on the data sheet in non-reflective ink. The timing mark is used to synchronize the motion of the document with the sensing unit of the 1231. Timing marks are located on the right-hand side of the data sheet.

Mark Positions: Areas printed in reflective ink that designate where marks are to be placed. A non-reflective mark in this area is read as a word or bit.

Word: Ten mark positions of a row. Words on the left half of the data sheet are odd words; words on the right half of the data sheet are even words.

Segments: Mark positions 0 through 4 and 10 through 14, or 5 through 9 and 15 through 19, of any word.

Non-Reflective Ink: A type of ink that is sensed by the 1231. Usually, timing marks are the only non-reflective printing on the data sheet. The recommended non-reflective ink is black.

Reflective Ink: A type of ink not sensed by the 1231. Reflective inks are used for headings, data sheet instructions, mark position outlines and any other data that is not to be read.

Marking the Data Sheet

Marks that are to be read by the IBM 1231 must be dark enough for positive machine reading, yet erase easily and completely. For these reasons, a number 2 pencil is recommended.

Marks made with a number 1 pencil, or an IBM ELECTROGRAPHIC (R) pencil, are difficult to erase. Even after an erasure is made, a residue remains that could be read as a mark by the machine.

Erasures should always be made carefully and completely. Any incomplete erasure could be read as a mark.

When response positions are marked, the mark should be made the full length of the mark position, and should fill at least two-thirds of the space between the top and bottom of the guide lines. A mark that extends no more than 1/16" past the ends of the response position is acceptable in all but the last even-word position (next to the timing marks). In this position, a mark must not extend beyond the right end of the guide lines or it could be read as a timing mark. This would result in erroneous reading of the rest of the data sheet.

FUNCTIONAL DESCRIPTION

The 1231 uses sonic delay lines for storing controls and data. Controls are marked on the regular data sheet and entered into delay line storage during the program load cycle. This data sheet is referred to as a program control sheet. The program control sheet is automatically placed in the select (upper) stacker during the load cycle.

As data sheets are read, data is stored in the delay lines according to instructions from the program control sheet. Each word to be stored on the delay line must be programmed by the program control sheet.

When a data sheet passes under the photoelectric read head, each word is tested for conditions, such as no-mark, multi-mark, or other-than-one. Switches on the 1231 control panel in conjunction with the program control sheet control the test of these conditions. Any word that does not pass the requirements of the switch settings causes the data sheet to be routed to the select stacker.

Document Path

The data sheet begins its movement through the optical mark page reader when it is fed from the hopper by CPU program control. The document than passes under a read head and is next transported through the transport area, past a selection station, and on into one of the two gravity stackers.

Message Format

Each word transferred from the 1231 to the 1130 reads into a single position of core storage. Words are transferred one segment at a time to the A buffer and the B buffer in the attachment; all odd segments (A buffer) enter positions 0-4 and 14, and all even segments (B buffer) enter positions 5-9 and 15 (Figure 41). If the 1231 is programmed for only one segment, all segments enter positions 0-4 and 14. Words with marks in positions 0, 1, 2, 3 or 4 transfer to core storage as an odd segment and marks in positions 5, 6, 7, 8, or 9 transfer to the 1130 as an even segment. Combinations of the bits make up a valid character which must be translated by the 1130 stored program. Any or all of the marking positions on the data sheet may contain marks.

Data is read by the 1231 from left to right, top to bottom, a row at a time. Information from a data sheet is stored in the following sequence:

1. Segment one of the first word programmed to read.
2. Segment 2 of the first word.
3. Segment 1 of the second word programmed to read.
4. Segment 2 of the second word.

If only one segment of any word is programmed to read, then each segment goes into a separate core storage word.

Mark Recognition and Discrimination

During the reading of data sheets, the Optical Mark Page Readers categorize marks according to their degree of light reflectance (Figure 42). A mark falls into one of the following categories:

1. Good
2. Poor
3. Uncertain

A good mark is recognized as a positive indication of a mark; a poor mark (or good erasure) is not recognized as a mark, and an uncertain mark (light mark or poor erasure) is one whose light reflectance level comes somewhere in between a good mark and a poor mark but cannot be positively identified as either. The reading or rejection of uncertainties can be customer-controlled.

Three read-mode switches, each associated with a set of field-checking switches, allow operator control of mark discrimination on a field-by-field basis. Documents containing uncertainties can be selected for a visual check if desired.

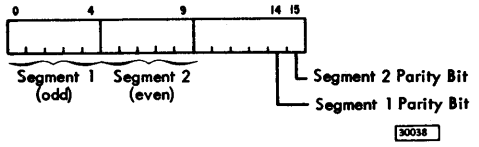


Figure 41. 1231 Data Format

Each of the three read-mode switches has four settings: SING RESP (single response), MULT RESP (multiple response), SING RESP SEL UNC (single response select uncertainties), MULT RESP SEL UNC (multiple response select uncertainties).

The setting of each read-mode switch affects mark discrimination as follows:

1. SING RESP:
   a. Marks in area A are accepted.
   b. Marks in area B that are not accompanied by a mark in area A of the same word or segment are accepted.*
   c. Marks in area B that are accompanied by a mark or marks in area A of the same word or segment are not accepted.
2. SING RESP SEL UNC:
   a. Marks in area A are accepted.
   b. Marks in area B that are not accompanied by a mark in area A of the same word or segment cause the data sheet to be selected.
   c. Marks in area B that are accompanied by a mark in area A of the same word or segment are not accepted as marks.
3. MULT RESP:
   a. Marks in area A are accepted.
   b. Marks in area B are accepted.
4. MULT RESP SEL UNC:
   a. Marks in area A are accepted.
   b. Marks in area B cause the document to be selected.

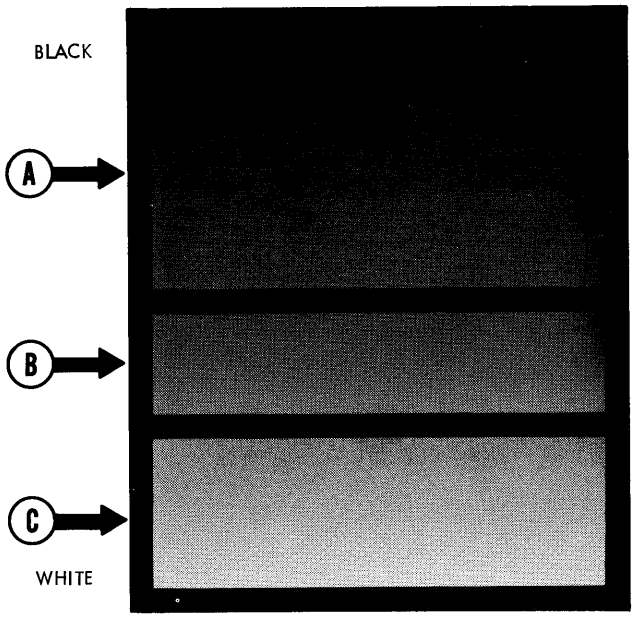
Whenever a data sheet is selected by the 1231, storage is cleared and data from that data sheet is prevented from being transferred to the computer.

Data Flow

Before the 1231 can act as an input device to a data processing system, the controls for the internal functions must be loaded and switches must be set to establish the conditions required for the particular run.

Two storage devices (sonic delay lines) are used to store and control the data as it is read from the data sheets. One of these storage devices, the "master" line, is used to store all the controls from the program control sheet. If the 1231 is equipped with the master mark special feature then master-mark data and controls associated with master-mark data are also stored.

*Number of mark positions included in any one mark discrimination test is determined by the setting SEGMENT or WORD of the check-length switch.



A: Good Mark
B: Light Mark or Poor Erasure
C: Poor Mark or Good Erasure (Marks in this area are not read)

27108

Figure 42. Mark Reflectance Relationship

The other storage line, the data line, is used to store information from the data sheet. As the data sheet is read, the two storage lines work concurrently and in synchronism. The master line, which contains the program instructions, determines which information from the data sheet is to be retained.

The following sequence is used for entering data into a fully equipped 1231 and for making this information available to the processing system (Figure 43).

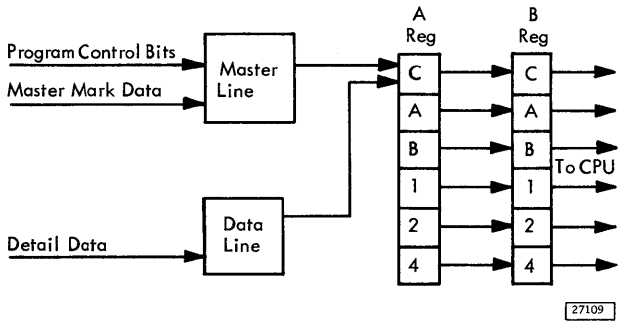


Figure 43. IBM 1231 Data Flow

1. Line mark and word mark bits are generated by internal circuitry to establish the starting point of the data on the delay lines. These bits go into the data delay line.
2. Program control bits are loaded into the 1231 from the program control sheet and go into the master line.
3. Master-mark information (if master-mark special feature is installed and being used) is transferred the same as detail data. It is up to the system to store this information and merge it with each succeeding data sheet until new master data is transferred.
4. Detail data reads into positions 12 through 111 of the data line.
5. The first word of data (master or detail) is sent to the 1231 attachment buffer in the 1131. When this buffer is loaded it causes an interrupt to be generated. The 1131 program must then issue a read command to transfer this word into core storage. This process is repeated until all the data is transferred.

Field Checking

In field checking, each word programmed to be read is checked for mark conditions which may indicate invalid data. Three switch-controllable mark conditions, each of which will cause the document to be selected, can be checked. These conditions are: multi-marks, no marks, or other-than-one. The switch also has an off position.

Data sheets are usually designed with "fields" of similar data in vertically consecutive words on the left or right sides of the data sheet.

A field checking field differs from a data sheet field primarily in functional grouping. A data sheet field groups similar information for ease of marking and reference. A field checking field may contain part of, or several, data sheet fields. The primary requirement of field checking is that all mark data within a field's area of coverage be checked for the same conditions (multi-mark, no mark, etc.).

The field checking fields on the data sheet are defined by special codes (start-of-checking codes) which are entered into 1231 storage from the program control sheet. A field checking field can be from 1 to 100 words in length.

Three start-of-checking control codes allow any specific area of the data sheet to be checked according to one of three groups of field checking switches. The three groups of field checking switches are labeled, field I, field II, and field III.

The checking of a field checking field by a particular group of switches begins on the word in which the field checking control code is recognized. On the program control sheet, a mark in position 6 designates the start of data checking according to conditions set up on field I switches; a mark in position 7 designates the start of data checking according to conditions set up in field II switches; marks in positions 6 and 7 designate the start of data checking according to field III switches.

Three switches are assigned to each field: (1) a read mode switch, which determines how uncertain marks are handled, (2) a check length switch, which determines whether information is to be checked on a word or segment basis, and (3) a select condition switch, which determines the conditions for which a data sheet will be selected.

Because the data sheet is read from left to right, top to bottom, row by row, field checking becomes an important factor when a new data sheet is to be designed. If, for instance, the data sheet is to be used for a yes or no survey, the yes and no mark positions should be within one segment or word in order to allow checking for both, either, or neither answer.

The field checking feature can be summarized as follows:

1. Each of the three field-checking switches can be set to one of the select conditions, or to the off position.
2. Unless programmed for another field, checking always returns to field I at the start of a new data sheet.
3. Field checking by a given set of field checking switches begins in the word programmed and continues (in all words programmed to be read) until a new field checking command has been recognized.
4. A field can begin or end on either the left or the right word.
5. A field can be from 1 to 100 words in length.
6. Three conditions can be checked: other-than-one (multi-mark and/or blank detection), no-mark (blank detection only), and multi-mark (multi-mark detection only).

When the data sheet is selected because of a field checking condition, or because of an uncertain reading, the 1231 causes the delay line to be reset. Some data may have been transferred. This data is considered to be invalid.

Alphabetic Coding

An alphabetic coding capability is necessary and desirable in many applications. Three schemes of alphabetic coding are illustrated in Figure 44.

Scheme 1: To code an alphabetic character, a mark must appear in the appropriate marking position of both the odd (left hand) and even (right hand) words of the same horizontal row. For example; to indicate the letter K, one mark must be made in the marking position immediately below the caption "J" through "R" in the odd word, and one mark must be made in the marking position immediately below the caption "BKS" in the even word of the same horizontal row. The odd word in this scheme represents the zone portion of the character.

Scheme 2: Each letter of the alphabet can be preprinted on the data sheet in reflective ink. The letters and/or numbers may be printed above, on, or below the mark positions. In this approach, the identity of each character is determined by its position in the matrix, which is programmable.

The entering of the marks is simple; however, considerable space is required on the data sheet to represent all the alphabetic and numberic characters.

Scheme 3: Each letter of the alphabet, or digits 0 through 9, can be represented by using only one word. An alphabetic character must be represented by a mark in each segment of the word selected for this purpose. (Z is an exception.) To indicate a K, marks in the 2-position of the left-hand segment and in the 5-position of the right-hand segment are required.

PROGRAMMING

Programming for the IBM 1231 Optical Mark Page Reader depends upon two sources of control: controls stored within the 1231 and controls received from the 1130. Controls stored within the reader are entered into storage from a program control sheet.

Program Control Sheet

A program control sheet is a data sheet with certain operational controls marked in the data areas. Each word from the data sheet consists of ten positions. Each word in the delay line storage consists of 16 positions: ten for the positions on the data sheet, and six for storing operational and internal controls generated by circuitry of the 1231.

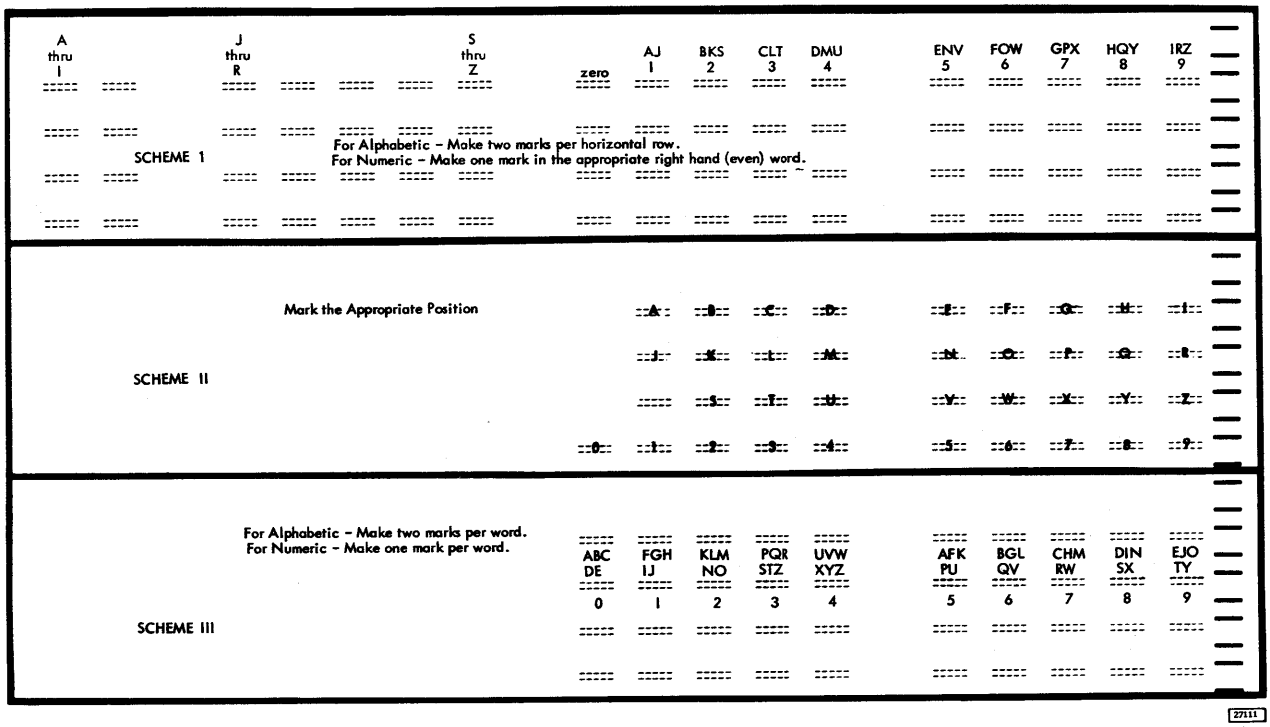


Figure 44. Alphabetic Coding Schemes

Every word that is to be retained for transferral to the 1130 must have an operational control marked in that word on the program control sheet. When operational control information is entered into storage, the controls go into some of the six control positions associated with each word.

During the program load cycle, the mark positions used as control positions are:

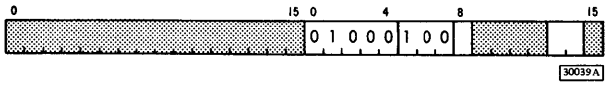
1. A mark in position 8. This designates that a word is to be stored. This word is available later for readout to the 1130 system. There must be two timing marks for each row on the detail data sheet programmed to read by the program control sheet.
2. A mark in position 0 and a mark in position 8. This stores data from segment 1 only (bits 0-4 or 10-14 of the data sheet).
3. A mark in position 5 and a mark in position 8. This stores data from segment 2 only. (Bits 5-9 or 15-19 of the data sheet).
4. A mark in position 6. This indicates the start of field checking according to the settings of field I switches.
5. A mark in position 7. This indicates the start of field checking according to the settings of field II switches.
6. A mark in position 6 and a mark in position 7. This indicates the start of field checking according to the settings of field III switches.

System Programming

Three I/O commands are used with the 1231: control (100), read (010), and sense device (111). In addition the control command uses three modifier bits to expand the number of commands.

The 1231 is addressed by the five-digit device code (01000=area code 8).

Control (100)



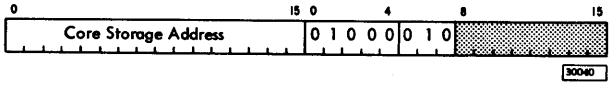
This control command uses three modifier bits:

Read Start (bit 13): Causes the document to move through the read station. Data is collected and placed on the delay line controlled by the control sheet and switch settings. As soon as the first word programmed for output is placed on the delay line, it is made available to the processor.

I/O Disconnect (bit 14): Terminates the read operation from the document and clears the delay line storage by signaling the 1231 that no more data is desired. This command should be given to prevent a read (overrun) error on the next document if all data from the previous document is not cleared from the delay line storage.

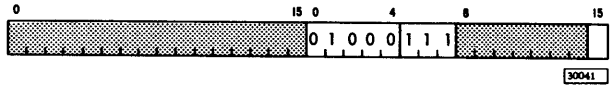
Select Stacker (bit 8): Causes the document just read to enter the select stacker. This command can be given within 40 ms after the last word is placed on the delay line. Indicator bit 5 in the device status word is on if it is permissible to select the document. This bit should be tested prior to issuing the select command. If the bit is off and a command is issued the command will be ignored.

Read (010)



This command causes the next word in the 1231 attachment buffer to be loaded into the core storage location specified by the address.

Sense Device (111)



The sense device command causes the 1231 DSW (Figure 45) to be placed in the accumulator. Modifier bit 15 resets the responses.

DSW Indicators

Read Request (Interrupt): Signals that a word (one or two segments) has been loaded in the 1231 attachment buffer and can be accepted by the CPU. An XIO sense DSW command with bit 15 turns off the read request and loads the 1231 DSW into the accumulator. An XIO read command is needed to turn off the read busy indicator and transfer the word to core storage.

Operation Complete (Interrupt): Occurs after the last word has been accepted by the CPU. End of transmission from the 1231 causes this interrupt. Timing mark error and read error can also cause this interrupt.

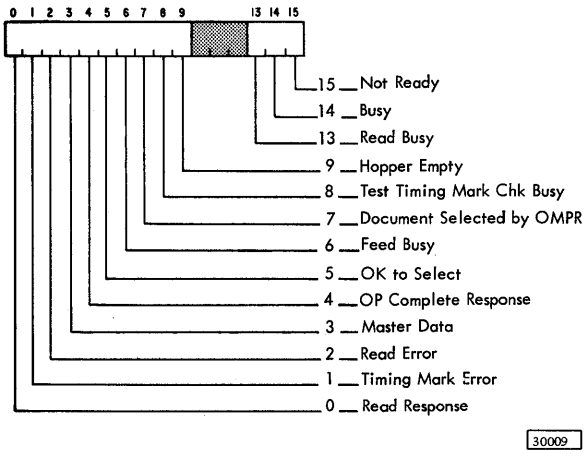


Figure 45. 1231 Device Status Word

It is possible to read the last character of a document and receive the end of transmission signal from the 1231 before the timing mark check is made if the timing mark switch is set to yes. This would turn on the operation complete interrupt, which would then be serviced by the processor. If the timing mark switch were set to YES, a timing mark error could occur after the operation complete routine. Were this to occur, the timing mark error indicator would be turned on. This indicator would remain on until the 1231 was placed in the ready state. The operator would be aware of this error either by program analysis or by visual inspection of the 1231 control panel.

Timing Mark Error (Interrupt): Indicates there is a timing mark error. Timing mark error turns on the operation complete interrupt. The error is dependent upon the settings of two switches: timing mark check switch and control timing mark switch. The timing mark check switch is an 11-position rotary switch. It has an off position and ten positions labeled 0 through 9. In the off position no checking is performed. If checking is desired, the switch is placed on the digit corresponding to the units value of the number of timing marks on the documents. The control timing mark switch has two positions labeled YES and NO. The no position is used for documents having the normal number of timing marks (100 or fewer); the yes position is used for documents having 106 timing marks.

Read Error (Interrupt): On if an even count (parity), overrun, or no bits error is detected. The error turns on operation complete interrupt to alert the processor and prevent further transfer of data from the attachment.

Not Ready: If this indicator is off the 1231 is ready to accept instructions from the CPU program. The following conditions are necessary for the 1231 to remain in a ready condition:

1. Power on.
2. Off line/on line switch set to ON LINE.
3. Control sheet loaded.
4. Hopper loaded.
5. No read or feed errors.
6. Start key is depressed after the program load light on the 1231 goes off after loading the control sheet.

Busy: This indicator comes on after the read start command has been issued and remains on until an operation complete interrupt is received or an XIO disconnect command is issued. This indicator being on indicates that a document is being read.

Read Busy: This indicates that the 1231 attachment buffer is full. It is turned off when the XIO read command for the 1231 is given.

Feed Busy: This indicator comes on when the XIO control command with modifier bit 13 is executed. It remains on until the first 1231 interrupt is turned off. Another start feed command should not be given while this indicator is on.

Read Response: This indicator comes on each time the attachment buffer is loaded. It is reset with the XIO sense device command with bit 15 on.

Operation Complete: This indicator is turned on by the end of transmission and signifies that the last word has been read by the CPU. It is also turned on by the timing mark error or read character error. It is turned off by an XIO sense device with bit 15 on.

Okay to Select: This indicator comes on when a document read is initiated and remains on for 40 ms after the document has been read.

Test Timing Mark Check: This indicator comes on just before the first character interrupt from the 1231 attachment and remains on for 90 ms after the last word has been placed on the delay line. If the control timing mark switch is set to YES, the timing mark check is not made until the end of the 90 ms period. If this indicator is to be tested, data processing should not take place until the indicator goes off.

Master Mark: This indicates that data to be transferred is master data. A master data subroutine should place master

data in a reserved area. The indicator does not come on if the master-mark switch is off.

Read Error: This indicates that a parity error (even count) occurred, an overrun condition occurred, or that no bits were entered onto one of the delay lines with either a data sheet or a control sheet. The not ready and the operation complete interrupt is turned on. The delay line storage is cleared so that it is not necessary to give an I/O disconnect.

Document Selected: This is caused by either a mark count reject or data uncertainty, according to the setting of the field checking switches and the program control sheet. It is turned off by an XIO sense device with bit 15 on. Document feeding is not inhibited. When this indicator is turned on the delay line storage is cleared of data and the transfer to the processor is terminated. The operator must be flagged by the program that this has happened. If the next feed command has been issued it will be necessary to refeed two documents. If the document selected indicator is turned on and serviced before the next feed command, only the top sheet in the select stacker should be processed. Data should not be processed if this indicator is on.

Hopper Empty: This indicates that the 1231 hopper is empty and turns on not ready.

1231 Usage Meter

This meter runs when both of the following conditions are present:

1. The unit is selected for operation by program control.
2. The 1131 usage meter is running.
3. The meter continues to run simultaneously with the 1131 meter until either the 1231 hopper becomes empty or it is stopped by the operator.

The IBM 2250 Display Unit, Model 4 (Figure 46) is a cathode-ray tube display unit that attaches to the IBM 1130 Computing System. The 2250 Display Unit operates under the control of a display order program and input/output (I/O) control commands. The program and commands are sent from the 1131 via the 1131 storage access channel (SAC) or 1133 storage access channel II (SAC II).

The 1130 system and 2250 model 4 operate asynchronously. The display program orders can be sent to the 2250 at a rate up to 40 frames per second (25 millisecond frame time); however, the 2250 operation can be delayed while the 1130 is servicing a device with a higher cycle-steal priority.

This section provides a brief description of the IBM 2250 Display Unit, Model 4. For a more complete description of the 2250 model 4. refer to the Systems Reference Library publication IBM 1130 Computing System Component Description -- IBM 2250 Display Unit, Model 4, Form A27-2723.

FUNCTIONAL DESCRIPTION

As soon as a 2250 operation is started by an I/O control command, the 2250 addresses CPU storage to execute the display program -- stealing core storage cycles from the CPU. Core storage cycle demands by the 2250, therefore, always have higher priority than those of the CPU program. Units that operate synchronously with the CPU are assigned higher cycle-steal priority than the 2250, eliminating 2250 interference with synchronous operations.

The 2250 can generate images of vectors (straight lines), points, and characters on the 21-inch cathode-ray tube. (The usable display area is 144 square inches -- 12 inches by 12 inches.) A visible display is produced when the electron beam in the cathode-ray tube strikes the phosphor-coated cathode-ray tube screen. The screen area struck by the beam glows briefly. Normally, the glow fades within a fraction of a second -- too rapid for human eye perception or recognition. For this reason, the display is continuously regenerated at a discernible rate.

As soon as regeneration is started by an 1130 I/O control command, the 2250 channel interface section -- where storage addressing is performed -- continuously fetches orders and data from the display program in storage. Orders are decoded in this section. Deflection information is transferred to the 2250 display section, where it is used to draw the appropriate display. Regeneration is accomplished by continuously repeating the display program. Orders and data in the display program can be modified during regeneration -- as directed by the CPU program or by the display program itself -- to update or change the display.

The 2250 display section, furthermore, performs various nondisplay services by providing the interface between the user and the problem program with three devices: the programmed function keyboard, the alphameric keyboard, and the fiber-optics light pen.

The programmed function keyboard provides communication between the user and a CPU program. The keyboard consists of keys, indicators, and sensing switches for use with replaceable descriptive overlays. The function of each key and indicator is defined by the CPU program. Punches in the top edge of each overlay identify the overlay to the CPU program. To identify the key and indicator functions to the operator, the key or indicator labels (or both) can be placed on the overlay. Each key can be used by the program to initiate a subroutine associated with the respective overlay, thereby performing the indicated function. For example, depression of a key might result in the enlargement, reduction, or deletion of the display image.

The alphameric keyboard makes it possible for character displays to be

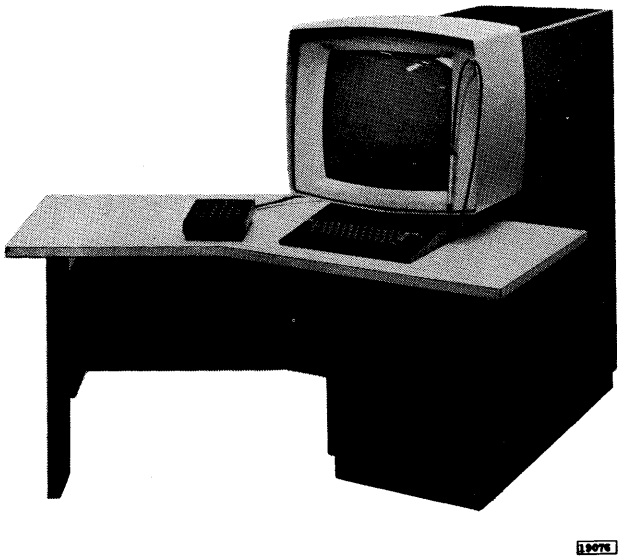


Figure 46. IBM 2250 Display Unit, Model 4

created, edited, or changed by the user. With the typewriter-like keyboard, alphameric messages can be entered into the display program for displaying and editing. The alphameric keyboard key codes can be interpreted by the CPU program and used for control purposes in a manner similar to operations with the programmed function keyboard.

The light pen provides the means by which the display program and the CPU program can identify the storage address of the order that initiated the display of a vector, point, or character. This information can be used for operations determined by the display program, by the alphameric keyboard, or by the programmed function keyboard. The user can identify a displayed image simply by pointing the light pen at the image or by pressing the tip switch (the point at the end of the light pen) against the image. The method of identification is determined by the display program.

DISPLAYS

Information positioning on the 2250 display area is controlled by a display program in the 1131 core storage. This program is sent to the 2250 (by a 16-bit word) via the 1131 storage access channel or 1133 storage access channel II. Orders in this program specify electron beam deflection to horizontal (X) and vertical (Y) coordinates on a square reference grid. This grid is composed of the 1,024 possible electron-beam-deflection end points.

Information can be displayed in the 2250 in either the graphic mode or character mode.

Graphic Mode

Either vector or point operations can be performed by the 2250 in graphic mode. If no specific graphic mode has been set previously by an order from the display program, the vector mode is set automatically. In graphic mode, the 2250 can receive (from the display program) either electron-beam positioning orders or an order to establish a different mode of operation, such as to set point mode from vector mode or to enter character mode from graphic mode.

Character Mode

The set of characters that can be displayed by the 2250 in character mode is defined by the programmer. This character set resides in 1130 storage as a subroutine of the display program. The character set can comprise any number of characters in any font. These characters can be modified at any time during execution of the display program. Characters in this set can be displayed in either of two sizes -- basic or large, as determined by the character-mode order.

In character mode, the current X, Y position of the beam on the 1,024-by-1,024 position display area becomes the center of a basic- or large-size character area. This area is maintained throughout one character mode operation. The program places the beam at a starting position on the display area (using a blanked point or vector) before a character display operation is started.

CHANNEL INTERFACE SECTION

The 2250 channel interface section interfaces the storage access channel (SAC) and the 2250 display section. It decodes and executes orders and commands, addresses CPU storage, and handles data transferred to or from CPU storage. Information transfer across the SAC/2250 interface is by a 16-bit word.

An address register in the 2250 channel interface section specifies (to CPU storage) the location at which information will be stored or from which it will be retrieved for 2250 operations. This address register is initially loaded by an initiate write (start regeneration) command from the CPU program. It can then be stepped automatically by the 2250, altered by the display program, or reloaded by the CPU program. Thus, display regeneration can be performed without CPU intervention.

The display program consists of display orders, associated data for image generation, and control orders for various nondisplay functions. Undefined order codes received by the 2250 are treated as no-operation orders or are interpreted as data if in the appropriate format.

The CPU program initiates 2250 operation by issuing an execute input/output (XIO) instruction. The I/O control command at the effective storage address specified by the XIO is then sent to the 2250. If the I/O control command is initiate write (start regeneration), the 2250 fetches display program information from main storage starting at the address specified by the I/O control command.

Display program information consists of orders and data. Orders either initiate a 2250 operation or establish a mode. Order-initiated operations include point and vector plotting, branching, and CPU interrupt generation.

Data is defined as information that does not contain an operation code. Character-stroke words are the only data received by the 2250. Even though a

character-stroke word may contain one or more control bits, these bits are used directly to perform an operation.

PROGRAMMING

The 1131 Central Processing Unit (CPU) uses input/output (I/O) control commands to control 2250 operations.

Input/Output Control Commands

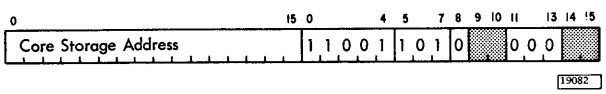
The 2250 is selected by the five-bit device code 11001. The three-bit function code specifies primary I/O functions. The modifier portion of the command provides additional information for the device and function specified. Command modifier bits 11, 12, and 13 must be 0's. Unassigned modifier bits are not decoded. Unassigned functions codes are treated as no-operation commands by the 2250. The execution time of each command is equal to the XIO instruction time plus one core storage cycle time for each cycle steal required for data transfer. When an XIO instruction is executed, the odd word of the I/O control command is sent to the 2250 via the storage access channel before the even word.

The I/O control commands associated with the 2250 are initiate write, initiate read, control, sense interrupt, and sense device.

Initiate Write

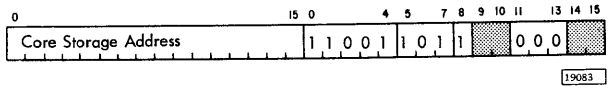
The two modifiers of the initiate write command are start regeneration (bit 8 = 0) and set programmed function indicators (bit 8 = 1). Both modifiers cause the corresponding even I/O control command word to be loaded into the 2250 address register. Words are then accessed from CPU storage by cycle stealing, starting at this address. An initiate write command can be executed only when the 2250 is not busy (not regenerating), and is treated as a no-operation command when the 2250 is busy. A control (reset display) command can be used to stop regeneration.

Start Regeneration Command: Starts execution of the display program at the address specified in the even command word.



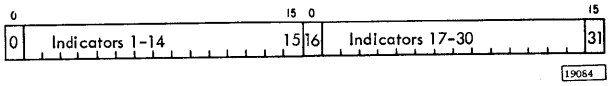
Regeneration continues under control of orders in the display program until terminated by a control (reset display) command or by a 2250 interrupt. The busy bit in the device status word is set during regeneration. (The device status word contains one bit of information for each indicator within the device.) The start regeneration command also clears the interrupt status indicator (device status word bits 0-2); if the keyboard interrupt bit is set, the command unlocks the 2250 keyboards, resets the data available bit, and clears initiate read command response word 4 and 5.

Set Programmed Function Indicators Command: Is used to load the programmed function keyboard indicators with the contents of two consecutive words in CPU storage.



The first of these two words is specified by the address word of this command. Two cycle-steal operations are performed.

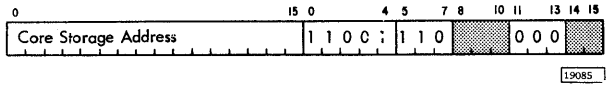
Each bit in the two indicator words corresponds to one programmed function keyboard indicator.



All 1-bits cause their associated indicators to light, and all 0-bits cause their associated indicators to be turned off. No interrupts are generated.

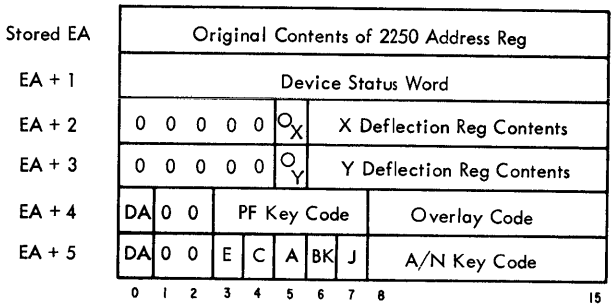
All programmed function indicators are turned off by a power-on reset (generated when 1130 power is turned on) and by a manual reset (generated when the 1131 reset pushbutton is pressed).

Initiate Read



The initiate read command causes six words of 2250 status information to be placed, by cycle stealing, into CPU storage starting at the address specified in the command. The original contents of the 2250 address register are saved (as the first word of status information) before the command address word is loaded but are not restored after execution of the command.

An initiate read command is normally issued immediately after a sense interrupt command in response to a 2250 interrupt; however, it can be executed any time the 2250 is not busy. Interrupts are not generated by the initiate read command, and the 2250 interrupt request is reset (if set). The six words of status information read by this command are as follows:



Legend: O_X = X Overflow
O_Y = Y Overflow
DA = Data Available
E = End Key
C = Cancel Key
A = Advance Key
BK = Backspace Key
J = Jump Key
PF = Programmed Function Keyboard
A/N=Alphameric Keyboard

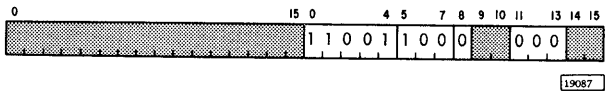
19086

These words reflect the status of the address register, device status word, X and Y deflection registers, programmed function keyboard, and alphameric keyboard at the time of the preceding interrupt. If a keyboard is not attached to the 2250 or does not have data available, the appropriate data available bit (bit 0) will be a 0. The device status word contents are defined in the sense device command description. The address register contents in the first word of this response, to be meaningful, may require modification as specified by address displacement bits 14 and 15 in the device status word.

Control

During control command execution, the 2250 address register is not loaded by an address from the I/O control command. Cycle steals are not used, and interrupts are not generated. The two control modifiers are no-operation (bit 8 = 0) and reset display (bit 8 = 1).

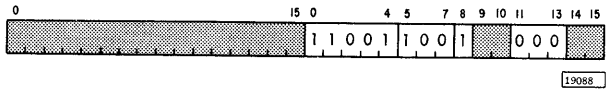
No-Operation Command: Is ignored by the 2250. It is reserved as a no-operation, and will not be assigned a function in the future.



19087

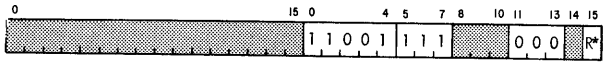
Reset Display Command: Stops regeneration immediately and generates a unit reset in the 2250, causing all registers, controls, and keyboards to be reset. Zero is the reset state of all registers except the X and Y deflection registers, which are reset to 512 each (the center of the reference grid). The display mode is reset to graphic mode (vector), light pen control is reset to the disable detects and defer interrupts condition, all pending interrupts are cleared, and the 2250 is made not busy.

In addition, the bit configuration in the odd word of the control (reset display) command (at EA + 1) is imaged twice in the programmed function indicators, once in indicators 0-15 and again in indicators 16-31. Each 1-bit lights two indicators, and each 0-bit clears two.



19088

Sense Device



*Reset (R): If set to 1, causes interrupt request to be reset.

19091

This command causes the 2250 to send a device status word (Figure 47) to the 1131, where it is loaded into the accumulator. Cycle steals are not used and interrupts are not generated. If the 2250 is regenerating (is busy), only bit 8 of the device status word will be set. When the 2250 is not busy, the device status word contents describe the control status of the 2250, as follows:

DSW Indicators

Order-Controlled Interrupt: Is generated when the 2250 is executing either the unconditional or conditional interrupt variation of the long branch/interrupt order. The conditional interrupt variation can cause an interrupt only when the light

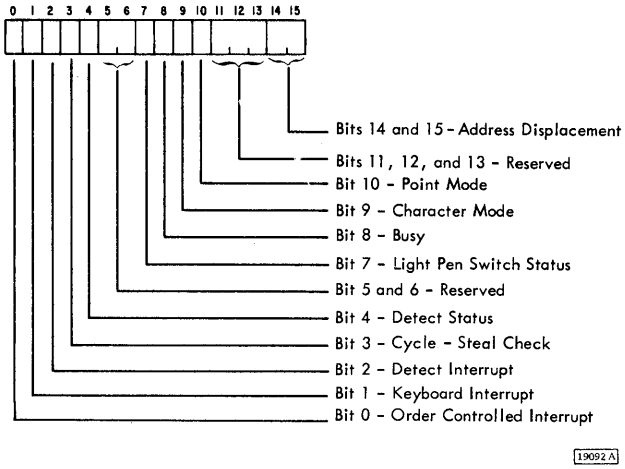


Figure 47. 2250 Device Status Word

pen detect and/or light pen switch status bits are tested successfully by the order. A 1 in device status word bit 0 indicates the occurrence of an order-controlled interrupt.

Following execution of an initiate read command, the address in the first word of status information points to the second word of the long branch/interrupt order, which may contain an address or other interrupt identification data. Bits 4 and 7 of the device status word indicate the light pen detect and light pen switch status at the time of interrupt. However, 4 is reset after it is tested successfully.

Keyboard Interrupt: Is set when a key has been pressed either on the alphameric keyboard or on the programmed function keyboard, and the next start timer order is decoded. An initiate read command reads the appropriate keyboard (response word 4 or 5). Both keyboards are locked, and light pen detects are inhibited at the time of interrupt and remain in this condition until an initiate write (start regeneration) command is executed. A 1 in device status word bit 1 indicates the occurrence of a keyboard interrupt.

If both keyboards are simultaneously activated, the programmed function keyboard is given priority by the 2250 and causes the interrupt; in this case, the alphameric keyboard is locked out. Bits 4 and 7 of the device status word indicate the light pen detect and light pen switch status at the time of interrupt.

Detect Interrupt: Is generated when the 2250 is enabled for light pen interrupts and when a detect has occurred. This interrupt is indicated by a 1 in device status word bit 2.

If the 2250 is enabled for light pen detects when a detect occurs, the address in the first initiate read response word depends on the type of data detected. Bits 9 and 10 of the device status word identify the display mode as character, vector, or point. Bits 14 and 15 of the device status word specify a displacement. This displacement should be subtracted from the initiate read response word 0 contents to obtain the address of the graphic positioning order causing display of the detected image, or the branch order to the detected character. Light pen switch status at the time initiate read was executed is indicated in device status word bit 7. In addition, the contents of the X and Y deflection registers (initiate read response words 2 and 3) might be significant. Detect Status indicates that the light pen has detected a point, vector, or character with interrupts deferred. This bit is reset whenever it is tested successfully or when device status word bit 2 is set.

Cycle-Steal Check Interrupt: Indicates the 2250 has stolen 32 consecutive cycles. This interrupt detects and stops a run-away cycle-steal situation in the 2250 caused by either 2250 malfunction or program error. If 32 consecutive cycles are stolen during any 2250 command, order, or character generation, regeneration is stopped, the busy bit is reset, cycle-steal requests are blocked, and an interrupt is requested. If a read status command is issued in response to this interrupt, no data is read since the cycle-steal request is blocked. The sense DSW command must be used following an unsuccessful read status command to examine the DSW and identify this interrupt. A reset display command or a manual reset must be given to clear the 2250 and prepare it for restarting the display.

Light Pen Switch Status: Indicates that the light pen switch was closed when last Start Timer order was executed.

Busy: Indicates that the display is currently regenerating in cycle-steal mode. This bit is always 0 if interrupt has occurred or display is not regenerating (or both).

Character Mode: Is equal to 1 when in basic or large-character mode; is equal to 0 when in graphic mode.

Point Mode: Is significant if bit 9 is equal to 0. Bit 10 is equal to 1 for point mode, or 0 for vector mode.

Address Displacement: Indicates the number of locations the address register (in first word of read status response) is ahead of address of the order being executed when detect interrupt occurred. Contains indeterminate value at any other time. Reset to 01.

2250 Model 4 Usage Meter: (This description applies to attachment via SAC.) This meter will run when the following conditions are present:

1. The 2250 enable/disable switch is in the enable position.
2. The 1131 usage meter is running.

The meter will continue to run simultaneously with the 1131 meter until the enable/disable switch is placed in the disable position. The condition required to change the status of the 2250 is: the CPU clock must not be running when the switch position is changed.

2250 Model 4 Usage Meter: (This description applies to attachment via SAC II.) This meter will run when the following conditions are present:

1. The 1133 in in an enabled status.
2. The 2250 enable/disable switch is in the enable position.
3. The 1131 usage meter is running.

The meter will continue to run simultaneously with the 1131 meter until the enable/disable switch is placed in the disable position, or the 1133 is disabled. The condition required to change the status of the 2250 is: the CPU must be in the stopped or wait state when the switch position is changed.

1133 Usage Meter

The 1133 meter runs when the following conditions are present:

1. The enable/disable switch is in the enable position.
2. The 1131 usage meter is running. This meter will continue to run simultaneously with the 1131 meter until the enable/disable switch is placed in the disable position. The condition required for honoring the switch in either the enable or disable position is that the 1131 clock must be stopped. The status of the 1133 cannot be changed (enable to disable or disable to enable) while the 1131 clock is running.

The storage access channel (SAC) provides the 1130 with additional I/O capability. If the 1403 Printer or 2310 Disk Storage is included in the system, then it is necessary to attach the 1133 Multiplex Control Enclosure to the SAC. However, an additional channel (SAC II) is provided by the 1133 as a special feature.

Through the facilities of SAC or SAC II, the user may attach his own device or the IBM 2250 Display Unit. The customer device may interrupt on any level from 2 through 5. Any bit within ILSW's 2 through 5 that has not been previously assigned may be used. This is also true for the assignment of area codes for the customer device. The customer device may be assigned any area code that has not been previously assigned.

FUNCTIONAL DESCRIPTION

The storage access channel feature allows external devices or systems to communicate directly with the 1131 core storage unit. The transfer of data to or from core storage and the SAC takes place in one of two modes.

1. Cycle Steal Mode: An XIO instruction, initiate read or initiate write, gives control of the data transfers to the SAC. When the SAC transfers a word or words to or from core storage, CPU cycles are stopped and a cycle steal cycle or cycles are taken. The CPU program has no control of or awareness of the cycle steal cycles.
2. Interrupt Mode: The external device can cause an interrupt of the CPU program by bringing up an interrupt-request-level 2, 3, 4, or 5 line, which is serviced by the CPU in the same manner as the basic interrupts.

Because of the SAC's ability to interrupt on levels 2, 3, and 5, interrupt level status words for these levels, as well as for level 4, are provided so that the CPU program may determine which device caused the interrupt.

When an interrupt is caused by a basic device, the CPU program must give an XIO sense interrupt command. The attachment for the device places the ILSW bit for that device on the I/O input bus, and reads the bit into the B-register and transfers to the accumulator. If a device on the SAC causes an interrupt, the CPU program must give an XIO sense interrupt command, and the device must decode the command and place its ILSW bit on the I/O input bus to be read into the B-register.

When an XIO sense device command is given to the SAC, the device must decode the command and set the status bits on the I/O input bus.

The customer must provide his own interrupt routines and controlling programs.

The customer may assign to devices on the SAC any area codes that are not already assigned to a basic device on his system. The decoding of the area codes is done in the devices on the channel.

The customer may assign any bit in the ILSW to a device on the SAC that is not assigned to a basic device on his system.

No change is made to the 1131 or the SAC attachment in the assignment of area codes, interrupt levels or ILSW bits.

Cycle-Steal Priority

There are four cycle-steal priority levels. The CPU Disk Storage is on level 0; SAC is on level 1; the 1132 is on level 2; and the 2501 is on level 3. There is no polling of cycle steal requests. That means the SAC, by keeping its request active, may completely block the 1132 printer and other lower priority devices.

PROGRAMMING

The storage access channel (SAC) operates on the IBM 1130 system under direct program control or cycle-steal control.

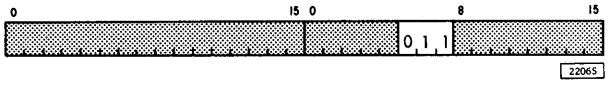
An XIO instruction addresses an I/O control command (IOCC) word, which is placed on the I/O output bus.

The devices or systems on the SAC must decode the IOCC area code to select one device or system for the operation.

The device or system selected must decode the function field and control the transfer of data to or from core storage.

I/O Control Commands

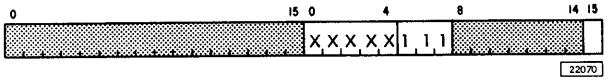
Sense Interrupt (011)



The sense interrupt IOCC is placed on the I/O output bus, and the interrupt level being serviced is sent to SAC. The device then sets its assigned bit on the I/O input bus. The CPU program then analyzes the ILSW and branches to the subroutine for the device.

The customer assigns interrupt status bits for the devices on the channel in his programs. The devices may bring up an interrupt status bit assigned by the customer. The interrupt status bits may be any bits not used by a basic device.

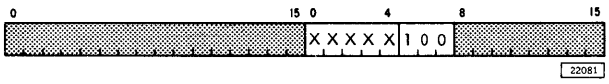
Sense Device (111)



This command sets the IOCC on the I/O output bus. The devices decode the area code and the selected device decodes the sense device function and sets the device status word (DSW) bits on the I/O input bus to read into the accumulator.

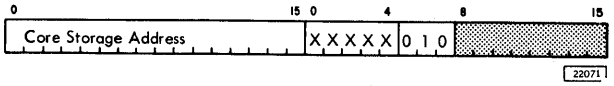
The conditions causing the interrupt are turned off by setting the modifier bit 15 to 1. If the device interrupts on more than one level, the conditions are turned off by modifier bit 15 for the highest level, bit 14 for the next highest level, etc.

Control (100)



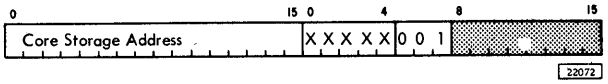
This command sets the IOCC on the I/O output bus. The devices decode the area code. The selected device decodes the control function and sets controls in the device to perform the action specified by the modifier bits (8-15) of EA + 1 or EA (address word). The device and the customer - provided programs control the function to be performed.

Read (010)



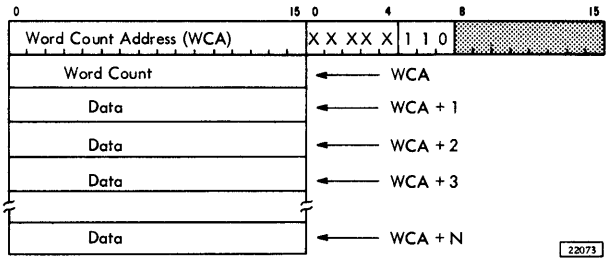
This command sets the IOCC on the I/O output bus. The devices decode the area code. The selected device decodes the read function and sets a single word on the I/O input bus.

Write (001)



This command sets the IOCC on the I/O output bus. The devices decode the area code. The selected device decodes the write function and takes the word from the I/O output bus.

Initiate Read (110)

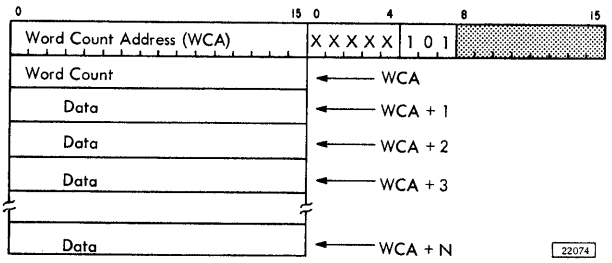


This command sets the IOCC on the I/O output bus. The devices decode the area code. The selected device decodes the initiate read function and sets the controls in the device for cycle-steal operation.

The word count address (WCA) is sent to the device.

The first cycle-steal cycle is taken and the word count is transferred to the device. The device then controls the transfer of data to the CPU core storage by cycle-steal level 1 cycles until the number of words specified by the word count has been transferred.

Initiate Write (101)



This command sets the IOCC on the I/O output bus. The devices decode the area code. The selected device decodes the initiate write function and sets the controls in the device for cycle-steal operation.

The word count address (WCA) is sent to the device.

The first cycle-steal cycle is taken and the word count is transferred to the device. The device then controls the transfer of data from the CPU core storage to the device by cycle-steal level 1 cycles until the number of words specified by the word count has been transferred.

For additional information regarding SAC, refer to IBM 1130/1133/SAC, Original Equipment Manufacturers' Information (A26-3645).

Special Power Sequencing Considerations

Since no power sequencing is provided by the CPU, one of the following procedures must be followed when powering up or down of the I/O device attached to the SAC to avoid possible loss of data:

1. Apply power to the I/O device. Then power up the CPU. Reverse procedure for powering down.
2. If CPU has power applied, hold dc reset depressed while powering up or down the I/O device.
3. CPU must be in a halt condition. Turn console mode switch to SS. I/O device may now be powered up or down without affecting CPU operation. When desired status of I/O device has been achieved, program operation may resume. (Halt is defined as the CPU stopped or in a wait condition with the run light out and no I/O devices operating.)

SYNCHRONOUS COMMUNICATIONS ADAPTER

The synchronous communications adapter special feature enables the IBM 1130 Computing System to function as a point-to-point or multipoint data transmission terminal, using either private or commercial common-carrier (switched or non-switched) line transmission facilities. The adapter sends data to or receives data from the line transmission facilities under control of the stored program in the 1130. It operates on an interrupt request basis similar to that used by other input/output devices in the IBM 1130 Computing System.

The synchronous communications adapter (SCA) provides data interchange between remote locations and a central data-processing location. The mode of communication may be either binary synchronous or synchronous transmit-receive and requires its own program. The mode is switch-selected by the operator. IBM supplies subroutines to support both modes.

The term "synchronous transmission" is used to describe continuous bit-stream transmission, without start-of-character identification. Thus, synchronous transmission is more efficient than start/stop transmission because fewer control bits are transmitted.

Binary Synchronous Communications (BSC)

The binary synchronous mode of data transmission provides for point-to-point and multipoint operation. The 1130 may be the primary station in a communication network or it may serve as a secondary station to a larger computing system. IBM programming systems provide primary and secondary station support for point-to-point operation and secondary station support for multipoint operation.

The capability of BSC mode to operate with any six-, seven-, or eight-bit level code provides the 1130 with the ability to communicate with a greater variety of devices. It is no longer necessary for a device to adhere to an eight-bit level code in order to communicate with the 1130 system.

Certain factors should be considered in selecting a character set if the user does not use the IBM-supported character sets. The six- and seven-bit codes provide a faster and more efficient type of communication because the data sets are rated in bits per second. Thus, the fewer number of bits to make a character, the more characters may be transmitted in any given segment of time. However, the number of separate characters that can be contained in a code is decreased, proportionately, as the number of bits used to make a character is decreased.

IBM programming systems support for the SCA in the BSC mode includes a subroutine for point-to-point operation and a subroutine for multipoint operation of a secondary station. Using these programs, text may be transmitted in either normal text (extended binary-coded-decimal interchange code, System/360 and 1130 internal code) or full-transparent text. Full-transparent text uses EBCDIC communication control characters. In normal text, data may not have the same bit configuration as any control character. In full-transparent text, control character recognition is handled by special procedure, thus making it possible to have data with the same configuration as control characters. Full-transparent text permits unrestricted coding of data within messages, and is useful in transmitting binary data, decimal data, and other data configurations.

A 2701 or 2703 Data Transmission Unit with the binary synchronous feature (SDA-2) must be attached to System/360 Models 30, 40, 50, 65 and 75 for communication in the BSC mode.

Synchronous Transmit-Receive (STR)

All synchronous transmit-receive (STR) devices use the four-of-eight line transmission code shown in Figure 48. The STR mode provides only point-to-point communication. It is used to communicate with the IBM 1009 Data Transmission Unit, the IBM 7701 and 7702 Magnetic Tape Transmission Terminals, the IBM 1013 Card Transmission Terminal, the IBM 7710 and 7711 Data Communication Units, and other STR devices.

The SCA provides the 1130 system with the ability to communicate with the communications adapter (#2073) of the Model 20 and with other System/360 configurations which have the IBM 2701 Data Transmission Unit attached. System/360 (other than Model 20) in the STR mode requires a 2701 (with the SDA-1 feature) attached to System/360.

LINE ATTACHMENT

The synchronous communications adapter is attached to either private or commercial line transmission facilities through a common-carrier data set. In the United States the interface for this data set is defined by EIA (Electronic Industries Association) Standard RS-232-B and requires a Western Electric data set model 201A3, 201A4, 201B1, 201B2, 202C1, 202D1, or an

Graphic	4 of 8 Code NXOR 8421	Graphic	4 of 8 Code NXOR 8421
blank	1111 0000*	F	0110 0110
¢	0110 1010	G	1000 0111
.	1000 1011	H	0111 1000
<	0110 1100	I	0110 1001
(	0101 0110	J	1101 0001
+	0011 0110	K	1101 0010
|**	1000 1101	L	1100 0011
&	1000 1110	M	1101 0100
!	1100 1010	N	1100 0101
$	0100 1011	O	1100 0110
*	1100 1100	P	0100 0111
)	0101 1100	Q	1101 1000
;	0011 1100	R	1100 1001
¬	0100 1101	none#	1010 1010
-	0100 1110	S	1011 0010
/	1011 0001	T	1010 0011
,	0010 1011	U	1011 0100
%	1010 1100	V	1010 0101
—	0101 1010	W	1010 0110
>	0011 1010	X	0010 0111
?	0010 1101	Y	1011 1000
:	0010 1110*	Z	1010 1001
#	0001 1011	0	1001 1010
@	1001 1100	1	1110 0001
'	0000 1111	2	1110 0010
=	0001 1110	3	1001 0011
"	0001 1101	4	1110 0100
A	0111 0001	5	1001 0101
B	0111 0010	6	1001 0110
C	0110 0011	7	0001 0111
D	0111 0100	8	1110 1000
E	0110 0101	9	1001 1001

*This is correct for System/360 Programs, but is not consistent with certain other STR devices. See the specific device manual.

**Group Mark

#Record Mark

114708

Figure 48. STR 4-of-8 Line Transmission Code

IBM-approved equivalent. Outside the United States the data set is defined by the CCITT (Consultive Committee on International Telephone and Telegraph) Standard and requires an IBM 3977 Modem or an IBM-approved equivalent.

The SCA can operate in half-duplex mode using either two-wire or four-wire line transmission facilities. Data rates, selected by the machine operator, are 600, 1,200, 2,000, or 2,400 baud (bits per second).

The adapter can be jumper wired to allow the program to control the data terminal ready condition in the data set interface. This selection will allow the program to control the disconnect of a switched data link.

Half-Duplex Operation

Half duplex is a mode of operation wherein either terminal can transmit or receive in conjunction with the remote terminal, but neither terminal can transmit and receive data simultaneously. In effect, the operation is quite similar to a normal telephone conversation; that is, one party talks while the other party listens. During the course of the conversation, each party may alternate between talking and listening as often as necessary.

Two-Wire Operation

Synchronous transmit-receive or binary synchronous operation with a two-wire half-duplex transmission system requires a delay of approximately 200 milliseconds when the adapter switches from receiving to transmitting data. This turnaround delay allows the data set and the communication lines to reverse the direction of transmission and line echo to settle. The amount of delay is therefore related to the character of the line and data set. Line turnaround time is controlled by the data set. When this turnaround is completed, the data set signals the adapter. The adapter does not transmit until the data set signals the completion of line turnaround.

Four-Wire Operation

When the adapter is connected to a four-wire half-duplex transmission system, line turnaround time is eliminated in the STR mode of operation. The adapter can remain synchronized by transmitting idles on one pair of wires while receiving data on the second pair of wires.

In binary synchronous mode (BSC) of transmission, the four-wire system eliminates line turnaround time, but resynchronization (which requires only the synchronous idle sequence) is always necessary before each transmission. Synchronization is always controlled by the sending device.

FUNCTIONAL DESCRIPTION

The entire synchronous communications adapter is contained within the 1131 Central Processing Unit. The adapter functions as an input/output control unit between the 1130 system and the transmission line. All data transfer is character-synchronous. This means that once an initial synchronous idle character is recognized, each subsequent character is recognized as a group of incoming data bits timed by an internal electronic clock for data terminal clocking or by the data set clock for data set clocking. Continuous regulation of the receiver's clock is provided in the case of data terminal clocking.

Incoming data from the transmission line is serial by bit and serial by character. As the data comes in, it is stored, one bit at a time, in the receive deserializer. When a complete character has been assembled, the character is transferred into the buffer register. Then the adapter initiates an interrupt request to notify the CPU that a character is ready to be read into core storage. When the interrupt request is serviced, the character is read in parallel into the high-order eight positions of a 16-bit word in core storage.

Outgoing data, from core storage to the transmission line, is taken in parallel from the high-order eight positions of the address location in core storage. The adapter initiates an interrupt request to notify the CPU that the adapter is ready to accept a character from core storage. When the interrupt request is serviced, the character is transferred in parallel to the adapter buffer register. Data from the register is subsequently sent to the transmission line one bit at a time.

Data transfer to or from the transmission line begins with the low order position. Each eight-bit character is located in bit positions 0-7 of a 16-bit core storage location as follows:

Bit Transfer Sequence	Bit Position in Core Storage
First	7
Second	6
Third	5
Fourth	4
Fifth	3
Sixth	2
Seventh	1
Eighth	0

The seventh and eighth (bit 6 and 7) bits are ignored when using a six-bit level code. The eighth bit (bit 7) is ignored when using a seven-bit level code (Figure 49).

Timers

There are three electronic timers in the SCA. Each timer is adjustable between 0.28 seconds and 3 seconds. One timer is set for 3 seconds and another is set for 1.25 seconds. The third timer is available for sync insertions in transparent mode, BSC. Its duration depends on the baud rate.

In the STR mode the three-second timer is designated as the receive timer and causes an interrupt and turns on DSW bit 3 when in the receive mode to signal the end of the listening period while establishing synchronization. This interrupt also occurs in the transmit mode if a clear to send is not received from the data set within a three-second period. Clear to send is a signal from the data set when it is ready.

The 1.25-second timer is used in the synchronize mode to signal the end of the transmission of idle characters for synchronization in the STR mode. It also causes an interrupt with DSW bit 3 on. This timeout is always coincident with a write response.

The third timer is inhibited in STR operation.

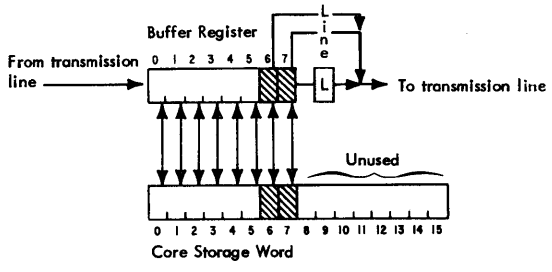
An XIO control (100) command with bit 10 on turns on a timer trigger which inhibits the 1.25- and 3-second timers when it is first issued. Issuing the command a second time removes the inhibited status, leaving the timers free to run. This command reverses the status of the timers each time it is issued.

The timers may be restarted at any time by issuing a sense device (111) command with bit 14 on if they are not inhibited.

In the BSC mode, the timers are set the same as for STR, but they have a different function. The receive timer (3 seconds) starts to run when the program enters the receive mode. The program should restart this timer when it detects the synchronous idle sequence (Figure 53). The sending station must transmit this sequence every 1.25 seconds. The 3-second timer also interrupts in the transmit mode if a clear to send is not received from the data set within 3 seconds. In either case DSW bit 3 is turned on.

The 1.25-second timer is used in the synchronize mode to signal the program that it is time to transmit the synchronous idle sequence.

The third timer (designated the program timer) will interrupt in either transmit or receive mode if it is allowed to run by the timer trigger. The IBM-supplied



Shaded areas show unused bit positions 6 and 7 for 6-bit and 7-bit codes respectively. 30023A

Figure 49. Communication Data Flow

subroutines use this timer and therefore it is not available for customer use when these subroutines are used.

An XIO control (100) command with bit 10 on inhibits the 1.25 and 3-second timers and starts the program timer. If the program timer is allowed to time out it resets the timer trigger and removes the inhibit condition from the other timers. Issuing another control command with bit 10 on also resets the timer trigger. A sense device (111) command with bit 14 on will restart any timer that is running.

Synchronous Transmit-Receive (STR) Operation

In order to communicate with a STR device, the STR/BSC switch must be placed in the STR position, and the 1130 must contain a program to control the communication. The program must use the four-of-eight code and must use STR line-control conventions. IBM provides a subroutine to control STR communication. This program is described in the manual IBM 1130 SCA Subroutines (C26-3706).

STR line-control conventions are described below. Most of the operations described are performed automatically when the IBM subroutine is used. These operations are described here for the user that wishes to write his own routines, and to provide a general understanding of STR communication.

IBM programming systems support for the SCA in the STR mode of operation uses the four-of-eight code. Two types of characters are used:

1. Control characters are used to control line functions; i.e., to acknowledge receipt of a message, to acknowledge synchronization, to signal the start of a message or the end of a transmission. The four-of-eight code, used by STR devices, contains special characters used to control line functions.
2. Data characters contain the information to be transferred to or from the adapter. The four-of-eight code contains 64 valid data characters; however, some STR devices do not utilize all of the 64 data characters. The 1130 system can recognize any or all of the 64 data characters as directed by the stored program, but the programmer should determine the character set recognized by the remote STR to avoid sending invalid characters.

Control Operations -- STR

The four-of-eight code contains special characters which are reserved for control functions. These control characters and their bit structures are shown in Figure 50. Control sequences are initiated by the 1130 program and are transmitted to the remote terminal as data. The remote terminal then has the responsibility of recognizing the control sequence and responding appropriately.

All operations of the adapter are controlled by the 1130 program. The program places the adapter in either the synchronize, transmit, or receive mode. In addition the program must initially store the idle character in the sync/idle register and, must generate the longitudinal redundancy check (LRC) character, which is transmitted at the end of each record.

The idle character is a special character which the adapter transmits automatically to the receiving terminal when no other data or control characters have been transferred to the adapter for transmission. This condition occurs

Control Characters	4 of 8 Code							
	N	X	O	R	8	4	2	1
Idle	0	0	1	1	1	0	0	1
Start of Record 1 or Acknowledge 1 (SOR 1 or ACK 1)	0	1	0	1	0	0	1	1
Start of Record 2 or Acknowledge 2 (SOR 2 or ACK 2)	0	0	1	1	0	0	1	1
Transmit Leader (TL)	0	0	1	1	0	1	0	1
Control Leader (CL)	0	1	0	1	0	1	0	1
End of Transmission (EOT)*	0	1	0	1	1	0	1	0
Inquiry or Error (INQ or ERR)	0	1	0	1	1	0	0	1
Telephone*	0	1	0	1	1	1	0	0
Group Mark	1	0	0	0	1	1	0	1
Longitudinal Redundancy Check (LRC)**	-	-	-	-	-	-	-	-

* Also used as a data character

** This character has a 0 bit in each bit position that contained an even number of 1 bits for that bit position in the data record. If that bit position in the record had an odd number of 1 bits the LRC character ranges from all 0s to all 1s and thus, is not in the 4 of 8 code.

16162A

Figure 50. STR Control Characters

during the synchronization mode at the start of each transmission, and when the program responds too slowly to the adapters request data. The idle character is not included in the LRC character. At least one idle character must be transmitted before each block of records. The adapter makes this transmission automatically on line turnaround.

Control characters are used generally in two-character sequences (Figure 51). Each sequence is made up of a leader character and a trailer character. Two of the control characters can be used as leaders of a control sequence. These are the transmit leader (TL) character and the control leader (CL) character. The special characters used as trailers each have two possible meanings depending on whether the TL or the CL character precedes them. For example, the INQ/ERR character is interpreted as an INQ character when preceded by the TL leader and is interpreted as an ERR character when preceded by the CL leader. The end-of-transmission sequence and the telephone sequence consist of one control character followed by one of two data characters. These data characters are interpreted as being part of a control sequence only when they are preceded by the CL character. When not preceded by the CL character, they are interpreted as data.

The inquiry control sequence is used by a terminal when it wishes to transmit a message. The terminal that is in control status may at any time send the inquiry control sequence, which notifies the other terminal of the desire to transmit and asks for permission to do so. If the other terminal is able to receive a message, it acknowledges the inquiry control sequence with an acknowledge sequence.

The start-of-record control sequence is transmitted immediately before each block of data. The start-of-record 1 (SOR 1) control sequence is transmitted before the first, third, fifth, etc., record of each message, while the start-of-record-2 (SOR 2) control sequence is transmitted before the second, fourth, sixth, etc., record of each message. This odd-even labeling of each record is used to ensure that no records of a message are lost or duplicated.

The end-of-transmittal record control sequence is sent immediately after each record of a message. The end-of-transmittal record control sequence contains the LRC character, which is used to check the validity of the transmission.

One of the acknowledge control sequences is sent by the receiving terminal after it correctly receives each block of data. This control sequence indicates to the transmitting terminal that it may proceed

Control Sequence	Control Character Sequence	
	Leader Character	Trailer Character
End of IDLE (EOI)*	CL	1 IDLE
Inquiry (Synchronized ?)*	TL	INQ
Acknowledge (Synchronized)	CL	ACK 2
Telephone Sequence *	CL	TEL
Acknowledge Telephone *	CL	TEL
Start of Record 1 (SOR 1) 1st or odd numbered record	TL	SOR 1
Start of Record 2 (SOR 2) 2nd or even numbered record	TL	SOR 2
End of Transmittal Record (EOTR)	TL	LRC
Acknowledge Record 1	CL	ACK 1
Acknowledge Record 2	CL	ACK 2
Repeat Last Record (ERROR)	CL	ERR
Intermediate LRC**	GM	LRC
End of Transmission (EOT)*	CL	EOT
Acknowledge EOT *	CL	EOT

*These sequences are always preceded by a 1.25 second transmission of IDLE characters.
** This sequence may be required on some terminals i.e. 1013, 7701, 7702

16163c

Figure 51. Control Sequences

to send another record. The acknowledge record 1 control sequence should be sent after a record that began with the start-of-record-1 control sequence is received, while the acknowledge record 2 control sequence should be sent after a record that began with the start-of-record-2 control sequence is received. This assures the sender that the receiver has not lost a record. The last acknowledgment is always sent in response to an inquiry.

The repeat last record (error) control sequence is sent by a receiving terminal if it receives a block of data that is in error. This sequence notifies the transmitting terminal that it should repeat the transmission of the last record.

The end-of-transmission control sequence is sent by the transmitting terminal after it has sent the last record of a message. This indicates that the message has been sent completely. A receiving terminal answers the end-of-transmission control sequence by sending back an end-of-transmission control sequence, thereby notifying the transmitting terminal that the receiving terminal has received the full message. After the transmission of these two end-of-transmission control sequences, the two terminals return to

synchronize mode of operation and exchange idle characters.

The telephone control sequence can be sent by either terminal and indicates that the terminal operator desires voice communication, via the handset, with the other terminal operator.

Synchronize Mode -- STR

The synchronize mode provides a means of synchronizing the transmitting and receiving terminals to ensure the proper recognition of data bits and characters as they are transmitted between terminals. The synchronize mode, sometimes referred to as handshaking, consists of the transmission of a series of characters for 1.25 seconds, followed by a control sequence and then turning around and listening for a similar series of characters from the other terminal for 3 seconds. The time intervals for transmit (1.25 seconds) and receive (3 seconds) are controlled by timers in the adapter. The timers are under control of the program. The character used in the synchronization sequence is called an idle character.

At the end of the 1.25-second transmission time, the transmitting terminal sends an end-of-idle control sequence -- a control leader (CL) followed by an idle character (Figure 55). This control sequence signals the receiving (remote) terminal to change from receive mode to transmit mode. When the turn-around is completed (200 ms for two-wire half-duplex) the remote terminal transmits the idle character for 1.25 seconds. At the end of this time the remote terminal sends the end-of-idle sequence. If neither terminal has a message to transmit, the synchronization sequence continues.

Transmit Mode -- STR

When a terminal has a message to transmit, that terminal sends 1.25 seconds of idle characters followed by the inquiry sequence. This sequence informs the remote terminal that a message is about to be transmitted. The remote terminal, if it is in synchronization and is ready to receive, sends an acknowledge control sequence. On receipt of the acknowledge sequence, the transmitting terminal transmits its message.

The first two characters of a message are the start-of-record-1 sequence. This sequence is preceded by one or more idle characters. This sequence is followed by the message data characters for this record. Some terminals may use or require an intermediate block check. (This sequence is GM-LRC.) At the end of the record, the end-of-transmittal-record (EOTR) sequence is sent. This sequence consists of a TL character and a longitudinal redundancy check (LRC) character. Two functions are performed by this sequence: it indicates the end of the record, and provides (via the LRC character) the receiving terminal with a method of checking for a complete message. The receiving terminal acknowledges the EOTR by sending the acknowledge 1 or 2 sequence (if LRC compares) or by the error sequence (if LRC does not compare).

Messages which contain more than one record indicate the start of the second record by sending a start-of-record-2 sequence. The start-of-record-1 sequence is used each time an odd-numbered record is transmitted, and the start-of-record-2 sequence is used each time an even-numbered record is transmitted. The use of the two different start-of-record sequences enables detection of lost or duplicated data records from a terminal.

When the receiving terminal has acknowledged the correct receipt of the last record of a message, the transmitting terminal sends the end-of-transmission sequence. This sequence consists of a CL character and an end-of-transmission (EOT) character. The receiving terminal acknowledges the EOT sequence by returning the same sequence. The terminals, if so programmed, return to the synchronize mode.

Receive Mode -- STR

In the receive mode, the adapter accepts data from other line devices and transfers it to the 1130 core storage. Prior to the transfer of data, the transmitting and receiving terminals must be synchronized.

In the receive mode, the adapter compares the incoming data to the character in the idle register. After at least one Idle character has been recognized, the first non-idle character detected and all subsequent characters including idles are transferred into core storage. Idle characters and control sequences are not included in the LRC. When the transmitting terminal signals the end of a record, the 1130 program checks the transmitted LRC character with the one compiled from the received record. If the two LRC characters are the same, the 1130 program generates the appropriate acknowledgment, which is then sent from the adapter to the transmitting terminal. If the LRC characters are not the same, the 1130 program responds with an error sequence which is then sent from the adapter to

the transmitting terminal. Normally, the 1130 program requests that the previous record be transmitted again. The number of transmission attempts is controlled by the programmer and may vary.

Special Programming

Special programming techniques are required in STR when an 1130 is used to communicate with a hardware device such as a 1013, 1009, or 7702. The special technique is required when either a 201 Data Set or an IBM 3977 Modem is used in a two-wire operation. These data sets do not allow data or control characters to modulate the carrier prior to the clear-to-send signal from the data set. Therefore, no idles are received from these devices before the control leader (CL) or transmit leader (TL). Since the 1130 SCA requires at least one recognizable character before interrupting the CPU, the following special technique should be used:

1. If the 1130 is the slave, it will be receiving records. After writing the acknowledgment character (ACK 1, ACK 2, or ERR), the program should load the sync/idle register with the TL. Since the TL is now the recognizable character, it is not loaded into the buffer for the CPU to read. The first character which interrupts the CPU is the trailer. The program must indicate to itself that the TL has already been received. This should be done when the first read interrupt occurs. If the 1130 times out, the remote station may send a message beginning with a TL or it may begin "handshaking" beginning with an idle character. To cope with either possibility, after a time-out, the 1130 program loads the sync register with a TL, and if another time-out occurs, the sync register is then loaded with an idle character. Once character phase is reestablished, the alternating of TL or idle characters ceases.
2. If the 1130 is the master, it is sending records. After writing an INQ, the LRC character of an EOTR, or the last character of an abort sequence (idle), the program should load the sync/idle register with the CL. Since the CL is now the recognizable character, it is not loaded into the buffer for the CPU to read. The first character which interrupts the CPU is again the trailer. The program must indicate to itself that the CL has already been received. This can be done either at the time the sync/idle register is loaded with the CL or when the first read interrupt occurs.

For both cases (1 and 2), the sync/idle register should be reloaded with the idle character prior to each transmission. An idle character should also remain in the sync/idle register after the program writes the idle of an end-of-idle sequence, or the TEL character, or the EOT character.

Since the above technique works for all data sets and STR devices, it is recommended that it be followed wherever a mixture of data sets and/or STR devices are used.

Binary Synchronous Communications (BSC) Operation

In binary synchronous operation the receiving terminal's ability to interpret the data it receives is the prime consideration in selecting the code to use for communication.

A variety of codes for communication is available. The user may select any code of six, seven, or eight levels. IBM programming systems for the 1130 use the extended binary-coded-decimal interchange code (EBCDIC) communication control characters for all BSC operations. Figure 52 shows the control characters and Figure 53 shows the sequences in which they are used. In full-transparent text, control character recognition should be handled by a special procedure, thus making it possible to have data with the same configuration as control characters. All characters are transferred to core storage in the CPU for program interpretation.

In the selection of a code, care must be taken in selecting the proper SYN character.
If only two characters are used for synchronization, the first bit of the SYN character must be 0. In all cases the bit configuration must be nonrepeating; that is, if several SYN characters are in a continuous flow, the bit configuration must be such that the beginning of each character can be recognized.

Control Operation -- Binary Synchronous

The binary synchronous communications control procedures are generally independent of the transmission code. Any code having a fixed number of bits (six, seven, or eight) per character may be used if the ten control characters are set aside and a proper choice is made for the synchronous idle character. The EBCDIC control sequences are presented in this manual (Figure 57).

The control sequences are initiated by the 1130 program and transmitted to the remote terminal as data. The remote

Character	Bit Configuration 0 1 2 3 4 5 6 7	Meaning
SYN	0 0 1 1 0 0 1 0	Synchronous Idle
DLE	0 0 0 1 0 0 0 0	Data Link Escape
ENQ	0 0 1 0 1 1 0 1	Enquiry
SOH	0 0 0 0 0 0 0 1	Start of Heading
STX	0 0 0 0 0 0 1 0	Start of Text
ETB	0 0 1 0 0 1 1 0	End of Transmission Block
ETX	0 0 0 0 0 0 1 1	End of Text
EOT	0 0 1 1 0 1 1 1	End of Transmission
NAK	0 0 1 1 1 1 0 1	Negative Acknowledgement
*ACK 0	0 1 1 1 0 0 0 0	Positive Acknowledgement (even record)
*ACK 1	0 1 1 0 0 0 0 1	Positive Acknowledgement (odd record)

* Control characters when preceded by DLE

30018

Figure 52. Binary Syncnronous EBCDIC Control Characters

Characters	Meaning
ENQ	Enquiry
SOH	Start of Heading
STX	Start of Text
DLE STX	Start of Transparent Text
ETB CRC-16*	End of Block
DLE ETB CRC-16	End of Transparent Block
ETX CRC-16	End of Text
DLE ETX CRC-16	End of Transparent Text
DLE ACK 1	Acknowledgement of Odd Record
DLE ACK 0	Acknowledgement of Even Record
NAK	Negative Acknowledgement
EOT	End of Transmission
DLE EOT	Disconnect Signal
SYN SYN	Synchronous Idle (Normal)
DLE SYN	Synchronous Idle (Transparent Text)

* CRC-16 is a 16-bit cyclic check character accumulated from text and heading data.

30019

Figure 53. Binary Synchronous Control Sequences

terminal then has the responsibility of recognizing the control sequences and responding appropriately.

All operations of the adapter are controlled by the 1130 program. The program places the adapter in either the synchronize (transmit) or the receive mode. In addition the program must initially store the synchronous idle (SYN) character in the sync/idle register. The program also accumulates the block check character (CRC-16), which is transmitted at the end of each record.

In BSC, data may be transmitted in two modes: normal (EBCDIC) text and full-transparent text. In normal text mode, data may not have the same bit configuration as any control character. In full-transparent text, data may contain any bit configuration since control character recognition is handled by a special procedure. Full-transparent text is quite useful in transmitting machine language and other codes that may contain control characters.

In full-transparent mode, the DLE STX sequence is a special sequence that is transmitted prior to transmitting full-transparent text. When a receiving terminal receives this sequence it will stop checking for control characters and treat all subsequent characters as transparent text. The only control character that is recognized is another DLE character. The detection of another DLE character switches the mode back to normal text mode, and the receiving terminal will start checking for control characters. If the next character is DLE or SYN the receiving program will treat the character as data or as synchronous idle and will return to the transparent mode. Therefore, in full-transparent text mode, all control characters, including SYN, must be preceded by the DLE character to be recognized by the receiving terminal. In full-transparent mode the program must store the DLE character in the sync/idle register. The SYN character must be stored after leaving full-transparent text mode.

Line Turnaround

When a terminal wishes to transmit, it sends two SYN characters followed by the ENQ character. Then the terminal goes to the receive mode and waits for an acknowledgment from the receiving terminal The receiving terminal detects the ENQ character as a request from the transmitting terminal, goes to the transmit mode, and replies with a positive acknowledgment (ACK0) if it is ready to receive. When the transmitting terminal receives the positive acknowledgment, it may start to transmit its record. If the receiving terminal is not ready to receive, it should respond with the NAK character (negative acknowledgment). If the terminal is unable to respond, the transmitting terminal will time out in 3 seconds.

Several of the control characters when detected by the program should cause line turnaround; that is, the transmitting

terminal switches to the receive mode and the receiving terminal switches to the synchronize (transmit) mode.

The end-of-block and the end-of-text (ETB and ETX) characters, (when transmitting without checking), also cause line turnaround. IBM subroutines always use the block check character. The acknowledgments alternate: ACK1 for the first record and all succeeding odd records, and ACK0 for the second record and all succeeding even records. If the block check character is used, the line turnaround follows it. When a station is through transmitting, it may relinquish its right to transmit by sending the end-of-transmission (EOT) character. The EOT character does not require an acknowledgment. The right to transmit reverts back to the master station or to contention if a master station is not designated.

Multi-Point Operation

In multi-point, centralized operation, IBM programming systems include subroutines that permit the 1130 to operate only as a slave station. Programs to support noncentralized operation must be supplied by the user. A slave station is one that may respond to a call from the control (master) station but cannot initiate the call. Initialization is performed when the control station sends polling or selection addresses. A particular polling address gives a unique station on the line an opportunity to transmit to the control station. The polled station responds with a positive response (data transmission) or a negative response (EOT). Selection addresses are used to request a particular station to receive data transmission. A selected station responds with its status, ready to receive (ACK0) or not ready to receive (NAK).

A nonselected terminal should restart the timers and reset character phase on recognition of all turnaround sequences.

In noncentralized operation, the operation is similar to centralized operation except the selected station (after being polled) must respond with its address and the address of the station to which it wishes to transmit. The selected station must reply with its address and a positive acknowledgment if it is ready to receive or a negative acknowledgment if it cannot receive.

Receive Mode-Binary Synchronous

In the receive mode, the adapter accepts data from the transmission line and transfers it to the 1130 core storage. Prior to the transfer of data, the adapter must be synchronized with the transmitting terminal. An initiate read command (110) with all modifier bits (8-15) set to 0 places the adapter in a receive mode.

In the receive mode, the adapter compares the incoming data to the character in the sync/ idle register. After at least two SYN characters have been recognized, the first non-SYN character detected and all subsequent characters including SYN characters are transferred to core storage. The receive mode is terminated by the program when it detects a valid turnaround sequence.

For a slave station, if a receive time-out occurs, an end operation command should be used to reset the clock and character phase. The slave should issue an initiate read command immediately after the end operation command. If a receive time-out occurs, the master should issue an initiate write command to send ENQ.

Data Transmission -- Binary Synchronous

To eliminate some marginal conditions associated with some data sets, all messages should be preceded by six SYN characters, two of which are sent automatically by the adapter. Two SYN characters are required when the 1130 is receiving. (This applies to both master and slave.) In four-wire operation, the SYN characters must be preceded by a pad character. The pad character should not be another SYN character. It could be a marking-line character.

When a terminal has a message to transmit, the terminal sends the synchronous idle sequence followed by the enquiry control character. The enquiry character informs the remote terminal that a message is about to be transmitted. The remote terminal, if it is ready, synchronizes on the SYN characters and acknowledges by sending the acknowledge control sequence (DLE ACK0). Upon receipt of the acknowledge control sequence, the transmitting terminal transmits its message. The entire message, including control characters and check characters, is generated and transmitted from core storage under control of the stored program in the CPU.

Synchronize Mode -- Binary Synchronous

The synchronize mode in binary synchronous communication is a transmit mode which allows a timeout to occur if the transmission is longer than 1.25 seconds. The program must insert the synchronous idle sequence after this timeout to ensure that the receiving terminal remains synchronized. Data transmission may continue after the synchronous idle

sequence. The receiving terminal will time out if it does not receive the synchronous idle sequence within 3 seconds. A control command (100) with bit 11 set to 1 places the adapter in the synchronize mode.

Transmit Mode -- Binary Synchronous

The transmit mode may be used in binary synchronous operation in lieu of the synchronize mode where a time-out is not required or desired. An initiate write command (101) with bit 9 set to 0 places the adapter in the transmit mode.

PROGRAMMING

All adapter operations are programmed using the 1130 XIO instruction (see execute I/O description in this manual). The effective address position of the XIO instruction specifies the address of the two-word IOCC which is required for the desired operation.

The adapter interrupts the 1130 system program on interrupt level 1. Bit position 1 of this interrupt level status word (ILSW) indicates that the interrupting device is the adapter. The program then senses the device status word (DSW).

The DSW is generated by the adapter to indicate the cause of the interrupt (Figure 54).
The DSW bit positions indicate the following conditions:

Bit 0 - The adapter is in receive mode (or diagnostic mode) and the buffer register in the adapter contains a data character which should be transferred into the 1130 core storage.

Bit 1 - The adapter is in transmit mode (or diagnostic mode) and requires a data character from the 1130 core storage for transmission.

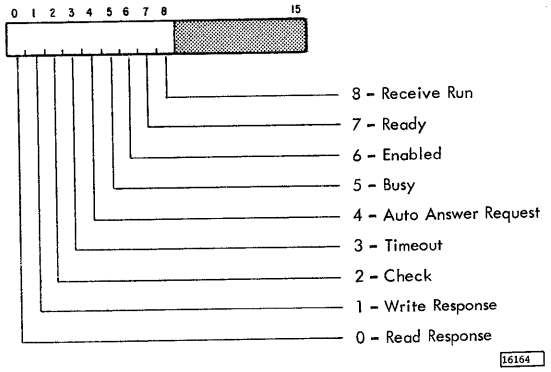


Figure 54. Device Status Word

Bit 2 - This bit indicates an error condition: data overrun or character gap.

Data overrun indicates that a character was still in the buffer when another character came, either from the transmission line (receive mode) or from core storage (transmit mode). This condition results in a loss of data. In the transmit mode, data overrun is the result of a program sending another character to the adapter without an interrupt request from the adapter. In the receive mode, this condition is the result of a program operating too slow; that is, a character is received from the transmission line before the preceding character has been transferred to core storage.

Character gap indicates that the data characters are being received by the buffer too slowly for correct adapter operation. In the transmit mode, the program is operating too slowly. (Note: The adapter automatically transmits the character in the sync/idle register.) In the receive mode, the program requested another character from the adapter without an interrupt request from the adapter.

Bit 3 - In STR this bit indicates the end of the 1.25-second timeout for transmission of idle characters, or the end of the three second listening time for synchronization. In BSC this bit indicates it is time to insert the synchronous idle sequence in synchronize mode, or a receive time-out occurred in the receive mode or a sync insertion is required in transparent mode.

Bit 4 - This bit indicates that the data-set phone is ringing.

Bit 5 - This bit indicates that the adapter is in either the receive or transmit mode.

Bit 6 - This bit indicates that the adapter has been enabled for an auto answer request interrupt. (See Bit 4.)

Bit 7 - This bit indicates that the data set is connected and ready to receive, synchronize, or transmit data.

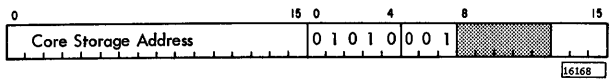
Bit 8 - This bit is used with two-wire half-duplex STR systems only.

It indicates that the adapter is in the "slave" mode. In slave mode the adapter transmits the normal acknowledge responses but does not transmit data records. Receive and transmit clocks are tied together.

I/O CONTROL COMMANDS (IOCC)

The adapter is addressed by the five-bit (bits 0 through 4) device (area) code in the IOCC. This code is 01010.

Write (001)



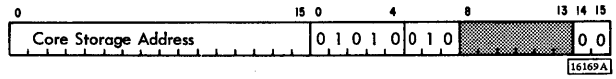
A write command without a modifier bit instructs the 1130 to transfer the contents of the specified core storage address to the adapter buffer. The adapter then serializes the contents of the buffer register onto the transmission line.

If modifier bit 13 is set, this command is used to set the sync/idle register. The 1130 transfers the data to the sync/idle register in the adapter. The idle character is transmitted during the synchronize mode or when the adapter is in transmit mode and has not received a data character for transmission.

Modifier bit 14 turns on the audible alarm trigger in the adapter.

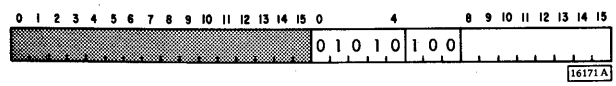
Modifier bit 15 turns off the audible alarm trigger.

Read (010)



The read command instructs the 1130 to transfer the contents of the adapter buffer to the core storage location specified in the address portion of the command. Modifier bits 14 and 15 must be 0's in application programs. When on, they are used for reading diagnostic words.

Control (100)



The control command is always used with a modifier bit. This command causes the adapter to accomplish the functions specified by the modifier.

Modifier bit 8, when set to 1, enables the adapter for auto answer operation. Auto answer allows the adapter to interrupt the 1130 program in response to a telephone ring from the remote terminal.

Modifier bit 9, when set to 1, disables the auto answer operation and does not allow a telephone ring from the remote terminal to interrupt the 1130 program.

Modifier bit 10 reverses the status of the timers from run to inhibit or from inhibit to run.

Modifier bit 11 sets the adapter to the synchronize mode. This is used to establish and maintain synchronization in the STR mode with minimum program interruption. Idles are transmitted without the program being interrupted until transmit time-out occurs.

In binary synchronous mode, modifier bit 11 allows the adapter to transmit in the synchronize mode. Write responses occur normally. A transmission longer than 1.25 seconds causes a time-out interrupt. The program must transmit the synchronous idle sequence before continuing to transmit data. The synchronous idle sequence is the only synchronization necessary in BSC. This usually consists of two sync characters.

The on condition of modifier bit 12 places the adapter in a diagnostic condition. Bit 12 should be off for all application programs. Because of the short time between interrupts in this condition, the diagnostic program should be run alone.

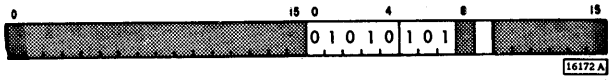
Modifier bit 13 is the end operation command. This command resets the adapter regardless of the mode of operation. If the adapter is in the transmit mode, the reset is delayed until a character gap of one character is detected. This allows the last character to get through the data set before the adapter is reset. It then resets the adapter and also resets the timers used in the synchronization mode and disconnects the adapter from the communication line if a switched network is used. In binary synchronous mode, this command should be issued after a receive time-out to reset the clock.

Modifier bit 14 is used to set the adapter for a six-bit character frame. Setting bit 14 automatically resets the seven-bit character mode.

Modifier bit 15 is used to set the adapter for a seven-bit character frame. Setting bit 15 automatically resets the six-bit character mode.

Both frame size modes are reset when the adapter leaves both the receive and transmit mode. Thus it becomes necessary to reenter the proper mode after each line turnaround. Attempting to set both bit 14 and bit 15 with the same instruction is ambiguous and may result in an error.

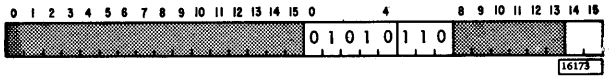
Initiate Write (101)



The initiate write command places the adapter in the transmit mode of operation.

Initiate write with modifier bit 9 on resets all conditions in the adapter.

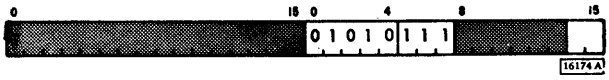
Initiate Read (110)



The initiate read command places the adapter in the receive mode of operation.

An initiate read command with modifier bit 14 on sets the send/receive run trigger and places the adapter in slave mode operation for STR operation. This mode of operation is used with two-wire half-duplex systems. In the slave mode the adapter should not be programmed to transmit data records to the master. The only transmissions that the adapter will make are the normal responses to the inquiry from the master and the normal acknowledgments. The start-read command and a modifier bit 15 clears the send/receive run trigger and removes the adapter from slave mode operations. This clearing places the adapter in the master mode, which is used for the transmission of data.

Sense Device (111)



The sense device command instructs the 1130 to sense the device status word (DSW). The DSW is generated by the adapter to indicate the cause of the interrupt. The DSW for the adapter is shown in Figure 54.

Sense DSW with modifier bit 14 on will restart the timers. If the synchronous idle sequence is received while in BSC receive mode, the program should restart the timer. If the timer is not reset within 3 seconds, the adapter will cause a time-out interrupt. Sense DSW with modifier bit 15 on resets the device status word responses.

TIMING FOR SCA PROGRAMMING

In order to prevent an overrun on receive, a character must be sent from the SCA buffer following a read response interrupt within the period shown in Figure 55. Also to prevent a character gap on transmission, a character must be written to the SCA buffer following a write response within the period shown in Figure 55.

Time Between Characters

Baud \ Char. Size	6 Bit	7 Bit	8 Bit
600	10.0 ms	11.6 ms	13.3 ms
1200	5.0 ms	5.8 ms	6.6 ms
2000	3.0 ms	3.5 ms	4.0 ms
2400	2.5 ms	2.9 ms	3.3 ms

Character Rate

Baud \ Char. Size	6 Bit	7 Bit	8 Bit
600	100 cps	85.7 cps	75 cps
1200	200 cps	171 cps	150 cps
2000	333.3 cps	286 cps	250 cps
2400	400 cps	343 cps	300 cps

3002-44

Figure 55. Transmission Timing

OVERLAPPING INPUT/OUTPUT OPERATIONS AND THROUGHPUT CONSIDERATIONS

The 1130 system permits input/output devices to operate simultaneously; that is, to overlap their operation with other functions of the CPU. Overlapping I/O operations provides increased data throughput and more efficient utilization of the central processing unit.

This section will aid the programmer wishing to maximize I/O throughput in the 1130 system. A primary concern, however, is the possible loss of data if the capabilities of the system are saturated by overlapping too may operations. Loss of data can occur only on an 1130 system which has either a 1442, 1132, or synchronous communications adapter, and then only if too many I/O operations are overlapped with these devices.

The material provided here may be used by the programmer to calculate the maximum throughput for his system without loss of data.

Cycle-Stealing Concept

The cycle-stealing concept of the 1130 permits the CPU program to start an operation on an I/O device and then continue the mainline program while the I/O device is performing its operation. Each I/O device that operates in this manner takes (steals) a cycle from the CPU when it is needed.

The CPU is "tied up" only one cycle while a data character is being transferred. The frequency at which devices steal cycles depends on the type of device.

Since the CPU is much faster than any I/O device on the system, the CPU may be performing another function, such as arithmetic, at the same time an I/O operation is being performed. In fact, several I/O operations may be overlapped with each other and with other CPU functions. For example, the data transfer rate of a disk storage drive is 27.8 us. per word. Thus, each disk storage drive read/write operation requires one CPU cycle (3.6 or 2.2 us. out of each 27.8 us., leaving 25.6 or 24.2 us. CPU time available for other functions. If two disk storage drives are transferring data at the same time, then 23.4 or 20.6 us. is available for other CPU functions.

Direct Program Control (via Interrupt)

Direct program control applies to I/O devices that are totally dependent upon the CPU program. These devices interrupt the mainline program by requesting service. Once the service request is honored, the actual transfer of data also requires programmed commands. Servicing interrupt requests generally requires several CPU instructions (a subroutine). The system programmer must calculate the time of the I/O subroutines to determine the maximum data throughput (without data loss) of his system configuration. See IBM 1130 Subroutine Library, Form C26-5929, for the execution times for IBM-supplied subroutines.

Conditions causing I/O interrupt requests are preserved in the device status word (DSW) of the I/O devices until the interrupt is accepted by the CPU.

The sequence of events after an interrupt request is received is:

1. Instruction in progress is allowed to continue to completion.
2. Interrupt request is accepted if higher level interrupt is not in progress.
3. Branch to an appropriate interrupt subroutine to service the request.
4. Housekeeping program routines must store all registers and linkage addresses to allow mainline program to continue after the interrupt is serviced.
5. Examine the interrupt level status work (ILSW) to determine the interrupting device.
6. Examine the DSW of the interrupting I/O device to determine the action required to service the request and reset the interrupt response bit in the DSW.
7. Service the request and restore the necessary register and address information to resume the mainline program operation or to service other interrupts. Turn off the interrupt level with a BOSC instruction (a BSC instruction with a bit 9 set to 1).

Exposure to Loss of Data

Some direct program control devices and all non-buffered cycle-steal devices are time-dependent (require service within a specified time). Time-dependent devices are subject to losing data if not serviced within specified times. These times vary depending upon the device type and function being performed.

A significant factor that must be considered by the programmer is the priority levels of the devices in his system configuration and the times in which these devices must be serviced.

Device Priority

Overlapping I/O operations in a computing system requires that a priority sequence be established. In the 1130 system, the priority levels for both cycle-steal and interrupt are established for each device that can be attached to the system.

Cycle-Steal Priority

A cycle-steal request may be honored at the end of any core storage cycle. Cycle stealing allows an external device to intervene during the processing of a CPU operation and use one or more core storage cycles in order to communicate directly with CPU core storage. At the completion of the cycle-steal operation, CPU operation is resumed at the point where the cycle-steal request occurred.

CPU cycle-steal level 0 - Single Disk Storage (in the CPU)
CPU cycle-steal level 1 - SAC/2250 (see the multiplex levels below)
CPU cycle-steal level 2 - 1132 Printer
CPU cycle-steal level 3 - 2501 Card Reader

Cycle-steal level 1 is subdivided by the channel multiplexer (when the 1133 is attached to SAC) as follows:

Multiplex level 0 - 1st 2310 Disk Storage Drive
Multiplex level 1 - 2nd 2310 Disk Storage Drive
Multiplex level 2 - 3rd 2310 Disk Storage Drive
Multiplex level 3 - 4th 2310 Disk Storage Drive
Multiplex level 4 - Reserved (RPQ)
Multiplex level 5 - Reserved (RPQ)
Multiplex level 6 - SAC II/2250
Multiplex level 7 - 1403 Printer
Multiplex level 8 - Reserved
Multiplex level 9 - Reserved
Multiplex level 10 - Reserved
Multiplex level 11 - Reserved (RPQ)

The preceding assignments are given in consideration of expansion for the user who may wish to expand his system at a later date. The cycle-steal levels listed as "Reserved (RPQ)" are for RPQ (Request for Price Quotation from IBM) activity.

Interrupt Priority

Interrupts are caused by a request for service from an I/O device or by the termination of an I/O operation. The interrupt facility provides an automatic branch (to one of core storage locations 8-13) from the normal program sequence in order to react to an external request or condition.

At the completion of any program instruction, any pending interrupt requests are serviced if no higher level interrupt is in progress.

Interrupts are assigned priority levels to allow the most efficient use of all attached I/O devices in the system.

Level	Device
0	1442 Card Read Punch (column read, punch)
1	1132 Printer, Synchronous Communications Adapter
2	Disk Storage, Storage Access Channel (SAC)
3	1627 Plotter, SAC, 2250 Display Unit
4	1442 (operation complete), keyboard console printer; 1134 Paper Tape Reader, 1055 Paper Tape Punch, 2501 Card Reader 1403 Printer, 1231 Optical Mark Page Reader, SAC
5	Console (program stop switch and interrupt run), SAC

Service Request Limitations

The I/O devices in the 1130 system are subject to loss of data or extremely reduced throughput if service request, either cycle-steal or interrupt, is not honored within times given next. (Refer to Figure 56 for a summary of service request times.)

Device	Time allowable to service I/O request without data loss		Time allowable to service I/O request to maintain rated speed		Time allowable to service end of operation interrupt to maintain rated speed	Frequency of request		
	Interrupt	Cycle-steal	Interrupt	Cycle-steal		I/O Interrupt	I/O Cycle-steal	End Operation Interrupt
Disk Storage Drives	-	27.8 µsec	-	27.8 µsec	500 µsec	-	27.8 µsec	9 ms
1403 Printer Model 6	-	-	-	3 ms out of 32 ms	29 ms		18 µsec	176 ms
1403 Printer Model 7	-	-	-	2 ms out of 19.9 ms	17 ms		11 µsec	100 ms
1132 Printer	1.5 ms	16 consecutive cycles within 300 µsec	1.5 ms	16 consecutive cycles within 300 µsec		11.2 ms	16 cycles every 11.2 ms	1344 ms
2501 Card Reader Model A1	-	466 µsec	-	466 µsec	18.3 ms	-	482 µsec	100 ms
2501 Card Reader Model A2	466 µsec	-	466 µsec	-	3.0 ms	-	482 µsec	60 ms
2250 Graphic Display	-	-		-				
1442-6 Card Punch	300 µsec	-	300 µsec	-	25 ms	12.5 ms	-	1216 ms
1442-5/7 Card Punch	300 µsec	-	300 µsec	-	25 ms	6.25 ms	-	663 ms
1442-6 Card Read	800 µsec	-	800 µsec	-	35 ms	2.5 ms	-	200 ms
1442-7 Card Read	700 µsec	-	700 µsec	-	25 ms	1.87 ms		150 ms
(SCA (8-bit, 2400 baud)	3.3 ms	-	3.3 ms	-	200 ms (depends on line turn around)	3.3 ms	-	(depends on number of characters per record)
1231 OMPR	-	-	13.2 ms	-	130 ms	13.2 ms	-	2000 ms
1134 Paper Tape Reader	-	-	500 µsec	-	16 ms	16.7 ms	-	16.7 ms
1055 Paper Tape	-	-	8 ms	-	8 ms	66.7 ms	-	66.7 ms

30119

Figure 56. Table of I/O Timing Requirements

Cycle-Steal Devices

Disk Storage Drives require one CPU cycle every 27.8 us. while an XIO initiate read/write operation is in progress. The end of operation interrupt request may wait indefinitely without losing data but should be completely serviced within 500 us. to gain maximum throughput. The disk storage drives are assigned to the highest cycle-steal priority levels in the system because of their fast data transfer rate.

1403 Printer requests one CPU cycle every 11 us. (model 7) or every 18 us. (model 6) while an XIO initiate write is in progress. However, the 1403 is fully buffered and is not subject to losing data if its request remains unhonored. In fact, the 1403 is designed to prevent it from interfering with time-dependent devices on lower priority levels.

The programmer does not need to consider the 1403 regarding loss of data but should consider it regarding throughput.

In order to maintain 340 lines per minute with model 6, the space command (XIO control) must be issued within 117 ms following the transfer complete interrupt. Also, the print complete interrupt must be serviced and the buffer loaded for the next print line within 32 ms (the time required to space one line). If the 1403 received all of its cycle-steal requests without interference, 3 ms would be required to load the model 6 buffer. Therefore, approximately 29 ms is available to service the print complete interrupt.

In order to maintain 600 lines per minute with the model 7, the space command

must be issued within 72 ms following the transfer complete interrupt. Also, the print complete interrupt must be serviced and the buffer loaded for the next print line within 19.9 ms. The model 7 takes about 2 ms to load the buffer. Therefore, approximately 17.9 ms is available to service the print complete interrupt.

Note: If the space command is not issued until after the print complete interrupt has occurred, the 1403 will not maintain rated speed.

1132 Printer operates in both the cycle-steal mode and the direct program control (interrupt) mode. The 1132 requires 16 consecutive CPU cycles within 300 us. following a read emitter instruction. The 1132 also requests an interrupt (direct program control) every 11.2 ms. This request must be honored within 1.5 ms.

2501 Card Reader requires one CPU cycle every 466 us. while and XIO initiate read operation is in progress. The end operation interrupt request may wait indefinitely without losing data but should be serviced within 18.3 ms (model A1) or 3.0 ms (model A2) to maintain rated speed.

2250 Graphic Display is designed to prevent it from interfering with the 1442, 1132, 2501, or synchronous communications adapter by inhibiting its cycle-steal request while these devices are being serviced. In effect, this inhibiting causes the 2250 to be on a priority level lower than any device except the 1403. Because the 2250 is not subject to losing data (actually, the brilliance of the screen could fade on a 3.6 - us. system), it may be overlapped with any or all devices in the system.

The time demand from the 1131 CPU varies depending on the mode (character or vector), the number of characters displayed on the screen, the actual characters displayed, and the state of the CPU (wait or processing). The greatest time demand (CPU in the wait state) could be almost every CPU cycle. The least time demand could be about two CPU cycles every 25 ms.

The average interference with CPU processing is:

For 3.6 us. core storage - character mode
80 % of CPU cycles
vector mode --
20 % of CPU cycles
For 2.2 us. core storage - character mode
66% of CPU cycles
vector mode --
18% of CPU cycles

Program Control Devices

1442 Card Punch requires the punch interrupt request be serviced within 300 us. (model 5, 6, and 7) to prevent loss of data. The end operation interrupt should be serviced within 25 ms to obtain rated speed.

1442 Card Read requires the read interrupt request be serviced within 800 us. (model 6) or 700 us. (model 7) to prevent loss of data. The end operation interrupt must be serviced within 35 ms (model 6) or 25 ms (model 7) to obtain rated speed.

Synchronous Communications Adapter operates at one of several transmission speeds. The time between character transfer interrupts depends on the speed selected by the speed selection switch and the number of bits per character. The times between characters (interrupts) for the various combinations of bit speed and character size are shown in the following chart:

Baud \ Char Size	6 Bit	7 Bit	8 Bit
600	10.0 ms	11.6 ms	13.3 ms
1200	5.0 ms	5.8 ms	6.6 ms
2000	3.0 ms	3.5 ms	4.0 ms
2400	2.5 ms	2.9 ms	3.3 ms

30118

Data may be lost if the SCA read request is not honored within the times shown. If the 1130 is transmitting, data will not be lost but fill characters will be automatically inserted, thus reducing actual effective baud rate.

1231 Optical Mark Page Reader requests a read response interrupt each time the one-character buffer in the attachment is loaded. If the request is honored within 13.2 ms, the requests occur every 13.2 ms until the entire data sheet has been read. However, the 1231 has a sonic delay-line buffer capable of storing all characters from a single data sheet. Therefore, data will not be lost if the read response interrupt request remains unhonored longer than 13.2 ms.

To maintain the rated throughput of 2,000 data sheets per hour, the 1231 read response interrupt request should be honored within 13.2 ms, and the read start command (XIO control with bit 13 on) for the next data sheet should be issued within 130 ms after the first read response interrupt has been serviced. The

programmer must be aware of the possibility of a read error causing a data sheet to be selected and the operation terminated before all characters have been read and transferred to core storage. If data from data sheets is directly related to the previous sheets (and assuming the previous sheet has been read correctly), then the read start must not be issued until after the operation complete interrupt has occurred and the error indicators have been tested.

The 1231 is on interrupt level 4 and will not impact the throughput of other devices on the system.

1134 Paper Tape reader requests a read response 500 us. after a feed command. A read command should be given to accept the character stored in the paper tape attachment buffer before the next feed command is issued. The continuous read rate is 16.7 ms per character. A feed command must be issued 16 ms after the response interrupt to maintain rated speed. The 1134 is not subject to losing data unless two feed commands are issued consecutively. The 1134 is on interrupt level 4 and will not impact the speed of the other devices on the system.

1055 Paper Tape Punch requests a punch response interrupt every 66.7 ms if punching continuously and should be serviced within 8 ms following the interrupt request to maintain rated speed. The 1055 is on interrupt level 4 and will not impact the speed of other devices on the system.

APPENDIX A. CHARACTER CODES

Ref. No. ①	EBCDIC Binary	EBCDIC Hex	IBM Card Code Rows	IBM Card Code Hex ④	Graphics and Control Names	Console Printer Hex	PTTC/8 Hex ②	1132 Hex ③	1403 Hex
0	00000000	00	12,0,9,8,1	B030	NUL				
1	0001	01	12,9,1	9010	SOH				
2	0010	02	12,9,2	8810	STX				
3	0011	03	12,9,3	8410	ETX				
4	0100	04	12,9,4	8210	PF Punch Off				
5*	0101	05	12,9,5	8110	HT Horiz Tab	41 ⑤	6D(U/L)		
6*	0110	06	12,9,6	8090	LC Lower Case		6E(U/L)		
7*	0111	07	12,9,7	8050	DEL Delete		7F(U/L)		
8	1000	08	12,9,8	8030					
9	1001	09	12,9,8,1	9030					
10	1010	0A	12,9,8,2	8830	SMM				
11	1011	0B	12,9,8,3	8430	VT				
12	1100	0C	12,9,8,4	8230	FF				
13	1101	0D	12,9,8,5	8130	CR				
14	1110	0E	12,9,8,6	80B0	SO				
15	1111	0F	12,9,8,7	8070	SI				
16	00010000	10	12,11,9,8,1	D030	DLE				
17	0001	11	11,9,1	5010	DC1				
18	0010	12	11,9,2	4810	DC2				
19	0011	13	11,9,3	4410	DC3				
20*	0100	14	11,9,4	4210	RES Restore	05 ⑥	4C(U/L)		
21*	0101	15	11,9,5	4110	NL New Line	81 ⑦	DD(U/L)		
22*	0110	16	11,9,6	4090	BS Backspace	11	5E(U/L)		
23	0111	17	11,9,7	4050	IDL Idle				
24	1000	18	11,9,8	4030	CAN				
25	1001	19	11,9,8,1	5030	EM				
26	1010	1A	11,9,8,2	4830	CC				
27	1011	1B	11,9,8,3	4430	CU1				
28	1100	1C	11,9,8,4	4230	FLS				
29	1101	1D	11,9,8,5	4130	GS				
30	1110	1E	11,9,8,6	40B0	RDS				
31	1111	1F	11,9,8,7	4070	US				
32	00100000	20	11,0,9,8,1	7030	DS				
33	0001	21	0,9,1	3010	SOS				
34	0010	22	0,9,2	2810	FS				
35	0011	23	0,9,3	2410					
36	0100	24	0,9,4	2210	BYP Bypass				
37*	0101	25	0,9,5	2110	LF Line Feed	03	3D(U/L)		
38*	0110	26	0,9,6	2090	EOB End of Block		3E(U/L)		
39	0111	27	0,9,7	2050	PRE Prefix				
40	1000	28	0,9,8	2030					
41	1001	29	0,9,8,1	3030					
42	1010	2A	0,9,8,2	2830	SM				
43	1011	2B	0,9,8,3	2430	CU2				
44	1100	2C	0,9,8,4	2230					
45	1101	2D	0,9,8,5	2130	ENQ				
46	1110	2E	0,9,8,6	20B0	ACK				
47	1111	2F	0,9,8,7	2070	BEL				
48	00110000	30	12,11,0,9,8,1	F030					
49	0001	31	9,1	1010					
50	0010	32	9,2	0810	SYN				
51	0011	33	9,3	0410					
52	0100	34	9,4	0210	PN Punch On				
53*	0101	35	9,5	0110	RS Reader Stop	09 ⑧	0D(U/L)		
54*	0110	36	9,6	0090	UC Upper Case		0E(U/L)		
55	0111	37	9,7	0050	EOT End of Trans				
56	1000	38	9,8	0030					
57	1001	39	9,8,1	1030					
58	1010	3A	9,8,2	0830					
59	1011	3B	9,8,3	0430	CU3				
60	1100	3C	9,8,4	0230	DCA				
61	1101	3D	9,8,5	0130	NAK				
62	1110	3E	9,8,6	00B0					
63	1111	3F	9,8,7	0070	SUB				

① Codes identified by * are recognized by all Monitor System conversion subroutines. Codes that are not asterisked are recognized only by the SPEED subroutine.

② U = Upper Case, L = Lower Case; ③ EBCDIC subset

④ Hexadecimal codes identified by ④ can also be entered from the console keyboard.

Console Printer Codes: ⑤ Tabulate, ⑥ Shift to Black, ⑦ Carrier Return

⑧ Shift to Red

29249

Ref. No. ①	EBCDIC Binary	EBCDIC Hex	IBM Card Code Rows	IBM Card Code Hex ④	Graphics and Control Names	Console Printer Hex	PTTC/8 Hex ②	1132 Hex ③	1403 Hex
64*	01000000	40	No Punches	0000④	blank (space)	21	10(U/L)	≠	7F
65	0001	41	12,0,9,1	B010					
66	0010	42	12,0,9,2	A810					
67	0011	43	12,0,9,3	A410					
68	0100	44	12,0,9,4	A210					
69	0101	45	12,0,9,5	A110					
70	0110	46	12,0,9,6	A090					
71	0111	47	12,0,9,7	A050					
72	1000	48	12,0,9,8	A030					
73	1001	49	12,8,1	9020					
74*	1010	4A	12,8,2	8820④	¢	02	20(U)		
75*	1011	4B	12,8,3	8420④	. (period)	00	6B(L)	4B	6E
76*	1100	4C	12,8,4	8220④	<	DE	02(U)		
77*	1101	4D	12,8,5	8120④	(	FE	19(U)	4D	57
78*	1110	4E	12,8,6	80A0④	+	DA	70(U)	4E	6D
79*	1111	4F	12,8,7	8060④	|(logical OR)	C6	3B(U)		
80*	01010000	50	12	8000④	&	44	70(L)	50	15
81	0001	51	12,11,9,1	D010					
82	0010	52	12,11,9,2	C810					
83	0011	53	12,11,9,3	C410					
84	0100	54	12,11,9,4	C210					
85	0101	55	12,11,9,5	C110					
86	0110	56	12,11,9,6	C090					
87	0111	57	12,11,9,7	C050					
88	1000	58	12,11,9,8	C030					
89	1001	59	11,8,1	5020					
90*	1010	5A	11,8,2	4820④	!	42	5B(U)		
91*	1011	5B	11,8,3	4420④	$	40	5B(L)	5B	62
92*	1100	5C	11,8,4	4220④	*	D6	08(U)	5C	23
93*	1101	5D	11,8,5	4120④	)	F6	1A(U)	5D	2F
94*	1110	5E	11,8,6	40A0④	;	D2	13(U)		
95*	1111	5F	11,8,7	4060④	¬(logical NOT)	F2	6B(U)		
96*	01100000	60	11	4000④	- (dash)	84	40(L)	60	61
97*	0001	61	0,1	3000④	/	BC	31(L)	61	4C
98	0010	62	11,0,9,2	6810					
99	0011	63	11,0,9,3	6410					
100	0100	64	11,0,9,4	6210					
101	0101	65	11,0,9,5	6110					
102	0110	66	11,0,9,6	6090					
103	0111	67	11,0,9,7	6050					
104	1000	68	11,0,9,8	6030					
105	1001	69	0,8,1	3020					
106	1010	6A	12,11	C000					
107*	1011	6B	0,8,3	2420④	, (comma)	80	3B(L)	6B	16
108*	1100	6C	0,8,4	2220④	%	06	15(U)		
109*	1101	6D	0,8,5	2120④	_ (underscore)	BE	40(U)		
110*	1110	6E	0,8,6	20A0④	>	46	07(U)		
111*	1111	6F	0,8,7	2060④	?	86	31(U)		
112	01110000	70	12,11,0	E000					
113	0001	71	12,11,0,9,1	F010					
114	0010	72	12,11,0,9,2	E810					
115	0011	73	12,11,0,9,3	E410					
116	0100	74	12,11,0,9,4	E210					
117	0101	75	12,11,0,9,5	E110					
118	0110	76	12,11,0,9,6	E090					
119	0111	77	12,11,0,9,7	E050					
120	1000	78	12,11,0,9,8	E030					
121	1001	79	8,1	1020					
122*	1010	7A	8,2	0820④	:	82	04(U)		
123*	1011	7B	8,3	0420④	#	C0	0B(L)		
124*	1100	7C	8,4	0220④	@	04	20(L)		
125*	1101	7D	8,5	0120④	' (apostrophe)	E6	16(U)	7D	0B
126*	1110	7E	8,6	00A0④	=	C2	01(U)	7E	4A
127*	1111	7F	8,7	0060④	"	E2	0B(U)		

‡ Any code other than those defined for the 1132 will be interpreted by the PRNT 1 subroutine as a blank.

29250

Ref. No. ①	EBCDIC Binary	EBCDIC Hex	IBM Card Code Rows	IBM Card Code Hex ④	Graphics and Control Names	Console Printer Hex	PTTC/8 Hex ②	1132 Hex ③	1403 Hex
128	10000000	80	12,0,8,1	B020					
129	0001	81	12,0,1	B000	a				
130	0010	82	12,0,2	A800	b				
131	0011	83	12,0,3	A400	c				
132	0100	84	12,0,4	A200	d				
133	0101	85	12,0,5	A100	e				
134	0110	86	12,0,6	A080	f				
135	0111	87	12,0,7	A040	g				
136	1000	88	12,0,8	A020	h				
137	1001	89	12,0,9	A010	i				
138	1010	8A	12,0,8,2	A820					
139	1011	8B	12,0,8,3	A420	[				
140	1100	8C	12,0,8,4	A220					
141	1101	8D	12,0,8,5	A120					
142	1110	8E	12,0,8,6	A0A0					
143	1111	8F	12,0,8,7	A060					
144	10010000	90	12,11,8,1	D020					
145	0001	91	12,11,1	D000	j				
146	0010	92	12,11,2	C800	k				
147	0011	93	12,11,3	C400	l				
148	0100	94	12,11,4	C200	m				
149	0101	95	12,11,5	C100	n				
150	0110	96	12,11,6	C080	o				
151	0111	97	12,11,7	C040	p				
152	1000	98	12,11,8	C020	q				
153	1001	99	12,11,9	C010	r				
154	1010	9A	12,11,8,2	C820					
155	1011	9B	12,11,8,3	C420	]				
156	1100	9C	12,11,8,4	C220					
157	1101	9D	12,11,8,5	C120					
158	1110	9E	12,11,8,6	C0A0					
159	1111	9F	12,11,8,7	C060					
160	10100000	A0	11,0,8,1	7020					
161	0001	A1	11,0,1	7000					
162	0010	A2	11,0,2	6800	s				
163	0011	A3	11,0,3	6400	t				
164	0100	A4	11,0,4	6200	u				
165	0101	A5	11,0,5	6100	v				
166	0110	A6	11,0,6	6080	w				
167	0111	A7	11,0,7	6040	x				
168	1000	A8	11,0,8	6020	y				
169	1001	A9	11,0,9	6010	z				
170	1010	AA	11,0,8,2	6820					
171	1011	AB	11,0,8,3	6420					
172	1100	AC	11,0,8,4	6220					
173	1101	AD	11,0,8,5	6120	[				
174	1110	AE	11,0,8,6	60A0					
175	1111	AF	11,0,8,7	6060					
176	10110000	B0	12,11,0,8,1	F020					
177	0001	B1	12,11,0,1	F000					
178	0010	B2	12,11,0,2	E800					
179	0011	B3	12,11,0,3	E400					
180	0100	B4	12,11,0,4	E200					
181	0101	B5	12,11,0,5	E100					
182	0110	B6	12,11,0,6	E080					
183	0111	B7	12,11,0,7	E040					
184	1000	B8	12,11,0,8	E020					
185	1001	B9	12,11,0,9	E010					
186	1010	BA	12,11,0,8,2	E820					
187	1011	BB	12,11,0,8,3	E420					
188	1100	BC	12,11,0,8,4	E220					
189	1101	BD	12,11,0,8,5	E120	]				
190	1110	BE	12,11,0,8,6	E0A0					
191	1111	BF	12,11,0,8,7	E060					

29251

Ref. No. ①	EBCDIC Binary	EBCDIC Hex	IBM Card Code Rows	IBM Card Code Hex ④	Graphics and Control Names	Console Printer Hex	PTTC/8 Hex ②	1132 Hex ③	1403 Hex
192	11000000	C0	12,0	A000	(+ zero)				
193*	0001	C1	12,1	9000 ④	A	3C/3E	61(U)	C1	64
194*	0010	C2	12,2	8800 ④	B	18/1A	62(U)	C2	25
195*	0011	C3	12,3	8400 ④	C	1C/1E	73(U)	C3	26
196*	0100	C4	12,4	8200 ④	D	30/32	64(U)	C4	67
197*	0101	C5	12,5	8100 ④	E	34/36	75(U)	C5	68
198*	0110	C6	12,6	8080 ④	F	10/12	76(U)	C6	29
199*	0111	C7	12,7	8040 ④	G	14/16	67(U)	C7	2A
200*	1000	C8	12,8	8020 ④	H	24/26	68(U)	C8	6B
201*	1001	C9	12,9	8010 ④	I	20/22	79(U)	C9	2C
202	1010	CA	12,0,9,8,2	A830					
203	1011	CB	12,0,9,8,3	A430					
204	1100	CC	12,0,9,8,4	A230					
205	1101	CD	12,0,9,8,5	A130					
206	1110	CE	12,0,9,8,6	A0B0					
207	1111	CF	12,0,9,8,7	A070					
208	11010000	D0	11,0	6000	(- zero)				
209*	0001	D1	11,1	5000 ④	J	7C/7E	51(U)	D1	58
210*	0010	D2	11,2	4800 ④	K	58/5A	52(U)	D2	19
211*	0011	D3	11,3	4400 ④	L	5C/5E	43(U)	D3	1A
212*	0100	D4	11,4	4200 ④	M	70/72	54(U)	D4	5B
213*	0101	D5	11,5	4100 ④	N	74/76	45(U)	D5	1C
214*	0110	D6	11,6	4080 ④	O	50/52	46(U)	D6	5D
215*	0111	D7	11,7	4040 ④	P	54/56	57(U)	D7	5E
216*	1000	D8	11,8	4020 ④	Q	64/66	58(U)	D8	1F
217*	1001	D9	11,9	4010 ④	R	60/62	49(U)	D9	20
218	1010	DA	12,11,9,8,2	C830					
219	1011	DB	12,11,9,8,3	C430					
220	1100	DC	12,11,9,8,4	C230					
221	1101	DD	12,11,9,8,5	C130					
222	1110	DE	12,11,9,8,6	C0B0					
223	1111	DF	12,11,9,8,7	C070					
224	11100000	E0	0,8,2	2820					
225	0001	E1	11,0,9,1	7010					
226*	0010	E2	0,2	2800 ④	S	98/9A	32(U)	E2	0D
227*	0011	E3	0,3	2400 ④	T	9C/9E	23(U)	E3	0E
228*	0100	E4	0,4	2200 ④	U	B0/B2	34(U)	E4	4F
229*	0101	E5	0,5	2100 ④	V	B4/B6	25(U)	E5	10
230*	0110	E6	0,6	2080 ④	W	90/92	26(U)	E6	51
231*	0111	E7	0,7	2040 ④	X	94/96	37(U)	E7	52
232*	1000	E8	0,8	2020 ④	Y	A4/A6	38(U)	E8	13
233*	1001	E9	0,9	2010 ④	Z	A0/A2	29(U)	E9	54
234	1010	EA	11,0,9,8,2	6830					
235	1011	EB	11,0,9,8,3	6430					
236	1100	EC	11,0,9,8,4	6230					
237	1101	ED	11,0,9,8,5	6130					
238	1110	EE	11,0,9,8,6	60B0					
239	1111	EF	11,0,9,8,7	6070					
240*	11110000	F0	0	2000 ④	0	C4	1A(L)	F0	49
241*	0001	F1	1	1000 ④	1	FC	01(L)	F1	40
242*	0010	F2	2	0800 ④	2	D8	02(L)	F2	01
243*	0011	F3	3	0400 ④	3	DC	13(L)	F3	02
244*	0100	F4	4	0200 ④	4	F0	04(L)	F4	43
245*	0101	F5	5	0100 ④	5	F4	15(L)	F5	04
246*	0110	F6	6	0080 ④	6	D0	16(L)	F6	45
247*	0111	F7	7	0040 ④	7	D4	07(L)	F7	46
248*	1000	F8	8	0020 ④	8	E4	08(L)	F8	07
249*	1001	F9	9	0010 ④	9	E0	19(L)	F9	08
250	1010	FA	12,11,0,9,8,2	E830					
251	1011	FB	12,11,0,9,8,3	E430					
252	1100	FC	12,11,0,9,8,4	E230					
253	1101	FD	12,11,0,9,8,5	E130					
254	1110	FE	12,11,0,9,8,6	E0B0					
255	1111	FF	12,11,0,9,8,7	E070					

29252

ACC 4
Accumulator 4
Accumulator Extension 4
Add 10
Add Double 10
Addressing 2
AFR 4
AND 12
Arithmetic Factor Register 4
Arithmetic Instructions 10

Binary Synchronous Communications 68
  Control Characters 75
  Control Sequences 75
  Operation 74
Branch and Store IAR 16
Branch Instructions 15
Branch or Skip on Condition 15

Carry Indicator 5
CCC 5
Central Processing Unit 2
CPU 2
  Arithmetic Instructions 10
    Add 10
    Add Double 10
    AND 12
    Divide 11
    Exclusive OR 12
    Multiply 11
    OR 12
    Subtract 11
    Subtract Double 11
  Branch Instructions 15
    Branch and Store IAR 16
    Branch or Skip on Condition 15
    Modify Index and Skip 16
  Execute I/O 18
  Load and Store 8
    Load ACC 8
    Load Double 9
    Load Index 9
    Load Status 10
    Store Accumulator 9
    Store Double 9
    Store Index 10
    Store Status 10
  Shift Instructions 13
    Rotate Right ACC and EXT 15
    Shift Left ACC 13
    Shift Left ACC and EXT 13
    Shift Left and Count ACC 13
    Shift Left and Count ACC and EXT 14
    Shift Right ACC 14
    Shift Right ACC and EXT 14
  Controls 5
  Core Storage 2
  Data Flow 5
  Wait 17

CPU Usage Meter 26
Chad Paper Tape 40
Communication Data Flow 70
Console Entry Switches 25
  CES/Read 26
  Keyboard Interrupt 26
Console Printer 23
  Data Coding 23
  Programming 23
  I/O Control Commands 23
    Sense Device 23
    Write 23
  Console/Keyboard DSW Indicators 24
    Busy 24
    Not Ready 24
    Service Response Interrupt 24
    Interrupt Request Interrupt 25
    Keyboard Response Interrupt 25
Control Operation -- Binary Synchronous 74
Control Operations -- STR 71
Customer Engineering Meter 26
Cycle Control Counter 5
Cycle-Steal
  Devices 82
  Mode 65
  Operation 21
  Priority 65,81
Cycle-Stealing Concept 80

Data Format, CPU 3
Data Transmission Timing 79
Data Sheet Terminology, 1231 51
Data Transmission -- Binary Synchronous 76
Data Word Format
  Double Precision 3
  Single Precision 3
Device Code 18
Device Priority 81
Device Status Word 21
Direct Addressing 6
Direct Program Control Via Interrupt 80
Disk Storage Drives 27
  Capacity 28
  Cylinder Schematic 28
  Data Organization 28
  Device Status Word 31
  Disk Access Mechanism 27,28
  Disk Cartridge 27
  Disk Check Bit Chart 29
  Data Checking 29
  Data Organization 27
  DSW Indicators 31
    Data Error 31
    Disk Busy R/W or Carriage 32
    Disk Not Ready 32
    Carriage Home 32
    Operation Complete 32
    Sector Count 32
  I/O Control Commands 30
    Control 31

I/O Control Commands (Continued)
Initiate Read 30
Initiate Write 30
Read 30
Read-Check 30
Sense Device 31
Programming Considerations 32
Sector Numbers 28
Timing 28
Displacement 3
Divide 11
Double Precision Word 3
Double Precision Data Word Format 3

Effective Address Determination 6
Effective Address Generation 6
Error-Correction Routines 32
Exclusive OR 12
Execute I/O 12
Exposure to Loss of Data 80
EXT 4

F Bit 3
Fiber-Optics Light Pen 59
4-8 Line Transmission Code 69
Four-Wire Operation 69
Function Code 18

Half-Duplex Operation 69

IA Bit 4
IAR 5,6
Index Registers 4,6
Indirect Addressing 7
Input/Output Control Command 18
I/O Timing Requirements 82
Instruction Address Register 5
Instruction Formats 3
Interrupt 19
Identification 21
Level Status Word -- Level 1 20
Level Status Word -- Level 2 20
Level Status Word -- Level 3 20
Level Status Word -- Level 4 20
Level Status Word -- Level 5 21
Mode 65
Philosophy 19
Priority 81
Recognition Procedure 22
Run Mode 25
Vector Location 19
IOCC 18
Address 18
Device 18
Function 18
Control 18
Initiate Read 18
Initiate Write 18
Read 18
Sense Device 19
Sense Interrupt 18
Write 18
Keyboard 24
Data Format 24

Keyboard (Continued)
Device Status Word 24
DSW Indicators 25
Entry 26
Programming 24
I/O Control Commands 24
Control 24
Read 24
Sense Device 24

Light Pen Switch Status 63
Line Attachment 68
Line Turnaround 75
Load ACC 8
Load and Store 8
Load Double 9
Load Index 9
Load Mode Read 34
Load Status 10
Long Instruction 6
Long Instruction Format 3

Machine Registers 4
Marking the 1231 Data Sheet 51
Modifier 19
Modifier Bits 4
Modify Index and Skip 16
Modulo 4 29
Multi-Point Operation 76
Multiply 11

Negative Number 3
Non-Reflective Ink 51
Normal Mode Read 34

OP 5
OP Code 3
Operation Register 5
OR 12
Overflow Indicator 6
Overlapping Input/Output Operations 80

Plotter Command Codes 50
Printers 43
Program Control Devices 83
Program Load from 1134 41
Program Load from 1442 35
Program Operation, Interrupt 20
Programmed Function Keyboard 59
Programming Single Disk Storage 30
Program Stop Key 25
Punched Card Input/Output Devices 33

Read-Check, Disk 30
Receive Mode -- STR 73
Receive Mode -- Binary Synchronous 76
Registers 4
Index Registers 4
Machine Registers 4
Reflective Ink, 1231 51
Regeneration, 2250 59
Reserved Core Storage 2
Rotate Right ACC and EXT 15

SAC/SAC II 65
I/O Control Commands 66
Control 66
Initiate Read 66
Initiate Write 67
Read 66
Sense Device 66
Sense Interrupt 66
Write 66
Programming 65
Special Power Sequencing Considerations 67
Sample Interrupt Recognition Procedure 22
SAR 4
SBR 4
Service Request Limitations 81
Shift Instructions 13
Shift Left ACC 13
Shift Left ACC and EXT 13
Shift Left and Count ACC 13
Shift Left and Count ACC and EXT 14
Shift Right ACC 14
Shift Right ACC and EXT 14
Short Instruction 6
Short Instruction Format 3
Single Precision Word 3
Special FORTRAN Characters 43
Special Keyboard Console Programming 25
Special Programming -- STR 74
Storage Access Channel 65
Storage Address Register 4
Storage Buffer Register 4
Store Accumulator 9
Store Double 9
Store Index 10
Store Status 10
STR 68
Control Characters 71
Control Sequences 72
Operation 71
Subtract 11
Subtract Double 11
Synchronize Mode -- Binary Synchronous 76
Synchronize Mode -- STR 73
Synchronous Communications Adapter 68
Bit Transfer Sequence 70
Device Status Word 77
I/O Control Commands 78
Control 78
Initiate Read 79
Initiate Write 79
Read 78
Sense Device 79
Write 78
Timers 70
Timing for SCA Programming 79
Synchronous Transmit-Receive (See STR) 68
Sync/Idle Register 76

T Bits 3
Tag 5
Tag Register 5
Tape Character Code 40
Tape Specifications 40
TAR 4
Temporary Accumulator 4
Timing for SCA Programming 79
Transmit Mode -- Binary Synchronous 77
Transmit Mode -- STR 73
Two-Wire Operation 69

Usage Meter
1131 26
1132 45
1133 63
1231 58
1403 48
1442 37
2250 64
2310 32
2501 39

Wait 17
Wraparound Addressing 2

XIO 18

2's Complement 3

1055 Paper Tape Punch 40

1130 Instruction Set 8

1131 Central Processing Unit 2
1131 CPU Usage Meter 26
1131 Single Disk Storage 27

1132 Printer 43
Data Format 43
Device Status Word 44
DSW Indicators 44
Carriage Busy 45
Carriage Control Channels 45
Not Ready 45
Print Scan Check 45
Read Emitter Response Interrupt 44
Space Response Interrupt 44
Skip Response Interrupt 44
Forms Control 43
I/O Control Commands 43
Control 44
Read Emitter 44
Sense Device 44
Space 44
Start Carriage 44
Start Printer 44
Stop Carriage 44
Stop Printer 43
Programming 43
Programming Notes 45
Usage Meter 45

1133 Usage Meter 63

1134 Paper Tape Reader 40
1134/1055 Programming 41
Paper Tape DSW Indicators 41

1134/1055 Programming (Continued)
Punch Busy 42
Punch Not Ready 42
Punch Response Interrupt 42
Reader Response Interrupt 41
Reader Not Ready 42
Reader Busy 42
Paper Tape I/O Control Commands 41
Control 41
Read 41
Sense Device 41
Write 41
Paper Tape Device Status Word 42
Paper Tape Input/Output Devices 40
Paper Tape/Core Storage Format 42

1231 Optical Mark Page Reader 51
Alphabetic Coding Schemes 55
Data Flow 53
Data Format 41
Data Sheet 51
Device Status Word 57
DSW Indicators 56
Busy 57
Document Selected 58
Feed Busy 57
Hopper Empty 58
Master Mark 57
Not Ready 57
Okay to Select 57
Operation Complete 57
Operation Complete Interrupt 56
Read Busy 57
Read Error 58
Read Error Interrupt 57
Read Response 57
Read Request Interrupt 56
Test Timing Mark Check 57
Timing Mark Error Interrupt 57
Field Checking 54
Mark Positions 51
Mark Recognition 52
Master Mark 51
Message Format 52
Program Control Sheet 55
Programming 55
Segments 51
System Programming 56
Control 56
I/O Disconnect 56
Read 56
Read and Start 56
Select Stacker 56
Sense Device 56
Timing Mark 51
Usage Meter 58
Word 51

1403 Printer 45
Control Tape 46
Data Format 46
Device Status Word 47
DSW Indicators 47
Carriage Busy 47
Carriage Channel 9 47
Carriage Channel 12 47
Carriage Interrupt 47
Not Ready 48

1403 Printer (Continued)
Parity Check 47
Print Buffer 47
Print Check 47
Printer Busy 48
Print Complete Interrupt 47
Sync Check 47
Transfer Complete Interrupt 47
Maximum Printing Speed 45
Programming 46
Control 47
Initiate Write 46
Sense Device 47
Write 47
Spacing and Skipping 46
Usage Meter 48

1442 Card Read Punch 33
Card Feeding 34
Card Punching 34
Card Reading 34
Data Coding 33
Device Status Word 36
DSW Indicators 36
Busy 36
Error Check 36
Last Card 36
Not Ready 36
Operation Complete Interrupt 36
Punch Response Interrupt 36
Read Response Interrupt 36
I/O Control Commands 35
Control 35
Feed Cycle 35
Read 35
Sense Device 35
Start Punch 35
Start Read 35
Stacker Select 35
Write 35
Last Card Sequence 35
Program Load 35
Programming 35
Programming Note 36
Speeds 33
Usage Meter 37

1627 Plotter 49
Device Status Word 50
Drum 49
DSW Indicators 50
Busy 50
Not Ready 50
Plotter Response Interrupt 50
I/O Control Command 50
Sense Device 50
Write 50
Operation Characteristics 49
Pen 49
Pen and Carriage 49
Programming 50

2250 Display Unit 59
Alphameric Keyboard 59
Channel Interface Section 60
Character Mode 60
Device Status Word 63
Displays 60

2250 Display Unit (Continued)
Graphic Mode 60
Input/Output Control Commands 61
Control 62
Initiate Read 61
Initiate Write 61
Sense Device 62
Set Programmed Functions Indicators Command 61
Start Regeneration Command 61
Reset Display Command 62
Programming 61
Usage Meter 64

2310 Disk Storage (See Also Disk Storage Drives) 29
Usage Meter 32

2315 Disk Cartridge Assembly 27

2501 Card Reader 37
Card Feeding 37
Card Reading 37
Card Throughput 39
Data Coding 37
Device Status Word 38
DSW Indicators 38
Busy 38
Error Check 39
Last Card 38
Not Ready or Busy 38
Operation Complete Interrupt 38
I/O Control Commands 38
Initiate Read 38
Sense Device 38
Last Card Sequence 38
Programming 38
Program Load 37
Reader and System Timing 39
Timing Schematic 39
Usage Meter 39

Printed in U. S. A. A26-5881-4

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

READER'S COMMENT FORM

IBM 1130 Functional Characteristics

Form A26-5881-4

- How did you use this publication?

  As a reference source ☐
  As a classroom text ☐
  As..... ☐

- Based on your own experience, rate this publication...

As a reference source:	..... Very Good	..... Good	..... Fair	..... Poor	..... Very Poor
As a text:	..... Very Good	..... Good	..... Fair	..... Poor	..... Very Poor

- What is your occupation? .....

- We would appreciate your other comments; please give specific page and line references where appropriate. If you wish a reply, be sure to include your name and address.

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS PLEASE . . .

This SRL bulletin is one of a series which serves as a reference source for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

fold fold

FIRST CLASS
PERMIT NO. 2078
SAN JOSE, CALIF.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY . . .

IBM Corporation
Monterey & Cottle Rds.
San Jose, California
95114

Attention: Product Publications, Dept. 455

fold fold

IBM 1130 Printed in U.S.A. A26-5881-4

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]