IBM

Field Engineering
Theory of Operation
(Manual of Instruction)

1130 Computing System

227-5978-2

IBM®

Field Engineering
Theory of Operation
(Manual of Instruction)

Issued to: _____

Branch Office: _____

Department: _____

Address: _____

If this manual is misplaced, it should
be returned to the above address.

1130 Computing System

PREFACE

The IBM Theory of Operation (Manual of Instruction) 1130 Computing System contains information on the 1131 Central Processing Unit and Console, and the attachment circuits for the Input/Output units: Console Printer, Console Keyboard, and Disk File Storage. Descriptions of the attachment circuits for the IBM 1442 Card Read-Punch, IBM 1132 Printer, IBM 1134 Paper Tape Reader, IBM 1055 Paper Tape Punch and IBM 1627 Plotter, are found in the Field Engineering Manual of Instruction 1130 Computing System-Features.

Information regarding the Input/Output units may be found in the manual of instruction for each unit as listed in the Bibliography, Appendix A.

The users of this manual are cautioned that specifications are subject to change at any time and without prior notice by IBM. Wiring diagrams (logics) at the engineering change level of that specific machine are included in each machine shipment.

This manual form 227-5978-2 is a major revision of form 227-5978-1. The latter is made obsolete by this revision.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.
Comments concerning the contents of this publication may be addressed to:
IBM, Product Publications Department, San Jose, Calif. 95114

CONTENTS

DESCRIPTION OF SYSTEM

- The IBM 1130 is a solid state system composed of a Central Processing Unit (including core storage and disk storage) and I/O Units.
- The Central Processing Unit (CPU) has Console Keyboard and Console Printer.
- Core storage can contain 4,096 or 8,192 words.
- Each 16-bit core storage word can be addressed.
- Alphameric and special characters may be stored.
- The program is stored in core storage.
- Program instructions may be sequential or non-sequential.
- 1130 is an add-to-accumulator computer.
- Addition, subtraction, multiplication and division are accomplished by a special adder circuit.

The IBM 1130 Computing System is a solid-state electronic computer system, designed specifically for technical applications. The IBM 1134 Paper Tape Reader, the IBM 1055 Paper Tape Punch, the IBM 1627 Plotter, the IBM 1132 Printer, and the IBM 1442 Card Read Punch are input-output units available for various customer applications (Figure 1-1). Information on these input-output units is included in the IBM Customer Engineering Manuals of Instruction for each unit.

The IBM 1131 Central Processing Unit contains the logic circuitry, the console panel, the console keyboard, console printer, and the disk storage drive (Models 2A and 2B only). Figure 1-2 shows the data flow of the system.

Data and instructions entered into the system are placed in core storage as 16-bit binary words. Each word in core storage can be addressed individually by a 4-digit decimal address and can store one or more characters of information. Core storage addresses extend from 0000 to 4095 for 4,096 words of core storage and from 0000 to 8191 for 8,192 words of core storage. This addressing system enables selection of any word or group of words within core storage. The system processes binary information only. All alphabetic, numerical and special characters are coded and decoded by programming for input/output operations. For example, in order to print information on the 1132 Printer, data must be coded in extended binary coded decimal coding.

The system is capable of performing 24 basic operations. Each operation is specified by an instruction which can be one or two words long. The instruction contains an operation code, data which generates an effective address, and/or modifying bits. Instructions comprising a program are normally stored in consecutive locations in core storage and executed sequentially. However, the sequence of operations may be altered at any point in the program by conditional skip and branch instructions. Conditional skip and branch instructions make logical decisions by performing tests on indicators set by the computer or the program.

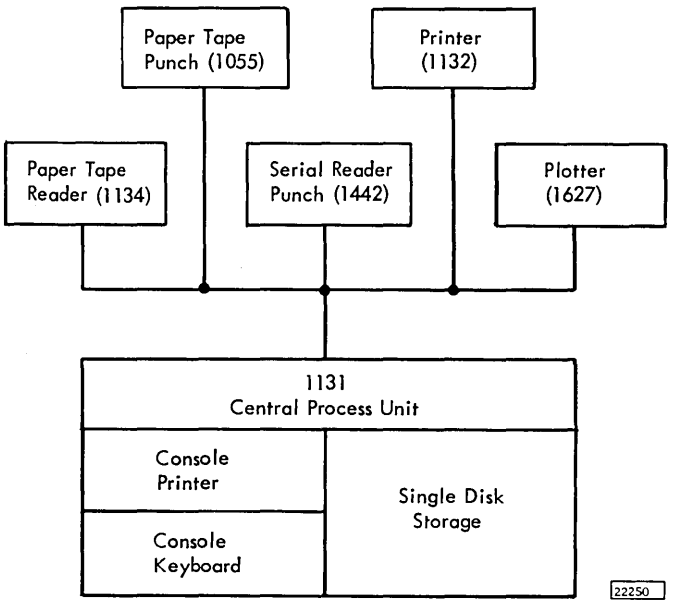


Figure 1-1. 1130 System Configurator

* NOTE: 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.

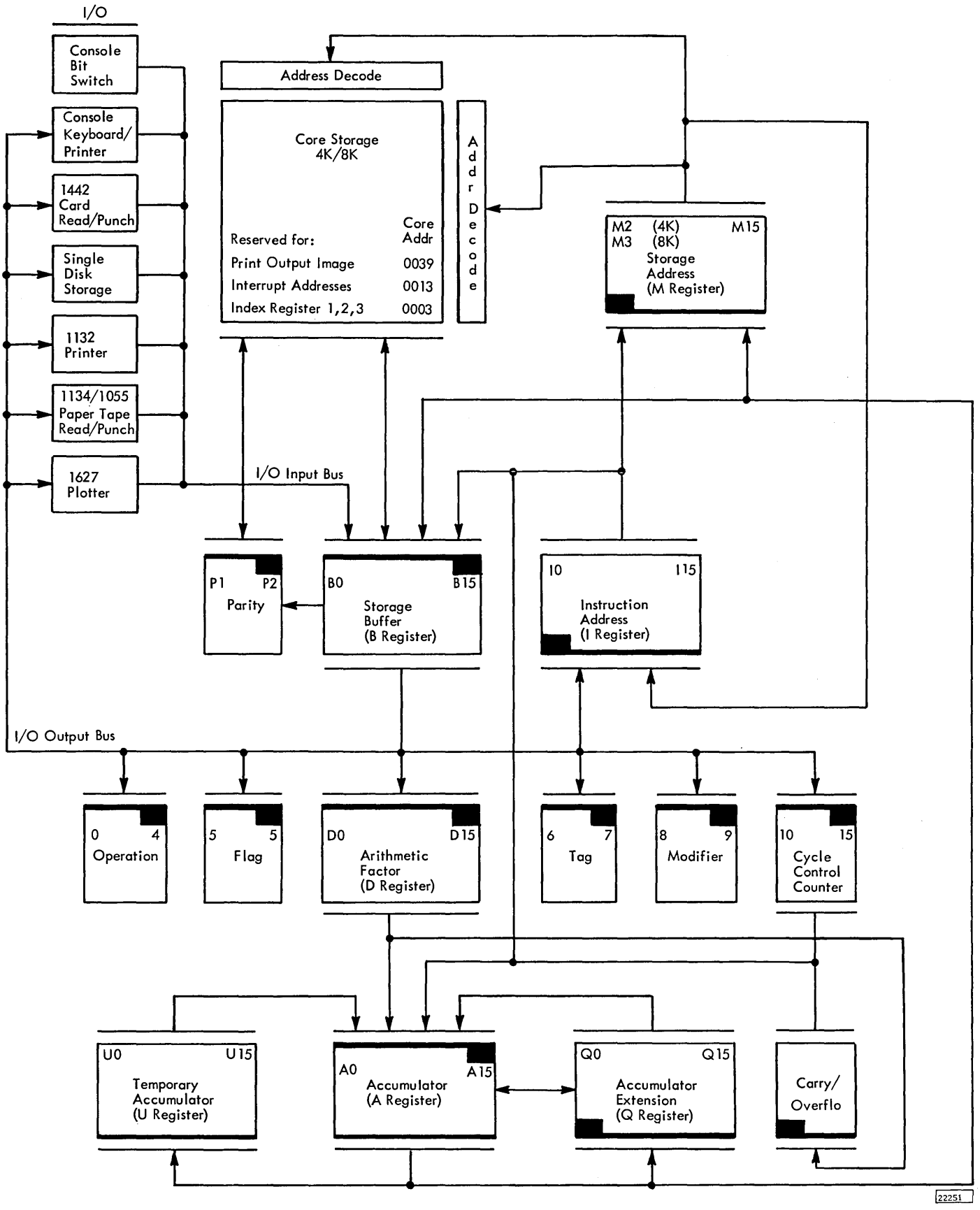


Figure 1-2. 1130 System Data Flow

The 1130 is an add-to-accumulator computer. That is, a word of core storage is added to the contents of an accumulator to develop a result.

Addition, subtraction, multiplication, and division operations are accomplished by the binary logic adder circuits of the accumulator.

The 1130 console contains control keys, switches, an indicator panel, a console keyboard and a console printer. The control keys and switches are used by the operator to communicate with and control operation of the system. The indicator panel provides a visual indication of the contents of various registers and the status of control circuitry within the computer. The console keyboard and console bit switches are used for manual entry of data and instructions into core storage. The console printer prints serially at the rate of 15.5 characters per second.

Information is entered into the system by the console bit switches, the console keyboard, the IBM 1134 Paper Tape Reader, or the IBM 1442 Card Read Punch. Output data is recorded by the console printer, the IBM 1442 Card Read Punch, the IBM 1627 Plotter, IBM 1055 Paper Tape Punch, or the IBM 1132 Printer. Information also may be entered into or removed from the system by way of the replaceable IBM 2315 Disk Cartridge.

When the computer is reading from, or writing on, and input/output device, each character received from, or sent to, the input-output device is represented in core storage as a single 16-bit word.

1.1 MACHINE LANGUAGE

1.1.1 Character Code

- Data is stored, transferred, and processed in binary form.
- Each word is made up of a combination of binary bits.
- Each core storage word has 18 bit positions: Bit 0 through bit 15 and two parity bits.
- Parity (P) bits are used to maintain an odd number of bits in each half word in core storage and the B register. Parity bit one maintains odd parity for bits 0-7, parity bit two for bits 8-15.

All data stored, transferred, or processed with the 1130 is represented in binary form. A number is represented by a particular combination of bits. The bit positions of each number consist of 16 data (B0 through B15) bits, and two parity (P) bits Figure 1-3). The decimal value of a word is the sum represented by the bits present in the data bit positions. A zero is represented by the two P bits. The digit 6 is represented by a bit 13 and a bit 14 plus both P bits; the digit 257 is represented by a bit 7 and a bit 15, no P bits (Figure 1-4).

The P bits are used for parity checking purposes. Each half word within core storage must consist of an odd total number of bits, or a parity error will be indicated. The P bit is present in a digit when the number of bits present in the numerical bit positions consists of an even number of bits.

1.1.2 Instruction Format

- Instructions contain one or two 16-bit words.
- The operation code specifies what is to be done.
- The effective address specifies the word to be processed.
- Single word instructions have: A 5 bit operation code and 11 controlling bits which form an effective core address or which control the operation. Figure 1-5.
- Two word instructions have: A 5 bit operation code, a 16 bit core address, and modifier bits which provide specific control to the operation.

The operation code consists of 5 binary bits which specify the operation to be performed. Figure 1-6 is a chart of basic operation codes and their associated mnemonics.

The functions of the modifier parts of an instruction depend upon the particular operation to be performed. The address part of an instruction represents the core storage address of a single or double word, or of an instruction, or of another address. It may also contain special information

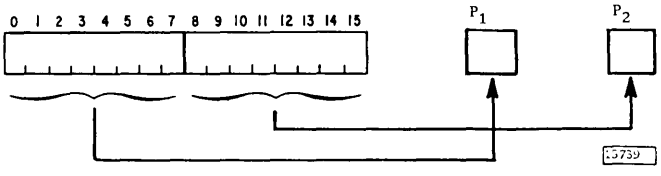


Figure 1-3. Bit Positions

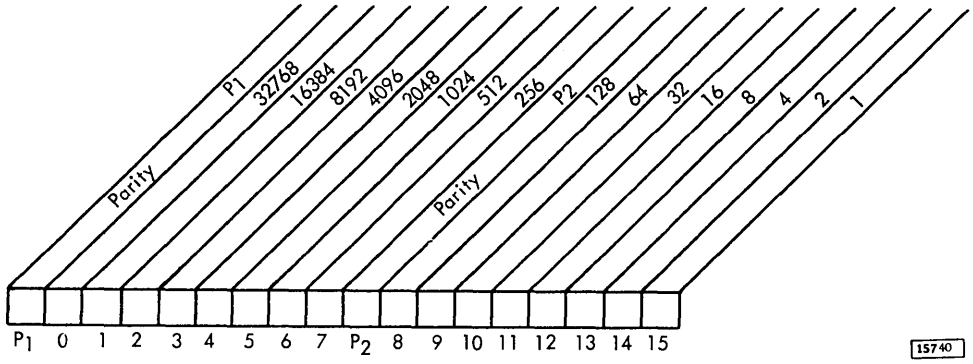


Figure 1-4. Bit Configuration - Numeric

required by the operation such as how far to shift or on what condition to branch or skip. The specific use of the address and modifier parts of the instruction are described with each operation.

1.1.3 Data Format

- Data may be arranged in core storage to form data tables.
- A data table is a number of consecutive words related to input/output operations.
- Data tables are normally processed, starting with the high-order word (low numbered core storage position).

Data tables can be of any length from one core storage position to the maximum number of positions in core storage. Data in core storage can be defined as words or tables.

A word occupies only one core storage position and has its own specific address.

A table in core storage consists of one or more words of data related to input/output operations. On output operations, a table in core storage is addressed at the position which occupies the lowest-numbered core storage position of the table. Tables are processed serially into successively higher core storage positions. Input operations enter data into core storage starting at the addressed position and continuing into successively higher core storage locations until terminated by the program or by an "end of transmission" signal from the device.

1.2 STORED PROGRAM CONCEPT

- A program in a series of instructions that cause the system to perform a function.
- The working instructions are stored in core storage.
- Instructions are read from core storage into registers for decoding to control the system operation.
- During the time that a program is being executed additional instructions and data may be entered into core storage from cards, disk storage, or paper tape.

To solve a problem or to process data, a programmer selects, from the instructions which the system is capable of performing, those instructions which are required to accomplish the desired results. The series of instructions, which designate the operations to be performed and the sequence in which they are to be performed, is called a program. Because the instructions comprising a program are written into core storage from an input device and read from core storage for interpretation and execution, the 1130 is called a stored program computer.

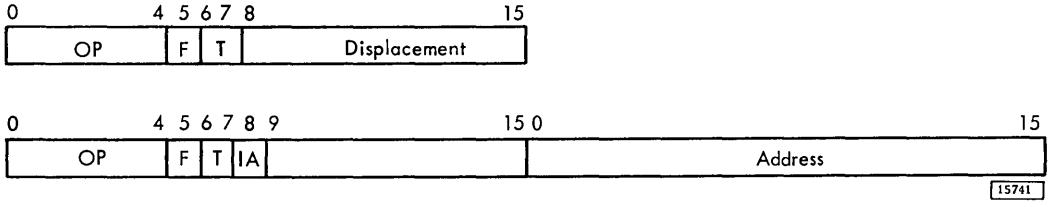


Figure 1-5. Instruction Formats

Binary	Mnemonic	
00000*	WAIT	Wait
00001	XIO	Execute I/O Instruction
00010	SLA	Shift Left A
00010	SLT	Shift Left A and Q
00010	SLCA	Shift Left and Count A
00010	SLC	Shift Left and Count A and Q
00011	RTE	Rotate Right A and Q
00011	SRA	Shift Right A
00011	SRT	Shift Right A and Q
00100	LDS	Load Status
00101	STS	Store Status
01000	BSI	Branch and Store Instruction Counter
01001	BSC	Branch or Skip On Condition
01100	LDX	Load Index
01101	STX	Store Index
01110	MDX	Modify Index and Skip
10000	A	Add
10001	AD	Double Add
10010	S	Subtract
10011	SD	Double Subtract
10100	M	Multiply
10101	D	Divide
11000	LD	Load Accumulator
11001	LDD	Double Load
11010	STO	Store Accumulator
11011	STD	Double Store
11100	AND	Logical AND
11101	OR	Logical OR
11110	EOR	Logical Exclusive OR

* All unassigned Operation Codes are defined as Wait Operations.

L5742

Figure 1-6. 1130 Operation Codes

For interpretation by the computer, an instruction must be read from core storage into registers.

Instructions within a program are normally interpreted and executed sequentially; that is, execution of the first instruction is followed by interpretation and execution of the second instruction, etc. (Figure 1-7). However, this sequence can be altered by the use of branch instructions that direct the computer to an instruction location at a location other than the next sequential core storage location.

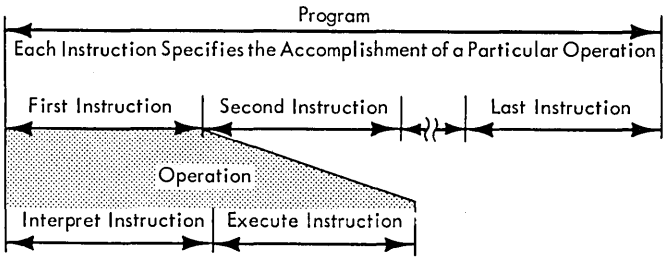


Figure 1-7. Program

While a program is being executed, additional instructions and/or data may be read into core storage from an input device or read out of core storage into an output device.

The only distinction between instructions and data in core storage is the manner in which they are interpreted by the computer. If data is placed in core storage locations assigned to an instruction, the data will be acted upon as if it were an instruction. Conversely, the operation code or address part of an instruction may be modified by treating the instruction as data.

1.2.1 Program Location

Program instructions and data may be located anywhere in core storage. However, some blocks of core storage have assignments and should not be used for other uses unless they are not used for the assigned purpose. As shown in Figure 1-8, locations 0001 to 0003 are used as index registers, 0008-0013 for interrupt level address storage, and 0032-0039 for 1132 Printer print control.

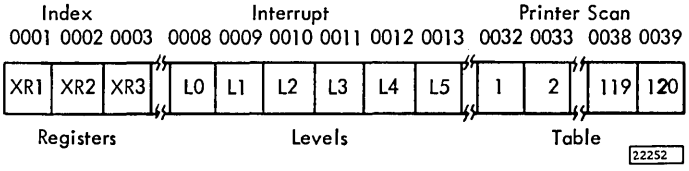


Figure 1-8. Core Storage Location Assignments

DATA FLOW AND CORE STORAGE ADDRESSING

1.3 DATA FLOW

- Data is normally processed serially by word, parallel by bit.
- Instructions are executed by a series of machine cycles.
- Each machine cycle reads and writes an addressed core storage position.

The IBM 1130 System normally processes data serially by word, parallel by bit. The 18 bits which are read from core storage for each address are processed simultaneously under most conditions, but the bits can be processed individually in some operations.

Data is confined to core storage, the storage buffer (B) register, the arithmetic factor (D) register, the accumulator (A) register, the temporary accumulator (U) register, the accumulator extension (Q) register, and the input/output areas. Digits which are presented to other registers by the B register are used to develop core storage (CS) addresses or to control program execution (Figure 1-2).

Each instruction of a program is interpreted and executed by a series of machine cycles. A machine cycle consists essentially of: addressing core storage, reading out of a core storage location, and writing into a core storage location.

Digits read from core storage, that are used for addressing or control purposes, are set into registers from the B register within the cycle or are retained in the B register until the next cycle. Digits are presented to an output device from the B register or received from an input device through the B register within the machine cycle. Figure 1-9.

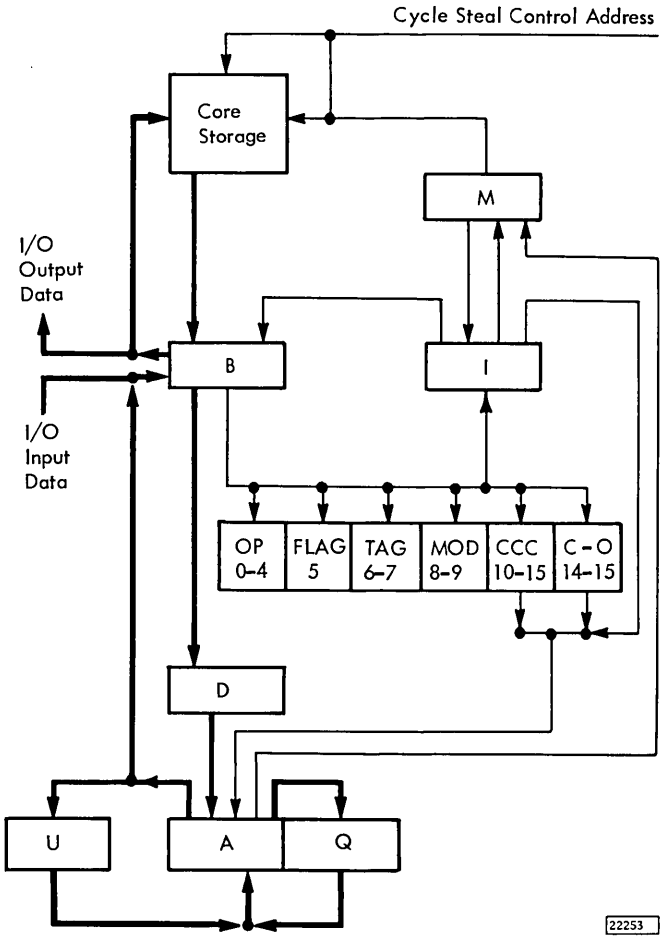


Figure 1-9. Data Flow

1.4 CORE STORAGE

- All core storage is housed within the IBM 1131 Central Processing Unit.
- 4,096 or 8,192 words of core storage are available.
- One 18-bit word is read on each machine cycle.

Core storage is addressed by means of two lines which select a single "column" of 18 cores (one core in the same position in each plane). Because each word in core storage consists of 18 bits, selecting one column of 18 cores causes a full word to be read out each time core storage is addressed (Figure 1-10).

The core storage address is in the form of a 13 bit binary coded word. This address word is set into the storage address (M) register at the beginning of each machine cycle in which a new word is required from core.

With the installation of additional core storage, the addressing scheme is expanded to include circuitry which selects addresses greater than 4096.

1.5 CORE STORAGE READ AND WRITE

- Words (16 bits) are read from core storage into the storage buffer (B) register.
- P bit flip-flop's (F F's) are not part of the B register.

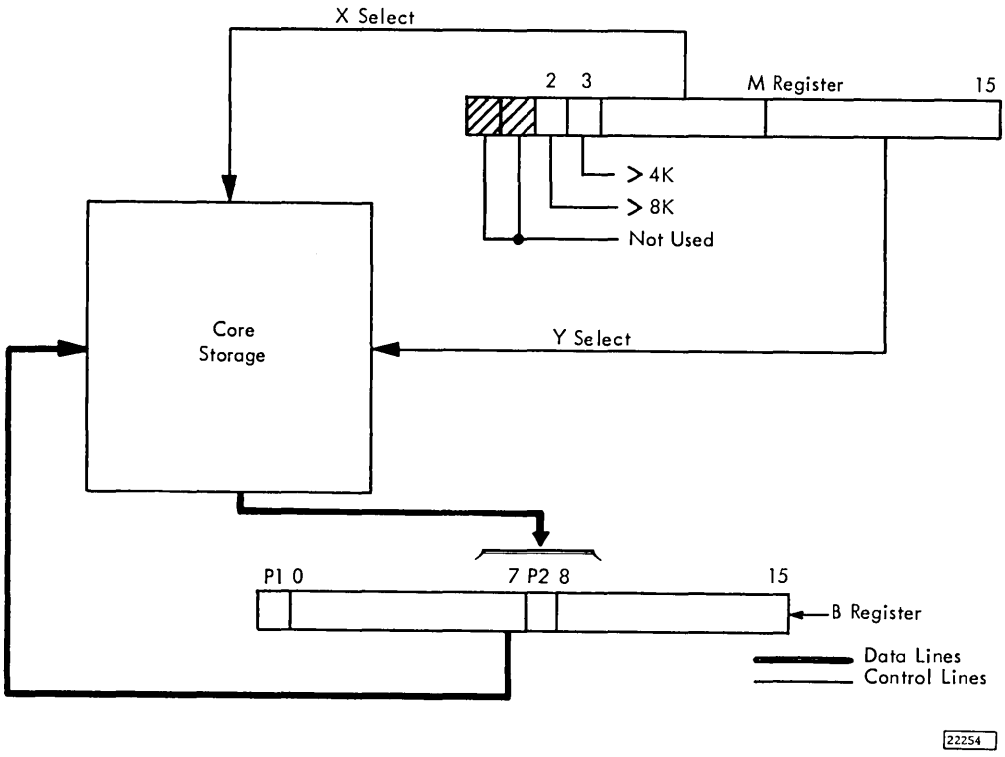


Figure 1-10. Addressing Core Storage

- Words are written into core storage from the B register and the P bit F F's.
- Read-out of core storage is destructive, therefore regeneration is required.

The 18 bits which are read out of memory are sensed by sense amplifiers which in turn set FF's in the B register and the P FF's. The B register is a 16-position register with one FF for each position in the word (Figure 1-11).

Read out of cores is destructive; therefore, to retain the word it must be written back into the cores from which it was read. This is accomplished by retaining the word in the B register. Whatever is in the B register at the end of a machine cycle has automatically been written into core storage.

1.5.1 Reading Out of Core Storage

When core storage is addressed, the B register F F's are turned off, and the core storage output is sensed by the sense amplifiers. The output of each sense amplifier is associated with a F F in the B register. Therefore, each sense amplifier that senses an output from core storage turns on its associated B register FF. Each sense amplifier that does not sense an output from core storage does not affect its associated B register FF.

Parity (odd number of bits) of the word in the B register is checked during each machine cycle after the register is set from the sense amplifiers.

1.5.2 Writing Into Core Storage

The word that is in the B register at the end of the machine cycle is automatically set into core storage at the location specified by the M register.

When a new word is to replace a word previously read out of core storage, the old word is destroyed as the new word is set into the B register.

1.6 DATA DISTRIBUTION

After the 18-bit word is read from core into the B register and P bit FF's, it is distributed to various control registers or the data (D) register as dictated by the operation code and machine cycle being used.

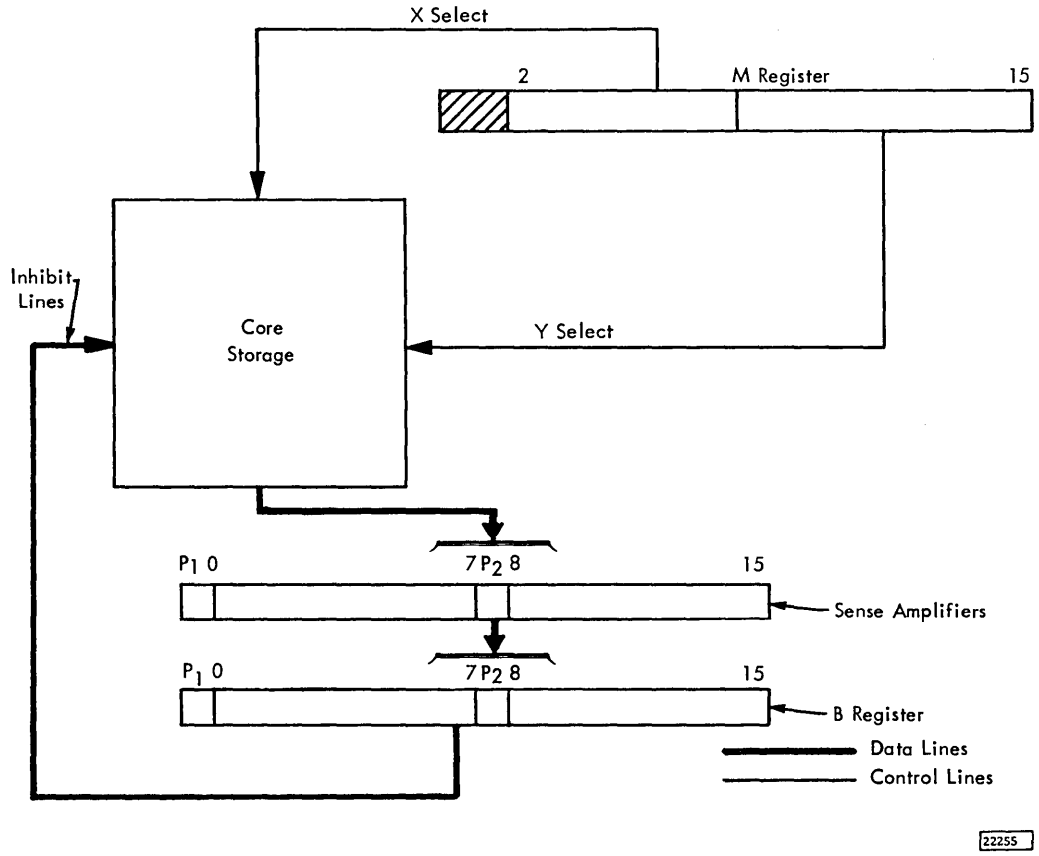


Figure 1-11. Reading/Writing Core Storage

1.7 CONSOLE FUNCTION

- The console is an integral part of the IBM 1311 Central Processing Unit.
- The console consists of the input keyboard, the output printer, and the display panel.

The input keyboard is used to enter data and instructions into core storage. Functions of the control switches and lights are described in this section. Functions of the data input keys are described in Chapter 3 Input/Output Operations.

The output printer is used to print out information that is in core storage. Functions of the printer attachment are described in Chapter 3 Input/Output Operations, 3.36 Console Printer.

1.8 CONSOLE DISPLAY PANEL

- The contents of the registers within the computer are displayed on the console panel (Figure 1-12).
- Each bit in each register position is represented by a light.
- The light is on when the bit which it represents is present in the word displayed.
- The mode switch selects the operating mode of the system.

1.8.1 Indicator Displays

Instruction (I) Address Register: Consists of one row of 14 lamps. Each lamp displays the status of one flip-flop (FF) in the I register.

Storage (M) Address Register: Consists of one row of 14 lamps. Each lamp displays the status of one FF in the M register.

Storage Buffer (B) Register: Consists of one row of 16 lamps. Each lamp indicates the status of one FF in the B register. At the end of each machine cycle the B register reflects the bits that were read into core storage on the cycle just completed.

Arithmetic Factor Register (D): Consists of one row of 16 lamps. Each lamp indicates the status of one FF in the D register.

Accumulator (A) Register: Consists of one row of 16 lamps. Each lamp indicates the status of one FF in the A register.

Accumulator Extension (Q) Register: Consists of one row of 16 lamps. Each lamp indicates the status of one FF in the Q register.

Clock Timer (T): Consists of one row of 8 lamps. These lamps indicate the last clock step completed.

Machine Cycle (I and E): Consists of one row of 7 lamps. These lamps indicate the type of machine cycle in process when in single step mode. They indicate the machine cycle just completed when in any other mode.

X7: Indicates when X7 clock is active.

Parity (P_1, P_2): Consists of two lamps. P_1 indicates parity for B register bits 0-7. P_2 indicates parity for B register bits 8-15.

Control Flip-Flop's: Consists of two rows of 7 lamps. These lamps indicate the status of various control FF's: add, AC-arith control, SC-shift control, AS-accumulator sign, TC-temporary carry, ZR-zero remainder, and W-wait operation.

CE Lights: Consists of two rows of 6 lamps. Each lamp can be wired by a CE to give a visual indication of any status condition in the machine.

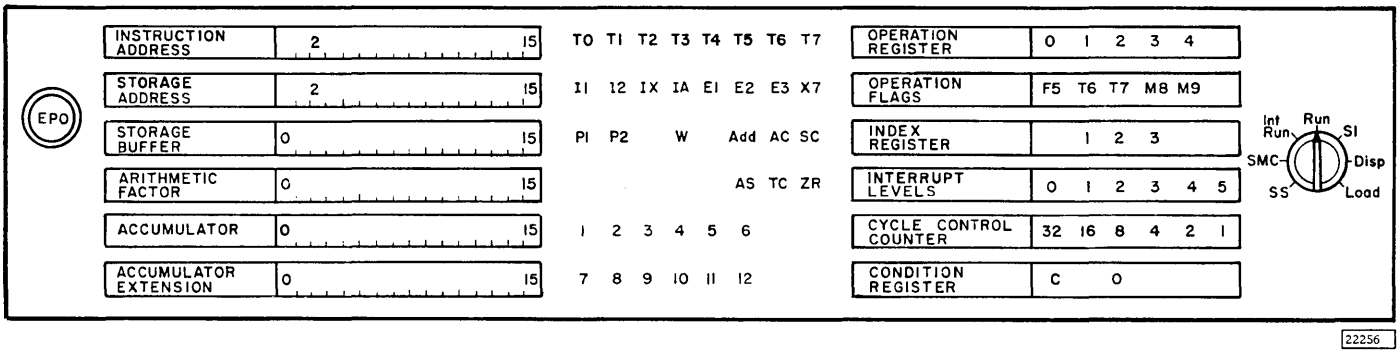


Figure 1-12. Console Display Panel

Operation (OP) Register: Consists of one row of 5 lamps. These lamps indicate the operation in process when in single step mode or single machine cycle mode. They indicate the operation just completed when in any other mode.

Operation Tags: Consists of one row of 5 lamps. These lamps indicate the status of the format, tag, and modifier registers.

Index Register: Consists of one row of 3 lamps. These lamps indicate index register address.

Interrupt Levels: Consists of one row of 5 lamps. These lamps indicate the interrupt levels that are active.

Cycle Control Counter (CCC): Consists of one row of 6 lamps. These lamps indicate the binary value contained in the CCC.

Condition Register: Consists of one row of 2 lamps. These lamps indicate the status of the carry FF and the overflow FF.

1.8.2 Mode Switch

The mode switch selects one of 7 operating modes.

Single Step (SS): With the mode switch set to SS, each depresssion of the start key causes the 1131 clock to advance one step (for example from T1 to T2).

Single Machine Cycle (SMC): With the mode switch set to SMC, each depression of the start key causes the 1131 to advance one machine cycle (for example from I-1 to I-2).

Interrupt Run (INT RUN): With the mode switch set to INT RUN, pressing the start key causes the 1131 to advance through its stored program. A level 5 interrupt occurs after each mainline program instruction is completed.

Program Run (RUN): With the mode switch set to RUN, pressing the start key causes the 1131 to advance through its stored program.

Single Instruction (SI): With the mode switch set to SI, each depression of the start key causes the 1131 to interpret and execute a single instruction.

Display (DISP) Core Storage: With the mode switch set to DISP, pressing the start key will display (in the B register) the core storage word at the location specified by the address in the instruction address register and advance IAR by one.

Load Core Storage (LOAD): With the mode switch set to LOAD, pressing the start key will load the data from the console entry switches (on the 1053 printer) into core storage at the location specified by the address in the instruction address register.

Pressing the load IAR CE switch in this mode transfers the data in the bit switches into the I register.

1.8.3 Emergency Power OFF Switch

This pull type switch drops power to the system and must be reset by the CE.

1.9 CONSOLE PRINTER

- Mechanically an IBM 1053 Printer.
- 16 console entry switches are on the front panel.
- Console entry switches are used to enter data, instructions and addresses.
- See Chapter 3 for a description of the printing functions.

The console printer (Figure 1-13) is located directly beneath the console display panel. It is an IBM 1053 Printer with 16 bit (console entry) switches mounted on the front panel. Each switch is directly related to a bit position of a core storage word.

These switches are used in conjunction with the mode switch on the display panel and the control switches on the console keyboard.

1.9.1 Bit Switch Data Input

To enter data from the bit switches under program operation the console/keyboard switch must be in the console position. The CPU program selects the bit switches, (Area 7), and reads the data from the bit switches on to the I/O input bus and sets it into the B register.

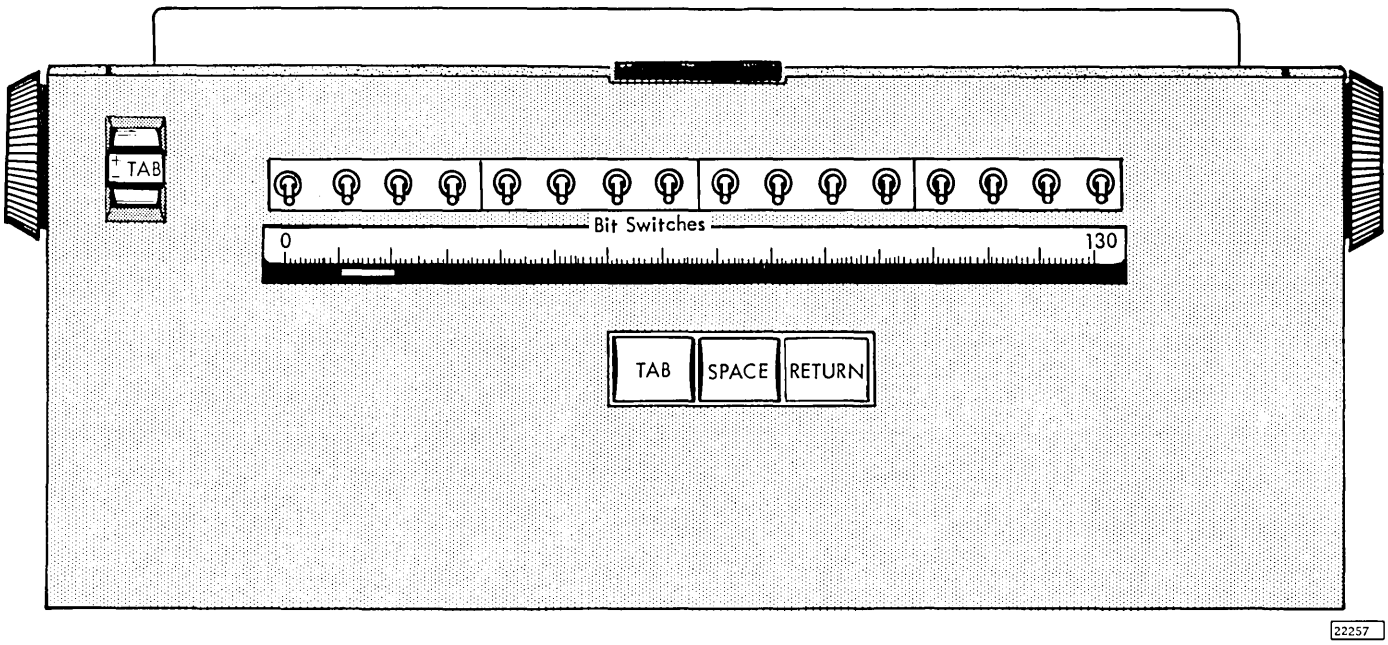


Figure 1-13. Console Printer

1.10 CONSOLE KEYBOARD

- Basically an 029 Card Punch keyboard.
- Contains 8 status indicator lamps.
- Contains 7 control switches.
- See Chapter 3 for a description of the keyboard data input functions.

The console keyboard (Figure 1-14) is located directly in front of the console printer. It is made up of an IBM 029 Card Punch keyboard, a status indicator lamp panel, and a control-switch panel.

1.10.1 Status Indicator Panel

Forms Check: this lamp is turned on when the last form has been detected by the console printer forms contact.

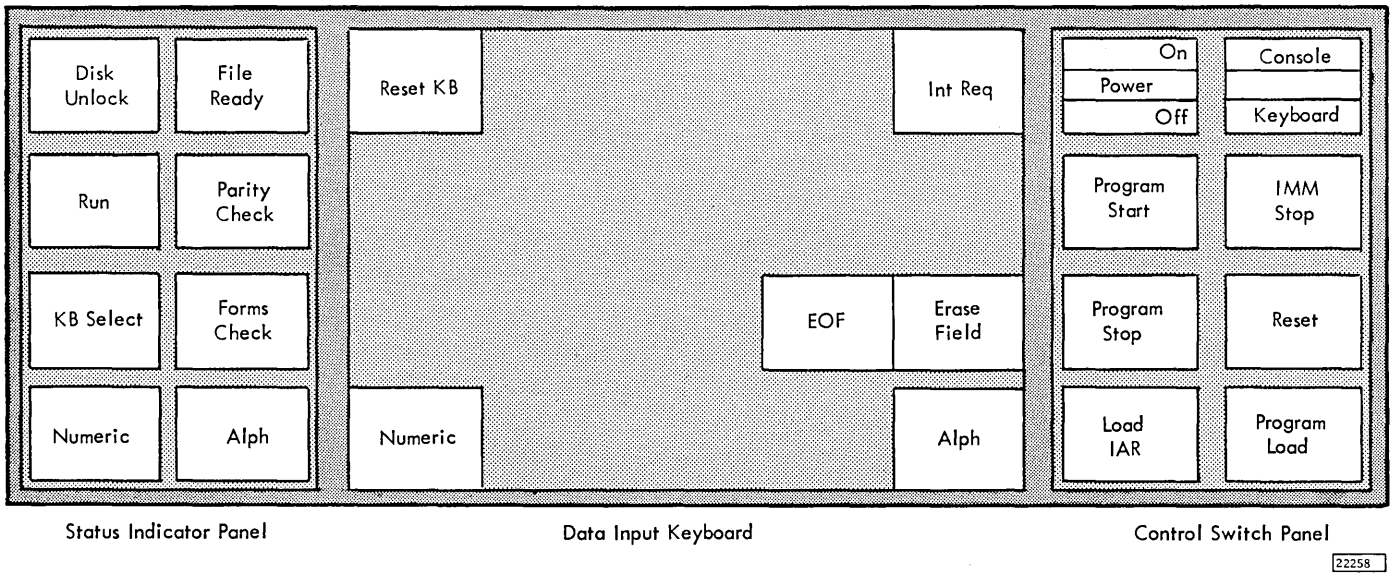


Figure 1-14. Console Keyboard

Keyboard Select: This lamp is turned on by a programmed instruction (XIO Control) that requests input data from the keyboard.

Ready: When on, this lamp indicates that the system is in a "ready-to-operate" status.

Parity Check: This lamp is turned on when a parity error (even number of bits) is detected in the word in the B register.

Alphabetic: This lamp indicates that the keyboard is in alphabetic (lower case) shift. The letters and symbols which appear in the bottom portion of the keyboard character keys can be entered when the keyboard is in alphabetic shift.

Numeric: This lamp indicates that the keyboard is in numeric (upper case) shift. The letters and symbols which appear on the top portion of the keyboard character keys can be entered only when the keyboard is in numeric shift.

Run: This lamp is turned on when the system is executing a program in run mode.

File Ready: This lamp is turned on when the disk cartridge is loaded and ready to operate.

1.10.2 Control Switch Panel.

Power: This two position toggle switch controls system power.

Console/Keyboard: This two position toggle switch indicates the source of the console input data to the program. The source is either the keyboard or the console entry switches.

Start: Pressing this pushbutton switch causes the machine to take one clock step or machine cycle and continue to take additional cycles if required by the setting of the mode switch.

Stop: Pressing this pushbutton switch causes an immediate stop of the CPU, interrupt, and cycle-stealing cycles. I/O devices will complete the cycle they are in when STOP is pressed. Data from I/O devices will be lost if they are operating at the time the stop key is pressed. A complete program restart is normally required.

Program Stop: Pressing this pushbutton switch causes the CPU and I/O devices to "cycle down" to a stop. The "cycle down" procedure is accomplished by turning on the program stop bit in the console device status word and causing a level 5 interrupt. After the program has satisfied all interrupts for levels 1 through 4 (all I/O devices except the console) it enters a routine for level 5. Because the console is the only unit on level 5, the device status word for the console can be sensed without interrogating the interrupt level status word. The program stop bit in the device status word directs the program to a wait loop that blocks all mainline operations until the console operator intervenes.

Reset: Pressing this pushbutton switch resets all machine registers except the M and U registers, cycle FF, control FF, and status indicators. The reset key is inoperative while the machine is run mode.

Load IAR: Pressing this pushbutton switch causes the contents of the bit switches to be placed in the instruction address register.

Program Load: Pressing this pushbutton switch causes a program to be loaded into core storage.

If the system has an IBM 1442 Card Read Punch, the card data is loaded into storage locations 0000-0079 and the CPU will go to location 0000 for the next instruction.

If the system does not have an IBM 1442, the load circuit is added to the IBM 1054 Paper Tape Reader attachment circuits. When the program load key is pressed, the program is loaded into core storage starting at location 0000. Only tape channels 4, 3, 2, and 1 are used. These bits are used in groups of four to form 16-bit words. When a channel 5 punch is read, the read operation is stopped and the CPU will go to 0000 for the next instruction.

1.10.3 Data Input Keyboard. (Figures 1-15 and 1-16)

Keyboard entries are not automatically printed unless the CPU is programmed to provide an output of the entry on the printer. To enter data from the keyboard, the CPU program must select the keyboard (Area 1), with the console/keyboard switch in the keyboard position, and the select light on. The

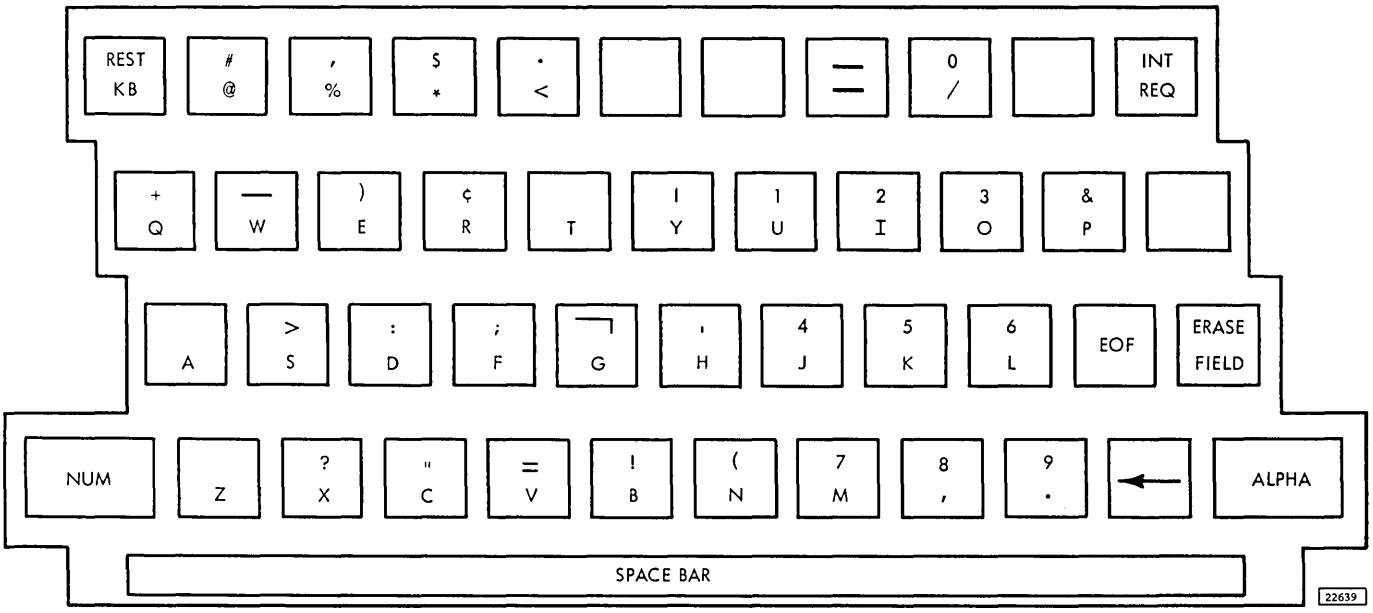


Figure 1-15. Console Keyboard Entry Keys

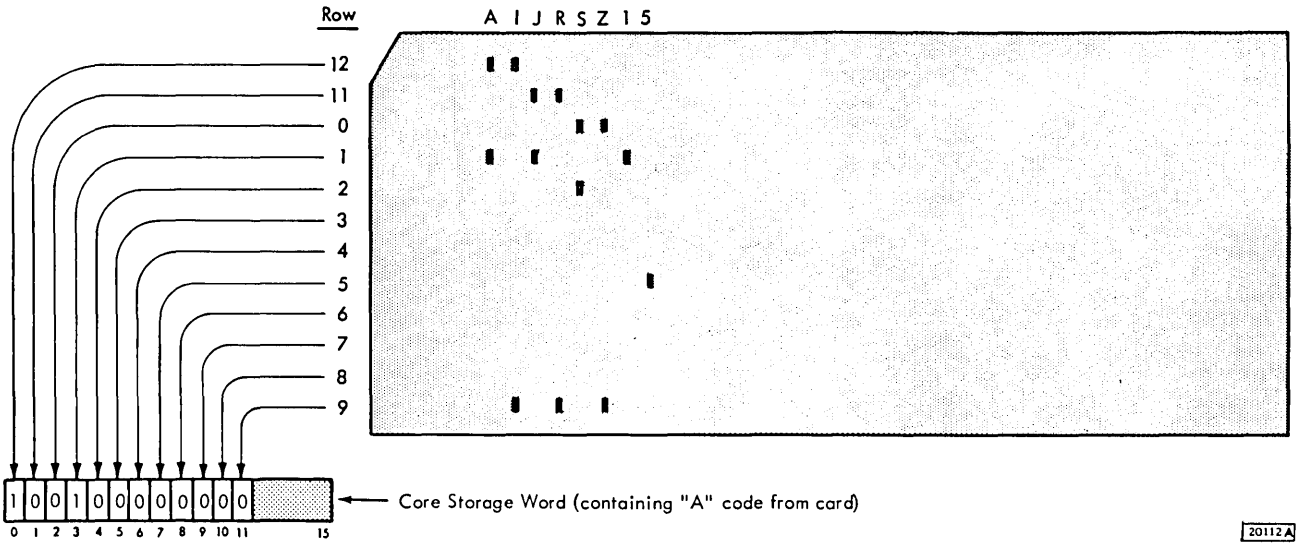


Figure 1-16. Console Keyboard Bit Entry

keyboard emits a coded character for each key struck by the operator. These characters are related to IBM card coding. Striking the 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; etc. See Figure 1-16. The data from the keyboard is entered on the I/O input bus and set into the B register.

1.10.4 Keyboard Function Keys

Interrupt Request: This key initiates a keyboard restore and causes an interrupt in the CPU.

End of Field (EOF): When the CPU reads in response to this key, a word containing a 12 bit only is placed in memory. Analysis of this word allows the program to determine that no further characters are to be sent in this message.

Backspace (←): When the CPU reads in response to this key, a word containing a 13 bit only is placed in memory. Analysis of this word allows the program to determine that the last character received is to be replaced by the next character to be entered.

Erase Field: When the CPU reads in response to this key, a word containing a 14 bit only is placed in memory. Analysis of this word allows the program to determine that the message being entered is to be deleted and replaced by a corrected message.

Mode: They are two mode keys: numeric (upper case shift) and alphabetic (lower case shift). These keys place the keyboard in the indicated mode. The keyboard remains in the selected mode until changed by the operator. If the numbers or symbols which appear on the top portion of the keys are desired, the keyboard must be placed in numeric mode.

Reset Keyboard: This key allows the operator to restore the keys if they become locked.

1.10.5 Keyboard Operation

The following procedure describes a typical use of the keyboard.

1. The operator presses the keyboard request key with the console/keyboard switch in the keyboard position, to initiate a request interrupt and place the keyboard in a restore status.
2. The CPU honors the request interrupt.
3. When the keyboard has been determined to be the device that caused the interrupt, the CPU issues a control command to select the keyboard. When the keyboard is selected, the select light is turned on to signal the operator that a character can be entered.
4. When a character key is pressed, the keyboard initiates a service interrupt to the CPU.
5. In response to the service interrupt, the CPU performs a read command which enters the character into core storage and removes the keyboard from the selected status.
6. Before another character can be entered, the CPU stored program must issue another control command to select the keyboard.

If the CPU performs a read command when the keyboard is not selected, no bits are entered.

1.11 CE PANEL

The CE panel (Figure 1-17) is located under the right end of the console display panel and contains six switches.

Lamp Test: This switch lights all lamps on the display panel to check their operation.

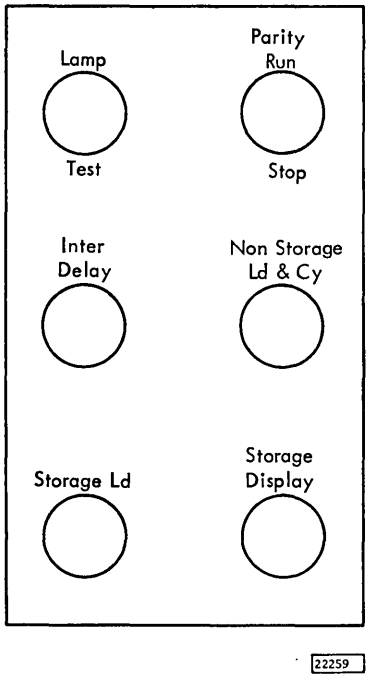


Figure 1-17. CE Panel

Parity Run/Stop: This two position toggle switch is normally set to STOP. If the CE wishes the program to bypass a parity error for diagnostic purposes, this switch is set to RUN and the program continues to run even if a parity error is detected.

NOTE: The parity run/stop switch must be in STOP when the system is returned to the customer.

Interrupt Delay: This switch when on blocks setting of the interrupt FF's.

Non-Storage Load and Cycle: Pressing the start key with this switch on allows the information in the bit switches to enter the B register. Input and output of core storage is inhibited. Depending on the setting of the mode switch, a cycle or step is taken without changing storage. This operation allows the CE to check the operation of the CPU without changing data in core storage.

Storage Load: When this switch is on, the information in the bit switches is entered into core. Depending on the setting of the mode switch, one or all core positions will be changed. For example, to set all core storage positions to zero, set all bit switches off, set the mode switch to run, set the storage load switch on and press the start key. Parity checking is under control of the parity run/stop CE switch.

Storage Display: Pressing the start switch when this switch is on causes the information in core storage to be displayed in the B register. Parity checking is under control of the parity run/stop CE switch.

INSTRUCTION TIME

1.12 INSTRUCTION CYCLES

- Instruction time (I cycles) interprets the instruction and develops an effective address (EA).
- Reads the instruction from the core storage location specified by the instruction address register.
- Six instruction types are used:
  - Single-word format, without indexing.
  - Single-word format, with indexing.
  - Double-word format, without indexing, without indirect addressing.
  - Double-word format, with indexing, without indirect addressing.
  - Double-word format, without indexing, with indirect addressing.
  - Double-word format, with indexing, with indirect addressing.
- Decodes the instruction format bit to determine whether the instruction is one or two words in length.
- Decodes the instruction tag bits to determine if the instruction is an indexing instruction.
- Decodes the instruction modifier bit to determine if the instruction has an indirect address.
- Instruction cycle 1 (I-1) is used by all instructions.
- Instruction cycle 2 (I-2) is used by all double-word-format instructions.
- Indirect addressing cycle (I-A); is used by indirect addressing instructions.
- Indexing cycle (I-X) is used by indexing instructions.
- Reference Maintenance Diagrams: AA101, AA211, AA601.

In the performance of the stored program instructions, the computer proceeds through instruction time (I-cycles), and generally execution time (E-cycles) for each operation (Figure 1-18). Instruction cycles read the instruction from core storage, store the instruction in control registers, and decode these control registers to specify a distinct operation.

The time required for the instruction cycles is dependent upon the type of instruction (Figure 1-19) and the factors to be used to develop the effective address.

All four I-cycles (I-1, I-2, I-X, and I-A) read and interpret instructions and develop an effective address (EA) for subsequent execute cycles.

Normal turn-on time and normal turn-off time of all I-cycle FF's is T0. A slight overlap of I-cycle FF's may occur at T0. This overlap will extend from the turn-on of one latch to the turn-off of the previous FF. For example, I-1 conditions the circuits required to turn on I-2. As I-2 is being turned on, I-1 is being turned off. An overlap of the two FF's may occur, depending upon the timing of the internal circuits of the FF.

Only functions common to all I-cycle FF's are performed during T 0 . The turn-on of successive FF's is interlocked by control gates to prevent more than one cycle (I-1, I-2, I-A, or I-X) FF from being on simultaneously during instruction time.

1.13 INSTRUCTION CYCLE 1 (I-1)

- Used by all instructions.
- Stores the 16 bits of the first word of the instruction in control registers.
- Decodes the contents of the control registers to determine the operation to be performed.
- Reference Maintenance Diagram: AA601.

This cycle reads the data from core storage at the address specified by the storage address register. (The instruction address register was set to the correct address during the previous instruction.) The core storage word is read into the B register, distributed from B register to the various control registers (Figure 1-20), and decoded to indicate the operation to be performed.

There are three major types of I-1 cycles:

1. I-1 used for single-word format instructions without indexing.
2. I-1 used for single-word format instructions with indexing.
3. I-1 used for all double-word format instructions.

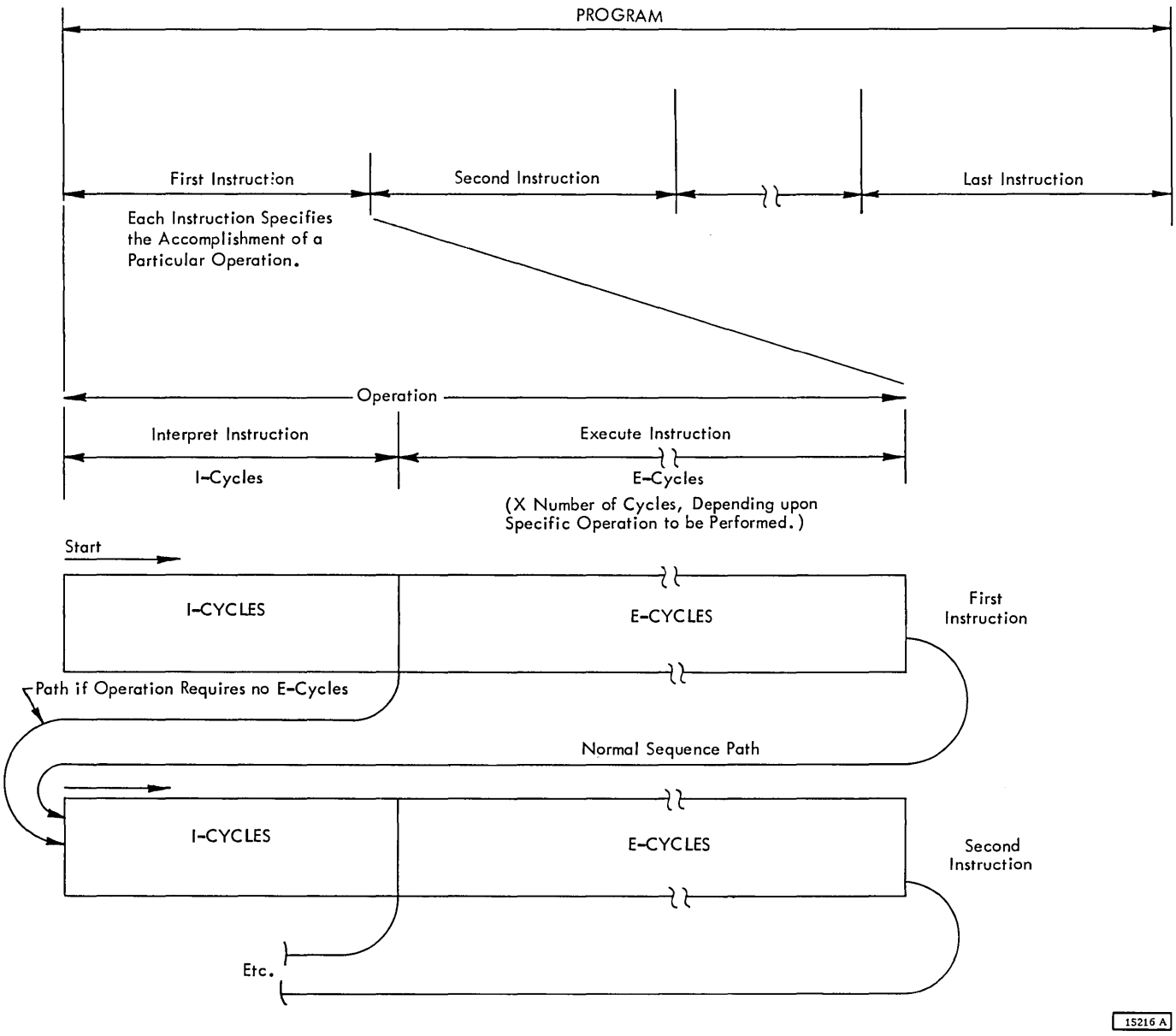


Figure 1-18. Program I Cycles - E Cycles

Type Of Instruction	Cycle Required			
	I-1	I-2	IX	IA
Single - Word	x			
Single - Word With Indexing	x		x	
Double - Word	x	x		
Double - Word with Indexing	x	x	x	
Double - Word With Indirect Addressing	x	x		x
Double - Word With Indexing And Indirect Addressing	x	x	x	x

22211

Figure 1-19. Instruction Cycles

B Register Bits	Register
0 to 4	Operation (OP)
5	Format (Flag)
6 and 7	Index Address (Tag)
8 and 9	Modifier (Mod)
10 to 15	Cycle Control Counter (CCC)
0 to 15	Data (D)
14 and 15 by way of D reg.	Carry - Overflow (C/O)

22212

Figure 1-20. B Register to Control Register Transfer

Each type of I-1 cycle has a specific function in addition to the normal functions previously described.

1.13.1 Single-Word Format Without Indexing (F=0, Tag=00)

An effective address must be generated during this I-1 cycle that subsequently addresses the core storage word that is manipulated during the execute portion of the instruction.

The effective address is generated by loading the contents of I register into the accumulator and algebraically adding the displacement (bits 8 through 15 of the instruction) to the accumulator (I register contents).

Circuit Objectives

T0:

1. Transfer I register to M register.
2. Increment I register.

T1:

1. Transfer ACC to U register (save contents of ACC for future use).

T2:

1. Read word at location specified by M register to B register.
2. Transfer B register bits 0-4 to operation register.
3. Transfer B register bit 5 to flag register (this bit must be off to specify single-word format).
4. Transfer B register bits 6 and 7 to tag register (these bits must be off to specify no indexing required).
5. Transfer B register bits 8 and 9 to MOD register.
6. Transfer B register bits 8-15 to D register bits 8-15 (second factor of effective address).

T3:

1. Set arithmetic control (required to add D register bits 8-15 to ACC).
2. Transfer I register to ACC (first factor of effective address).

T4:

1. Add displacement (D register bits 8-15) to accumulator.

T5, T6:

1. Continue add operation.

T7:

1. Continue add operation (extend T7 if required).
2. Exit to E-1 cycle, or I-1 cycle as indicated by the operation code.

1.13.2 Single-Word Format With Indexing (F=0, Tag $\neq$ 00)

The effective address for this type of instruction is not generated completely during the I-1 cycle. However, the displacement portion of the instruction is placed in the accumulator so that the contents of the index register can be added to the displacement during the I-X cycle.

Circuit Objectives

T0:

1. Transfer I register to M register.
2. Increment I register

T1:

1. Transfer ACC to U register (save contents of ACC for future use).

T2:

1. Read word at the location specified by M register to B register.
2. Transfer B register bits 0-4 to operation register.
3. Transfer B register bit 5 to flag register (this must be off to specify single-word format).
4. Transfer B register bits 6 and 7 to tag register (one of these bits must be on to specify indexing).
5. Transfer B register bits 8 and 9 to MOD register.
6. Transfer B register bits 8-15 to D bits 8-15 (first factor of effective address).

T3:

1. Set arithmetic control.

T4:

1. Add displacement (D register bits 8-15) to accumulator.

T5, T6:

1. Continue add operation.

T7:

1. Exit to I-X Cycle.

1.13.3 Double-Word Format (F=1)

The effective address for this type of instruction is generated completely by I-cycles other than I-1. Therefore, this I-1 cycle does not set any factors into the accumulator other than the address of the address word.

Circuit Objectives

T0:

1. Transfer I register to M register
2. Increment I register.

T1:

1. Transfer ACC to U register (save contents of ACC for future use).

T2:

1. Read word at location specified by M register to B register.
2. Transfer B register bits 0-4 to operation register.
3. Transfer B register bit 5 to flag register (this bit must be on to specify double-word format).
4. Transfer B register bits 6 and 7 to tag register.
5. Transfer B register bits 8 and 9 to MOD register.

T3:

1. Transfer I register +1 to the ACC, if Tag $\neq$ 00.

T5:

1. Set CCC to 1.

T7:

1. Exit to I-2 cycle.

1.14 INSTRUCTION CYCLE 2 (I-2)

- Used by all double-word format instructions.
- Reads the 16 bits of the second word of the instruction into ACC via B register and D register. The second word in a double-word format instruction is the effective address unless it is modified by indexing or indirect addressing. In any event, the 16-bit address is placed in the accumulator during I-2. Any modification is made to the accumulator during IX or IA cycles.
- Reference Maintenance Diagram: AA601.

Circuit Objectives

T0:

1. Transfer I register to M register (I register was incremented by +1 during the previous I-1 cycle).
2. Increment I register.

T2:

1. Read word at location specified by M register into B register.
2. Transfer B register to D register.

T4:

1. Transfer D register to ACC.

T7:

1. If either tag register FF is on (bit 6 or 7 of instruction);
   a. Set XR FF.
   b. Exit to I-X cycle.
2. If neither tag register FF is on, test MOD register for indirect addressing bit (Bit 8 of the instruction).
   a. If IA bit is on, set IA FF and exit to IA cycle.
   b. If IA bit is off, exit to E-1 cycle or I-1 cycle as indicated by the operation code.

1.15 INDEXING CYCLE (I-X)

- Reads the contents of the index register addressed by the tag bits of the instruction (bits 6 and 7).

- Generates an effective address (add XR to contents of accumulator) for the data to be manipulated during the following execute time.
- Indexing takes precedence over indirect addressing.
- Instruction address is not changed.
- Arithmetic check indicators are not altered.
- Reference Maintenance Diagram: AA601.

The I-X cycle reads the contents of the index register specified by the tag bits of the instruction (bits 6 and 7 now in the tag register) and adds the contents of the XR to the contents of the accumulator (displacement if single-word format; address word if double-word format).

The contents of the accumulator is always considered as positive (absolute). The instruction as it appears in core storage is not changed by the indexing operation.

When an address is indexed and indirect, the indexing procedure has precedence. The resulting effective address after indexing is the new effective indirect address. The address at the indirect address must be a direct address (only one indexing and one indirect addressing operation are allowed for each instruction).

The three index registers are located in the first three core storage locations (addresses 0001, 0002, and 0003). The index registers are addressed by circuits which block the normal output of the M register and decode the contents of tag register to energize the M register output lines for bits 14 and 15.

Circuit Objectives

T7 of previous I-1 or I-2 cycle.

1. Set XR address FF (this FF blocks the normal M register output).

T0 of I-X cycle.

1. Select index register by raising correct M register bit 14 and/or bit 15 output.

T1:

1. Note that the I-X cycle does not transfer ACC to the U register. This is because U register contains the accumulator data developed during a previous instruction and the accumulator contains the address that must be indexed.

T2:

1. Read word at location specified by M register to B register.
2. Transfer B register to D register.

T3:

1. Set arithmetic control.

T4:

1. Algebraically add D register (contents of index register) to ACC (effective address).

T5, T6:

1. Continue add operation.

T7:

1. Continue add operation, if required.
2. Extend T7 cycle if required.
3. Exit to I-A cycle if Modifier bit 8 is on.
4. Exit to I-1 or E-1 as indicated by operation code, if IA bit is off.

1.16 INDIRECT ADDRESSING CYCLE (I-A)

- Reads a direct address from the core storage location specified by the indirect address in the accumulator.
- Transfers the direct address to the accumulator to replace the indirect address.
- Reference Maintenance Diagram: AA601.

A direct address can be generally defined as the address of data in core storage. An indirect address can then be defined as the address of a direct address. For most 1130 instructions, the effective address that is generated during I-1, I-2, and I-X cycles is a direct address.

However, the presence of a "one" in the bit 8 position of the instruction indicates that the effective address is really an "indirect" address.

This "indirect" address is the core storage address of a "direct" address.

The I-A cycle reads the indirect address into the M register, reads the direct address from the core storage location specified by the M register

and transfers the direct address to the accumulator via B register and D register.

Only one level of indirect addressing is available in the 1130 system. This means that the indirect address of an instruction always addresses a direct address and never addresses another indirect address.

Circuit Objectives

T0:

1. Transfer accumulator (indirect address) to M register.

T1:

1. Note that the I-A cycle does not transfer the ACC to the U register. This is because the U register contains the accumulator data developed during a previous instruction and the accumulator contains the indirect address.

T2:

1. Read the word (direct address) from the core location specified by M register (indirect address) into B register.

2. Transfer B register (direct address) to D register.

T3:

1. Transfer D register to ACC.
2. Note: Indirect address is cleared out of ACC.

T7:

1. Exit to I-1 cycle or E-1 cycle as indicated by operation code.

EXECUTION TIME

1.17 EXECUTION CYCLES

- Execute the instruction read from core storage and interpreted during the associated instruction time.
- Execution time consists of a number of 3.6 μsec machine cycles (E cycles) under the control of E-cycle FF's.
- Some instructions do not require any execution time.
- Reference Maintenance Diagrams: AA101, AA211, AA601.

The performance of a computer operation is normally divided into two parts, instruction time (I Cycles) and execution time (E cycles) (see Figure 1-21).

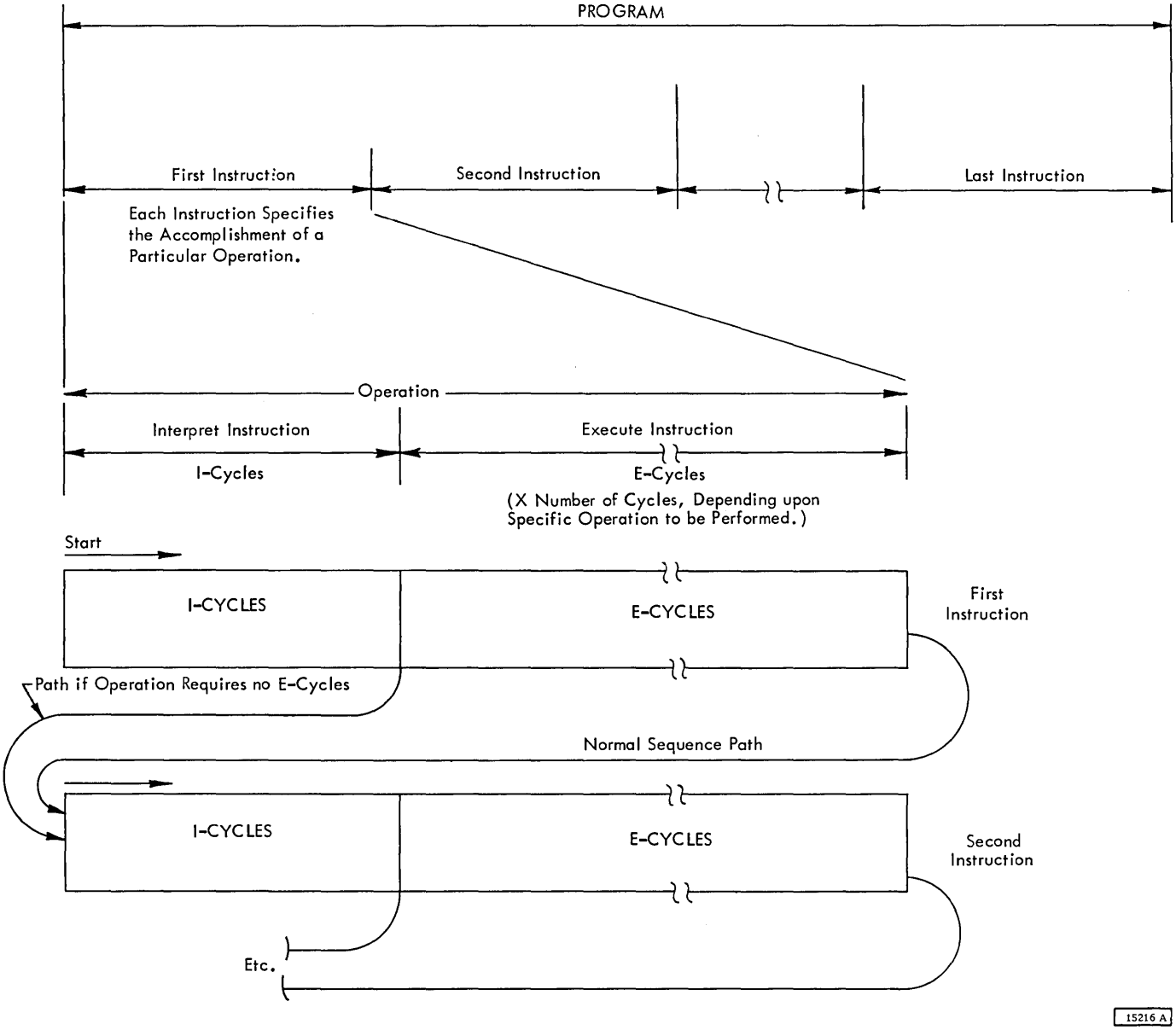


Figure 1-21. Sequences

Each execution time is immediately preceded by its associated instruction time during which the instruction is read from core storage and interpreted.

The function of execution time is to accomplish the objectives of the particular operation specified by the instruction. The required number of machine cycles depends upon the specific operation to be performed and upon the factor(s) involved in the operation.

The following instructions are complete at the end of their instruction time and require no execute time:

1. Wait (00000)
2. Shift left (00010) if not shift left and count and ACC bit 0 = 1 .
3. Shift right (00011)
4. Load status (00100)
5. Branch and store 1 register (01001) if condition not met.
6. Branch or skip on condition (01001).
7. Load index (01100) if no index register is specified.
8. MDX (10100) if flag = 0 and tag = 00.

When the execution time for an operation is complete, the computer is directed to enter the instruction time for the next instruction in sequence.

1.17.1 E-Cycle Flip Flops

In general, each E-cycle FF controls all of the functions which must be accomplished during the machine cycle for which it is on. At the beginning of the following machine cycle a T0 clock pulse turns on the next E-cycle FF which is to assume control and turns off the E-cycle FF which was on.

Because identical or similar functions are accomplished during the execution of different computer operations, the E-cycle FF's are used with more than one operation, and more than once in any specific operation. For example, the load double (11001) operation is executed in two execute cycles, using E-cycle FF's E-1 and E-2 (Figure 1-22). The load accumulator (11000) operation is executed in one execute cycle, using E-cycle FF E-1. FF E-1 is used with both operations to accomplish a similar function, that of reading a 16-bit word from core storage and loading into the accumulator.

The functions of individual E-cycle FF's are described in detail in other sections of this manual as they are used to accomplish the objectives of the various computer operations. However, each E-cycle FF has a "normal" function and objective. These "normal" functions and objectives are performed each cycle that the FF is on except during the cycles that they are specifically blocked. These "normal" functions and objectives are:

T0: Transfer the effective address that was generated during I-cycles from the accumulator to the storage address register.

T1:

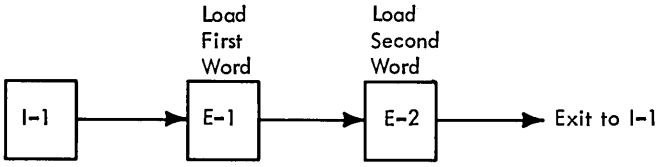
1. Transfer the contents of the temporary accumulator (U) register to the accumulator (ACC). (The contents of ACC is placed in U register at the beginning of instruction time, and is merely placed back into ACC at the beginning of the first E-cycle of an instruction.)
2. Decrement the cycle control counter. The cycle control counter is set during instruction time for the number of cycles required to complete the operation.

T2: Read the word at the core storage location specified by M register in B register.

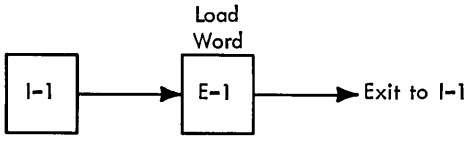
1. Transfer B register to D register.

T4:

1. Store the contents of B register back into the core storage location specified by M register.



Instruction: Load Double (Load two words from core storage into Accumulator).



Instruction: Load Accumulator (Load one word from core storage into Accumulator).

24038

Figure 1-22. Accumulator Load Instructions

2. Transfer, compare, or add D register to the contents of the accumulator depending upon the operation code.

T5, T6, T7:

1. Continue the compare or add operation if required.
2. Extend T7 time if required.

The sequential advance from one numbered E-cycle FF to another is controlled by operation code gating. At the beginning of each machine cycle, the operation code and the conditions of the operation gate on the next cycle FF. The previous cycle FF is turned off by a T0 clock pulse.

1.17.2 Auxiliary Flip-Flops

Auxiliary FF's are used to establish or recognize conditions within the computer which must be considered to properly accomplish the execution of an instruction.

Functions of the most commonly used execution auxiliary FF's are described here. These and other auxiliary FF's are described in detail in the sections of this manual where they are used in the various computer operations.

Overflow

1. This FF and its associated indicator are turned on by:
   a. An add, subtract, or divide operation that produces a result larger than can be accurately represented in the accumulator.
   b. A load status (00100) instruction that specifies the turn on of the overflow indicator.
   c. An error detected during a multiply or divide operation.
2. This FF and its associated indicator are turned off by:
   a. A branch instruction that tests the overflow FF.
   b. A load status (00100) instruction that specifies the turn off of the overflow indicator.
   c. A store status (00101) operation.

Carry

1. This FF and its indicator are turned on by:
   a. Arithmetic and shift operations, to indicate a one bit in the high order position of the accumulator or a carry out of the high order position of the accumulator.
   b. A load status (00100) instruction that specifies the turn on of the carry FF.
   c. A transfer of the CCC to the accumulator.
2. This FF and its associated indicator are turned off by:
   a. A branch and skip conditionally (01001) instruction that tests the carry FF.
   b. A load status (00100) instruction that specifies the turn off of the carry FF.
   c. A store status (00101) operation.
   d. A shift left and count operation.

Double Precision

This FF indicates that the factor of an instruction is contained in two core storage words. These two 16-bit words must be treated as one 32-bit word.

The effective address (EA) of a double-precision pair of words must be even because the double-precision circuits are designed to read the core storage words at EA and EA + 1. If EA is already odd, the +1 function is impeded and EA is read twice, thereby loading or storing the same word twice.

PROGRAMMING

1.18 DEVELOPMENT OF A PROGRAM

- A program is a series of instructions that perform an entire procedure.
- A program will normally accept input data, develop a result, and deliver output data.

Programming consists of defining the steps required to receive data, processing the data, and displaying the results in terms of the operations which the computer system is capable of performing. Each step must be written as an instruction to the computer. A series of instructions pertaining to an entire procedure constitutes a program.

A program is normally entered into core storage by a program load operation from cards or paper tape.

A manually entered program to accomplish entry of factors, solution of a simple problem, and display of the result is presented here to illustrate 1130 computer programming (Figure 1-23) and provide instructions for use of the console.

1.19 COMPUTE X + Y = Z

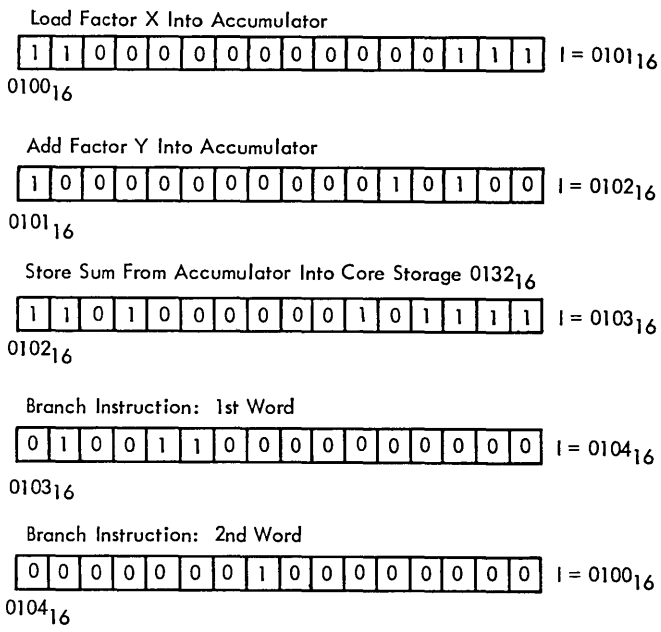
1. Enter factors X and Y into core storage by means of the console bit switches.
   a. Store factor X at core storage location hex 0108.
   b. Store factor Y at core storage location hex 0116.
2. Develop the sum (Z) of X + Y.
   a. Put factor X into accumulator.
   b. Add factor Y to factor X. (The developed sum will replace factor X in the accumulator).
3. Store the sum (Z) at core storage location hex 0132.
4. Display the sum (Z) on the console.

1.20 LOAD FACTORS INTO CORE STORAGE

- Factors can be entered from the console bit switches without the need for a stored program.
- Other input devices can enter factors only when conditioned for entry by a stored program.

Enter factor X (4762) at address hex 0108.

X + Y = Z



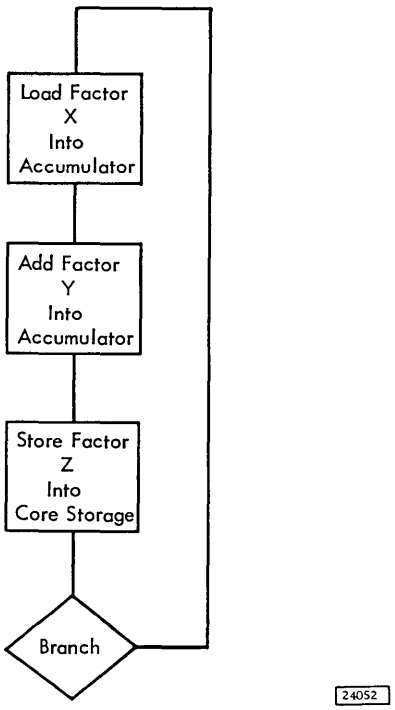


Figure 1-23. X + Y = Z Program

1. Set console bit switches 7 and 12 (address hex 0108).
2. Set mode switch to LOAD.
3. Press LOAD IAR (instruction address register).
4. Set console bit switches ON for factor X (4762): 3, 6, 8, 11, 12, 14.
5. Press START.

Enter factor Y (3429) at address hex 0116.

1. Set console bit switch 7, 11, 13 and 14 (address hex 0116).
2. Press LOAD IAR.
3. Set console bit switches ON for factor Y (3429): 4, 5, 7, 9, 10, 13, 15.
4. Press START.

1.21 LOADING A PROGRAM INTO CORE STORAGE

- Instructions can be entered from the console bit switches, the card reader, and the paper tape reader (if the system has no card reader) without the need for a stored program.

- Instructions can be entered from the console keyboard and disk storage only when conditioned for entry by a stored program.

Because the IBM 1130 CPU is a stored program computer, the instructions comprising a program must be written in core storage to be available to the computer so it can accomplish the program objectives.

Enter the program at addresses hex 0100, hex 0101, hex 0102.

1. Store the first instruction - Load factor X into accumulator.
   a. Set mode switch to LOAD.
   b. Turn on console bit switch 7 (address hex 0100).
   c. Press LOAD IAR.
   d. Set console bit switches for first instruction (Load factor X into accumulator). 1100 0000 0000 0111.
   e. Press START.
2. Store second instruction - Add factor Y to accumulator.
   a. Turn off all console bit switches.
   b. Set console bit switches for second instruction (add factor Y into accumulator). 1000 0000 0001 0100
   c. Press START
   d. Turn off all console bit switches.
3. Store third instruction - Store sum (Z) from accumulator into core.
   a. Turn off all console bit switches.
   b. Set console bit switches for third instruction: (Store sum from accumulator into core at address hex 0132). 1101 0000 0010 1111
   c. Press START.

1.22 EXECUTION OF A PROGRAM

- Instructions are normally interpreted and executed serially.

- Instructions are read from core storage starting with core storage word hex 0100 in the above program.

Instructions within a program are normally interpreted and executed sequentially; that is, the first instruction is stored in core storage location hex 0100, the execution of that instruction is normally followed by the interpretation, and, execution of the instruction stored at location 0101, etc.

Each instruction is read from core storage and stored in registers for interpretation.

Because execution of the program requires the use of factors X and Y, but does not request them from an input device, they must be present in core storage before the program is started.

Pressing the start key causes the program to begin execution in the mode selected by the mode switch.

The mode switch allows the program to execute in one of four modes:

Single clock step mode (SS).
Single machine cycle mode (SMC).
Single instruction mode (SI).
Run mode (RUN).

1.22.1 Execute Program

1. Set mode switch to LOAD.
2. Turn on the console bit switch 7 (address hex 0100).

3. Press LOAD IAR
4. Set mode switch to SI (single instruction)
5. Press START. Factor X is now in the accumulator, and the B register.
6. Press START again. Factor Y is now in the B register. The sum of X and Y (Z) is now in the accumulator.
7. Press START again. The sum (Z) is now in the B register and in core storage location hex 0132.

1.23 DISPLAY RESULT

- Any core storage word can be displayed.
- Console bit switches select the word to be displayed.
- Mode switch selects display mode.
- Pressing START causes actual display.

1.23.1 Display Sum (Address hex 0132)

1. Set the mode switch to LOAD.
2. Turn on console bit switch 7, 10, 11 and 14 (address hex 0132).
3. Press LOAD IAR.
4. Set the mode switch to DISP (display).
5. Press START. The sum (at address hex 0132) is now displayed in the B register.

1.24 PROGRAM REPETITION (PROGRAM LOOP)

- A program can be "looped" to repeat all of the instructions using different input data.
- A program is looped by returning ("branching") from the last instruction back to the first instruction.

The program entered computes the sum for only one problem, but by returning to the first instruction of the program and repeating the program as a loop, the sums for any number of problems can be computed. A program loop can be initiated by entering new factors and repeating the program execution.

1.24.1 Enter Different Factors

Different factors can be entered by repeating the steps listed under Enter Factor X or Enter Factor Y. Step 3, however, must be changed to the correct bit structure of the new factor.

1.24.2 Store Branch Instruction

1. Set the mode switch to LOAD.
2. Turn off all console bit switches.
3. Turn on console bit switch 7, 14, and 15 (address hex 0103).
4. Press LOAD IAR.
5. Set console bit switches for branch instruction: 0100 1100 0000 0000 (First word)
6. Press START.
7. Set console bit switches for branch instruction: 0000 0001 0000 0000 (Second word)
8. Press START.

1.24.3 Repeat Program Execution

A new sum can be developed from the new factors by repeating the execute program routine.

DISK STORAGE

- Disk storage is available on IBM 1130 Models 2A and 2B Systems only.
- Disk storage capacity is 512,000 on line 16 bit data words.
- The disk storage is contained in the 1131 cabinet.
- 2315 Disk Cartridge is used as the recording medium.
- The cartridge is easily changed for unlimited storage capacity.
- The speed of the disk is 1500 RPM or 40 ms per revolution.
- Uses two recording surfaces and two read write heads controlled by one access mechanism.
- Accesses in increments of one or two tracks.
- Can record in 200 tracks and cylinders.
- Each cylinder contains 8 sectors, 4 on the upper track and 4 on the lower track. The sectors are numbered 0 through 7.
- Each sector can contain up to 321 words.
- Each word recorded on the disk consits of 16 data bits and four check bits.
- Word rate is 27.8 μsec per word.
- Bit rate is 1.39 μsec per bit.
- The first word of each sector should be used to identify the record.
- More detailed information on the Disk Drive is given in the IBM Single Disk Storage (Serial Numbers 00001 through 39999), Field Engineering Theory of Operation (Manual of Instruction). See the bibliography for the form number.
- Reference Maintenance Diagram: XF401.

1.25 DISK STORAGE DESCRIPTION

Disk storage provides the IBM 1130 Computing System with low-cost random or sequential access data storage. On-line data capacity is 512,000 words. Off-line capacity is virtually unlimited because the interchangeable disk cartridge is easily removed and replaced with another. Thus, the large storage capacity, comparable to that of magnetic tape, coupled with the unique advantage of random access, affords the 1130 Computing System great flexibility in handling engineering, scientific, industrial, and commercial programs.

1.25.1 Disk Storage Unit

Disk storage for the 1130 system is contained in the CPU cabinet and is connected to the CPU by the disk storage attachment. It has two components: the disk drive assembly and the access mechanism.

1.25.2 Disk Assembly

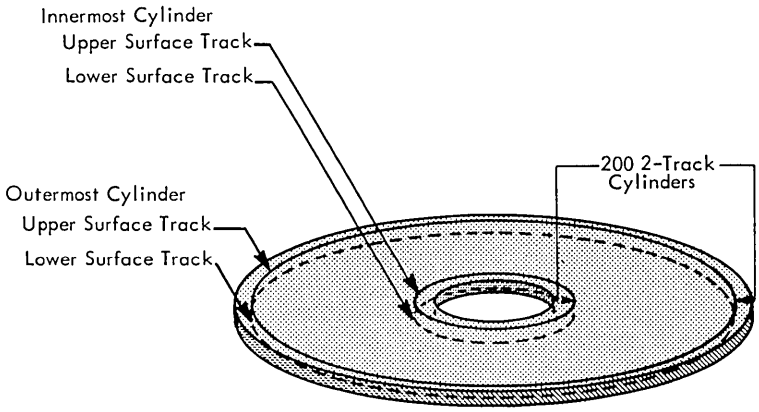
The assembly is a single disk, completely enclosed in a protective housing or cartridge. The recording medium is an oxide-coated disk that provides two surfaces for the magnetic recording of data. The disk rotates at the rate of 1500 revolutions per minute.

1.25.3 Access Mechanism

The disk storage access mechanism has two arms mounted horizontally on a vertical assembly. 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 when the disk cartridge is inserted.

When the disk drive switch is turned off, the access mechanism retracts the heads from the cartridge.



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

Figure 1-24. Disk Cylinder Concept

1.25.4 Disk Organization and Capacity

The access mechanism is moved back and forth by program instructions and can be placed in any one of 200 positions, from the point of greatest withdrawal to the point of farthest advance. At each position, either head reads or writes in a circular pattern on a surface of the disk, as it revolves. These 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, together are called a cylinder. Figure 1-24 shows the innermost and outermost cylinders of two tracks each. To complete the picture, the 198 intermediate cyclinders, or pairs of tracks, should be visualized as they were omitted for the sake of clarity in 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 in the cylinder, from 0 through 7, as shown in Figure 1-25. Sectors 0-3 divide the upper surface track, and Sectors 4-7, the lower. A sector contains 320 data words and is the largest segment of data that can be read or written with a single instruction.

In addition to 320 data words, each sector has an additional word that can be used as needed by a particular program. 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 sector number. Therefore, the first word of the sector must be used with caution by the programmer if the Assembler program or other components of the Monitor system are to be used. Otherwise, this word may be used for a sector address, data, or other purposes. The format of a track is shown in Figure 1-26. The zero bit field and the sync word are written by the attachment controls and are used by the attachment circuits to sync to the data when reading back.

A disk storage word is comprised of 16 data bits and four check bits. Figure 1-27 shows the layout of a word. Clock bits are recorded every 1.39 μsec and if a "one" bit is to be written, it is placed between the two clock bits. The check bits in the last four bits of each 20 bit word depend on the number of data "one" bits written. The modulo 4 counter in the attachment is used to write the check bits. The relationship of the number of data bits to the modulo 4 counter and the checks bits written is shown in Figure 1-28.

Figure 1-29 shows the organizational components of disk storage. Note that capacities are based on the 320-word sector.

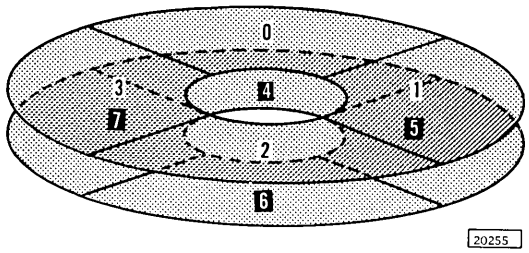


Figure 1-25. Sector Numbers of Upper and Lower Disk Surfaces

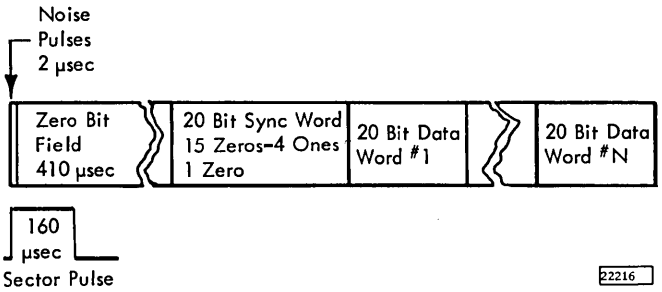


Figure 1-26. Sector Format

1.25.5 Disk Storage 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.

Access. The access mechanism moves in increments of one or two cylinders at the rate of 15 ms per increment. All movements of the access will be in increments of two, except the first movement if the number of cylinders to be moved is odd. During the 28 ms stabilization period that follows the last incremental movement, a read or write instruction can be given and will be started at the end of the stabilization period.

$$\text{Access time (ms)} = 7.5\ (N) + 28$$

Read/Write. Reading or writing of data in disk storage is at the rate of 27.8 µsec per word. Average rotational delay time is 20 ms, based on 1500 rpm, or 40 ms per revolution. Thus, a sector can be read or written in an average of 30 ms. There are no timing considerations for switching heads because there is an interval of 450 µsec between sectors; the interval is increased by 27.8µsec 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.

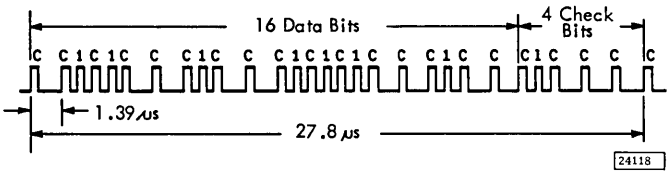


Figure 1-27. Disk File Data Word

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 1-28. Check Bit Chart

CPU Time. An interrupt in a disk storage operation occurs only at the end of the seek or read/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 µsec for the transmission of the next word. Thus, except for the normal instruction times, the CPU is busy only 14 ms of the 100 ms required to read or write a full cylinder. The remaining 86 ms are available for other program operations.

1.25.6 Disk Storage Data Checking

Data is checked on each transmission between disk storage and core storage. When writing on disk storage the number of bits of 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. 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 (Figure 1-28).

No. of Per	Disks	Cylinders	Tracks	Sectors	Words
Cylinders	200				
Tracks	400	2			
Sectors	1,600	8	4		
Data Words	512,000	2,560	1,280	320	
Bits					
Data	8,192,000	40,960	20,480	5,120	16
Check	2,048,000	10,240	5,120	1,280	4
Total	10,240,000	51,200	25,600	6,400	20

20261

Figure 1-29. Disk Storage Organization of Data

2.1 DESIGN PHILOSOPHY

Much of the 1130 system logic design utilizes the characteristics of the multi-input flip-flop (FF) and its associated gated-inputs. A basic understanding of these circuits is helpful. This section includes a description of the multi-input FF and its gated-inputs.

Throughout the system, signal line names reflect the active state of the originating circuits. In many cases, the fall of a signal rather than the rise of a signal is the activating shift for other circuits. In the automated logic diagrams (ALDS), this is evidenced by a positive line entering a negative shift input.

The use of the multi-input trigger in the 1130 system allows the transfer of data from one register to a second register and the transfer of data in the second register back to the first register in one cycle. This principal is used in arithmetic and shift operations.

2.1.1 Multi-Input (MI) Flip-Flop

The MI FF (Figure 2-1) is used in clock-ring and register circuits and as a standard flip-flop. Direct-coupled emitter followers are used for the flipping action and inverters are used for the output functions. The FF may be connected for binary operation, single gated a-c input, dual gated a-c input, or d-c set input. Both in-phase and out-of-phase outputs are provided.

A-C Set Input: External RC and FTX transistors packs generate the gated input pulse. The set pulse is a 3v negative shift (+3v to 0v) with a minimum duration of 30 nsec. The gate must be conditioned (at 0v) for at least 90 nsec before arrival of the set pulse.

D-C Set Input: For d-c flipping, a down level of 0v is applied to the d-c set (reset) input. The down input signal must be at least 30 nsec in duration. For a-c set operation, the d-c input must be at up (+3v) level.

Binary Operation: The FF can be adapted for binary operation by making the following external connections:

Pin 11 to pin 11
Pin 4 to pin 4
Pin 9 to pin 9
Pin 5 to pin 5

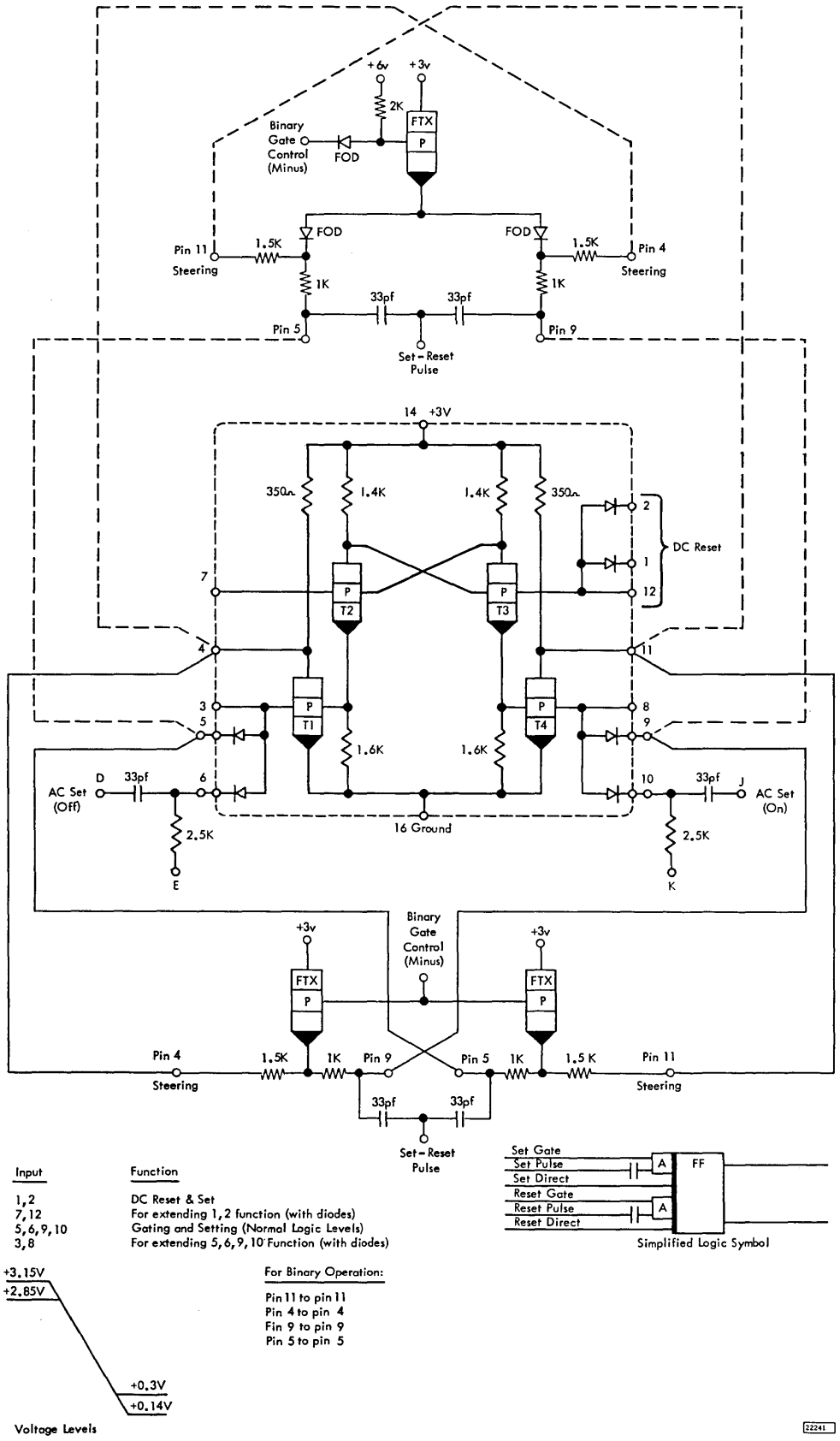
The binary input pulse is applied to the two a-c inputs. The binary gate may be connected to an external gate or tied to 0v.

Circuit Description

Assume that the FF (Figure 2-1) is off and is to be turned on with an a-c set pulse. In the OFF condition, T4 is off and pin 11 is at +3v; T1 is on and pin 4 is at 0v. T3 is in a state of low conduction and the T3 emitter voltage (about +0.2v) is insufficient to forward bias T4. T2 is in a state of heavy conduction and the T2 emitter voltage (about +0.7v) forward biases T1 and holds it on.

There are two versions of the binary operation R-C packs. One uses two FTX transistors and the other only uses one. The operating principle is the same for both. When the gate is at +3v, the FTX transistors are conditioned and the common point of the resistors is held at +3v and prevents the sample pulses from effecting the FF. When the gate is shifted to 0v, the FTX transistors are deconditioned and the binary sample can effect the circuit.

A down level of 0v is applied to the binary gate while the binary sample is still at +3v. The input capacitor charges to 3v during the conditioning period. When the set signal at pin G goes to 0v, a negative shift of 3v is seen on the emitter of T3. T3 immediately goes into heavy conduction and the reduced collector voltage switches T2 into a state of low conduction. The emitter of T2 goes from +0.7v to +0.2v and cuts off T1. As soon as the a-c input transient recedes, the T3 emitter voltage (+0.7v) holds T4 on.



Input	Function
1,2	DC Reset & Set
7,12	For extending 1,2 function (with diodes)
5,6,9,10	Gating and Setting (Normal Logic Levels)
3,8	For extending 5,6,9,10 Function (with diodes)

For Binary Operation:

Pin 11 to pin 11
Pin 4 to pin 4
Fin 9 to pin 9
Pin 5 to pin 5

Figure 2-1. Multi-Input (MI) Flip-Flop

It is important to note at this point, that should another gated a-c set pulse be applied to the ON input while the FF is on, the negative shift momentarily tries to turn T4 off. Consequently, a narrow positive pulse is generated at the OFF output (pin 11). Likewise, when the FF is off, a gated a-c set pulse applied to the OFF input causes a positive pulse at the ON output (pin 4).

2.2 CLOCK AND TIMING

- The 1130 system (3.6 μsec machine cycle) uses a 2.25 megacycle free-running oscillator for the basic pulse generator.
- Figure 2-2 and 2-3 show the clock, clock advance circuits, and a timing chart of the clock.
- Phase FF provides sub-clock-step pulses for clock advance generation and other system timing purposes.
- Advance FF is turned on by the first A phase after the start key is operated.
- Delay FF is turned on by the advance FF and the next A phase or by an initial program load request.
- Run FF is turned on by the delay FF or by an interrupt.
- Clock advance pulses occur at every A phase time after the run FF is turned on.
- The clock ring consists of eight triggers, T0 through T7.
- A basic machine cycle is one complete cycle of the clock ring, 3.6 μsec.
- A cycle is extended for various operations by holding on the T7 FF.
- Reference Maintenance Diagram: AA211.

The clock is turned off with the T7 FF ON. Also, when certain arithmetic and shift operations require more steps than one cycle provides, the clock is stepped to T7 and then clock advances are prevented, holding T7 on while the phase FF output.

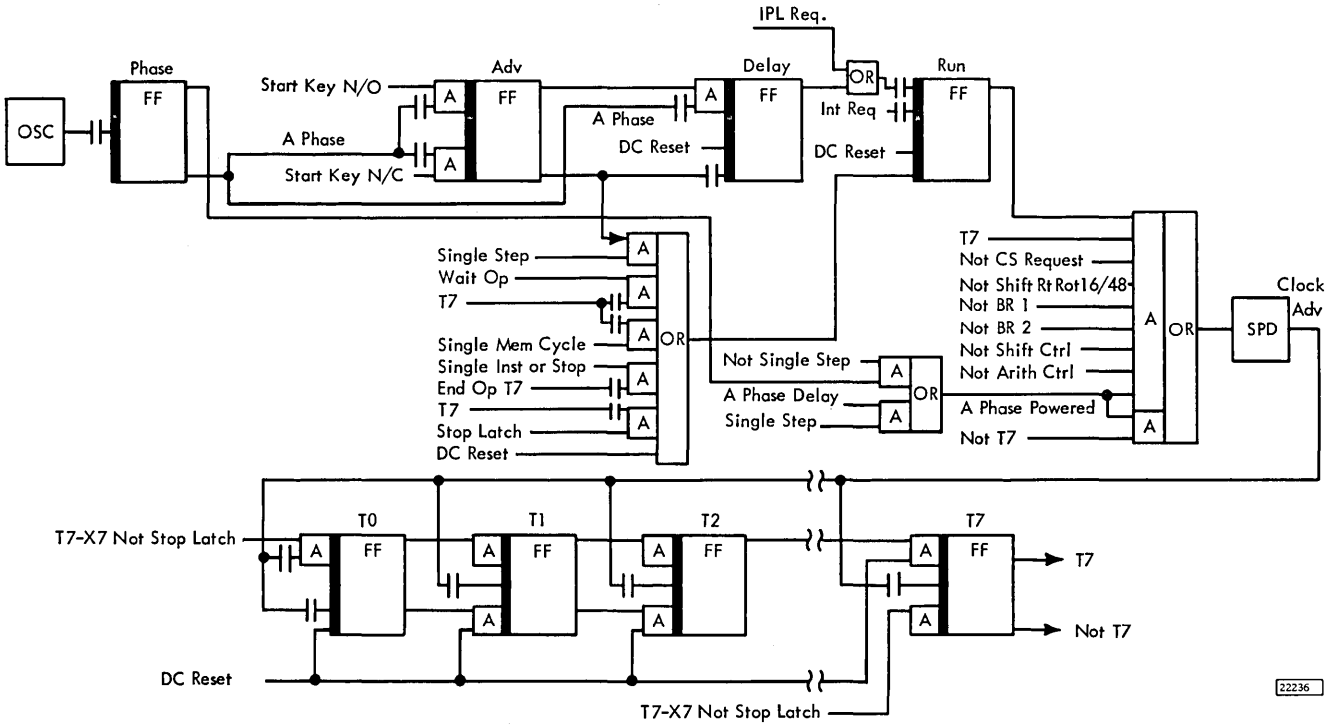


Figure 2-2. Clock Control

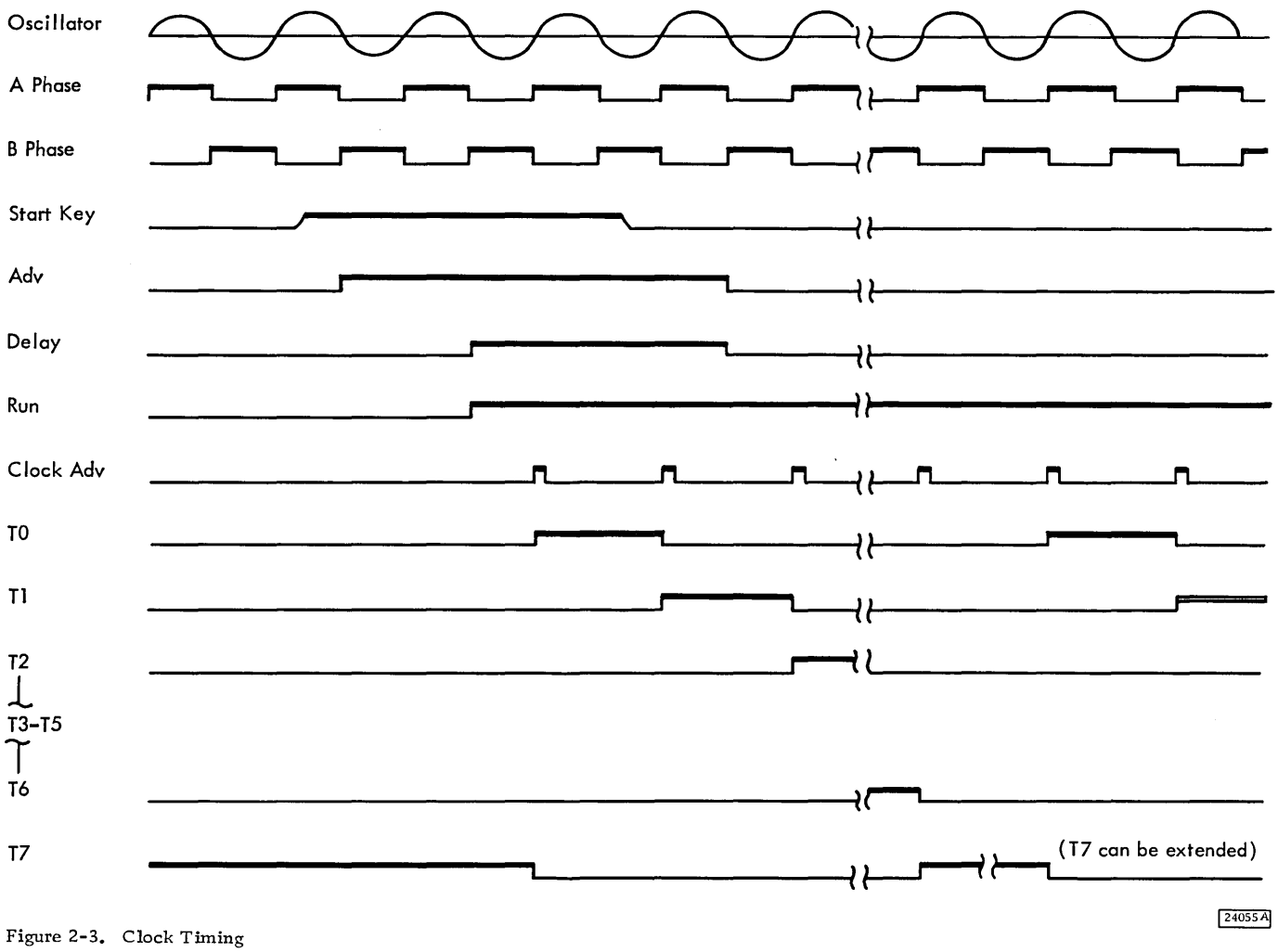


Figure 2-3. Clock Timing

provides timing pulses for the operation. Advancement from T7 requires that run FF be on and that any of the clock-extending conditions be absent. Once T7 is turned off, the clock advances through T6 unless the CPU is in the single step mode.

The clock is stopped at T7 after one cycle (basic or extended) if the CPU is in the single cycle mode or performing a wait operation, or if the stop key is operated. The clock is stopped at the last T7 step of an operation (end op T7) if the CPU is in the single instruction mode.

Single step mode prevents clock advances by blocking the A phase pulse except when the start key is operated.

2.3 CYCLE TIMER

- Composed of seven flip-flops: I-1, I-2, IX, IA, E, E-1, and E-3 (Figures 2-4 and 2-5).
- Specifies the type of cycle being performed.
- Gates the necessary circuits for that type of cycle.
- Functions of the various cycles are described in the I and E cycles section of the manual.
- Reference Maintenance Diagram: AA211.

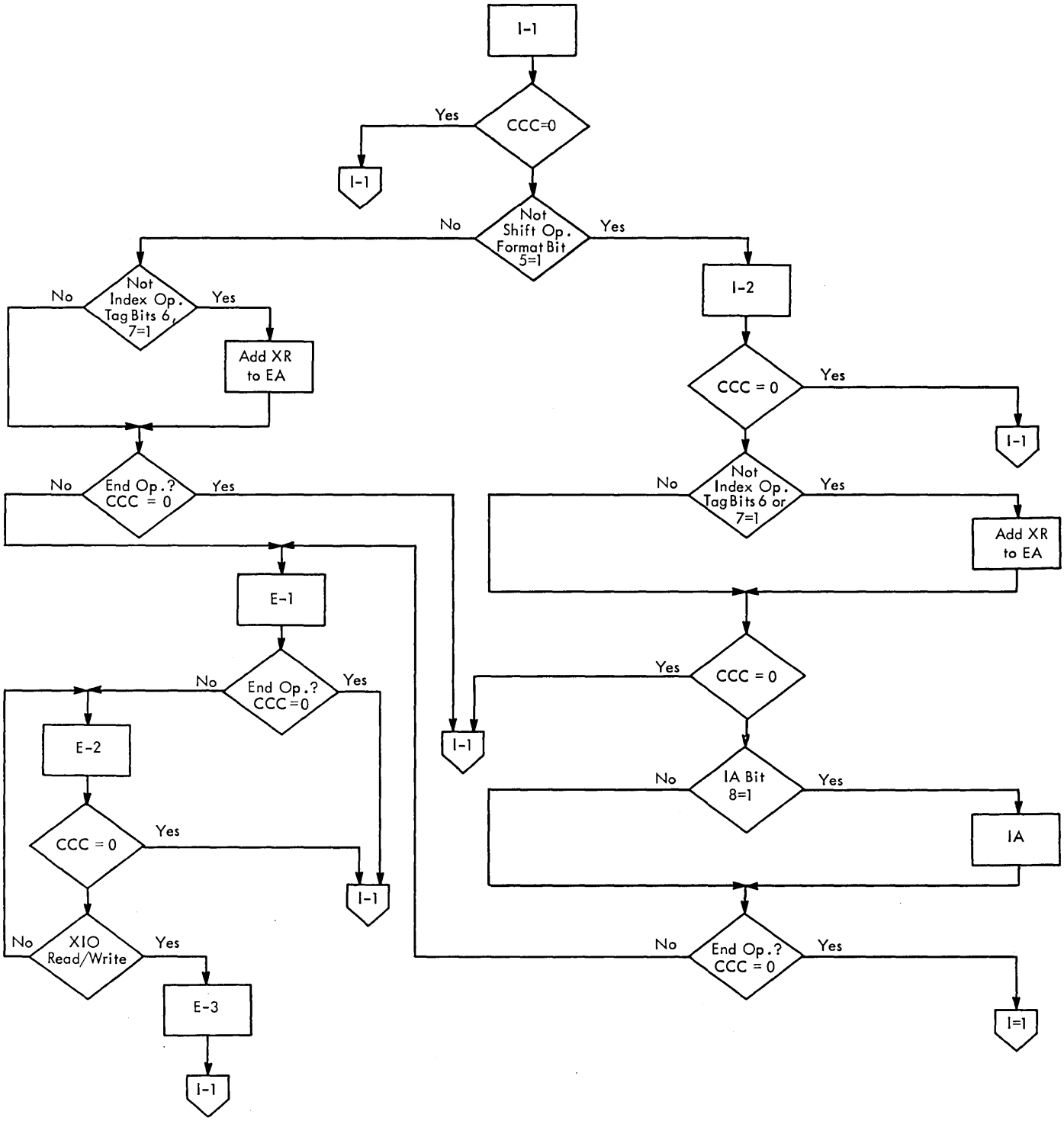


Figure 2-4. I and E Cycles

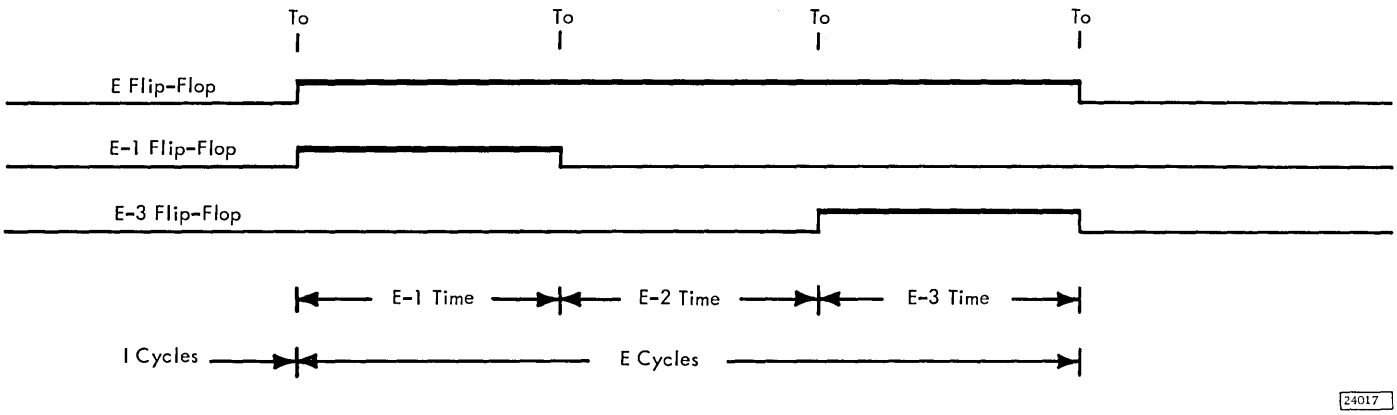


Figure 2-5. Development of E Cycle Times

2.3.1 I-1

- DC reset on to begin every operation with I-1 cycle.
- Turned on after the last cycle of every operation (end op, T0).
- Turned off at the next T0 if not end op.

2.3.2 I-2

- Turned on at T0 following I-1 if the format bit (bit 5 of instruction read during I-1) is a one, and not shift operations.
- Turned off at next T0.

2.3.3 I-X

- Turned on at T0 following I-1 or I-2 if either tag bit is a one (bits 6 and 7), and not index operations (read, store, or modify index).
- Turned off at next T0.

2.3.4 IA

- Turned on at T0 of a two word instruction following I-2 or IX (bit 5 = 1) if the IA bit (bit 8) is a one.
- Turned off at next T0.

2.3.5 E

- Turned on at T0 following any I cycle if the following do not exist: End op or conditions to turn on I-2, IX, or IA.
- Turned off at T0 with end op. Remains on for E-1, E-2, and E-3 times.

2.3.6 E-1

- Turned on with E flip-flop.
- Turned off with next T0.

2.3.7 E-2

- The E-1 off condition is ANDed with the E flip-flop output to provide E-2 time.
- Turned off within the next T0.

2.3.8 E-3

- Turned on at T0 with an XIO read/write instruction and E-2 time.
- Turned off at the next T0.

2.4 CORE STORAGE

- The core storage unit is self-contained on a single SLT board.

- Figures 2-6 and 2-7 are diagrams of the data flow and control of core storage.

- The core storage unit cycle time is 3.6 μsec.

2.4.1 Magnetic Core Theory

A magnetic core is a small doughnut-shaped ring that is uniformly constructed of ferrite particles bonded together by a ceramic material. The ferrite particles have good magnetic properties and the core has a high retentivity of the magnetic flux lines after the magnetizing force is removed. It is this property of retentivity that makes a magnetic core useful as a storage device.

The operation of a magnetic core can best be described by reference to the hysteresis curve, Figure 2-8. This curve is a plot of the relationship between a magnetizing current (I_m) and the flux density.

A magnetic core is capable of maintaining indefinitely one of two stable magnetic states, either at point A or at point D on the hysteresis curve. Because the core has two stable states, it can be used as a binary storage device. At point A the core has a residual flux in a negative direction, and at point D a residual flux in the positive direction. These two directions can be arbitrarily assigned as binary "zero" and binary "one," respectively.

I_m is the amount of current necessary to change the state of the core. Plus I_m is that amount of current flowing in one direction, and minus I_m the same amount of current flowing in the opposite direction.

On the hysteresis curve, it can be observed that a magnetizing current of plus I_m will change the magnetism of the core from point A, a binary zero, to the magnetic saturation value in the positive direction at point C. When the current is removed, the total amount of magnetization drops back to point D (binary one). If, instead of full plus I_m, a current of plus 1/2 I_m were applied, the flux would change only the small amount from point A to point B on the curve, and when the current returned to zero, the flux would return to its original value at point A.

A reverse current, minus I_m, develops flux of opposite polarity and changes the magnetic field of the core from point D to the magnetic saturation value in the negative direction at point F. The total amount of magnetization drops back to point A (binary zero) when the driving current is removed.

When a matrix of magnetic cores is constructed to store multiple bits of information, a specific core is selected (addressed) by the coincidence of 1/2 I_m flowing through each of two wires threaded through the core, with a total effective current of full I_m. The state of the core is determined by the direction of the current flowing through these wires. A current I_m is passed to store a "1" in the core; that is called "writing" into cores. A current I_m is passed in the opposite direction to change a stored value of "1" to a "0" and can be considered a current of minus I_m; this is called "reading" a core.

Reading a memory core depends on a system of sensing the state of the flux field within the core. When the core contains a "0" and a current of minus I_m read current is passed through the core, the field changes from point A to point F on the hysteresis curve, which is a very small change in total flux density. If the core contains a "1" and a minus I_m read current is passed through the core,

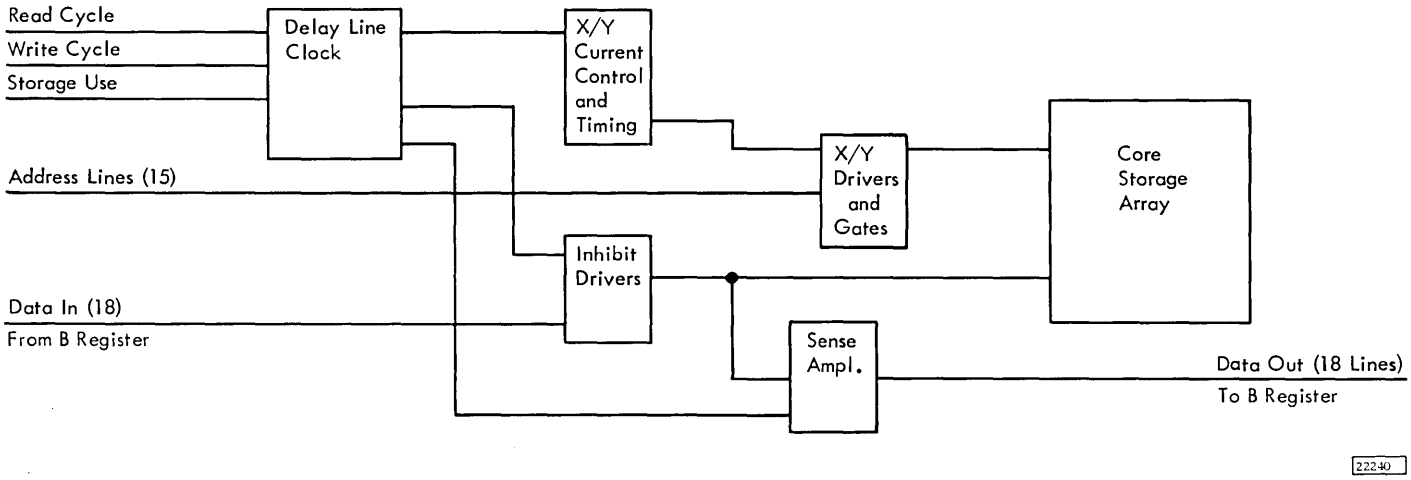


Figure 2-6. Core Storage Data Flow and Control

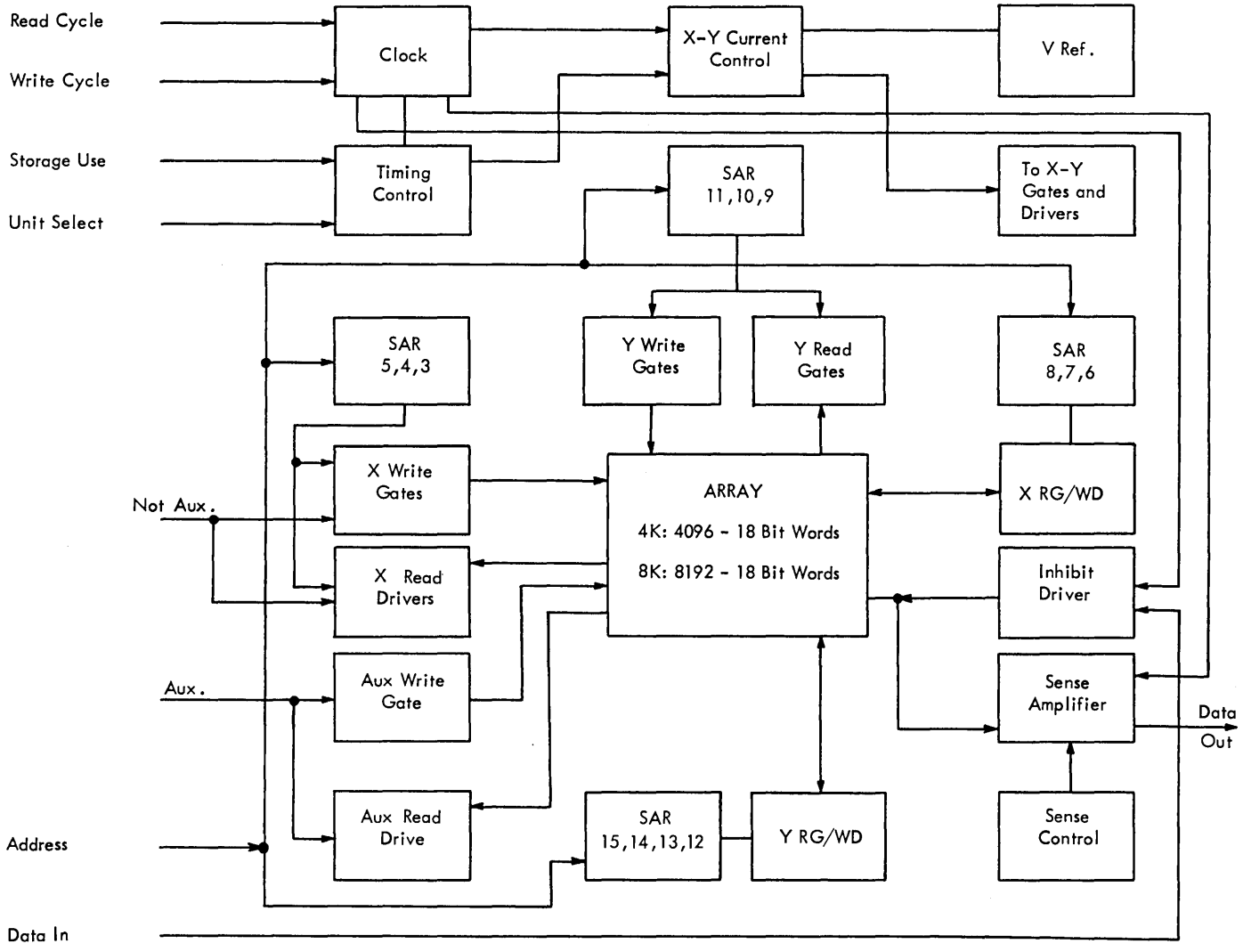


Figure 2-7. Core Storage Data Flow

the field changes from point D to point F and a large change in the flux field occurs. A third wire, called the "sense" wire, is threaded through the core to recognize these changes in the magnetic field. Circuits are used to discriminate against low value "0" signals and amplify the large signal that results when a "1" is read from a core.

2.4.2 Addressing

- The address lines from the M register are decoded to condition X and Y read/write drivers and gates.
- One active X R/W line and one active Y R/W line coincide at one core in each plane. (In a 4 k array, 9 planes, coincidence occurs at two cores in each plane, Figure 2-9.)
- Each active address line carries half-select current.
- Coincidence of Y half-select and X half-select currents address the 18 cores (one word).
- Reference Diagrams (RD): SD 011, SD 012, SD 041, SD 042. Figures 2-10, 2-11, 2-12, and 2-13.

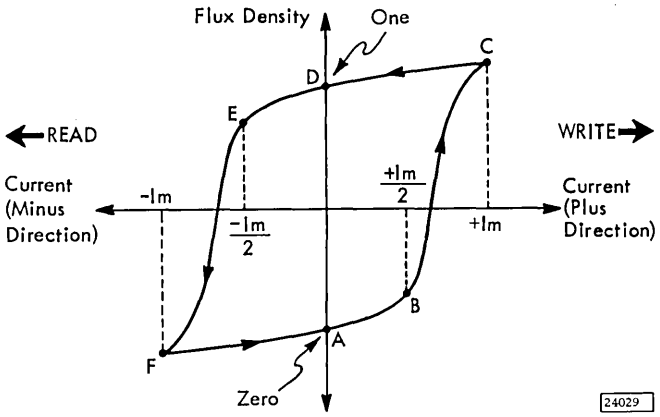
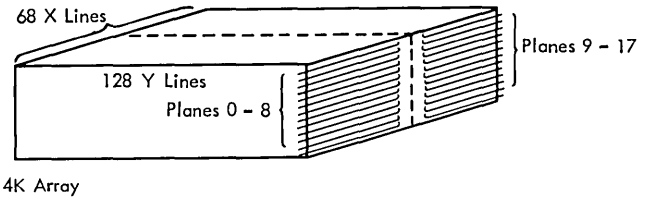


Figure 2-8. Hysteresis Curve



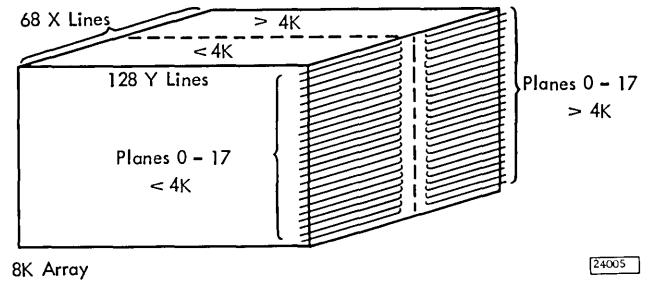


Figure 2-9. 4K and 8K Arrays

2.4.2.1 Addressing 8K Array

The magnetic cores are arranged in matrices of 128 x 64 cores, called planes (Each plane is actually 128 x 68, providing 512 positions of auxiliary storage that are not program addressable.). The 8 k array consists of 18 of these planes with each plane assigned to one bit position of a core storage word. Corresponding core positions in each plane are addressed simultaneously by the X and Y drive lines to select one 18-bit word. Since there are 8192 cores in each plane, the 8 k array has a capacity of 8192 words.

The 128 x 64 matrix in reference diagram (part of core ALD's) SD 041 represents one plane of the 3.6 μsec core storage, 8 k array. The one core that is shown is selected by the address in the 13 positions of the address register (M register).

On the A side, the 64 X drive lines, which run through all planes, are divided into eight groups of eight lines each. One group (000) is activated by a decode of M register positions 3, 4, and 5. On the C side of the array, one line in each group of eight is activated by a decode of M register positions 6, 7, and 8 (111). Thus, a single X line is activated and carries half-select current.

At side B, the 128 Y lines which run through all planes are divided into eight groups of 16 lines each. One group of 16 lines is activated by a decode of M register positions 9, 10, and 11 (111). At side D, one line in each group of 16 is activated by a decode of M register positions 12, 13, 14, and 15 (0000). A single Y line is thereby activated and carries half-select current.

In each plane, the one core at the intersection of the active X and Y drive lines receives full-current and is the only core in the plane that is selected.

2.4.2.2 Addressing 4K Array

One plane of the 3.6 μsec core storage array is shown in reference diagram SD 042. The 4 k and 8 k arrays occupy the same space in the SLT board. The core planes used in the 4 k array are identical to those used in the 8 k array but only nine planes are used. For addressing purposes, each 128 x 64 plane is divided into two 128 x 32 half-planes. Each of the 18 half-planes is assigned to one bit position of the core storage word. Since there are 4,096 cores in each half-plane, the 4 k array has a capacity of 4,096 words. (An auxiliary storage of 256 words is actually part of the 4 k array.) The 32 X drive lines are divided into four groups of eight lines each (one-half as many groups as in the 8 k array). The X lines run through the nine B half-planes, then loop back through the nine D half-planes. One group is activated by a decode of M register positions 4 and 5 (010), position 3 is not used with 4 k core storage. At the other end of the X lines, one line in each group is activated by

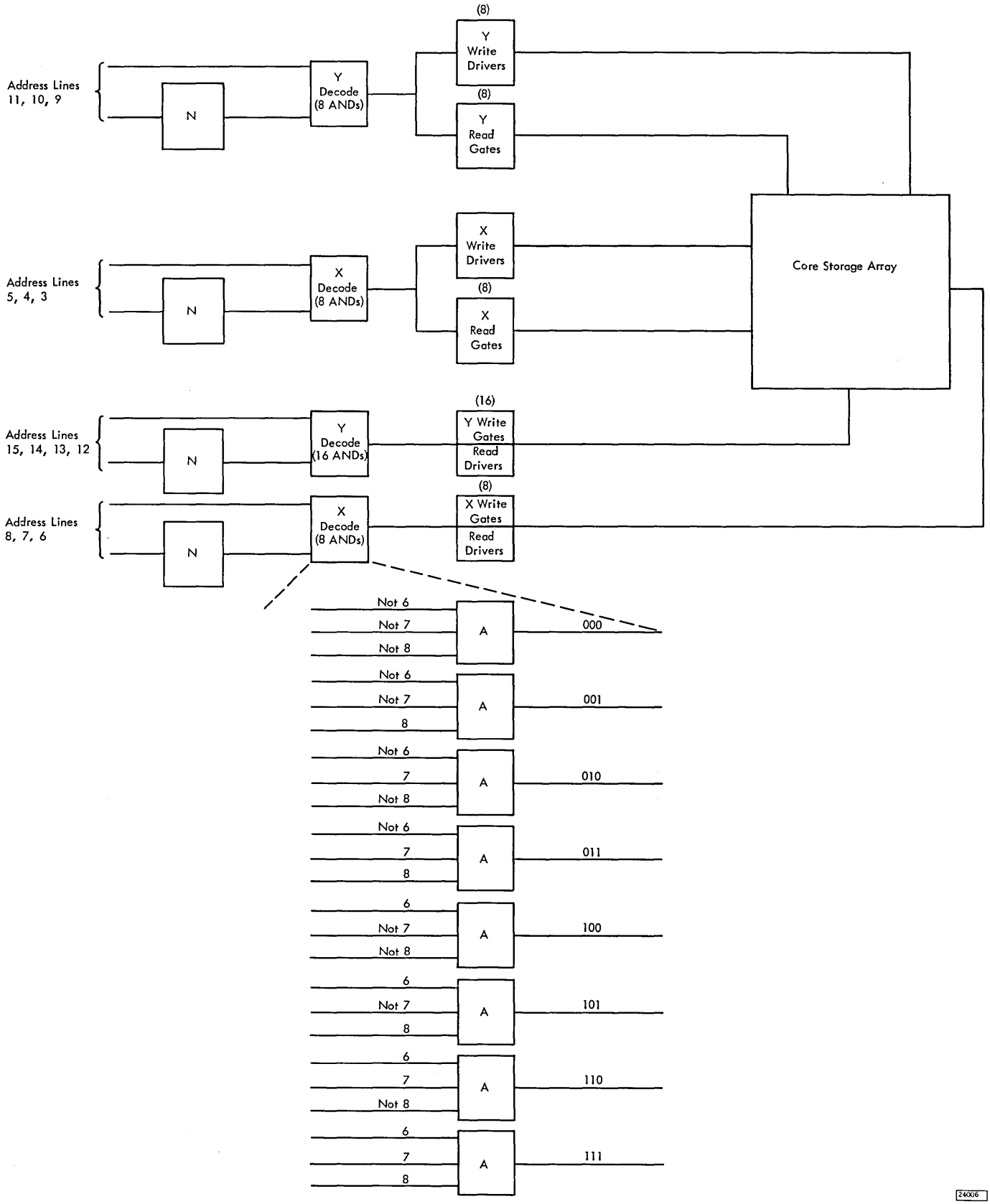


Figure 2-10. Address Decode

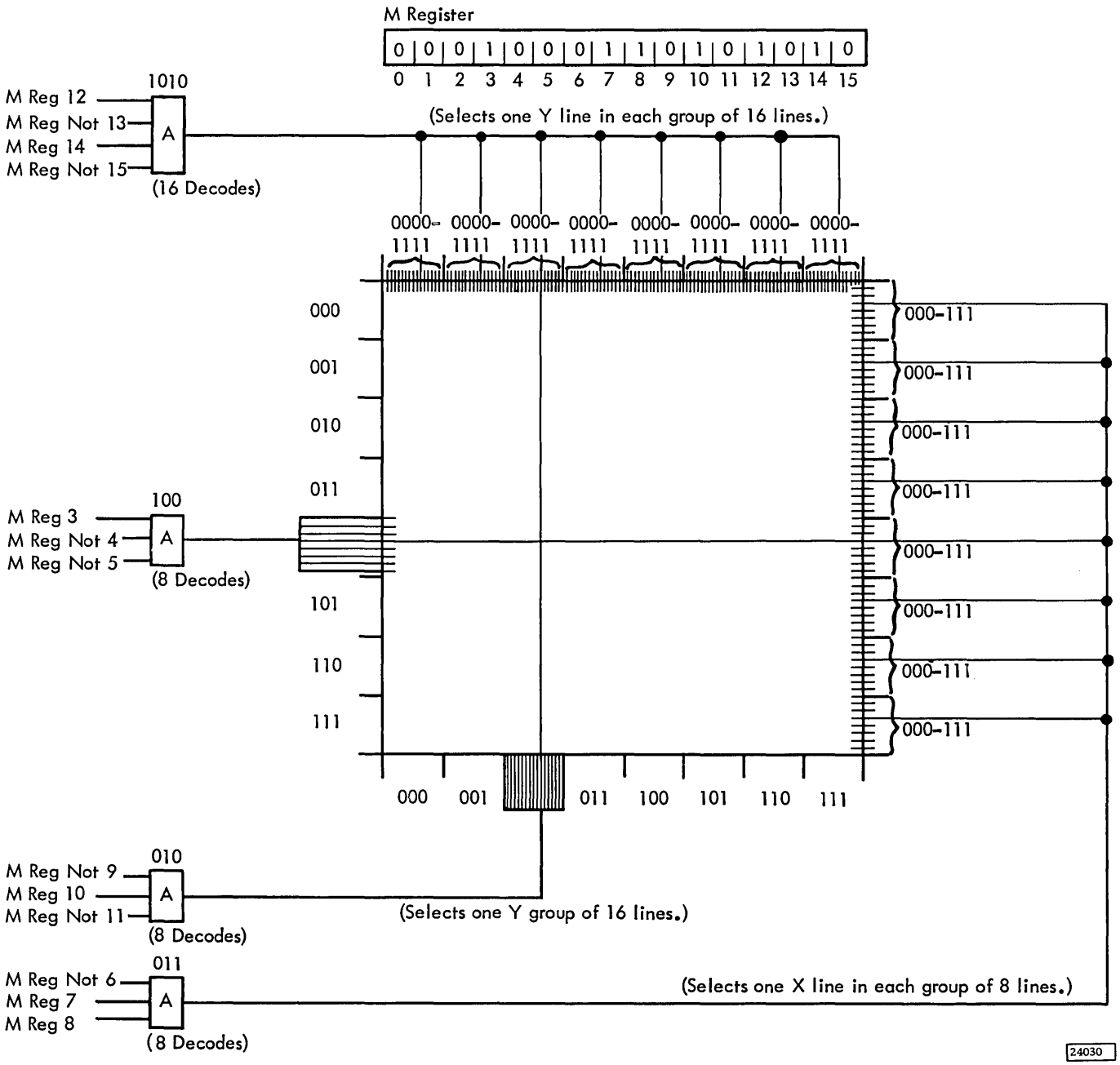


Figure 2-11. Addressing Scheme

a decode of M register positions 6, 7, and 8 (111). A single X line is thereby activated and carries half-select current. The 128 Y drive lines are divided into eight groups of 16 lines each, as in the 8 k array, and run through all nine planes. One group of 16 lines is activated by a decode of M register positions 9, 10 and 11 (111). On the D side of the array, one line in each group of 16 is activated by a decode of M register positions 12, 13, 14, and 15 (0000). This active Y line intersects the active X line at two cores in each plane; therefore, nine planes provide 18 bits for each of 4096 words.

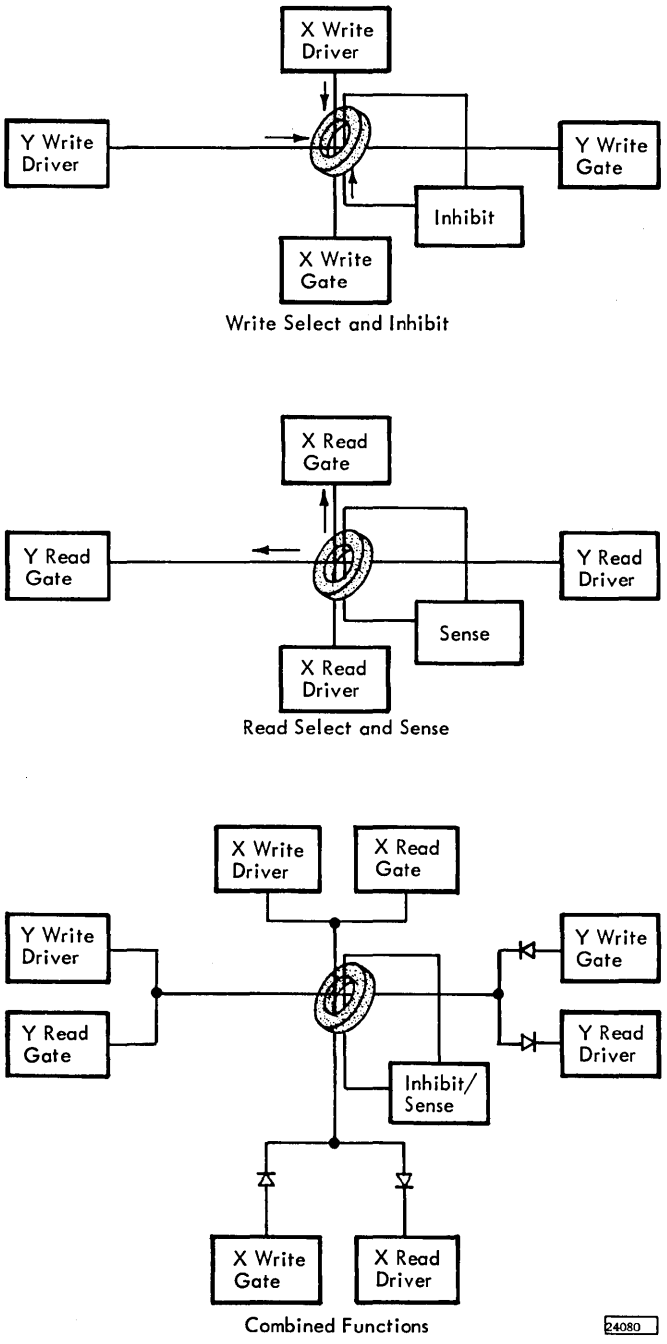


Figure 2-12. Core Storage Write and Read

2.4.3 Core Storage Arrays

- Core storage arrays are available in two sizes, 4096 (4 k) words and 8192 (8 k) words. Figure 2-9.
- Each word consists of 18 bits.
- Both arrays consist of core planes in a 168 x 68 matrix (168 x 64 program addressable).
- The 8 k array contains 18 physical planes, each of which contains one bit-position for each word.
- The 4 k array contains 9 physical planes, each of which contains two bit-positions for each word.
- Reference Diagrams: SD 012, SD 021, SD 071, SD 072, SD 081, SD 082.

The core storage array is mounted on a SLT large board. The 4 k unit consists of 9 planes and the 8 k unit consists of 18 planes.

On the 8 k array the X and Y drive lines address one bit in each plane for one word. On the 4 k array and the X and Y drive lines address two bits on each plane for one word.

The array consists of a diode board, the core planes, a bus board and connecting pins.

The array is mounted on the large board with four holding screws. Electrical connections are made to the pins which go through the large board, by jumper blocks to the circuit pins of the large board.

A 4 k system may be changed to an 8 k system by unplugging the pin connectors, removing the 4 k array, replacing it with an 8 k array and installing the connectors plugging in twelve new SLT cards.

2.4.3.1 Eight-K Array

The 8 k core storage array consists of a bottom board, 18 core planes, and a diode board. The bottom board plugs into the SLT board and provides connection from the SLT circuits to the X, Y, and inhibit/sense. Reference diagram SD 071 shows the land pattern for the terminals; the connectors are not shown. The even addressed Y write driver and read gate lines go from the connectors, through the land pattern, to terminals B (4)-(35) and B(46)-(77); the odd addressed lines go to terminals D(4)-(35) and D(46)-(77). The even addressed X write driver and read gate lines go to terminals C(2)-(17) and C(20)-(35); the odd addressed lines to terminals A(2)-(17) and A(20)-(35). The write driver/read gate windings of bit 17 plane connect directly to the bottom board terminals.

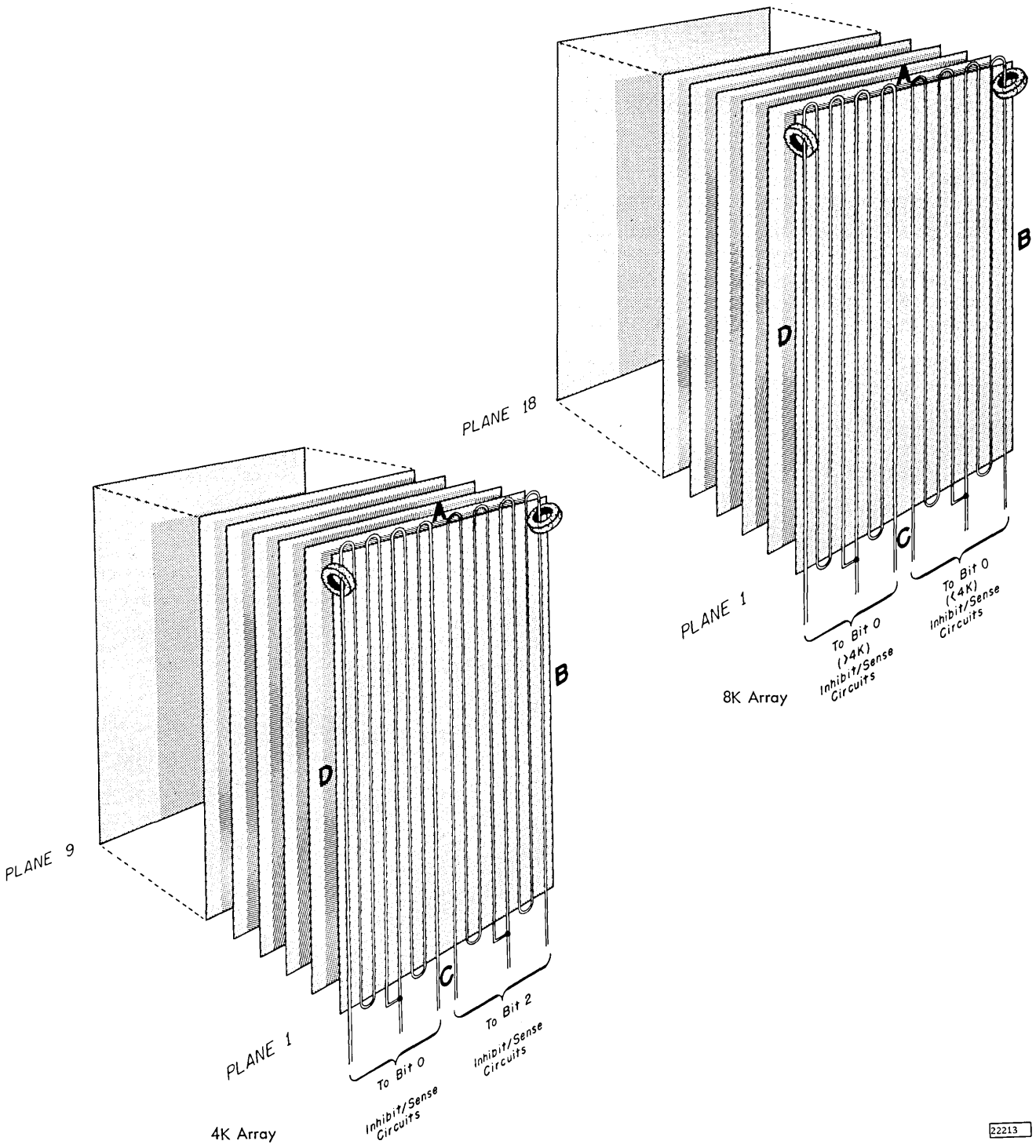


Figure 2-13. Inhibit/Sense Lines - 4K and 8K

The X and Y write gate and read driver lines go from the connectors, through the bottom-board land-pattern, to the following terminals: even addressed lines to B(1)-(3), B(36)-(45), and B(78)-(80); odd addressed lines to D(1)-(3), D(36)-(45), and D(78)-(80). The lines then go along the side of the array to the corresponding terminals of the diode board. The diode board land pattern (reference diagram SD 081) connects the lines to the common terminals of the diode packs. From the individual diode packs. From the individual diodes, the land pattern connects to the terminals at the sides of the diode board.

Diode Board Terminals. Refer to reference diagram SD 081 and note that the storage address register is divided into X and Y high order and low order positions. If the X portion of the specified address is even (pos. 8 = 0), the active X terminal is on side C of the array. If the X portion of the address is odd (pos. 8 = 1), the active X terminal is on side A of the array.

If the Y portion of the address is even (pos. 15 = 0), the active Y terminal is on side B of the array; if odd (pos. 15 = 1), the active Y terminal is on side D.

Locate the specified terminals by finding the group of lines associated with the X and Y high order positions of the address. The relative position of the line within the group is the same for all low order values and is identified in the enlarged drawing (within broken lines) for each side of the diode board.

Array Windings. The diode board terminals can be related to the terminals on each plane and the lines can be traced through the array by referring to Figure 2-14. This figure shows the relative location of the 18 cores that form the word at core storage address hex 0000. Note that the Y address line for this address enters bit 0 plane at side B and is activated by positions 9, 10, and 11 (Y high order) of the address register. The X line enters the bit 0 plane at side C and is activated by positions 3, 4, and 5 (X high order) of the address register. The two lines go through every plane, intersecting at the core at location hex 0000 in each plane.

At bit 17 plane, the Y line exits at side B and is activated by positions 12-15 (Y low order) of the address register. The X line exits at side C and is activated by positions 6, 7, and 8 (X low order) of the address register.

The X and Y lines for an odd address enter bit 0 plane and exit bit 17 plane at the sides of the array opposite those for an even address, as shown in the insert in Figure 2-14.

Bottom Board Terminals. From the terminals of bit 17 plane, the lines go to the bottom board terminals (reference diagram SD 071), through the land pattern to the connectors (not shown), and through the SLT board land pattern to the drivers and gates (reference diagram SD 012).

The bottom board also provides connection from the inhibit/sense connectors (not shown), that plug into the SLT board, to terminals (1S)-(54S) at side A and C.

2.4.3.2 Four-K Array

The 4 k array consists of a bottom board, 9 core planes, and a diode board. As described in the Addressing section, the core planes used in the 4 k and 8 k arrays are the same. The Y windings in the two arrays are connected in the same way but the X windings are connected differently. Because each physical plane must provide two bit-planes, the X windings go through the B half of each plane in the array and then through the D half. Reference diagram SD 072 shows the land pattern of the 4 k bottom board. Note that at sides A and C the four groups of X lines in the B half are common to the four groups in the D half.

The bit 16 and 17 X lines shown in Figure 2-15 connect to the bottom-board terminals at side A and C. Side A of the bottom board land pattern thus provides the "jumpers" for the even-addressed X lines between bit 16 and 17 and side C provides the "jumpers" for the odd-addressed X lines. The reference diagram SD 072 shows the land pattern for the terminals; the connectors are not shown.

The X and Y write driver and read gate lines go from the connectors, through the bottom-board land-pattern, to terminals at the sides of the array. The write driver/read gate windings of bit 16 and 17 planes connect directly to the bottom board terminals.

The X and Y write gate and read driver lines go from the connectors, through the bottom-board land-pattern to the indicated terminals, and along the sides of the array to the corresponding terminals of the diode board. The diode-board land-pattern (reference diagram 082) connects the lines to the common terminals of the diode packs (reference

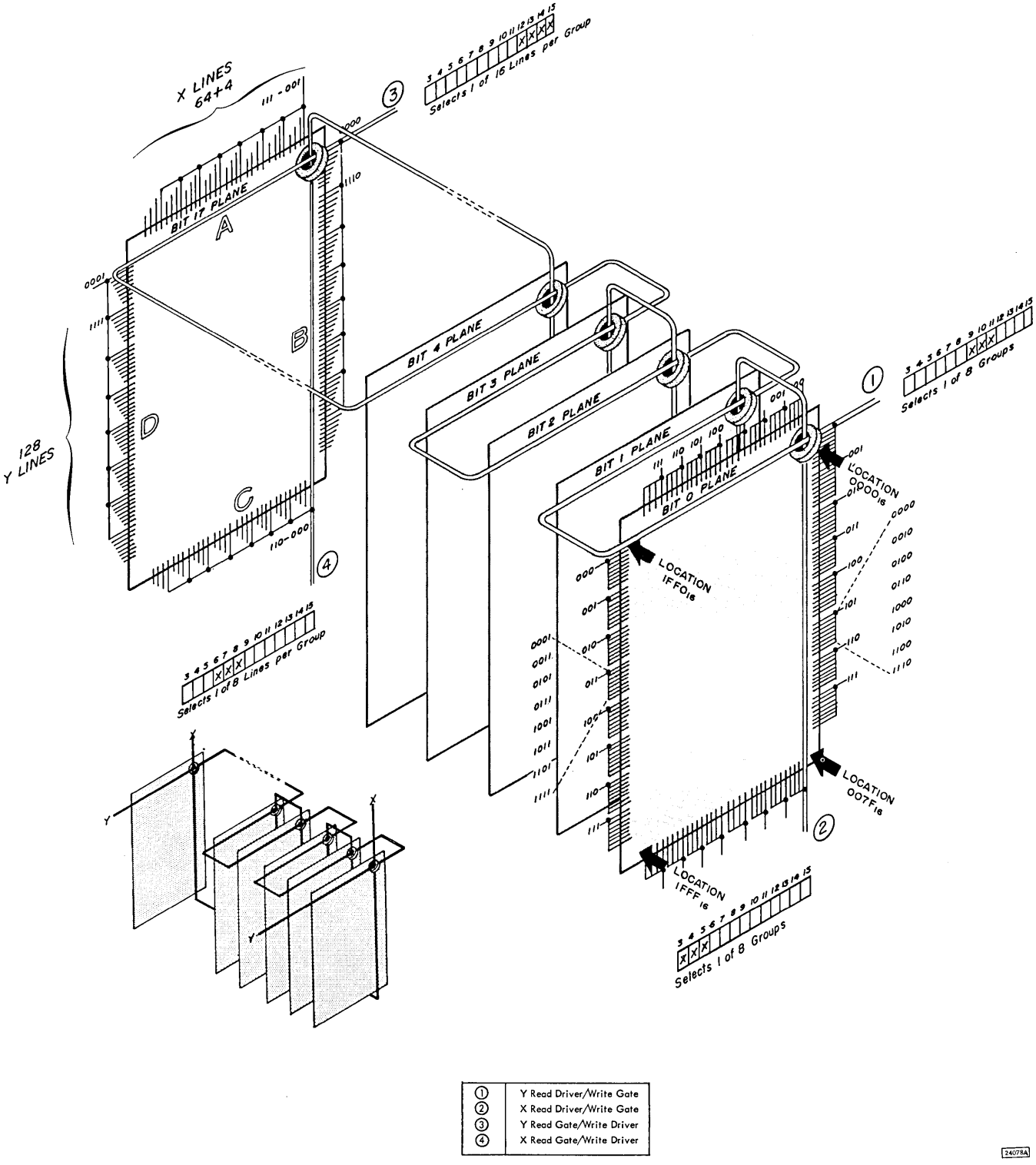


Figure 2-14. X and Y Select Lines for Location 0000-8K Array

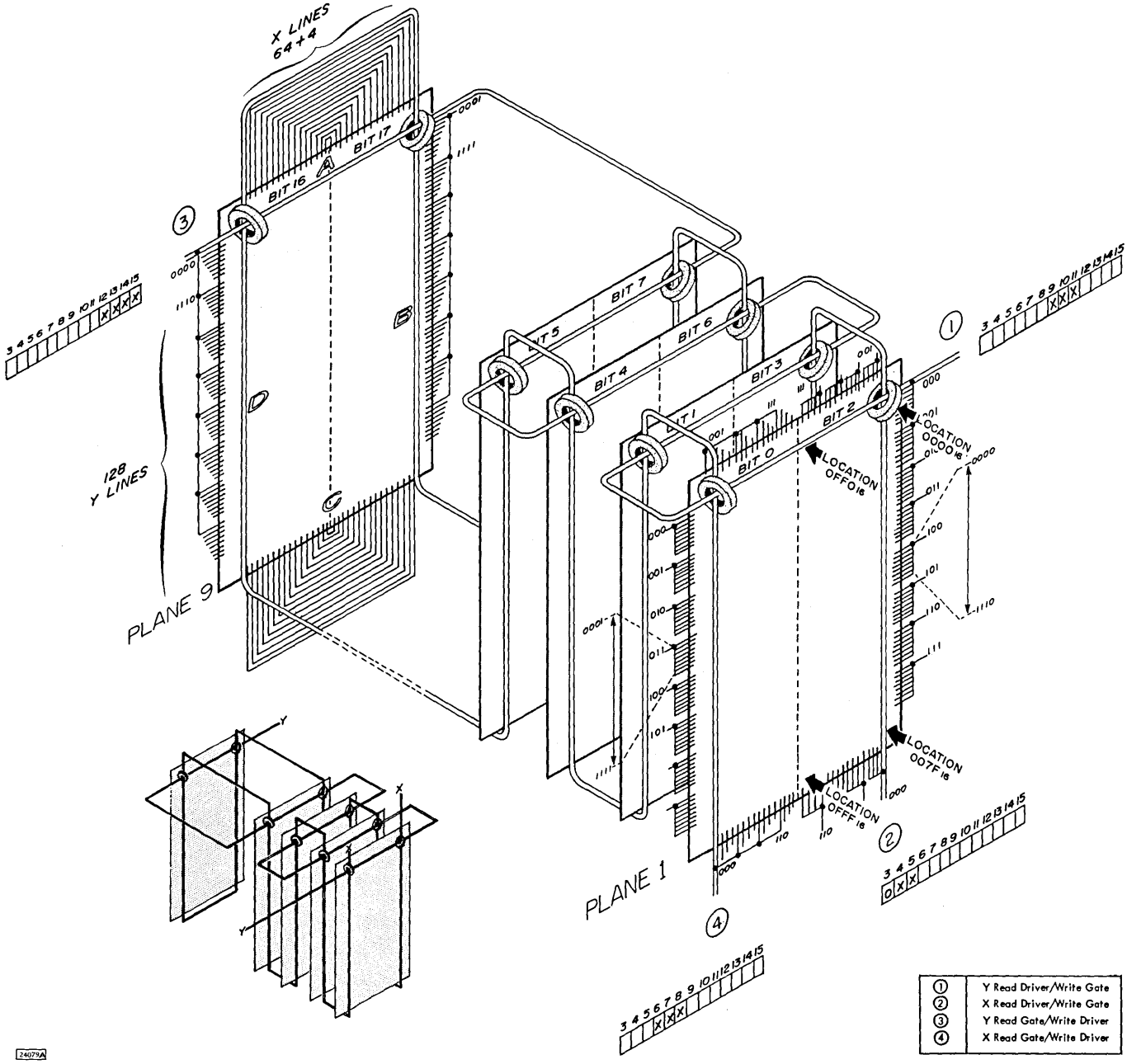


Figure 2-15. X and Y Select Lines for Location 0000-4K Array

diagram SD 012). From the individual diodes, the land pattern connects to the terminals at the sides of the diode board.

Diode Board Terminals. The function of the 4 k diode board (reference diagram SD 082) is the same as the previously described 8 k diode board. One-half of the X read driver/write gate diodepacs are not used because there are half as many X drive lines in the array. The terminals are numbered differently and sides A and C are used differently.

The B half of sides A and C provide connection from the diodes to the X read driver/write gate lines in groups of eight lines as do the corresponding terminals of the 8 k diode board. The D half of sides A and C provide connection from the other end of the lines to the X read gate/write driver lines within the groups.

Array Windings. Figure 2-15 illustrates the 4 k array bit-plane layout, the paths of the X and Y windings, and the relation of the windings to the terminals of the bottom board and diode boards. The windings within the planes are the same as those in the planes of the 8 k array; it is the bottom board and diode board that cause the array to function as 4096 eighteen-bit words.

The even-addressed Y lines, from the read driver/write gate circuits, enter plane 1 through the Bit-2-plane half (side B), go through every plane, and exit plane 9 at the bit-16-plane half (side D). Odd-addressed Y lines (Figure 2-15 inset) enter plane 1 at side D and exit plane 9 at side B. Even-addressed X lines, in four groups of eight each, enter plane 1 at the B half of side C, go through the B half of every plane, and loop back (by the bottom-board land-pattern) to the D-half of plane 9. The lines then go through the D-half of every plane and exit plane 1 at side C to connect to the diode board. The lines are connected, through the diode board and the bottom board, to the X read gate and write driver circuits.

Odd-addressed X lines (Figure 2-15 inset) enter plane 1 at side A, go through every plane, loop back (by side C of the bottom board), and exit plane 1 at side A.

2.4.4 Reading/Writing

- Sense/inhibit winding is common to every core in one half of a plane, in a 4k array.
- In the 8k array two sense amplifiers and two inhibit drivers operate in parallel for each plane.
- During read time, a pulse caused by the "flipping" of the selected core is conducted to the sense amplifier.
- The strobe pulse conditions the sense amplifiers at the optimum time to amplify the sensed pulse.
- During write time, inhibit current prevents the selected core from "flipping".
- Inhibit time conditions the inhibit drivers.
- Sense and inhibit circuits are divided into less than and more than 4k by the M 03 address line.
- Reference diagrams: SD 011, SD 043, SD 044, SD 051. Figures 2-12 and 2-13.

Each of the 18 (8k) planes has a sense/inhibit winding that passes through every core in that plane. This winding is effectively in parallel with the X drive line.

On the 4k array the sense/inhibit winding is split for each half of each of the 9 planes.

During read time, (SD 043) if a core "flips" from a "one" to a "zero" by having a coincidence of X and Y read current, the resulting magnetic field induces a pulse on the sense winding. This sense pulse is conducted to the sense amplifier (SD 051) associated with that plane and is set into the corresponding position of the B register. Because the X and Y current coincides at all 18 cores at one core storage location, the contents of the 18 selected cores (one word) is set into the 18 positions of the B register.

During write time, (SD 044) X and Y write current (opposite direction from X and Y read current) causes any selected core that is at a "zero" state to "flip" to a "one" state. Therefore, if it is not prevented, all 18 cores at the core storage location where X and Y write currents coincide, will be set to a "one". The sense/inhibit winding (SD 051) is parallel to the X winding. Therefore, if a current equal to, but opposite in direction to, the X current flows in this winding, it will cancel the effect of the X current and the selected core in that plane will not

be set to a "one". This cancelling current is called inhibit current and is controlled by the output of the B register.

2.4.5 Timing Circuits

- Read drivers and gates are conditioned at T0 time for a duration of approximately 1 μsec.
- Write drivers and gates are conditioned at T4 time for a duration of approximately 1 μsec.
- Y read/write time is delayed to allow X drive line noise to subside.
- Emitter strobe deactivates the first stage of the sense amplifiers during write time.
- Sense strobe gates data to the output stage of the sense amplifiers during write time.
- Inhibit time gates data from B register to the inhibit drivers.
- Reference Diagram: SD 031.

Read drivers and gate are conditioned by timing pulses during read cycle time (T0 through T3). Write drivers and gates are conditioned by timing pulses during write time (T4 through T7). Further timing of the drivers and gates is necessary due to the presence of noise at the rise of the X drive pulse. This noise is a current spike resulting from the discharge of core capacitance. The sense windings are paralleled to the X drive windings; therefore, if the X and Y drive currents occurred simultaneously, the noise would be sensed.

The X drivers and gates are conditioned by long time. After the X drive line noise has subsided, the Y drivers and gates are conditioned by short time.

2.4.5.1 Core Storage Clock

- A time delay "clock" circuit (Figure 2-16) developes the necessary timing pulses for addressing, reading from, and writing into core storage.
- Long time - A time delay circuit provides a one μsec pulse at T0 (read cycle) and at T4 (write cycle).
- Short time - a latch provides a pulse that begins after the rise of long time and ends with the fall of long time.
- Strobe - A single shot, activated by short time, provides a pulse that is ANDed with read cycle and an address line to gate the sense amplifiers.
- Inhibit time - long time and write cycle are ANDed to develop inhibit time.

2.4.5.2 Time Delay Circuit

The time delay circuit (Figure 2-17) is used to develop short time for noise rejection. Read cycle and write cycle are connected to opposite ends of a one μsec inductive delay line. The low resistance of the delay line causes the levels at the inputs (points 1 and 2) to balance. For example, if read cycle is 0 volts and write cycle is +3 volts, points 1 and 2 balance at +1.5 volts.

When read cycle and write cycle change levels (at T0 and T4 time), the input points reflect the respective levels for one μsec then balance again. The same effect is evident at the taps (points 3 and 4) except for a shorter duration.

2.4.6 Current Control

- X/Y driver and gate control line development is shown in Figure 2-10.
- X/Y drive current is temperature compensated.
- Reference Diagrams: SD 043, SD 044.

X/Y driver sink (termination of a current source line) current is developed by combining the X/Y driver and gate control lines with a temperature compensated voltage reference (V Ref). Temperature affects the amount of current required in the X and Y drive lines; a decrease in temperature requires an increase in current for satisfactory operation. A thermistor regulates the sink current to provide temperature compensation.

SD 043 and SD 044 show the current control for read and write operations. The drive and gate control circuits are current switching type circuits. The current will flow in one of the three transistors at all times. When not in a storage cycle, the current flows in the center transistor, biased by the 1.4 voltage. When a storage cycle

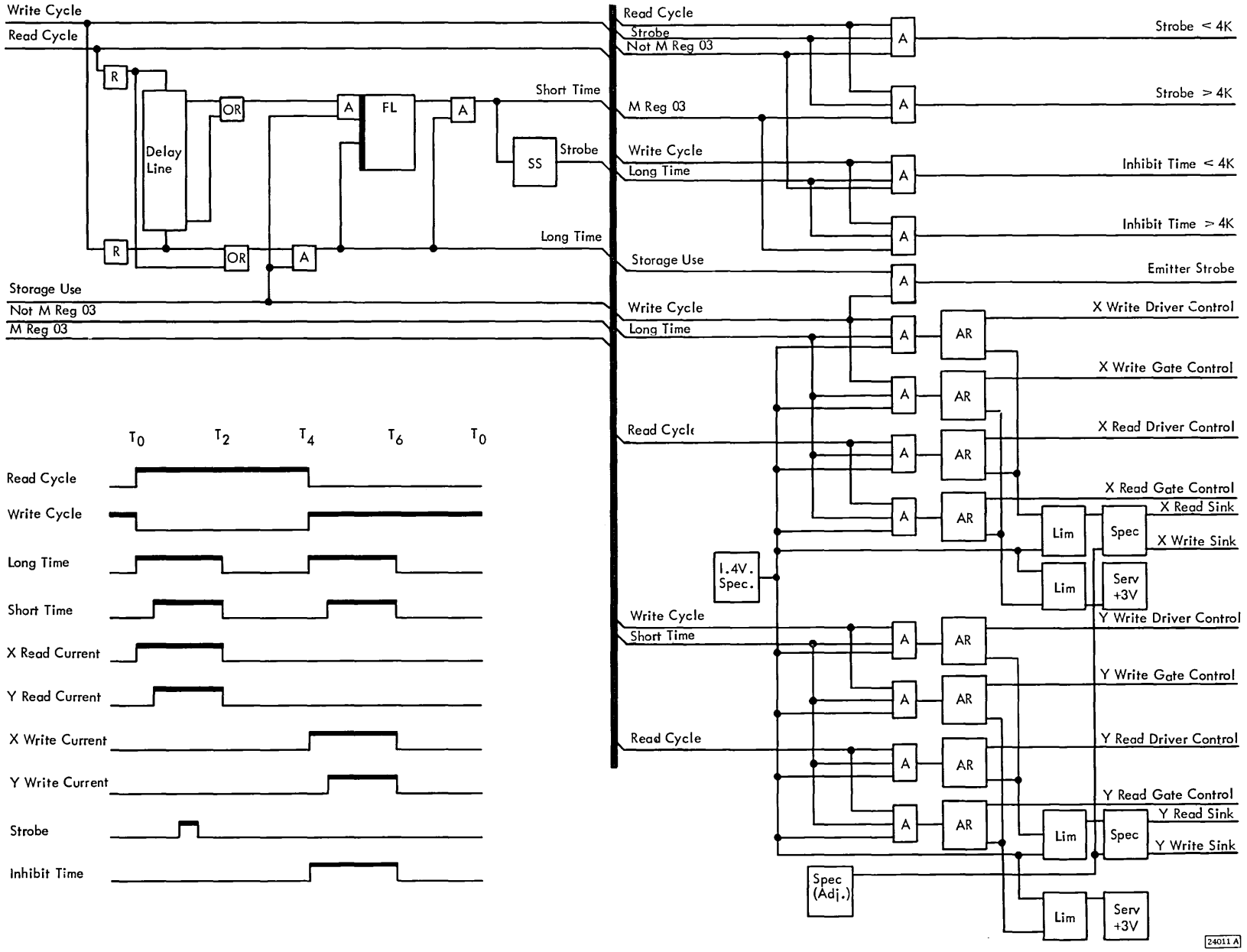


Figure 2-16. Core Storage Clock and Logic Timing

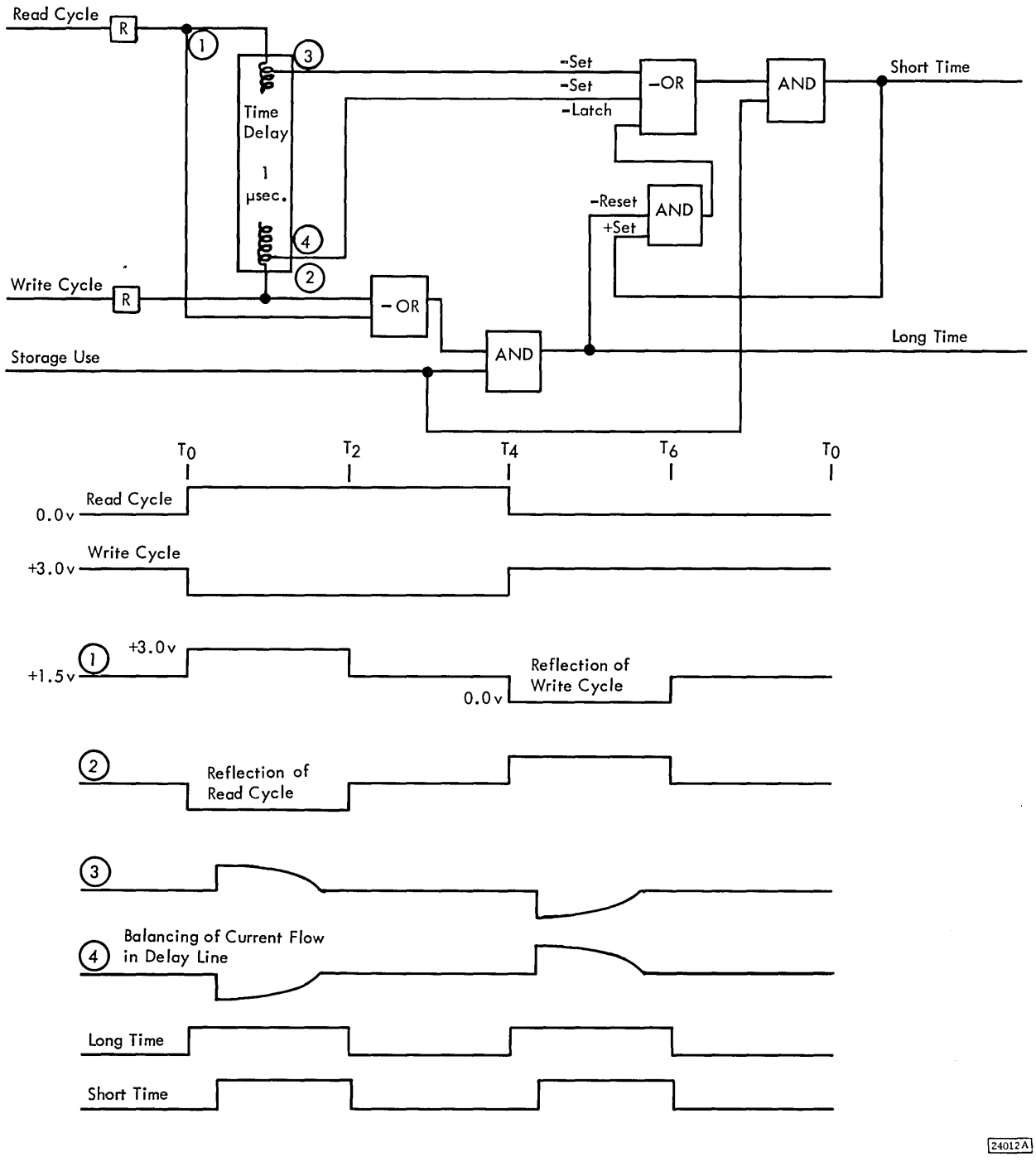


Figure 2-17. Time Delay Clock Circuit

is entered, the read transistor is conditioned and the emitter voltage rises to cut off the center transistor.

2.5 REGISTERS

- The typical flip-flop circuit that comprises the registers is shown in Figure 2-1.
- The output is independent of the triggering action.
- The flip-flop can be set and sampled at the same time.
- The gate must be present 170 nanoseconds before the pulse.
- Reference Maintenance Diagrams: AA101, AA221, Figures 2-18, 2-19.

Registers are used at different cycle times and for different purposes during the various instructions. The times and conditions are explained for each instruction in the Theory of Operation section of this manual.

2.5.1 Storage Buffer Register (B)

- Storage buffer register, 16 positions.
- Data and instruction words read out of core storage are set into the B register.
- Data and instruction words to be written into core storage are contained in the B register.
- Odd parity is generated for each half of the B register during each core storage cycle.
- Inputs to the B register are from:

  Sense amplifiers (core storage)
  I/O input bus
  I counter
  Accumulator

- Outputs from the B register go to:

  Inhibit circuits for core storage
  D register
  I/O out bus
  I counter
  Op register (B_0 - B_4)
  FLAG register (B_5)
  TAG register (B_{6-7})
  Modifier register (B_{8-9})
  Cycle control counter (B_{10-15})

2.5.2 Arithmetic Factor Register (D)

- The arithmetic factor register has 16 positions.
- Stores one factor for arithmetic and logic operations; the other factor is in the accumulator.
- The D register and the accumulator have interconnections that implement arithmetic operations.
- The normal input to the D register is from the B register.
- For operations requiring transfer of data from the D register to the accumulator the control lines; add + sub + EOR gate, and AND gate are brought up.

2.5.3 Accumulator (A)

- Is a 16 position register.
- Stores one factor of an arithmetic operation; the D register contains the other factor.
- Contains the result of any arithmetic operation.
- Can be loaded from storage with a load accumulator instruction; contents can be stored in core storage with a store accumulator instruction.
- Its interconnection with the D register enables the arithmetic and logical operations.
- The contents can be shifted right or left.
- Inputs:
  A0-15
    D register
    I register
    Q register
    U register
  A0
    Q15-shift right, rotate entry
    Shift right sign entry
    Multiply sign entry

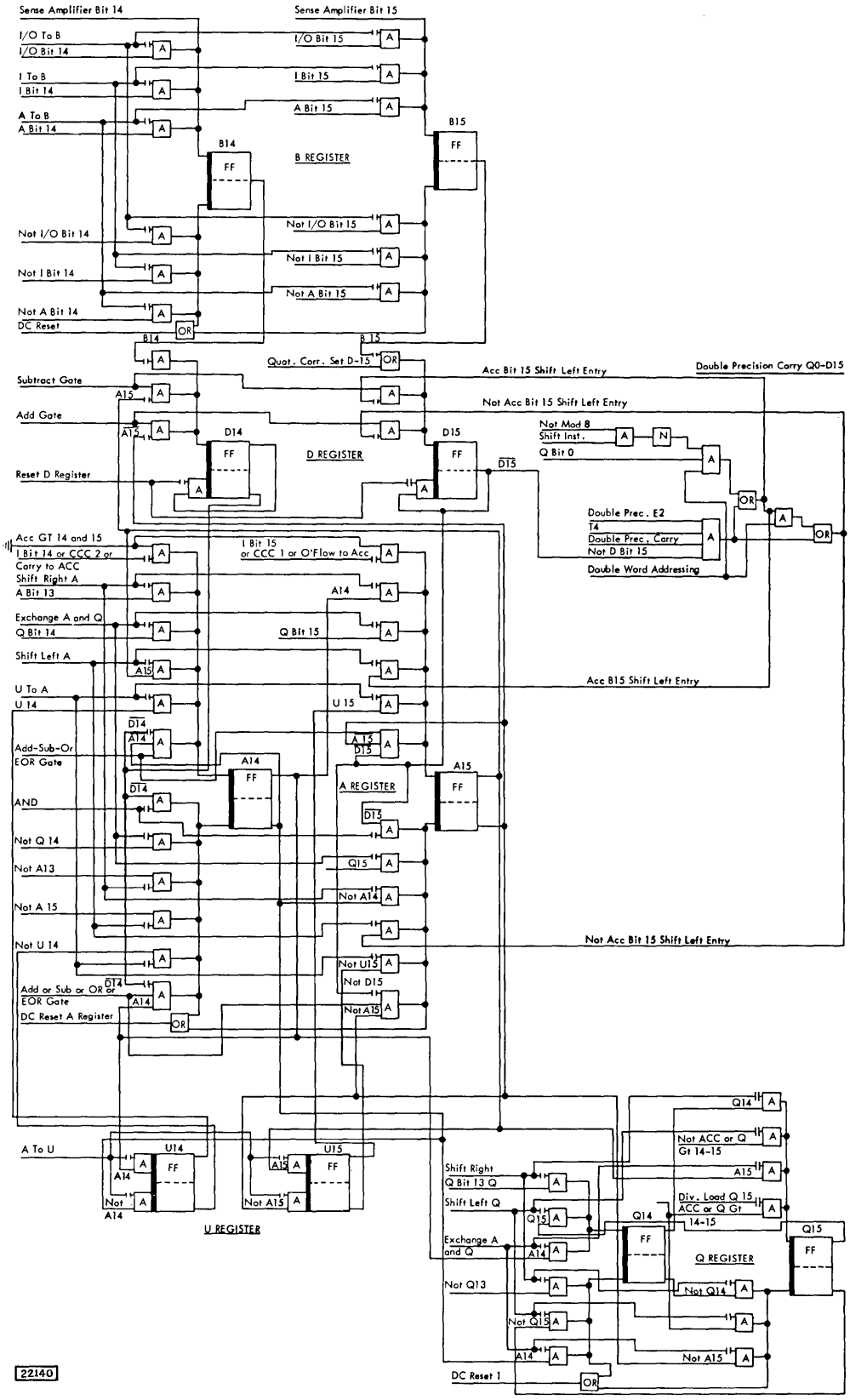


Figure 2-18. Simplified Logic Diagram - Basic Data Flow

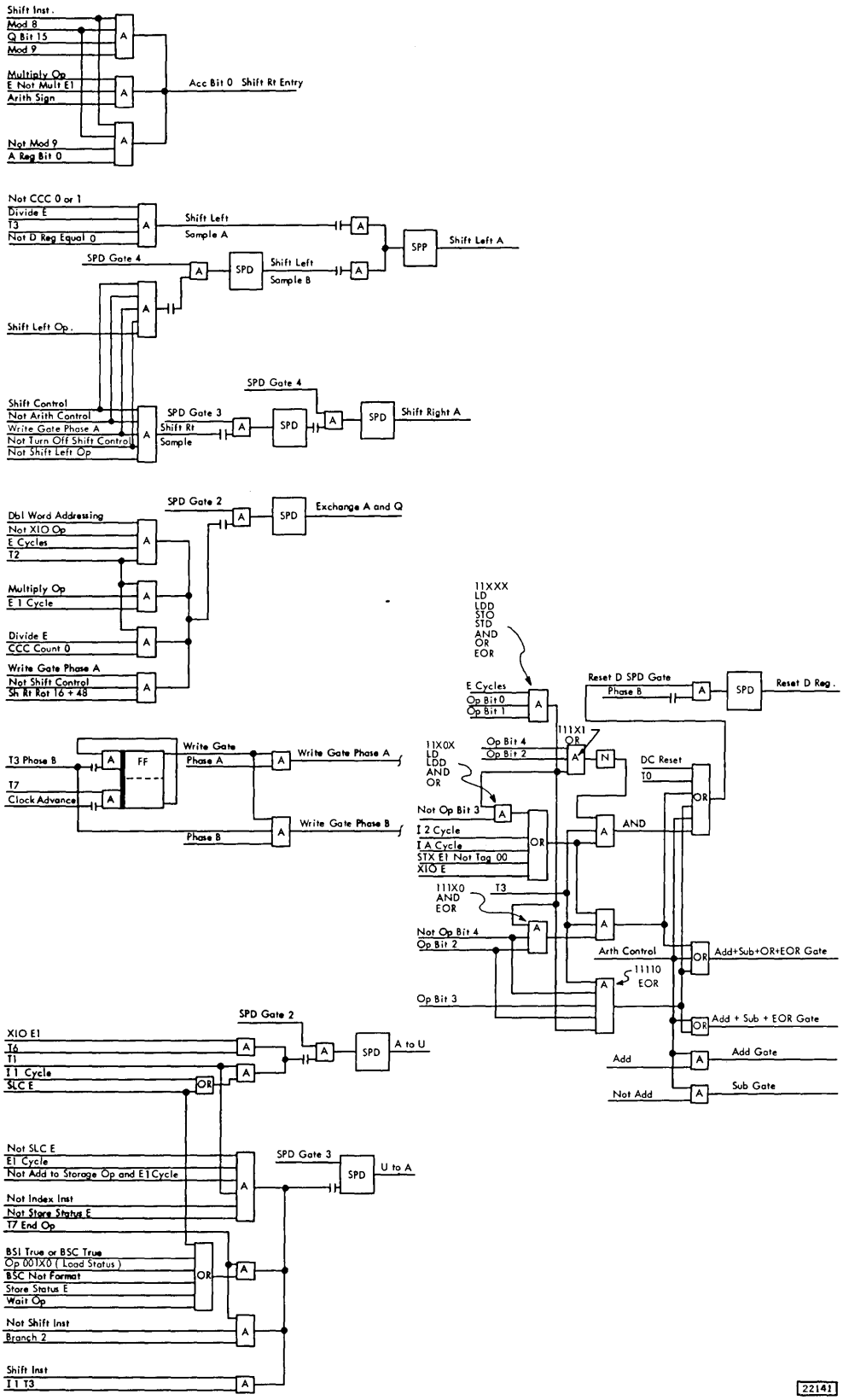


Figure 2-19. Simplified Logic Diagram - Data Flow Controls

A10-15
Cycle control counter
A14
Carry
A15
Overflow
Q0-shift left entry

- Outputs:
A0-15
Q register
U register
B register
M register
A15
Q0-shift right entry

2.5.3.1 Adder

- There is no separate functional unit called the adder.
- The "adder" is composed of the D register, the accumulator, their interconnections, and control.
- D factor is added to the A factor in parallel.
- If D_n is a one, the state of A_n (same bit position as D_n) is changed.
- If A_n changes from a one to a zero, the next higher order of $D(D_{n+1})$ is set to a one. If not, D_{n+1} is set to a zero.
- The add is complete when the D register contains all zeros.
- The D register reset pulse that causes this action occurs at phase A time of clock steps T4 through T7.
- At phase B of the same clock step, the D register is checked for all zeros.
- Figure 2-20 is an example of an add operation.
- T7 is extended until the add is completed, if the operation requires more than four add cycles (T4-T7).

	0 0 1 0 1 1 0 1	D
Start	0 0 0 1 0 1 1 1	A
Carries	0 0 0 0 1 0 1 0	D'
Partial Sum	0 0 1 1 1 0 1 0	A'
	0 0 0 1 0 1 0 0	D"
	0 0 1 1 0 0 0 0	A"
	0 0 1 0 0 0 0 0	D^{III}
	0 0 1 0 0 1 0 0	A^{III}
	0 1 0 0 0 0 0 0	D^{IV}
	0 0 0 0 0 1 0 0	A^{IV}
	0 0 0 0 0 0 0 0	D^{V}
Result	0 1 0 0 0 1 0 0	A^{V}

Sum Rule: Each 1 in D changes value of corresponding bit in A.

Carry Rule: A change from 1 to 0 in A sets next higher-order position in D to 1.

24031A

Figure 2-20. Example of an Add Operation

- The circuit connections for the add operation (Figure 2-21) are:

A_n	turn on AND	Gate	- A_n off
		Pulse	- D_n going off
	turn off AND	Gate	- A_n on
		Pulse	- D_n going off
D_n	turn on AND	Gate	- Add gate
		Pulse	- A_{n-1} going off
	turn off	Pulse	- Reset D register

- Negative factors and sums are expressed in two's complement form.
- Subtraction is accomplished by additional inputs to the D_n position turn on and turn off as follows (Figure 2-22):

D_n	turn on AND	Gate	- Subtract gate
		Pulse	- A_{n-1} turning on.
D_n	turn off AND	Pulse	- Reset D register

- If D_n position is a one, the state of A_n is changed.

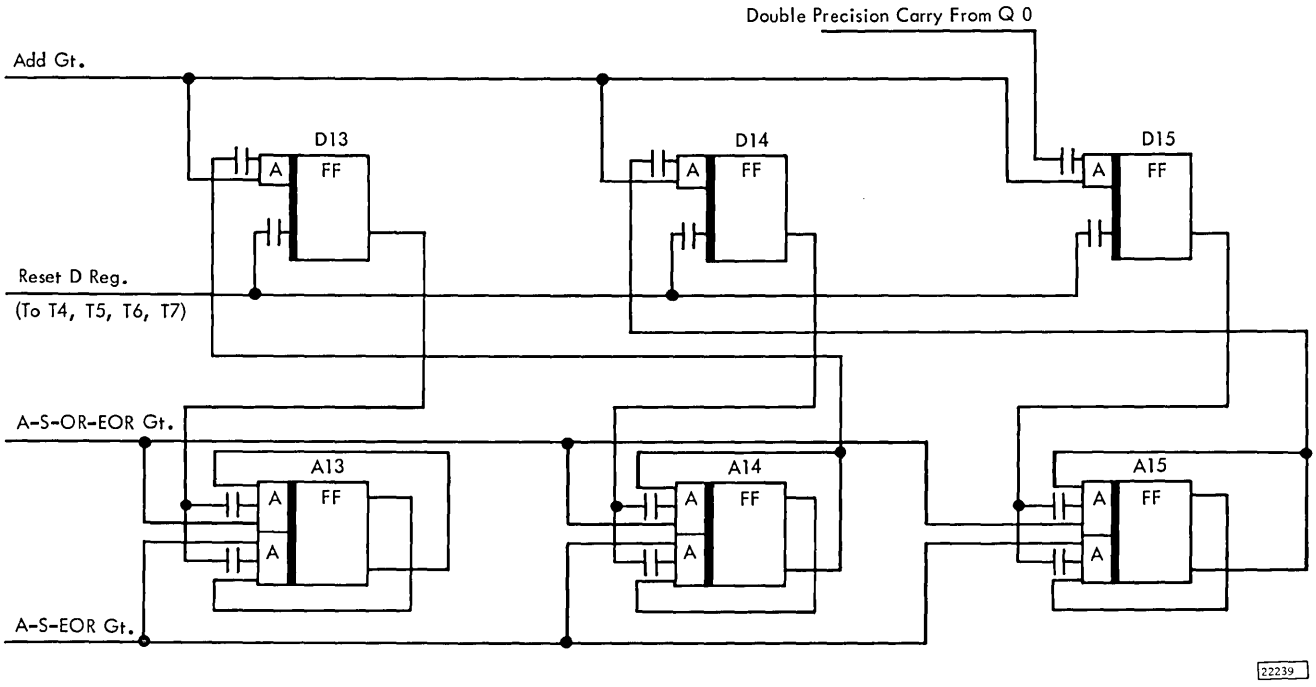


Figure 2-21. A and D Register - Add Operation

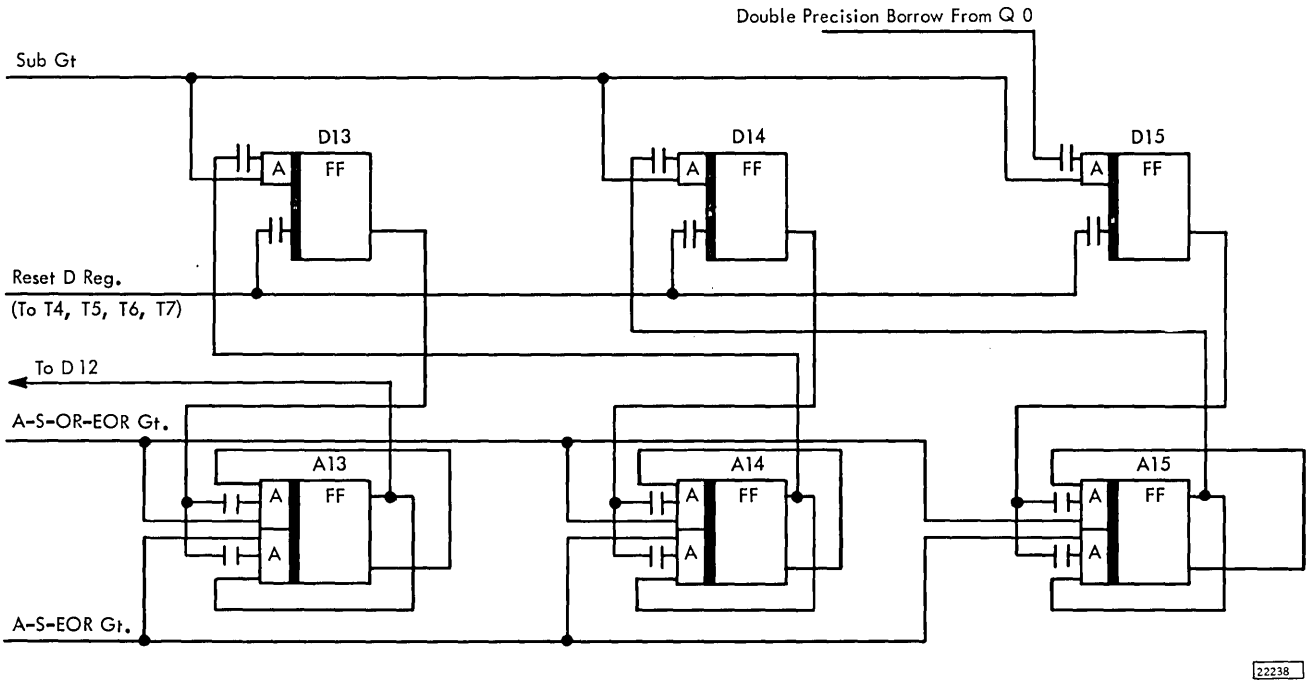


Figure 2-22. A and D Register - Subtract Operation

- The next higher order of D (D_{n+1}) is set to a one, if A_n changes from a zero to a one. If not, the next higher order (D_{n+1}) is set to a zero.

2.5.4 Temporary Accumulator Register (U)

- Temporary storage for the contents of the accumulator; 16 positions.
- Input and output is to/from the corresponding positions of the accumulator.
- Output goes to I/O attachments for function decodes.

2.5.5 Accumulator Extension Register (Q)

- An extension of the low order end of the accumulator; 16 positions.
- Inputs are from the corresponding positions of A; the Q0 position has shift right input from A15. Q15 has an input from divide quotient load.
- Outputs are to the corresponding positions of A. Q0 has a shift left output to A15.
- Contents can be shifted right or left.

2.5.6 Instruction Address Register (I)

- Sixteen position register connected as a counter to maintain the address of the next instruction.
- Contents are transferred to the M register at T0 time of I-1 and I-2.
- Contents are then increased by one for the next sequential address.
- Contents are transferred to the A register for effective address generation during each one-word instruction cycle.

2.5.6.1 I Register Incrementing

The 16 flip-flops of the I register are connected as a binary counter (Figure 2-23). All the flip-flops are gated by increment gate except during the operations in which incrementing is not needed. The pulse that turns on or off I 15 is active every end op T0 time (I to M pulse). I 15 going off turns on or off I 14: I 14 going off turns on or off I 13. This binary-counter method is used for all the flip-flops except positions I 11, I 7, and I 3. These FF's are turned on or off by the lowest order position of the group (e.g., I-4 to I-7) going off when all the positions of the group are on. This speeds up the rippling of a full section.

2.5.7 Storage Address Register (M)

- Storage address register consists of 16 positions.
- Addresses core storage with positions 2 through 15.
- Loaded from the I counter or the accumulator.
- Output is inhibited for index and cycle steal operations.
- During disk storage operations, output is replaced by the cycle steal address register.
- Position 15 output can be forced during double precision instructions; 14 and 15 can be replaced during index instructions.
- During printer scans the output is replaced by the printer address gate controls.

2.5.8 Cycle Control Counter (CCC)

- Is a six position binary counter.
- Is loaded at the beginning (I-1, T5) of every operation to control the number of cycles to be taken.
- Is loaded at the beginning of a shift instruction to control the number of positions to be shifted.
- Is decremented at T1 of each execute cycle (Figure 2-24).
- A count of 0 causes the operation to end.

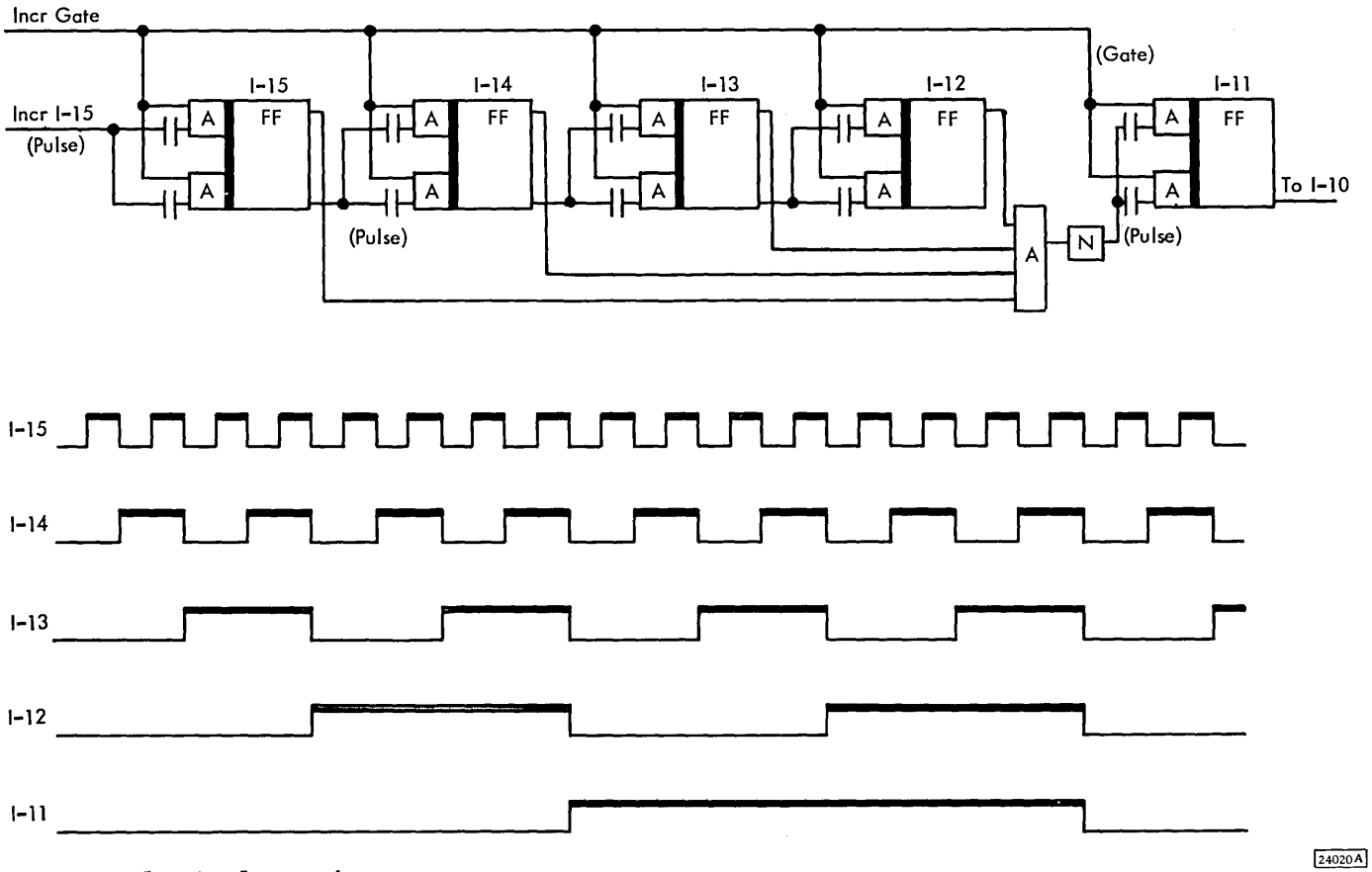


Figure 2-23. I Register Incrementing

The shift counter consists of six flip-flops; CCC-1, CCC-2, CCC-4, CCC-8, CCC-16, and CCC-32. It can be set to a count of 1, 16, or 18 or can be loaded from the B register. The output is tested for count-equal-zero condition and is gated to the A register for the shift left and count instruction. Figure 2-24 shows the shift counter set to a count of 18 and decremented to zero.

2.6 INPUT/OUTPUT CONTROLS

- Control transferring of data to or from the CPU.
- Direct program control - each data word is transferred by an individual execute I/O (XIO) read or write instruction.
- Disk storage - an XIO instruction initiates the operation, the I/O device proceeds and transfers data to or from core storage by cycle stealing.
- Provides for interrupt circuits to signal the CPU that a device has information ready for the program.
- Provides for cycle steal circuits to allow the disk storage and the printer to insert cycles, for data transfer, into other CPU operations.
- Reference to Maintenance Diagrams: AA101, AA211, AA231, AA621, AA611.
- Descriptions of the console printer, console keyboard, and disk storage I/O operations are in Chapter 3 Theory of Operations.

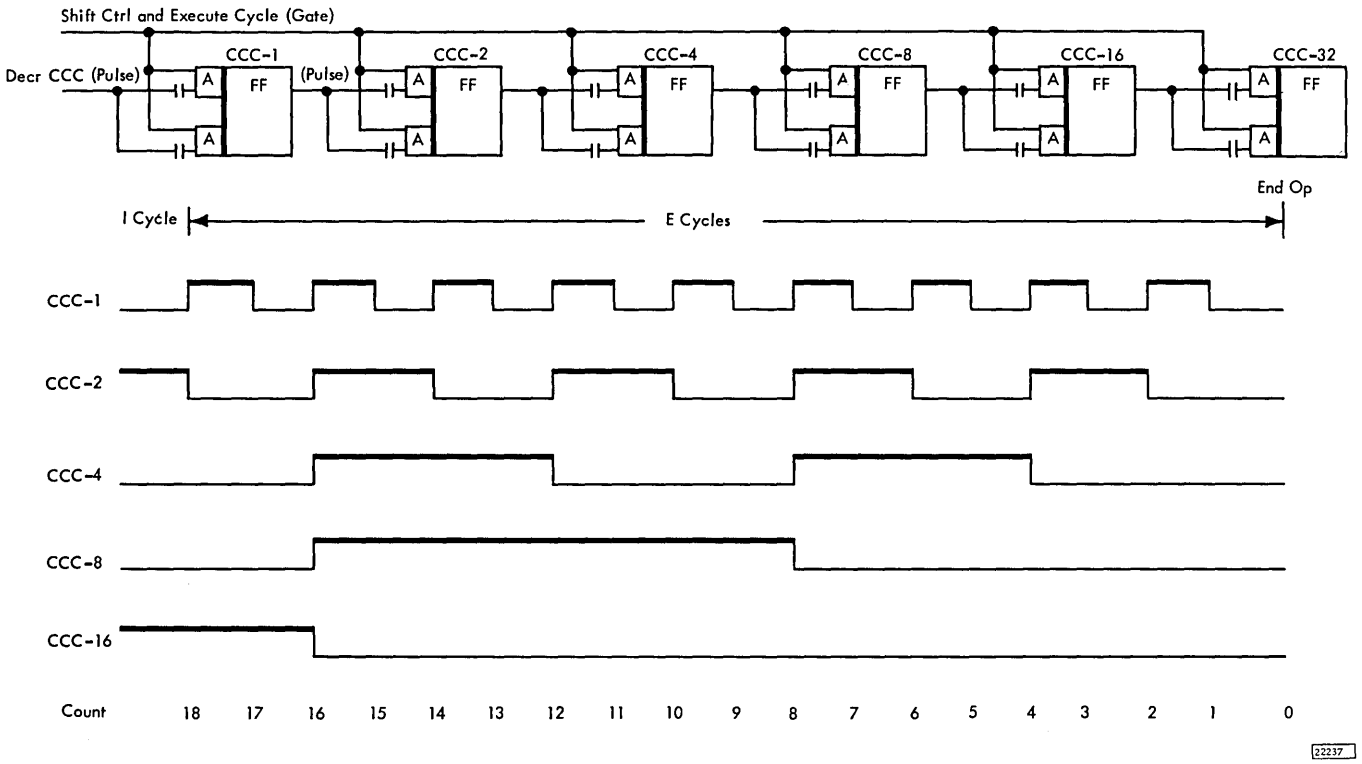


Figure 2-24. Cycle Control Counter Decrementing from 18

2.6.1 Cycle Steal Circuits

2.6.1.1 X Clock

- Consists of 8 F-F's (X0 through X7).
- Advance X0 through X7, advances come at phase A.
- Starts with the second phase A of T7; Add, T7 extended cycles, may continue after the X clock has started.
- Runs under control of cycle steal levels 0 and 1.
- Reference to Maintenance Diagram AA610.

2.6.1.2 Cycle Steal Controls

- Consists of 2 F-F's.
- Cycle steal trigger 0 set with request from the Disk Storage attachment.
- Reset at end of cycle steal cycle.
- Cycle steal trigger 1 set with request from the 1132 Printer attachment.
- Either CS0 or CS1 will block T clock advance from T7 to T0.
- CS0 overrides CS1 so that disk storage operations take precedence over printer operations.
- Either CS level blocks addressing of core storage by the M register.
- CS0 causes core storage to be addressed by the file core address register.
- CS1 causes core storage to be addressed from the printer address gates.
- Reference to Maintenance Diagram: AA231, AA611, AA621.

Cycle stealing is the ability of I/O devices to intervene in the processing of a CPU operation and insert one, or more, cycles to transfer data between core storage and an I/O device. At the end of the cycle steal operation, the CPU operation is resumed at the point where the cycle steal cycle(s) occurred.

2.6.2 Interrupt Controls

- Provides for simultaneous operation of I/O devices with CPU operations.
- The interrupt facility provides an automatic branch from the normal program sequence based upon an external condition.
- A maximum of six interrupt levels are available with the 1130 Computing System.
- Interrupt level assignments are:

Device	Level
1442 Card Read-Punch Response	0
1132 Printer	1
Disk Storage	2
1627 Plotter	3
Console Printer	4
Console Keyboard	
1134 Paper Tape Reader	
1055 Paper Tape Punch	
1442 Card Read-Punch Operation Complete	
Console Stop switch or Interrupt Run Mode switch	5

- Level 4 interrupts can be decoded by reading the interrupt level status word (ILSW) into the ACC with a sense interrupt control command (011).
- Reference Maintenance Diagrams: AA211, AA231, AA611.

2.6.2.1 Interrupt Philosophy

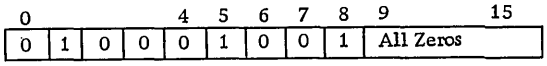
Because of the number and types of interrupt requests, it is not always possible to cause a branch to a unique address for each interrupt condition. For the same reason, it is frequently not desirable to cause one branch for all interrupt requests and require the program to determine the individual request(s) requiring service. Therefore, it is expedient to group the many individual request lines into a lesser number of priority levels. This 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.

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

2.6.2.2 Program Operation

An interrupt may be recognized by the CPU at the completion of any program instruction (Figure 2-25). It is initiated by the basic interrupt control, which forces execution of a CPU-generated branch and store IAR (BSI) instruction. This is accomplished by inhibiting the normal access to storage by the stored program and generating into the SBR (storage buffer register) a BSI (indirect) instruction from the interrupt control.



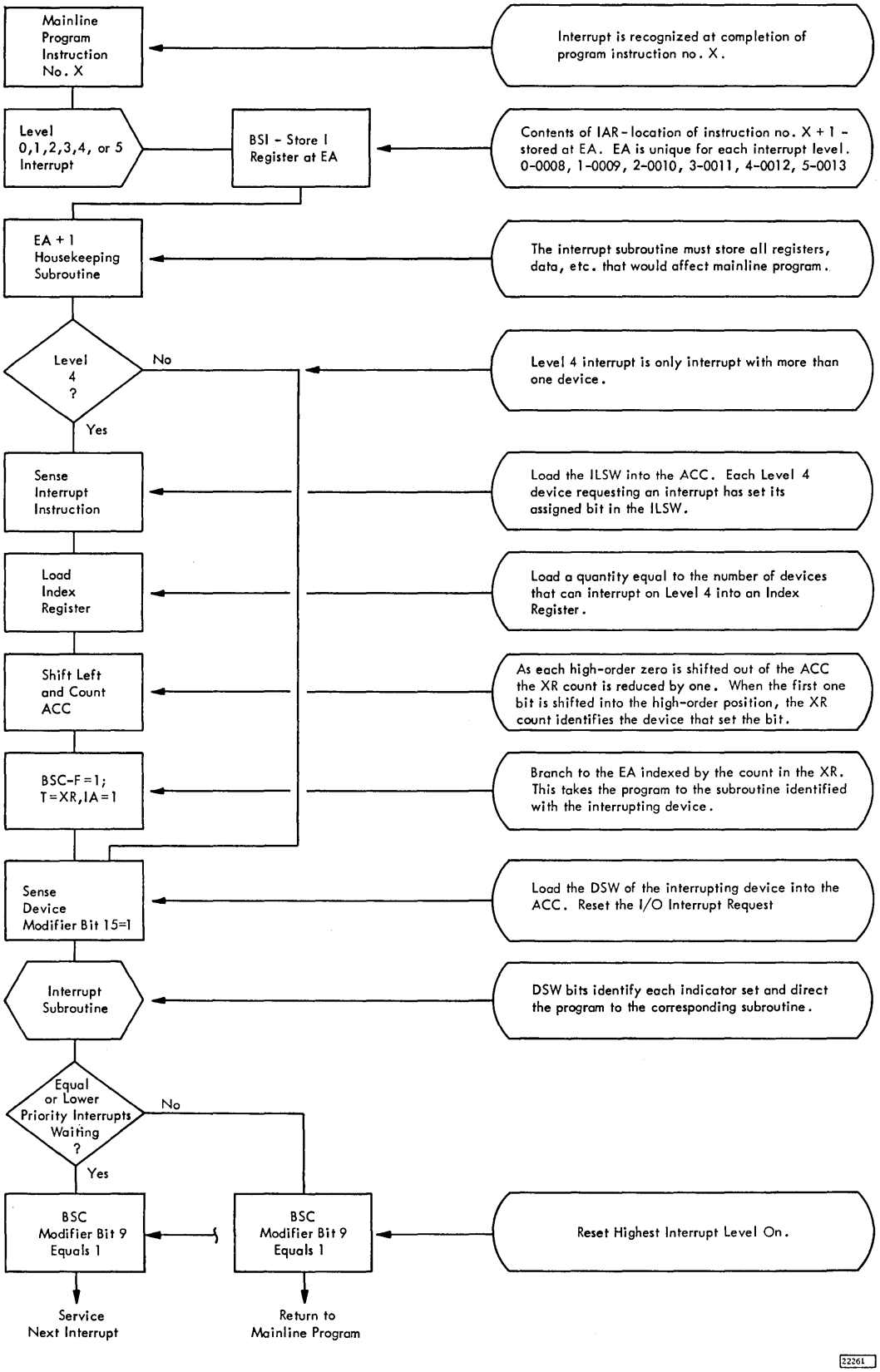
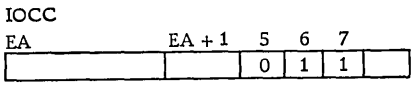


Figure 2-25. Interrupt Procedure Block Diagram

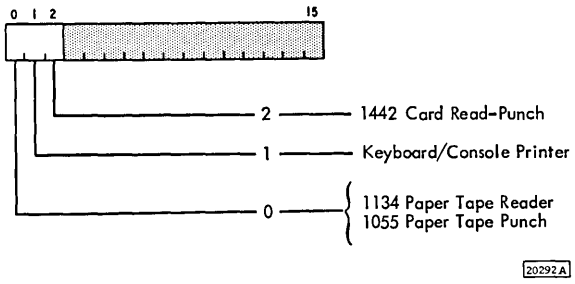
The BSI instruction EA is developed by emitting an address depending on the level of the interrupt as shown in Figure 1-8. The constant stored there is used as the EA and the I register is stored at the EA (effective address) and a branch to EA + 1 occurs. It is the responsibility of the interrupt subroutine 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.

Several devices can request an interrupt on level 4. 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 (sense ILSW).



The sense interrupt function is decoded and sent to all I/O devices, along with the current interrupt level being serviced. The sense interrupt command will therefore produce meaningful results only if executed in a program sequence that is a result of interrupt level 4, and before a sense DSW or branch out command is executed in this routine.

Interrupt Level Status Word



Although a 16-bit ILSW could exist for each priority level, only level 4 uses the ILSW in the 1130 System. Each device with an interrupt request signal assigned to priority level 4 is given a particular bit position in its ILSW to indicate its interrupt request status; a 1-bit if on, and, a 0 bit if off. The status indicator(s) is 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).

2.6.2.3 Interrupt Identification

Following loading of the ILSW word 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 interrupt priority level 4, followed by the shift left and count instruction (SLC). The resulting count in the index register is unique and corresponds to the first non-zero bit of the ILSW in the accumulator. (It is also possible to execute a shift left and count of both the ACC and EXT. The SLC 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 SLC. This provides, in conjunction with a branch table, a unique branch for each non-zero bit of the ILSW.

Programming Note: When an interrupt request has been detected by a priority level, the program is directed to service the request by interrupting. During the servicing, all interrupt requests of equal or lower status are effectively constrained from interrupting while the servicing of the higher priority is in progress. 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 of the highest level that is on. This reset permits lower priority requests (including those that may have been temporarily constrained, but recorded) to be accepted once again by the CPU. Therefore a branch out instruction is given. This is effected by making bit 9 = 1 in this instruction. This programmed recognition of waiting interrupts should not be confused with a normal subroutine linkage back to a mainline program in which bit 9 should be set to zero.

The BSC is a conditional instruction and when bit 9 = 1, the interrupt level is reset only when the branch or skip occurs.

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 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).

2.6.2.4 Device Status Word (DSW)

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.

2.6.2.5 Interrupt Circuit Description

- Consists of six level FF's (0-5) and six request FF's (0-5).
- The six request FF's determine if the CPU has started servicing the interrupt. If On, the interrupt has not been serviced.
- The six level FF's determine if the CPU has finished servicing the interrupt. If On, the level has not finished its routine.
- Each higher priority level degates all lower levels and causes a new interrupt.
- At the end of an instruction, after the level and request FF's come on, the interrupt gates set up a BSI instruction with an address determined by the highest priority level.
- The address of the next instruction is stored so that the routine left may be returned to at the right place.
- If one level is being serviced and an interrupt of a higher level is set, the first interrupt routine is stopped and the new one started.
- As each interrupt routine is finished, the CPU is returned to the next lower order interrupt level unserviced or unfinished.
- Reference to Maintenance Diagrams: AA231, AA611.

2.6.2.6 Interrupt Logic Flow

- Enter AA611 sheet 1.
- Wait for end of instruction.
- Don't branch if CE switch-interrupt delay is on.
- Wait for end of higher or same level interrupt.
- Set level and request FF for this level of interrupt.
- Set interrupt request FF.
- If in wait operation, set run FF and branch.
- Cycle steal overrides interrupts.
- Block addressing of core storage-drop storage use.

I-1:

- Force BSI operation code and modifier bits on I/O input bus.
- Set B register.
- Enter AA601 sheet 2 at T1 time.
- Transfer ACC to U register.
- Note: Storage to B register transfer is blocked.
- Transfer B register to operation, tag, format and modifier registers.

- Transfer B to D register (displacement).
- Set CCC to 1.

I-2:

- Enter AA611.
- Gate interrupt address on to I/O input bus.
- Set B register.
- Enter AA601 Sheet 5 - Time 2.
- Transfer B to D register.
- Transfer D to A register (address at interrupt level).
- Enter AA611 at Time 7.
- Reset request FF for level being serviced.
- Unblock storage use.
- Enter AA601 - Sheet 7 (IA Cycle).

IA:

- Set IA FF.
- Transfer A to M register - address interrupt word.
- Read core storage to B register.
- Transfer B to D register.
- Transfer D to ACC register.

E-1:

- Transfer A to M register (EA).
- Transfer U to A register.
- Decrement CCC.
- Transfer I to B register.
- Write B register into core storage (address of next instruction in program).
- Transfer M to I register.
- Increment I register (address of first instruction in interrupt sub-routine).
- End operation, enter I-1 cycle.

2.6.3 Disk Storage Attachment

2.6.3.1 Word Counter

- Is a 9 bit binary counter.
- Is set, in complement form, from the I/O out bus on the first cycle steal cycle of a read or write operation.
- Is set, in complement form, from the I/O bus on the E-2 cycle of a seek command.
- On read or write operations, it keeps track of the number of words to be read.
- On seek operations, it keeps track of the number of cylinders to be moved.
- On read or write operations it is incremented, by a bit counter position, for each word.
- On seek operations, it is incremented with each access drive pulse.
- When all counter positions are on, a full word count decode ends the operation.
- Reference Maintenance Diagrams: XF501, XF511, XF521.

2.6.3.2 File Data Register

- Sixteen bit binary register.
- On write operations it is set from the I/O output bus on cycle steal cycles after the first cycle steal cycle.
- The first word of a data field contains the word count of the field and is not set in the file data register.
- On write operations bit 15 is the input to the write circuit.

- On write operations the register is shifted right, to move each bit into bit position 15, at each clock time.
- On read operations the register bit 0 is set by each read data bit and reset by each not read data bit.
- On read operations the register is shifted right, to move bits read into bit 0 to the right, at each clock time.
- On read operations the register is gated to the I/O input bus on cycle steal cycles after the first cycle steal cycle.
- Reference Maintenance Diagrams: XF501, XF511.

2.6.3.3 File Address Register

- Fourteen bit binary register.
- Set from the I/O output bus, during the E-2 cycle, on read and write operations.
- Is used to address core storage when stealing cycles during disk storage operations.
- It incremented +1 on each cycle steal cycle.
- Reference Maintenance Diagrams: XF501, XF511.

2.6.3.4 Modulo 4 Counter

- Is used to check that bits are not picked up or dropped.
- Two position binary counter.
- Incremented with each data bit written or read.
- On read operations, the counter must be zero at the end of each word.
- On write operations, the counter is used to write bits 17-20 and the counter is reduced to zero.
- Reference Maintenance Diagrams: XF501, XF511.

2.6.3.5 Bit Counter

- Five position binary counter labeled A through E.
- Positions A through D provide timing for the sixteen data bit positions of each word read or written.
- Position E controls the writing or reading of the four modulo 4 bits.
- Bit position A is set or reset with each read or write clock pulse.
- Reference Maintenance Diagrams: XF501, XF511, XF701, XF711.

2.6.3.6 Sector Register

- Three position register.
- Set from the U register bits 13, 14, 15 on the E-2 cycle.
- The high order position is used to select the head, off = upper head, on = lower head.
- The two low order positions are used to decode the sector to be read or written.
- Reference Maintenance Diagrams: XF502, XF511.

2.6.3.7 Sector Counter

- Is used to keep track of the sector available to the heads.
- Two position binary counter.
- Reset off by the index pulse each revolution of the disk.
- Advanced by the sector pulse from the disk.
- Is compared with the sector register. When equal, start the read or write operation.
- Reference Maintenance Diagrams: XF502, XF511.

2.7 CONSOLE KEYBOARD

2.7.1 Keyboard Mechanics

- The keyboard mechanism converts a manual key depression into an electrical impulse.
- The keyboard interlocks so that an operator can press only one key at a time.
- The two basic units in the keyboard are:
  The permutation unit and
  The key components unit.
- Some keys perform an electrical and mechanical function, and some keys perform only an electrical function.

The keyboard incorporates a square SELECTRIC®-type keybutton, and the keyboard layout corresponds to the layout used in the latest machines having operator keyboards.

Figure 2-26 shows the combination keyboard with its cover removed and the keyboard disassembled into its two major units (key components unit and permutation unit). The key unit contains all the keys. The alphabetic key, numeric key, and reset keyboard key perform only electrical functions. These keys operate keystem contacts (Figure 2-26).

Figure 2-27 shows one latch pull bar, one bellcrank, one latch, and one permutation bar. All the keys, except those operating keystem contacts, operate a latch pull bar through a keystem bellcrank.

For details of keyboard electrical functions, refer to 2.7.5 Keyboard Electrical Functions.

The key button rubber bumper, a rubber washer under the key button, is associated with the keystem. The amount of downward travel on the key when it is pressed is partly determined by the thickness of the rubber bumper. As a key is pressed, the bellcrank pivots and moves the latch pull bar toward the key stem. The hooked portion of the latch pull bar rests in a notch in the top of the latch (Figure 2-27). Each latch pull bar operates a latch. When a key stem is pressed and the latch pull bar pulls the latch off the latch bar, near the end of the operation the keystem is restored by its spring.

The mechanical operation upon pressing a key follows (Figure 2-27, 2-28, and 2-29):

1. The keystem bellcrank moves its latch pull bar forward.
2. The latch assembly drops off the latch bar.
3. Individual keystem springs restore the keys and the pull bars to normal.
4. A separate flat spring holds each pull bar against its latch assembly and makes sure it relatches in the notch in the latch.

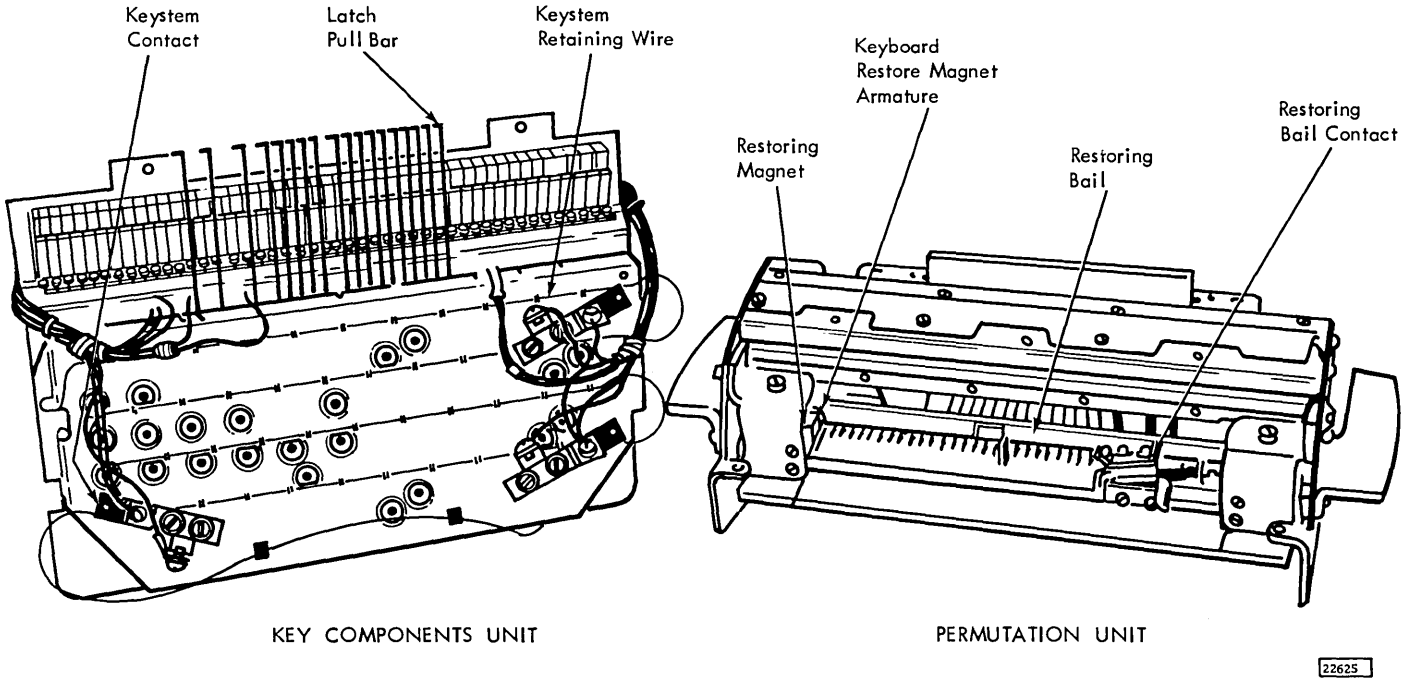


Figure 2-26. Keyboard Units

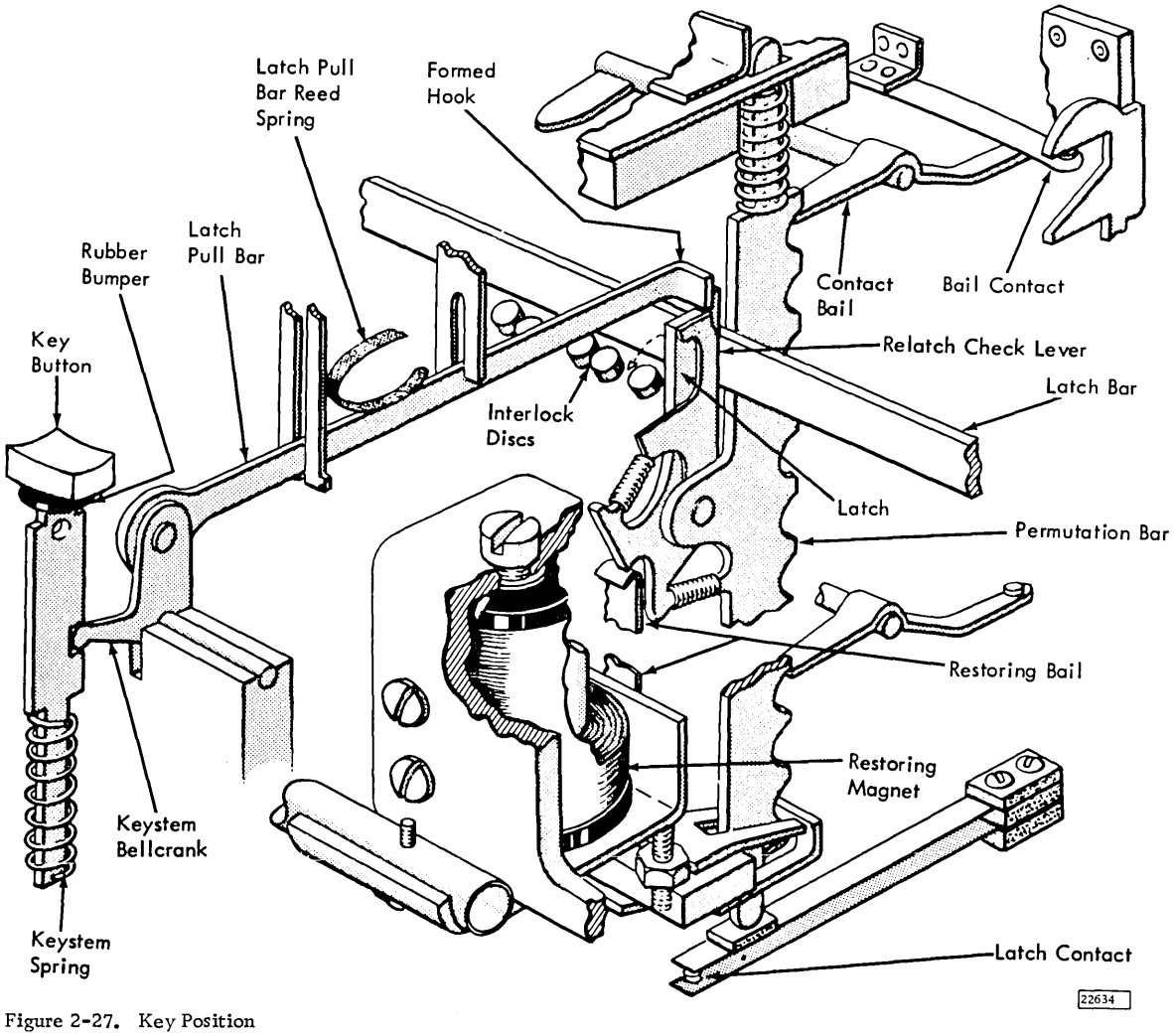


Figure 2-27. Key Position

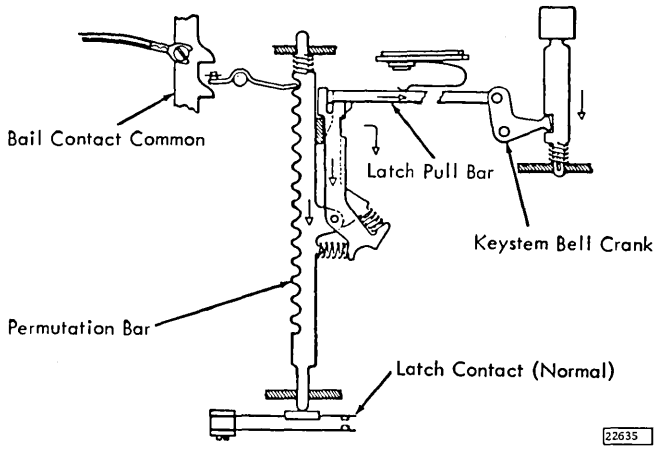


Figure 2-28. Permutation Bar and Keyboard Latch (Normal)

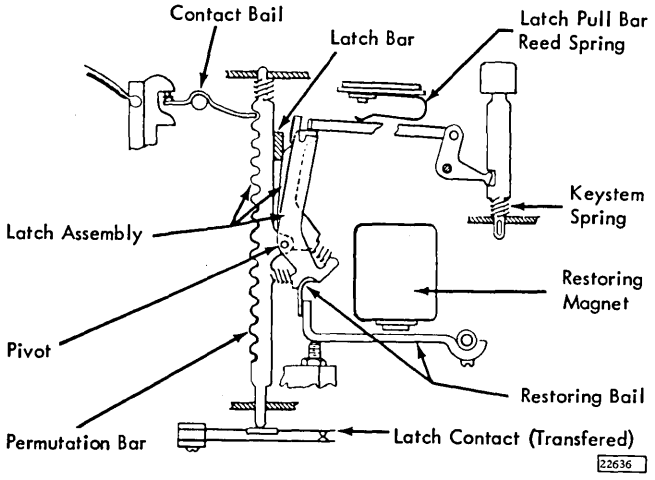


Figure 2-29. Permutation Bar and Keyboard Latch (Tripped)

Note in Figure 2-29 that the latch assembly has three parts that, although attached by a rivet, are free to pivot on the rivet. Each part has its own function.

The permutation bar is contained in the permutation unit of the keyboard. The permutation bar supports the operating spring and pivots the contact bails to close the bail contacts. Some permutation bars operate latch contacts. The latch pivots on the permutation bar. The latch pull bar reed spring can be seen in Figure 2-29. The latch pull bar has a 90 degree hook formed at its end. The hooked end of the pull bar rests in a notch that is cut into the top of the latch, and it is held there by the U-shaped reed spring.

Permutation means tranformation or change in grouping. In the keyboard, the operation of the latch and bail contacts is controlled by the permutation bars. The latch hooks over the latch bar, and holds the latch assembly and its permutation bar inoperative until a key is pressed. The purpose of the key unit is to facilitate selection of individual characters and, after selection, to operate a permutation bar. The permutation bar transfers the bail contact and a latch contact (Figure 2-29) necessary to form the correct character. The permutation bar is spring-operated and held restored by the latch. The latch bar is not a moving part. When the latch is hooked onto the latch bar, the permutation bar is held restored. When the latch is pulled off the latch bar, the permutation bar is then operated by the force of its spring.

The function of the permutation bar requires it to move. This movement is guided by the upper front guide rail and the latch stop plate at the bottom. The lower end of a permutation bar, when operated, transfers a latch contact.

2.7.2 Bail and Latch Contacts

- These contacts are operated by permutation bars to energize the correct circuit for each character.
- Fifteen bails span the width of the keyboard.
- Bails, when pivoted, transfer a bail contact.
- A latch contact is transferred by its permutation bar.

The contact bail is pivoted by a tab attached to the bail and resting in a notch that is cut in the front (as viewed in Figure 2-30) edge of a permutation bar. A permutation bar has fifteen notches cut in its front edge for operating any of the possible fifteen contact bails.

A keyboard reference chart, located on ALD ZX121, shows the combination of latch contacts and bail contacts needed to punch a desired character. Odd-numbered bail contacts are on the right side of the keyboard, and even-numbered bail contacts are on the left side of the keyboard (as viewed from front of keyboard).

The chart in Figure 2-30 identifies the key-stem numbering of the combination keyboards. Pressing any one key can never transfer more than one latch contact, but more than one bail contact may be transferred. Latch contacts are mounted on the keyboard lower frame directly under their corresponding permutation bar.

2.7.3 Keyboard Restoring Components

- The keyboard is restored by two restoring magnets operating a restoring bail.
- Pressing a key pulls a latch off the latch bar.
- The latch is restored to the latch bar by the restoring bail.

The restoring bail operates a normally closed bail contact. When the keyboard restoring magnets are energized, the restoring bail restores the bail to the latch bar, and the restoring bail contact is opened.

Figure 2-31 shows the latch and relatch check levers in two different positions. The large over-all drawing shows the latch on the latch bar (restored), and the inset view shows the latch off the latch bar (tripped). The latch and relatch check levers pivot on a stud (pivot) on the permutation bar. The latch is held on the latch bar by the force of its spring. The relatch check lever is operated by its spring.

The machine operation is faster than the operator's finger movement on the keys.

When the latch is pulled off the latch bar (Figure 2-31 inset), the relatch check lever pivots toward the latch bar. The pull bar disengages from the latch because it cannot follow the latch downward. The pull bar cannot engage in the slot in the latch again until the latch and pull bar have both been restored. A key can be pressed and held pressed while the latch is pulled off the latch bar and restored to the latch bar. The pull bar re-engages in the latch only if the key is re-

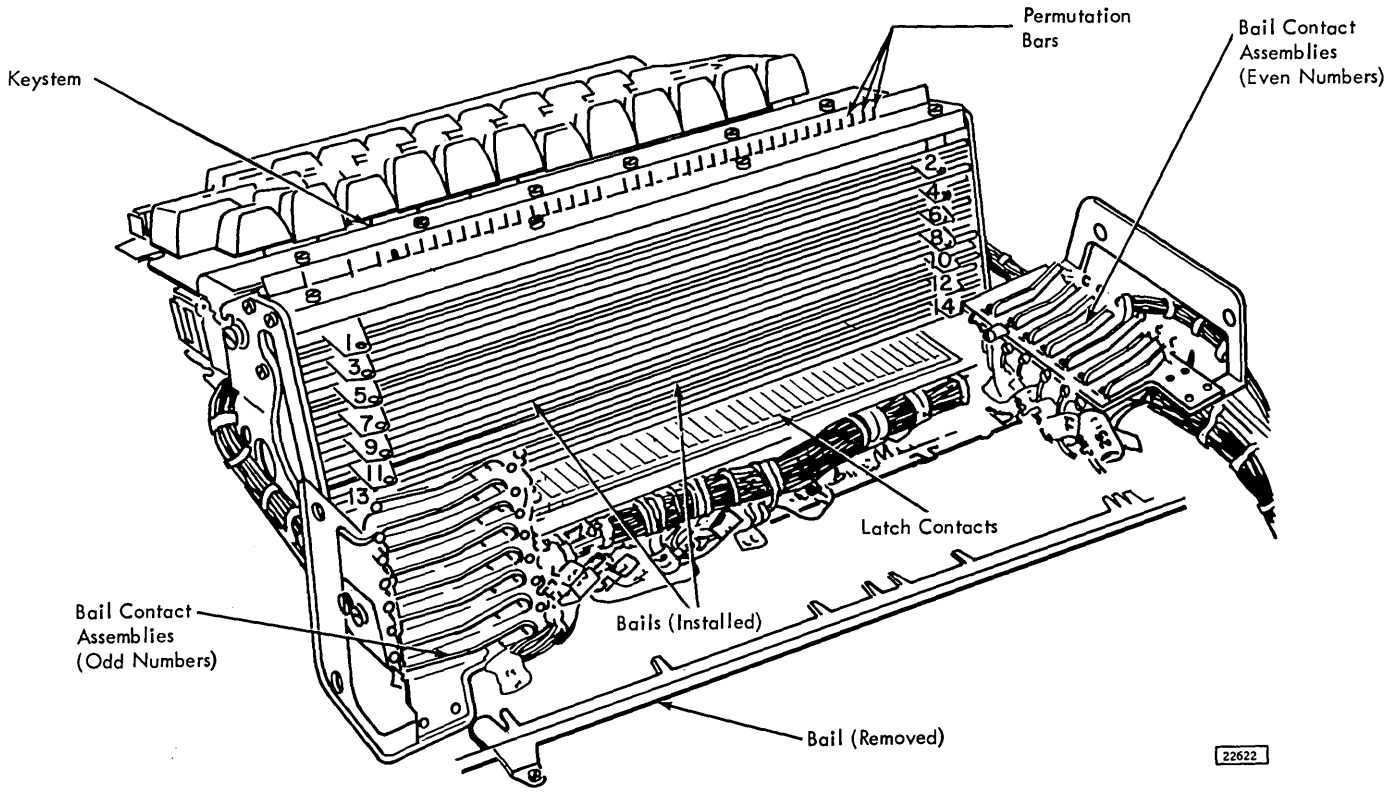


Figure 2-30a. Permutation Unit and Keystem Numbering Chart

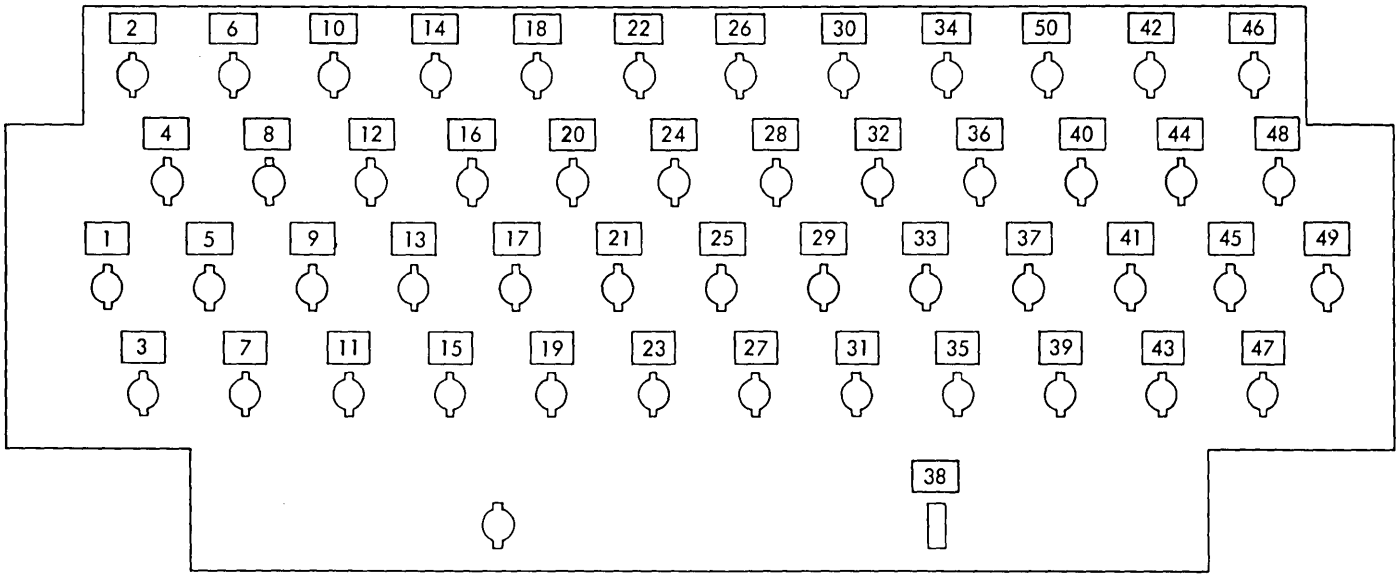


Figure 2-30b. Permutation Unit and Keystem Numbering Chart

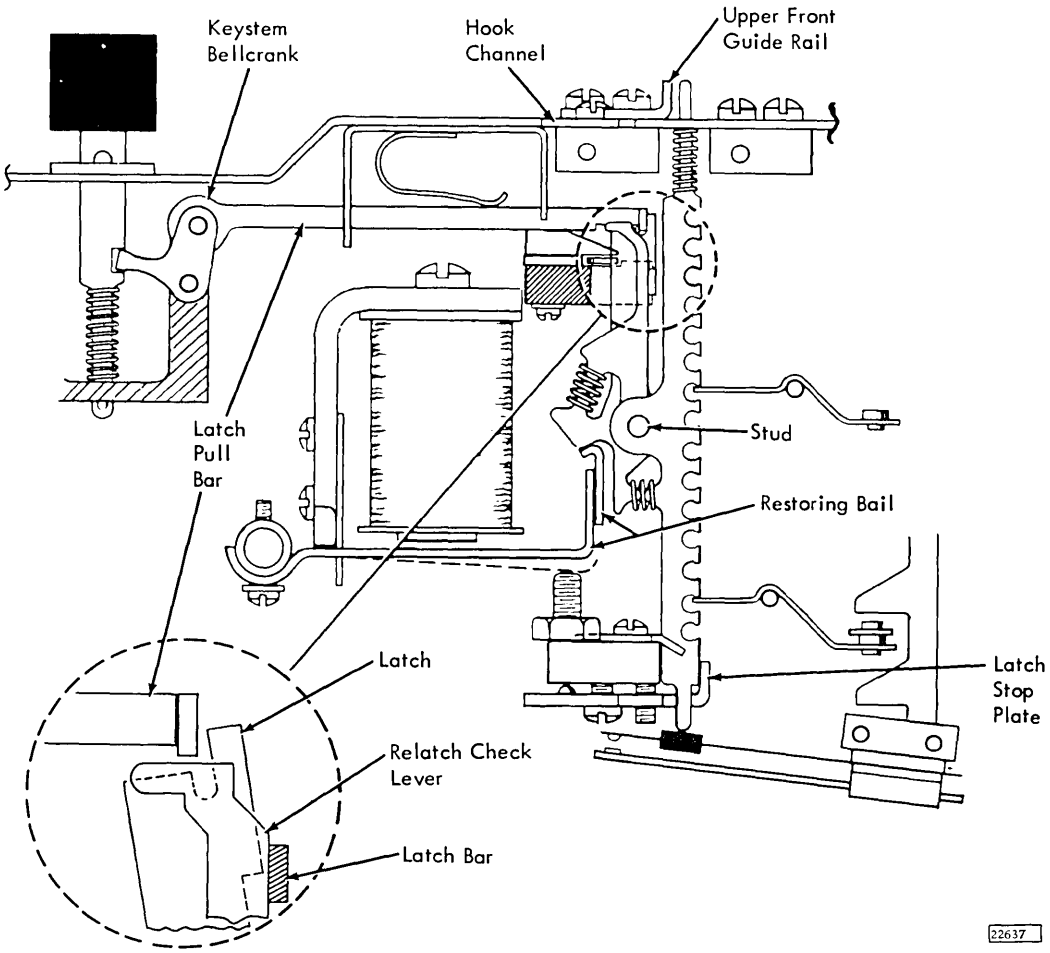


Figure 2-31. Key Position (Latched and Tripped)

stored. When a latch is pulled off the latch bar, the pull bar engaging slot is covered by the relatch check lever.

The permutation bar is restored along with the latch by the latch restoring bail. Energizing the restoring magnets operates the restoring bail and opens the restoring bail contact. The restoring bail is attached to the armatures of the keyboard restoring magnets. The restoring bail restores the latch to the latch bar. The keyboard contains several latch contacts, fifteen bail contacts, but only one restoring bail contact. The opening of the keyboard restoring bail contact signals that the latch has been restored to the latch bar. For correct keyboard operation, only one latch at a time can be pulled off the latch bar, thus preventing the operation of more than one key at a time.

2.7.4 Keyboard Interlocks

- Interlock disks prevent more than one latch at a time from being pulled off the latch bar.
- Interlock disks prevent more than one key at a time from being pressed.

Figure 2-32 shows the interlock disks in the trough in which they are contained. The permutation bars and their latch assemblies are positioned side by side equal distances apart. The interlock disks are located between each latch. The disks are placed side by side in the trough, and the line of disks spans the row of latches. Any time two disks are separated, all the disks will be moved.

Figure 2-33 shows five interlock disks with latch B first between two of them. When a latch is forced between any two disks, the disks to the right and to the left of it will move. A key was pressed in Figure 2-33 and latch B was pulled off the latch bar. Before latch B could be restored, the operator pressed the key for latch D. Latch D cannot be pulled off the latch bar until latch B is restored.

When a latch is operated and the disks are separated, all the other latches are inoperative

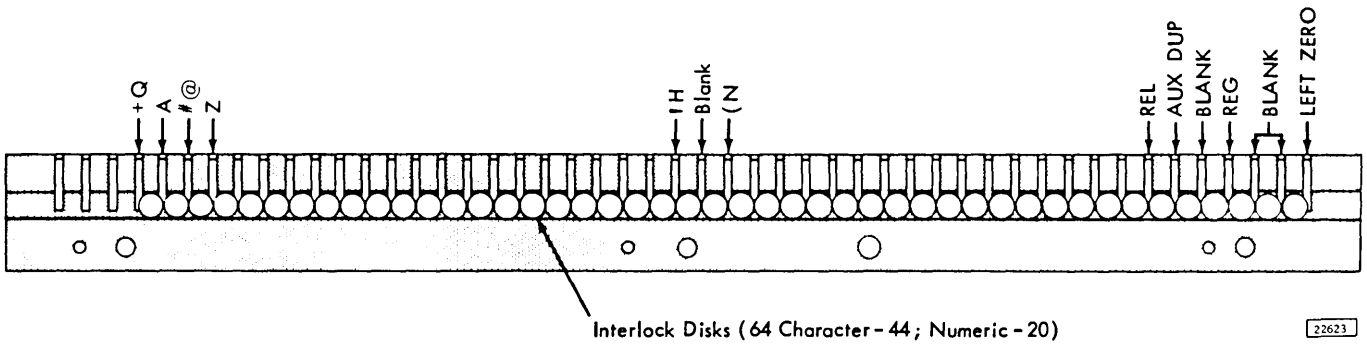


Figure 2-32. Interlock Disks

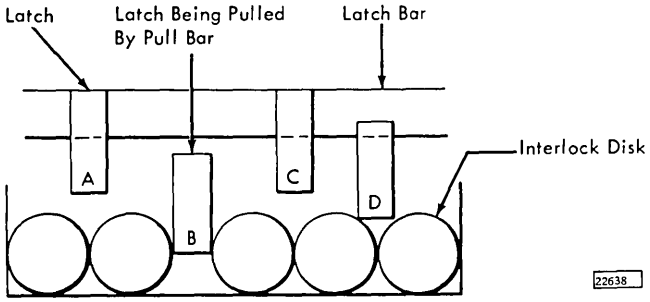


Figure 2-33. Interlock Disks - Detail

because the interlock disks cannot be spread any further apart. When a key is pressed, the latch must be restored before another key can be pressed.

When a keyboard is locked up (interlocked), it can be restored (unlocked) by pressing the keyboard restore key. This key has a section cut away in its latch so that it can be pulled off the latch bar regardless of the position of the interlock disks.

2.7.5 Keyboard Electrical Functions

Pressing a keybutton on the keyboard forces a keystem down and actuates a permutation bar. This mechanical linkage closes a latch contact, a bail contact, a keystem contact, or a combination of latch and bail contacts. Certain function keys (other than character keys) close keystem contacts only. All permutation bars do not operate both latch and bail contacts. The keyboard reference chart on page ZK121 of the ALDs, shows all the combinations of contacts, the card code, and the symbol associated with each keystem. Page ZK101 of the ALDs shows the wiring of the keyboard.

LOAD AND STORE OPERATIONS

- Load operations normally transfer data from core storage to the machine register specified in that instruction.
- Store operations normally transfer data from the machine register specified in the instruction to core storage.
- The machine register is effectively reset to zero before the new data is loaded into it.
- The core storage location addressed by a store operation is effectively reset to zero before the new data is transferred into it.
- Reference Maintenance Diagrams: AA101, AA211, AA221.

Load and store operations transfer data within the system and set up factors in arithmetic operations.

The word "load" can be interpreted as "transfer data into". For example, the instruction load accumulator means: transfer data into the accumulator: the instruction load index register means: transfer data into an index register.

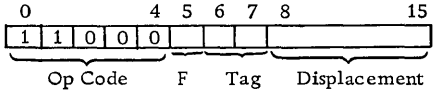
The word "store" can be interpreted as "transfer data out of". For example, the instruction store accumulator means: transfer data out of the accumulator into core storage. The instruction store index register means: transfer data out of the index register into core storage.

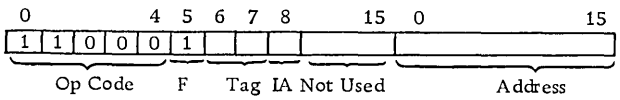
Combinations of these instructions are used to transfer data from one core storage location to another or from one register to another.

The system uses an add-to-accumulator circuitry. Therefore, one factor of each arithmetic of logical operation must be loaded into the accumulator before the actual arithmetic or logical instruction is given.

3.1 LOAD ACCUMULATOR

Mnemonic	Op Code	Format	IX	IA
LD	11000	0/1	No/Yes	Yes





- Transfer the contents of the core storage location specified by the effective address (EA) of the instruction into the accumulator (A).
- The contents of the core storage location are unchanged.
- The carry and overflow indicators are not changed.
- Hexadecimal operation code is C000, assuming no F, tag, IA, or displacement bits.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA681.

Table 3-1. Determining Effective Address

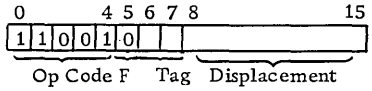
	F = 0 (Direct Addressing)	F = 1, IA = 0 (Direct Addressing)	F = 1, IA = 1 (Indirect Addressing)
T = 00	EA = Disp + IAR	EA = Add	EA = C/Add
T = 01	EA = Disp + XR1	EA = Add + XR1	EA = C/Add + XR1
T = 10	EA = Disp + XR2	EA = Add + XR2	EA = C/Add + XR2
T = 11	EA = Disp + XR3	EA = Add + XR3	EA = C/Add + XR3

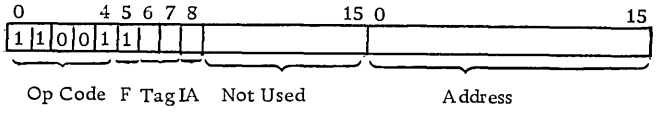
Disp = Contents of Displacement field of instuction.
Add = Contents of Address field of instruction.
C = Contents of Location specified by Add or Add + XR.

201118

3.2 LOAD DOUBLE

Mnemonic	Op Code	Format	IX	IA
LDD	11001	0/1	Yes/No	Yes/No

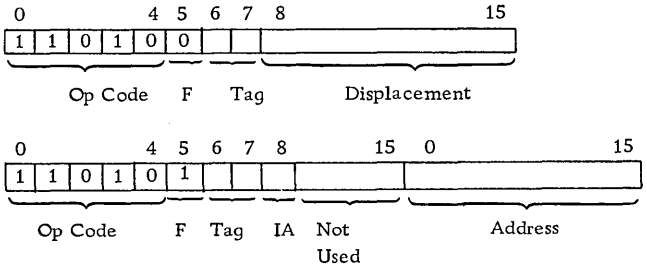




- Load the contents of the memory location specified by the instruction (EA) and the next higher memory location into the accumulator and its extension, respectively.
- This instruction provides double precision load for use with the double precision arithmetic.
- The EA of the instruction must be an even ad dress for correct operation.
- If the EA is odd, the contents of that location will be entered into both the accumulator and its extension.
- The contents of memory remain unchanged.
- Carry and overflow indicators are not changed by this operation.
- Hexadecimal operation code is C800, assuming no F, tag, IA, or displacement bits.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA682.

3.3 STORE ACCUMULATOR

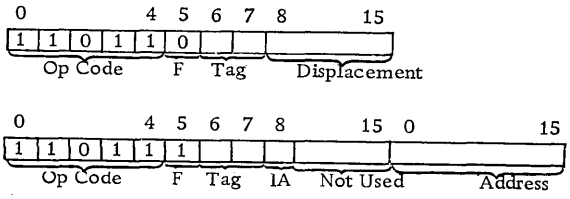
Mnemonic	Op Code	Format	IX	IA
STO	11010	0/1	Yes/No	Yes/No



- Store the contents of the accumulator at the memory location specified by the effective address.
- The contents of the accumulator are unchanged.
- Carry and overflow indicators are not changed by this operation.
- Hexadecimal operation code is D000, assuming no F, Tag, or displacement bits.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA683.

3.4 STORE DOUBLE

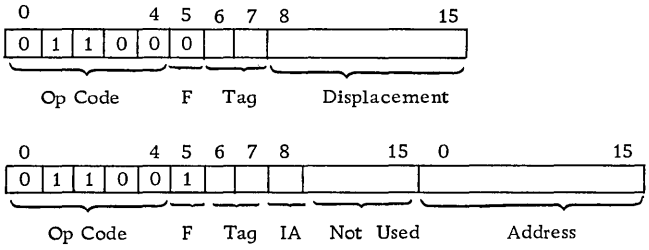
Mnemonic	Op Code	Format	IX	IA
STD	11011	0/1	No/Yes	No/Yes



- Load the contents of the Accumulator (A) and its extension (Q) into the memory locations specified by the effective address (EA) and the EA + 1.
- This instruction provides double precision store for use with double precision arithmetic.
- The EA of this instruction must be an even address for correct operation.
- If the EA is odd, the contents of the accumulator extension does not appear in core storage.
- The contents of A and Q remain unchanged.
- Carry and overflow indicators are not changed by this operation.
- Hexadecimal operation code is D800, assuming no F, tag, or displacement bits.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA684.

3.5 LOAD INDEX

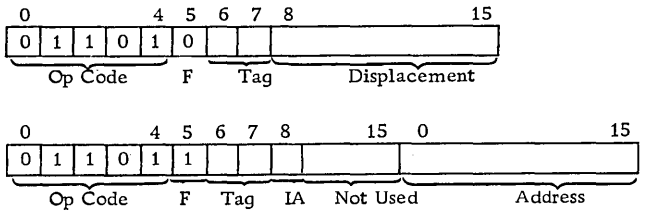
Mnemonic	Op Code	Format	IX	IA
LDX	01100	0/1	No	No/Yes



- The hexadecimal operation code of the LDX instruction is 6000 assuming no F, Tag, and IA bits.
- Tag = 00, data is loaded into IAR.
- Tag = 00, load index is an unconditional branch to the address loaded into IAR.
- Tag ≠ 00, data is loaded into the register specified by the tag bits.
- F = 0, displacement data is the data loaded.
- When the displacement is used, the eight high-order positions of the specified register are filled with the value of the sign bit (bit position 8 of instruction) to complete the 16-bit word.
- F = 1, IA = 0, address word is the data loaded.
- F = 1, IA = 1, contents of storage specified by the address word is the data loaded.
- Carry and overflow indicators are not affected.
- Reference Maintenance Diagram: AA661.

3.6 STORE INDEX

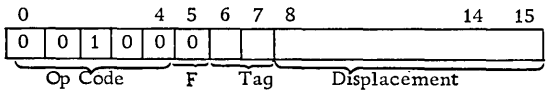
Mnemonic	Op Code	Format	IX	IA
STX	01101	0/1	No	No/Yes



- The hexadecimal operation code of the STX instruction is 6800 assuming no F, Tag, and IA bits.
- Tag = 00, store contents of IAR at the effective address.
- Tag ≠ 00, store contents of the specified index register at the effective address.
- Refer to Table 1 for effective address.
- Carry and overflow indicators are not affected.
- Reference Maintenance Diagram: AA662

3.7 LOAD STATUS

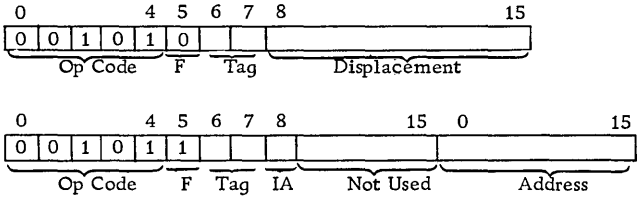
Mnemonic	Op Code	Format	IX	IA
LDS	00100	0	No	No



- This instruction uses the single-word format only.
- Load the carry and overflow indicators with the on/off status of the bits in positions 14 (carry) and 15 (overflow) of the instruction.
- The carry and overflow status is normally stored into this instruction by a previous store status instruction.
- Core storage is not changed.
- The hexadecimal operation code of this instruction is 200x, assuming no F and T bits and zero bits in positions 8-11.
- The units hexadecimal operation code integer depends on how the indicators are to be set.
- Reference Maintenance Diagram: AA641.

3.8 STORE STATUS

Mnemonic	Op Code	Format	IX	IA
STS	00101	0/1	Yes/No	No/Yes



- Store the on/off status of the carry and overflow indicators in bits 14 (carry) and 15 (overflow) of the word at the effective address (EA).
- Bits 0 through 7 of the word remain unchanged and bits 8 through 13 are reset to zero.
- Carry and overflow indicators are reset as they are stored.
- The hexadecimal operation code for this instruction is 2800 assuming no F, Tab, or displacement bits.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA642.

BRANCH AND SKIP OPERATIONS

- Branch instructions in a program permit alteration of the sequential execution of the program.
- Unconditional branch instructions always alter the sequential execution of a program.
- Conditional branch instructions alter the sequential execution of a program if the tested condition is present.
- Reference Maintenance Diagram: AA221.

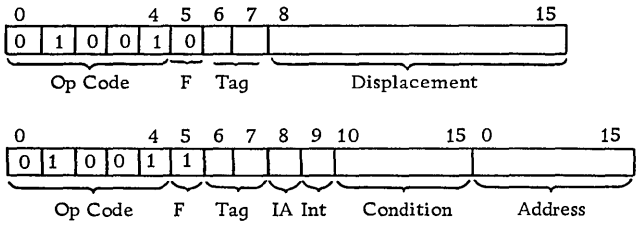
Branch instructions may be classified into two categories:

1. Unconditional branch instructions:
   a. Branch and store instruction counter, with F = 0.
   b. Load index, when tag = 00.
2. Conditional branch instructions:
   a. Branch or skip on condition.
   b. Branch and store instruction register with F = 1.

Whether or not branching occurs as a result of conditional branch instructions is dependent upon the status of a condition within the computer.

3.9 BRANCH OR SKIP ON CONDITION

Mnemonic	Op Code	Format	IX	IA
BSC	01001	0/1	Yes/No	Yes/No



- If F = 0: skip next sequential one word instruction if any specified condition is present. If modifier bit 9 = 1, reset highest level interrupt on.
- If F = 0; and the condition bits are all zeros, the instruction acts a no-op.
- If F = 1: branch to the instruction at the location specified by the effective address if no specified condition is present. If modifier bit 9 = 1, reset highest level interrupt on.
- If F = 1, and the condition bits are all zeros, the instruction acts as a unconditional branch.
- Must be sequentially followed by a single-word instruction if this instruction is a single word.
- Core storage and accumulator extension are not changed by this instruction.
- Overflow indicator is turned off if tested.
- Carry indicator is not turned off.
- Accumulator is not changed by testing.
- Six conditions associated with the accumulator can be tested (Figure 3-1).
- Hexadecimal operation codes for this instruction are 48 (no tag), 49 (tag 01), 4A (tag 10), 4B (tag 11).
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA652.

There are six testable conditions (Figure 3-1) associated with the accumulator. These conditions are tested by indicating the bit pattern in the displacement of the instruction.

If F = 0 (single-word instruction), the instructions performs as a skip on condition. If any of the skip conditions specified by the displacement exist, the address in the instruction address register is increased by 1. Modifying the instruction address register in this manner causes the next instruction (which must be a single word) to be skipped.

If none of the skip conditions exist, the next instruction (single-word) in sequence is entered.

If F = 1 (Double-word instruction), this instruction performs as a branch no condition. If none of the skip conditions specified by the displacement

Displacement Bit Position	Condition Tested
15	Overflow Indicator Off
14	Carry Indicator Off
13	Accumulator Even
12	Accumulator Plus (>0)
11	Accumulator Negative (<0)
10	Accumulator = Zero

24039

Figure 3-1. Testable Conditions

exist, the effective address is placed into the I register. Modifying the instruction address register in this manner causes a branch to the instruction located at the effective address.

If any of the skip conditions exist, the next instruction (single-or-double-word) in sequence is entered.

An example of a BSC operation is shown in Figure 3-2.

Programming Note. When the IA bit is equal to a one (IA = 1), this instruction enables the program to return to a mainline program from a program subroutine or interrupt routine. This is accomplished by making the EA of this instruction identical to the EA of a previously executed branch and store instruction register (BSI) instruction. The EA as calculated below is loaded into the instruction register.

When an interrupt request has been detected by a priority level, the program is directed to service the request by interrupting. During the servicing, all interrupt requests of equal or lower status are effectively constrained from interrupting while the servicing of the higher priority is in progress. If a request for a higher priority level is detected, 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 turn off the priority-status of the highest level that is on. This reset permits lower priority requests that have been temporarily constrained but recorded to be accepted once again by the CPU. This is effected by making bit 9 = 1 (BOSC operation code hex 4840) in this instruction. This programmed recognition of waiting interrupts should not be confused with a normal subroutine linkage back to a mainline program in which bit 9 should be set to zero.

The BSC is a conditional instruction. When bit 9 = 1, the reset of the interrupt level occurs when the branch or skip occurs. If the branch or skip does not occur, the interrupt level is not reset.

Indicators. The overflow indicator is reset if tested. The carry indicator is not reset by testing. The contents of the accumulator are not changed by testing.

Instruction Word								Skip Conditions						Result
F		Displacement								Accumulator Status				
5	-	10	11	12	13	14	15	O'flow	Carry	Even	Plus	Minus	Zero	
0		0	0	1	0	1	0	-	-	[Yes]	-	No	-	Skip
0		1	0	1	0	0	1	On	-	-	No	-	No	No Skip
0		0	1	0	0	1	0	-	[Off]	-	-	No	-	Skip
0		1	0	1	0	0	1	[Off]	-	[Yes]	-	-	No	Skip
0		1	1	1	1	1	1	On	On	No	No	No	[Yes]	Skip
0		0	0	0	0	0	0	-	-	-	-	-	-	No Skip
1		1	0	1	1	0	1	On	-	[Yes]	No	-	No	No Branch
1		0	0	0	0	0	0	-	-	-	-	-	-	Branch
1		1	1	1	1	1	1	On	On	[Yes]	No	[Yes]	No	No Branch
1		1	0	1	0	1	1	[Off]	-	No	-	No	No	No Branch
1		0	0	1	0	1	0	-	-	No	-	No	-	Branch

[ ] Indicates condition causing skip or preventing branch.

24040 A

Figure 3-2. BSC Examples

Hexadecimal. The hexadecimal versions of the BSC instruction depend on the F, T, IA, INT and condition bits.

48XXX	Skip on any condition
4C00	Branch to ADDR unconditionaly
4C0X	Branch to ADDR on no condition
4D0X	Branch to contents of XR1 + ADDR on no condition
4E0X	Branch to contents of XR2 + ADDR on no condition
4F0X	Branch to contents of XR3 + ADDR on no condition

The tens digit for the four instructions above (4C0X - 4F0X) may actually be any digit from 0 to 7.

4C8X	Branch to contents of ADDR on no condition
4D8X	Branch to contents of XR1 + contents of ADDR on no condition
4E8X	Branch to contents of XR2 + contents of ADDR on no condition
4F8X	Branch to contents of XR3 + contents of ADDR on no condition

The tens digit for the four instructions above may actually be any integer from 8 through F.

3.9.1 Circuit Description - One Word Instruction - BSC

1-1: Read and interpret first word of the instruction:

T1 - Transfer ACC to the U register.
T2 - Read core storage to B register.
Transfer displacement bits to D register.
Transfer bits 0-9 to Op, tag, flag and mod register.
T3 - Set add FF.
Transfer I register +1 to the ACC (normal I-1 operation - not used)
Compare skip condition bits - B10 - B15
Reset overflow FF if tested - B15.
T4 - If one or more of the conditions tested are true set skip condition FF. If not skip to T7b.
Write B register back into core storage.
T5 - If skip condition is met increment the I register (skip next one word instruction).
T7a - If skip condition and mod bit 9 are on, reset highest level interrupt that is on.
T7b - Transfer U register to the ACC.
End of operation, enter I-1 cycle for next instruction.

3.9.2 Circuit Description - Two Word Instruction - BSC

I-1: Read and interpret first word of the instruction:

T1 - Transfer ACC to the U register.
T2 - Read core storage to B register.
Transfer displacement bits to the D register.
Transfer bits 0-9 to Op, tag, flag, and mod registers.
T3 - Set add FF.
Transfer I register +1 to the ACC (normal I-1 operation - not used).
Compare skip condition bits - B10 - B15.
Reset overflow FF if tested - B15.
T4 - If one or more of the conditions tested are true set skip condition FF (No branch). If not, skip to T5b.
Write B register back into core storage.
T5a - Increment the I register (skip address word).
T7a - True condition and mod bit 9 is on, turn off the highest level interrupt that is on.
Transfer U register to ACC.
End operation, enter I-1 cycle of next instruction.
T5b - Set CCC to 1. (Prepare to enter I-2.)

I-2: Develop EA

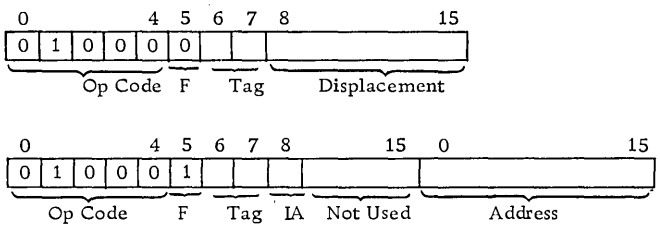
T0 - Set I-2 F-F
Transfer I to M register (read address word).
Increment the I register.
T2 - Read address word to the B register.
Transfer the B register to the D register.
T4 - Transfer the D register to the ACC register. (address word = EA)
T6 - Set Branch 1 FF.
T7 Extended - Set branch 2 FF.
Transfer the ACC to the M register. (EA)
Turn off branch 1 FF.
Turn off the CCC FF's.
Transfer the M register to the I register. (EA).
Transfer the U register to the ACC register.
Turn off the branch 2 FF.
End of operation, enter I-1 cycle at EA for next instruction.

NOTE: The ADDR specified may be indirect, in which case the EA is the contents of ADDR. This instruction is frequently used as the last instruction of an interrupt subroutine to return to the mainline program.

This is accomplished by making the EA of this instruction identical to the EA of the forced branch and store IAR instruction that caused the branch to the interrupt subroutine.

3.10 BRANCH AND STORE INSTRUCTION REGISTER

Mnemonic	Op Code	Format	IX	IA
BSI	01000	0/1	Yes/No	Yes/No



- Store contents of instruction address register at the core storage location specified by the effective address.
- Branch by setting instruction address register to the value of the effective address plus one.
- If F = 0, the branch and store functions are unconditional.
- If F = 1, the branch and store functions are performed only if none of the BSC skip conditions (Figure 3-1) specified by the displacement are present.
- Overflow indicator is turned off if tested.
- Carry indicator is not turned off.
- Accumulator is not changed by testing.
- Six conditions associated with the accumulator can be tested. (Figure 3-1.)
- Hexadecimal operation codes for this instruction are 40 through 47.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA651.

When F = 0 (one word format), the contents of the instruction register are stored in the core memory location specified by the effective address. The stored address is that of the next instruction in the normal sequence. The instruction register is then set to the value of the effective address plus one, and program execution proceeds from that point.

For example, a BSI instruction located at core memory address 0500, with an effective address of 0600, would store the address 0501 at location 0600 and then branch to 0601.

3.10.1 Circuit Description - One Word Instruction - BSI

I-1: Read and interpret first word of the instruction:

T1 - Transfer the ACC to the U register.
T2 - Read core storage to the B register.
Transfer displacement bits to the D register.
Transfer bits 0-9 to operation, tag, flag and mod registers.
T3 - Set add FF.
Transfer I register +1 to the ACC (will be used to calculate EA).
Set arith control FF.
T4 - T7 - Perform add operation (displacement to I + 1 = (EA) T5, set CCC to 1 (take one E cycle).
Write B register back into core storage.

E-1: Store I + 1 at EA.

T0 - Transfer ACC to M register. (EA)
T1 - Transfer the U register to the ACC register.
Decrement the CCC.
T3 - Transfer the I register to the B register. (old address +1)
T4 - Store B register in core storage.
Transfer the M register to the I register. (EA)
T5 - Increment the I register. (EA +1)
T7 - End of operation, enter I-1 cycle for next instruction.

3.10.2 Circuit Description - Two Word Instruction -BSI

I-1: Read and interpret first word of the instruction:

T1 - Transfer the ACC to the U register.
T2 - Read core storage to the B register.
Transfer the displacement bits to the D register.
Transfer bits 0-9 to operation, tag, flag and mod registers.
T3 - Set add FF.
Transfer the I register +1 to the ACC (will be used to calculate EA).
Compare skip condition bits - B10 - B15.
Turn off overflow FF if tested - B15.
T4 - If one or more of the conditions tested are true set skip condition FF. (no branch) If not skip to T5b.
Write B register back into core storage.
T5a - Increment the I register (skip address word)
T7a - True condition and mod bit 9 is on, reset highest level interrupt that is on.
Transfer the U register to the ACC.
End operation, enter I-1 cycle of next instruction.
T5b - Set CCC to 1. (Prepare to enter I-2)

I-2: Develop EA.

T0 - Set I-2 FF.
Transfer the I register to the M register (read address word).
Increment the I register.

T2 - Read the address word to the B register.
Transfer the B register to the D register.

T4 - Transfer the D register to the A register. (address word = EA)

T7 - Enter XR cycle if tag ≠ 00, enter IA cycle if IA = 1 or E-1 cycle if tag = 00, IA = 0.

E-1: Store I +1 at EA

T0 - Transfer the A register to the M register. (EA)

T1 - Transfer the U register to the A register. Decrement the CCC.

T3 - Transfer the I register to the B register (address of next instruction)

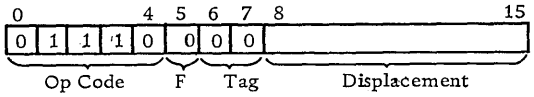
T4 - Store the B register in core storage. Transfer the M register to the I register. (EA)

T5 - Increment the I register. (EA +1)

T7 - End of operation, enter I-1 cycle of next instruction at EA +1.

3.11 MODIFY INDEX, F = 0, TAG = 00

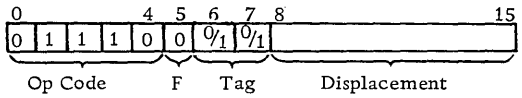
Mnemonic	Op Code	Format	IX	IA
MDX	01110	0	No	No



- Add contents of displacement to the contents of the I register plus one (effective address).
- The displacement may be negative.
- Transfer the effective address to the I register.
- Branches to new instruction at the effective address.
- The contents of the accumulator are unchanged.
- Carry and overlow indicators are not changed by this operation.
- Only one I (I-1) cycle used.
- Reference Maintenance Diagram: AA663, Sheet 1.

3.12 MODIFY INDEX, F = 0, TAG ≠ 00

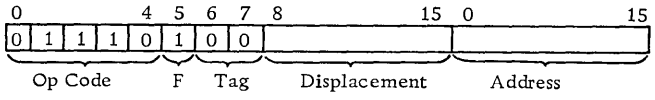
Mnemonic	Op Code	Format	IX	IA
MDX	01110	0	No	No



- The displacement is added to the contents of the index register specified by the tag bits.
- Set up the displacement as the effective address in the I1 cycle.
- Select the index register.
- Add effective address to the selected index register.
- Skip next one word instruction if the index register changes sign or becomes zero.
- I-1, E-1 and E-2 cycles used.
- Carry and overflow indicators are not changed by this operation.
- The contents of the accumulator are not changed by this instruction.
- Reference Maintenance Diagram: AA663, Sheet 2.

3.13 MODIFY INDEX, F = 1, TAG = 00

Mnemonic	Op Code	Format	IX	IA
MDX	01110	1	No	IA

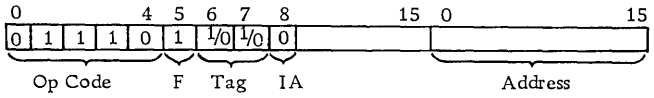


- The displacement of the first word of the instruction is added to the contents of the core location specified by the address word.
- I-1 and I-2 cycles - Set contents of address word (modify factor address) in the ACC.

- Force a IA cycle - set contents of the address word location (modify factor) in the ACC.
- E-1 cycle - address MDX instruction word, add the displacement to the modify factor.
- E-2 cycle - Store new factor at the modify factor location.
- Skip next one word instruction if the modify factor changes sign or becomes zero.
- Carry and overflow indicators are not changed by this operation.
- The contents of the accumulator are not changed by this instruction.
- Reference Maintenance Diagram: AA663, Sheet 3 and 4.

3.14 MODIFY INDEX, F = 1, TAG ≠ 00, IA = 0

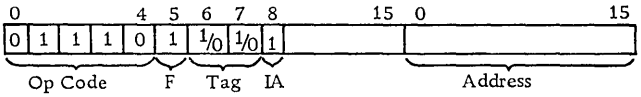
Mnemonic	Op Code	Format	IX	IA
MDX	01110	1	No	No



- The contents of the address word of the instruction is added to the index register specified by the tag bits.
- I-1 and I-2 cycles - set contents of the address word (modify factor) in the ACC.
- Select the index register.
- E-1 cycle, add contents of the index register to the ACC (modify factor).
- E-2 cycle store the new factor in the index register.
- Skip the next one word instruction if the index register changes sign or becomes zero.
- Carry and overflow indicators are not changed by this operation.
- The contents of the accumulator are not changed by this instruction.
- Reference Maintenance Diagram: AA663, Sheet 2.

3.15 MODIFY INDEX, F = 1, TAG ≠ 00, IA = 1

Mnemonic	Op Code	Format	IX	IA
MDX	01110	1	No	Yes



- The contents of the core storage location specified by the indirect address is added to the specified index register.
- I-1, I-2 and IA cycles - Set contents of the address word location (modify factor) in the ACC.
- E-1 add the contents of the index register to the ACC (modify factor).
- E-2 cycle store the new factor in the index register.
- Skip the next one word instruction if the index register changes sign or becomes zero.
- Carry and overflow indicators are not changed by this operation.
- The contents of the accumulator are not changed by this instruction.
- Reference Maintenance Diagram: AA663, Sheet 2.

Table 2 shows the relationships of the different MDX instructions.

Table 2. MDX Functions

Format	Tag	IA	Add	Skip
0	00	-	Displ. to I	Unconditional
0	01		Displ. to XR-1	Conditional*
0	10	-	Displ. to XR-2	
0	11	-	Displ. to XR-3	
1	00	-	Displ. to Contents of core storage (specified by Address word).	
1	01	0	Address word to XR-1	
1	10	0	Address word to XR-2	
1	11	0	Address word to XR-3	
1	01	1	Contents of core storage specified by Addr. word - C (Addr) to XR-1	
1	10	1	C (Addr.) to XR-2	
1	11	1	C (Addr.) to XR-3	

Skip following MDX if modified factor reaches zero or changes sign while being modified.

22264

SHIFT OPERATIONS

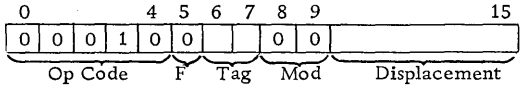
- All shift instructions are single word format only (F = 0).
- The index registers or the displacement specify the number of shifts required.
- If the shift count is zero, the instruction performs as a no-operation.
- Shift instructions are divided into classes by operation code and into subclasses by bit positions 8 and 9.
- Reference Maintenance Diagram: AA221.

Shift instructions are divided into two major classes, shift left and shift right. These major classes are defined by the operation code of the instruction. Each of these major classes is divided into subclasses. These subclasses are defined by decoding at bits 8 and 9 (modifier (Mod) bits) of the instruction (Figure 3-3).

The location of the shift count is defined by the tag bits of the instruction (Figure 3-4).

3.16 SHIFT LEFT A

Mnemonic	Op Code	Format	IX	IA
SLA	00010	0	No/Yes	No



- Shift the accumulator bits to the left the number of positions specified by the cycle control counter.

Instruction	B8	B9	Tag
Shift Left			
Shift Left A	0	0	
Shift Left A + Q	1	0	
Shift Left and Count A	0	1	≠00
Shift Left and Count A + Q	1	1	≠00
Shift Right			
Shift Right A	0 or 0	1 or 0	
Shift Right A and Q	1	0	
Rotate Right A + Q	1	1	

22222

Figure 3-3. Shift Modifiers

XR Bits		Shift Count Located In:
B6	B7	
0	0	Low-Order 6 bits of DISPLACEMENT
0	1	Low-Order 6 bits of XR-1
1	0	Low-Order 6 bits of XR-2
1	1	Low-Order 6 bits of XR-3

24042

Figure 3-4. Tag Bits - Shift Instructions

- If tag bits ≠ 00 set CCC with contents of the XR register.
- Fill all vacated positions of the accumulator with zeros.
- Shift bits out of the high-order of the accumulator into the carry indicator.
- Core storage and the accumulator extension are not changed by the instruction.
- Hexadecimal operation codes for this instruction are 10 (no tag), 11 (tag 01), 12 (tag 10), 13 (tag 11).
- The overflow indicator is not affected by this instruction.
- Reference Maintenance Diagram: AA631.

The accumulator is shifted left the number of spaces specified by the displacement or by the contents of a XR register. Bits leaving the high-order (bit 0) position are shifted into the carry indicator.

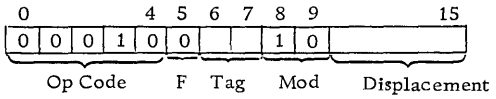
The carry indicator is turned on for each one and off for each zero shifted left from the high-order position of A.

Hexadecimal versions of the SLA instruction depend on the tag bits (XX is the unknown value of the 6 low-order bits of the displacement when the tag bits are 00).

10XX	Shift count in displacement
1100	Shift count in XR1
1200	Shift count in XR2
1300	Shift count in XR3

3.17 SHIFT LEFT A & Q

Mnemonic	Op Code	Format	IX	IA
SLT	00010	0	No/Yes	No



- Shift the accumulator and its extension (treated as a 32 bit double precision word) to the left the number of positions specified by the shift count.
- Fill all vacated positions of the accumulator and its extension with zeros.
- Shift bits out of the high-order of the accumulator into the carry indicator.
- Core storage is not changed.
- Hexadecimal operation codes for this instruction are: 10 (No tag), 11 (tag 01), 12 (tag 10), 13 (tag 11).
- Overflow indicator is not affected.
- Reference Maintenance Diagram: AA631.

The accumulator (A) and its extension (Q) are shifted left as a 32-bit double-precision register. Bits leaving the high-order position (bit position 0 of A) are shifted into the carry indicator.

The carry indicator is turned on for each one and off for each zero shifted left from high-order position of A.

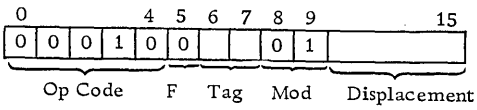
The hexadecimal versions of this instruction depend on the tag bits.

*108X	Shift count is displacement
1180	Shift count in XR1
1280	Shift count in XR2
1380	Shift count in XR3

*The tens position of 108X can actually be any integer from 8 through B, depending on the unknown bit values in positions 10 and 11. The unknown values of bit positions 12-15 are represented by X.

3.18 SHIFT LEFT AND COUNT ACC

Mnemonic	Op Code	Format	IX	IA
SLCA	00010	0	No/Yes	No



- If tag bits = 00, this instruction performs like a shift left A instruction.
- If tag bits ≠ 00, transfer the contents of XR register to the CCC and shift the accumulator to the left, one position at a time.
- Stop shifting when a 1 bit is detected in the bit 0 position of the accumulator or the shift counter is decremented to zero. Set shift count remainder into the addressed index register and set the carry indicator.
- Set all vacated positions to zero.
- Carry indicator is turned on by a one bit in the high-order position of the accumulator.
- Overflow indicator is not effected.
- Hexadecimal operation codes for this instruction are: 10 (no tag), 11 (tag 01), 12 (tag 10), 13 (tag 11).
- Reference Maintenance Diagram: AA631.

If the tag bits specify an index register, the shift count is transferred from the low-order six bits of the addressed index register to the CCC. If the tag bits are 00 set the displacement into the CCC. This count is decremented by one for each bit position that the word in the accumulator is shifted to the left.

The shift terminates when a 1 is shifted into the high-order position of A or when the shift count is decremented to zero.

After the shift is completed, the decremented count is then loaded back into the six low-order bit positions of the index register. Bit positions 0-7 of the index register remain unchanged at completion of the instruction.

Indicators. The carry indicator will be off if the shift is terminated by the detection of the count reaching zero. The carry indicator will be on if the shift is terminated by the detection of a 1 in the A0 position before the shift count reaches zero.

If tag bits = 00, the carry indicator is set as in the shift left instruction.

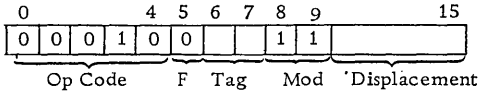
The hexadecimal version of this instruction depends on the tag bits.

104X	Shift count in displacement
1140	Shift count in XR1

1240 Shift count in XR2
1340 Shift count in XR3

3.19 SHIFT LEFT AND COUNT A & Q

Mnemonic	Op Code	Format	IX	IA
SLC	00010	0	No/Yes	No



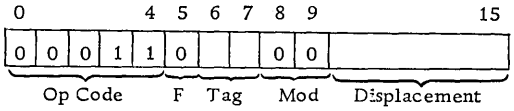
- If tag bits = 00, this instruction performs like a shift left A & Q instruction.
- If tag bits ≠ 00, transfer the contents of the XR register to the CCC and shift the accumulator and its extension to the left one position at a time.
- Bits shifted out of the high-order position of the extension are transferred into the low-order position of the accumulator.
- Stop shifting when CCC is decremented to zero or when a one bit is detected in the high-order position of the accumulator.
- Set all vacated extension positions to zero.
- Carry indicator is turned on by a one bit in the high-order position of the accumulator, or if the shift is terminated by an A0 bit = 1.
- Overflow indicator is not effected.
- Hexadecimal operation codes for this instruction are: 10 (no tag), 11 (tag 01), 12 (tag 10), 13 (tag 11).
- Reference Maintenance Diagram: AA631.

This instruction is the same as the shift left and count A except that both the accumulator and its extension are shifted. Bit position 0 of Q is shifted into bit position 15 of A and vacated positions at the right of Q are set to zero.

The hexadecimal versions of the SLC instruction are 10XX, (the tens position must be C through F), 11C0, 12C0, and 13C0.

3.20 SHIFT RIGHT A

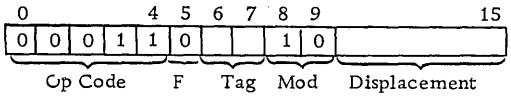
Mnemonic	Op Code	Format	IX	IA
SRA	00011	0	No/Yes	No



- Shift the accumulator bits to the right the number of positions specified by the cycle control counter.
- Tag = 00, set the cycle control counter with the displacement.
- Tag ≠ 00, set the cycle control counter with the contents of the index register.
- Fill all vacated positions of the accumulator with zeros.
- Low-order bits of the accumulator are lost.
- Core storage and the accumulator extension are not changed by this instruction.
- Hexadecimal operation codes for this instruction are 18 (no tag), 19 (tag 01), 1A (tag 10), 1B (tag 11).
- Carry and overflow indicators are not affected by this instruction.
- Reference Maintenance Diagram: AA632.

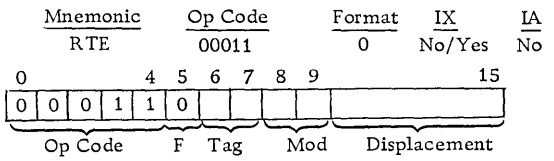
3.21 SHIFT RIGHT A & Q

Mnemonic	Op Code	Format	IX	IA
SRT	00011	0	No/Yes	No



- Shift the accumulator and its extension to the right the number of positions specified by the displacement. The accumulator and extension are shifted right as a 32-bit double precision register.
- If tag bits = 00, set the CCC with the displacement.
- If tag bits ≠ 00, set the CCC with contents of the XR register.
- Fill all vacated accumulator positions with the sign of the accumulator (1 bit if negative, 0 bit if positive).
- Shift A15 to Q0.
- Low-order bits of the accumulator extension are lost.
- Core storage is not changed.
- Hexadecimal operation codes for this instruction are: 18 (no tag), 19 (tag 01), 1A (tag 10), 1B (tag 11).
- Carry and overflow indicators are not affected.
- Reference Maintenance Diagram: AA632.

3.22 ROTATE RIGHT A & Q



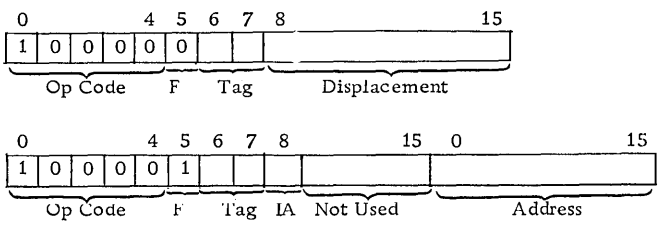
- Rotate the accumulator and its extension to the right the number of positions specified in the cycle control counter.
- If tag bits = 00, set the CCC with the displacement.
- If tag bits ≠ 00, set the CCC with the contents of the XR register.
- Bits from position 15 of the accumulator are transferred to position 0 of the extension.
- Bits from position 15 of the extension are transferred to position 0 of the accumulator.
- Carry and overflow indicators are unaffected.
- Core storage is not changed.
- Hexadecimal operation codes for this instruction are: 18 (no tag), 19 (tag 01), 1A (tag 10), 1B (tag 11).
- Reference Maintenance Diagram: AA632.

The accumulator and extension are rotated to the right as a 32 bit-double-precision register the number of bit positions specified by the displacement. If tag bits are not 00, the number of positions shifted are set by the XR register contents. Bit position 15 of the extension (A) is linked to bit position 0 of the accumulator (A) to form a continuous loop so that the high-order positions of the accumulator pick up the bits shifted from the low-order positions of the extension.

ARITHMETIC OPERATIONS

3.23 ADD

Mnemonic	Op Code	Format	IX	IA
A	10000	0/1	Yes/ No	No/Yes



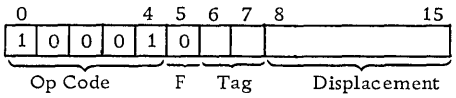
- Add the 16-bit word specified by EA to the 16-bit word in ACC.
- Replace the word in ACC with the result.
- Two's complement arithmetic is used; that is, negative operands and/or sums are in two's complement form.
- Core storage remains unchanged.
- The carry indicator is turned off at the beginning of the operation and is set by a subsequent carry out of the high-order bit position of the accumulator.
- The overflow indicator is turned on if the sum is too large to be accurately represented in the accumulator.
- If overflow was previously on, it is not changed.
- The hexadecimal operation code for this instruction is 8000, assuming no F, tag, or displacement bits.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA671.

Indicators. When the instruction is completed, the carry indicator represents the results of this instruction, not previous instruction. The carry indicator is set on by detection of a carry out of the high-order position of the accumulator.

The overflow indicator is turned on by this instruction if the sum is greater than $+2^{15}-1$ or less than -2^{15}. If this indicator was on before the instruction, no change will occur. If off, it will be turned on when the result is too large to be accurately represented. This is detected as shown on AA671.

3.24 DOUBLE ADD

Mnemonic	Op Code	Format	IX	IA
AD	10001	0/1	Yes/No	No/Yes



- Add the 32 bit-word specified by EA and EA +1 to the 32-bit word in the accumulator and its extender.
- The EA must be an even address.
- The sum replaces the contents of the accumulator and its extension.
- Core storage remains unchanged.
- The carry indicator is reset at the beginning of the operation and is set by a subsequent carry out of the high-order position of the accumulator.
- The overflow indicator is turned on if the result is too large to be accurately represented in the accumulator and its extender.
- The hexadecimal operation code for this instruction is 8800; assuming no F, tag, or displacement bits.
- Refer to Table 1 for effective address.
- References Maintenance Diagram: AA672.

This instruction provides double precision addition where the accumulator and its extension are considered as one 32-bit accumulator.

The effective address formed by this instruction must be an even address for correct operation. If EA is odd, the word at the EA location is added to both the accumulator and its extension.

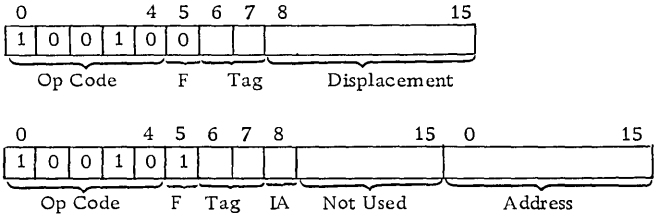
Indicators. The overflow indicator is turned on if the magnitude of the difference is too large to be represented in the accumulator and its extender; that is, it is greater than $+2^{31}-1$ or less than -2^{31}.

If overflow was previously on, it will not be changed. (Overflow can be reset by testing or by a load or store status instruction. See branch or skip on condition instruction). An overflow condition is detected by a borrow or carry from only one of the two high-order bit positions of the accumulator (AA671).

The carry indicator is set by a carry from the high order position of the accumulator.

3.25 SUBTRACT

Mnemonic	Op Code	Format	IX	IA
S	10010	0/1	Yes/No	No/Yes



- Subtract the word at the memory location specified by the effective address from the word in the accumulator.
- The result replaces the contents of the accumulator.
- Two's complement arithmetic is used; that is, any negative operand or results are in two's complement form.
- Core storage remains unchanged.
- The overflow indicator is turned on if the difference is too large to be accurately represented in the accumulator.
- The carry indicator is turned on by a carry or borrow from the high-order position of the accumulator.
- The hexadecimal operation code for this instruction is 9000, assuming no F, tag, or displacement bits.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA671.

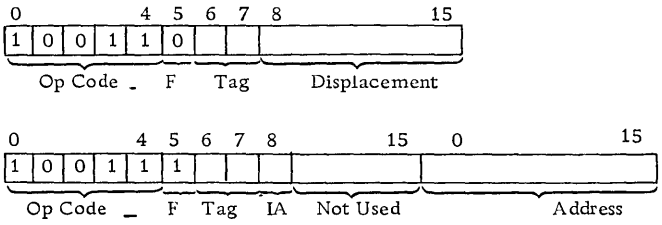
Indicators. The overflow indicator is turned on if the magnitude of the difference is too large to be represented in the accumulator; that is, greater than $+2^{15}-1$ or less than -2^{15}.

If overflow was previously on, it will not be changed. (Overflow can be reset by testing or a load or store status instruction. See branch or skip on condition instruction.) This is detected as shown on AA671.

The carry indicator is set by a borrow from the high order position of the accumulator.

3.26 DOUBLE SUBTRACT

Mnemonic	Op Code	Format	IX	IA
SD	10011	0/1	Yes/No	No/Yes



- Subtract the 32-bit word specified by EA and EA + 1 from the 32-bit word in the accumulator and its extender.
- The EA must be an even address.
- The difference replaces the contents of the accumulator and its extender.
- Core storage remains unchanged.
- If the difference is too large to be accurately represented in the accumulator and its extension, turn on the overflow indicator.
- The carry indicator is turned on by a borrow from the high-order position of the accumulator.
- The hexadecimal operation code for this instruction is 9800, assuming no F, tag and displacement bits.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA672.

This instruction provides double precision subtraction where the accumulator and its extension are considered as one 32-bit accumulator.

The effective address formed by this instruction must be an even address for correct operation. If the effective address is odd, the word at the location specified by EA is subtracted from both the accumulator and its extension.

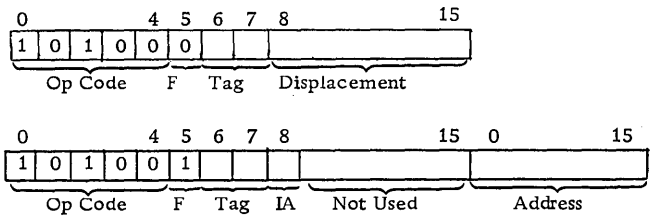
Indicators. The overflow indicator is turned on if the difference is too large to be accurately represented in the accumulator (A) and its extension (Q), or more specifically, greater than $+2^{31}-1$ or less than -2^{31}. The overflow condition is detected as shown on AA671.

If overflow was previously on, it will not be changed. (Overflow can be reset by testing or by a load or store status instruction. See branch or skip on condition instruction.)

The carry indicator is set by a borrow from the high-order position.

3.27 MULTIPLY

Mnemonic	Op Code	Format	IX	IA
M	10100	0/1	Yes/No	No/Yes



- Multiply the 16-bit word at the EA by the 16-bit word in the accumulator.
- Place the 32-bit product into the accumulator and its extender.
- Core storage is not changed.
- Overflow and carry indicators are not changed.
- Reference Maintenance Diagram: AA673.
- The hexadecimal operation code for this instruction is A000, assuming no F, tag, or displacement bits.
- Refer to Table 1 for effective address.

The word at the core-storage location specified by the effective address (multiplicand) is multiplied algebraically by the word in the accumulator (multiplier). The 32-bit product replaces the contents of the accumulator (A) and its extension (Q). The most significant bits of the product are in the accumulator. The product is in the double precision format.

1130 Multiplication depends on the fact that any binary number may be represented by powers of two.

Thus it is not necessary to form the partial product by adding for each bit position. The machine may examine the multiplier (two lowest order bits at a time) to determine when to add the multiplicand, when to subtract, or when to just shift the multiplier.

This system of multiplication permits the 1130 to use fewer add cycles than would be possible with conventional multiplication.

Multiply E2 cycles are entered when an examination of the Q15 bit indicates that it is desired to add or subtract the multiplicand to/from the accumulator partial product.

Multiplier Q14	Q15	Previous Operation	New Action	Explanation
0	0	Add	Shift	No Action
0	1	Add	Add, Shift	Single One In String of Zeros
1	0	Add	Shift	No Action
1	1	Add	Sub, Shift	Start String of Ones
0	0	Sub	Add, Shift	End of String of Ones
0	1	Sub	Shift	No Action
1	0	Sub	Sub, Shift	Single Zero In String of Ones
1	1	Sub	Shift	No Action

In the first part of the E2 cycle the Q14 bit is examined to determine whether addition or subtraction is desired.

In the second part of the E2 cycle shifting is continued until the Q15 bit indicates that arithmetic action is again required.

The largest product that can be developed is 2^{30}. This occurs when the multiplier and multiplicand are both the largest negative numbers, -2^{15}.

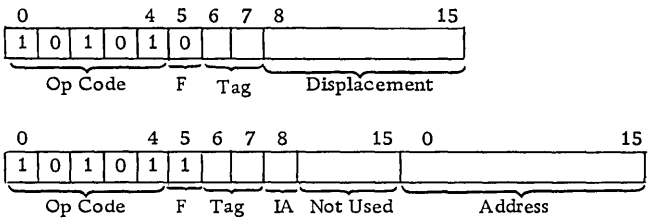
Indicators. Neither the overflow nor the carry indicators are changed.

Overflow and carry conditions cannot exist, because multiplying two 16-bit factors cannot produce

a result larger than can be represented in the accumulator and its extender.

3.28 DIVIDE

Mnemonic	Op Code	Format	IX	IA
D	10101	0/1	Yes/No	No/Yes



- Divide the 32-bit word in the accumulator and its extension by the 16-bit word at the location specified by the EA.
- Place the quotient in the accumulator and the remainder in the extension.
- Overflow indicator is turned on if a divide-by-zero is attempted or if the quotient cannot be accurately represented in the accumulator.
- The carry indicator is not changed.
- The hexadecimal operation code for this instruction is A800, assuming no F, tag, or displacement bits.
- The sign of the remainder is the same as the sign of the dividend.
- Refer to Table 1 for effective address.
- Reference Maintenance Diagram: AA674.

The word in the accumulator and its extension (a 32-bit double-precision dividend) is divided by the word at the core storage location specified by the effective address. The quotient replaces the contents of the accumulator and the remainder is placed in the extention (Q). The "sign" of the remainder will be the same as the dividend.

The division operation is similar to the multiplication operation in that a predetermined number (16) of shift, add, or subtract cycles are taken. Early in each of these sixteen reduction cycles, a comparison of the sign of the divisor and the sign of the accumulator determine whether the operation on this cycle is add or subtract. The same comparison determines whether or not to set a quotient bit. Next the A and Q are shifted left and the add/subtract operation performed. Two more cycles follow the 16 reduction cycles. During the 17th cycle both quotient and remainder are tested for the necessity of correction. If the remainder needs correction, it is also accomplished on cycle 17. During the 18th cycle, the accumulator (remainder) and accumulate extension (quotient) are exchanged. If quotient correction is required, it is accomplished on cycle 18, and the operation ends.

This shift-add/subtract procedure is illustrated as follows:
Divide binary 00111001 (57) by 0101 (5)

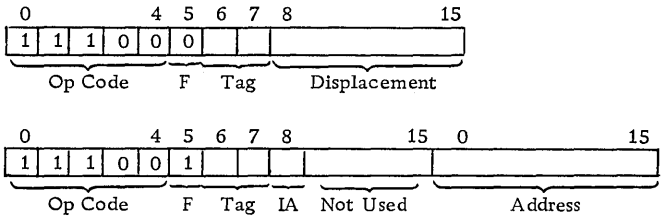
		Quotient
Shift and sub:	0 0 1 1 1 0 0 1	
	0 1 0 1	
	+ 0 0 0 1 0 0 0 1	1
Shift and sub:	0 1 0 1	
	- 1 1 1 1 1 1 0 1	0
Shift and add:	0 1 0 1	
	0 0 0 0 0 1 1 1	1
Shift and sub:	0 1 0 1	
	+ 0 0 0 0 0 0 1 0	1

Answer: Quotient 1011 (11) remainder 0010 (2).

Indicators. The overflow indicator is turned on if a divide-by-zero operation is attempted or if a quotient-overflow condition is detected. A quotient overflow occurs when the factors are such that the quotient would exceed the range of -2^{15} to $+2^{15}-1$. An overflow condition leaves the accumulator and its extension in an undefined state.

3.29 LOGICAL AND

Mnemonic	Op Code	Format	IX	IA
AND	11100	0/1	Yes/No	No/Yes



- AND the word specified by the effective address with the word in the accumulator.
- Positions in the accumulator that have a matching bit in the core storage word are left on; all other accumulator positions are turned off.
- Core storage is not changed.
- Carry and overflow indicators are not changed by this operation.
- Hexadecimal operation code for this instruction is E000, assuming no F, tag, or displacement bits.
- Reference Maintenance Diagram: AA691.
- Refer to Table 1 for effective address.

The word at the core storage location specified by the effective address is ANDed bit by bit with the word in the Accumulator:

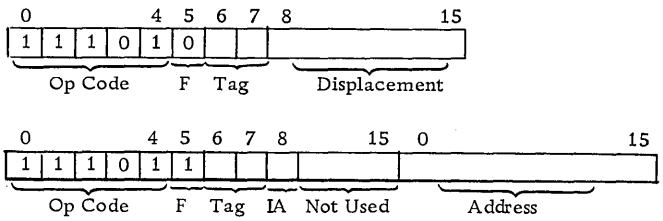
AND

Memory	1	1	0	0
Accum.	1	0	1	0
Result	1	0	0	0

The result replaces the word in the accumulator.

3.30 LOGICAL OR

Mnemonic	Op Code	Format	IX	IA
OR	11101	0/1	Yes/No	Yes/No



- OR the word specified by the effective address with the word in the accumulator.
- The bits in the accumulator remain on; bits in the core storage word turn on associated bits in the accumulator; all other positions remain off.
- Core storage is not changed.
- Carry and overflow indicators are not affected by this operation.
- Hexadecimal operation code for this instruction is E800, assuming no F, tag, or displacement bits.
- Reference Maintenance Diagram: AA691.
- Refer to Table 1 for effective address.

The word at the core storage location specified by the effective address is ORed bit by bit with the word in the accumulator.

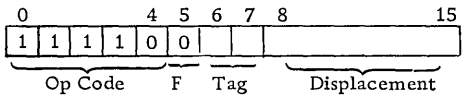
OR

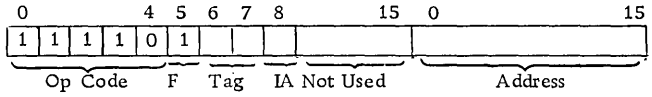
Memory	1	1	0	0
Accum.	1	0	1	0
Result	1	1	1	0

The result replaces the word in the accumulator.

3.31 LOGICAL EXCLUSIVE OR

Mnemonic	Op Code	Format	IX	IA
EOR	11110	0/1	Yes/No	No/Yes





- Exclusive OR the word specified by the effective address with the word in the accumulator.
- The bits in the accumulator without matching bits in the core storage word, remain on; bits in the core storage word without matching bits in the accumulator word, turn on accumulator bits; all other accumulator positions remain off.
- Core storage remains unchanged.
- Carry and overflow indicators are not changed by this operation.
- Hexadecimal operation code for this instruction is F000, assuming no F, tag, or displacement bits.
- Reference Maintenance Diagram: AA691.
- Refer to Table 1 for effective address.

The word at the core storage location specified by the effective address is exclusive ORed bit by bit with the word in the Accumulator.

Exclusive OR

Memory	1	1	0	0
Accum.	1	0	1	0
Result	0	1	1	0

The result replaces the contents of the accumulator.

INPUT/OUTPUT OPERATIONS

- Two methods are used to control input/output devices and to transfer data between core storage and the devices.
  1. Direct program control - initiated by XIO instructions. Each data transfer or physical action of the I/O device is initiated by an individual XIO instruction.
  2. Combined control - initiated by XIO instructions. Data transfers and physical actions of the I/O device are controlled by a selected combination of cycle steal operations and XIO instructions.

- Both methods are initiated by the execute I/O instruction.

- XIO instruction addresses a two-word I/O control command (IOCC) which selects the I/O device, specifies the function to be performed, and provides the data address.

- Effective address of the IOCC must be an even address.

- Hexadecimal operation code for this instruction is 0800 assuming no F, tag, or displacement bits.

- Refer to Table 1 for effective address.

- F = 1, tag ≠ 00, IA = 1: EA = address at location specified by the contents of address + the contents of XR.

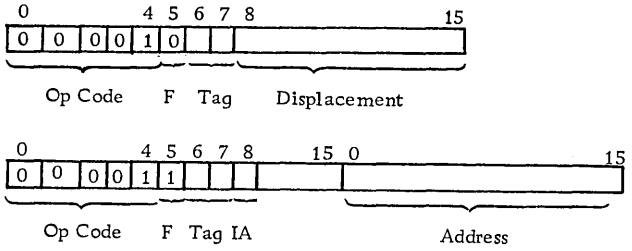
- Reference Maintenance Diagrams: AA621, AA611, AA231.

The basic Differences between a direct program control operation and a combined control operation are shown in the following chart:

Cycle	Direct Program Control	Combined Control
E-1 Cycle	Transfer Control Word to I/O Adapter	Same
E-2 Cycle	Data Address to M Register	Data Address to Cycle Steal Address Register. Terminate Op.
E-3 Cycle	Transfer data word to or from I/O Adapter using M register address. Terminate Op. Interrupt to I/O subroutine for subsequent data word transfers.	None
Cycle Steal Cycle	None	Transfer data word to or from I/O Adapter using Cycle Steal address. Take C. S. cycles when needed for subsequent data word transfers.

3.32 XIO INSTRUCTION

Mnemonic	Op Code	Format	IX	IA
XIO	00001	0/1	Yes/no	Yes/No



- The 1130 uses only one I/O instruction: Execute I/O.

- Place the input/output control command word (IOCC) into the U register.

- Decode the U register to select the device and operation.

- Reference Maintenance Diagram: AA621.

E-1 Cycle. This cycle is referred to as the XIO control cycle. The effective address (EA), loaded into the accumulator during the I cycle, must be an even address. This is because the address, which is the location of the IOCC, is loaded into the M register and the M_{15} bit output is forced on. This causes EA + 1, the location of the IOCC control word, to be addressed. The IOCC control word, which includes the area, function and modifier bits, is placed in the U register.

E2, E3 Cycles. The U register output is decoded and the I/O attachment for the device specified by the area controls the operation to be done.

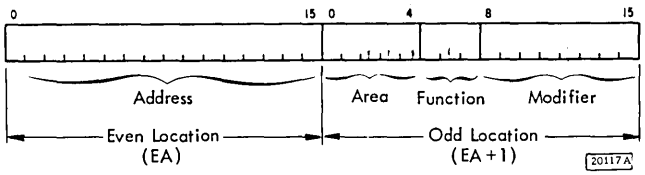
3.32.1 I/O Control Commands

The address portion of the execute I/O instruction specifies the core storage location of a two-word I/O control command and must be even.

Seven I/O control command functions are provided for the IBM 1130:

Read
Write
Initiate read
Initiate write
Control
Sense device
Sense interrupt

All I/O control commands have four parts:



3.32.1.1 Address

The meaning of this 16-bit field depends on the function of the I/O control command:

1. If the function is initiate write (101) or initiate read (110), the address specifies the starting address of a table in storage (an I/O block). This table contains data words and control information. Initiate write and initiate read functions are used only with disk storage control.
2. If the function is control (100), and the area specifies the disk storage device, the address indicates the number of tracks the access must be moved.
3. If the function is sense device (111) or sense interrupt (011), the address field is ignored. Instead, an increment of time, equivalent to a core storage cycle, is taken, during which the selected I/O device or interrupt level places its status code into the accumulator.
4. If the function is write (001) or read (010), the address specifies the core storage location of the data word. These functions are used only with direct program control.

3.32.1.2 Area

This 5-bit field specifies a single device (1442 Card read-punch, 1132 Printer, etc.).

00010(2) 1442 Card Read-Punch
00110(6) 1132 Printer
00100(4) Disk Storage
00101(5) 1627 Plotter
00011(3) 1134 Paper Tape Reader, 1055 Paper Tape Punch
00001(1) Console Keyboard, Console Printer
00111(7) Console Entry Switches

3.32.1.3 Function

The seven primary I/O functions are specified by the 3-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 core storage location is provided by the address field of the I/O control command.
010--Read
This code is used to transfer a single word from an I/O unit to core storage. The address of the core storage location is provided by the address field of the I/O control command.
011--Sense Interrupt
This code directs the I/O devices requesting interrupt recognition on the interrupt level specified by the modifier field of the I/O control command to make their interrupt status available.
100--Control
This code causes the selected device to interpret the modifier or address field as a specific control action.
101--Initiate write
This code initiates a write operation on the disk storage unit which will subsequently make data transfers from core storage under disk storage control.
110--Initiate read
This code initiates a read operation from the disk storage unit which will subsequently make data transfers to core storage under disk storage control.
111--Sense device
This code directs the selected device to make its current indicator status available for automatic placement into the accumulator.

Programming Note:

The current contents of the accumulator are destroyed by the execution of an execute I/O instruction.

Therefore, it is the programmers responsibility to save the accumulator contents, if necessary.

3.32.1.4 Modifier

This 8-bit field provides additional definition for either function or area. For example, if the area specifies an 1132 Printer, and the function specifies control (100), a particular modifier bit specifies the operation. In this case, the modifier extends the function.

3.32.2 Input/Output Termination

Input/output operations using devices under direct program control are terminated when the device has completed the single function requested by the program.

Disk storage operations are terminated when the number of data words specified have been transferred.

3.32.3 Input/Output Interrupts

Input/output interrupts are caused by termination of an I/O operation or by operator intervention at the I/O device. Input/output interrupts enable the CPU to provide appropriate programmed responses to conditions that occur in I/O devices.

Input/output interrupts are assigned priority levels to allow the most efficient use of all I/O devices.

Interrupting Device	Interrupt Level
1442	0, 4
1132	1
Disk Storage	2
Plotter	3
Keyboard-Printer, 1134/1055	4
Console	5

Conditions responsible for I/O interrupt requests are preserved in the device status word of the I/O devices until they are accepted by the CPU.

3.32.4 Interrupt Level Status Word

Although a 16-bit ILSW could exist for each priority level, only level 4 is used with the 1130 System. Each device, with an interrupt request signal assigned to priority level 4, is given a particular bit position in its ILSW to indicate its interrupt request status; one (1) bit if on, and zero (0) bit 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 ILSW must be checked by the program to determine the device causing the interrupt.

The identification of the interrupting condition within the device is accomplished by sensing the device status word (DSW).

3.32.5 Device Status Word

The 16-bit device status word consists of bit indicators that define the status of the I/O device and its control. The indicators vary from device to device. However, each indicator falls into one of two groups: Interrupt indicators or non-interrupt indicators.

As the name implies, interrupt indicators are associated with conditions that interrupt the stored program execution. For example, in the 1442 operation, reader service response and punch service response are interrupt indicators. Non-interrupt indicators are associated with conditions that do not interrupt the program; for example, busy, and not ready.

3.33 I/O ATTACHMENTS

- These I/O units can be attached to the 1130 system:
  - Console Printer-Keyboard (Basic-Models 1 and 2)
  - 1134 Paper Tape Reader
  - 1055 Paper Tape Punch
  - 1442 Card Read Punch
  - 1627 Plotter
  - 1132 Printer

  } Features)

  - Disk Storage (Basic-Model 2)
- The console printer-keyboard is standard on the models 1 and 2.
- The disk storage is standard on the model 2.
- The circuits that facilitate attachment of I/O units to the 1131 comprise the I/O unit adapters.
- I/O unit adapters provide buffer, storage control, and I/O interface circuitry.
- The I/O interface enables the CPU to control all I/O operations with only one type of instruction.

- IBM 1130 system I/O units operate in either direct program control, or by combined (disk storage or 1132 printer) control.

3.33.1 I/O Interface

So that the CPU may control a wide variety of I/O devices, all control features are designed to respond to a standard set of signals. This I/O unit to CPU connection is called the I/O interface. It enables the CPU to handle all I/O operations with only one type of instruction. I/O interface is described in Chapter 2 - Functional Units, 2.6 I/O Control.

3.34 DIRECT PROGRAM CONTROLLED OPERATION

- Initiated by an execute I/O (XIO) instruction with an I/O control command (IOCC) function of control, sense interrupt level, sense device, read or write.

- Each XIO instruction performs the following functions:
  1. Addresses the IOCC.
  2. Places the control word (area, function, modifier) in the U register.
  3. Transfers the first data word to or from the core storage location specified by the address portion of the IOCC.

- Subsequent data words may be transferred by an I/O program subroutine reached by an interrupt initiated by the device.

- The data flow of the direct program controlled operation is shown in the I/O operation diagrams for each device.

- Reference Maintenance Diagrams: AA231, AA611, AA621.

- Devices operating in this mode include:

  Console printer-keyboard
  1627 Plotter
  1134 Paper Tape Reader
  1055 Paper Tape Punch
  1442 Card Read-Punch
  1132 Printer (also combined control)

3.34.1 Read/Write Function

When the function specified is a read or write, the address word provides the storage address from or to which data is transferred. At the conclusion of the storage cycle for data transfer, the XIO instruction is terminated and the next sequential instruction is executed.

I Cycle. Loads the accumulator with the effective address (see I cycle description). An XIO effective address (EA) must be an even address.

E-1 Cycle. Transfers EA to the M register and makes the M_{15} bit output line active. This causes the core storage location at EA+1 (odd address) to be selected. The word at this location is loaded into the B register. This word (EA+1) is the control word of the I/O control command; it contains the area, function, and modifier bits. It is placed in the U register to be analyzed by the I/O Adapters.

E-2 Cycle. Loads the accumulator with the address word portion of the IOCC (located at EA, even address).

E-3 Cycle. Transfers the address word from the accumulator (loaded during E-2 Cycle) to the M register. Using this M register address, the data word is transferred to or from the I/O adapter via the I/O out or I/O in bus.

Normally, transmission of several words is required to complete the data or message transfer. This is accomplished by allowing the CPU to respond to an interrupt request from the device when the device cycle has been completed and a subsequent data word is required. At an appropriate time, the CPU acknowledges the interrupt, identifies the request to the particular device, and by another XIO instruction transmits the next sequential word of the message as specified by the IOCC.

It is the responsibility of the program subroutine answering the device interrupt request to modify the address word of the IOCC, provide table look-up to translate to the device character set if required, and maintain a program word count to indicate the end of message if necessary. At the completion of each interrupt subroutine, it is also required to exit from the routine utilizing a branch or skip on condition instruction (BSC) with B9 = 1 which accomplishes the branch out of interrupt function. (Refer to branch or skip on condition instruction description.) This is necessary to restore the interrupt hardware so that future interrupt requests at the same or lower interrupt levels can be acknowledged.

3.34.2 Control Function

With a specified function of control, the modifier field specifies to the device, the particular control

operation to be executed. Examples of such control operations are start printer, feed cycle, initiate read.

I Cycle. Same as read/write function.

E-1 Cycle. Same as read/write. Set up area, function, and modifier decode circuits to I/O adapter. Some I/O adapters use this cycle to perform a particular control function.

E-2 Cycle. Certain I/O adapters interpret the address word at EA to determine the particular function and initiate that function during this cycle. For other I/O adapters, this may be a dummy cycle.

E-3 Cycle. Not used with a control function.

3.34.3 Sense Function

The sense functions of the XIO instruction are used to read the interrupt level status word (ILSW) and/or the device status word (DSW).

I-Cycle. Same as other functions.

E-1 Cycle. Same as other functions except the XIO inhibit memory FF is turned on because the data to be transferred from the I/O adapter is not to be stored in core storage.

E-2 Cycle. The status word, placed on the I/O in bus by the I/O adapter is loaded into the B register, then transferred to the accumulator.

E-3 Cycle. Not used with the sense function.

3.34.4 Busy Condition

It is possible for the program sequence to issue an XIO instruction to a device that is busy responding to a previous XIO. Each device that can have this condition provides a busy indicator in the DSW. This indicator signals that the device cannot accept data or control information, and that should data or control information be sent it is lost. It is up to the program to ensure, by testing the busy indicator, that the data is not lost. No hardware indication is given to signal incorrect use of the device.

3.35 CONSOLE PRINTER-KEYBOARD

- The 1131 provides a console-keyboard and a console-printer for the operator to enter information into the system and receive information from the system.

- For a detailed description of the printer mechanics, see I/O Printer (Modified Selectric Field Engineering Manual of Instruction. (See Bibliography.)

3.36 CONSOLE PRINTER

3.36.1 Printer Functional Description

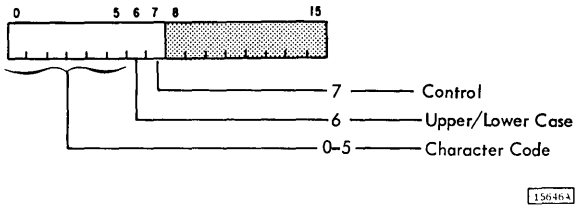
- Maximum output rate of the printer is 15.5 characters per second.

- Data to be printed is transferred from core storage to the typewriter by direct program control.

- Data and control characters (space, tabulate, etc.) are sent to the typewriter by means of the write command.

Because control characters and data characters are sent in the same manner, the core storage field 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:



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

3.36.1.1 Data Coding

Data printed by the console-printer is coded by the program into the typewriter code. Figure 3-5 shows the characters which can be printed by the standard print element.

The data-character codes also contain (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.

Character Code Bits						U/L Case		Ctrl
B0 T_1	B1 T_2	B2 R_1	B3 R_{2A}	B4 R_2	B5 R_5	B6=0 LC	B6=1 UC	B7
0	0	1	1	1	1	A	A	0
0	0	0	1	1	0	B	B	0
0	0	0	1	1	1	C	C	0
0	0	1	1	0	0	D	D	0
0	0	1	1	0	1	E	E	0
0	0	0	1	0	0	F	F	0
0	0	0	1	0	1	G	G	0
0	0	1	0	0	1	H	H	0
0	0	1	0	0	0	I	I	0
0	1	1	1	1	1	J	J	0
0	1	0	1	1	0	K	K	0
0	1	0	1	1	1	L	L	0
0	1	1	1	0	0	M	M	0
0	1	1	1	0	1	N	N	0
0	1	0	1	0	0	O	O	0
0	1	0	1	0	1	P	P	0
0	1	1	0	0	1	Q	Q	0
0	1	1	0	0	0	R	R	0
1	0	0	1	1	0	S	S	0
1	0	0	1	1	1	T	T	0
1	0	1	1	0	0	U	U	0
1	0	1	1	0	1	V	V	0
1	0	0	1	0	0	W	W	0
1	0	0	1	0	1	X	X	0
1	0	1	0	0	1	Y	Y	0
1	0	1	0	0	0	Z	Z	0
1	1	1	1	1	1	1	(	0
1	1	0	1	1	0	2	+	0
1	1	0	1	1	1	3	<	0
1	1	1	1	0	0	4	¬	0
1	1	1	1	0	1	5	)	0
1	1	0	1	0	0	6	;	0
1	1	0	1	0	1	7	*	0
1	1	1	0	0	1	8	'	0
1	1	1	0	0	0	9	"	0
1	1	0	0	0	1	0	|	0
1	1	0	0	0	0	#	=	0
1	0	1	1	1	1	/	–	0
1	0	0	0	0	1	-	?	0
1	0	0	0	0	0	,	:	0
0	1	0	0	0	1	&	>	0
0	1	0	0	0	0	$	!	0
0	0	0	0	0	1	@	%	0
0	0	0	0	0	0	.	¢	0

22265

Figure 3-5. Data Character Coding, Console Printer

A console-printer write command is modified by the B7 position of the output character word. If B7 equals one, the write command to the printer is interpreted as a control function. If B7 equals zero, the write command is interpreted as a print function.

The codes for console-printer control functions are shown in Figure 3-6.

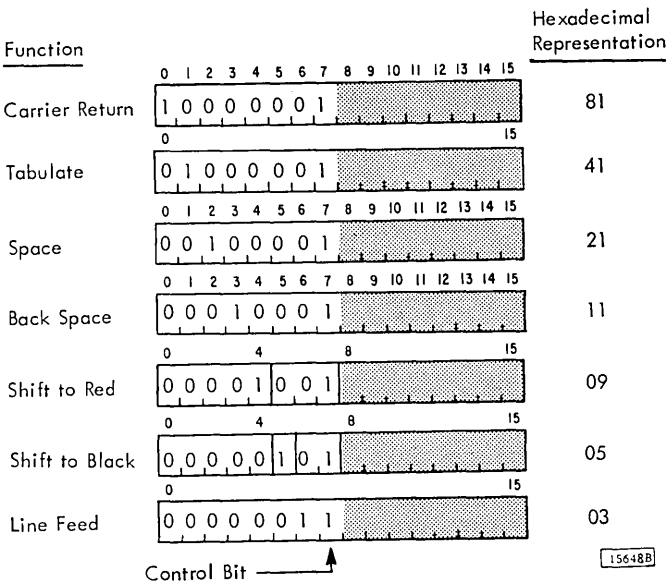


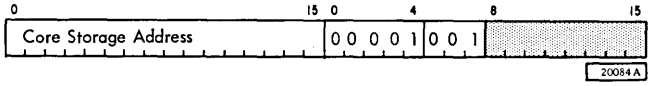
Figure 3-6. Console Printer Control Character

3.36.2 Console Printer Programming

- The console printer operates on the IBM 1130 under direct program control.
- An XIO instruction addresses an I/O control command which selects the printer, specifies a write or sense device function, and provides the data address.

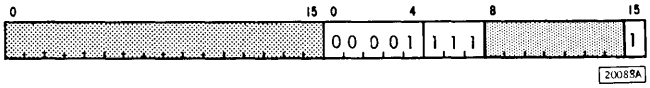
3.36.3 I/O Control Commands (IOCC)

3.36.3.1 Write (001).



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

3.36.3.2 Sense Device (111).



This command causes the device status word of the console-printer (Figure 3-7) to be placed in the accumulator.

Modifier bit 15 specifies that the responses are to be reset.

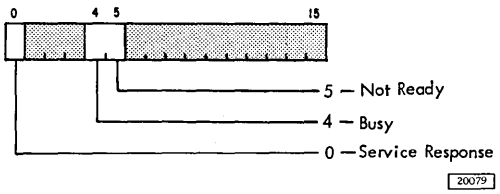


Figure 3-7. Console Printer Device Status Word

3.36.3.3 Interrupt

There is only one interrupt associated with the console-printer attachment, level 4.

Service Response. This interrupt occurs each time the console-printer has completed printing the data and/or the control operations required by the last word transmitted by the write command.

3.36.3.4 Indicators

The following indicators are associated with the console printer.

Not Ready. When off, indicates that the console 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.

If not ready is tested and found to be on, busy should then be tested. If busy is off, operator intervention is required. However, if not ready is on and busy is on, it indicates that the console-printer has not finished execution of the previous write command to the console printer.

Busy. When on, 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 console printer until the printer has completed the action required.

3.36.4 Console Printer Operation

- The execute I/O instruction places the control word of the IOCC on the I/O out bus.
- The area (Area 1), function and modifier bits are decoded by the console printer adapter.
- The write function can initiate either a print or control operation on the console printer.
- If the specified function is sense device, the indicator status bits are placed on the I/O in bus.
- One of the following functions is performed:
  1. Control
  2. Shift and print
  3. Print without a shift
  4. Sense device
- Reference Maintenance Diagrams: AA101, AA231, AA621, XW401, XW501, XW511, XW701.

3.36.4.1 Printer Write Operation

E-1 Cycle. This cycle is referred to as the XIO control cycle. The effective address (EA), loaded into the accumulator during the I cycle, must be an even address. This address, which is the location of the IOCC, is loaded into the M register and the M_{15} bit output line is made active. This causes EA+1, the location of the control word, to be addressed. The control word, which includes the area, function and modifier bits, is placed in the U register.

E-2 Cycle. The M_{15} bit line is no longer active so that EA, which is the location of the address word, is addressed. The address word is therefore read from core storage and loaded into the accumulator for use in the E-3 cycle.

E-3 Cycle. The address word of the IOCC, loaded into the accumulator during the E-2 cycle, addresses the data word which is placed on the I/O out bus and set into the console printer adapter. The XIO instruction execution is terminated and the program continues with the next instruction.

3.36.4.2 Print or Control Execution

1. Print or control - no shift.
   a. The fall of T6 in the E-3 cycle turns on the typewriter cycle FF, gates the I/O bus to the typewriter buffer, times single shot 1, and turns on typewriter interlock.
   b. While single shot 1 is timing, the typewriter buffer is gated to the tilt and/or rotate magnets, or to the control magnets (bit 7 = 1).
   c. The single shot 1 timing out turns on single shot 2, brings up the print interlock line, and resets the typewriter buffer.

d. The CB response circuit breaker provides approximately 35 milliseconds for a print function. On a control function, the function interlock is up until the control function is completed. The opening of the CB response circuit breaker, the dropping of function interlock, or the timing out of single shot 2 resets the typewriter cycle FF and drops the typewriter interlock line.
e. When the typewriter cycle FF goes off the response FF turns on and the level 4 interrupt is activated.

2. Print Function - Shift.
   a. The fall of T6 in the E-3 cycle turns on the typewriter cycle FF, gates the I/O bus to the typewriter buffer, times single shot 1, and brings up the typewriter interlock line. Buffer bit 6 turns on the shift status FF and prevents the gating of the buffer to the tilt and rotate magnets.
   b. The CB response circuit breaker makes because of the shift cycle.
   c. Single shot 1 timing out turns on single shot 2. The typewriter buffer is not reset because of shift status and/or because CB response is up.
   d. Typewriter interlock is dropped by the break of CB response.
   e. When typewriter interlock goes off, it starts single shot 1 timing which brings up typewriter interlock. Single shot 1 timing resets the shift status FF.
   f. With shift status off, the print buffer is gated to the tilt and rotate magnets.

The operation continues as in d. and e. in item 1.

The control buffer F-F's control the magnet drivers for the seven control functions as follows:

Data Word Bit	Function
0	Carrier return
1	Tabulate
2	Space
3	Backspace
4	Shift to red
5	Shift to black
6	Line feed

The function is performed and the buffers are reset by single-shot circuits. At the same time that the print buffer is reset, the service response FF is turned on to activate an interrupt request.

3.36.4.3 Printer Sense Operation

The sense device function places the printer device status word (previously described) on the I/O in bus and loads it into the accumulator for subsequent analysis by the program.

E-1 Cycle. This control cycle is the same as for the write function except the XIO inhibit memory FF is turned on as the data to be transferred from the printer is not to be stored in core storage.

E-2 Cycle. A one is placed on the I/O in bus in the bit positions associated with the status indicators that are on. If specified by the IOCC, the indicators are reset.

The DSW is set into the B register, loaded into the accumulator and the operation is terminated.

3.37 CONSOLE KEYBOARD

3.37.1 Console Keyboard Functional Description

- The keyboard is similar to the keyboard of the IBM 029.
- The keys are not connected to the printer. Therefore, keyboard entries are not printed unless the CPU is programmed to do so.
- Maximum input speed of the keyboard is 20 characters per second, but is usually limited by the speed of the operator.
- The keyboard operates under direct program control.
- The keyboard emits an IBM-Card-Coded character for each operation of a key.

The character coding emitted by the keyboard is shown in Figure 3-8. The character enters the CPU left justified. Thus bit 0 corresponds to 12 row, bit 1 to 11 row, etc. Bits 12, 13, and 14 have special significance and indicate end of message, erase field, and backspace. (See description of these three keys).

3.37.1.1 Keyboard Function Keys

Interrupt Request. This key causes an interrupt request on level 4.

Key	IBM Card Code	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
*	11,8,4		1					1				1					
/	0,1			1	1												
Ø	0			1													
1	1				1												
2	2					1											
3	3						1										
4	4							1									
5	5								1								
6	6									1							
7	7										1						
8	8											1					
9	9												1				
$	11,8,3		1				1					1					
.	12,8,3	1					1					1					
	0,8,3			1			1					1					
EOF	None													1			
Back Space	None														1		
ER FLD	None															1	
=	6,8									1		1					
'	5,8								1			1					
(	12,5,8	1						1				1					
)	12,5,8		1						1			1					
+	12,8,6	1								1		1					
-	11		1														
A	12,1	1			1												
B	12,2	1				1											
C	12,3	1					1										
D	12,4	1						1									
E	12,5	1							1								
F	12,6	1								1							
G	12,7	1									1						
H	12,8	1										1					
I	12,9	1											1				
J	11,1		1		1												
K	11,2		1			1											
L	11,3		1				1										
M	11,4		1					1									
N	11,5		1						1								
O	11,6		1							1							
P	11,7		1								1						
Q	11,8		1									1					
R	11,9		1										1				
S	0,2			1		1											
T	0,3			1			1										
U	0,4			1				1									
V	0,5			1					1								
W	0,6			1						1							
X	0,7			1							1						
Y	0,8			1								1					
Z	0,9			1									1				
Space	Blank	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
¢	12,8,2	1				1						1					
<	12,8,4	1						1				1					
|	12,8,7	1									1	1					
&	12	1															
!	11,8,2		1			1						1					
;	11,8,6		1							1		1					
¬	11,8,7		1								1	1					
%	0,8,4			1				1				1					
_	0,8,5			1					1			1					
>	0,8,6			1						1		1					
?	0,8,7			1							1	1					
:	8,2					1						1					
#	8,3						1					1					
@	8,4							1				1					
"	8,7										1	1					

24062B

Figure 3-8. Keyboard Character Code

End of Field. When the CPU reads in response to this key, a word containing a 12 bit only is placed in core storage. Analysis of this word allows the program to determine that no further characters are to be sent in this message.

Backspace. When the CPU reads in response to this key, a word containing a 13 bit only is placed in memory. Analysis of this word allows the program to determine that the last character received is to be replaced by the next character to be entered.

Erase Field. When the CPU reads in response to this key, a word containing a 14 bit only is placed in memory. Analysis of this word allows the program to determine that the message being entered is to be deleted and replaced by a corrected message.

Mode. There are two mode keys: numeric (upper case shift) and alphabetic (lower case shift). These keys place the keyboard in the indicated mode. The keyboard remains in the selected mode until changed. If the numbers or symbols which appear on the top portion of the keys are desired, the keyboard must be placed in numeric mode.

Reset Keyboard. This key allows the operator to restore the keys if they should become locked.

3.37.1.2 Keyboard Light

Keyboard Select. This light comes on when the CPU has performed a control command. This light goes off when a read command is performed.

3.37.2 Keyboard Operating Procedures

The following procedure describes a typical use of the keyboard.

1. The operator presses the keyboard request key.
2. The CPU honors the request interrupt.
3. The CPU places the keyboard in the select status and the select light is turned on to signal the operator that a character can be entered.
4. A character key is pressed; the keyboard is removed from the select status.
5. The CPU performs a read command which enters the character into core storage.

6. Before another character can be entered the program must return the keyboard to the select status.

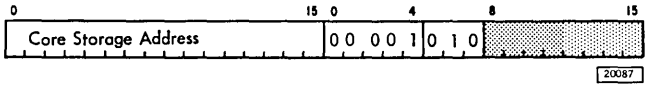
   If the CPU performs a read command when the keyboard is not in the select status, no bits are entered.

3.37.3 Console Keyboard Programming

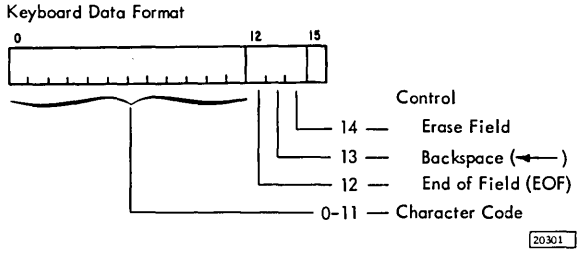
- The keyboard operates under direct program control of the IBM 1131.
- Selected with an area decode of 1.

3.37.4 I/O Control Commands (IOCC)

3.37.4.1 Read (010).



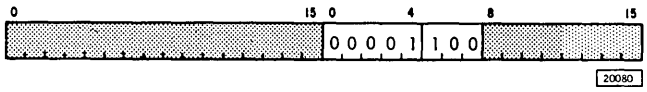
This command enters a single input character from the keyboard in the storage location specified by the address of the IOCC.



3.37.4.2 Sense Device (111).

This command reads the device status word (Figure 3-9) associated with the keyboard into the CPU. Modifier bit 15 specifies that the response is to be reset.

3.37.4.3 Control (100).



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

3.37.4.4 Interrupts

The two interrupts associated with the keyboard are assigned to the same level of priority. Level 4.

Request. This interrupt is initiated by the interrupt request key located on the keyboard.

Keyboard Response. This interrupt signals that a character key has been pressed and that a character is ready to be entered into core storage.

3.37.5 Keyboard Read Operation

- The interrupt request key initiates an interrupt request. When the interrupt is serviced, an XIO

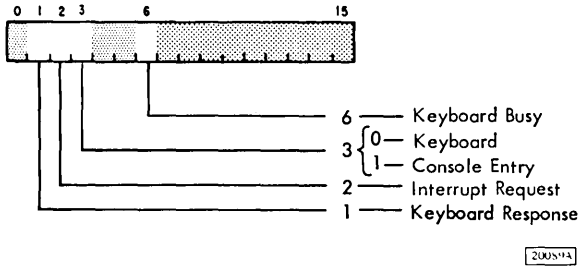


Figure 3-9. Device Status Word (DSW)

instruction with an IOCC (Area 1) specifying a control function is executed.

- The control operation selects the keyboard, placing it in select status, and turns on the select lamp.
- Operation of a character key places the card code of that character into the B register and initiates another interrupt request.
- When this interrupt is serviced, an XIO instruction with an IOCC specifying a read function is executed (Area 1).
- The read operation transfers the data word (card-code character) to core storage.
- Reference Maintenance Diagrams: AA101, AA231, XK401, XK501, AA621.

3.37.5.1 Interrupt Request

Operating the interrupt request key starts a 25 millisecond delay to allow time to restore the keyboard and energizes the restore magnet. After the delay, the adapter circuits initiate an interrupt request to the CPU.

3.37.5.2 Control Function

When the CPU satisfies all conditions to service the interrupt, a branch is forced to a subroutine containing an XIO instruction with an IOCC specifying the keyboard and a control function. The control operation turns on the keyboard select FF and lights the select light.

3.37.5.3 Character Key Function

When the keyboard is in the select condition, pressing a character key loads the card-code bits for that character onto the I/O input bus and initiates an interrupt request for the read operation by firing 28 ms single shot 1 and turning on the keyboard response FF.

3.37.5.4 Read Function

When this interrupt is serviced, a branch is forced to a subroutine containing an XIO instruction with an IOCC specifying a read function (Area 1). The effective address (EA) of the XIO instruction must be an even address.

E-1 (Control) Cycle. The M register bit 15 output line is made active to address EA + 1 (odd) which is the location of the control word of the IOCC. The control word is set into the B register and placed in the U register.

E-2 Cycle. The decoded output of the U Register (XIO Read) specifies the read function (code 010). The CPU drops the M_{15} bit output line and reads the IOCC address word (at EA) and loads it into the accumulator.

E-3 (Data Cycle). The decoded bits of the keyboard contacts are placed on the I/O in bus. The CPU stores this data word in core storage at the location specified by the address portion of the IOCC. The keyboard select FF is turned off and the 25 ms single shot 2 is timed to restore the keyboard. If additional characters are to be entered, the program must execute another XIO Control command to return the keyboard to the select condition. After the last character is entered, the end of message key is operated to indicate to the program that the subroutine need not be repeated.

3.37.6 Keyboard Sense Operation

- The sense device function places the keyboard device status word (previously described) on the I/O in bus and loads it into the accumulator for subsequent analysis by the program.

E-1 Cycle. This control cycle is the same as for the read function except the XIO inhibit memory FF is turned on as the data to be transferred from the keyboard is not to be stored in core storage.

E-2 Cycle. A one is placed on the I/O in bus in the bit positions associated with the status indicators that are on. If specified by bit 15 in the IOCC, the indicators are turned off.

The DSW is set into the B register, loaded into the accumulator, and the operation is terminated.

3.38 CONSOLE BIT SWITCHES

3.38.1 Bit Switch Functional Description

- Used to enter data into core storage in binary form.
- The bit switches operate under direct program control.

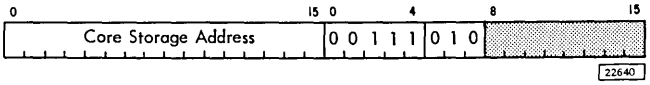
- The bit switches are also used as a source of data in load mode operations, load IAR, load core storage.

3.38.2 Programming

- The Bit switches operate under direct program control of the IBM 1131.
- The bit switches are selected with an area decode of 7.

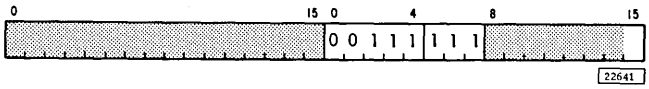
3.38.3 I/O Control Commands (IOCC)

3.38.3.1 Read (010)



This command enters a single input character from the bit switches into the storage location specified by the address of the IOCC.

3.38.3.2 Sense Device (111)



The bit switches use the same device status word as the console keyboard (Figure 3-9).

3.38.3.3 Interrupts

The interrupt associated with the bit switches is the interrupt request key, level 4.

3.38.4 Bit Switch Read Operation

- Interrupt request key initiates an interrupt request.
- When the interrupt is serviced, the program senses the DSW and finds the console/keyboard switch in the console position.
- The program gives a XIO instruction with an IOCC specifying a read function (Area 7).
- The read operation transfers the data word to core storage.

Reference Maintenance Diagram: XK501, ALD pages: ZS 101, XC 101, XC 131.

3.38.4.1 Read Function

When this interrupt is serviced, a branch is forced to a subroutine containing an XIO instruction with an IOCC specifying a read function (Area 7). The effective address (EA) of the XIO instruction must be an even address.

E-1 (Control) Cycle. The M register bit 15 output line is made active to address EA + 1 (odd) which is the location of the control word of the IOCC. The control word is set into the B register and placed in the U register.

E-2 Cycle. The decoded output of the U register (XIO read) specifies the read function (code 010). The CPU drops the M_{15} bit output line, reads the IOCC address word (at EA) and loads it into the accumulator.

E-3 (Data Cycle). The data bits of the bit switches are placed on the I/O in bus. The CPU stores this data word in core storage at the location specified by the address portion of the IOCC.

3.39 COMBINED CONTROL

- Initiated by execute I/O instruction.
- Used by disk storage attachment and 1132 Printer attachment.
- Uses cycle stealing to transfer data to core storage.
- On disk storage operations only one XIO instruction is used for each operation
- On 1132 Printer operations, XIO instructions must be given for each character but cycle steal cycles transfer the program generated printer scan field.

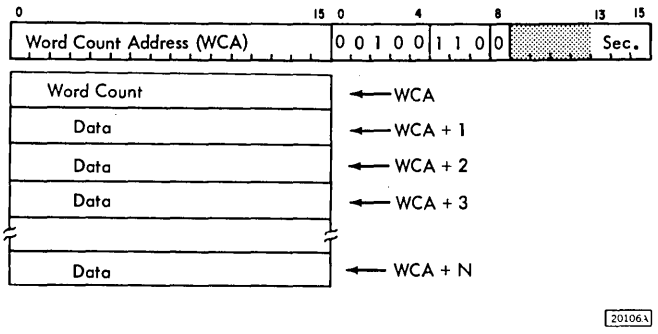
Disk storage operations are described in the next section. 1132 operations are described in 1130 Features FETO. (See Bibliography.)

3.40 DISK STORAGE OPERATIONS

- Initiated by an execute I/O (XIO) instruction with an I/O control command (IOCC) function of control, initiate read, initiate write, or sense device.
- The XIO instruction addresses the IOCC.
- Places the control word (area, function, modifier) in the U register.
- Sets the controls in the disk storage attachment.
- Word count and data words are transferred to or from the disk storage by cycle steal cycles controlled by the attachment.
- Data flow of the disk storage attachment is shown in Maintenance Diagrams: AA101, AA231, AA621, XF401, XF501, XF511, XF521, XF701, XF711, XF721.

3.40.1 I/O Control Commands (IOCC)

3.40.1.1 Initiate Read (110)



This instruction causes the number of words specified by the word count to be read from the disk storage sector (0-7) as identified by modifier bits 13-15. The address word of the instruction contains the WCA (word count address), and modifier bit 8 determines whether the command is a read instruction (0) or a read-check instruction (1).

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

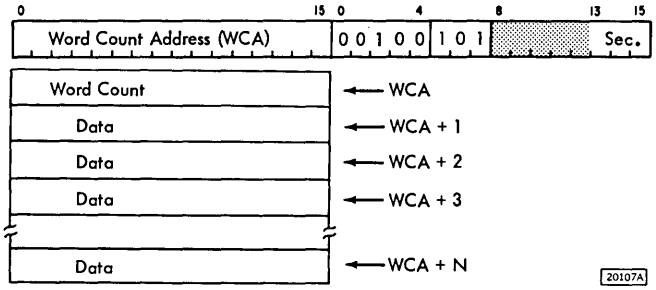
An operation-complete interrupt level 2 occurs when the number of words in the word count has been transmitted.

Read Instruction (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, which is 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 incoming disk storage data so that useful data is not written over and lost.

Read-Check Instruction (Bit 8 = 1). Data is read from disk storage, as in the read instruction, and the number of bits of each word is checked for modulo 4. If the modulo 4 check for any word is not zero, the data error indicator bit is set in the disk storage DSW. Neither disk storage nor core storage is affected by the read-check instruction.

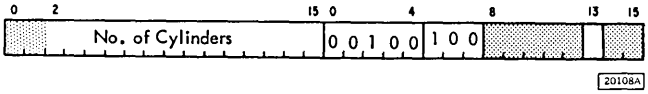
3.40.1.2 Initiate Write (101)



This instruction 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 address word of the instruction contains the address of the word count (WCA). The data is transmitted from core storage location WCA + 1 and ascending addresses. A full sector, 321 words, is the maximum transmission with one instruction. Succeeding sectors, or parts of sectors, require an initiate write instruction for each one.

An operation-complete interrupt, level 2, occurs when the number of words in the word count has been transmitted.

3.40.1.3 Control (100)



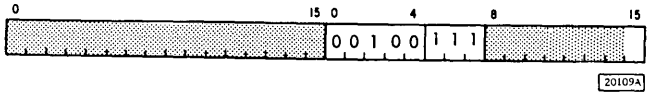
This instruction causes the access mechanism to move in increments of one or two cylinders for the number of cylinders specified by the address word of the instruction. If the number of cylinders is odd, the first increment consists of one cylinder and the remainder will be increments of two cylinders.

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.

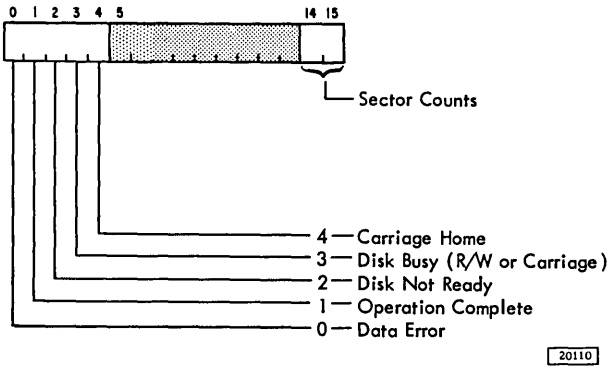
When the access mechanism has moved the number of cylinders specified, an operation-complete, level 2, interrupt occurs.

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.

3.40.1.4 Sense Device (111)



This instruction causes the device status word of disk storage to be read into the accumulator. All indicators associated with the specified interrupt level are reset if modifier bit 15 is a 1.



3.40.1.5 Interrupt - Level 2

Operation Complete. This is the only interrupt associated with disk storage, and is turned on at the end of a read, read-check, or write operation. It is also turned on when the access mechanism has reached the designated cylinder during a control instruction.

Indicators

Operation Complete. This indicator is turned on at the end of a read, read-check, write, or control (access movement) operation. It is turned off by a sense device instruction with modifier bit 15 set to one.

Disk Busy. This indicator is on during execution of a disk storage instruction. It is turned off when the operation is complete.

Data Error. This indicator is turned on when a parity (modulo 4) error is detected during a read, read-check, or write instruction. The data error is set also if a sector pulse comes while reading or writing. This indicates that the word count was too large. It is turned off by a sense device instruction with modifier bit 15 set to one.

Disk Not Ready. This indicator is on when disk storage is not ready to receive an instruction, which means that the disk is busy or that a disk interlock has not been properly satisfied.

Carriage Home. This indicator is on when the access mechanism is at the home position (cylinder 00).

Sector Count. These bits represent the sector number 0 to 3 of the current position of the read/write heads. Thus, the quantity represented by the bits is the sector number on the upper track.

3.40.2 Circuit Descriptions

3.40.2.1 Control-Seek

- Move the access mechanism the number of cylinders specified in the IOCC.
- Move in the direction indicated by bit 13: 0 = forward, 1 = backward.
- Move one cylinder first if number of cylinders to move is odd.
- Move balance of cylinders in increments of two.
- Signal operation complete, level 2 interrupt, when read/write head is positioned at the correct cylinder.

- Reference Maintenance Diagrams: AA101, AA231, AA611, XF401, XF521, XF721.

An XIO instruction with an IOCC of control with area code of 4 selects the disk storage attachment. The control function causes motion of the access mechanism as specified in first word of the IOCC.

E1-Cycle. Transfers EA to the M register and makes the M_{15} bit output line active. This causes the core storage location at EA + 1 (odd address) to be selected. The word at this location is loaded into the B register. This word (EA + 1) is the control word of the I/O control command; it contains the area, function, and modifier bits. It is placed in the U register to be analyzed by the I/O adapters.

E2-Cycle. The U register decode of XIO control and area 4 selects the disk storage attachment. The dropping of M_{15} bit addresses EA, which is the number of cylinders to be moved. This is placed on the I/O out bus. This is gated, in complement form, into the word counter. Depending on the U register bit 13, the file load sample pulse turns the head request FF on or off, thus controlling the access direction line. If the FF is on, the heads move forward, if the FF is off, the heads move backward. The file load sample pulse sets the access control FF, activating the access busy line, if the word counter is not full (word count of zero would give word counter full). If a control command is given with a word count of zero, operation complete is not set and no interrupt occurs.

E3-Cycle. Not used.

Access Functions. The access control FF on, the access ready line from the disk drive, and not full word count sets the access drive latch and set the access drive controls in the drive. If the word count was odd, the word counter bit 15 is off and the access moves one increment. The access drive pulse brings up the increment word counter line, but, as the word counter bit 15 was off, the bit 14 position cannot be changed. As the access starts to move in the drive, the access ready line drops and the access drive latch is turned off. The word counter bit 15 is turned on.

When the access completes the movement, the access ready line comes up and a new access drive pulse is developed. This brings up the increment word counter line to word counter 14 and changes its state as bit 15 is now on. The balance of the word counter is connected in binary fashion and as each stage goes off, it changes the state of the next position (Figure 3-10).

I/O Output Bus	15	14	13	12	11	10	9	8	7
Word Count Three	1	1	0	0	0	0	0	0	0
Word Counter Positions	15	14	13	12	11	10	9	8	7
Set Complement	0	0	1	1	1	1	1	1	1
Move One Cylinder	1	0	1	1	1	1	1	1	1
Move Two Cylinders (Word Counter Full)	1	1	1	1	1	1	1	1	1
Operation Complete	0	0	0	0	0	0	0	0	0

22225

Figure 3-10. Word Counter Operation

At the end of each access movement when access ready comes up, the word counter is tested, and when all bits are on, the access control FF is reset. This sets the operation complete FF to give an interrupt level 2.

When the heads are located at cylinder 0 they are at the home position. If a seek command is given in the backward direction while at home, or a word count is given that would take the access past home as the access attempts to move past home to the crash stop, the access home latch is set and the access control FF is turned off. This prevents programming from driving the access into the crash stop repeatedly.

3.40.2.2 Initiate Read

- Read and read check operations are the same with the exception that no cycle steal requests are given on the read check operation.
- Transfer one sector or less of data from the disk storage to core storage on the read operation.
- Modulo 4 check of all data read is made.
- XIO instruction IOCC word selects the disk storage attachment (Area 4) and addresses the word count word in core storage.
- Start the read operation when sector called for in the IOCC word reaches the read/write heads.
- Read zeros until the sync word is read. Sync the bit counter with the sync word.
- Request a level 0 cycle steal cycle after each data word is read.
- Reference Maintenance Diagrams: AA101, AA231, AA611, XF401, XF511, XF711.

An XIO instruction with an IOCC of initiate read with an area code of 4 selects the disk storage attachment.

E-1 Cycle. Transfers EA to the M register and makes the M_{15} bit output line active. This causes the core storage location at EA + 1 (odd address) to be selected. The word at this location is loaded into the B register. This word (EA + 1) is the control word of the I/O control command; it contains the area, function, and modifier bits. It is placed in the U register to be analyzed by the I/O adapters.

E2-Cycle. The U register decode of XIO start read and area 4 selects the disk storage attachment. The dropping of the M_{15} bit addresses EA, which is the address of the word count word in core storage, it is placed in the B register and on the I/O out bus. The file address register is reset and load sample sets the address register with the address of the word count word.

File load sample gates the sector address from the U register to the sector address register, sets the cycle steal request level 0 FF, and turns on the read/write request FF. (Turns off the read/write operation FF and sets the read check operation FF if U register bit 8 = 1).

Cycle Steal Cycle. At the end of the E2 cycle, with cycle steal level 0 up, a cycle steal cycle is taken. The M register output to core storage is blocked and core storage is addressed from the file address register. The word count word is read out to the B register and placed on the I/O output bus. Because this is the first cycle steal cycle the load word counter line gates the word count data into the word counter in a complement form. The file address register is incremented +1. At the end of the cycle steal cycle the cycle steal request FF is turned off.

Search for Selected Sector. During the E2 cycle the number of the sector to be read was placed in the sector address register. Bit 13 from the U register is set into the high order position of the sector address register and is used to select the upper (0) or lower (1) head. Bits 14 and 15 are used to decode the sector to read.

The two position sector counter is turned off by the index pulse each revolution of the disk and advanced with each sector pulse. When the sector address register and the sector counter are equal, the fall of sector pulse causes the read/write select FF to turn on and the read/write request FF to turn off.

Read Selected Sector. The data word format is shown in Figure 3-11 and the sector format is shown in Figure 3-12. With the read/write select FF on, read clock and read data pulses from the disk storage are gated to the attachment. The read clock pulses become counter sample pulses. Nothing happens until the ones (11110) in the sync word are read. The read data pulse turns on the input FF and this gates the modulo 4 counter and the set bit 0 line to the file data register. When the next read clock pulse comes, the modulo 4 counter and the file data register are advanced, and input FF is turned off. The counter sample pulse also fires the shift sample pulse and sets the bit 1 position of the file data register. The shift sample pulse is developed to shift the bit in position 0 of the file data register to position 1 and turn off bit 0. The read data pulses repeat the sequence until four ones and a zero have been read. Figure 3-13 shows the timing relationship of the data from the disk and the file data register advancement. When four ones and a zero have been read, the read/write condition FF is turned on. This gates the bit counter advance. The bit ring advances with each read clock pulse. Figure 3-14 shows the sequence of the bit counter. As each data bit is read, the modulo 4 counter is incremented, and at the end of bit counter E the modulo 4 counter is checked for zero. If it is not zero, the error FF is turned on.

At the end of the data portion of each word, the word counter is checked to see if it is full. If it is not full, the cycle steal request FF is turned on. A level 0 cycle steal cycle is taken and the file data register is gated onto the I/O input bus and placed in core storage. The file address register is incremented + 1 to advance the data field address for the next word.

At the start of bit counter E, the word counter is incremented for the next word. When the word counter is incremented to all bits on, the full word count line comes up, and at the end of bit counter E the

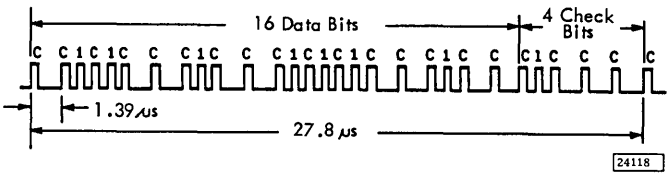


Figure 3-11. Disk File Data Word

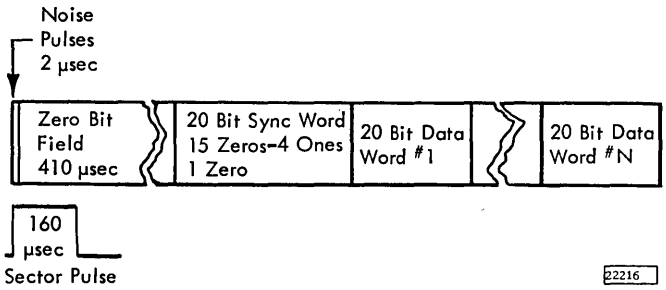


Figure 3-12. Sector Format

operation complete FF is turned on. This gives a level 2 interrupt request so that the program can know that the sector has been read.

Operation complete coming on turns off the cycle steal request, word counter, file data register, bit counter, modulo 4 counter, sector counter, read check operation FF, the input FF's read/write select FF and the read/write condition FF.

3.40.2.3 Initiate Write

- Write one sector or less of data from core storage onto disk storage.
- Write modulo 4 check bits with each word.
- XIO instruction IOCC word selects the disk storage attachment and addresses the word count word in core storage.
- Start write operation when sector called for in the IOCC word reaches the read/write heads.
- Write zero field and sync word.
- Request level 0 cycle steal cycle before each data word is written.
- Reference Maintenance Diagrams: AA101, AA231, AA611, XF401, XF501, XF701.

An XIO instruction with an IOCC of initiate write with an area code of 4 selects the disk storage attachment.

E-1 Cycle. Transfers EA to the M register and makes the M_{15} bit output line active causing the core storage location at EA + 1 (odd address) to be selected. The word at this location is loaded into the B register. This word (EA + 1) is the control word

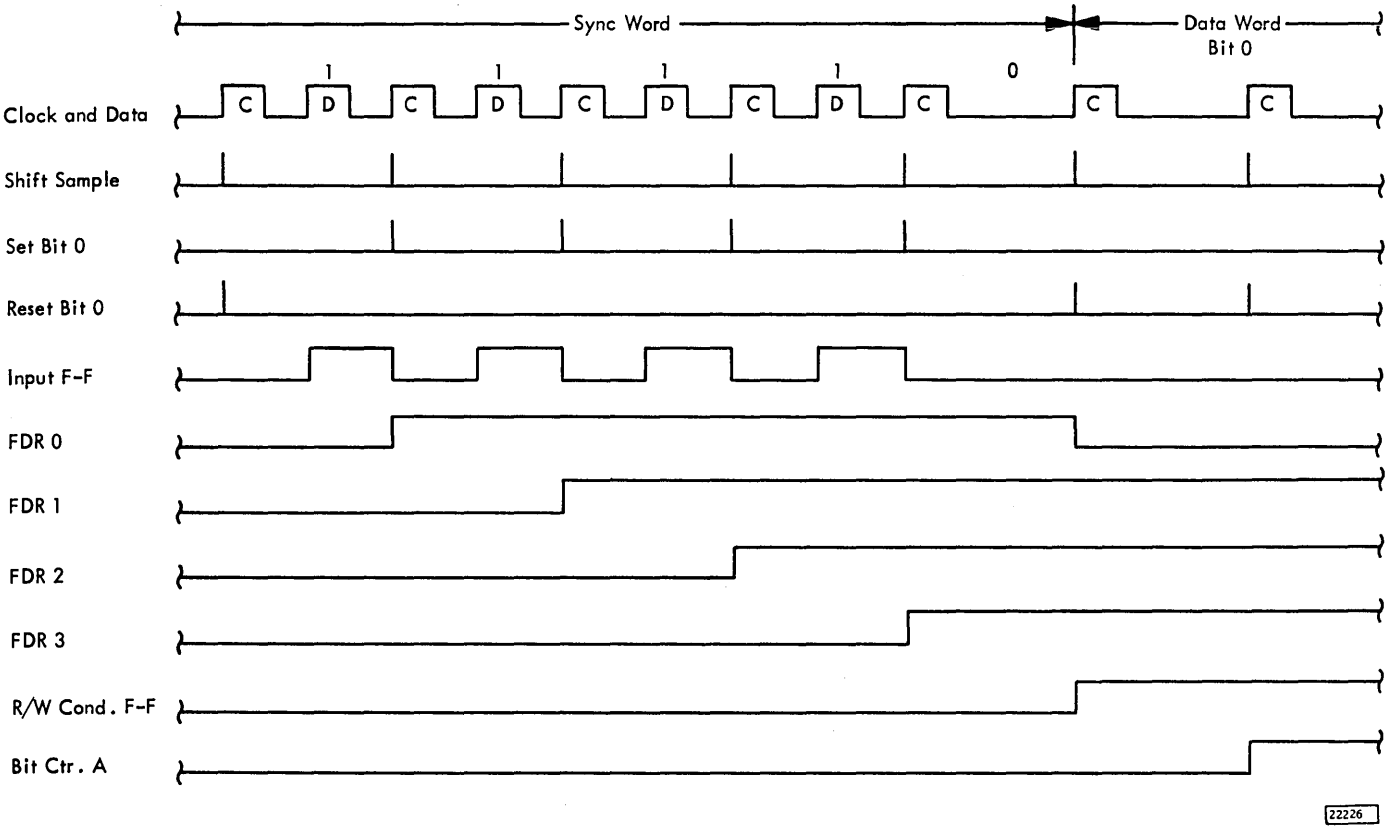


Figure 3-13. Timing Chart - Sync Bit Counter

Bit Positions	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Bit Ctr. A		x		x		x		x		x		x		x		x		x		x
Bit Ctr. B			x	x			x	x			x	x			x	x			x	x
Bit Ctr. C					x	x	x	x					x	x	x	x				
Bit Ctr. D									x	x	x	x	x	x	x	x				
Bit Ctr. E																	x	x	x	x

22227

Figure 3-14. Bit Counter Sequence

of the I/O control command; it contains the area, function, and modifier bits. It is placed in the U register to be analyzed by the I/O adapters.

E-2 Cycle. The U register decode of XIO initiate write and area 4 selects the disk storage attachment. The dropping of the M_{15} bit line addresses EA, which is the address of the word count word in core storage. The word count word address is placed in the B register and on the I/O out bus. The file address register is loaded with the address of the word count word. File load sample gates the sector address from the U register to the sector address register, sets the cycle steal request level 0 FF and turns on the read/write request FF and the read/write operation FF.

Cycle Steal Cycle. At the end of the E2 cycle, with cycle steal level 0 up, a cycle steal cycle is taken. The M register output to core storage is blocked and core storage is addressed from the file address register. The word count word is read out to the B register and placed on the I/O output bus. Because this is the first cycle steal cycle, the load word counter line gates the word count data into the word counter in a complement form. The file address register is incremented + 1. At the end of the cycle steal cycle, the cycle steal request FF is turned off.

Search for Selected Sector. During the E2 cycle, the number of the sector to be written was placed in the sector address register. Bit 13 from the U register is set into the high order position of the sector address register and is used to select the upper (0) or lower (1) head. Bits 14 and 15 are used to decode the sector to be written.

The two position sector counter is turned off by the index pulse each revolution of the disk and advanced with each sector pulse. When both sector address register and the sector counter are equal, the rise of sector pulse causes the read/write select FF to turn on, and the read/write request FF to turn off. Bit 0 of the file data register is set to one with the load sync word line.

Write Selected Sector. The sync word consists of 15 zeros - 4 ones and a zero. The read/write select FF gates the write clock pulses from the drive and they become counter sample pulses. Write clock pulses are sent back to the drive as file data gate to write zeros on the disk.

The fall of the sector pulse times the read/write single shot to control the number of zeros written before the sync word.

When the read/write single shot times out, the next counter sample pulse sets the write sync FF, this becomes bit counter gate.

Bit counter gate and counter sample advance the bit counter. Bit counter gate, not bit counter E, and counter sample give shift pulses to the file data register.

The file data register started with a one in position 0 so 15 zero bits and 1 one are written. A data bit is written and the modulo 4 Counter is incremented each time a 1 is in the file data register position 15. The modulo 4 counter then writes 3 ones and 1 zero to complete the sync word.

At the end of bit counter D, the cycle steal request FF is turned on. At the end of the next CPU cycle, a level 0 cycle steal cycle is taken and the first data word is read out of core storage into the B register and onto the I/O output bus. The load data register line gates the word into the file data register.

At the end of bit counter E, the read/write condition FF turns on and the write sync FF is turned off. The read/write condition FF now brings up the bit counter gate line.

The bit counter gate increments the word counter at the end of each bit counter D time.

When the word counter is full, all bits on, the cycle steal request is blocked and the operation complete FF is turned on. This gives a level 2 interrupt request, and file reset to: cycle steal request FF, word counter, sector counter, file data register, bit counter, the modulo 4 counter, read/write select and read/write condition FF's.

4.1 AC VOLTAGES AND SEQUENCING

- The IBM 1130 System requires a single phase 115 vac-60 cps or 195, 220 or 235 vac-50 cps source of power.
- The line-voltage supply cable terminates at the filter box in the 1131.
- Ac voltage distribution and sequencing is shown on YP101.
- I/O power distribution is shown on ZB101.

The 115 vac input power is filtered to reduce noise, enters the 30 amp circuit breaker and 24 vac step down transformer (Figure 4-1). With the emergency power off switch closed, turning on the power on/off switch picks the main line contactor K1. This powers the convenience outlet and through the C.E. switch the -3 vdc, +3 vdc, +6 vdc supplies, blowers, disk drive, 1132, 1442 and picks RLY2. As the -3 vdc and +6 vdc supplies come up RLY1 is picked to give sequenced power to the 1132. RLY1-1 powers the +12 vdc and +48 vdc supplies. The +48 vdc picks RLY3 and drops the power on reset.

Turning off the power on/off switch drops K1, dropping RLY2 and power to the supplies. The +48 vdc dropping drops RLY3 to set up for power on reset.

4.2 DC VOLTAGES

- +3 vdc, -3 vdc and +6 vdc supplies are standard mid-pac supplies with amplifier card.
- The supplies are shown on YP003 and YP004 and the amplifier cards on YP005.
- The +12 vdc supply is a half wave, resistor-capacitor filtered, ferro-resonant supply shown on YP006.
- The +48 vdc supply is a full wave, resistor-capacitor filtered, ferror-resonant supply shown on YP007.
- The +3 vdc, -3 vdc and +6 vdc supplies are protected for over-voltage and over-current.
- The +12 vdc and +48 vdc supplies are protected for over-current only.
- DC voltage distribution is shown on YP141, YP151 and YP161.
- Reference SLT Power Supplies FEMI. See the bibliography.

4.3 50 CYCLE

The 1130 system provides for 50 cycle - 195, 220, or 235 vac power inputs. Input power is used for the convenience outlets, blowers, 1442, 1132, and the power supplies.

An auto transformer is used to step down the input power to 115 vac for the RLY2 coil, the console printer, and the 1627 Plotter.

Power sequencing remains the same.

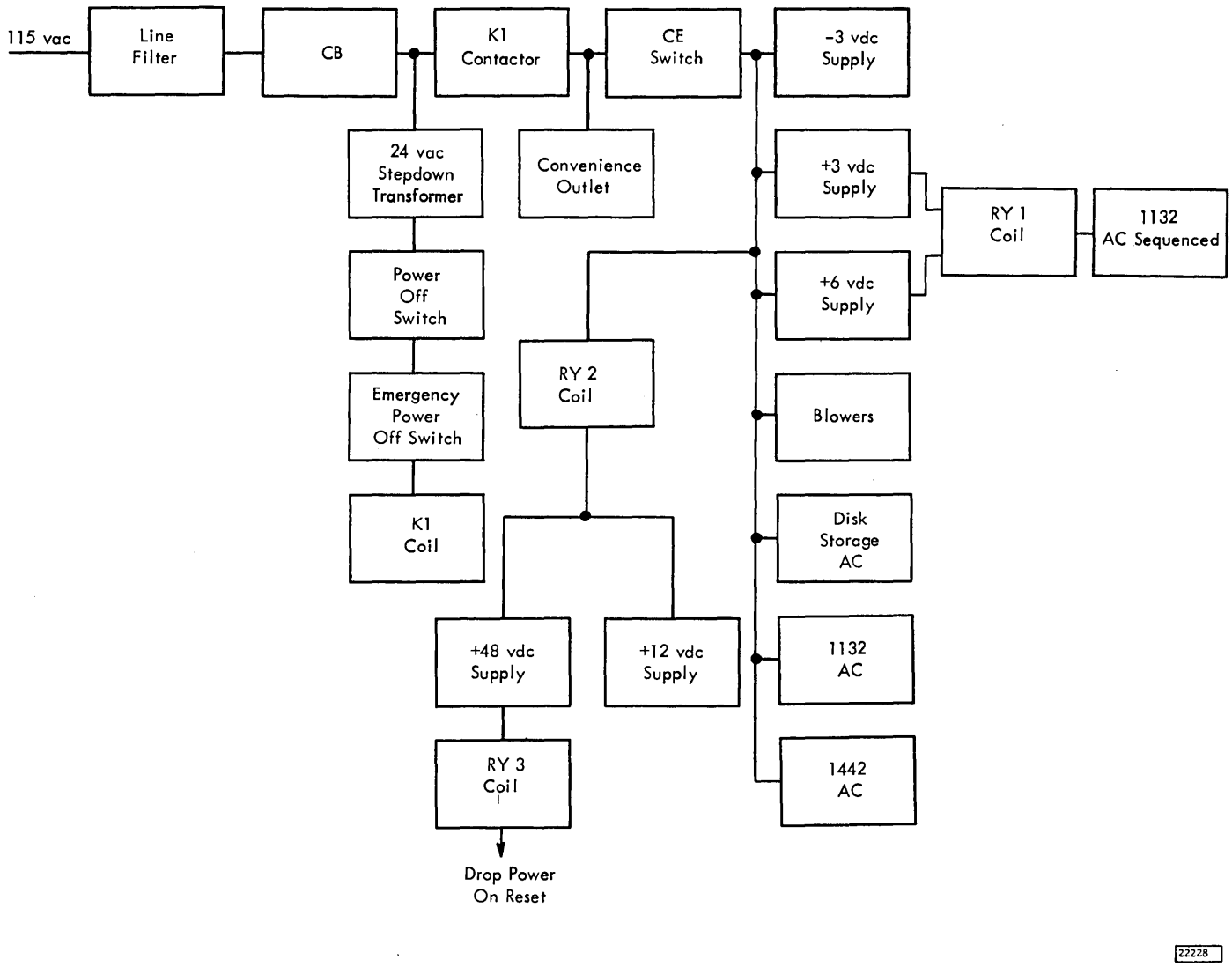


Figure 4-1. Voltage Distribution

APPENDIX A. BIBLIOGRAPHY

This is a list of manuals that contain information that is of value in servicing the IBM 1130 Computing System.

Manual Name	Type	Form Number
IBM 1130 Computing System Functional Characteristics	SRL	A26-5881
IBM 1130 Computing System	FETO	227-5978
Includes: IBM 1131 Central Processing Unit		
IBM Disk Storage Feature		
IBM 1130 Computing System - Features	FETO	227-3670
Includes: IBM 1442 Card Read Punch Feature		
IBM 1132 Printer Feature		
IBM 1627 Plotter Feature		
IBM 1134/1055 Paper Tape Reader-Punch Feature		
IBM 1130 Computing System	FEMM	227-5977
IBM 1130 Computing System	PC	127-0808
I/O Printer (Modified IBM SELECTRIC ®)	FEMI	225-6595
I/O Printer (Modified IBM SELECTRIC ®)	FEMM	225-1726
IBM 1130 Reference Card		X26-3566
Solid Logic Technology Packaging	FEIM	
Solid Logic Technology Component Circuits	FEMI	
Tektronix Oscilloscopes	FEMI	223-6725
Transistor Component Circuits	FEMI	223-6889
Transistor Theory Illustrated	FEMI	223-6794
Transistor Theory and Application	FEMI	223-6783
SLT Power Supplies	FEMI	223-2799
IBM Serial Reader-Punch	FEMM	321-0026
IBM 1442 Card Read-Punch	PC	121-0518
IBM Serial Reader-Punch	FEMI	231-0025
IBM 1442 Models 6 and 7	FEMI	231-0091
IBM 1442 Models 5, 6, and 7	FEMM	231-0098
IBM Single Disk Storage (Serial Numbers 00001 through 39999)	FEMM	227-3668
IBM Single Disk Storage (Serial Numbers 00001 through 39999)	FETO	227-3669
IBM 1134 Paper Tape Reader	FEIM	227-3662
IBM 1134 Paper Tape Reader	PC	123-0542
IBM 1132 Printer	FEMM	227-3621
IBM 1132 Printer	PC	127-0806
IBM 1132 Printer	FEMI	227-3622
IBM 1627 Plotter	FEIM	227-5980
IBM 1627 Plotter	PC	127-0780
IBM 1055 Paper Tape Punch	FEMM	225-3178
IBM 1055 Paper Tape Punch	PC	124-0062
IBM 1055 Paper Tape Punch	FEMI	225-3082

FEMI = Field Engineering Manual of Instruction
FEMM = Field Engineering Maintenance Manual (Formerly Reference Manual)
FEIM = Field Engineering Instruction - Maintenance
FEISD = Field Engineering Instructional System Diagrams
PC = Parts Catalog
FEILD = Field Engineering Intermediate Level Diagrams
FES = Field Engineering Manual Supplement
FESI = Field Engineering Service Index
FEDM = Field Engineering Diagram Manual
FETO = Field Engineering Theory of Operation (Manual of Instruction)

INDEX

READER'S COMMENT FORM

FROM:

NAME _____ DATE _____

OFFICE/DEPT. _____

CHECK ONE OR MORE OF THE COMMENT BLOCKS AND EXPLAIN IN THE SPACE PROVIDED.

☐ ADDITION: PAGE(S) FIGURE(S)

☐ DELETION: PAGE(S) FIGURE(S)

☐ ALTERATION: PAGE(S) FIGURE(S)

FOLD FOLD

EXPLANATION:

CUT ALONG LINE

FOLD FOLD

NO POSTAGE NECESSARY
IF MAILED IN U.S.A.

FOLD ON TWO LINES,
STAPLE, AND MAIL

STAPLE

STAPLE

FOLD

FIRST CLASS
PERMIT NO. 2078
SAN JOSE, CALIF.

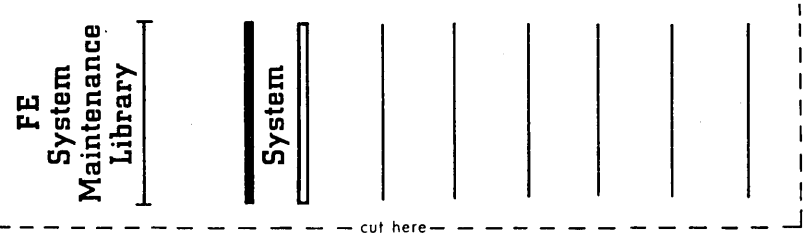
BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY
IBM CORPORATION
MONTEREY & COTTLE RDS.
SAN JOSE, CALIFORNIA
95114

ATTN: PRODUCT PUBLICATIONS DEPT. 455

CUT ALONG LINE

FOLD



227-5978-2

IBM 1130 Printed in U.S.A. 227-5978-2

IBM
International Business Machines Corporation
Field Engineering Division
112 East Post Road, White Plains, N.Y. 10601