CONTROL DATA®

GRID

DISPLAY SUBSYSTEM

HARDWARE REFERENCE MANUAL

PRELIMINARY
DRAFT

- General Description
- Operation
- Programming

CONTROL DATA CORPORATION

DATA DISPLAY DIVISION
2401 NORTH FAIRVIEW AVENUE
SAINT PAUL, MINNESOTA 55113

GRID
DISPLAY SUBSYSTEM
HARDWARE REFERENCE MANUAL

SECTIONS IN THIS MANUAL:

Section I —— General Description
Section II —— Operation
Section III —— Programming

Any comments concerning this
publication should be addressed to:

Control Data Corporation
Data Display Division
Technical Services Department
2401 North Fairview Avenue
St. Paul, Minnesota 55113

Publication No. PD82134500
September, 1969
PD-I-0

GRID

DISPLAY SUBSYSTEM

HARDWARE REFERENCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
PD-0-0	Released August 1968
PD-0-1	Interim change written to correct various errors. Changed pages i, 1-3, 1-4, 1-10, 1-12, 1-15, 1-16, 2-3, 2-5, 2-6, 2-12, 2-13, 2-13a, 2-14, 2-15, 2-16, 2-17, 2-18, 2-19, 2-20, 2-22, A3-6, A3-9, B3-3, B3-4, B3-5, B3-7, B3-9, B3-14, B3-15, B3-19, B3-20, B3-20A, B3-21, B3-34B, B3-38, B3-38B, B3-38C, B3-38D, B3-39, B3-40, B3-47, B3-48, B3-49, B3-51, B3-52, B3-55, B3-56, B3-63, B3-64, and B3-68. Added pages A, 1-3A, and B3-20B.
PD-1-0	Manual reprint includes and obsoletes all previous changes, also includes miscellaneous other changes.

FOREWORD

This manual presents an overall view of the Grid Subsystem together with detailed operating and programming information. Explanations assume only a basic knowledge of data processing and display methods and address three general classifications of readers: system planner, operator, and programmer.

The text is in three sections. The first explains both operational and functional characteristics of the subsystem. Section two thoroughly discusses the operator controls and indicators and their use. Programming information in section three, including flow charts and sample programs, enables effective user application of the subsystem.

The following additional publications may be ordered through the Control Data Literature Distribution Center.

Control Data
Literature Distribution Services
9200 Bloomington Freeway
Minneapolis, Minnesota, 55431

A three-book Customer Engineering Manual documents the Display Controller.

- Book 1 (publication no. PD82134600) — contains six sections; General Description, Operation and Programming, Installation and Checkout, Theory of Operation, Maintenance, and Maintenance Aids. The first two sections act as a supplement to the Hardware Reference Manual. Illustrated instructions for installing the equipment at the site as well as procedures for making it operational constitute the third section. The Theory of Operation is concerned with internal operations. A functional approach is used to eliminate the necessity of tracing through the logic diagrams. This approach links the various functional areas in the controller and provides insight into techniques used to perform various operations. After absorbing this material, the reader should have little difficulty understanding the logic diagrams. Maintenance consists of preventive and corrective maintenance procedures while Maintenance Aids provides useful information for customer engineers.

- Book 2 (publication no. PD82134700) — this book contains logic diagrams, timing charts, block diagrams, card placement charts, schematic diagrams and interconnection diagrams. In many instances, the back of the preceding sheet is used to explain functions shown on the diagram. This serves as an additional "Theory of Operation," from a more analytical viewpoint.

- Book 3 (publication no. PD82134800) — this book contains replaceable parts information for the Display Controller. A provisioning parts list identifies parts called out on illustrated parts drawings. It also contains a list of vendors.

DISPLAY CONSOLE

A single-volume Hardware Reference/Customer Engineering Manual (publication no. PD82134900) is available which has its sections structured in a manner similar to that for the Display Controller. The theory of operation has been integrated with the various schematics and waveforms which have been added to interconnection diagrams.

TABLE OF CONTENTS

Section			Page
I	—	GENERAL DESCRIPTION	
		Subsystem Characteristics	1-2
		Operational Description	1-5
		On Line/Off Line Subsystem Status	1-5
		Internal Subsystem Operating Modes	1-6
		Functional Description	1-10
		Display Controller	1-11
		Display Console	1-15
		Physical Description	1-16
		Environmental Data	1-16
		Power Data	1-18
II	—	OPERATION	
		Monitor Controls and Indicators	2-1
		Operator Controls	2-4
		Light Pen	2-5
		Alphanumeric and Function Keyboard	2-9
		Operator Control Panel	2-13
III	—	PROGRAMMING	
		PART A	
		Interface	A3-1
		Data Transfer	A3-1
		Connect	A3-2
		Functions	A3-3
		Subsystem Status	A3-8
		PART B	
		Arithmetic	B3-1
		Memory	B3-3
		Memory Registers	B3-3
		Bank Control	B3-4

TABLE OF CONTENTS (CONT)

Section | Page

PART B (CONT)

Section	Page
Processor Instruction Repertoire	B3-5
Word Format	B3-6
Address Modes	B3-6
Instructions	B3-13
Stop Instructions	B3-15
No Operation Instruction	B3-15
Data Transmission Instructions	B3-15
Arithmetic Instructions	B3-20
Shift Instructions	B3-23
Logical Instructions	B3-26
Memory Bank Controls	B3-29
Jump Instructions	B3-30
Interrupts	B3-34B
Input/Output Instructions	B3-38A
Display Repertoire	B3-39
Enter Mode Commands	B3-39
Jump Commands	B3-55
Identifier Word	B3-56
Start Refresh	B3-60
Operator Device Control	B3-61
Symbol Repertoire	B3-63
Display State and Keyboard Registers	B3-65
Display State Register	B3-66
Keyboard Register (R_2)	B3-67
Programming Hints	B3-69
Example Program	B3-70

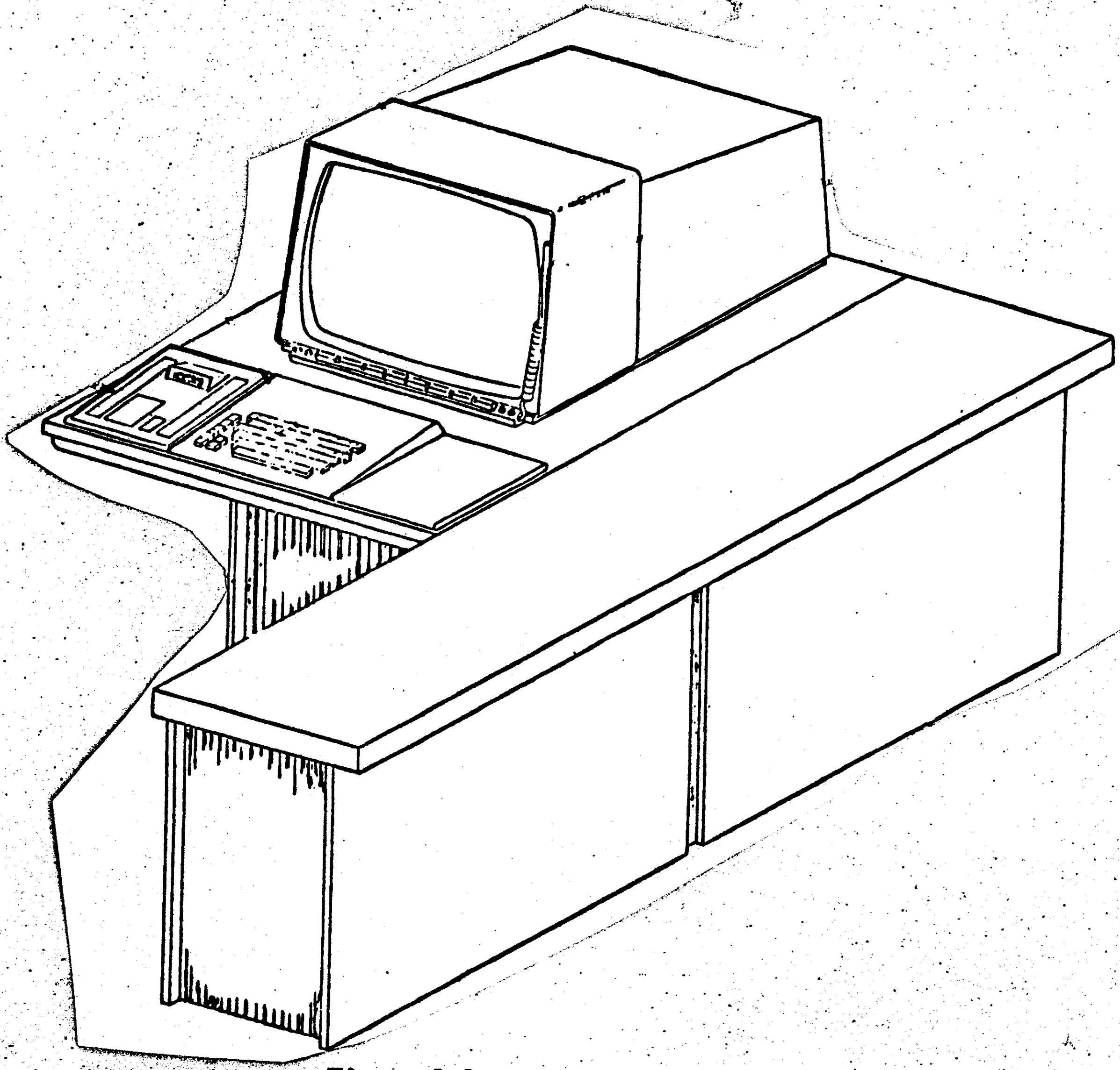


Figure 1-1. Graphic Display Subsystem

SECTION I

GENERAL DESCRIPTION

The graphic display subsystem is a computer-controlled, integrated circuit, stored program, visual data processing subsystem. The basic subsystem consists of a Display Controller and Display Console. In conjunction with a large-scale data processing system, the graphic display subsystem enables solving complex scientific and data processing problems. The subsystem features an internal stored program processor with an optional input/output channel for peripheral equipment communication, enabling small-scale data processing, effective use of the operator controls — alphanumeric/function keyboard and light pen — and visual display. Stored program and input/output capabilities minimize subsystem to computer communications, freeing the large system for other tasks. This enables more effective use of the large system and provides the subsystem with access to increased data handling capability.

SUBSYSTEM CHARACTERISTICS

- Stored program
- Parallel mode of operation
- Binary arithmetic---modulus $2^{12}-1$ (one's complement)
- Single address logic
- 12-bit storage word
- Magnetic core storage

  Basic --- 4096 words

  Expanded --- 8192 words or 12,288 words

  Cycle time --- 1.2 microseconds

  Access time --- 0.8 microseconds

  Relative addressing

  Direct addressing

  Indirect addressing
- Input/Output

  CONTROL DATA 3000 Series interface

  12-bit bytes/parallel transfer

  Maximum transfer rate --- 208,000 bytes/second

  Up to 200 ft. logic cables --- subsystem to data source
- Interrupt system
- Flexible processor and display repertoires

- Processor execution times—2.4 to 9.2 microseconds
- Processor input/output channel

  Provides peripheral equipment communication

  Maximum transfer rate—96 KHz

  Maximum cable length—75 feet

  Maximum number of peripheral controllers—5
- Display times

  Random point plot—4.0 to 12.0 microseconds

  Incremental point plot—4.0 microseconds

  Random symbols—5.2 to 18.4 microseconds

  Incremental symbols—5.2 to 8.4 microseconds

  Tabular symbols—2.8 to 8.4 microseconds

  Vectors

  Full-scale diagonal (approximately 16 inches) — 30 microseconds

  2-inch vector—less than 12 microseconds

  1/2-inch vector—less than 9 microseconds
- Display characteristics

  21-inch electromagnetic deflection cathode ray tube (crt)

  Display area—12 inches by 12 inches

  Spot size—0.030 inch (maximum)

Positioning accuracy— ±1%

Large symbols—0.210 inch high by 0.160 inch wide (nominal)

Small symbols—0.166 inch high by 0.125 inch wide (nominal)

Display refresh rate—nominally 50 hertz (variable)

Display intensity—0 to 50 foot lamberts

Basic Symbol set—64 symols

Extended Symbol set—64 symbols

- Solid state logic

  Digital ---DTL integrated circuits
  ---TTL adder

  Analog ---discrete components

  Easy access pluggable logic board

- Operator controls

  Alphanumeric/function keyboard

  Light pen

  Operator Panel

- Power

  60-hertz, 3-phase, 208 volt

  60-hertz, single-phase, 120 volt

OPERATIONAL DESCRIPTION.

The data source executive program controls all communication between the data source and display subsystem. Before data source/ subsystem communication can take place, the data source must connect the subsystem to the communication line by transmitting the subsystem connect code. After connection, function codes set up subsystem operating conditions and initiate operations. Upon request, the data source can obtain information on the operating status of the subsystem.

ON LINE/OFF LINE SUBSYSTEM STATUS.

When the display subsystem is on line and ready for service, the data source transfers data or programs to the subsystem for storage into memory. The subsystem can also transmit data from memory to the data source.

The off-line condition provides for maintenance and program debugging. The operator uses the operator control panel to simulate data source operations or monitor internal operations of the display subsystem. Except for the interface module which is locked out while the subsystem is off line, operation within the display subsystem is the same in either the on-line or off-line states.

INTERNAL SUBSYSTEM OPERATING MODES.

There are two basic modes of operation within the subsystem, display mode and processor mode. Since both display and processor share the same arithmetic and memory circuitry, they cannot operate simultaneously. Operating mode can be selected by any one of three sources: data source; operator; or subsystem internal program.

Stored processor and display programs permit independent subsystem data processing and graphic display. In conjunction with stored programs, operator controls -- light pen and alphanumeric/function keyboard -- provide manual data entry and display intervention capabilities. Stored programs enable independent subsystem operation, freeing valuable data source computation time.

Processor.

A flexible processor instruction repertoire enables solving complex data processing and engineering design problems. An interrupt system enables effective interchange of subsystem and data source information.

Interlacing execution of processor and display programs presents graphic and alphanumeric displays under operator control. Enabled operator controls generate manual interrupts when activated. A manual interrupt initiates a processor interrupt routine. By interrogating appropriate registers, the interrupt routine identifies the interrupt source and initiates a pre-programmed response to the interrupt. In this manner, the operator enters program parameters, selects alternate courses of action (different program subroutines), or modifies specific display entities.

Display.

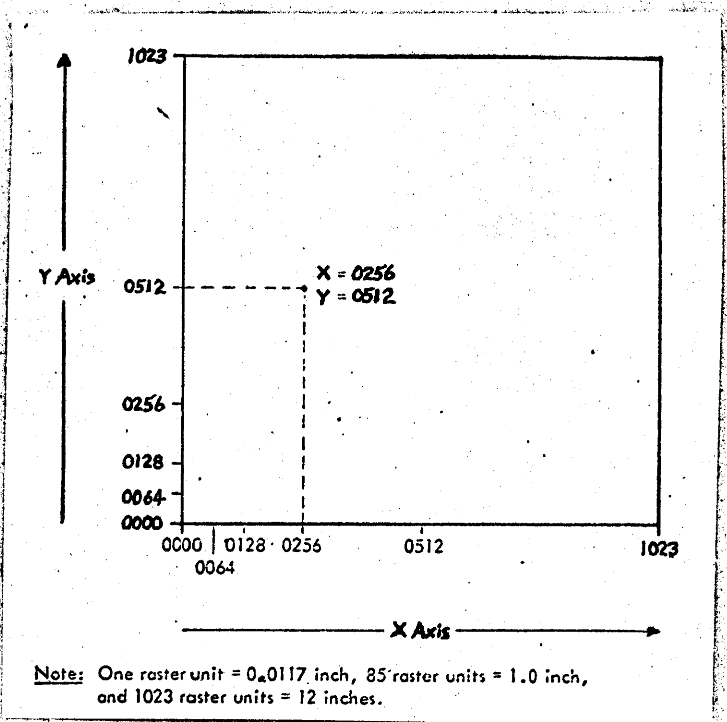
In display mode, a display program generates images on the 12-inch by 12-inch usable display area of the 21-inch monitor crt. Vectors (straight lines), points, and symbols comprise a display image.

Image Generation.

An electron beam in the crt produces a visible display when it strikes the phosphor coated crt screen, causing that portion of the phosphor to glow briefly. Normally the glow fades within a fraction of a second, too soon for the human eye to perceive and identify the image. For this reason, the display image must be redrawn continuously (refreshed) at a rate which makes the display appear steady and of uniform intensity to the observer. Display refresh is under program control. To prevent damaging the crt phosphor, the display program must initiate a hardware refresh delay at the end of each main display file.

Deflection.

Position words, associated with display commands, control electron beam positioning (deflection) on the crt screen. These position words specify electron beam deflection to horizontal (X) and vertical (Y) coordinates on a square grid composed of possible electron beam positions. This grid (called a raster) covers the 12-inch by 12-inch display area on the crt screen. There are 1024 equally spaced X positions and 1024 equally spaced Y positions, figure 1-2.



Note: One raster unit = 0.0117 inch, 85 raster units = 1.0 inch, and 1023 raster units = 12 inches.

Figure 1-2. Display Area Coordinate System

Position words in the display program select the X and Y coordinates for each element of a display (each point, vector end point, and symbol matrix center). A point appears directly on a coordinate position. A vector is a straight line between any two coordinate positions. The crt beam paints a symbol within an imaginary 5 by 7 symbol matrix centered around a coordinate position.

The distance between two sequentially addressable lines on the raster is called a "raster unit". A raster unit represents 1/1023 of an image in either the X or Y direction.

FUNCTIONAL DESCRIPTION.

The functional units of the display subsystem breakdown as shown in figure 1-3.

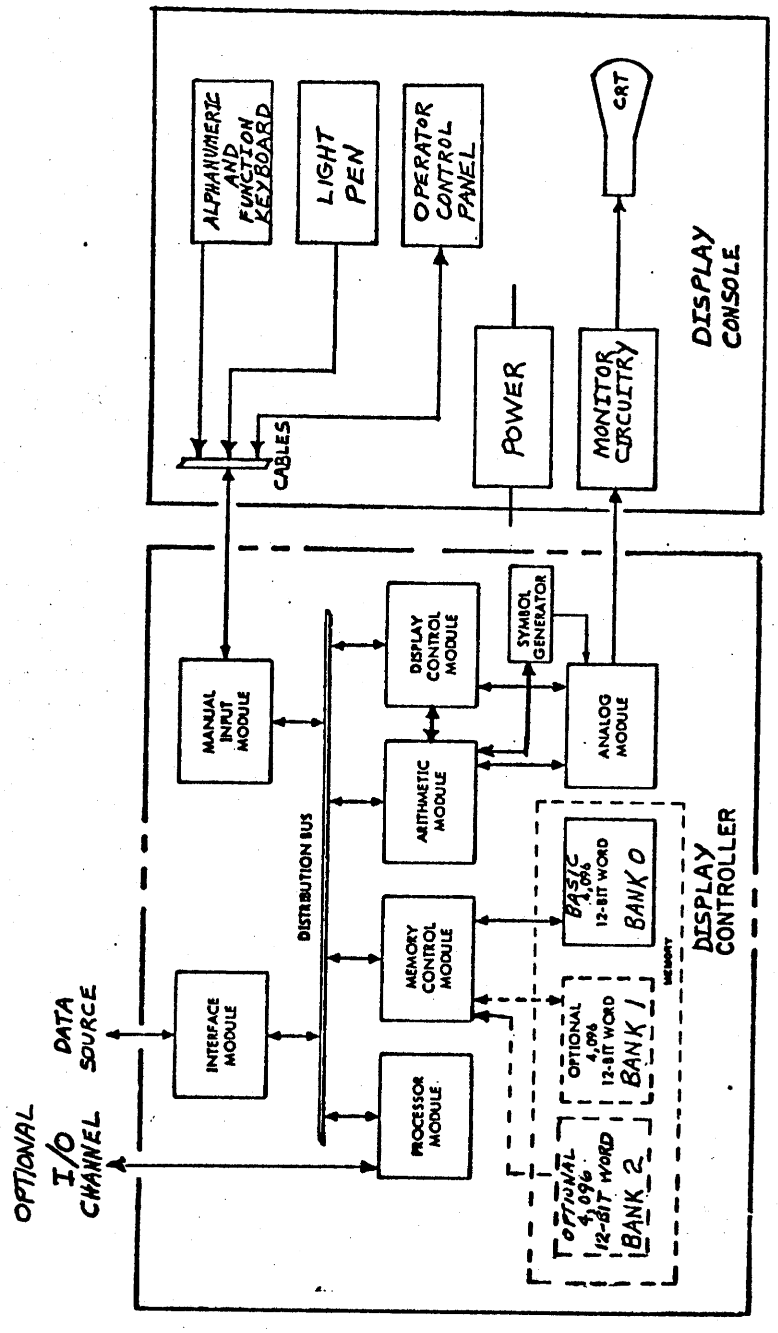


Figure 1-3. Subsystem Functional Diagram

DISPLAY CONTROLLER.

Seven functional modules in the controller perform all the data transfer, processing, and visual display operations.

Six of the functional modules control all logical operations. Each of the six modules connect to a distribution bus which, as the name implies, distributes data transferred from one block to the next. Specific combinations of blocks are active during execution of programs. Inactive blocks ignore distribution bus data.

The analog module interfaces to, and operates in conjunction with, the display control and arithmetic modules. Circuits in the analog module perform digital-to-analog conversions required for presentation of a crt display.

Interface Module.

Via communication lines, the interface module links the display subsystem to and controls communication with a data source.

Processor Module.

When in processor mode, the processor module functions as the principal control element governing execution of program instructions. The processor accesses memory to obtain instructions for execution and to obtain operands and data for processing. The processor continues executing instructions until termination of processor mode. Optional processor control enables communication with peripheral equipment via an unbuffered input/output channel.

Display Control Module.

Display control logic directs actions of the analog, arithmetic, and memory control logic in generating a display. Symbols, points, and vectors comprise all displays. Display logic accesses memory for display commands and continues executing commands until termination of display mode.

Arithmetic Module.

Arithmetic logic serves both the processor and display modules.

During display mode, arithmetic logic computes crt beam positions and vector lengths.

During processor mode, arithmetic logic adds, subtracts, multiplies, shifts, and performs logical operations required for data manipulation.

Memory Control Module.

Memory control accepts commands from the processor, the interface, manual input, or the display control logic to initiate memory read/write cycles. The memory control logic includes the circuitry necessary for random addressing and automatic address incrementing. Data stored in or read from the core memory is transferred into or out of the memory module via the distribution bus.

Core Memory.

Each core memory operates in conjunction with the memory control module. Within the memory control module, special address translation circuits allow the control circuits to recognize the memory bank selected for data storage or data retrieval.

Manual Input Module.

Manual input logic provides a data transfer path between the controller and operator controls. This transfer path operates during message entry or during operator modification of a graphic display using the light pen or keyboard. Manual input logic also operates in conjunction with controls on the operator panel, permitting off-line subsystem operation.

Analog Module.

Circuits within the analog module convert binary data, in the form of position words and symbol words, into the analog signals received at the monitor. Analog signals deflect the crt electron beam, generating display images specified by the display module.

DISPLAY CONSOLE.

The Display Console is the subsystem unit with which the operator views graphic displays, modifies displays, enters program data, contols off-line operation, and controls subsystem power on/off. There are three major functional areas in the Display Console: monitor, operator controls, and power cabinet.

Monitor circuits receive anolog signals from the Display Controller analog module and convert these signals into visual images on a 21-inch electromagnetic cathode ray tube. Controls permit operator adjustment of display position and quality.

Operator controls -- alphanumeric/function keyboard, light pen, and operator control panel -- provide manual data entry and display intervention capabilities as well as program debugging facilities.

The power cabinet contains power distribution control circuitry and logic, analog, and memory power supplies. A running time meter, voltage adjustments, and temperature sensing controls are provided.

PHYSICAL DESCRIPTION.

The display subsystem is 52-1/2 inches wide, 72 inches deep, and 48 inches high. All table tops or operator working areas are 29 inches from the floor. Figure 1-4 shows unit dimensions.

All external cabling is through the bottom and rear of the equipment. The equipment weighs approximately 820 pounds. Motor generator weighs approximately 125 pounds.

ENVIRONMENTAL DATA.

Table 1-1 lists the environmental ranges fur subsystem operating and nonoperating status.

TABLE 1-1. ENVIRONMENTAL DATA

CONDITION	OPERATIONAL	NONOPERATIONAL
Temperature	+40 F to +110 F	-30 F to + 150 F
Relative Humidity	10% to 90%	5% to 95% (including condensation)
Altitude	-1,000 to 8,000 feet	-1,000 to 15,000 feet

Ambient air cools the subsystem. A blower forces air through a plenum at the bottom of the controller cabinet and up through the logic. Air exhausts through vents beneath the cabinet top. Internal console blowers draw in and exhaust cooling air through vents at the rear. Heat dissipation is 8350 BTU/hour.

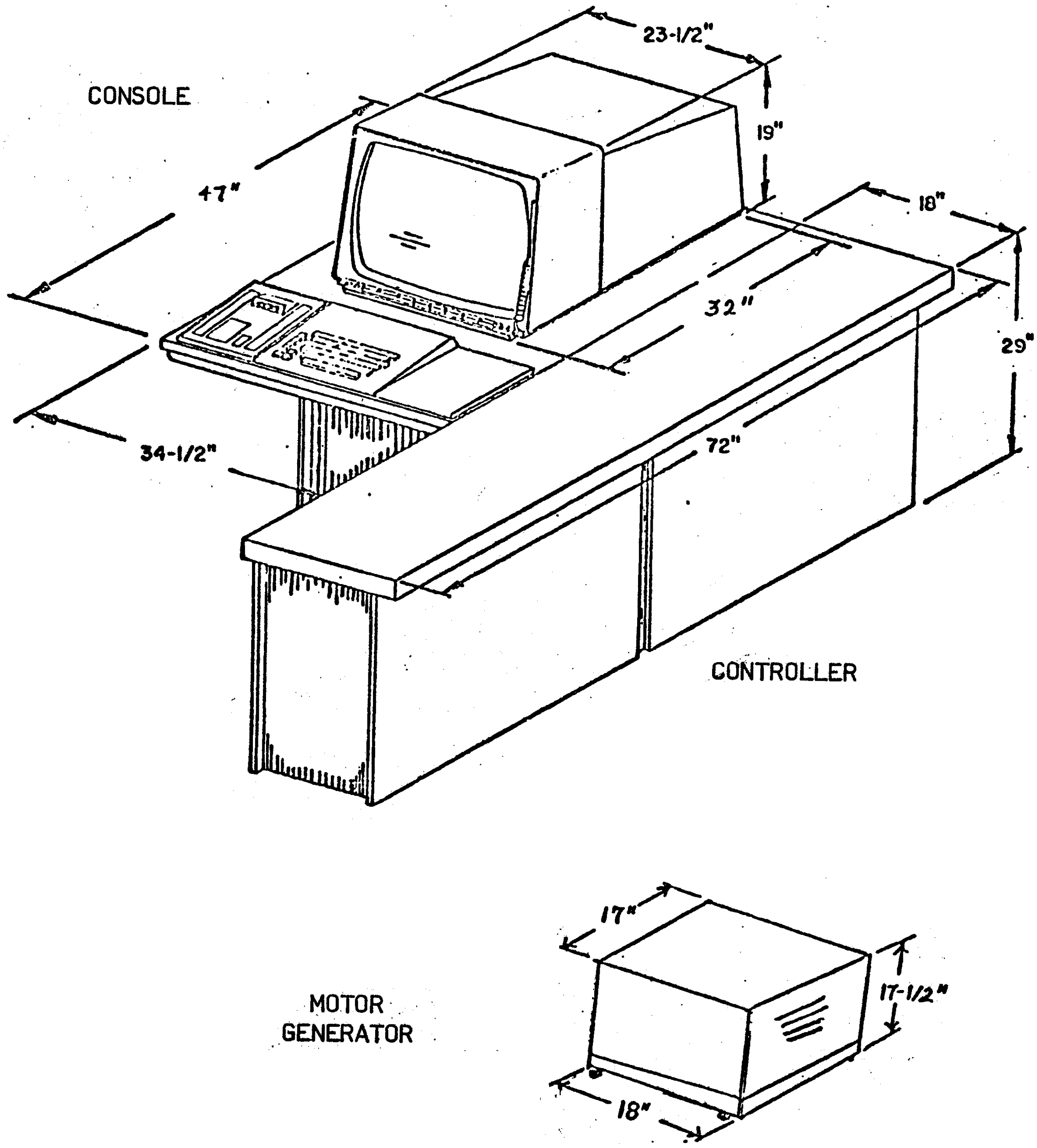


Figure 1-4. Controller, Console, and Motor Generator Dimensions

POWER DATA.

The display subsystem operates on 60-hertz, 3-phase, 208-volt power at approximately 10 amperes and 60-hertz, single-phase, 120-volt power at approximately 3 amperes. Total power consumption is less than 2,000 watts.

POWER ON and POWER OFF switch/indicators on the monitor front panel control power application.

SECTION II

OPERATION

The display subsystem has two major groups of controls and indicators, monitor and operator. Monitor controls and indicators control display quality and indicate subsystem operating conditions. Operator controls link the operator, display subsystem, and data source in a man/machine communication loop.

MONITOR CONTROLS AND INDICATORS.

Figure 2-1 shows the monitor and associated controls. Four main controls adjust the display. These are: HORIZ (horizontal) and VERT (vertical) CENTERING controls, for adjusting display positioning; and FOCUS and INTENSITY controls for adjusting display appearance. Table 2-1 lists and explains the remaining monitor controls and indicators.

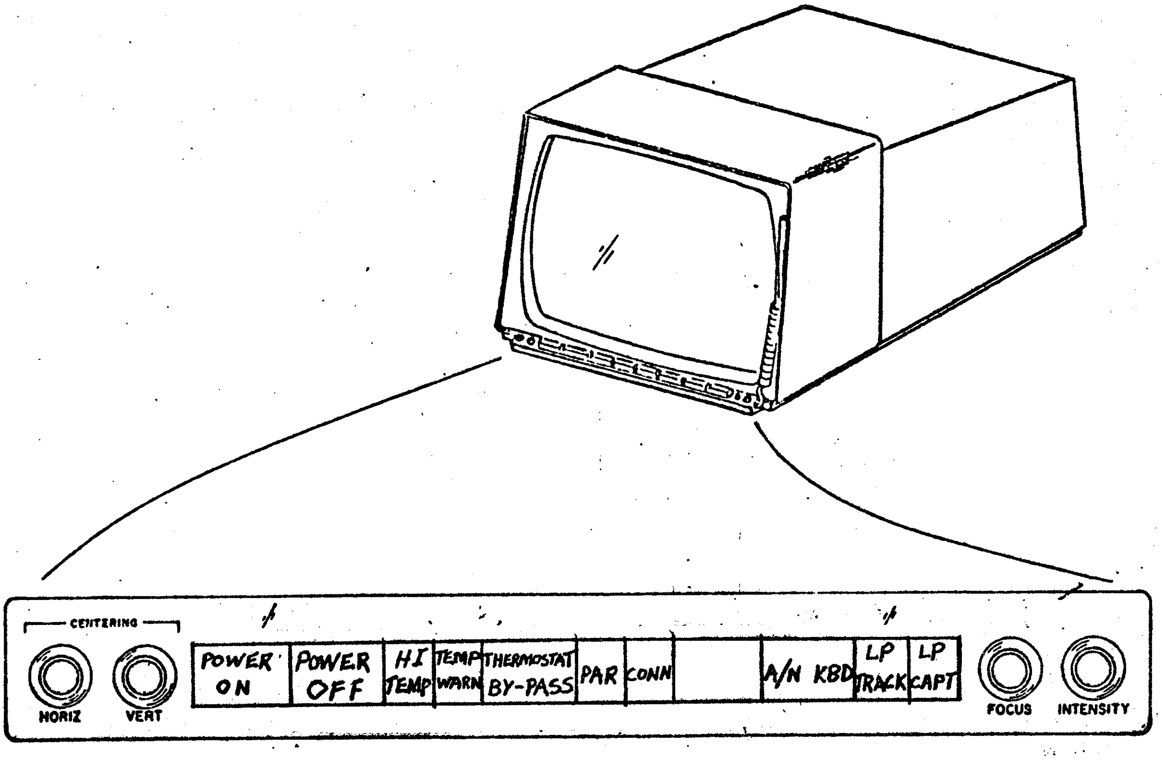


Figure 2-1. Monitor

TABLE 2-1. MONITOR FRONT PANEL CONTROLS AND INDICATORS

NAME	TYPE	DESCRIPTION
POWER ON	Switch momentary Indicator	Depression applies subsystem power. Indicator lights while power is on.
POWER OFF	Switch momentary	Depression removes subsystem power.
HI TEMP/TEMP WARN	Indicator split	HI TEMP lights when an interior thermostat determines an unsafe temperature and removes subsystem power. TEMP WARN lights when subsystem interior air rises above a preset temperature.
THERMOSTAT BY-PASS	Switch alternate action Indicator	Depression bypasses the hi temp thermostat and allows continued operation. Indicator lights when the subsystem is in a bypass condition.
PAR/CONN	Indicator split	PAR indicates a parity error in transmission from the data channel to the subsystem. Turned off by master clear. CONN indicates subsystem connection to the data channel for input/output operations.
BLANK	—	Reserved.
A/N KEYBOARD	Indicator	Lights after program selection of the keyboard.
LP TRACK/LP CAPT	Switch alternate action Indicator split	Lights after program selection of the light pen, switch depression selects the desired light pen operating mode. LP TRACK lights for tracking mode. LP CAPT lights for capture mode.

OPERATOR CONTROLS.

Operator controls — light pen, alphanumeric/function keyboard, and operator control panel — enable effective man/machine communication. Figure 2-2 shows the operator controls.

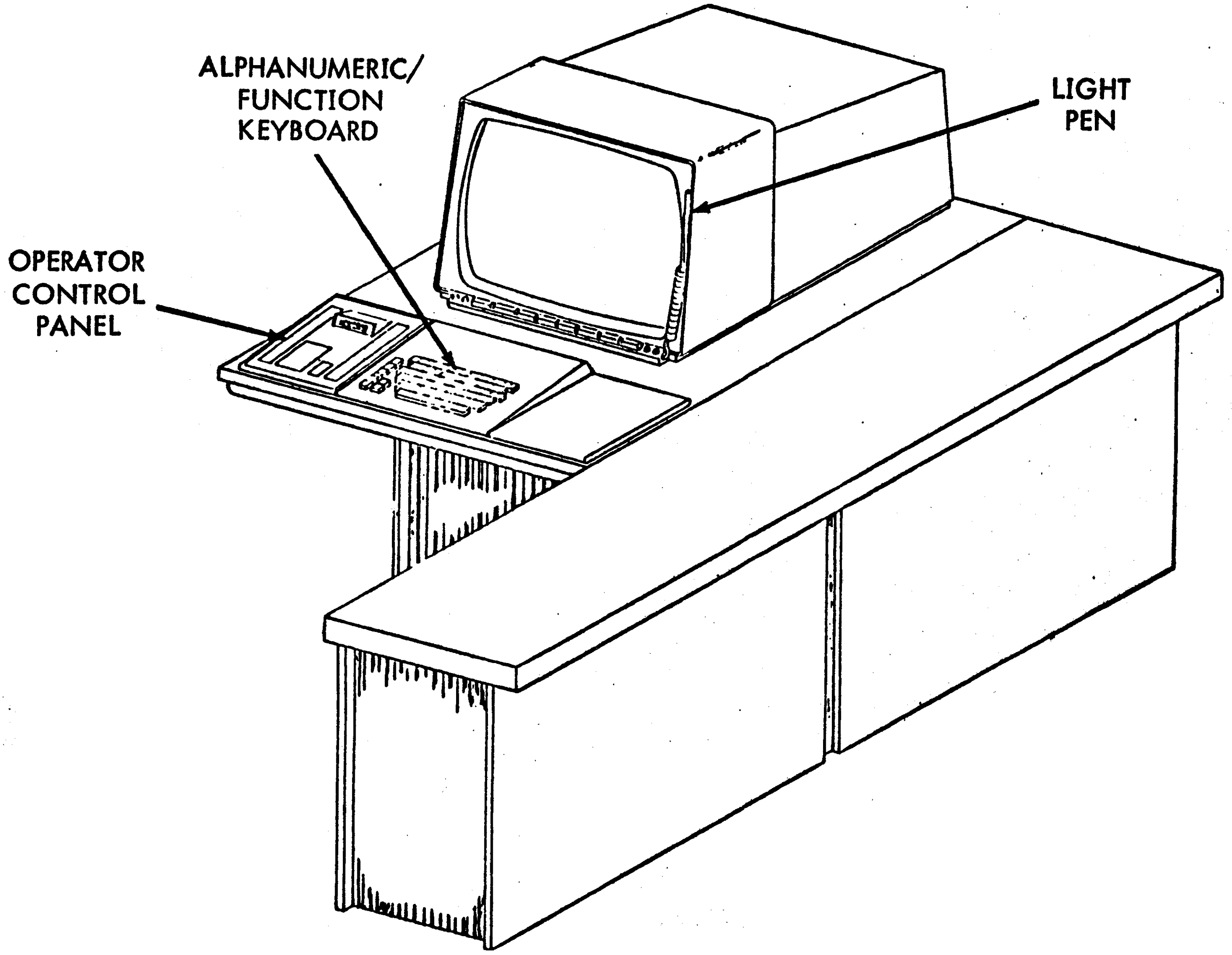


Figure 2-2. Operator Controls

LIGHT PEN.

The light pen, figure 2-3, is a solid-state, high-speed, photosensitive stylus actuated by a two-position, momentary switch. It operates in either of two program selected modes, track or capture. Aiming parentheses of light, emitted from the light pen tip, enable operator selection of a specific display item. By placing the tip of the light pen on or near the face of the crt, and depressing the activate switch, the operator may selectively modify the displayed image under program control.

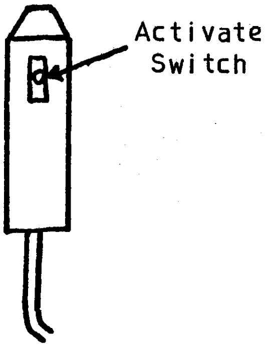


Figure 2-3. Light Pen

Because the light pen senses changes in light intensity only, it "sees" data on the crt at the instant the electron beam passes (unblanks) that data. When the light pen senses unblank ("hit"), it generates an interrupt. The interrupt halts display operation, stores the next program address at direct address 10_8, and initiates a processor interrupt routine at relative address 11_8. This routine is set up by the programmer, not hardwired. In general, the interrupt routine performs the following tasks:

(a) Stores X position register contents.

(b) Determines interrupt source.

(c) Indentifies the light penned item or group.

(d) Performs associated subroutine.

(e) Clears interrupt lockout .

(f) Returns to display mode.

Program interrogation of the manual input register determines the interrupt source. Examination of the X, Y position register, Direct memory address 10_8, or display state register identifies the light penned display item or group. After determining which item or group the operator pointed to, a jump to the associated processor subroutine occurs. The subroutine may modify the display or interrupt the data source. Upon subroutine completion, the processor enables further manual interrupts by executing a clear interrupt lockout (120_8) instruction. The processor may restore the original display parameters and resume the display program from the interrupted point.

Operating Modes.

Before processor recognition of light pen hits, the display program must enable the light pen. With the light pen enabled, the operator selects "track" or "capture" mode via the LP TRACK/LP CAPT switch/indicator on the monitor front panel. The respective indicator informs the operator of the selected mode.

Light pen mode determines handling of light pen hits. In track mode, all light pen hits generate manual interrupts as long as the activate switch on the light pen remains depressed. In capture mode, only one light pen hit per activate switch depression generates a manual interrupt.

By appropriate positioning of light pen enables and disables throughout a program, a programmer may protect individual display items from light pen operations.

Typical Application.

One application is drawing images on the crt with the aid of a programmed tracking cross, figure 2-4. Also, reference the identifier word description in part B of Section III for an example. A tracking cross is an array of dots on the crt used for locating points and lines or drawing. Under program control, light pen selection of a dot in the tracking cross draws a vector from the center dot of the cross in the direction of the selected dot. The program determines vector length. The program repositions the center of the tracking cross to the end of the vector just drawn. In this manner, the operator draws on the crt. Vector length and number of tracking cross dots determine line resolution.

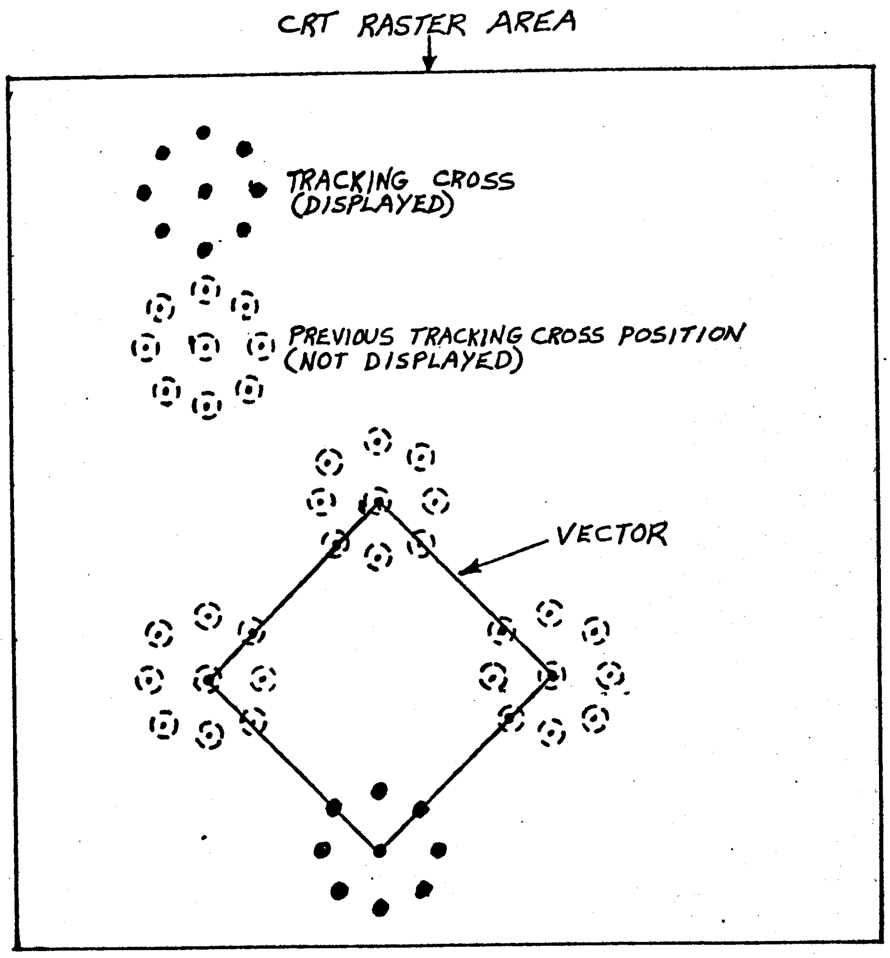

CRT RASTER AREA
TRACKING CROSS (DISPLAYED)
PREVIOUS TRACKING CROSS POSITION (NOT DISPLAYED)
VECTOR

Figure 2-4. Tracking Cross Application (Not to Scale)

ALPHANUMERIC AND FUNCTION KEYBOARD.

Figure 2-5 shows the alphanumeric/function keyboard.

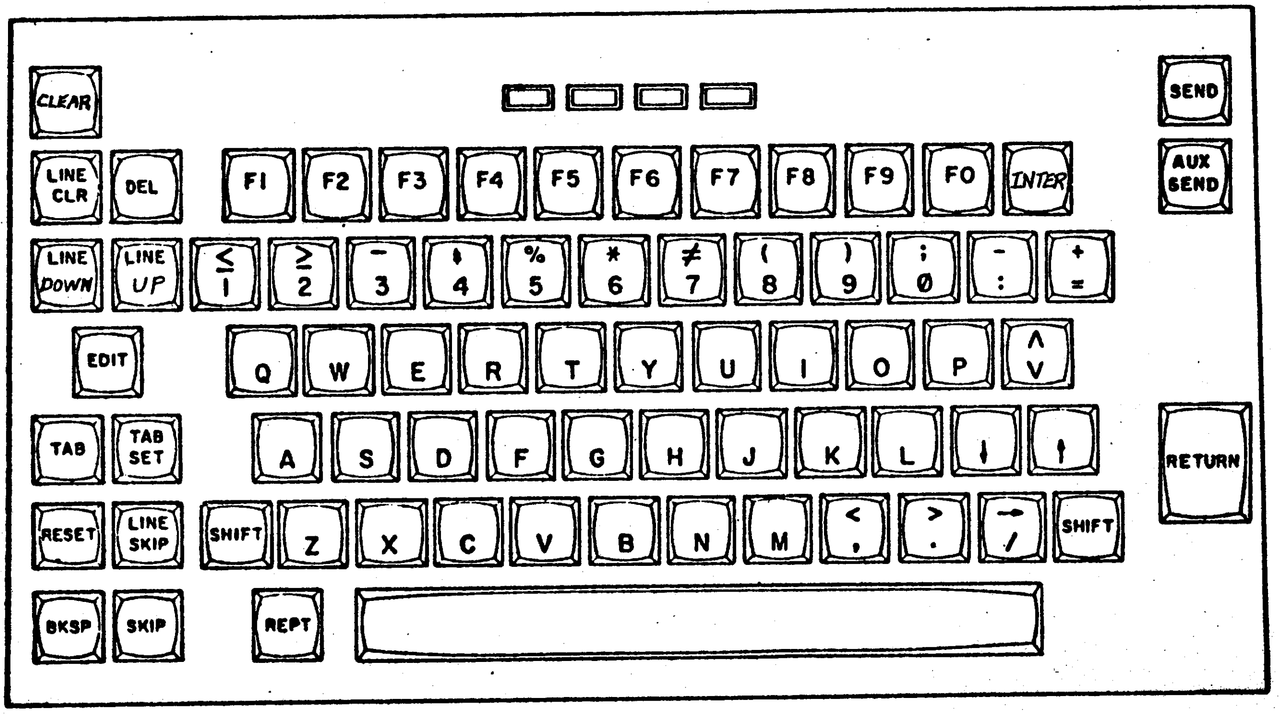


Figure 2-5. Alphanumeric/Function Keyboard

Alphanumeric.

The alphanumeric portion of the keyboard is a switch actuated, data-entry device. Alphanumeric keys enable message editing and composition by an operator.

Operation.

Enabling or disabling of the keyboard is under display program control. Before the keyboard can be used, it must be enabled. The A/N KBD indicator on the monitor indicates the enabled state to the operator. Depressing an alphanumeric/

edit key generates a manual interrupt and places an edit/alphanumeric key word in keyboard register (R_2), figure B3-17A. All keyboard operations are flexible with interpretation of each key code, table B3-28, determined by processor program.

Typical Application.

A typical alphanumeric keyboard application is composing a message on the crt and sending the completed message to the data source, figure 2-6.

With the keyboard enabled, each key depression generates a manual interrupt. By interrogating the manual input register, the processor interrupt routine identifies an a/n keyboard interrupt. After determining the a/n keyboard generated the manual interrupt, the interrupt routine interrogates the keyboard register.

If the keyboard register contains a SEND key code, the processor interrupts the data source. The data source then requests the data message from the Display Controller memory.

An edit key initiates the indicated editing function. The interrupt routine stores any other a/n key code in a message storage area in memory. After performing the required operation, the interrupt routines clears interrupt lockout and returns to the display routine. The display routine presents the operator message on the crt as it is composed.

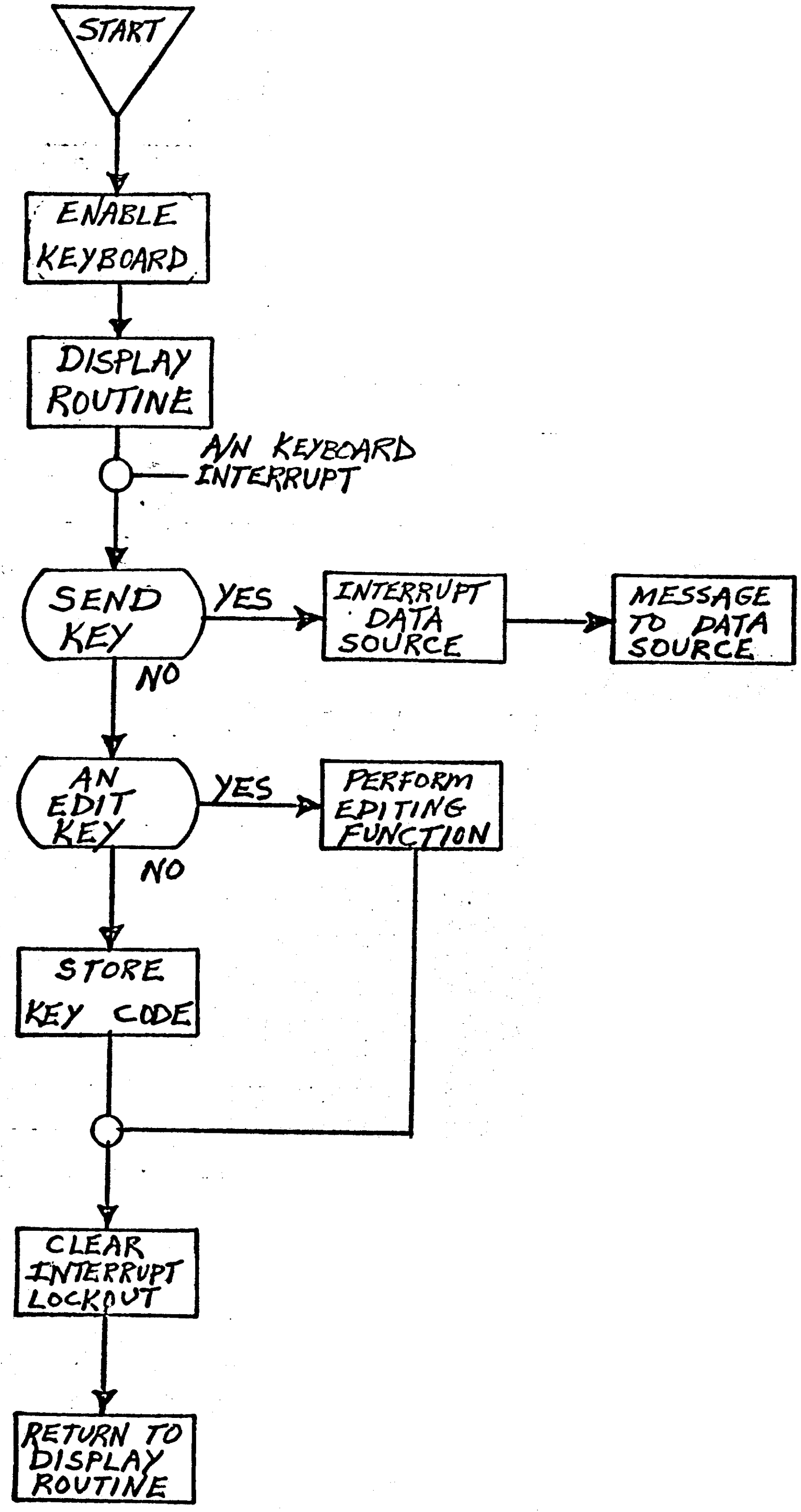


Figure 2-6. Typical A/N Keyboard Application

Function.

The function portion of the keyboard includes 10 function keys (F0 through F9), 4 status switches, and an INTER (interrupt) key. Function keys and status switches, in conjunction with the INTER key, serve as a data entry device.

Operation.

Before operation, the keyboard must be enabled. The A/N KBD indicator on the monitor indicates the enabled state to the operator.

Depressing a function key or INTER key generates a manual interrupt and sets up the keyboard register as shown in figure B3-18A. The processor program interprets function key codes, table B3-28, as desired.

Depressing a status switch locks it down, actuated position. The second depression of a status switch releases it. Any combination of status switches may be actuated simultaneously. The operator depresses the INTER key or function key to gererate a manual interrupt and enter the status code into the keyboard register, figure B3-18A. There are 16 possible status switch combinations which may be interpreted as desired by program.

Typical Application.

Status switches provide a means of selecting program subroutines. As an example, assume the operator wishes to erase a displayed symbol with the light pen. Via processor program, the operator selects an erase subroutine by depressing the proper combination of status switches and the INTER key. After erase subroutine selection, light penning a displayed symbol erases that symbol.

OPERATOR CONTROL PANEL.

The operator control panel, figure 2-7, interfaces between the operator and the display subsystem. The panel incorporates switches to control the mode of operation, switches for internal register selection, an operator keyboard for data entry, and an octal readout indicator for monitoring selected internal registers.

Register Selectors and Octal Readout Indicator.

The right bank of interlocking switch indicators designate by illumination the register being monitored. Depressing a switch indicator displays the contents of the respective register in the octal readout indicator, (reference table 2-2).

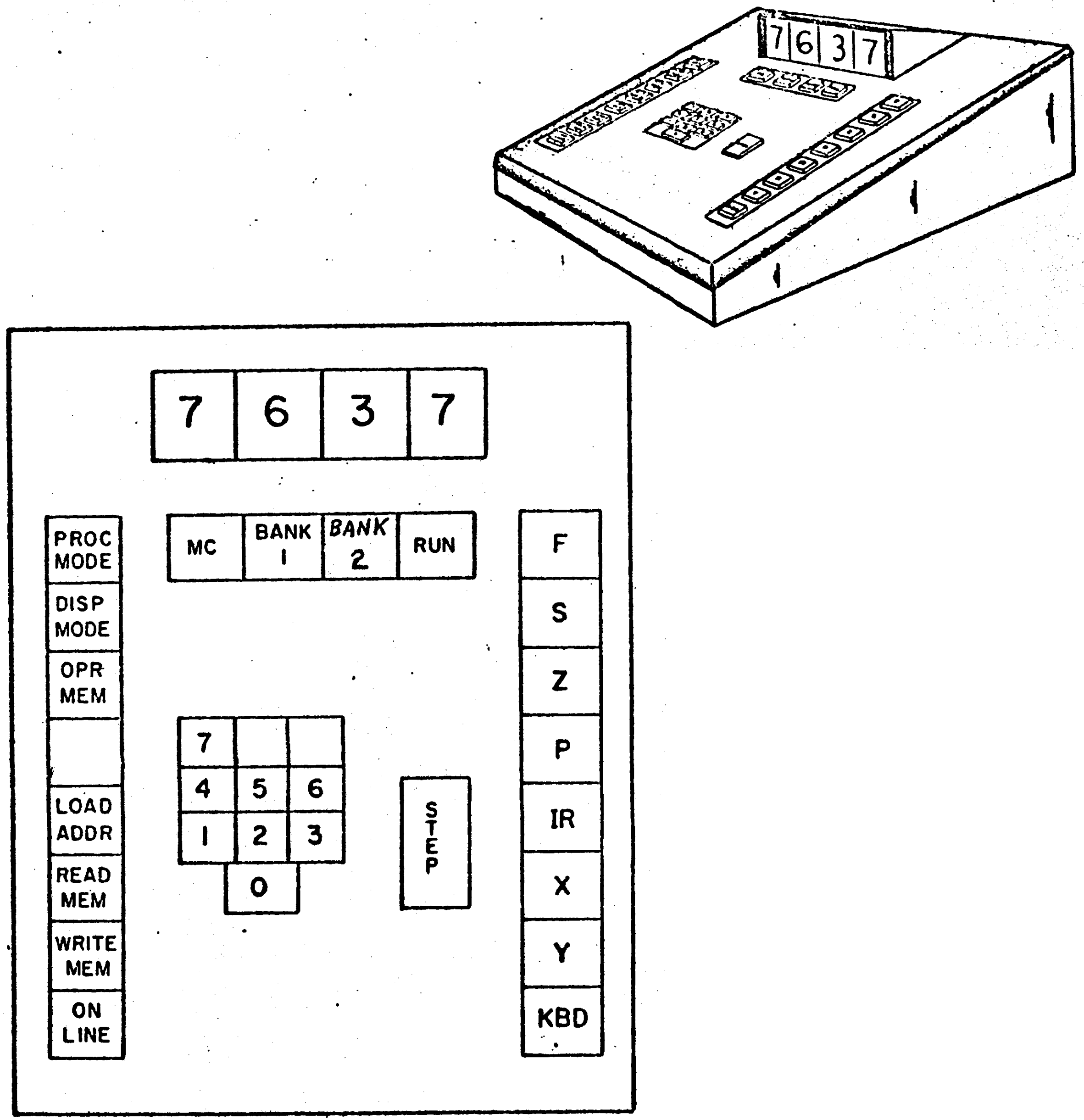


Figure 2-7. Operator Control Panel

TABLE 2-2. REGISTER SELECTORS

DESIGNATOR	DESCRIPTION
F	Displays the processor function register. It contains the processor function (instruction) to be executed or in the process of execution.
S	Displays the storage address register. It contains the storage address for the next memory reference.
Z	Displays the transfer register which holds data read from or written into memory during the last memory reference.
P	Displays the program address register which contains the address of the current program instruction.
IR	Displays the interface input register which contains the last data received from the data source.
X	Displays X register contents. When in processor mode, it contains results of arithmetic, logical, and shifting operations. When in display mode, it contains the horizontal (X) display position.
Y	Displays the Y register which contains the vertical (Y) display position.
KBD	Displays the keyboard register which contains data loaded by the numeric keyboard.

Operator Keyboard.

The operator keyboard is a switch actuated data entry device. Each time the operator depresses a numeric key, the octal code for that key enters the lowest octal position of the keyboard register. UNLABLED KEYS do not have a 3-digit octal code, AND do not function. With the keyboard register displayed, KBD switch depressed, the number associated with a depressed key appears in the lower octal of the readout indicator. Previous keyboard register contents shift left one octal position for each key depression. Four key depressions change the entire keyboard register contents.

In conjunction with the LOAD ADDR, WRITE MEM, and STEP switches, the operator keyboard enables storing data in memory by supplying the initial memory address and data for storage.

Control Switches.

The left and top bank of switches, together with the STEP switch, are a combination of interlocking and momentary contact switches. These switches enable operator control of the display subsystem. Table 2-3 describes the switches.

TABLE 2-3. CONTROL SWITCHES

NAME	SWITCH TYPE	FUNCTION
PROC MODE	Momentary Indicator	Selects processor mode and remains illuminated until selection of display mode. *
DISP MODE	Momentary Indicator	Selects display mode and remains illuminated until selection of processor mode. *
OPR MEM	Interlocking Indicator	When operating off-line, it must be depressed to enable selection of processor or display mode. Lights while depressed.
LOAD ADDR	Interlocking Indicator	Enables entering a 12-bit memory address via the operator keyboard. STEP transfers this address to the P and S registers. Lights while depressed.
READ MEM	Interlocking Indicator	Enables reading data from memory by depressing STEP or RUN. It is useful for checking programs stored in memory. Lights while depressed.
WRITE MEM	Interlocking Indicator	Enables storing keyboard data into memory by depressing STEP or RUN. It permits manual entry of program data. Lights while depressed.

* Operate only when OPR MEM is depressed.

TABLE 2-3. CONTROL SWITCHES (CONT)

NAME	SWITCH TYPE	FUNCTION
ON LINE	Interlocking Indicator	Selects the display subsystem operating status, on line or off line. When depressed, the subsystem is on line and the ON LINE indicator lights. On line status enables communication with a data source. Off line status enables the operator control panel. Depressing any one of the other interlocking left bank switches places the subsystem off line.
MC	Momentary	Master clears the display subsystem.
BANK 1	Momentary Indicator	Selects memory bank 1. Lights while bank 1 is selected.
BANK 2	Momentary Indicator	Selects memory bank 2. Lights while bank 2 is selected.
RUN	Momentary Indicator	Enables and disables high-speed program execution. When illuminated, high-speed program execution is enabled and vice versa.
STEP	Momentary	When off line, it initiates a complete operational cycle for each depression. This feature aids program debugging.

Operational Sequences.

Definite operational sequences permit the operator to write into memory, read from memory, start processor operations, and start display operations.

Write Memory.

Enter program data into memory with the following sequence of operations.

(a) MC (master clear).

(b) Depress the LOAD ADDR switch.

(c) Using the operator keyboard, enter the 12-bit starting address into the keyboard register.

(d) Depress the STEP switch.

(e) Select the memory bank for the write, if other than 0, by depressing the BANK 1 or BANK 2 switch.

(f) Depress the WRITE MEM switch.

(g) Using the operator keyboard, enter the desired 12-bit word into the keyboard register.

(h) Depress the STEP switch.

For continuous operation, repeat steps (g) and (h) until the entire program has been entered into memory. While writing memory, the following registers may be monitored by the operator.

- Keyboard register - indicates data entered from the operator keyboard.
- Z register - indicates the data written into memory at the last depression of the STEP switch.
- S register - indicates the next storage address.
- P register - indicates the last storage address.

To store the same data word thoughout memory, depress RUN instead of STEP.

Read Memory.

Read data from memory with the following sequence of operations.

(a) MC (master clear).

(b) Depress the LOAD ADDR switch.

(c) Using the operator keyboard, enter the 12-bit starting address into the keyboard register.

(d) Depress the STEP switch.

(e) Select the memory, bank for the read, if other than bank 0, by depressing the BANK 1 or BANK 2 switch.

(f) Depress the READ MEM switch.

(g) Depress the STEP switch.

Repeated depressions of the STEP switch results in reading memory data from sequential addresses. While reading memory, the following registers may be monitored on the octal readout indicator.

- Z register — indicates the contents of the address referenced on the last depression of the STEP switch.
- S register — indicates the next address to be referenced.
- P register — indicates the last address referenced.

Start Processor.

To initiate the processor mode of operation perform the following operations in the indicated sequence.

(a) MC (master clear).

(b) Depress the LOAD ADDR switch.

(c) Using the operator keyboard, enter the 12 bit starting address into the keyboard register.

(d) Depress the STEP switch.

(e) Select the memory bank for the processor operation, if other than bank 0, by depressing the BANK 1 or BANK 2 switch.

(f) Depress the OPR MEM switch.

(g) Depress the PROC MODE switch.

(h) For high-speed program execution, depress the RUN switch. For program debugging, operate the STEP switch.

When program debugging, each depression of the STEP switch causes the processor to perform one memory cycle. The number of memory cycles required to perform an instruction varies with the instruction code.

The following registers may be monitored by the operator on the octal readout indicator.

- Z register — indicates the contents of the address referenced on the last depression of the STEP switch.
- S register — indicates the next address to be referenced.
- P register — indicates the last address referenced.
- X register — indicates results of arithmetic, logical, and shifting operations.
- F register — displays instruction to be executed or in process of execution.

Start Display.

To initiate the display mode of operation, perform the following operations in the indicated sequence.

(a) MC (master clear).

(b) Depress the LOAD ADDR switch.

(c) Using the operator keyboard, enter the 12 bit starting address into the keyboard register.

(d) Depress the STEP switch.

(e) Select the memory bank for the display operation, if other than bank 0, by depressing the BANK1 or BANK 2 switch.

(f) Depress the OPR MEM switch.

(g) Depress the DISP MODE switch.

(h) For high-speed display operation, depress the RUN switch. For program debugging, operate the STEP switch.

The following registers may be monitored by the operator on the octal readout indicator.

- Z register — indicates the contents of the address referenced on the last depression of the STEP switch.
- S register — indicates the next address to be referenced.
- P register — indicates the last address referenced.
- X register — indicates the present X beam position.
- Y register — indicates the present Y beam position.

INTERFACE.

The subsystem Display Controller interface module is compatible with and controls communication with a standard CONTROL DATA 3000 Series Data Channel. Refer to the associated 3000 Series Reference Manual for detailed input/output programming information.

DATA TRANSFER.

Transfer of 12-bit data bytes between the Display Controller memory and data channel is bidirectional. A write address function selects memory addresses for data storage and retrieval. Odd parity information accompanies each data byte.

If a parity error occurs on data during a write operation (data to subsystem), the following occurs.

(a) Subsystem transmits Parity Error signal and Reply.

(b) PARITY ERROR indicator on the monitor cabinet lights. It remains on until receipt of a Master Clear signal or depression of the operator panel MC switch.

(c) Subsystem stores the data in memory.

CONNECT.

The subsystem interprets the connect code, figure A3-1, received from the data channel as follows.

(a) Checks bits 11 through 9 for a matching EQUIPMENT SELECTOR switch number.

(b) Bits 8 through 0 are not translated.

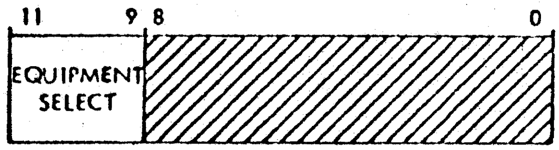


Figure A3-1. Connect Code Format

Reply and Reject signals are the possible connect code responses. A parity error results in no response.

Reply.

The subsystem connects and transmits a Reply signal if the following conditions exist.

(a) Power is on.

(b) The connect code and EQUIPMENT SELECTOR switch setting match.

(c) The subsystem is on-line.

(d) No transmission parity error.

Reject.

The subsystem does not connect and transmits a Reject signal if the subsystem is off-line. All other conditions are the same as for a Reply. Status lines are enabled until receipt of the next Connect signal.

Parity Error.

A connect code parity error results in the following.

(a) The subsystem does not connect; if connected, it disconnects.

(b) The subsystem does not return a Reply, Reject, or Parity Error signal.

(c) The PARITY ERROR indicator on the monitor cabinet lights. It remains on until receipt of a Master Clear signal or depression of the operator panel MC switch.

FUNCTIONS.

The data source initiates display subsystem functions with 12-bit function codes. Table A3-1 lists the function codes recognized by the subsystem. The subsystem interprets a function code, figure A3-2, as follows.

11					6	5					0
0	0	0	0	0	0	X	X	X	X	X	X

Figure A3-2. Function Code Format

(a) Translates bits 5 through 0 to initiate the function.

(b) Bits 11 through 6 must equal 0's or the function code translates as an illegal code.

Reply, Reject, and Parity Error signals are the possible responses to a function code.

Reply.

If the specified function can be performed, the subsystem initiates the function and transmits a Reply. The accepted function locks out all other functions until completion. All the following conditions enable a Reply.

(a) Display subsystem is connected.

(b) No transmission parity error exists.

(c) Function code can be performed.

Reject.

If the specified function cannot be performed, the subsystem ignores the function code and transmits a Reject. With the exception of being unable to perform the function, Reject and Reply subsystem conditions are identical.

Parity Error.

If a parity error exists in a function code and the subsystem is connected, the following occur.

(a) The subsystem returns a Parity Error signal in lieu of a Reply or Reject.

(b) The PARITY ERROR indicator on the monitor cabinet lights. It remains on until receipt of a Master Clear signal or depression of the operator MC switch.

(c) The subsystem does not perform the function.

TABLE A3-1. FUNCTION CODES

CODE (OCTAL)	FUNCTION	DESCRIPTION
0000	Release and Disconnect	Disconnects the display subsystem and clears the interface module logic.
0020	Select Interrupt on Ready and Not Busy	Enables starting another i/o operation when the subsystem is on line, the Channel Busy line is inactive, and peripheral i/o is inactive. Clears previously generated interrupt on read and not busy.
0021	Clear Interrupt on Ready and Not Busy	Clears interrupt on ready and not busy selection and any resultant interrupt.
0022	Select Interrupt on End of Operation	Enables an interrupt upon termination of a read, write, or peripheral i/o operation or when the processor encounters a halt instruction. Clears previously generated interrupt on end of operation.
0023	Clear Interrupt on End of Operation	Clears interrupt on end of operation selection and any resultant interrupt.
0024	Select Interrupt on Abnormal End of Operation	Enables an interrupt when the processor encounters an error instruction (0000_8). Clears previously generated interrupt on abnormal end of operation. operation.
0025	Clear Interrupt on Abnormal End of Operation	Clears the interrupt on abnormal end of operation selection and any resulting interrupt.

TABLE A3-1. FUNCTION CODES (CONT)

CODE (OCTAL)	FUNCTION	DESCRIPTION
0026	Select Processor Interrupt	Enables an interrupt when the processor encounters an interrupt data source instruction (0122_8). Clears previously generated processor interrupt.
0027	Clear Processor Interrupt	Clears the processor interrupt selection and any resulting interrupts.
0040	Write Address	Directs the next two write data words to the storage address(S) register. The first data word contains the lower 12 bits of the address. The lowest octal of the second data word selects memory bank 0, 1, or 2 (binary 010=2, etc). Processor or display operation halts. The program address (P) register remains unaltered.
0041	Write Memory	Directs succeeding data words from the data channel to consecutive memory addresses. Normally, a write address function sets up the starting address prior to the write memory function. Processor or display operation halts. The storage address(S) register increments automatically. The program address (P) register remains unaltered.

TABLE A3-1. FUNCTION CODES (CONT)

CODE (OCTAL)	FUNCTION	DESCRIPTION
0042	Read Memory	Directs data from consecutive memory addresses to the data channel. Normally a write address function precedes a read memory function, setting up the initial address. Processor or display operation halts. The storage address(S) register increments automatically. The program address (P) register remains unaltered.
0050	Start Processor	Starts processor operation at the address in the S register.
0051	Start Display	Starts display operation at the address in the S register.
0052	Resume	Transfers the P register to the S register and directs operation to resume in whichever mode (display or processor) was operating last. This enables resuming a display or processor program that was stopped by a read or write i/o operation.

SUBSYSTEM STATUS.

When connected, the subsystem indicates its operating status on 12 Channel Status lines. Table A3-2 lists the status bits.

TABLE A3-2. STATUS BITS

BIT POSITION	NAME	DESCRIPTION
2^{0}	Ready	On line.
2^{1}	Busy	Connected and channel busy or peripheral i/o busy.
2^{2}	Processor Active	Processor active or last active mode of operation.
2^{3}	Display Active	Display active or last active mode of operation.
2^{4}	Not Used	
2^{5}	Not Used	
2^{6}	Processor Interrupt	Processor interrupt occurred.
2^{7}	Ready and Not Busy Interrupt	Ready and not busy interrupt occurred.
2^{8}	End of Operation Interrupt	Processor encountered a halt instruction (77_8).
2^{9}	Abnormal End of Operation Interrupt.	Processor encountered an error instruction (0000_8), read operation terminated, write operation terminated, or perophera l i/o operation terminated.
2^{10}	Not Used	
2^{11}	Not Used	

SECTION III

PROGRAMMING

PART B

Part B of this section provides arithmetic, memory, processor repertoire, display repertoire, and symbol repertoire information. Programming hints and examples encompass and clarify overall programming of the subsystem.

ARITHMETIC.

A processor word contains 12 binary digits. Figure B3-1 shows bit position numbering, binary representation of a 12-bit quantity, and octal equivalent of that quantity. All arithmetic is binary, one's complement notation. Although the processor operates in binary, processor words are represented as four octal digits for programming convenience.

<table>
<tr><td>11</td><td>10</td><td>09</td><td>08</td><td>07</td><td>06</td><td>05</td><td>04</td><td>03</td><td>02</td><td>01</td><td>00</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td colspan="3">3</td><td colspan="3">6</td><td colspan="3">1</td><td colspan="3">3</td></tr>
</table>

Figure B3-1. Processor Word Representation

In one's complement notation, positive numbers are represented by their binary equivalent; negative numbers are represented by the one's complement of the equivalent positive number. Form the one's complement by reversing each bit of a word.

Example:

+5 = 000 000 000 101 (binary) = 0005 (octal)

-5 = 111 111 111 010 (binary) = 7772 (octal)

The arithmetic module performs all arithmetic operations by subtraction. Addition results from subtracting the complement of the addend from the augend. Subtraction does not require complementing. Arithmetic is modulus $2^{12} - 1$. That is, a borrow generated from the high order end of a 12-bit word subtracts from the low order end to provide the correct result (end-around-borrow).

Zero may be represented in one's complement notation in either of two ways:

000 000 000 000 (plus zero)

111 111 111 111 (minus zero)

Both plus and minus zero are acceptable as arithmetic operands. There are only two cases in which a zero arithmetic result will be minus zero; in all other cases, the result will be plus zero. These two cases are:

$$-0 + (-0) \text{ and } -0 - (+0)$$

The programmer may interpret and use arithmetic operands and results of arithmetic operations as desired. For example, he may or may not treat bit 11 as a sign bit, depending upon program requirements. Treating bit 11 as a sign bit limits operands to a maximum value of $+3777_8$ and a minimum value of -3777_8. The limit for unsigned operands is 7777_8.

MEMORY.

Memory provides magnetic core storage for 12-bit display and processor information words. The interface, processor, display, and operator all have access to memory on a time sharing basis.

MEMORY REGISTER.

There are three registers associated with memory: P, S, and Z.

P - 12-bit program address register. It contains the address of the last memory reference.

S - 12-bit storage address register. It contains the next reference address. Upon completion of each memory cycle, the S register automatically advances 1.

Z - 12-bit transfer register which holds data being written into or read from memory.

BANK CONTROL.

When the Display Controller contains three memory banks of 4096, 12-bit words each, bank control becomes necessary. Each bank has $10{,}000_8$ addresses numbered from 0000_8 to 7777_8. Bank control allows independent selection of banks for relative, direct, and indirect addressing. When the Display Controller contains only bank 0, relative, direct, and indirect references to other than bank 0 yield instructions and operands of 7777_8.

Relative Bank Control (r).

Reference memory bank control instructions for selection of the relative (r) storage bank. All instructions are in the relative bank. Also, instructions indicating relative or constant addressing reference the relative bank. The relative bank also contains the first address for memory instructions.

A change in relative bank initiates a processor jump. Reference the set memory bank controls instruction in the processor repertoire.

Direct Bank Control (d).

Reference memory bank control instructions for selection of the storage bank for reference by all instructions having direct addressing operation codes. For operation codes indicating indirect addressing, (d) selects the direct bank containing the indirect address.

Indirect Bank Control (i).

Reference the memory bank control instructions for selection of the indirect (i) storage bank. When instructions have indirect or memory addressing operation codes, control logic references the indirect bank for final operands. Because jump instructions cause an indirect transfer of control, they are exceptions. JPI and JFI transfer control within the relative bank.

PROCESSOR INSTRUCTION REPERTOIRE.

While in the processor mode of operation (activated by manual interrupt, interface function code, operator panel, or display jump instruction), a program stored in memory controls processor operation. The stored program may be located in bank 0, bank 1, or bank 2.

Utilizing memory, manual input, and arithmetic modules, processor logic performs instructions similar to those of a CONTROL DATA 160-A computer. The 12-bit X register in the arithmetic module receives the results of all arithmetic, logical, and shifting operations.

WORD FORMAT.

A processor instruction word divides into a 6-bit function code (f) and a 6-bit execution address (E). Most instructions require only one word of storage but expanded instructions occupy two words. Figure B3-2 shows the processor instruction word formats. The first word contains a 6-bit function code and a 6-bit execution address field which always equals zero in two-word instructions. The second word contains a 12-bit address or operand (G), depending on the instruction. Both words 1 and 2 must be located in sequential storage addresses in the same memory bank.

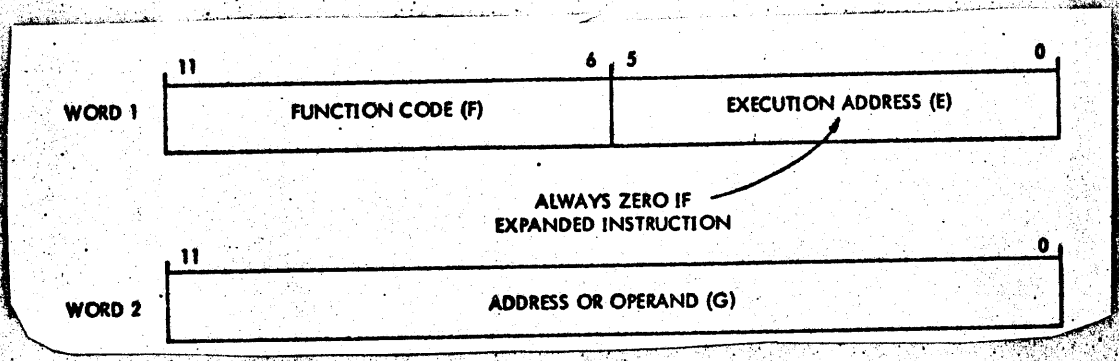


Figure B3-2. Processor Instruction Word Formats

ADDRESS MODES.

Seven memory address modes listed in table B3-1 provide maximum 6-bit address flexibility.

TABLE B3-1. MEMORY ADDRESS MODES

ADDRESS MODE	DESCRIPTION
No Address (N)	When the E portion of a processor instruction is a 6-bit operand, the instruction initiates arithmetic and logical operations using this 6-bit operand as a constant. By combining constant and instruction, this mode conserves memory locations.
Direct Address (D)	Refers to a 12-bit operand in one of the first 100$_8$ memory locations in the direct memory bank.
Indirect Address (I)	Provides for operand references and jump addresses. For processor instructions employing indirect addressing, E refers to one of 77$_8$ of the first 100$_8$ memory locations starting at location 0001 of the direct storage bank. The contents of this address become the address of the operand or become the jump address. address.
Relative Address Forward (F)	Generates operand or jump address by adding E to the current contents of P. This specifies one of 77$_8$ addresses immediately following the address of the current processor instruction.
Relative Address Backward (B)	Generates operand or jump address by subtracting E from the current contents of P. This specifies one of 77$_8$ addresses immediately preceding the address of the current processor instruction.
Constant Address (C)*	The G portion of a 24-bit processor instruction contains the operand.
Memory Address (M)*	The G portion of a 24-bit processor instruction contains the address of the operand.

*This mode uses two sequential storage locations, designation for the second location is G. In this mode, E must always equal zero.

No Address Mode (N).

In no address mode, E is the lower 6 bits of an implied 12-bit operand. The upper 6 bits of the operand always equal zero. Thus, the E portion of the instruction word becomes the operand.

Example:

Location	F	E
(r)0100	LDN	43 (load no address)
(r)0101	next instruction	

A load instruction transmits the operand to the X register. The 12-bit operand for LDN 43 is 0043.* The number 0043 goes to the X register. At the completion of a no address (N) instruction, control continues at the location in the relative storage bank (r) specified by the contents of P + 1. In this case, control continues at location (r) 101.

Direct Address Mode (D).

In the direct address mode, E selects one of the first 100_8 locations in the direct storage bank (d) as the operand address.

Example:

Location	F	E
(d)0076	12	34
(r)1075	LDD	76 (load direct)
(r)1076	next instruction	

*All numbers in example programs are octal unless stated otherwise.

E specifies the operand address as (d)0076. This address contains the quantity 1234 which will be transferred to the X register. At the completion of a direct address (d) instruction, control continues at the location in the relative storage bank specified by the contents of P + 1. In this case, control contunues at (r)1076.

Indirect Address Mode (I).

In indirect address mode, E references 77_8 of the first 100_8 locations, starting at location 0001 of the direct storage bank (d). The location (d)00E is then referenced and the contents of (d)00E become the operand address in the indirect bank (i).

Example:

Location	F	E
(d)0045	33	65
(i)3365	46	57
(r)4121	LDI	45 (load indirect)
(r)4122	next instruction	

E calls for referencing (d)0045, which contains the address 3365. A final reference is now made to (i)3365, which contains the number 4657. The quantity 4657 goes to the X register. Notice that both the direct (d) and indirect (i) storage bank controls are involved in the indirect address mode. At completion of an (i) instruction, control continues at the location in the relative storage bank specified by the contents of P + 1. In the above example, control contunues at (r)4122.

There are two (I) instructions which are exceptions to the above rules, JPI and JFI:

(a) JPI initially references location (d)00E. A transfer of control then takes place within the relative (r) bank to the location specified by the contents of (d)00E.

(b) JFI initial reference is relative forward. A transfer of control then takes place within the relative (r) bank to the address specified in the relative forward reference.

Relative Forward Address Mode (F).

In relative forward address mode, E adds to the contents of the P register. This sum becomes the effective operand address in the relative storage bank (r).

	Location	F	E
Example:			
	(r)0233	LDF	22 (load forward)
	(r)0234	next instruction	
	(r)0255	77	03

Adding E to the P register yields address (r)0255. The contents of (r)0255 go to the X register. At completion of this instruction, X contains 7703. At completion of an (F) instruction which does not cause transfer of control, control continues in the relative storage bank (r) at the location specified by contents of P + 1. In the above example, control continues at location (r)0234. Forward Jump instructions transfer control E locations forward in the relative bank.

Relative Backward Address Mode (B).

The relative backward address mode functions similar to relative forward (F) mode. In relative backward address mode, subtraction of (E) from the contents of the P register forms an address in the relative storage bank.

Constant Address Mode (C).

All constant address mode instructions occupy two sequential storage locations. The G portion of the 24-bit instruction word contains the operand. E always equals zero.

Example:

Location	F	E	G	
(r)0101	LDC	00		(load constant)
			7337	
(r)0103	STC	00		(store constant)
			2345	
(r)0105	next instruction	00		

At location (r)0101 is a load constant (LDC) instruction. The operand address is (r)0102. The quantity 7337goes to the X register. Upon completion of a (C) instruction, control continues in the relative storage bank (r) at the location specified by the contents of P + 2. In this case, control continues at (r)0103.

This address contains a store constant (STC) instruction. This transfers X register contents to the operand address. In the above example, the operand address of the STC instruction is (r)0104. The quantity 7337, in the X register as a result of the LDC instruction in (r)0101, goes to location (r)0104, replacing the constant 2345 now in (r) 0104. Final contents of (r)0104 are 7337. Control continues at (r)0105.

Memory Address Mode (M).

All memory address mode instructions occupy two sequential storage locations. The G portion of the 24-bit instruction word contains the address of the operand. E always equals zero.

Example:	Location	F	E	G	
	(r)3477	LDM	00	1111	(load memory)
	(r)3501	STM		0024	(store memory
	(r)3503	next instruction			
	(i)1111	67	66		
	(i)0024	02	34		

Location (r)3477 contains a load memory (LDM) instruction. The location (i)1111 becomes the operand address. The quantity 6766 goes to the X register. Upon completion of an (M) instruction, control continues in the relative storage bank (r) at the location specified by the contents of P + 2. In this case, control continues at location (r)3501 which contains a store memory (STM) instruction~~s~~. The operand address of this instruction becomes (i)0024. This stores X register quantity 6766 in location (i)0024, replacing quantity 0234. Control continues at location (r)3503.

INSTRUCTIONS.

Table B3-2 lists the processor instructions and execution times. For explanations of instructions that follow, these points apply.

(a) All numbers are in octal notation unless stated otherwise.

(b) In operation code descriptions, operations appear in sequence.

(c) Instruction descriptions are by general function rather than numerical order. Instructions which may assume more than one address mode appear under the general function.

(d) A description lists the G portion of an instruction only with instructions occupying two storage words.

TABLE B3-2. PROCESSOR REPERTOIRE AND EXECUTION TIMES

OPERATION	MNEMONIC	CODE	MODE	TIME (microseconds)
ERROR STOP	ERR	0000	n	2.4
NO OPERATION	NOP	000X	n	2.4
MEMORY BANK CONTROLS	SRJ	001X	n	2.4
	SIC	002X	n	2.4
	IRJ	003X	n	2.4
	SDC	004X	n	2.4
	DRJ	005X	n	2.4
	SID	006X	n	2.4
	ACJ	007X	n	2.4
MEMORY BANK CONTROL TO X	CTX	0130	n	2.4
TRANSFER INTERNAL REGISTERS	STX	0100	n	2.4
	PTX	0101	n	2.4
	YTX	0170	n	2.4
	RTX	0171	n	2.4
	RMX	0172	n	2.4
	*RSX	0173	n	2.4
	XTY	0174	n	2.4
	*XTA	0175	n	2.4
	*XMA	0176	n	2.4
	*XTR	0177	n	2.4
LEFT SHIFT ONE	LS1	0102	n	2.4
LEFT SHIFT THREE	LS3	0110	n	2.4
LEFT SHIFT SIX	LS6	0111	n	2.4
MULTIPLY BY 12	MUT	0112	n	2.4
CLEAR INTERRUPT LOCKOUT	CIL	0120	n	2.4
SET INTERRUPT LOCKOUT	SIL	0121	n	2.4
INTERRUPT DATA SOURCE	**IDS	0122	n	2.4
LOGICAL PRODUCT	LPN	02XX	n	2.4
	LPD	10YY	d	4.8
	LPM	1100	m	6.8
	LPI	11YY	i	6.8
	LPC	1200	c	4.8
	LPF	12XX	f	5.6
	LPB	13XX	b	5.6
SELECTIVE COMPLEMENT	SCN	03XX	n	2.4
	SCD	14YY	d	4.8
	SCM	1500	m	6.8
	SCI	15YY	i	6.8
	SCC	1600	c	4.8
	SCF	16XX	f	5.6
	SCB	17XX	b	5.6
LOAD	LDN	04XX	n	2.4
	LDD	20YY	d	4.8
	LDM	2100	m	6.8
	LDI	21YY	i	6.8
	LDC	2200	c	4.8
	LDF	22XX	f	5.6
	LDB	23XX	b	5.6
LOAD COMPLEMENT	LCN	05XX	n	2.4
	LCD	24YY	d	4.8
	LCM	2500	m	6.8
	LCI	25YY	i	6.8
	LCC	2600	c	4.8
	LCF	26XX	f	5.6
	LCB	27XX	b	5.6
ADD	ADN	06XX	n	2.4
	ADD	30YY	d	4.8
	ADM	3100	m	6.8
	ADI	31YY	i	6.8
	ADC	3200	c	4.8
	ADF	32XX	f	5.6
	ADB	33XX	b	5.6
SUBTRACT	SBN	07XX	n	2.4
	SBD	34YY	d	4.8
	SBM	3500	m	6.8
	SBI	35YY	i	6.8
	SBC	3600	c	4.8
	SBF	36XX	f	5.6
	SBB	37XX	b	5.6
STORE	STD	40YY	d	4.8
	STM	4100	m	6.8
	STI	41YY	i	6.8
	STC	4200	c	4.8
	STF	42XX	f	5.6
	STB	43XX	b	5.6
LEFT SHIFT 1 AND REPLACE	SRD	44YY	d	5.6
	SRM	4500	m	9.2
	SRI	45YY	i	9.2
	SRC	4600	c	7.2
	SRF	46XX	f	8.0
	SRB	47XX	b	8.0
REPLACE AND ADD	RAD	50YY	d	7.2
	RAM	5100	m	9.2
	RAI	51YY	i	9.2
	RAC	5200	c	7.2
	RAF	52XX	f	8.0
	RAB	53XX	b	8.0
REPLACE AND ADD 1	AOD	54YY	d	7.2
	AOM	5500	m	9.2
	AOI	55YY	i	9.2
	AOC	5600	c	7.2
	AOF	56XX	f	8.0
	AOB	57XX	b	8.0
ZERO JUMP	ZJF	60XX	f	3.2
	ZJB	64XX	b	3.2
NONZERO JUMP	NZF	61XX	f	3.2
	NZB	65XX	b	3.2
POSITIVE JUMP	PJF	62XX	f	3.2
	PJB	66XX	b	3.2
NEGATIVE JUMP	NJF	63XX	f	3.2
	NJB	67XX	b	3.2
JUMP INDIRECT	JPI	70YY	d	4.4
RETURN JUMP	JPR	7100	m	6.8
JUMP FORWARD INDIRECT	JFI	71XX	fi	5.2
JUMP TO DISPLAY	JPD	74XX	m	4.4
INPUT/OUTPUT	INP	72XX	fi	
	OUT	73XX	fi	
	EXC	7500	m	
	EXF	75XX	n	
	INX	7600	m	
	OTX	7677	n	
HALT	HLT	77XX	n	2.4

* Applicable on remote subsystems only.
** Not applicable on remote subsystems.

(e) When the Display Controller contains only one memory bank, only memory bank 0 can be referenced. Reference to a bank other than 0 yields operands and instructions of 7777_8.

(f) XXXX represents an octal operand.

(g) YYYY represents an octal address.

STOP INSTRUCTIONS.

Stop instructions, table B3-3, stop the processor unconditionally. When the processor encounters an ERR or HLT instruction, computation stops and, if enabled, interrupts the data channel.

TABLE B3-3. STOP INSTRUCTIONS

F	E	MNEMONIC	NAME
00	00	ERR	Error Stop
77	XX	HLT	Halt

NO OPERATION INSTRUCTION.

A no operation instruction (NOP), octal code 000X, does not perform any function. Program control passes to the next instruction at location (r)(P)+1.

DATA TRANSMISSION INSTRUCTIONS.

Data transmission instructions enable transferring data between the X register and memory, and between the X register and other selected registers.

Load Instructions.

Load instructions, table B3-4, transfer an operand to the X register. Execution of a load instruction does not alter the operand in storage. After transferring the operand to X for each address mode, control continues as described under address modes.

NOTE

The E portion of LDI and LDF instructions must not be zero. If zero, the processor interprets the operation code as LDM or LDC, respectively.

TABLE B3-4. LOAD INSTRUCTIONS

F	E	G	MNEMONIC	NAME
04	XX		LDN	Load No Address
20	YY		LDD	Load Direct
21	00	YYYY	LDM	Load Memory
21	YY		LDI	Load Indirect
22	00	XXXX	LDC	Load Constant
22	XX		LDF	Load Forward
23	XX		LDB	Load Backward

Description: operand → X

Load Complement Instructions.

Load complement instructions, table B3-5, transfer the one's complement of the operand to X. A load complement instruction does not alter the operand in storage. After transferring the one's complement of the operand to X for each address mode, control continues as described under address modes.

NOTE

The E portion of LCI and LCF instructions must not be zero. If zero, the processor interprets the operation code as LCM or LCC, respectively.

TABLE B3-5. LOAD COMPLEMENT INSTRUCTIONS

F	E	G	MNEMONIC	NAME
05	XX		LCN	Load Complement No Address
24	YY		LCD	Load Complement Direct
25	00	YYYY	LCM	Load Complement Memory
25	YY		LCI	Load Complement Indirect
26	00	XXX	LCC	Load Complement Constant
26	XX		LCF	Load Complement Forward
27	XX		LCB	Load Complement Backward

Description: $\overline{\text{operand}} \rightarrow X$

Store Instructions.

Store instructions, table B3-6, transfer an operand from the X register to the operand memory address. Execution of a store instruction does not alter contents of the X register. After storing the operand for each address mode, control continues as described under address modes.

NOTE

> The E portion of STI and STF instructions must not be zero. If zero, the processor interprets the operation code as STM or STC, respectively.

TABLE B3-6. STORE INSTRUCTIONS

F	E	G	MNEMONIC	NAME
40	YY		STD	Store Direct
41	00	YYYY	STM	Store Memory
41	YY		STI	Store Indirect
42	00	XXXX	STC	Store Constant
42	XX		STF	Store Forward
43	XX		STB	Store Backward
Description: (X)→operand address				

Transfer Internal Registers.

The transfer internal register instructions (table B3-7) enable saving, modifying, and sampling the contents of the indicated registers.

TABLE B3-7. TRANSFER INTERNAL REGISTERS

F	E	MNEMONIC	FUNCTION
01	00	STX	State → X
01	01	PTA	P → X
01	70	YTX	Y → X
01	71	RTX	R_1 → X
01	72	RMX	R_2 → X
*01	73	RSX	Remote Status → X
01	74	XTY	X → Y
*01	75	XTA	X → A_1
*01	76	XMA	X → A_2
*01	77	XTR	X → Remote Control

*Applicable on remote systems only

Example:

Location	F	E	
1313	01	72	
(R_2)	00	61	
(X)	77	57	(original)
(X)	00	61	(final)

Transfer Bank Controls.

The transfer bank controls to X register (CTA) instruction enables sampling the current memory bank selections. Figure B3-2A is an octal breakdown of the X register after executing a CTA instruction. Each of the four octals in X designates the number of its respective memory bank. For example, if memory bank 2 is the relative bank, the binary equivalent of the lowest octal in X is 010. Storage bank controls remain unchanged and program control continues at (r)(P)+1.

11 9	8 6	5 3	2 0
DISPLAY BANK	INDIRECT BANK	DIRECT BANK	RELATIVE BANK

Figure B3-2A. X Register Breakdown

ARITHMETIC INSTRUCTIONS.

Arithmetic instructions include multiply, subtract, replace add, and replace add one instructions.

Multiply.

The multiply instruction, table B3-9, multiplies the contents of X (multiplicand) by 12_8 (multiplier). The product appears in X as an 11-bit signed number or a 12-bit unsigned number, depending on the value of the multiplicand. For a multiplicand range of -314_8 to $+314_8$, the product is algebraically correct. The upper bit of X is the sign. A 1 indicates a negative number and 0 indicates a positive number. If the contents of X are $> +314_8$ or $< -314_8$ before multiplication, the magnitude of the product is correct modulus $2^{12}-1$. That is, the product becomes a 12-bit unsigned number. The maximum multiplicand for a meaningful product is 631_8.

TABLE B3-9. MULTIPLY INSTRUCTION

F	E	MNEMONIC	NAME
01	12	MUT	Multiply X by 12_8
Description: 12_8 (X) → X			

Example: Multiply $+314_8$ by 12_8.

Original (X) = 0314

Multiplier = 0012

Final (X) = 3770 = $+3770_8$

Example: Multiply $+315_8$ by 12_8.

Original (X) = 0315

Multiplier = 0012

Final (X) = 4002 = [4002]

Add Instructions.

Add instructions, table B3-9A, place in X the sum of the original contents of X and the operand. The operand remains unchanged. Control continues as described under address modes.

NOTE

> The E portion of ADI and ADF instructions must not be zero. If zero, the processor interprets the operation code as ADM or ADC, respectively.

TABLE B3-9A. ADD INSTRUCTIONS

F	E	G	MNEMONIC	NAME
06	XX		ADN	Add no Address
30	YY		ADD	Add Direct
31	00	YYYY	ADM	Add Memory
31	YY		ADI	Add Indirect
32	00	XXXX	ADC	Add Contant
32	XX		ADF	Add Forward
33	XX		ADB	Add Backward

Description: (X) +operand → X

Subtract Instructions.

Subtract instructions, table B3-10, place in X the difference between the original contents of X and the operand. The operand remains unchanged. Control continues as described under address modes.

NOTE

The E portion of SBI and SBF instructions must not not be zero. If zero, the processor interprets the operation code as SBM or SBC, respectively.

TABLE B3-10. SUBTRACT INSTRUCTIONS

F	E	G	MNEMONIC	NAME
07	XX		SBN	Subtract No Address
34	YY		SBD	Subtract Direct
35	00	YYYY	SBM	Subtract Memory
35	YY		SBI	Subtract Indirect
36	00	XXXX	SBC	Subtract Constant
36	XX		SBF	Subtract Forward
37	XX		SBB	Subtract Backward
Description: (X) - operand → X				

<u>Replace Add Instructions.</u>

Replace add instructions, table B3-11, sum the contents of X with an operand. Upon completion, both the operand address and X contain the new sum. Control continues as described under address modes.

NOTE

The E portion of RAI and RAF instructions must not be zero. If zero, the processor interprets the operation code as RAM or RAC, respectively.

TABLE B3-11. REPLACE ADD INSTRUCTIONS

F	E	G	MNEMONIC	NAME
50	YY		RAD	Replace Add Direct
51	00	YYYY	RAM	Replace Add Memory
51	YY		RAI	Replace Add Indirect
52	00	XXXX	RAC	Replace Add Constant
52	XX		RAF	Replace Add Forward
53	XX		RAB	Replace Add Backward

Description: (X) + operand →X
(X) →operand address

Replace Add One Instructions.

Replace add one instructions, table B3-12, form the sum of the operand plus one in X and transfer this sum to the operand address. Upon completion, both the operand address and X contain the original operand increased by 1. Control continues as described under address modes.

NOTE

The E portion of AOI and AOF instructions must not be zero. If zero, the processor interprets the operation code as AOM or AOC, respectively.

TABLE B3-12. REPLACE ADD ONE INSTRUCTIONS

F	E	G	MNEMONIC	NAME
54	YY		AOD	Replace Add One Direct
55	00	YYYY	AOM	Replace Add One Memory
55	YY		AOI	Replace Add One Indirect
56	00	XXXX	AOC	Replace Add One Constant
56	XX		AOF	Replace Add One Forward
57	XX		AOB	Replace Add One Backward

Description: operand → X
(X) + 1 → X
(X) → operand address

SHIFT INSTRUCTIONS.

There are two groups of shift instructions, shift and shift replace.

Shift.

Shift instructions, table B3-13, perform an end around (circular) left shift on information in the X register. Bits shifted out of bit position 11 move to bit position 00. From bit position 00, bits shift into bit position 01, etc. Control continues at location (r) (P) + 1.

TABLE B3-13. SHIFT INSTRUCTIONS

F	E	MNEMONIC	NAME
01	02	LS1	Left Shift One
01	10	LS3	Left Shift Three
01	11	LS6	Left Shift Six

Description: Shift X left the number of bit positions specified.

Example:

Location	F	E
(r)6173	LS3	
(X)	61	02

E is not specified since all the shift instructions use E as part of the operation code. X contains 6102. At completion of LS3, X contains 1026 and control continues at (r)6174.

Shift Replace.

Execution of a replace shift instruction, table B3-14, transfers the operand to X, shifts it left (circular) one bit position, and transfers the contents of X back to the operand address. After completion of a shift repl ace, both X and the operand address contain the shifted operand. Control continues as described under address modes.

NOTE

The E portion of SRI and SRF instructions must not be zero. If zero, the processor interprets the operation code as SRM or SRC, respectively.

TABLE B3-14. SHIFT REPLACE INSTRUCTIONS

F	E	G	MNEMONIC	NAME
44	YY		SRD	Shift Replace Direct
45	00	YYYY	SRM	Shift Replace Memory
45	YY		SRI	Shift Replace Indirect
46	00	XXXX	SRC	Shift Replace Constant
46	XX		SRF	Shift Replace Forward
47	XX		SRB	Shift Replace Backward

Description: operand —> X
shift X left circular 1 bit position
(X) —> operand address

LOGICAL INSTRUCTIONS.

Logical instructions complement or extract any portion of the X register according to a designated bit pattern.

Logical Product Instructions.

Logical product instructions, (table B3-15) form, in X, the logical product of the operand and the original contents of X. Using the operand as a mask, selected portions of X may be cleared or retained. The operand in storage remains unchanged.

Example:

(X) (binary)	111 010 001 101
operand (binary)	010 010 000 101
logical product (binary)	010 010 000 101

Control continues as described under address modes.

NOTE

The E portion of LPI and LPF instructions must not be zero. If zero, the processor interprets the operation code as LPM or LPC, respectively.

TABLE B3-15. LOGICAL PRODUCT INSTRUCTIONS

F	E	G	MNEMONIC	NAME
02	XX		LPN	Logical Product No Address
10	YY		LPD	Logical Product Direct
11	00	YYYY	LPM	Logical Product Memory
11	YY		LPI	Logical Product Indirect
12	00	XXXX	LPC	Logical Product Constant
12	XX		LPF	Logical Product Forward
13	XX		LPE	Logical Product Backward

Description: Logical product of (X) (operand) → X

Selective Complement Instructions.

Selective complement instructions (table B3-16) form, in X, the bit complement of X for each bit in the operand equal to 1. The operand in storage remains unchanged.

Example:

(X) (binary)	111 010 001 101
operand (binary)	010 000 100 111
selective complement (binary)	101 010 101 010

Control continues as described under address modes.

NOTE

The E portion of SCI and SCF instructions must not be zero. If zero, the processor interprets the operation code as SCM or SCC, respectively.

TABLE B3-16. SELECTIVE COMPLEMENT INSTRUCTIONS

F	E	G	MNEMONIC	NAME
03	XX		SCN	Selective Complement No Address
14	YY		SCD	Selective Complement Direct
15	00	YYYY	SCM	Selective Complement Memory
15	YY		SCI	Selective Complement Indirect
16	00	XXXX	SCC	Selective Complement Constant
16	XX		SCF	Selective Complement Forward
17	XX		SCB	Selective Complement Backward

Description: Selective complement X_n for $\text{operand}_n = 1$

MEMORY BANK CONTROLS.

Memory bank control instructions, table B3-17, assign address modes to the three memory banks before transferring program control to a given bank reference

TABLE B3-17. MEMORY BANK CONTROL INSTRUCTIONS

F	E	MNEMONIC	NAME
00	1X	SRJ	Set Relative Bank Control and Jump
00	2X	SIC	Set Indirect Bank Control
00	3X	IRJ	Set Indirect and Relative Bank Control and Jump
00	4X	SDC	Set Direct Bank Control
00	5X	DRJ	Set Direct and Relative Bank Control and Jump
00	6X	SID	Set Indirect and Direct Bank Control
00	7X	ACJ	Set Direct, Indirect, and Relative Bank Control and Jump

Instructions SIC, SDC, and SID allow program continuation at (r)(P)+1. The remaining instructions, SRJ, IRJ, DRJ, and ACJ, are the only instructions which transfer program control between memory banks. The act of setting the relative bank control (r) alters the bank from which the next program instruction will be read. After assigning the relative mode to a memory bank, program control transfers to the relative address specified by the contents of the X register. Execution of a memory bank control instruction does not alter the contents of X. Not only may (r) be set by itself, combinations of memory bank controls may be set at the same time. The following example sets the direct and relative bank to 2 and transfers program control to storage bank 2, location 4600.

Example:	Location	F	E	G
	(r = 0) 3100	LDC	00	4600
	(r = 0) 3102	DRJ	52	

When there is only one memory bank, memory bank control instructions become no operation (NOP) instructions. After a NOP, control continues at the next address.

JUMP INSTRUCTIONS.

Jump instructions terminate the current program sequence and initiate a new sequence at a different memory address. Certain jump instructions unconditionally change the program sequence while contents of the X register condition other jump instructions.

Conditional Jump Instructions.

Conditional jump instructions, table B3-18, test the X register for the condition stated. If the specified condition exists, control transfers forward or backward in the relative (r) bank XX locations. If the specified condition does not exist, control continues at P + 1.

TABLE B3-18. CONDITIONAL JUMP INSTRUCTIONS

F	E	MNEMONIC	NAME
60	XX	ZJF	Zero Jump Forward
61	XX	NZF	Non-Zero Jump Forward
62	XX	PJF	Positive Jump Forward
63	XX	NJF	Negative Jump Forward
64	XX	ZJB	Zero Jump Backward
65	XX	NZB	Non-Zero Jump Backward
66	XX	PJB	Positive Jump Backward
67	XX	NJB	Negative Jump Backward

Test conditions are:

(a) X zero — contents of X equal positive zero (0000). Negative zero (7777) does not meet the jump condition.

(b) X not zero — X contains any quantity other than 0000.

(c) X positive — bit 11 of X is 0.

(d) X negative — bit 11 of X is 1.

Unconditional Jump Instructions.

Unconditional jump instructions, table B3-19, transfer program control regardless of X register contents.

TABLE B3-19. UNCONDITIONAL JUMP INSTRUCTIONS

F	E	G	MNEMONIC	NAME
70	YY		JPI	Jump Indirect
71	00	YYYY	JPR	Return Jump
71	XX		JFI	Jump Forward Indirect
74	XX	XXXX	JPD	Jump Display

Jump Indirect.

The jump indirect instruction (JPI) transfers program control to the relative (r) bank location specified by the contents of direct (d) bank location 00YY. After executing the JPI 43 instruction in the following example, since (d) 0043 contains 3662, program control transfers to location (r)3662.

Example:	Location	F	E
	(r) YYYY	JPI	43
	(d) 0043	36	62
	(r) 3662	Continue	

Return Jump.

The return jump (JPR) instruction transfers control from the main program sequence to a program subroutine. In doing this, reference figure B3-3, it provides a method of returning from the subroutine to the next step in the main program.

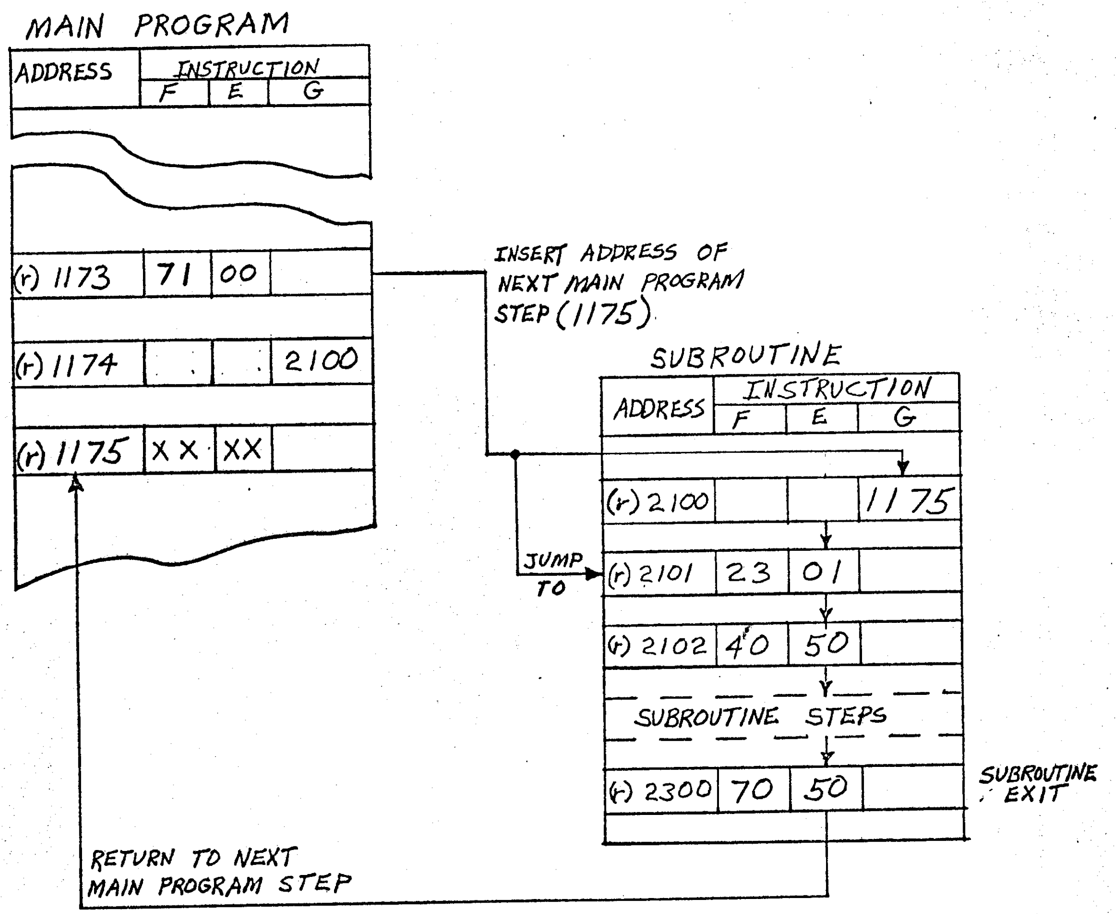


Figure B3-3. Return Jump (JPR) Application

A return jump increases the contents of P by 2, providing the address of the next main program step (r)1175 in this example . The processor transfers this address [(r)1175] to location (r) YYYY [(r)2100 in this example]. Program control transfers to location (r)YYYY + 1 [(r)2101]. Load backward and store direct instructions at the beginning of the subroutine, figure B3-3, enable an indirect jump to address (r)1175 of the main program after subroutine completion.

Jump Forward Indirect.

Jump forward indirect instructions transfer program control to the relative (r) bank address specified by contents of the storage location XX positions forward. After executing JFI07 in the following example, the processor references the relative bank location 0007 positions forward, (r)5171. Program control transfers to the address contained in this location, (r)0423.

Example:	Location	F	E
	(r)5162	JF1	07
	(r)5171	04	23

NOTE

The E portion of the JF1 instruction must not be zero. If zero, the processor interprets the operation code as a JPR instruction.

Jump Display.

Jump display (JPD) instructions transfer program control to the display mode of operation at the address specified by G. In the E portion of JPD instructions, bits 0 and 1 specify the display memory bank as follows.

BIT 1	BIT 0	BANK
0	0	0
0	1	1
1	0	2

Bit 5 of E determines whether to resume or clear the display. If bit 5 is a 1, display resumes where it left off. If bit 5 is a 0, display clears before starting.

INTERRUPTS.

Interrupt lines and instructions enable effective processor handling of the manual input devices (alphanumeric/funtion keyboard and light pen), remote interface communication, and peripheral equipment connected to the input/output channel.

Lines.

Interrupt signals, transmitted on one of four interrupt lines, notify the processor of external requests for action. An interrupt initiates transfer of processor program control to a fixed memory location without losing the information needed for return to the main program. There are four interrupt lines: 10, 20, 30, and 40. When an interrupt occurs on one of these lines, the processor stores the contents of P at location (d)0010, (d)0020, (d)0030, or (d)0040, depending on the line which generates the interrupt, and takes its next instruction from (r)0011, (r)0021, (r)0031, or (r)0041. The address at location (d)0010, (d)0020, (d)0030, or (d)0040 enables returning to the interrupted program with a jump indirect instruction (70YY).

Interrupt 10.

Interrupt 10 is a manual interrupt generated by a manual input device. Depressing a key on the keyboard or a light pen hit generates this interrupt.

Interrupt 20.

This line is only for subsystems with a remote interface. At completion of a processor initiated input or output to or from a modem, the remote interface generates an interrupt on line 20.

Interrupts 30 and 40.

Interrupt lines 30 and 40 may be activated by any peripheral equipment which provides an interrupt signal. The meaning of these interrupts is a function of the equipment generating the signal. Because several equipments may be connected to each line, each line must be interrogated following an interrupt to determine which equipment generated the interrupt.

Interrupt Servicing.

A scanner sequentially interrogates the four interrupt lines. Detection of an interrupt locks out further interrupts while the processor services the honored interrupt. When the processor services interrupts 10 and 20, it automatically clears the serviced interrupt. Interrupts 30 and 40 must be cleared with an external function to the interrupting peripheral unit. The Display Console MC switch clears all interrupts.

Instructions.

Interrupt instructions, table B3-20, enable effective usage of manual input devices (alphanumeric/function keyboard and light pen), remote interface communication, and input/output channel communication.

TABLE B3-20. INTERRUPT INSTRUCTIONS

F	E	MNEMONIC	NAME
01	20	CIL	Clear Interrupt Lockout
01	21	SIL	Set Interrupt Lockout
0/	22	IDS	Interrupt Data Source

Clear/Set Interrupt Lockout.

After detection of an interrupt or execution of an external function instruction, all further interrupts are locked out until execution of a clear interrupt lockout (CIL) instruction. Any interrupt line which becomes active during interrupt lockout remains active until execution of a CIL, at which time all interrupt lines will be sequentially scanned for activity. The set interrupt lockout (SIL) instruction unconditionally locks out all interrupts.

Interrupt Data Source.

The interrupt data source (IDS) instruction permits processor interruption of the data source when a predetermined condition occurs. If properly enabled by an 0026 function code, this instruction places a logical 1 on the data channel Interrupt line and appropriate Status line.

Typical Manual Interrupt Routine.

Figure B3-4 is a flow chart for a typical processor manual interrupt routine. Activating a manual input device (alphanumeric/function keyboard and light pen) interrupts the display or processor routine. When interrupted, the next address in the active routine goes to direct address 00/0 and a processor interrupt routine begins at relative address 0011. All manual interrupts are locked out until execution of clear interrupt lockout instruction. Because the X register serves as both an X position register and arithmetic register, depending upon operating mode, its contents must be stored before interrogating the manual input register. Interrogating the manual input register enables determining the interrupt source (function/status, a/n keyboard or light pen) with three sequential groups of register transfer, selective complement, and zero jump forward instructions. Table B3-21 is an example of interrogating the manual input register for a light pen interrupt. Determination of the interrupt source transfers control to the corresponding interrupt processing routine. After processing the interrupt, a clear interrupt lockout instruction enables further manual interrupts.

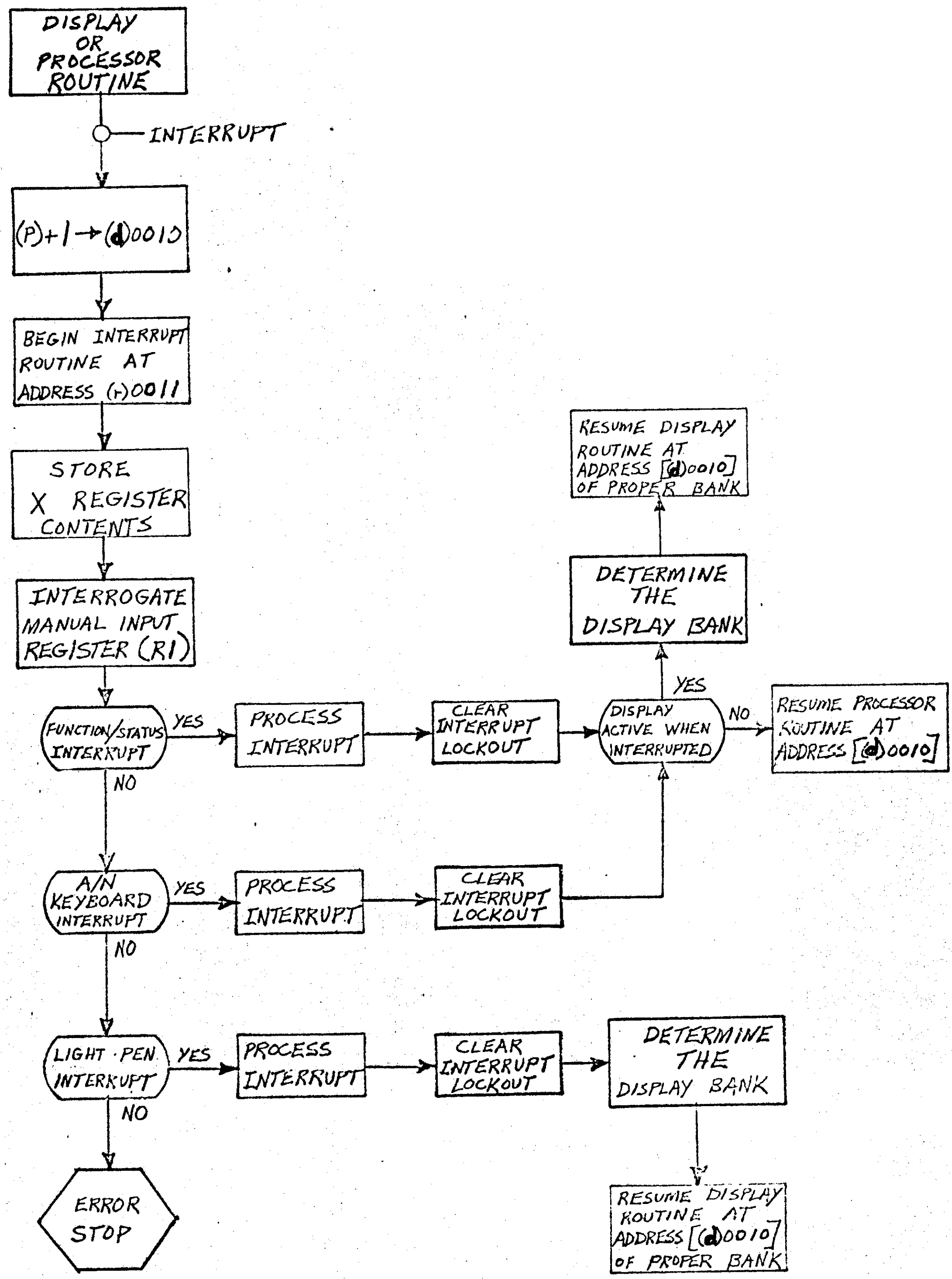

DISPLAY OR PROCESSOR ROUTINE
INTERRUPT
(P)+1 → (d)0010
BEGIN INTERRUPT ROUTINE AT ADDRESS (r)0011
STORE X REGISTER CONTENTS
INTERROGATE MANUAL INPUT REGISTER (R1)
FUNCTION/STATUS INTERRUPT
YES
PROCESS INTERRUPT
CLEAR INTERRUPT LOCKOUT
DISPLAY ACTIVE WHEN INTERRUPTED
YES
DETERMINE THE DISPLAY BANK
RESUME DISPLAY ROUTINE AT ADDRESS [(d)0010] OF PROPER BANK
NO
RESUME PROCESSOR ROUTINE AT ADDRESS [(d)0010]
NO
A/N KEYBOARD INTERRUPT
YES
PROCESS INTERRUPT
CLEAR INTERRUPT LOCKOUT
NO
LIGHT · PEN INTERRUPT
YES
PROCESS INTERRUPT
CLEAR INTERRUPT LOCKOUT
DETERMINE THE DISPLAY BANK
RESUME DISPLAY ROUTINE AT ADDRESS [(d)0010] OF PROPER BANK
NO
ERROR STOP

Figure B3-4. Typical Processor Manual Interrupt Routine

TABLE B3-21. MANUAL INPUT REGISTER INTERROGATION

ADDRESS (OCTAL)	F	E	G	DESCRIPTION
0043	74	01		R1 —>X
0044	16	00		SELECTIVE COMPLEMENT
0045			0400	Mask For Light Pen Interrupt Bit
0046	60	20		X =0000 — Jump to Interrupt Routine X ≠ 0000 — Go to Next Instruction

In the case of FUNCTION/status and a/n keyboard interrupts, examine bit 11 of the display state register. If bit 11 is a logical 1, display mode was interrupted. Before returning to the display routine, determine the display bank. Knowing the display bank and next address in the display routine, contents of direct address 0010, return to the display routine with a jump display. If bit 11 of the display state register is a logical 0, return to the interrupt processor routine with a jump indirect to the contents of direct address 0010.

In the case of a light pen interrupt, return to the display routine as previously described.

INPUT/OUTPUT INSTRUCTIONS.

Input/output instructions, table B3-21A, permit communication between the processor and peripheral equipment (unit).

TABLE B3-21A. INPUT/OUTPUT INSTRUCTIONS

F	E	G	MNEMONIC	NAME
*72	XX	YYYY	INP	Normal Input
*73	XX	YYYY	OUT	Normal Output
75	00		EXC	External Function Constant
75	XX		EXF	External Function Forward
76	00		INX	Input to X
76	77		OTX	Output from X

* (r)(P) + 00XX = FWA of the input or output area
YYYY = LWA + 1 of the input or output area

Normal Input/Output.

An input operation consists of reading information from a previously selected peripheral unit and storing it in a specified memory area. Transmitting information from a specified memory area to a previously selected peripheral unit is an output operation. These instructions define the input/output memory area as follows:

(a) The first word address (FWA) of the input/output memory area is XX locations forward in the relative memory bank (r). This location, (r) P + 00XX, specifies an address in the indirect memory bank (i).

(b) The last word address (LWA) +l of the input/output area is location YYYY in the indirect memory bank (i)YYYY.

If the peripheral unit has been properly selected, the input or output operation will take place and at its completion, control will continue at (r)(P)+2. If no peripheral unit has been properly selected, the processor will be indefinitely delayed (hung up).

Information transfers initiated by INP and OUT instructions are not buffered. That is, execution of processor instructions halts during an input or output. Upon completion of the input/output operation, the processor executes the next instruction. The contents of X at completion of an INP or OUT instruction indicate the LWA+l actually written into or read out of memory during the input or output. Although the FWA and LWA of the input/output memory area are in the relative memory bank (r), they specify locations in the indirect memory bank (i). The following program example reads 1000_8 words from a peripheral unit and stores them in memory locations $(i)1000_8$ through $(i)1777_8$.

NOTE

The E portion of INP or OUT instructions must not equal zero.

Example:

Location				
(r)1134	INP	33	2000	
(r)1136	Next instruction			
(r)1167	10	00		(FWA)
Final (X)	20	00		(LWA + 1)

External Functions.

External function constant and external function forward instructions transmit 12-bit function codes to peripheral units. The operand (function code) is selected and control continues as described under address modes. At completion of an external function instruction, X contains the 12-bit external function code.

An external function instruction selects a peripheral unit for performance of a specific function. With the exception of a status request code, attempting an illegal selection delays the processor indefinitely (hangs it up). One example of an illegal selection is attempting to select a magnetic tape unit for reading when the unit is off.

Most peripheral units have a status request code. When such a code is given and followed with an INA instruction, a 12-bit status response code will be sent to X. By examining this response code, it is possible to determine whether further selection of the unit is possible. A status request may be given even when a peripheral unit is off.

Only one unit may be selected at any time. Selection of any unit automatically disconnects any other selected unit. If a select is given and some other unit on the channel is busy, the processor will be delayed until a selection can be made. If a unit is selected for some function and another select is made on the same unit for some other function, the previous select is nullified. This applies to status request so that, if a unit is selected for reading and then status is requested of that unit, another read selection must be made before any further reading can take place.

X Input/Output.

INA and OTA instructions read or write, on a previously selected peripheral unit, one word to or from X. On equipments which transmit less than one full computer word at a time, information transfers to and from the low order portion of X. Program control continues at (r)(P) +1. Attempting an IN A or OTA with no peripheral unit selected hangs-up the processor.

DISPLAY REPERTOIRE.

The display mode of operation (activated by interface function code, processor jump-display instruction, or operator panel) enables displaying symbols, points, and vectors on the crt. Display commands may be located in bank 0, bank 1, or bank 2. While in display mode, the arithmetic module controls beam positioning and relative coding. The X register controls X beam position. The Y register controls Y beam position. Table B3-22 presents the display repertoire. For the following explanations, all numbers are octal unless stated otherwise.

ENTER MODE COMMANDS.

Codes 40 through 47 define display operating modes. Even codes select mode I. Odd codes select mode II. Bits 0 through 5 of an enter mode command assign display parameters for the selected mode, figure B3-5.

Mode I Position Words.

Mode I position words, figure B3-6, contains X positional information in odd words (first, third, etc) following the enter mode command and Y positional information in even words. Two position words follow each enter mode command.

TABLE B3-22. DISPLAY REPERTOIRE

	11	10	9	8	7	6	5	4	3	2	1	0
ENTER PLOT POINT MODE I (40)	1	0	0	0	0	0	☒	BLI	☒	☒	☒	
ENTER PLOT POINT MODE II (41)	1	0	0	0	0	1	SCA	BLI	☒	☒	☒	
ENTER PLOT SYMBOL MODE I (42)	1	0	0	0	1	0	☒	BLI	OR	☒	SIZE	
ENTER PLOT SYMBOL MODE II (43)	1	0	0	0	1	1	SCA	BLI	OR	☒	SIZE	
ENTER TABULAR SYMBOL MODE I (44)	1	0	0	1	0	0	☒	BLI	OR	☒	SIZE	
ENTER TABULAR SYMBOL MODE II (45)	1	0	0	1	0	1	SCA	BLI	OR	☒	SIZE	
ENTER VECTOR MODE I (46)	1	0	0	1	1	0	☒	BLI	DASH	☒	☒	
ENTER VECTOR MODE II (47)	1	0	0	1	1	1	SCA	BLI	DASH	☒	☒	
JUMP (52)	1	0	1	0	1	0	☒	☒	☒	P	B	B
RETURN JUMP (53)	1	0	1	0	1	1	☒	☒	☒	☒	☒	☒
IDENTIFIER (54)	1	0	1	1	0	0	ID					
START REFRESH CYCLE (55)	1	0	1	1	0	1	☒					
ENABLE OPERATOR DEVICE (56)	1	0	1	1	1	0	☒	☒	☒	LP	☒	A/N
DISABLE OPERATOR DEVICE (57)	1	0	1	1	1	1	☒	☒	☒	LP	☒	A/N
"X" AND "Y" MODE I POSITION WORDS	0	R	X+ΔX									
	0	BLA	Y+ΔY									
MODE II POSITION WORD	0	SIGN	ΔX				BLA	SIGN	ΔY			
SYMBOL WORD	SYMBOL 1						SYMBOL 2					
ADDRESS WORD	ADDRESS											

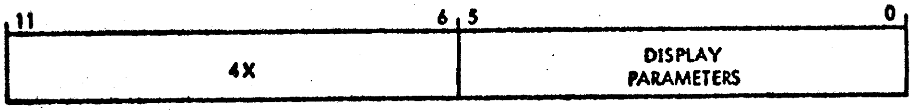


Figure B3-5. Enter Mode Command Format

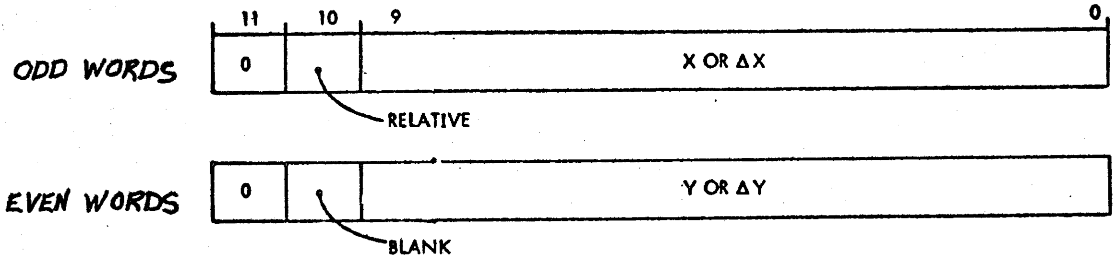


Figure B3-6. Mode I Position Words

Bit 11 of both position words must be zero or display logic interprets them as display commands, clearing the current display mode.

A 0 in bit 10 of the X position word directs the display logic to interpret the X and Y positional information as absolute values. A 1 in bit 10 of the X position word indicates that X and Y positional information is in relative mode. A 1 in bit 10 of the Y position word allows the beam to be positioned but not unblanked.

Absolute Positioning.

Absolute positioning, 0 in bit 10 of X position word, transfers bits 0 through 9 of the X and Y position words to bits 0 through 9 of the X and Y position registers, respectively. This enables positioning the beam to any X, Y coordinate on the crt raster. Minimum and maximum X, Y position values are 0000 and 1777, respectively.

Example:

COORDINATE	INITIAL POSITION	POSITION WORD I	FINAL POSITION
X	1232	0441	0441
Y	0573	1172	1172

Relative Positioning.

Relative positioning, 1 in bit 10 of X position word, enables updating the current X, Y position coordinates. The 10-bit ΔX, ΔY values in the position words combine with the current X, Y coordinates, yielding new X, Y coordinates. ΔX and ΔY are either positive values or negative values expressed in 1's complement. Minimum and maximum positive Δ values are 0000 and 0777, respectively. Minimum and maximum negative Δ values are 1777 and 1000, respectively. Table B3-23 lists some examples of X, ΔX relative positioning combinations resulting from 1's complement arithmetic.

TABLE B3-23. POSSIBLE X, ΔX COMBINATIONS, RELATIVE POSITIONING

EXAMPLE	CURRENT X POSITION	ΔX	NEW X POSITION	DIRECTION OF BEAM MOVEMENT (BLANKED)	MAGNITUDE OF CHANGE IN X POSITION
1	0000	0000	0000	NONE	0000
2	0000	0777	0777	→	+0777
3	1000	0777	1777	→	+0777
4	1001	0777	0001	←	- 1000
5	1777	0777	0777	←	- 1000
6	1777	1777	1777	NONE	0000
7	1777	1000	1000	←	- 0777
8	1000	1000	0001	←	- 0777
9	.0001	1000	1001	→	+ 1000

ALL NUMBERS ARE OCTAL

Mode II Position Word.

Mode II position words, figure B3-7, contain the relative X and Y positions in the same 12-bit word. Interpretation of the ΔX, ΔY values depends on the scale selected by the enter mode II command. A 1 in bit 5 of an enter mode II command selects scale 3. Scale 3 results in ΔX, ΔY values being left-shifted 3 positions before being added to the previous X, Y position. A 0 in bit 5 of an enter mode II command selects scale 1. Scale 1 results in ΔX, ΔY values being left-shifted 1 position before being added to the previous X, Y position. Bits 10 and 4 are sign bits for ΔX and ΔY, respectively. A 1 indicates a negative Δ value in 1's complement. A 0 indicates a positive Δ value in standard notation. A 1 in bit 5 allows beam positioning without unblank. Table B3-24 tabulates the Δ values for both scales.

Example:

COORDINATE	INITIAL POSITION	POSITION WORD II	FINAL POSITION	
X	1232	0532	1244	SCALE 1
Y	0573		0561	
X	1232	0532	1302	SCALE 3
Y	0573		0523	

11	10	6	5	4	0
0	SIGN	ΔX	BLANK	SIGN	ΔY

Figure B3-7. Mode II Position Word

TABLE B3-24. POSITION WORD 2 AND ΔX, ΔY VALUES

POSITION WORD 2 (OCTAL)	ΔX, ΔY (OCTAL)	
	SCALE 1	SCALE 3
0000	+ 0	+ 0
0101	+ 2	+ 10
0202	+ 4	+ 20
0303	+ 6	+ 30
0404	+10	+ 40
0505	+12	+ 50
0606	+14	+ 60
0707	+16	+ 70
1010	+20	+100
1111	+22	+110
1212	+24	+120
1313	+26	+130
1414	+30	+140
1515	+32	+150
1616	+34	+160
1717	+36	+170
2020	-36	-170
2121	-34	-160
2222	- 32	-150
2323	-30	-140
2424	-26	-130
2525	-24	-120
2626	-22	-110
2727	-20	-100
3030	-16	- 70
3131	-14	- 60
3232	-12	- 50
3333	-10	- 40
3434	- 6	- 30
3535	- 4	- 20
3636	- 2	- 10
3737	- 0	- 00

Plot Point Mode.

Codes 40 and 41, figure B3-8 select plot point mode. Code 40 initiates mode I which uses mode I position words. Code 41 initiates mode II which uses a mode II position word. A 1 in bit position 4 of either mode causes blinking of displayed points. Display logic remains in plot point mode until recognition of a new enter mode command.

	11	10	9	8	7	6	5	4	3 – 0
PLOT POINT MODE I (40)	1	0	0	0	0	0	▨	BLINK	▨
PLOT POINT MODE II (41)	1	0	0	0	0	1	SCALE	BLINK	▨

Figure B3-8. Plot Point Words

Mode I.

Plot point mode I requires two position words for each point. Odd position words following the enter plot point mode I command contain the X position and even position words contain the Y position. A 1 in bit 10 of the Y position word allows crt beam positioning but inhibits unblanking the point. A 0 in bit 10 allows both positioning and point unblank.

The following example program plots points at the corners and center of the crt raster. The center point blinks.

Program	Comments
4000	Enter plot point mode I
0000	Plot point at lower left corner
0000	
0000	Plot point at upper left corner
1777	
1777	Plot point at upper right corner
1777	
1777	Plot point at lower right corner
0000	
4020	Enter plot point mode I - blink
3000	(ΔX =-0777) Blink point at center
0777	(ΔY=+0777)

Mode II.

Plot point mode II requires only one position word for each point. Both ΔX and ΔY values are in a mode II position word. ΔX and ΔY binary weights, table B3-24, dependent upon the scale selected in the enter mode command. A 0 in bit 5 of the enter plot point mode II command selects scale I and a I selects scale 3. Scales I and 3 shift the ΔX, ΔY values left I and 3 positions, respectively, before adding them to the previous X, Y position. A I in bit 5 of the position word allows crt beam positioning but inhibits unblanking the point. A 0 in bit 5 allows both positioning and point unblank.

Assuming an initial crt beam position of X=1000 and Y=1000, the following program plots points at the corners of an imaginary square centered on the crt raster. Each side is 24 positions long.

Program	Comments
4100	Enter plot point mode II
3232	Plot point at lower left corner (X=0766, Y=0766)
0012	Plot point at upper left corner (X=0766, Y=1012)
1200	Plot point at upper right corner (X=1012, Y=1012)
0025	Plot point at lower right corner (X=1012, Y=0766)

Plot Symbol Mode.

Codes 42 and 43, figure B3-9, select plot symbol mode. Code 42 initiates mode I which uses mode I position words. Code 43 initiates mode II which uses a mode II position word. Display logic remains in plot symbol mode until recognition of a new enter mode command. Plot symbol mode command automatically selects the basic symbol set.

	11	10	9	8	7	6	5	4	3	2	1	0
PLOT SYMBOL MODE I (42)	1	0	0	0	1	0	▨	BLINK	OR	▨	SIZE	▨
PLOT SYMBOL MODE II (43)	1	0	0	0	1	1	SCALE	BLINK	OR	▨	SIZE	▨

Figure B3-9. Plot Symbol Words

Bit positions 4, 3, and 1 of either mode select blink, orientation, and size respectively. A 1 in bit position 4 selects blinking of following symbols and 0 inhibits blinking.

A I in bit position 3 plots symbols rotated 90 degrees counterclockwise and 0 plots symbols in normal orientation. A I in bit position I selects large size symbols and 0 selects small size. Reference table B3-25 symbol spacing.

TABLE B3-25. SYMBOL SPACING

BIT POSITION 1	HORIZONTAL SYMBOL SPACING (OCTAL)	HORIZONTAL SYMBOLS/LINE (DECIMAL)	VERTICAL SYMBOLS/COLUMN (DECIMAL)
0 (small)	14	86	64
1 (large)	20	64	43

Display logic automatically interprets the first word following the enter plot symbol command as a symbol word. Bits 6 through II of the symbol word, figure 3-I0, define symbol set. Symbol code I7 selects the extended symbol set, and all other codes select the basic symbol set for the symbol code in bits 5 through 0. Reference the symbol repertoire heading for basic and extended symbol sets.

11 6	5 0
Set Code	Symbol Code

Figure B3-I0. Symbol Word for Plot Symbol Mode

Display logic plots the symbol at positions specified by position words following the symbol word. Reference the plot point mode description for details on mode I and II plotting.

The following example program, using plot symbol mode I, plots four, small size, 90 degree letter "B's" equally spaced along the left edge of the raster.

Program	Comments
4210	Enter plot symbol mode (90 degrees small)
0062	Symbol word(B)
0000 0000	Plot "B" at lower left corner (absolute)
2000 0525	Plot "B" at X = 0000, Y = 0525
2000 0525	Plot "B" at X = 0000, Y = 1252
0000 1777	Plot "B" at upper left corner

Tabular Symbol Mode.

Codes 44 and 45, figure B3-11, select tabular symbol modes I and II, respectively. For mode I selection, the next two words following the tabular symbol word determine the initial X, Y position. In mode II, the position word following the tabular symbol word provides the initial X, Y position.

	11	10	9	8	7	6	5	4	3	2	1	0
TABULAR SYMBOL MODE I (44)	1	0	0	1	0	0	▨	BLINK	OR	▨	SIZE	▨
TABULAR SYMBOL MODE II (45)	1	0	0	1	0	1	SCALE	BLINK	OR	▨	SIZE	▨

Figure B3-11. Tabular Symbol Word

A 1 in bit 4 of either mode causes following symbols to blink. The size of the symbol (bit 1) determines the automatic spacing between symbols as follows.

Bit 1	Spacing	Symbol/Line	Vertical Symbols/ Column (Decimal)
0	14_8	86	64
1	20_8	64	43

The selected orientation determines whether the X or Y position registers increment. Normal orientation causes spacing in the X direction and 90 degree orientation causes spacing in the Y direction. Accomplish line spacing or carriage returns by escaping tabular mode and reentering at the desired X, Y position.

An enter tabular mode command automatically selects the basic symbol set. Upon recognition of a set shift code (17), the symbol set changes. If using the basic symbol set, shift code (17) selects extended symbols, and vice versa.

Except for the positional information, tabular symbol mode I and mode II are identical. Display logic treats all words following the position as symbol words, figure B3-12, until recognition of an escape code (32) in either symbol position. Reference the symbol repertoire heading for basic and extended symbol sets.

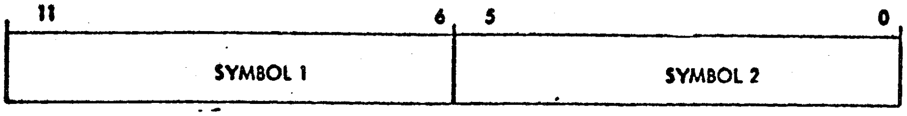


Figure B3-12. Symbol Word for Tabular Symbol Mode

Assuming an initial beam position of X=0100, Y = 1767, and large symbol size, the following program displays the word "Large" in normal orientation along the X axis at the top of the raster.

Program	Comments
4542	Enter tabular symbol mode II (scale 3, large size, normal orientation)
2701	ΔX = - 100, ΔY = 10
4317	L, case shift
6151	a, r
6765	g, e
3200	escape

Vector Mode.

Codes 46 and 47, figure B3-13, select vector mode. Mode I, code 46, enables painting full-length vectors for absolute position words and half-length vectors for relative position words. Code 47 initiates mode II. Scale 1 (bit 5 = 0) allows vector lengths up to 40_8 and scale 3 (bit 5 = 1) allows vector lengths up to 200_8 in both X and Y directions. The distance between crt beam positions is 0.0117 inch. Either mode permits selecting blinked or dashed vectors with a 1 in bit positions 4 and 3, respectively.

	11	10	9	8	7	6	5	4	3	2	1	0
VECTOR MODE I (46)	1	0	0	1	1	0	▨	BLINK	DASH	▨		
VECTOR MODE II (47)	1	0	0	1	1	1	SCALE	BLINK	DASH	▨		

Figure B3-13. Vector Words

Regardless of vector mode, the X, Y position registers contain the vector end point at completion of each vector. When using absolute coding, the position words contain the vector end point and become the starting point for the next vector. In relative coding, ΔX and ΔY values add to the current vector starting point, forming the next vector starting point. This feature enables "head-to-tail" vectors by simply programming a series of end points.

Example 1 : Draw a box around the display area.

Program	Comments
4600	Enter vector mode I
0000 2000	Position blanked crt beam to lower left corner
0000 1777	Upper left corner
1777 1777	Upper right corner
1777 0000	Lower right corner
0000 0000	Lower left corner

Example 2: With each side equal to 100, draw a square around the center of the raster. Use dashed vectors and assume an initial crt beam position of X = 1000, Y = 1000.

Program	Comments
4750	Enter vector mode II (scale 3, dash)
3373	$\Delta X=-40$, $\Delta Y=-40$, blank
0010	$\Delta X=00$, $\Delta Y=+100$
1000	$\Delta X=+100$, $\Delta Y=00$
0027	$\Delta X=00$, $\Delta Y=-100$
2700	$\Delta X=-100$, $\Delta Y=00$

JUMP COMMANDS.

Codes 52 and 53, figure B3-14 provide memory jump capability. For a direct jump, code 52, the 12-bit word following the jump command becomes the jump address. Program control transfers to the jump address in the memory bank specified by bits 0 and 1 of the jump command.

Bit 2 of the jump command selects the mode of operation, display or processor, following execution of the jump. A 0 continues display mode. A 1 selects processor mode.

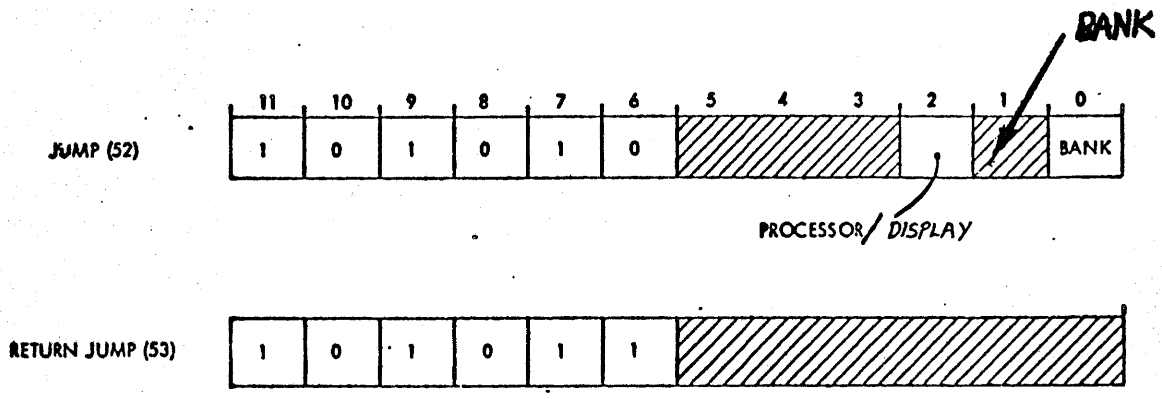


Figure B3-14. Jump Words

A return jump command, code 53, occurs within the same memory bank as the instruction, no bank selection. A 12-bit address word follows the return jump command. Executing a return jump stores the present address plus two at the location specified by the address word. Program control transfers to the contents of the address word plus one. Reference the return jump application, figure B3-3.

IDENTIFIER WORD.

Code 54, figure B3-15, stores bits 0 through 5 of the identifier word in the identity register. The internal processor examines the identity register by reading the state register. The last received identification code appears in bits 0 through 5 of the state register.

11	10	9	8	7	6	5 0
1	0	1	1	0	0	ID

Figure B3-15. Identifier Word

Figures B3-16 and B3-17 illustrate a typical method of using the identifier word for a circuit analysis application. Via program, an engineer displays the desired circuit and performs a mathematical analysis of circuit operation. To change circuit component values, the engineer points the light pen at the desired component and activates the light pen switch. For example, assume light pen designation of resistor R_3. Refreshing R_3 generates a light pen interrupt. Interrupt routine identification of a light pen interrupt initiates state register interrogation. In this example, the state register contains the identification code for R_3. After determining R_3 generated the interrupt, enabling the keyboard permits entering a different resistance value for R_3. In this manner, the design engineer modifies a circuit for desired characteristics.

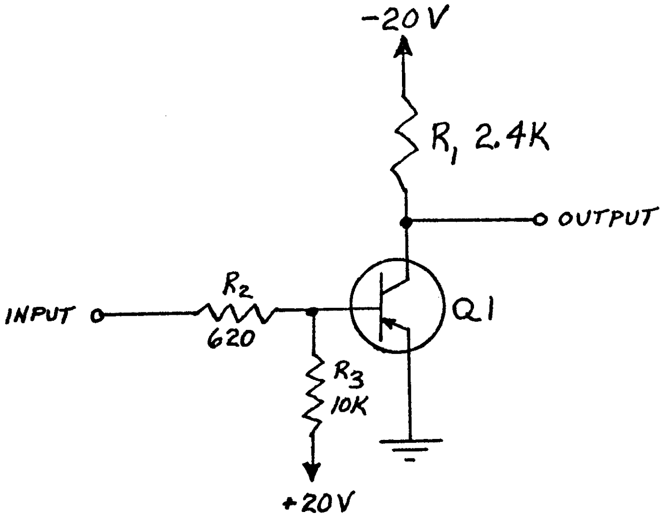


Figure B3-16. Typical Circuit Display

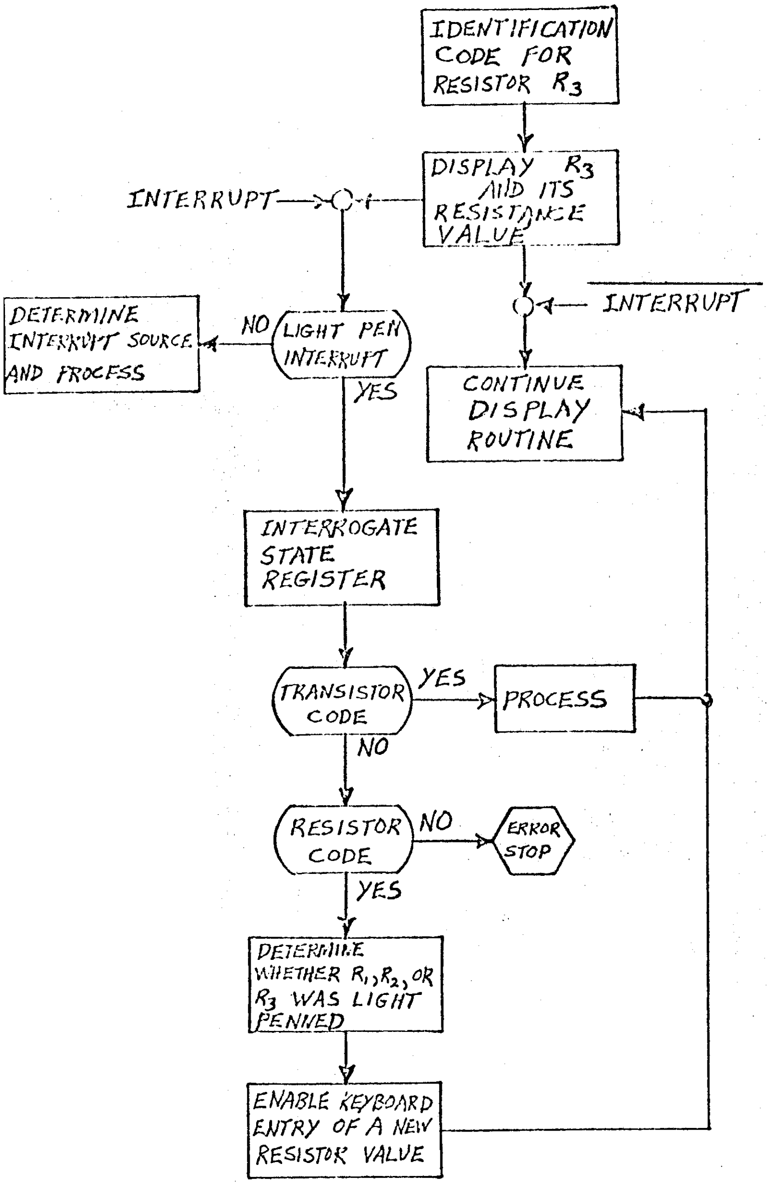

IDENTIFICATION CODE FOR RESISTOR R_3
DISPLAY R_3 AND ITS RESISTANCE VALUE
INTERRUPT
INTERRUPT
DETERMINE INTERRUPT SOURCE AND PROCESS
NO
LIGHT PEN INTERRUPT
YES
CONTINUE DISPLAY ROUTINE
INTERROGATE STATE REGISTER
TRANSISTOR CODE
YES
PROCESS
NO
RESISTOR CODE
NO
ERROR STOP
YES
DETERMINE WHETHER R_1, R_2, OR R_3 WAS LIGHT PENNED
ENABLE KEYBOARD ENTRY OF A NEW RESISTOR VALUE

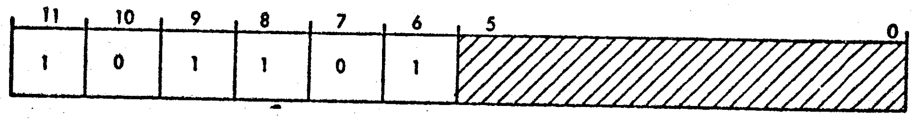
Figure B3-17. Typical Identifier Word Usage

START REFRESH.

The start refresh command, figure B3-18, controls the display pattern refresh rate. Recognition of code 55 triggers a 20-millisecond delay. Detection of another code 55 before delay termination stops display. After delay termination, display resumes and another 20-millisecond delay begins. Detection of code 55 after delay termination begins another 20-millisecond delay and display continues without interruption. This instruction must be used in each display pattern to assure a constant intensity, flicker-free display and prevent damage to the crt phosphor. Reference figure B3-19 for a start refresh example.

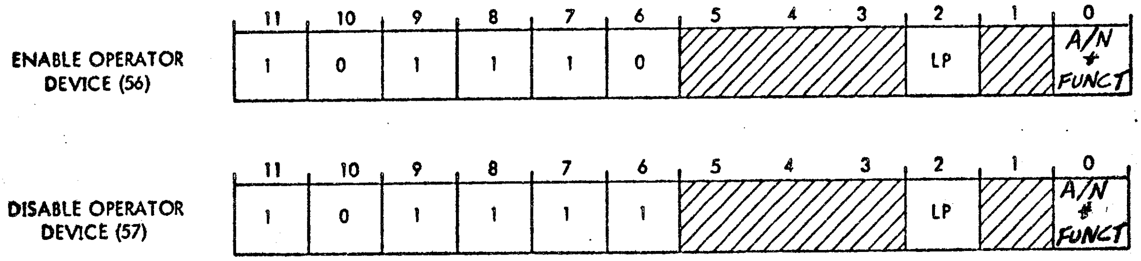
WARNING

Each display pattern must contain a start refresh command or damage to the crt phosphor may result.



11	10	9	8	7	6	5 0
1	0	1	1	0	1	▨

Figure B3-18. Start Refresh Cycle Word


ENABLE OPERATOR DEVICE (56)
11 10 9 8 7 6 5 4 3 2 1 0
1 0 1 1 1 0 LP A/N + FUNCT
DISABLE OPERATOR DEVICE (57)
11 10 9 8 7 6 5 4 3 2 1 0
1 0 1 1 1 1 LP A/N + FUNCT

FigureB3-20. Operator Control Device Words

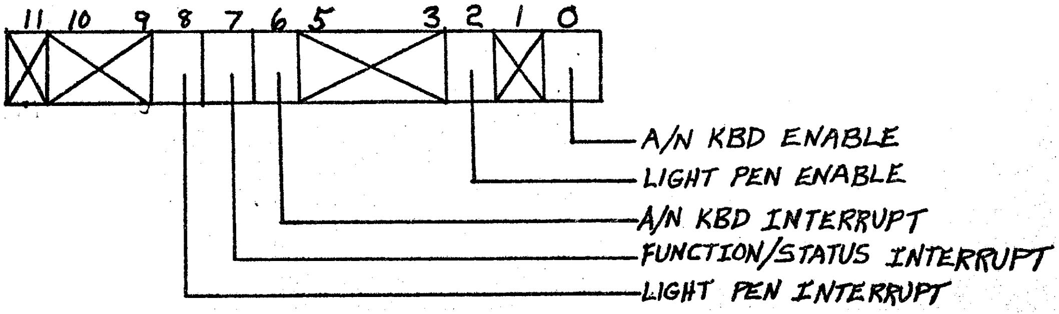


Figure B3-21. Manual Input Register (R_1)

SYMBOL REPERTOIRE.

Tables B3-26 and B3-27 list the basic and extended sumbol sets and codes, repectively.

TABLE B3-26. BASIC SYMBOLS

CODE (OCTAL)	SYMBOL	CODE (OCTAL)	SYMBOL	CODE (OCTAL)	SYMBOL
00	: (colon)	26	W	54	*
01	1	27	X	55	↑
02	2	30	Y	56	↓
03	3	31	Z	57	>
04	4	32	Tabular Escape	60	+
05	5	33	,	61	A
06	6	34	(	62	B
07	7	35	→	63	C
10	8	36	_ (underline)	64	D
11	9	37	∧	65	E
12	Ø	40	-	66	F
13	=	41	J	67	G
14	≠	42	K	70	H
15	≤	43	L	71	I
16	%	44	M	72	<
17	SET Shift	45	N	73	. (period)
20	Space	46	O	74	)
21	/	47	P	75	≥
22	S	50	Q	76	]
23	T	51	R	77	;
24	U	52	V		
25	V	53	$		

Uppercase codes 17 and 32 are control codes and do not have associated symbols.

When the Display Controller contains the optional extended symbol set, detection of code 17 initiates a set shift. With the basic symbol set selected, it shifts to extended symbol set and vice versa.

Code 32 causes an escape from the tabular symbol mode. This escape code may appear in either symbol position of a symbol word. If an escape code appears in symbol 1 of a symbol word, symbol 2 is ignored. Display control logic interprets the 12-bit word following the symbol word containing an escape code as a display command.

TABLE B3-27. OPTIONAL EXTENDED SYMBOLS

OCTAL CODE	SYMBOL DISPLAYED	DESCRIPTION
00	β	Small Beta
01	1	Subscript 1
02	2	Subscript 2
03	3	Subscript 3
04	ψ	Psi
05	ρ	Rho
06	γ	Gamma
07	φ	Small phi
10	ω	Small Omega

B3-27. OPTIONALLY EXTENDED SYMBOLS (CONT)

OCTAL CODE	SYMBOL DISPLAYED	DESCRIPTION
11	α	Small Alpha
12	∇	Del
13	δ	Small delta
14	Σ	Sigma
15	σ	Small sigma
16	μ	Mu
17		Shift to Basic Symbol Set
20		Space
21	←	Left Arrow
22	s	
23	t	
24	u	
25	v	
26	w	
27	x	
30	y	
31	z	
32		Tab Mode Escape
33	⊡	
34	⊙	
35	◬	
36	⟐	
37	∇̇	
40	±	Plus or minus
41	j	

B3-27. OPTIONALLY EXTENDED SYMBOLS (CONT)

OCTAL CODE	SYMBOL DISPLAYED	DESCRIPTION
42	k	
43	l	
44	m	
45	n	
46	0	
47	p	
50	q	
51	r	
52	∫	Integral sign
53	∂	Partial sign
54	°	Degree sign
55	Δ	Delta
56	π	Pi
57	√‾	Square root
60	θ	Theta
61	a	
62	b	
63	c	
64	d	
65	e	
66	f	
67	g	
70	h	
71	i	

B3-27. OPTIONALLY EXTENDED SYMBOLS (CONT)

OCTAL CODE	SYMBOL DISPLAYED	DESCRIPTION
72	2	Superscript 2
73	3	Superscript 3
74	"	Quotation mark
75	'	Prime mark
76	∞	Infinity
77	?	Question Mark

DISPLAY STATE AND KEYBOARD REGISTERS.

The display state and keyboard (R_2) register supply information on display conditions and keyboard data, repectively.

DISPLAY STATE REGISTER.

The display state register, figure B3-16A, is an internal control register available to the processor for determining conditions in the display control mode.

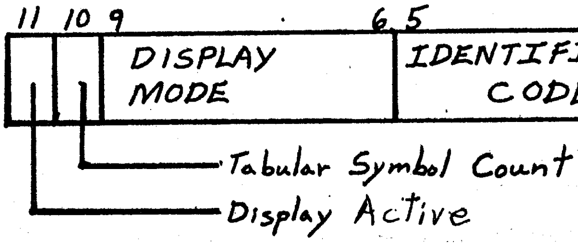


Figure B3-16A. Display State Register

Bit 11 indicates the last operating mode. A 1 indicates the display and a 0 indicates the processor was last active.

Bit 10 denotes the symbol count when in tabular mode. A 1 indicates that symbol code 2 was last displayed and a 0 indicates that symbol code 1 was last displayed.

Bits 6 through 9 equal the lower 4 bits of the present display mode.

Bits 0 through 5 equal the last received identification code.

KEYBOARD REGISTER (R_2).

The keyboard register (R_2) accepts data from the alphanumeric/function keyboard.

With the keyboard enabled, depression of any key other than a function, status, or the ~~INT~~ INTER key results in the R_2 setup shown in figure B3-17A.

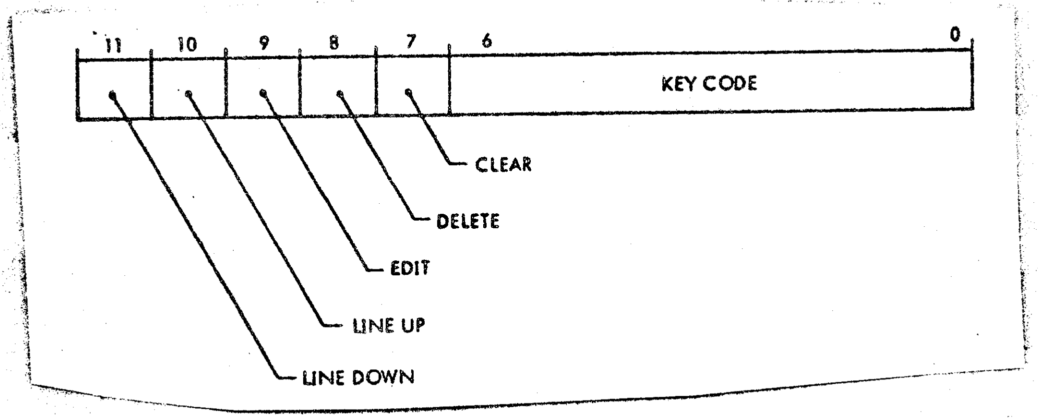


Figure B3-17A. Edit and Alphanumeric Key Word (R_2)

Depression of a function key or the ~~INT~~ INTER key results in the R_2 setup shown in figure B3-18A.

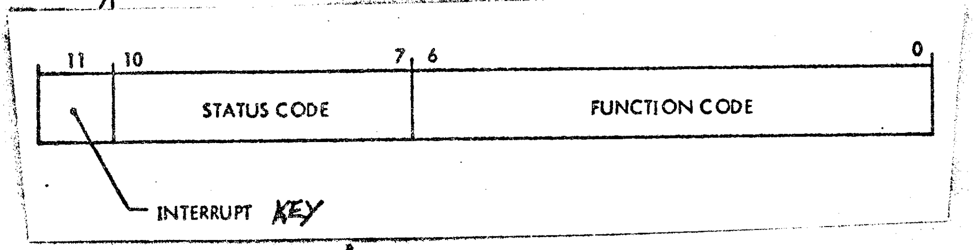


Figure B3-18A Function and Interrupt Word (R_2)

Table B3-28 lists the keyboard code assignments.

TABLE B3-28. R2 AND KEYBOARD CODE ASSIGNMENTS

R2 CODE	SYMBOL	R2 CODE	SYMBOL	R2 CODE	SYMBOL	R2 CODE	FUNCTION KEY
000	: (colon)	041	J	101	Return	160	F0
001	1	042	K	102	Send	161	F1
002	2	043	L	103	Not used	162	F2
003	3	044	M	104	↑	163	F3
004	4	045	N	105		164	F4
005	5	046	O	106		165	F5
006	6	047	P	107		166	F6
007	7	050	Q	110		167	F7
010	8	051	R	111		170	F8
011	9	052	V	112		171	F9
012	Ø	053	$	113			
013	=	054	*	114			
014	≠	055	↑	115			
015	≤	056	↓	116	↓		
016	%	057	>	117	Not used		
017	Not used	060	+	120	Bksp		
020	Space	061	A	121	Tab		
021	/	062	B	122	Line Skip		
022	S	063	C	123	NoT Used		
023	T	064	D	124	Line Clr		
024	U	065	E	125	Not used		
025	V	066	F	126	↑		
026	W	067	G	127	↓		
027	X	070	H	130	Not used		
030	Y	071	I	131	Skip		
031	Z	072	<	132	Reset		
032	Not used	073	. (period)	133	Not used		
033	, (comma)	074	)	134	Not used		
034	(	075	≥	135	Not used		
035	→	076	Not used	136	Tab Set		
036	_ (underline)	077	; (semicolon)	137	Not used		
037	∧	100	Not used	140	Aux Send		
040	- (minus)						

PROGRAMMING HINTS.

1. Use start refresh command in each display pattern.
2. After transferring to processor mode from display mode, store the X register contents if they are of value.
3. Place all negative constants in one's complement notation.
4. Express negative ΔX and ΔY relative position values in one's complement notation.
5. Always enable an operator device before attempting usage.
6. Always select a peripheral unit before initiating an input or output.

EXAMPLE PROGRAM.

By employing processor, display, and manual operator device commands, the example program provides a working familiarity with display subsystem programming. Figure B3-19A is a flow chart of the program listing in table B3-30.

Via the data source, store the program in Display Controller memory bank 0 beginning at address 0000. After activating display mode with the proper interface function code or the DISP MODE switch on the operator control panel, begin the program at address 0313. Table B3-29 lists program operating procedures. Figures B3-20A and B3-21A show the tuning pattern and box with tracking cross displays, respectively.

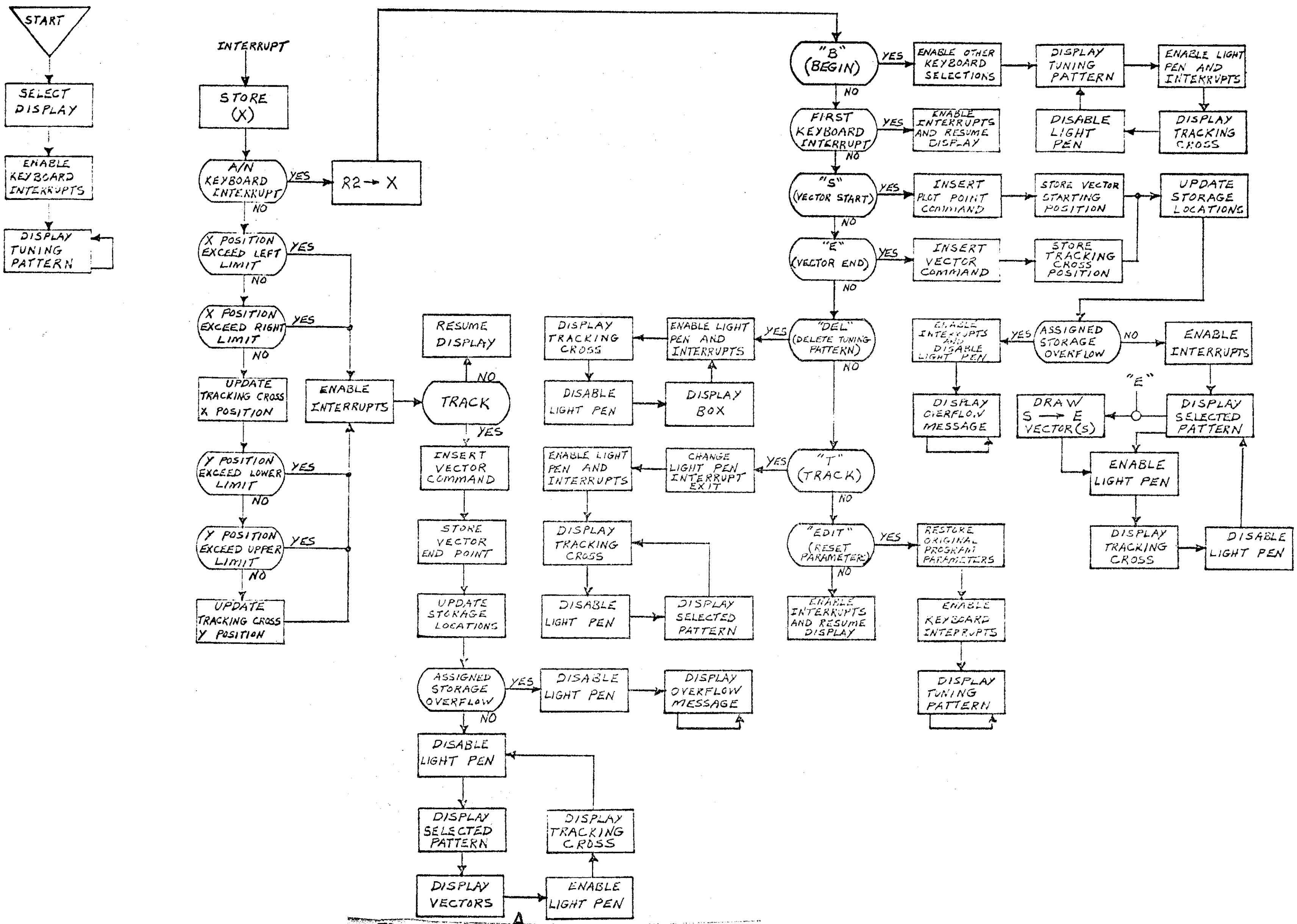


Figure B3-19A Flow Chart of Example Program

TABLE B3-29. PROGRAM OPERATING PROCEDURES

ALPHANUMERIC KEY	DESCRIPTION
B	B displays the tracking cross and enables keyboard selection of other functions. With the light pen, track or capture mode, the operator positions the tracking cross by light penning a point in the inner or outer circle. The tracking cross repositions to the selected point. The outer display pattern box defines the limit of tracking cross movement.
DEL	DEL deletes the tuning pattern and displays only a box together with the tracking cross.
S	S designates the present tracking cross position as a vector starting point.
E	E designates the present tracking cross position as a vector end point and draws a vector from S to E. S must be activated before E.
T	T selects track. In track, the operator may draw a continuous line with the light pen and tracking cross. The program storage area limits line length. A displayed message signals tracking data overflow.
EDIT	EDIT provides a program restart.

TO BE SUPPLIED

Figure B3-20A Tuning Pattern

TO BE SUPPLIED

Figure B3-21A Box With Tracking Cross

TABLE B3-30. EXAMPLE PROGRAM

MEMORY LOCATION	F	E	COMMENTS
0000	57	04	Disable light pen
1	40	00	Plot point
2	15	50	Tracking cross X position
3	13	70	Tracking cross Y position
4	52	00	Jump → tracking cross
5	04	36	
6	17	27	Mask
7	74	40	Resume display
0010	XX	XX	Interrupt address + 1
1	71	01	Jump to interrupt routine
2	00	41	Interrupt routine address
3	05	33	Jump address
4	05	32	Mode address
5	77	00	Storage limit
6	71	01	Jump to kbd routine
7	00	73	
0020	XX	XX	Temporary X position storage
1	05	33	X insert address
2	05	34	Y insert address
3	05	32	Jump command address
24 ↕			}0024 → 0040 Storage area
41	40	20	Store X position
2	01	71	R1 → X
3	12	00	Logical product constant
4	01	00	Mask 2^6 — a/n keyboard bit
5	65	27	(X) ≠ 0 jump to kbd routine
6	20	20	X position → X
7	07	50	Check left limit
0050	63	16	(X) neg — jump forward 16
1	06	50	Restore (X)
2	34	06	Check right limit
3	62	13	(X) pos — jump forward 13
4	30	06	Restore (X)
5	40	02	Update X position
6	01	70	(Y) → X
7	07	50	Check lower limit
0060	63	06	(X) neg — jump forward 6
1	06	50	Restore (X)
2	34	06	Check upper limit
3	62	03	X pos — jump forward 3

TABLE B3-30. EXAMPLE PROGRAM (CONT)

MEMORY LOCATION	F	E	COMMENTS
0064	30	06	Restore (X)
5	40	03	Update Y position
6	01	20	Enable interrupts
7	71	01	Resume interrupted routine
0070	00	71	
1	74	40	Resume display
2	03	13	
3	01	72	R2→X
4	03	62	
5	60	32	(X) = 0 Begin "B"
6	71	01	First Key ≠ "B", resume display
7	00	07	Becomes "S" check after "B"
0100	60	42	(X) = 0 → Vector start "S"
1	01	72	R2→X
2	03	65	
3	60	44	(X) = 0 Vector end "E"
4	01	72	R2→X
5	16	00	
6	04	00	
7	60	12	(X) = 0 Delete "DEL"
0110	01	72	R2→X
1	03	23	
2	60	11	(X) = 0 Track "T"
3	01	72	R2→X
4	16	00	
5	10	00	
6	60	07	(X) = 0→Reset "EDIT"
7	71	01	Resume display
0120	00	07	
1	71	01	Jump→delete
2	02	07	
3	71	01	Jump→track
4	02	21	
5	71	01	Jump→reset
6	02	27	
7	23	14	Begin routine
0130	43	32	
1	22	00	
2	03	22	

TABLE B3-30. EXAMPLE PROGRAM (CONT)

MEMORY LOCATION	F	E	COMMENTS
0133	43	34	Provide "S" check data
4	04	00	
5	41	00	Enable tracking cross display
6	05	33	
7	01	20	Enable interrupts
0140	74	40	Resume display
1	03	13	
2	22	00	Vector start routine
3	40	00	
4	41	14	Insert plot point command
5	71	01	
6	01	52	
7	22	00	Track routine
0150	46	00	
1	41	14	Insert vector command
2	20	02	
3	41	21	Insert X position
4	20	03	
5	41	22	Insert Y position
6	04	03	
7	50	14	Mode address +3
0160	04	03	
1	50	21	X address +3
2	04	03	
3	50	22	Y address +3
4	04	03	
5	50	23	Jump command address +3
6	04	03	
7	50	13	Jump address +3
0170	22	00	
1	52	00	
2	41	23	Move Jump command
3	04	00	
4	41	13	Change jump address
5	20	14	
6	10	15	
7	34	15	

TABLE B3-30. EXAMPLE PROGRAM (CONT)

MEMORY LOCATION	F	E	COMMENTS
0200	60	04	(X) = 0 — assigned storage overflow
1	01	20	Enable interrupts
2	74	40	Resume display
3	03	13	
4	01	20	Enable interrupts
5	74	40	Jump → overflow message
6	02	70	
7	22	00	Delete routine
0210	05	14	Address of box
1	41	00	Delete tuning pattern
2	05	13	
3	41	00	
4	02	03	
5	40	72	
6	01	20	Enable interrupts
7	74	40	Resume display
0220	00	00	
1	22	00	Initiate track
2	01	47	Address of track routine
3	40	70	
4	01	20	Enable interrupts
5	74	40	Resume display
6	00	00	
7	22	00	Reset routine
0230	71	01	
1	41	00	
2	00	76	
3	04	07	
4	41	00	
5	00	77	
6	22	00	
7	03	13	Tuning pattern address
0240	41	00	
1	05	13	Tracking cross exit address
2	41	00	
3	02	03	
4	41	13	
5	40	72	
6	04	71	

TABLE B3-30. EXAMPLE PROGRAM (CONT)

MEMORY LOCATION	F	E	COMMENTS
0247	40	70	
0250	22	00	
1	05	32	
2	40	14	
3	40	23	
4	22	00	
5	05	33	
6	40	21	
7	40	13	
0260	22	00	
1	05	34	
2	40	22	
3	22	00	
4	52	00	
5	41	23	
6	74	40	Jump → Tuning pattern
7	03	13	
0270	55	00	Overflow message routine
1	57	04	Disable light pen
2	44	02	Tabular symbols — large-normal
3	05	40	
4	12	00	
5	23	51	T, R
6	61	63	A, C
7	42	71	K, I
0300	45	67	N, G
1	20	64	Space, D
2	61	23	A, T
3	61	20	A, Space
4	46	25	O, V
5	65	51	E, R
6	66	43	F, L
7	46	26	O, W
0310	32	00	Escape tabular mode
1	52	00	Repeat message
2	02	70	
3	57	04	Tuning pattern routine
4	46	00	Vector mode I

TABLE B3-30. EXAMPLE PROGRAM (CONT)

MEMORY LOCATION	F	E	COMMENTS
0315	00	00	Lower left (blank)
6	20	00	
7	00	00	↑
0320	17	77	
1	17	77	→
2	17	77	
3	17	77	↓
4	00	00	
5	00	00	←
6	00	00	
7	17	77	↗
0330	17	77	
1	00	00	↙
2	00	00	
3	00	00	Upper left (blank)
4	37	77	
5	17	77	↘
6	00	00	
7	00	00	↖
0340	17	77	
1	04	00	Begin 2^9 box
2	24	00	
3	04	00	↑
4	14	00	
5	14	00	→
6	14	00	
7	14	00	↓
0350	04	00	
1	04	00	←
2	04	00	
3	44	02	Tabular symbols — large — normal
4	07	50	Top line
5	16	00	
6	67	51	G, R
7	71	64	I, D
0360	32	00	Escape tabular mode
1	44	12	Tabular symbols — large — 90 degree
2	16	00	Right line
3	07	30	

TABLE B3-30. EXAMPLE PROGRAM (CONT)

MEMORY LOCATION	F	E	COMMENTS
0364	23	24	T, U
5	45	71	N, I
6	45	67	N, G
7	32	00	Escape tabular mode
037 0	44	02	Tabular symbols — large — normal
1	07	20	Bottom line
2	01	77	
3	47	61	P, A
4	23	23	T, T
5	65	51	E, R
6	45	32	N, escape tabular mode
7	44	12	Tabular symbols — large — 90 degree
040 0	01	77	Left line
1	07	20	
2	65	27	E, X
3	61	44	A, M
4	47	43	P, L
5	65	32	E, escape tabular mode
6	42	02	Plot symbol — large — normal
7	00	40	Dash (-)
041 0	10	00	Top (-)
1	13	00	
2	13	00	Right (-)
3	10	00	
4	10	00	Bottom (-)
5	05	00	
6	05	00	Left (-)
7	10	00	
042 0	42	12	Plot symbol - large - 90 degree
1	00	40	Dash (ı)
2	10	00	Top (ı)
3	13	00	
4	13	00	Right (ı)
5	10	00	
6	10	00	Bottom (ı)
7	05	00	
043 0	05	00	Left (ı)
1	10	00	

TABLE B3-30. EXAMPLE PROGRAM (CONT)

MEMORY LOCATION	F	E	COMMENTS
0432	56	01	Enable Keyboard
3	01	20	Enable interrupts
4	52	00	Jump → refresh
5	05	30	
6	46	00	Tracking cross routine
7	20	00	↑
0440	00	50	
1	20	00	↓ (blank)
2	37	27	
3	20	50	→
4	00	00	
5	37	27	← (blank)
6	20	00	
7	20	00	↓
0450	17	27	
1	20	00	↑ (blank)
2	20	50	
3	37	27	←
4	00	00	
5	20	50	→ (blank)
6	20	00	
7	56	04	Enable light pen
0460	41	00	Plot point mode II
1	00	05	ΔX = 00 ΔY = + 12 (small circle)
2	03	35	ΔX = +06 ΔY = - 04
3	02	34	ΔX = +04 ΔY = - 06
4	35	34	ΔX = - 04 ΔY = - 06
5	34	35	ΔX = - 06 ΔY = - 04
6	34	02	ΔX = - 06 ΔY = +04
7	35	03	ΔX = - 04 ΔY = +06
0470	02	03	ΔX = +04 ΔY = +06
1	03	40	ΔX = - 06 ΔY = 00, blank
2	00	11	ΔX = 00 ΔY = +22 (large circle)
3	05	36	ΔX = + 12 ΔY = - 02
4	03	35	ΔX = +06 ΔY = - 04
5	03	33	ΔX = +06 ΔY = - 10
6	01	32	ΔX = +02 ΔY = - 12
7	36	32	ΔX = - 02 ΔY = - 12

TABLE B3-30. EXAMPLE PROGRAM (CONT)

MEMORY LOCATION	F	E	COMMENTS
0500	34	33	ΔX = -06 ΔY = -10
1	34	35	ΔX = -06 ΔY = -04
2	32	36	ΔX = - 12 ΔY = -02
3	32	01	ΔX = - 12 ΔY = +02
4	34	02	ΔX = -06 ΔY = +04
5	34	04	ΔX = -06 ΔY = +10
6	36	05	ΔX = -02 ΔY = +12
7	01	05	ΔX = +02 ΔY = +12
0510	03	04	ΔX = +06 ΔY = +10
1	03	02	ΔX = +06 ΔY = +04
2	52	00	Jump to current
3	03	13	display pattern
4	57	04	Box routine
5	46	00	Vector mode I
6	00	00	Lower left (blank)
7	20	00	
0520	00	00	↑
1	17	77	
2	17	77	→
3	17	77	
4	17	77	↓
5	00	00	
6	00	00	←
7	00	00	
0530	55	00	Start refresh
1	57	04	Disable light pen
2	52	00	Jump to current
3	03	13	display pattern
			Storage
			↓