



Z20-1848-0

SALES and SYSTEMS GUIDE

PRELIMINARY USER'S GUIDE FOR IBM 1130/2250 MODEL 4 GRAPHIC SYSTEM

This manual introduces IBM sales and systems representatives to the new, low-cost IBM 1130/2250 Model 4 Graphic System. The information discussed is derived from preliminary specifications and analyses. The reader, therefore, should refer to the latest IBM publications for prices, features, and specifications as they become available.

IBM CONFIDENTIAL

FOREWORD

This preliminary user's guide introduces the IBM Marketing Representative to the new and dynamic 1130/2250 Model 4 Graphic System.

The information contained herein was taken from preliminary specifications and analyses. It is suggested that the reader refer to the latest IBM publications as they are made available.

IBM CONFIDENTIAL

Copies of this and other IBM publications can be obtained through IBM branch offices. Address comments concerning the contents of this publication to IBM, Technical Publications Department, 112 East Post Road, White Plains, N. Y. 10601

CONTENTS

INTRODUCTION	1
STAND-ALONE SYSTEM	6
LOCAL AND REMOTE SYSTEM	7
SYSTEM HIGHLIGHTS	14
HARDWARE CAPABILITY	26
Cycle Stealing and Interference	26
Display Commands	28
Display Interrupts	28
Display Orders	29
SOFTWARE CAPABILITY	31
Machine Requirements	31
Language Compatibility	31
Structure of a GSP Program	31
Initializing the Graphic Subroutine Package	32
Creating, Modifying, and Displaying an Image Entity	33
Communication between a GSP Program and the 2250 Operator ...	37
APPLICATIONS	44
Graphic Applications	44
Applications for the Stand-Alone System	54
Applications for the 1130/2250 Attachment to System/360	54
APPENDIX: DESCRIPTION OF DISPLAY ORDERS	55



INTRODUCTION

The 1130/2250 system provides low-cost entry into graphic data processing for customers with no System/360. Before the 1130/2250, the minimum graphic system was a System/360 Model 30 with 2250 at approximately \$10,000 a month. In addition to providing a low-cost graphic system, the 1130/2250 can be attached to the System/360 for a local or remote configuration. Each of the 2250 models is geared for different environments and loads on the System/360:

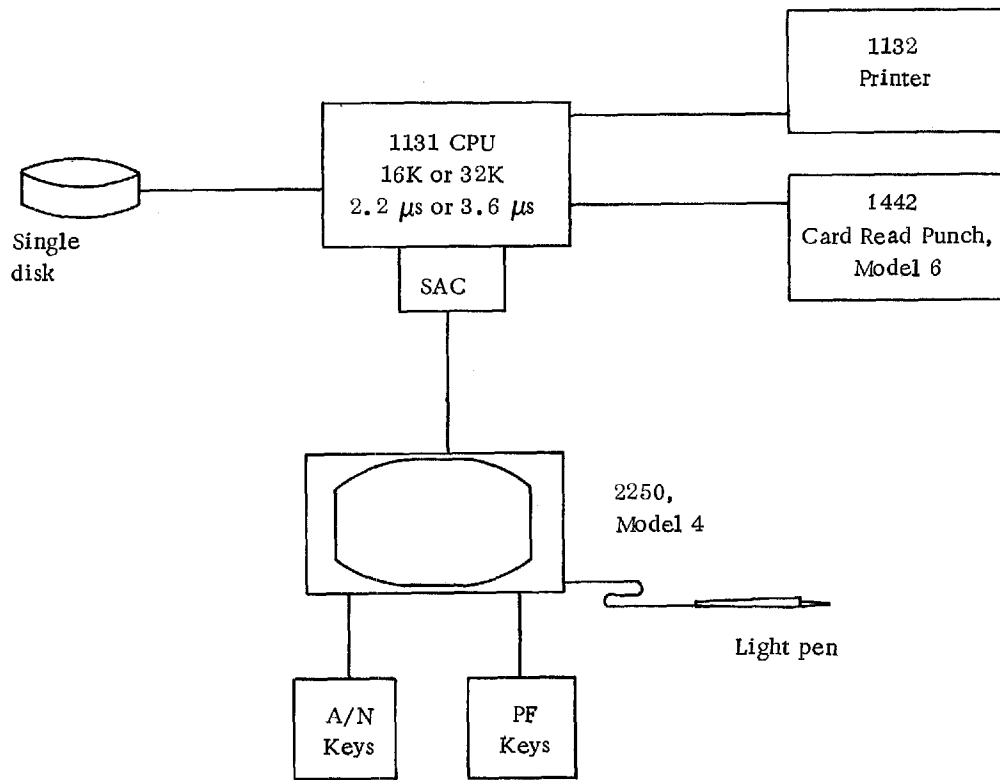
- The 2250 Model 1 has very little logical function and relies primarily on the System/360 for attention handling and application functions.
- The 2250 Model 3 provides additional functions not included in the Model 1. In addition, multiple units may share the 2840 Control Unit. (The 2250 Model 1 has a separate buffer for each unit.) The 2250 Model 3 does reduce the load on the System/360.
- The 1130/2250 attached to the System/360 provides a well balanced system where the 1130 can perform attention handling and application functions and fulfill most computational requirements. A significant reduction of the load on the System/360 is a key factor in this configuration.

In addition, the 1130/2250 Model 4 can be remotely attached to the System/360 when there is an application requirement for communicating with System/360. It can also function as a stand-alone during part or all of the application solution time.

Each application environment must be investigated as to the requirements for the 2250 unit. Normally, a customer would start small -- that is, his introduction to graphics would be either an 1130/2250 or the attachment of a 2250 Model 1 to an existing System/360. As the graphics workload increases, the customer would normally expand to a System/360 with 2250 Model 3s in a multiconsole environment. He may also expand to several 1130/2250s attached to the System/360. Figure 7 shows the entry price for each of the various 2250s attached to System/360. Figures 1-4 describe the various stand-alone configurations for the 1130/2250.

<u>Single-Disk Configuration</u>	<u>Feature</u>	<u>Rental</u>
1131 Model 2C		\$1,510
16K words 3.6 μ s		
Single disk		
Storage Access Channel	7490	25
1132 Printer (80-110 lpm)		268
1132 Attachment	3616	10
Expansion Adapter	3854	5
1442 Card Read Punch, Model 6 (300 cpm/80 cpm)		265
1442 Attachment	4454	36
2250 Model 4		2,400
Optional Features:		
A/N Keyboard		75
PF Keyboard		150
		<hr/>
		\$4,744
 <u>Expansions of Single-Disk Configuration</u>		<u>Rental</u>
Single-disk configuration with:		
16K 3.6 μ s		\$4,744
16K 2.2 μ s		5,184
32K 3.6 μ s		5,744
32K 2.2 μ s		6,384

Figure 1. 1130/2250 Model 4 configurations -- single disk



Rental: \$4,744 to \$6,384

Figure 2. Single-disk configuration

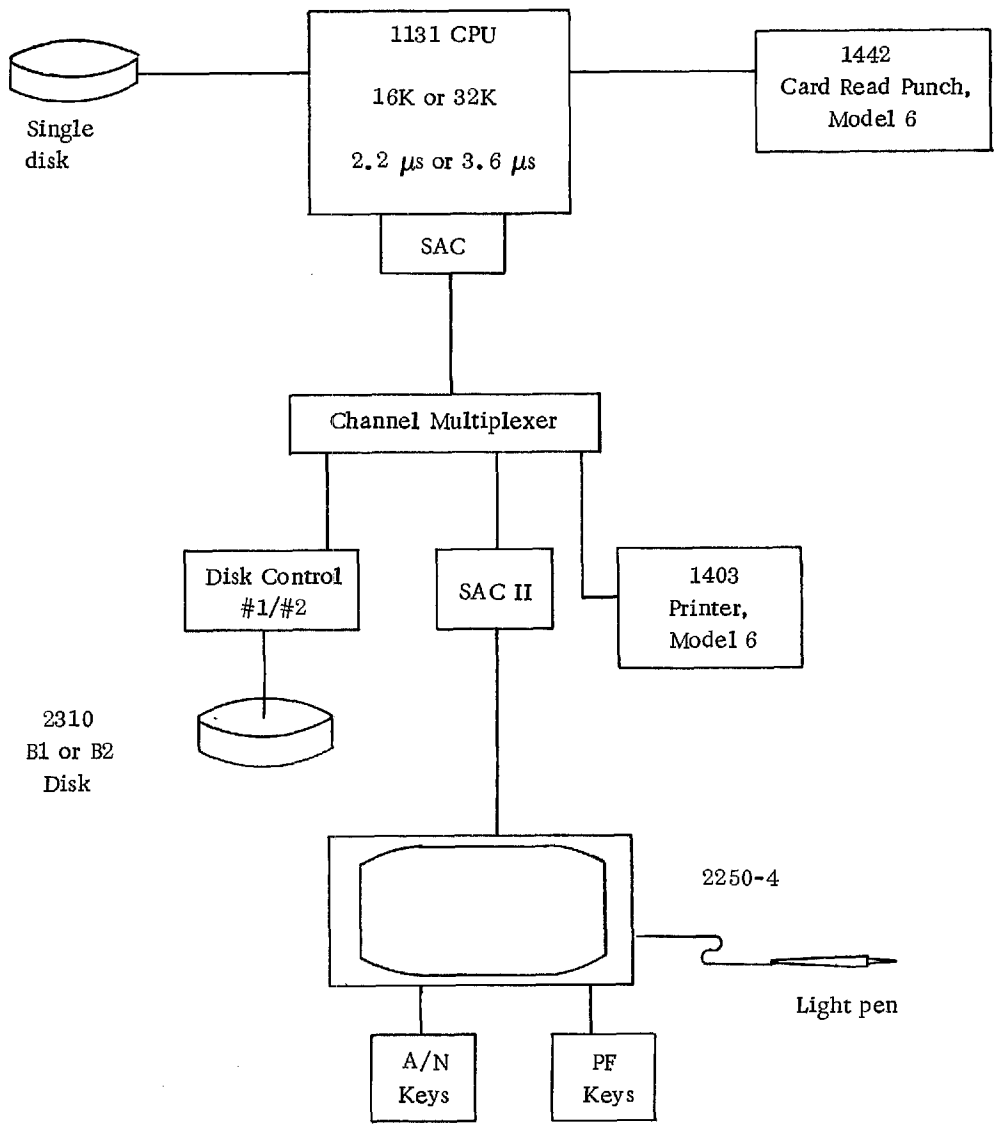
<u>Multiple-Disk Configuration*</u>	<u>Feature</u>	<u>Rental</u>
1131 Model 2C		\$1,510
16K words 3.6 μ s		
Single Disk		
Storage Access Channel	7490	25
Storage Access Channel II	7492	25
1133 Multiplex Enclosure		40
Channel Multiplexer	1865	165
Disk Control 1	3201	45
2310 Model B1 Disk Storage		270
1403 Model 6 Printer (340 lpm)		400
1403 Attachment	4424	450
1442 Card Read Punch, Model 6 (300 cpm/50 cpm)		265
1442 Attachment	4454	36
2250 Model 4		2,400
Optional Features		
A/N Keyboard		75
PF Keyboard		150
		<u>\$5,856</u>

*Also includes higher-speed printer

Expansions of Multiple-Disk Configuration

Core Size and Speed		No. of Disk Units	
		2	3
16K	3.6 μ s	\$5856	\$6061
16K	2.2 μ s	6296	6501
32K	3.6 μ s	6856	7061
32K	2.2 μ s	7496	7701

Figure 3. 1130/2250 Model 4 configurations -- multiple disks



Rental: \$5,856 to \$7,701

Figure 4. Multiple-disk configuration

STAND-ALONE SYSTEM

The 1130/2250 Model 4 stand-alone system provides a low-cost entry into graphic applications. The normal graphic application would require a 16K, 3.6-microsecond, single-disk 1130/2250 system. Figure 1 depicts such a system.

If the user is going to use the FORTRAN IV Graphic Subroutine Package, he will need at least the system described in Figure 1 in order to have sufficient capability for an engineering application. (See "Applications" for a discussion of the application considerations for the stand-alone system.)

LOCAL AND REMOTE SYSTEM

The attachment of an 1130 to a System/360 permits a flexible configuration where the 1130/2250 Model 4 becomes a local or remote subsystem that performs the conversational and display portions of an application and calls on the central system for computational assistance and/or access of a central data base.

The application environment of this system can be illustrated by an example of an engineering analysis type application where the System/360 is needed for its fast computational capabilities.

The analyst would first insert the disk cartridge of his program and data onto an 1130 system disk drive. Using the display, he would then start defining his input data. The input may be in graphic form (lines, points, circles, or full part drawing), alphameric form (parameters to a mathematical function, weight and dimension of a beam, etc.), or a combination of both. During this input phase the 1130/2250 Model 4 subsystem is essentially a stand-alone graphic system that is assembling and preprocessing the input data.

When assembled to the satisfaction of the analyst, the set of input data would be communicated to the central computing system together with control type information that identifies the program to be executed and certain control functions, such as the type of plots to be calculated. This information would cause the central system to start the analysis phase of the application.

During this phase the System/360 is doing high-powered and complicated computation while the 1130/2250 system is waiting for the results or preparing a new set of input parameters. We shall assume now that the analysis program in System/360 is written so that it computes "n" data sets and stores them in System/360. The interim results of one of these data sets could be periodically transmitted to the remote subsystem and displayed to the analyst. On the basis of this, the analyst could:

1. Stop the execution of the program.
2. Change an input parameter and restart the program.
3. Allow the execution of the program to continue.

When the program is completed the user starts an iteration phase where he requests the transmission and display of the data sets that were generated and stored in System/360. The form of the display is specified by the analyst and might be a tabular listing of output parameters, a graph, or a combination of both.

After the display of the output results the analyst may change some of his original parameters and restart the program or start a new problem.

Now that we have seen the application environment of this system, let us consider some of the specific aspects of his configuration.

Q. What is the functional difference between the 1130/2250 Model 4 and the 2250 Model 1 and Model 3?

A. The Model 1 has limited functional capability and relies on the System/360 for most functions. The Model 3 has many additional functions that significantly reduce System/360 interference. The 1130/2250 Model 4 has computing capability which does not require System/360 assistance except when large scale applications require a S/360.

* * *

Q. If the 1130 can handle the input aspect of an application, why do we need the System/360?

A. We need it for:

1. Applications that require computational needs beyond that of the 1130.
2. Applications that require a centralized or large data base.

* * *

Q. How is the increased cost of this system justified over the 2250 Model 1 or Model 3?

A. It is justified by:

1. The dollar value of remote capabilities (saved travel time, shipping of programs, increased turnaround time).
2. The dollar value of stand-alone capability.
3. Saving in System/360. The 1130/2250 graphic subsystem possesses considerable processing capabilities. It will handle the conversational aspect of an application, thereby eliminating the associated overhead and reducing the core requirement in the System/360. For customers for whom graphics is only part of

the total system, this means that more productive time will be available for other System/360 uses.

* * *

- Q. Will the 300-cps or higher-speed (5000 to 30,000 cps) transmission between the 1130 and System/360 satisfy a particular application? The 2250 Models 1 and 3 operate at much higher data rates.
- A. Recognize first that the 1130/2250 is not always communicating with System/360. The application environment explained above clearly indicates that the 1130/2250 subsystem is in a stand-alone mode for some duration of time within an application. During this mode of operation there is no transmission to System/360 and the response time due to operator actions should be very fast.

The effect of the data rate on response time should, therefore, be considered only when the 1130 is communicating to the System/360. Specifically, the turnaround time for transmission to the System/360, System/360 computation or access of data, and then a transmission from System/360 to the 1130 must be considered.

In such a case, depending on the type of application, the justification of the 300 cps or higher speed is based on any or all of the following:

1. Whether small amounts of data or complete buffer loads of data are being transmitted between the two systems.
2. Whether the System/360 computation or data access time is large compared with the transmission time.
3. Whether data display at the 1130 could be overlapped with transmission. For example, a quick response time can be achieved by starting to display portions of the data as soon as the data is received in the 1130 subsystem.

Last and most important, it must be recognized that the 1130 computing capability reduces the frequency of communication between the display and System/360. If some operator action causes slower response than is normally acceptable to the operator, it will be tolerated as long as it does not occur frequently.

* * *

- Q. In general, what type of application can be satisfied with 300 cps and which ones are likely to need 5000 cps?
- A. In general, 300 cps should be satisfactory for analysis-type applications where small amounts of data (messages, control parameters, single graphs, etc.) are transmitted between the subsystem and the central processor; 5000 cps may be needed for applications requiring frequent transmission of large amounts of data.

* * *

- Q. What guidelines would help in organizing an application so that maximum performance could be achieved at minimum cost?
- A. 1. Do as much as possible in the 1130 subsystem. This saves 360 time and core. It also reduces frequency of transmission to System/360, which in turn may result in reducing the bandwidth requirement of the application.
2. If you have to communicate with System/360, minimize the transmission time by compacting the data and overlapping image generation with transmission.

Figure 5 shows the functional capabilities of each of the 2250 display units. The workload and core requirements for the System/360 are reduced as the power of the various 2250 systems is increased. Figure 6 shows the functional capabilities of each of the 2250 display units for specific application-oriented functions. Figure 7 shows price comparisons for "typical" configurations.

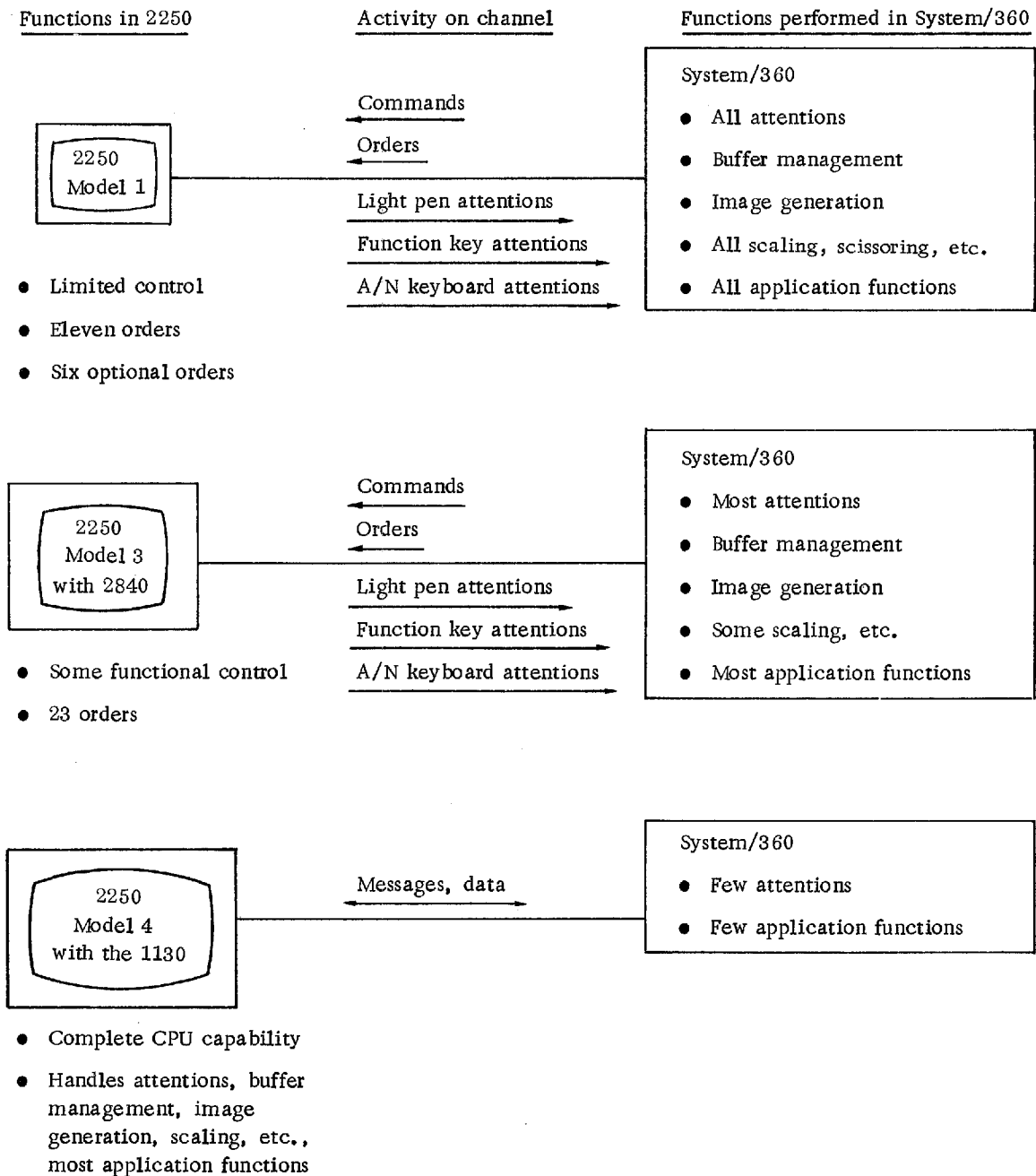


Figure 5. Functional comparison of 2250 Display Units attached to System/360

Function	IBM 2250 Model 1	IBM 2250 Model 1 w/GDF	IBM 2250 Model 3	IBM 2250 Model 4 attached to 1130*
Light pen tracking	CPU req'd	Assists CPU	CPU not req'd	CPU not req'd
Incremental vectors	No	Yes	Yes	Yes
Continuous detect	No	Yes	Yes	Yes
Subroutine in buffer	No	No	Yes	Yes
Copy an image	CPU req'd	Assists CPU	CPU not req'd	CPU not req'd
"Rubber-banding"	CPU req'd	Assists CPU	CPU not req'd	CPU not req'd
Move an image	CPU req'd	Assists CPU	CPU not req'd	CPU not req'd
Selective detect	CPU req'd	Assists CPU	CPU not req'd	CPU not req'd
Highlighting	CPU req'd	CPU req'd	CPU not req'd	CPU not req'd
Reduced buffer requirements with small vectors	No	Yes	Yes	Yes
Arithmetic capability within buffer	No	No	No	Yes
Local I/O capability (readers, disk, etc.)	No	No	No	Yes
Stand-alone capability	No	No	No	Yes
Remote attachment	No	No	No	Yes

* In this column, CPU refers to System/360 CPU. The 1130 is considered as the "buffer".

Figure 6. Functional capability of 2250 displays attached to the System/360

Number of Consoles	2250 Model 1			2250 Model 3			2250 Model 4 attached to System/360* (Average Configuration)
	Simple Applic.	Complex Applic.	Maximum	Simple Applic.	Complex Applic.	Maximum	
1	1,925	2,550	2,925	5,600	5,825	5,825	4,500
2	3,850	5,100	5,850	7,200	7,650	7,650	9,000
3	5,775	7,650	8,775	9,200	9,875	9,875	13,500
4	7,700	10,200	11,700	10,800	11,700	11,700	18,000

* The 1130/2250 system prices include the complete stand-alone configuration. These prices do not consider the attachment prices and transmission costs for remote systems using 1130/2250 Model 4s.

Figure 7. Prices for 2250 Display Units attached to System/360

SYSTEM HIGHLIGHTS

The 1130/2250 Model 4 provides a well balanced combination of man-machine graphics capability and CPU computation power. Since the CPU core storage is used for both the display buffer and application program, it leads to efficient control of the display as well as other economies. The system takes advantage of the fast interrupt handling capability of the 1130 to perform certain functions which were implemented with hardware in the 2250 Model 3. For instance, pen tracking (a function which is handled totally within the buffer of the Model 3 and which involves following the random motions of the light pen anywhere on the screen) does require some CPU assistance. However, it is performed with less CPU assistance than the 2250 Model 1 with the Graphic Design Feature (GDF) on System/360. Character generation is performed by a blend of hardware and software to provide user-specified character sets.

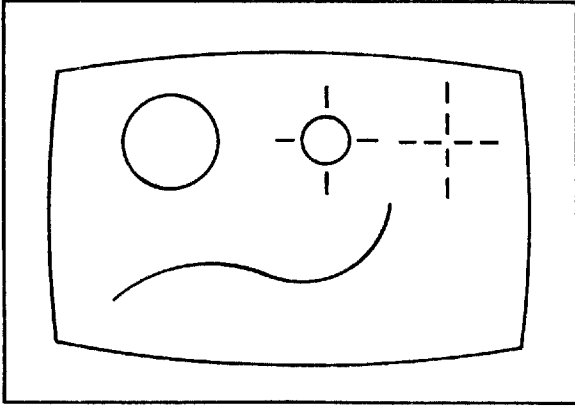
Highlights of the 1130/2250 Model 4 are as follows:

- Incremental and absolute line drawing and point plotting
- A programmable character set completely specified by the user
- Variable display buffer requirements -- any section or sections of CPU core can be addressed by display orders
- Fast update of display list by using CPU core for display buffer
- An efficient method of pen tracking
- Fast line drawing and point plotting for small incremental changes
- Upper- and lowercase alphabetic input capability from optional alphameric keyboard
- Superscript and subscript capability for display of alphamerics
- Automatic off-screen scissoring of lines and points in incremental mode
- Only two basic display modes (graphic and character) to minimize the number of orders required to perform mode changes
- A short form (one word) for absolute horizontal or vertical vectors
- In graphic mode, provision for a random mixture of absolute and incremental beam motion orders

- Conditional branching and interrupt facilities based on pen detect and pen switch conditions
- A sophisticated level of automatic subroutines for generating character strokes
- Provision for multilevel subroutining in the display list, through indirect address branching and storing of return addresses

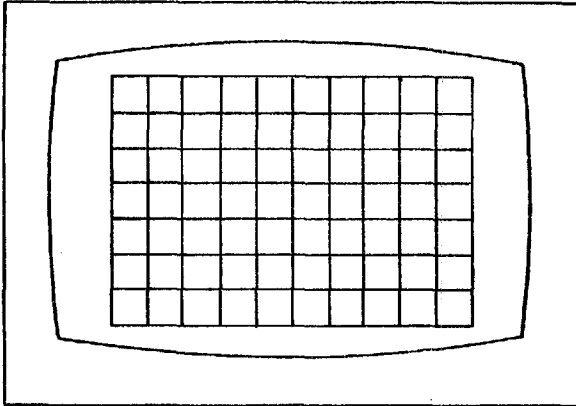
The remainder of this section illustrates these features and shows how they are used in an application environment.

Reduced Core Storage Required with Short Vectors, Circles, Arcs, Curves,
and Points



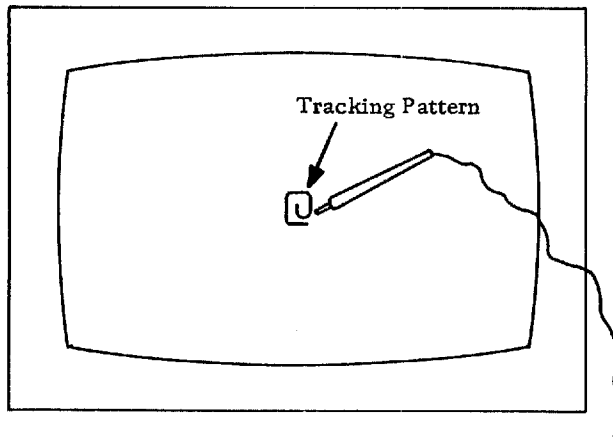
The incremental vector mode allows lines of up to .74" to be stored in one word of 1130 core as opposed to the two words per vector required with normal absolute vectors. This means, for instance, that circles, arcs, and curves composed of many small straight line segments can be displayed from the 1130 with a 50% reduction in core requirements. For example, a continuous curve containing 500 short vectors will require 500 words of 1130 storage compared with 1,000 words when using the absolute vector feature.

Reduced Core Storage Required for Grid Lines



The short absolute vector format requires only one word of 1130 core to store a horizontal or vertical line of any length up to 12 inches. For instance, beam motions required to draw a grid of 7 lines by 10 lines which covers the entire screen can be stored in 31 words of 1130 core, whereas 62 words are required using normal absolute vectors. The two-to-one saving in core storage will also apply for drawing bar charts, wiring diagrams, and integrated circuit layouts.

Input by Light Pen Tracking



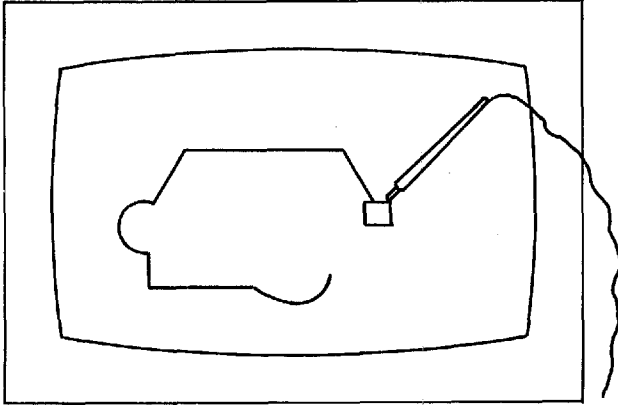
Tracking pattern indicates pen position

The light pen must sense a point, character, or vector; then the computer can determine the precise position of the pen. In many applications, the user may desire to point at a position on the screen where there is no source of light. A symbol or pattern may be displayed such that it follows the path of the light pen and can indicate to the CPU the present position of the pen. Some orders that make possible programming of this pen tracking technique are:

- | | |
|---------|------------------------------------------------------|
| SPM | - Set Pen Mode (enable detects and defer interrupts) |
| MBI/DBI | - Move/Draw Beam Incremental |
| GBC | - Graphic Branch Conditional |
| GIC | - Graphic Interrupt Conditional |

These enable the display of a user-designed pattern which will follow the pen being moved about the screen. These orders also make possible the detection of individual elements in the pattern, eliminate CPU interruption due to detection, and permit multiple detects without repeated depression of the pen switch. The net result is that light pen tracking can be performed in an efficient manner with these orders and the fast interrupt handling capability of the 1130. The user may use the tracking pattern provided with Type I support or can design his own pattern and tracking routine.

"Rubber-Banding"

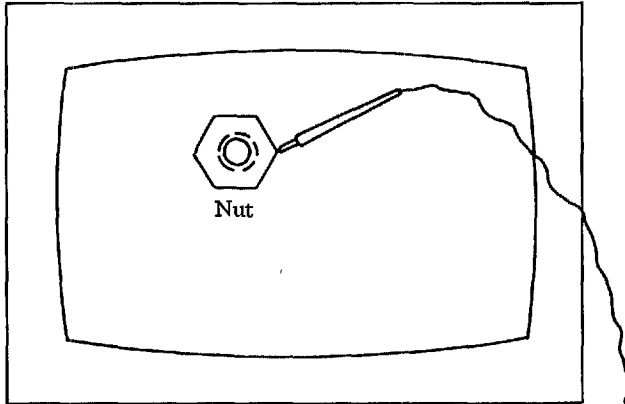


The line is stretched or shortened
by movement of the light pen

In many applications it is desirable to manipulate a line as if it were a rubber band and could be shortened or lengthened by the movement of the light pen. One end point of the line may remain stationary while the other follows the movement of the light pen. Thus, the line is continually adjusted to the proper length with interrupts to the CPU for pen tracking purposes only. The orders that may be used to program this technique include:

- SPM - Set Pen Mode (enable detects and defer interrupts)
- MBI, DBI - Move/Draw Beam Incremental
- GBC - Graphic Branch Conditional
- GIC - Graphic Interrupt Conditional

Identification of an Object



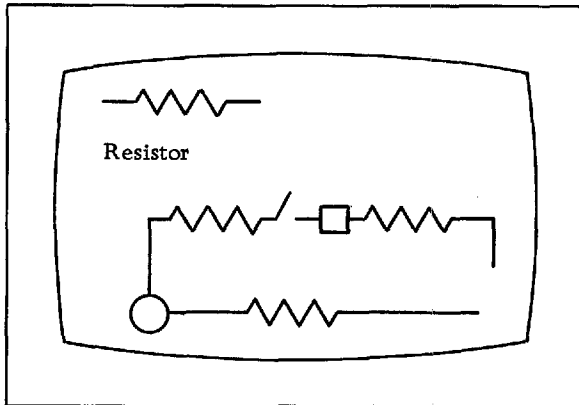
Identifying a complex object

In many cases, it is desirable to know what object has been referenced by the light pen regardless of what graphic element on that object has been detected. The display above shows an object, a "nut", being detected. The user may be interested only in the fact that the "nut" was detected and not that the detect occurred on one particular vector. The orders that make programming of this possible include:

- | | |
|-----|------------------------------------------------------|
| SPM | - Set Pen Mode (enable detects and defer interrupts) |
| GBC | - Graphic Branch Conditional |
| GIC | - Graphic Interrupt Conditional |

The orders allow the effective use within the 1130 core display list of hierarchical elements that can be labeled and treated as separate entities for purposes of identification. A typical hierarchy of elements in a design application would be assembly, subassembly, and parts.

Graphic Subroutine Ability



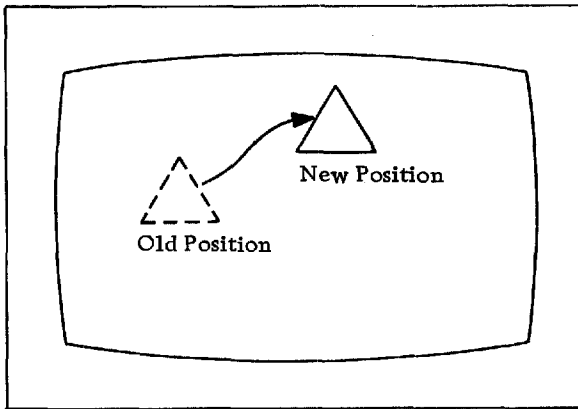
An image, the resistor, appears many times on the display but only once in the display program.

The 2250 Model 4 Display Unit enables the user to display a subroutine image, such as a resistor, in many places on the screen even though the orders appear only once in the display list. By use of logical display orders, the display program can retain core addresses for a return path from a subroutine. With this capability, an image subroutine can be stored in the 1130 core once and can be displayed many times throughout a regeneration cycle by subroutine linkages. The orders that provide this capability are:

- GB - Graphic Branch
- SRVT - Store Revert Register
- RVT - Revert

Within limits of screen and the 1130 core allocated to the display list, the image subroutine can be used to copy multiple images on the screen, and any image can be moved, deleted, or detected without interference with the other images generated by the subroutine.

Movement of Images

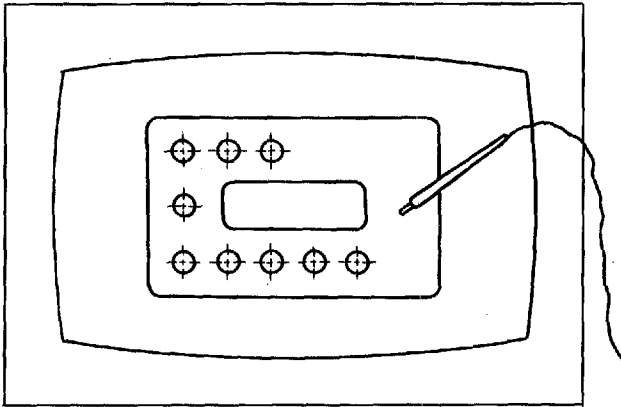


Moving an image

In many applications it is necessary to move the image on the display to a new position. This can be done with minimum 1130 interrupts using the display orders for the 2250 Model 4. The orders that make this possible include:

- MBI, DBI - Move/Draw Beam Incremental
- GBC - Graphic Branch Conditional
- GIC - Graphic Interrupt Conditional
- RVT - Revert

Copying Images

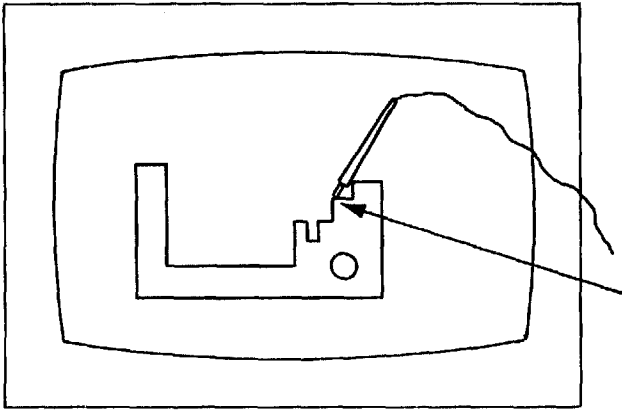


An image may be copied and placed anywhere on the display.

In many applications an object may appear several times on the drawing. It is very convenient if the user is allowed to display the image once and then, with the light pen, point at the other positions where he wants the object to appear. This kind of copy procedure can be preplanned, implemented, and executed with a minimum of 1130 interrupts. Orders that aid the copy function are:

GB	- Graphic Branch
RVT	- Revert

Highlighting a Detected Image

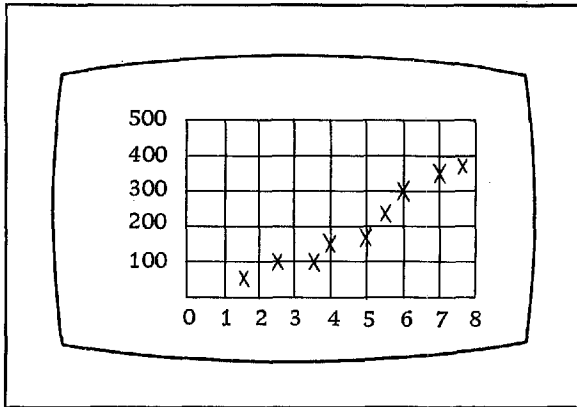


Element will be intensified to indicate that it has been detected.

It is highly desirable to highlight a graphic element that has been detected by the light pen. This procedure guarantees that the proper element has been selected by the user. One approach to this technique is to have the detected element temporarily intensified. This procedure can be implemented in the 2250 Model 4 and can be executed with minimum CPU interference. The repeated display of a detected element required to obtain intensification can be readily implemented with the following orders:

GB	- Graphic Branch
RVT	- Revert
SRVT	- Store Revert Register

Selective Detection without CPU Interrupt



Detect on data and not on
background grid

It is often necessary to ensure that only specific points, characters, or vectors are detected with the light pen while other information on the screen is ignored. If data is displayed on a background grid as in the accompanying display, we want to ensure that only the point data is detected while the grid is ignored. If the user aims the light pen at a data point that falls directly on a grid line, only the data point will be detected. The orders may be used to program the selective detection technique so that no interrupt to CPU will be generated during specific detects. These orders include:

- SPM - Set Pen Mode (disable detects)
- SPM - Set Pen Mode (enable detects)
- GBC - Graphic Branch Conditional
- GIC - Graphic Interrupt Conditional

HARDWARE CAPABILITY

The 2250 Model 4 is attached to the 1130 system via the Storage Access Channel (SAC) on the 1131 or SAC II on the 1133. This connection permits the CPU core storage to be used as the display buffer for image regeneration without CPU program intervention. The cycle stealing operation of the SAC and priority level assignments of all I/O devices (including 2250) guarantee that the 2250 will not interfere with core storage cycle requirements of disks, printer, or card equipment. The priority level of the 2250 also changes dynamically when a CPU program is running, so that instruction execution can continue while the display is regenerated. The dual load on CPU storage cycles does increase the instruction execution times and may cause slight throughput reduction on some I/O devices and may also cause flicker on the 2250 screen when displaying complex pictures.

CYCLE STEALING AND INTERFERENCE

Attachment of the 2250 Model 4 to the 1130 system via the Storage Access Channel (SAC) permits the operation of the display asynchronously with the CPU. That is, once the 2250 has been started, it continues to execute as a channel program by stealing core storage cycles from the CPU without CPU program intervention. The 2250 will function with either a $3.6\mu s$ or $2.2\mu s$ core storage cycle. The portion of the 1130 core having the 2250 orders representing the image and logical functions becomes essentially a buffer for the 2250. These orders are accessed by the SAC channel (by cycle stealing) and sent to the display up to 40 times per second. Since the 2250, I/O devices, and the CPU are requesting core storage cycles from a single source, some delay must occur. However, the design of the 2250 Model 4 prevents any significant interference with other I/O device operations.

The following are some facts on the cycle stealing operation:

- The central processing unit and certain I/O devices request core storage cycles. If more than one request for a storage cycle is present, the requests are serviced in a fixed order. The order of service for a basic 1130 system is as follows:
 1. Disk
 2. SAC (2250)
 3. Printer
 4. Card reader
 5. CPU

- The lowest-priority cycle steal is the CPU that is executing problem program or I/O interrupt routine instructions. Some devices require interrupt service within a specified length of time. An example of such a device is the card punch interrupt, which must be serviced within 300 μ s before the card moves to the next column. To allow normal card operation, the 2250 will not cycle-steal during the card interrupt routine.
- When the 2250 is drawing vectors, it steals one or two cycles to access the vectors. No cycle stealing will occur when the vector is being drawn.
- The maximum interference of the display cycle stealing on the system occurs when the 2250 is drawing characters.
- The maximum interference on the 2250 caused by other I/O devices and the CPU occurs when the CPU has to continuously process interrupts at levels 0 and 1. This interference will normally not be observed on the display. However, this is a function of the number and lengths of lines and characters displayed.

Table 1 shows the functions being performed in the 1130 system and relates them to the interference that they have on the display. A display regenerated at 40 cps is considered "flicker-free". As regeneration decreases to below 30 cps, there is noticeable flicker. The table indicates that for normal conditions the regeneration is 32 cps or more. The table assumes 3.6 μ s core storage and a complex picture requiring 100% of the CPU cycles to regenerate at 40 cps.

Table 1

1130 Function Requiring CPU	% of CPU Time Required	% Time of CPU Remaining for 2250	Regeneration Rate (cps)
Card reader	2.0	98.0	39
Card punch	2.5	97.5	39
Disk	13.0	87.0	35
300 cps	21.2	78.8	32
5100 cps	2.0	98.0	39

The figures are based on timing figures for subroutines supplied in Type I support.

DISPLAY COMMANDS

The reading and writing of display orders in the buffer storage area of the CPU can be performed by the normal 1130 instructions used to perform core-to-core moves.

The display commands are used to control other functions of the 2250 Model 4 from the 1130 via the Storage Access Channel in much the same way as the 2250 Models 1 and 3 are controlled over the Selector Channel in System/360. The display commands perform the following functions:

- Start regeneration of 2250 display at a specified 1130 core location
- Set indicator lights on optional Programmed Function Keyboard
- Sense to determine source of interrupt from 2250
- Read data required for interrupt handling: address register, X and Y deflection registers, keyboard data, device status word
- Reset all 2250 registers and status indicators as an aid to error recovery
- Perform no-op function when required

DISPLAY INTERRUPTS

All interrupts caused by actions at the 2250 will stop the display image regeneration. Display commands as outlined above are used to determine the source of the interrupt and to read the contents of the 2250 registers into specified 1130 core locations. The three types of interrupts are:

- Light Pen Detect — caused when a light pen detect occurs and detect interrupts are enabled
- Keyboard — caused by a key depression on the optional Alphameric or Programmed Function Keyboard
- Order-Controlled — caused by conditional or unconditional interrupt facilities in the display orders

After the interrupt has been serviced, a display command is used to restart display regeneration.

DISPLAY ORDERS

The display orders, X-Y data, and character strokes reside in the 1130 core storage and are sent to the 2250 via the SAC for decoding and execution. The orders determine the mode of operation (graphic, character, or control), the X-Y data governs the beam motion on the screen for line drawing and point plotting, and the character strokes create the short beam motions required for drawing alphameric. The display order list can be continuously executed (display regeneration) without CPU intervention by means of branching orders and a timer. The timer is required to prevent regeneration from exceeding 40 cps when small quantities of data are being displayed.

The graphic orders control beam motion within an X, Y coordinate system of 1024 by 1024 points which cover a 12" by 12" area on the screen. Point plotting and line drawing are performed in absolute or incremental coordinates. The absolute mode permits drawing a line (beam on) or moving the beam (beam off) between any two points on the 1024 by 1024 grid or plotting a point anywhere on this grid. The incremental mode permits faster drawing and plotting times when lines are short or successive points are closely spaced. The incremental distance from the current beam position can be anywhere within a 127 by 127 grid, considering the current position to be at the center of this grid.

The character mode orders set the 2250 to accept stroke X-Y formats for beam positioning rather than absolute or incremental vector beam motions. In the character mode the X-Y data for each stroke is contained in one byte or a halfword of the 1130 core. Each stroke can position the beam within a 7 by 8 grid of points. However, the spacing between these grid points is greater than the spacing between points on the 1024 by 1024 grid. The resulting character sizes in inches for the basic and large characters are .12" by .16" and .18" by .24". Since a stroke table for each character desired is stored in the 1130 core at execution time and can be changed under program control, the user is free to generate any number of 256-character sets desired for output on the screen. For input via the alphameric keyboard the number of displayable characters recognized by the hardware is 88. This keyboard-input displayable set includes upper- and lowercase alphabetic characters, numerics, and 26 special symbols.

The control orders are used in the display list to control display regeneration and reduce CPU program intervention, especially with respect to light pen detect interrupts and light pen tracking. The control orders fall into two major categories: (1) light pen control orders that enable and disable pen detects, and also enable and defer interrupts when detects occur, (2) branch orders, which provide the capability for regeneration, logical decision making,

character stroke subroutine tables, and subroutining display orders to create multiple images from one set of orders.

Table 2 contains a summary of the display orders. A complete list of all display orders and their functional descriptions can be found in the Appendix.

Table 2. Summary of Display Orders

SVML	Set Vector Mode Line
SVMP	Set Vector Mode Point
SCMB	Set Character Mode Basic
SCML	Set Character Mode Large
SPM	Set Pen Mode
MBI, DBI	Move/Draw Beam Incremental
MBA, DBA	Move/Draw Beam Absolute
MBAX, DBAX	Move/Draw Beam Absolute X
MBAY, DBAY	Move/Draw Beam Absolute Y
MBS, DBS	Move/Draw Beam Stroke
CS	Control Stroke
STMR	Start Regeneration Timer
SRVT	Store Revert Register
RVT	Revert
GNOP	Graphic No-Op
GSB	Graphic Short Branch
GB	Graphic Branch
GBC	Graphic Branch Conditional
GI	Graphic Interrupt
GIC	Graphic Interrupt Conditional

SOFTWARE CAPABILITY

The set of 45 subroutines available for use by the FORTRAN programmer is called the IBM 1130/2250 Graphic Subroutine Package (GSP). This package is not an extension of the FORTRAN IV language, but may be used in conjunction with it. The GSP allows a FORTRAN programmer to create displays on a 2250 Display Unit, Model 4, attached to an 1130 Computing System with disk. These displays can be constructed of lines, points, and characters. The execution of each subroutine is requested by issuing a CALL statement.

A program that uses the GSP includes calls to GSP subroutines in a sequence that causes displays to be produced and provides two-way communication between the 2250 operator and the program.

MACHINE REQUIREMENTS

Programs using the GSP may be executed on any 1130 Computing System with at least 16K and a disk. The graphic device attached to the computing system is the 2250 Display Unit, Model 4. The alphameric keyboard and programmed function keyboard are optional and are necessary only when an application requires them.

LANGUAGE COMPATIBILITY

The facilities provided by the GSP can be used by programs written in the 1130 Basic FORTRAN IV Language or in the 1130 Assembler Language.

STRUCTURE OF A GSP PROGRAM

Preparation of a program that uses the GSP involves the following basic steps:

1. Initialization and definition

- Initializing the GSP
- Initializing an image construction area
- Initializing a generation control area

2. Creating, modifying, and displaying the image entity

3. Establishing communication between the GSP program and the 2250 operator

Initializing the Graphic Subroutine Package

In order to activate the GSP and to establish its environmental characteristics, the first GSP subroutine to be called must be GSPIN, the GSP initialization subroutine. On the basis of the arguments provided by the programmer, GSPIN sets up the control structure needed by almost all other GSP subroutines:

1. Precision specifications, to define the format of integer and variable input data
2. Error returns, to which the GSP returns indications of any errors that might have been detected
3. Unit identification, to establish correspondence between the device address and its logical unit number

Before the programmer can begin creating images by means of the image generation subroutines, he must provide and initialize an image construction area (ICA), which will contain the image entity or entities that he is creating. In this area the input data, converted to 2250 format, is stored for subsequent display.

The ICA is also a control area. It contains, in addition to the image entity, information needed to define the structure of the entity.

Initializing a Generation Control Area

Another area that must be provided and initialized before the programmer can begin creating images is the generation control area (GCA). The GCA contains information needed by image generation subroutines to properly generate elements of the image entity. This information is instrumental in performing the following functions:

- Mapping (scaling and scissoring) input data to 2250 units representing the entire 2250 screen and a smaller area of the screen (grid)
- Converting graphic data (that is, X and Y coordinates in 2250 units and character data in 2250 format) to integer, real, or character arrays in program format which can be used by nongraphic FORTRAN statements.
- Allowing selection of input data from arrays by means of an index factor rather than sequentially

- Allowing specification of successive X and Y coordinates by means of an increment value instead of requiring an input data array to contain them.
- Specifying real or integer and absolute or incremental input data mode and absolute, incremental, or optimized graphic output mode

More than one GCA may be defined for a single GSP program. For example, different grid areas of the 2250 screen may be designated, and different scaling factors (for example, data range to grid area) may be specified, etc.

Creating, Modifying, and Displaying an Image Entity

Once the communication paths to GSP have been initialized (that is, GSPIN has been called and both an ICA and at least one GCA have been initialized), the programmer can begin defining an image entity. An image entity has three major characteristics:

1. It consists of one or more elements.
2. It has a structure.
3. It can be displayed.

The structure of an image entity is defined by a series of calls to the image management subroutines, while the elements of an image entity are defined by a series of calls to the image generation subroutines. In general, an element is defined by a call to an image management subroutine to define its starting point, one or more calls to the image generation subroutines to define its content (points, characters, lines, etc.), and finally a call to an image management subroutine to define its end point. For some elements a single call suffices to completely define the element.

The image generation subroutines are used to define the content of an element by converting the input data into 2250 format.

The following is a list of all the image generation subroutines and their functions:

<u>Mnemonic</u>	<u>Name and Function</u>
GCAIN	Generation Control Area Initialization -- initializes a GCA with standard values
SSCAL	Set Scaling Information -- sets or resets the scaling information in the GCA
SINDX	Set Index Values -- sets or resets the array indexing values in the GCA
SINCR	Set Increment Values -- sets or resets the increment values in the GCA
SDATM	Set Input Data Mode -- sets or resets the input data type for absolute or incremental, real or integer
SGRAM	Set Output Graphic Mode -- sets or resets the desired output data type for absolute, incremental, or optimized
SSCIS	Set Scissoring Option -- sets or resets the scissoring to grid boundaries or screen boundaries
MVPOS	Move Beam to a Position -- establishes the starting coordinates for the next element
IDPOS	Indicate Beam Position -- indicates the starting coordinates of the next element for proper scissoring after use of a linkage entity or when modifying an existing element
PLINE	Plot Line(s) -- converts input data to 2250 format for plotting lines
PPNT	Plot Point(s) -- converts input data to 2250 format for plotting points
PTEXT	Plot Text -- converts input data to 2250 format for plotting characters
PSGMT	Plot Line Segment(s) -- converts input data to 2250 format for plotting line segments
LKSUB	Linkage to a Subroutine -- establishes linkage to a subroutine entity
PGRID	Plot Grid Outline -- generates 2250 format data to plot the rectangular outline of the grid
PCOPY	Plot Copy of an Element -- generates a copy of an element at some other point in an image or subroutine entity

The image management subroutines are used to define and control the structure of an image or subroutine entity.

The following is a list of all the image management subroutines and their functions:

<u>Mnemonic</u>	<u>Name and Function</u>
ICAIN	Image Construction Area Initialization -- initializes an area for use by the GSP to contain the image entity and its control information
BELMT	Begin Element -- defines the beginning of a controlled, uncontrolled, subroutine, or image entity
EELMT	End Element -- defines the end of an element whose beginning was specified by BELMT, BUPDT, or XUPDT
BUPDT	Begin Update -- identifies an element whose content is to be completely modified by subsequent calls to image generation subroutines
XUPDT	Extend Update -- identifies an element whose content is to be extended by subsequent calls to image generation subroutines
DELMT	Delete Element -- identifies an element that is to be completely deleted from the image entity
STPLV	Step Verification Level -- modifies the level at which verification is to take place and the level of the correlation value to be returned on a light pen attention (see "The Controlled Entity" at the end of this section on software)
SATRB	Set Entity Attributes -- resets one or more of the attributes of a controlled entity
EXEC	Execute Display -- performs the processing required to display an image entity
TMDSP	Terminate Display -- performs the processing required to stop the display of an image entity
GXIT	2250 Exit -- provides a program exit and a dump of core storage

For example, to display a single circle using a series of points, the programmer might call the following series of GSP subroutines:

<u>Routine Called</u>	<u>Reason</u>
GSPIN	GSP initialization
ICAIN	ICA initialization
GCAIN	GCA initialization
BELMT	Define beginning of an image entity
PPNT	Generate the series of points in 2250 format to display the circle
EELMT	Define the end of the image entity
EXEC	Display the image entity on the 2250

This series of calls defines a single element, the image entity, whose content is a circle.

Now suppose the programmer wishes to change the display from that of a circle to that of a square. He might use the following series of routines:

<u>Subroutine Called</u>	<u>Reason</u>
TMDSP	Stop the display of the image entity for updating purposes
BUPDT	Identify the element to be changed
MVPOS	Position the beam to the corner of the square from which the lines will begin
PLINE	Generate the four sides of the square in 2250 format
EELMT	Define the end of the new contents of the element
EXEC	Display the image entity, now a square

Communication between a GSP Program and the 2250 Operator

The IBM 2250 Display Unit is a powerful tool for two reasons:

1. It can quickly display a large amount of data.
2. The 2250 operator can communicate and guide the running program by use of the light pen or keyboards.

Up to this point, this section has been concerned with how to format and display data. The following paragraphs deal with the four major areas in the operator-to-program communication:

1. Attention handling
2. Using the alphameric keyboard
3. Using the light pen
4. Using the programmed function keyboard

Attention Handling

An "attention" is a signal generated when the 2250 operator presses a programmed function key, presses an alphameric keyboard key, or points at something with the light pen. The attention from the device causes the CPU to interrupt and enter a GSP routine whose function is to gather all the available information about the attention and provide the information when requested.

The attention handling subroutines of the GSP are used to establish which attentions are to be processed and which are to be ignored in order to obtain attention information.

The following are the attention handling subroutines and their functions:

<u>Mnemonic</u>	<u>Name and Function</u>
SATNS	Set Attention Status — defines the attentions that the program will process
RQATN	Request Attention Information — requests that available attention information be formatted and provided to the program
ROCOR	Return Outer Correlation Value — identifies the element within which another element is nested

Four attention sources are available to the GSP programmer:

1. Light pen
2. END key
3. Alphameric keyboard (except END and CANCEL keys)
4. Programmed function keyboard

The attention information returned is different for the different sources.

The programmer determines which attention sources are meaningful to his program. He enables those and disables all others. At any point in the program he may change sources from enabled to disabled and vice versa.

When the program reaches a point that requires the 2250 operator to specify or have specified some information by means of an attention, the program issues a call to RQATN. RQATN formats any available attention information and places it into a programmer-defined array. The program can then determine whether an attention has occurred and, if so, whether it is the correct one. If no attention has occurred, the program might be able to continue processing. If an attention is required before further processing can take place, the program can issue a PAUSE followed by a GO TO to the RQATN call.

If an attention has occurred and it is the correct one, the program continues processing. If the attention is incorrect, the program can simply ignore it and reexecute RQATN or notify the 2250 operator that his attention is inappropriate.

At some point in the program it may be desirable to perform some unique function that requires the 2250 operator to respond with a series of attentions. However, the present attention status is to be reestablished following performance of this function. This may be accomplished by saving the argument used in the last call to SATNS. Another call to SATNS with a different argument establishes the new attention handling environment. The unique function can now be performed, and a final call to SATNS with the saved argument reestablishes the attention handling environment as it existed just before the unique function was performed.

In the previous examples of a circle and a square, an attention could have triggered the change in display. The following might have been the series of calls issued:

<u>Subroutine Called</u>	<u>Reason</u>
GSPIN, ICAIN, GCAIN	Initialization
BELMT	Define the beginning of the image entity
BELMT, MVPOS, PLINE, EELMT	Define a circle as a controlled entity in order to be light-pen-detectable and define the end of the image entity
SATNS	Enable light pen attentions
EXEC	Display the image entity
RQATN	Wait for the light pen attention on the circle, which indicates that the square is to be displayed
BUPDT, MVPOS, PLINE, EELMT, EXEC	Change the display to the square
RQATN	Wait for the PF keyboard attention to terminate the display
TMDSP	Terminate the display
GXIT	Terminate program execution

Using the Alphameric Keyboard

The alphameric keyboard provides for the means of communicating characters (alphabetic, numeric, and special) to the program. The characters could be codes or text, depending on their use by the program.

The following is a list of the alphameric keyboard and message-collection subroutines and their functions:

<u>Mnemonic</u>	<u>Name and Function</u>
DFMSG	Define Message Entity — defines an element into which alphameric keyboard characters will be placed
MSGIN	Message Entity Initialization — initializes a message entity with text
ICURS	Insert Cursor — places a cursor into a message entity and establishes the message-collection mode
RCURS	Remove Cursor — deletes the cursor and terminates the message-collection mode
TLMSG	Translate Message Data — converts alphameric data in a message entity to EBCDIC format for manipulation by the FORTRAN program

The programmer defines one or more message entities in his program and may initialize them with some text, such as instructions to the 2250 operator. By issuing a call to ICURS, the program enters message-collection mode. The alphameric keyboard attentions are no longer passed to the program, if the alphameric keyboard was enabled, but are sent to the GSP message-collection routine. When the 2250 operator presses a key, the character is placed into the message entity defined in the call to ICURS, and the cursor character advances one position. The character will appear on the screen. The 2250 operator can key in as many characters as the message entity can hold. Once the message entity is filled, all other characters keyed in will replace the last character until the JUMP key is depressed. When the GSP message-collection routine receives a jump signal, the cursor is moved to the next available message entity, and the 2250 operator can continue keying in more characters. He enters as many characters as necessary and signals that he is finished by pressing the END key.

When the programmer decodes an END key attention after a call to RQATN, he issues a call to RCURS to terminate the message-collection mode and then a call to TLMSG to convert the characters from 2250 format to EBCDIC for further manipulation.

Correlation value. In defining the elements of an image to the GSP, the programmer may require the facility of referring to a particular element at some later point in the program. This facility is provided by allowing an argument

called a "correlation value" to be specified when the element is defined. At any point in the program after the element is named (that is, assigned a correlation value), this name may be used as an argument to identify that element.

Using the Light Pen

The light pen is used to communicate one of at least two things to the program:

1. Identification of a particular element of the image entity being displayed
2. A series of X and Y coordinates defining one or more points or lines

The SATNS subroutine is used to enable or disable light pen attentions. If enabled, certain information is returned on each light pen attention. In particular, the correlation value of the element pointed at with the light pen is returned. The program can then manipulate or modify the element according to the application.

In addition to the above, a series of GSP light pen subroutines is provided to allow for data input. The following is a list of the light pen subroutines and their functions:

<u>Mnemonic</u>	<u>Name and Function</u>
MOVPN	Move the Position of the Light Pen — used to define a particular X and Y coordinate on the screen
LOCPN	Locate the Position of the Light Pen — used to identify the X and Y coordinate of the point being defined by the light pen
LOCND	Locate the Position of the Light Pen on No Detect — used to find the X and Y coordinate of the light pen if no attention occurred on a controlled entity during the display of the image entity
TRKPN	Track the Position of the Light Pen — defines a series of X and Y coordinates
PNCTL	Control Light Pen Tracking — changes light pen tracking status set by a prior call to TRKPN or PNCTL
DISPE	Disconnect Pen Entity — disconnects the pen entity from the temporary linkage in an image entity
CVTPN	Convert Light Pen Data — converts X and Y coordinates from 2250 format to integer or real FORTRAN format

LOCPN and LOCND cause a scanning pattern to be produced on the screen until a light pen detect occurs. When the detect occurs the scanning pattern disappears and the data about the light pen attention may be accessed by a call to RQATN.

MOVPN and TRKPN display a special symbol called a "tracking symbol". For MOVPN the 2250 operator places the light pen on the tracking symbol; as he moves the light pen across the screen, the tracking symbol moves. When the tracking symbol is at the point that the 2250 operator wishes to define to the program, he closes the switch on the pen and the tracking symbol disappears. The attention information contains the coordinates of the center of the tracking symbol and can be accessed by RQATN.

TRKPN uses the tracking symbol in much the same way as MOVPN, but in this case an entire pen entity is being defined as the operator identifies X and Y coordinates with the light pen. A call to TRKPN causes the tracking symbol to appear on the screen. The 2250 operator places the light pen on the tracking symbol, closes the switch, and begins defining X and Y coordinates.

As the tracking symbol is moved by the 2250 operator, the points being defined are placed into a pen entity, and the defined points or lines appear on the screen. The 2250 operator signals the program when he wishes to change from continuous sketching to rubber-banding -- that is, to defining points or straight lines, and vice versa. The 2250 operator must also signal the program that he has defined all the desired points and lines. The program then calls PNCTL to terminate light pen tracking by removing the tracking symbol from the screen. The program next calls CVTPN to convert the X and Y coordinates from 2250 format to a FORTRAN data format for further manipulation.

The controlled entity. The controlled entity has two controllable attributes: visibility and detectability. Visibility refers to the property that the controlled entity is part of the image entity, but may or may not appear on the screen when the image entity is displayed, depending on whether its visibility attribute is set to display or non-display. This attribute provides the means to generate elements and selectively display them on the screen.

Detectability refers to the property that the element can or cannot be identified by means of the light pen, depending on whether or not the detectability attribute is set to detect or non-detect. This provides for easily ignoring meaningless light pen detects. That is, on elements that the programmer does not wish to receive a light pen attention, he sets the controlled entity attribute as non-detect.

The attributes can be set or reset any time after the controlled entity is completely defined.

Using the Programmed Function Keyboard

The programmed function keyboard (PFKB) provides one of 32 signals to the program. The meaning of each signal depends strictly on the individual application program unless certain installation standards apply.

The programmed function keyboard is either enabled or disabled. If it is enabled, any key depressed causes attention data to be available by a call to RQATN. In addition to the key number, the overlay number is also returned with the PFKB attention. Since the overlay number is in the range 0-255 inclusive, there are actually 32 x 256 or 8192 unique signals available from the PFKB.

A subroutine is provided to enable the program to selectively light the programmed function keys. This subroutine, SPFKL, provides the programmer with the means of indicating to the 2250 operator which of the PFKB keys are meaningful at any particular point in the program. SPFKL will set on any or all of the 32 key lights while turning off those specified by the program.

APPLICATIONS

From the standpoint of configurations using the 1130/2250 system, we may separate applications into two distinct areas:

- For the 1130/2250 when used as a subsystem attached to System/360
- For the 1130/2250 when used as a stand-alone system

For some applications, either of these environments may be suitable for the solution of the problem. However, the choice of the system is highly dependent on (1) the extent of the application, (2) the computational requirements, (3) the general relationship of the application to a data base, and (4) the economics of the application on a small system (1130/2250 stand-alone) versus a large system (System/360 with an 1130/2250). Of course, other factors, such as equipment already installed, operational philosophy of the customer, software availability, etc., strongly influence the choice and use of the system. With these factors in mind, the following application discussion is divided into three parts:

1. Graphic application areas
2. Applications on the 1130/2250 stand-alone system
3. Applications on the 1130/2250 subsystem attached to System/360

GRAPHIC APPLICATIONS

The range of applications for graphics spans from simple data reduction and retrieval programs to complex engineering design and numerical control applications. The most common present-day applications are in engineering analysis. The engineering analysis programs are typically those that require many iterations in the solution of the problem. The key advantage of using graphic equipment in these applications lies in the fact that we achieve faster turnaround at the graphic console. That is, many iterations and decisions can be made at one console session rather than in the hours or days required in a normal batch system. A listing of typical applications in various industries can be found in Table 3.

In looking at the existing applications, we see that:

1. Faster turnaround time is an essential part of the application.
2. Graphic representation assists in the performance of the application. An example would be curve fitting or engineering drawings where graphic representation is a must.
3. The best talents of the man (experience, judgment and know-how) are combined with the computer capabilities (low cost per calculation, high speed, etc.).

Conceptually, we look at each application and try to leave to the user that which he can do best--namely, use his engineering judgment and know-how. We also leave to the computer those things which it can do best--namely, high speed calculations, accuracy, checking, etc. Normally, the justification of an application must be derived from the total turnaround time from start of a problem to completion. Figures 8-10 illustrate these points.

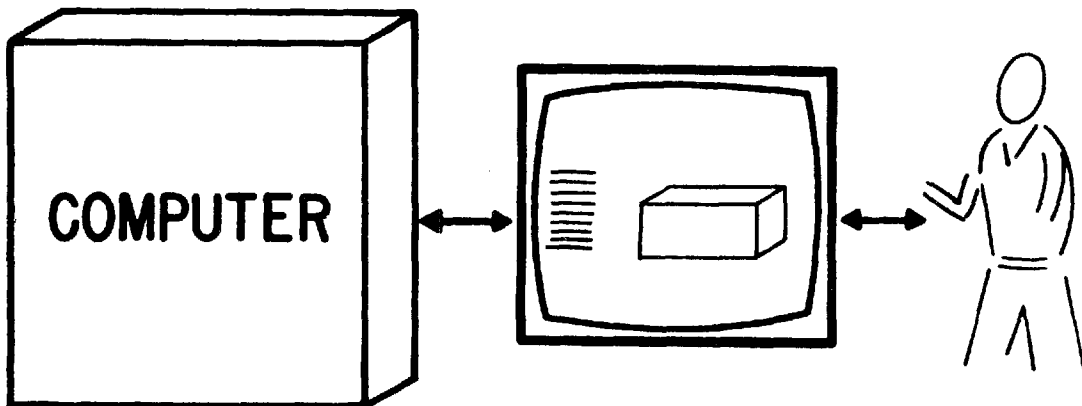


Figure 8. Man-computer conversation -- a common language of graphics and alphamerics

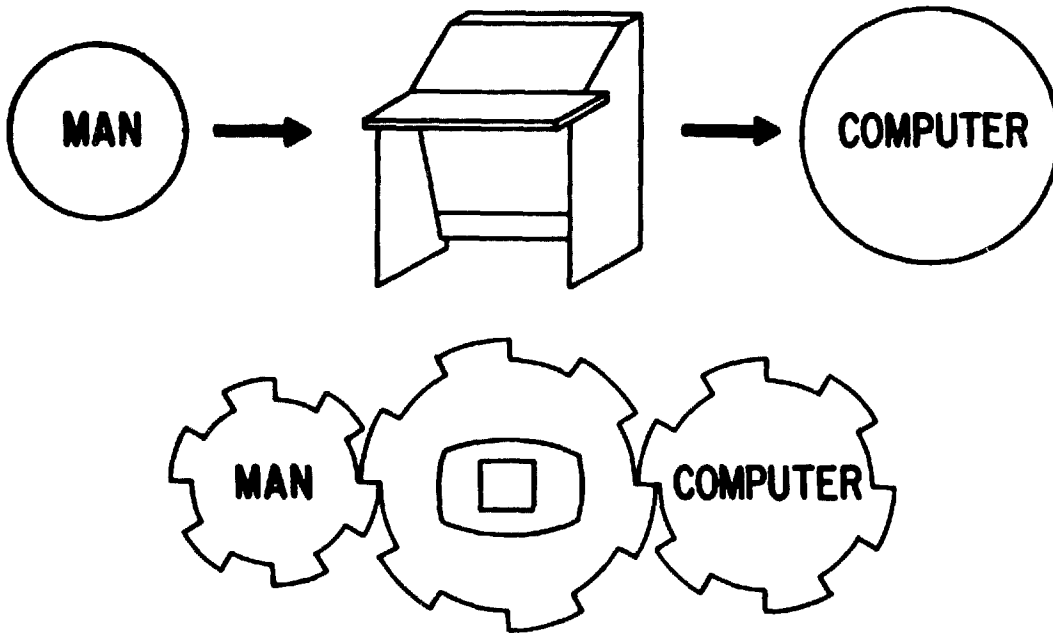


Figure 9. Man-machine interaction in problem solving

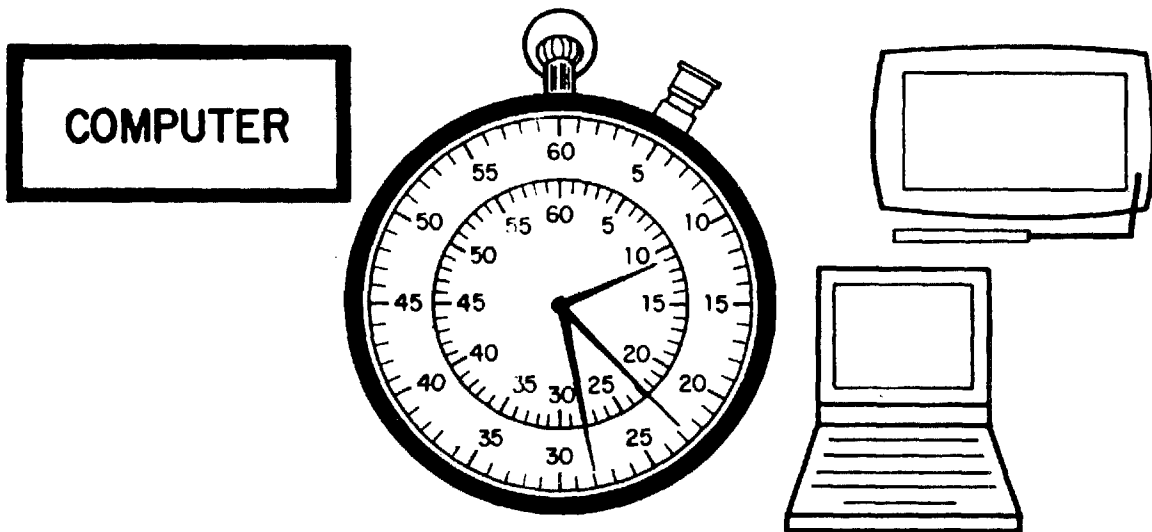


Figure 10. Instantaneous man-machine communication

TABLE 3. GRAPHIC APPLICATIONS IN INDUSTRY (1 of 7)

APPLICATION	INDUSTRY												
	Automotive	Electrical	Mechanical	Construction	Aerospace	Process	Universities	Communications	State & Local Govt.	Utilities	Medical	Consultants and Service Bureaus	Federal
Beam Design			x	x				x			x		
Cam, Gear, [etc.] Design	x	x									x		
Design of Process Equipment				x	x						x	x	
Process System Analysis				x	x							x	
Batch Process Simulation					x								
Unit Operations					x								
Data Correlation	x	x	x	x	x	x	x	x	x	x	x	x	x
Aircraft Course Plotting				x									
Ship Course Plotting												x	
Missile and Satellite Plotting				x									
Cartography Plotting				x		x							
Flight Test Recording				x									
Telemetry Data Plotting				x		x		x					
Linear Programming	x		x	x	x	x	x	x	x	x	x	x	x
Regression Analysis	x	x	x	x	x	x	x	x	x	x	x	x	x

TABLE 3. GRAPHIC APPLICATIONS IN INDUSTRY (3 of 7)

APPLICATION	INDUSTRY												
	Automotive	Electrical	Mechanical	Construction	Aerospace	Process	Universities	Communications	State & Local Govt.	Utilities	Medical	Consultants and Service Bureaus	Federal
Cable Layout		x			x			x		x			
Network Analysis		x			x			x	x			x	x
Teaching Machine							x						
Photographic Analysis							x		x		x		
Pattern Recognition					x		x	x	x				
Data Reduction	x		x		x	x	x		x		x	x	
Chemical Plant Simulation						x						x	
Mathematical Research					x		x				x	x	
Curve Fitting	x	x	x		x	x	x	x	x	x	x	x	x
Geometrical Problems	x		x	x	x	x	x	x	x	x		x	x
Servomechanism Design	x	x	x	x	x	x	x	x	x	x		x	x
Surface Design	x		x		x				x			x	x
PERT Diagrams	x	x	x	x	x	x	x	x	x			x	x
Verification of N/C Tapes	x		x		x							x	x

TABLE 3. GRAPHIC APPLICATIONS IN INDUSTRY (4 of 7)

APPLICATION	INDUSTRY												
	Automotive	Electrical	Mechanical	Construction	Aerospace	Process	Universities	Communications	State & Local Govt.	Utilities	Medical	Consultants and Service Bureaus	Federal
Bridge Design				x					x				
Reactor Design								x					x
Heat and Material Balancing					x	x				x		x	x
Dynamic Response					x	x							
Chemical Equilibrium						x				x			x
Bubble Chamber Film Analysis								x					x
Velocity Gradients in Fluid Flow						x	x		x	x			
Optimization Studies	x	x	x	x	x	x	x	x	x	x		x	x
Statistical Analysis	x	x	x	x	x	x	x	x	x	x	x	x	x
Flutter Analysis					x							x	x
Wind Tunnel Studies					x		x						x
Search Pattern Modification					x	x	x						x
Scanning X Rays											x		
Scanning EKG											x		
Medical Research											x		

TABLE 3. GRAPHIC APPLICATIONS IN INDUSTRY (5 of 7)

APPLICATION	INDUSTRY												
	Automotive	Electrical	Mechanical	Construction	Aerospace	Process	Universities	Communications	State & Local Govt.	Utilities	Medical	Consultants and Service Bureaus	Federal
Well Logs					x								
3-Dimensional Wire Routing	x	x		x	x		x		x		x	x	
3-Dimensional Line (piping, etc.) Routing			x		x		x		x		x	x	
Mechanical Parts Drawing	x		x	x									
Plans and Elevation Diagrams			x		x								
Structural Steel Detailing			x	x	x			x					
Concrete Detailing			x					x					
Heat and Ventilation Diagrams				x	x								
Power Supplies Diagrams		x	x	x	x								
Highway Cross Section Plotting								x	x				
Cut and Fill for Highways								x					
Subdivision Mapping			x					x					
Truss Analysis			x					x					
Stress Analysis	x	x	x	x	x	x		x					
Cross Sectional Drawings	x	x	x	x	x			x					

TABLE 3. GRAPHIC APPLICATIONS IN INDUSTRY (7 of 7)

APPLICATION	INDUSTRY												
	Automotive	Electrical	Mechanical	Construction	Aerospace	Process	Universities	Communications	State & Local Govt.	Utilities	Medical	Consultants and Service Bureaus	Federal
Motor design	x	x	x		x								
Transformer design		x											
Numerical Control Programming (via light pen and console)	x		x		x								
Petroleum Pipeline Monitoring & Dispatching						x				x			
Plant Optimization (Cloth, Steel, Aluminum - through Stock Display and Pattern Manipulation.)	x		x	x	x	x							
Traffic analysis through display and manipulation of traffic maps and simulated									x				
Aircraft Weight Distribution Analysis					x								
Panel Point Loading					x								
Scale Design		x	x										
Recorder Design		x											

APPLICATIONS FOR THE STAND-ALONE SYSTEM

The easiest way in which to install graphic applications on the 1130/2250 is to take existing programs and build graphics directly into them. Many of the existing programs can readily be improved with graphics capability. These programs can be modified to suit the specific needs of each customer. Normally, we are using the graphic console for input and editing, examining intermediate results in iteration-type problems, displaying answers in graphic form, and providing fast response in the decision-making process. Simply stated, we are building graphics into the front of the program, in the middle, and at the end. In using the GSP for the applications, you would normally require a minimum of 16K words in the 1130/2250 system. Applications with any high degree of complexity would normally require additional core. If an application already exists on the 1130 without graphics, an increase of 8K in core requirements is probably needed to build graphics into the program. The reader should refer to other IBM publications on graphic data processing for further information on this subject.

APPLICATIONS FOR THE 1130/2250 ATTACHMENT TO SYSTEM/360

In the 1130/2250 system attached to the System/360, there are application functions performed in both systems. Normally, we would do as much as possible in the 1130/2250 and use the System/360 only when computational requirements exceeded those of the 1130 system. In this environment, the load on the System/360 is significantly reduced in a given application. In addition, the 1130/2250 can be remotely placed relative to the System/360. This allows users who may be located away from the System/360 to utilize its power in the solution of their problem.

As previously stated, the application areas for this type of system are essentially those that can be performed in the stand-alone system. In addition, the 1130/2250 subsystem approach can be utilized for the most complex application areas.

APPENDIX: DESCRIPTION OF DISPLAY ORDERS

GRAPHIC MODE

SVML--Set Vector Mode Line

A one-word order that causes the display unit to enter a mode for incremental and/or absolute line drawing. The X-Y data that follows this order can be a random mixture of absolute and incremental data.

SVMP--Set Vector Mode Point

A one-word order that produces the same result as SVML for point plotting.

MBI, DBI--Move/Draw Beam Incremental

One-word orders that also contain ΔX , ΔY data for moving the beam +63 or -64 raster units from the present X, Y position. "Move" indicates beam off; "Draw" indicates beam on. In the incremental mode the beam position is retained for up to one screen width (1024 units) off the working screen area in all directions. Lines that are not totally visible in the 12" x 12" working area are completely blanked.

MBA, DBA--Move/Draw Beam Absolute

Two-word orders that also contain X, Y data for moving the beam (on or off) from the present position to anywhere within the 1024 x 1024 working area grid of points using absolute coordinates.

MBAX, MBAY, DBAX, DBAY--Move/Draw Beam Absolute X/Y

One-word orders that also contain X or Y data for moving the beam (on or off) along the X or Y coordinate only from the present beam position. These orders use absolute coordinates for X and Y, and permit the beam to be moved completely across the screen vertically or horizontally with data packed in one 1130 word.

CHARACTER MODE

SCMB--Set Character Mode Basic

A one-word order that causes the display unit to enter a mode for displaying basic size characters. When the display is in this mode, all data following a graphic branch order is handled as character strokes. The character mode is terminated only by a nonbranch/noninterrupt order.

SCML--Set Character Mode Large

A one-word order that causes the display unit to enter a mode for displaying large size characters. (See SCMB for other details.)

MBS, DBS--Move/Draw Beam Stroke

A one-word order that contains X, Y data for two character strokes. The beam is moved (on or off) from the present position to the new X, Y specified within the 7 by 8 grid of points used to create a character. A revert bit can be set in the last stroke of a character to cause an automatic branch back to the main display list, and automatic spacing to the next character position.

CS--Control Stroke

A one-word order containing one of five special control characters that provide control functions as follows:

- Subscript*-- The center point for the next character is offset down 3 units.
- Superscript-- The center point of the next character is offset up 3 units.
- New Line-- The beam is positioned to the extreme left X position and stepped down to the next line Y position.
- Null Character-- Spacing to the next character is suppressed and revert to main line display list is automatic.
- No-Operation-- A one-word no-op control stroke that produces no effect on display operation.

PEN CONTROL MODE

SPM--Set Pen Mode

A one-word order that causes the display unit to enter one of several modes for light pen operations. The options available within this order are:

- Enable light pen detects
- Disable light pen detects
- Enable light pen interrupts
- Defer light pen interrupts

*After execution of a superscript order (if revert bit is not set), execution continues with the second word after the superscript order. This allows the user to have a subscript order after the superscript order and thus requires only one character stroke subroutine to serve for superscript, subscript, or normal mode depending on the entry point to the subroutine.

Valid combinations of the above modes can be set within one Set Pen Mode order. Thus enable detects and enable interrupts can be set with one order as well as enable detects and defer interrupts. The deferred detects that result from the last combination can be tested by the Graphic Interrupt Conditional order.

GENERAL CONTROL MODE

STMR--Start Regeneration Timer

A one-word order that sets and tests a 25-millisecond timer. The order causes the timer to be tested to determine whether or not it has run down to zero. If zero, the timer is reset to 25 ms; if nonzero, the order stops regeneration until the timer does reach zero, then resets the timer and allows regeneration to continue. This timer guarantees a maximum regeneration of 40 cps, which protects the phosphor coating on the CRT. Higher regeneration rates due to small quantities of data on the screen can damage the CRT if STMR is not executed every display cycle. This order has no effect on displaying large quantities of data, which may take more than 25 ms per display cycle.

SRVT--Store Revert Register

A one-word order permitting multiple levels of display order subroutines. When a graphic branch order is executed, the next core address is stored in the Revert register. If additional branches are to be executed before going back to the main line display list, the contents of the Revert register can be stored in the second word of the SRVT order. A Graphic Branch Conditional (Indirect) is used to work back to the main line display.

RVT--Revert

A one-word order permitting single-level subroutining without using SRVT. The Revert order results in an unconditional branch to the core address presently stored in the Revert register.

GNOP--Graphic No-Op

A one-word order that has no effect on the display unit. The order can be used to reserve a single word in the display list for later modification.

GSB--Graphic Short Branch

A one-word order permitting a branch to a location in the first 8K of 1130 core storage. GSB can be used to save core storage when branching to frequently used subroutines, such as character stroke tables.

GB--Graphic Branch

A two-word order permitting an unconditional branch to any 1130 core location directly or indirectly. Direct branch uses the address in the second word of the order as the next address in the display list. Indirect branch uses the contents of the address in the second word as the next address in the display list.

GBC--Graphic Branch Conditional

A two-word order permitting a conditional branch, directly or indirectly, to any 1130 core address. The conditions specified are:

- Pen switch closed
- Pen detect
- Pen detect and switch closed
- Unconditional

If the condition is met, the branch occurs; otherwise, the order is a two-word no-op.

GI--Graphic Interrupt

A two-word order permitting an unconditional interrupt and stopping display regeneration. The second word can be used to store a symbolic address, number, or expression for any purpose the user desires.

GIC--Graphic Interrupt Conditional

A two-word order permitting a conditional interrupt. The conditions are the same as those for GBC. If the condition is met, the interrupt occurs; otherwise, the order is a two-word no-op. The second word is available for general purpose data at the user's option.