

Errata

Document Title: Using the Configurable Graphics Memory in Your
HP 7550A, 7580B, 7585B, or 7586B Plotter (AN 229-10)

Part Number: 5953-9883

Revision Date: August 1985

HP References in this Application Note

This application note may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this application note copy. The HP XXXX referred to in this document is now the Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

About this Application Note

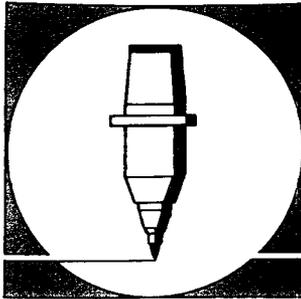
We've added this application note to the Agilent website in an effort to help you support your product. This manual provides the best information we could find. It may be incomplete or contain dated information, and the scan quality may not be ideal. If we find a better copy in the future, we will add it to the Agilent website.

Support for Your Product

Agilent no longer sells or supports this product. You will find any other available product information on the Agilent website:

www.agilent.com

Search for the model number of this product, and the resulting product page will guide you to any available information. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available.



Using the Configurable Graphics Memory in Your HP 7550A, 7580B,* 7585B,* or 7586B Plotter

What's inside this document

This note will help you understand and use a special feature within your plotter, the configurable graphics memory. You will learn when and how to allocate memory, the best use of each buffer, and how buffers work together to produce complex plots. Included are hints and sample programs which illustrate these concepts.

In this note, the term "drafting plotters" refers to the free-standing HP 7580B, 7585B, and 7586B plotters. "Graphics plotter" refers to the desktop model, HP 7550A. Although most information applies to both plotter types, watch for information that applies only to drafting plotters or to the graphics plotter.

All sample programs and discussions assume that you know how to send HP-GL and device-control instructions and that you understand their syntax. This note is intended to supplement your plotter's Interfacing and Programming Manual. Refer to that manual when you have additional questions about HP-GL instructions and their parameters.

Section 1: What Is the Configurable Graphics Memory?

The memory uses unique buffers

The configurable graphics memory is an area in the plotter reserved for temporary data storage. This storage area is divided into separate buffers, each with a unique function (buffer functions are described later in this section):

- Physical input/output buffer
- Polygon buffer
- Downloadable character buffer
- Replot buffer (graphics plotter only)
- Vector buffer (graphics plotter only)

*The configurable graphics memory is available on HP 7580B and 7585B plotters with a serial prefix of 2402 or higher.

*The memory is flexible,
to suit your needs*

You control the amount of data stored in each buffer. In this way, you can customize memory allocation to fit your plotting needs.

Think of the configurable graphics memory as a house with several rooms. Each room has a different function such as living room, dining room, kitchen, or bedroom. You decide how to furnish each room to suit your living needs and how to change furnishings when you plan a party or host guests from out of town. Similarly, if one plot includes complex polygons, and another requires specially designed characters, you can rearrange the graphics memory for each plot.

The concept of a configurable graphics memory is new and unlike other types of plotter memory. In the past, a plotter's internal memory was limited to input and output data storage (the equivalent of your plotter's physical I/O buffer). It would be analogous to a one-room house, limited in both storage and living flexibility.

What Can This Memory Do for You?

With the configurable graphics memory, you can tailor the available buffer space to your particular plotting needs. You may allocate a large storage area to a single buffer or a combination of buffers. For example, since each buffer stores different information, you could use one buffer to define a complex polygon and another to store your own character set, both at the same time.*

*Simplify the program-
ming task*

Any programmer appreciates a tool that simplifies the programming task. Try incorporating the HP-GL instructions that work with the buffers. You will write shorter, simpler programs that produce complex plots. For example, plotters without a configurable memory need software to generate unusual characters. Software characters contain many pen moves and defining these vectors increases program length and complexity. With a configurable memory, once you have used HP-GL instructions to define and store characters in a buffer, any program can retrieve these characters and use them repeatedly in labeling instructions.

*Use the plotter's "hidden
intelligence"*

The configurable memory also expands your plotter's intelligence. Let's assume that you want to define and fill a large polygon. With the configurable memory, your program can send vertices as parameters attached to HP-GL polygon instructions. These points are stored in the polygon buffer and accessed by HP-GL edge and fill instructions. The plotter assumes the burden of calculating each endpoint and drawing the polygon. Without a configurable memory, the program must calculate each endpoint, avoid windows, and use a complex algorithm to fill the polygon.

Which Buffers Does Your Plotter Use?

Both total memory capacity and the number of buffers differ between drafting plotters and the graphics plotter. Study the following section that pertains to your particular model to discover which buffers your plotter uses.

Drafting Plotters (HP 7580B, 7585B, and 7586B)

The configurable graphics memory in drafting plotters has a maximum of 18432 bytes available for data storage. This memory is divided between three buffers: physical I/O buffer, polygon buffer, and downloadable character buffer. Note that the drafting plotter contains neither the replot buffer nor the vector buffer. Also note that the combined default values of all buffers represent only a portion of the

*Multiple buffer use depends on a plot's complexity and the amount of storage space available to each buffer.

total memory capacity available. Therefore, you can increase the size of one or more buffers up to the memory limit. Buffer default values are provided in Section 2.

Graphics Plotter (HP 7550A)

The graphics plotter has a total storage capacity of 12 800 bytes in the configurable graphics memory. This memory includes five buffers: physical I/O buffer, polygon buffer, downloadable character buffer, replot buffer, and vector buffer. The combined default values of all five buffers uses the entire memory storage space. So, if you want to allocate extra bytes to one buffer, you must “borrow” storage space from another buffer to avoid exceeding the memory limits. Buffer default values are provided in Section 2.

What Does Each Buffer Do?

Buffers use HP-GL instructions to store and/or retrieve plot information

The **physical I/O buffer** provides an area for all incoming graphics (HP-GL) instructions on a first-in/first-out basis. The computer transfers blocks of instructions into this area.

The **polygon buffer** stores the vertices of a polygon defined by HP-GL instructions. Once these points have been stored, edge or fill polygon instructions retrieve these points and draw the polygon at a location you specify.

The **downloadable character buffer** will store from 1 to 94 characters that you design using the DL instruction. You can store a special alphabet, type style, logo, or any symbol not included in the plotter’s character sets. Once characters have been stored, the LB instruction retrieves these characters from the buffer and uses them to label your plot.

The **replot buffer** uses the BF instruction to store the graphics instructions in your plot. Once they are stored, the RP instruction retrieves these graphics instructions from the buffer, and causes up to 99 original plots to be drawn automatically without rerunning your program.

The **vector buffer** responds to the CV instruction which collects vectors in the buffer to be plotted as a group. Once the vectors are collected, the pen plots in a continuous motion, rather than starting and stopping at each vector endpoint. As a result, curves appear smoother. The effects are most noticeable on curves with short vectors, such as characters.

Section 3 will tell you more about buffers: when to change a buffer’s size, how much memory to allocate, and how each buffer uses HP-GL instructions.

Section 2: When and How to Allocate Memory

Allocate memory when your program uses specific buffers

Your plotter uses the configurable graphics memory with every plot. Although many plots can be run successfully without changing memory allocations, certain HP-GL instructions were designed to work with specific buffers. A buffer must have sufficient memory to store these instructions and data points. Generally, change memory allocations when:

- You want to minimize computer communication time by transferring large data blocks to the plotter (Physical I/O buffer).

- Your plot contains a complex polygon with many vertices (Polygon buffer).
- You need special symbols or characters not included in your plotter's character set (Downloadable character buffer).
- You want to generate multiple original plots without rerunning the graphics program (Replot buffer).
- You want to smooth curved lines or characters that show ink blotting at each vector's endpoint, or anytime your application requires maximum curve quality (Vector buffer).

Allocating memory is a two-step process. First, your program should send the plotter device-control instructions that control memory allocation. Once storage area has been created, then send HP-GL instructions that use the buffers to store or access plot information.

Step 1 — Allocate Memory

Use the ESC . T instruction to allocate memory

The three device-control instructions* listed below control memory allocation.

- The ESC . T instruction first clears the buffers, then allocates memory to all buffers at once.
- The ESC . L instruction helps ensure that memory has been allocated before any HP-GL instructions are sent to the plotter.
- The ESC . @ instruction specifies the size of the logical I/O buffer.

Only the ESC . T instruction is used every time you allocate memory. This instruction is discussed in detail later in this section. The use of the ESC . @ and ESC . L instructions depends on your application and on your computer system. The following discussion will help you decide if and when to include these instructions in your program.

Using ESC . @ and ESC . L

Use ESC . @ whenever you increase the physical I/O buffer

Use the ESC . @ instruction to increase the logical I/O buffer *after* you have increased the physical I/O buffer. The logical I/O buffer is part of the physical I/O buffer (it is not a separate buffer). Because the number of instructions stored in the physical I/O buffer depends on the size of the logical I/O buffer, set the logical and physical I/O buffers to the same size.

NOTE: You never need to reduce the logical I/O buffer's size. When you decrease the physical I/O buffer, the logical I/O buffer automatically assumes this new size. ■

Use ESC . L to prevent the loss of plot information

When writing a program for either plotter type, send ESC . L *before* ESC . T as a safeguard to prevent ESC . T from clearing the buffers until the previous plot is complete. When writing a program for the graphics plotter, always send ESC . L *after* ESC . T. In this case, the ESC . L instruction "buys time" for the memory allocation process to finish and will prevent you from losing plot data. Instead of ESC . L, drafting plotters require a software delay to ensure memory allocation.

*Your plotter's Interfacing and Programming Manual describes device-control instructions in more detail. Remember, all device-control instructions use the single ASCII escape character, **ESC** (decimal code 27). On many computers, you can access this character by striking a single key, or, as in our BASIC program examples, use a function such as CHR\$ to send its decimal code.

The ESC . L instruction does two things; it postpones data transmission until the physical I/O buffer is empty, then it outputs the size of the logical I/O buffer. In this case, the buffer's size is irrelevant. However, to avoid a plotter I/O error, your computer must read this output.

To read the output statement, follow each ESC . L instruction with a statement such as READ or ENTER. This statement is computer-dependent. The program listings in Section 4 use the statement ENTER 705. The 705 specifies the HP-IB interface select code and plotter address so the computer knows which device is sending information. Consult your Interfacing and Programming Manual and your computer documentation if you need to modify this statement.

Using ESC . T

The ESC . T instruction allocates memory for either plotter type. Each parameter in ESC . T reflects the number of bytes assigned to a single buffer. A parameter (buffer size) must stay within a specified range. You can send ESC . T with or without its parameters. An ESC . T instruction without parameters sets all buffers to their default values.

Because the number of buffers is different for drafting and graphics plotters, read the following section that describes how your plotter uses ESC . T.

How Drafting Plotters Use ESC . T

For drafting plotters, the ESC . T instruction allocates memory to the physical I/O, polygon, and downloadable character buffers. The following table lists the buffers (parameters) in the order they must appear in the ESC . T instruction. The table gives each buffer's range and default value in bytes. Note how the sample ESC . T instruction following the table uses both stated and default values to allocate memory.

*Range and default values
of drafting plotter buffers*

Buffer	Range	Default
Physical I/O buffer	2-18 432	1024
Polygon buffer	0-18 430	3072
Downloadable character buffer	0-18 430	0

The following ESC . T instruction sets the physical I/O buffer to its default value (1024 bytes); allocates 8000 bytes to the polygon buffer and 3000 bytes to the downloadable character buffer.

ESC . T 1024 ; 8000 ; 3000 :

How the Graphics Plotter Uses ESC . T

When using the ESC . T instruction, refer to the following table and send the buffer parameters in the order they are listed. The table lists the buffers' range and default values in bytes. Note how the sample ESC . T instruction following the table allocates memory with both stated and default buffer values.

**Range and default values
of graphics plotter buffers**

Buffer	Range	Default
Physical I/O buffer	2-12 752	1024
Polygon buffer	4-12 754	1778
Downloadable character buffer	0-12 750	0
Replot buffer	0-12 750	9954
Vector buffer	44-12 794	44

The following ESC . T instruction sets the physical I/O buffer to its default value (1024 bytes); allocates 3000 bytes to the polygon buffer, 2000 bytes to the downloadable character buffer, 6000 bytes to the replot buffer; sets the vector buffer at its default value (44 bytes).

ESC . T ; 3000 ; 2000 ; 6000 :

Using the GM Instruction (Graphics Plotter Only)

**Another way for the
graphics plotter to allo-
cate memory**

With the graphics plotter, you can also allocate memory with the HP-GL instruction, GM. The GM instruction allocates memory in the same manner as the ESC . T instruction, except that GM *does not* change the size of the physical I/O buffer. The physical I/O buffer remains at whatever it is currently set to. In the GM instruction, the sequence of the four parameters is the same as in ESC . T.

GM polygon buffer , downloadable character buffer ,
replot buffer , vector buffer ;

The GM instruction uses the same syntax conventions as other HP-GL instructions. (See your Interfacing and Programming Guide if you need help remembering HP-GL syntax.) For example, GM 3000 , 2000 , 6000 ; performs the same memory allocations as the previous sample ESC . T instruction (provided the physical I/O buffer is currently set at its default value).

NOTE: When you use the GM instruction, you do not need to send the ESC . L instruction. Also, be sure to allocate memory with GM *before* you send the HP-GL instructions that draw your plot. ■

Step 2 — Send HP-GL Instructions

After allocating memory, send the plotter the HP-GL instructions that use each buffer. For example, let's assume your program uses the PM instruction to define a large polygon and store it in the polygon buffer. You must first calculate the number of bytes the buffer needs to store your polygon. Next, increase the polygon buffer's size by sending memory allocation instructions. Now, send the PM instruction and the points defining your polygon. Once stored, this defined polygon will remain in the polygon buffer, ready to be accessed by the FP or EP instructions.

Section 3 provides a more complete view of which HP-GL instructions each buffer uses. Check the sample plots and program listings in Section 4 to see how memory allocation and HP-GL instructions work together.

Questions and Answers about Allocating Memory

Drafting plotters set default values for all buffers

The graphics plotter reduces the largest buffer

What If I Exceed the Maximum Storage Capacity?

When allocating memory, remember that the total memory capacity is divided between all buffers. In drafting plotters, if the sum of the three allocations exceeds 18432 bytes, the plotter overrides the ESC.T instruction. When this happens, the buffers automatically assume their default values and an error, "parameter out of range," occurs. A front-panel LED signals the error.

For graphics plotters, if you execute an ESC.T instruction and the parameters' sum is greater than 12800 bytes, the plotter uses an algorithm to reduce this sum to the maximum. For example, suppose you wish to plot a complex polygon such as the Olympic symbol in Section 4. You decide to allocate 10000 bytes to the polygon buffer, but forget to "borrow" storage space from another buffer. You have exceeded the maximum storage capacity. The plotter will perform the algorithm, and in this case, automatically reduce the polygon buffer to its default value. The buffer can store only part of your polygon.

What If I Don't Allocate Enough Storage Area?

When a buffer doesn't have enough storage area to accommodate incoming data, the plotter will not produce the results you expect and may indicate an error, "buffer overflow." The following table shows how the plotter responds when a buffer is too small and/or overflows.

How the plotter responds when a buffer is too small

Error Condition	Plotter Response
Physical I/O buffer too small	No error; data transmission is slowed down
Polygon buffer overflows	Error 7; plotter will not fill the polygon and may only partially edge it
Downloadable character buffer overflows	Error 7; buffer will store characters up to the allocation limit, then ignore subsequent character definitions
Replot buffer overflows	Error 7; plotter clears the replot buffer and exits buffer plot mode, but completes the first plot
Vector buffer too small	No error; effects of CV instruction are negligible

Use ESC.S to check buffer sizes

Can I Check the Capacity of Each Buffer?

You can verify the number of bytes currently allocated to each buffer by using the output configurable memory size instruction, ESC.S. Depending on which parameter you use, this instruction outputs either the total number of bytes currently allocated to the configurable graphics memory or the number of bytes in any of its buffers.

How Are Memory Allocations Reset?

The memory allocations you set remain in effect only until one of the following conditions occurs.

- You turn the plotter off; when the plotter is turned on again, the buffers are automatically cleared and set to default values.
- You send the reset instruction, ESC . R, to the plotter in a program. A reset by this method clears all buffers and defaults the buffer allocations.
- You send another ESC . T (or GM) instruction with different allocations; each new instruction clears the buffers before memory allocations are reset.

Section 3: Make the Best Use of Each Buffer

This section will help you decide when to change a buffer's size and how much memory to allocate to each buffer. You will also learn how to use HP-GL instructions with the buffers. Refer to your plotter's Interfacing and Programming Manual for additional HP-GL programming details and for the formulas to calculate the number of bytes needed to store polygons, characters, vectors, and replot data.

Physical I/O Buffer

A large physical I/O buffer uses computer time efficiently

The physical I/O buffer is used in all plotting operations. Generally, the default size is convenient for most plots. However, if you are using a time-sharing computer system and/or a serial interface (e.g., RS-232-C/CCITT V.24), consider allocating as much storage space as possible to this buffer. The larger the physical I/O buffer, the more data the computer can send to the plotter at a given time. Conversely, allocating too few bytes to this buffer slows down plot throughput.

NOTE: After increasing the size of the physical I/O buffer, you also need to increase the size of the logical I/O buffer using the ESC . @ instruction. Refer to Section 2 for details. ■

Polygon Buffer

A complex polygon needs more storage area than a simple polygon

The polygon buffer is used whenever your graphics program sends certain HP-GL polygon instructions to the plotter. If your program does not use these instructions, and you need extra bytes for the other buffers, decrease the polygon buffer to its minimum.

At the default setting, the polygon buffer can store any polygon containing up to 127 points for the graphics plotter and up to 219 points for drafting plotters. To determine how many bytes the buffer needs for your plot, convert the number of points in the largest polygon into bytes. (Use the conversion formula in your Interfacing and Programming Manual.) Be sure to allocate memory only for those HP-GL instructions that use this buffer.

The PM (polygon mode) instruction is the one HP-GL instruction that may require you to increase the polygon buffer above its default value. The PM instruction uses other HP-GL instructions to define and store a complex polygon. This polygon remains in the buffer and is not drawn until you send an EP or FP instruction to edge or fill it.

Several other HP-GL polygon instructions also use the polygon buffer. For drafting plotters, these instructions include WG, RA, and RR. For the graphics plotter, the instructions are: WG, EW, RA, RR, EA, and ER. None of these polygon instructions require significant storage area. Each instruction first clears the buffer, then uses the buffer to define a simple polygon before filling or edging it.

Downloadable Character Buffer

Enlarge this buffer before you define characters

The downloadable character buffer is only used when a program creates downloadable characters. The buffer's default size for all plotter models is zero. When you want to create characters, you must first allocate enough bytes to store them. This buffer requires a minimum of 444 bytes for overhead, plus enough bytes to store each character according to the formula in your Interfacing and Programming Manual.

After allocating storage, use the DL instruction to create individual characters. Once defined, these characters reside in the buffer as character set - 1. To use these characters in a plot, simply designate and select character - 1. Then send a labeling instruction (LB, BL, PB, or SM) to retrieve your characters in the same manner as any other character set.

Replot Buffer (Graphics Plotter Only)

To replot successfully, increase this buffer to the maximum size possible

The replot buffer is activated when the plotter receives HP-GL replot instructions. The buffer's default size is 9954 bytes. If you do not intend to replot, and you want to allocate these bytes to another buffer, reduce the replot buffer to 0 bytes. In many cases, the buffer's default size is large enough to store your entire graphics program. However, to avoid buffer overflow, allocate the maximum possible memory to the replot buffer.

The replot buffer remains empty until you issue the buffer plot instruction, BF. This instruction stores subsequent plot information in the buffer. The plot is drawn one time while it is being stored. BF continues to store information until the replot instruction, RP is executed (or until the replot buffer overflows). The RP instruction signals the plotter that the plot has been stored and to begin replotting the number of copies specified. As soon as replot begins, your computer is free to perform other functions.

NOTE: The front-panel **RELOT** function key emulates the BF and RP instructions; the **LOAD** function key is equivalent to the BF instruction and the **START** function key is equivalent to the RP instruction. ■

Vector Buffer (Graphics Plotter Only)

The vector buffer processes all vectors in every plot. These vectors are drawn one at a time. You can not reduce this buffer below the default value of 44 bytes.

Enlarge the vector buffer before sending the CV instruction

Increase the vector buffer when you intend to use the curved line generator instruction, CV. Using the formula in your Interfacing and Programming Manual, allocate enough bytes to accommodate the largest number of vectors that a single curve will have. For example, when plotting a circle, increase the vector buffer to

hold all vectors that define the circle. If the buffer is not large enough, the effects of the CV instruction are negligible.

Once you have allocated memory, send the CV instruction before you define the vectors. The CV instruction collects vectors in the buffer to be plotted as a group. The pen does not stop at each vector endpoint, so curves appear smoother. The effects are especially noticeable in circles or characters drawn with short vectors.

NOTE: If plot throughput is important, do not increase the vector buffer size too much. This could cause the plotter to spend unnecessary time waiting for vectors before plotting them as a group. ■

Section 4: How Buffers Work Together

This section uses examples to illustrate memory allocations for several different buffer combinations. Each example lists the device-control instructions that graphics and drafting plotters require to allocate memory. In Examples 2 through 4, HP-GL instructions produce increasingly complex programs and plots. The program in Example 4 uses the replot and vector buffer, and will run on the graphics plotter only.

How to use these examples

All sample programs were written on an HP 9000 Series 200 computer. You might need to change some of the BASIC statements if you are programming in another language, or if the BASIC on your computer is different. Before you run a graphics program, your computer and plotter must be properly connected. Also, if you are using the RS-232-C/CCITT V.24 interface, make sure your handshaking protocol is established.

Example 1 — One Buffer Uses the Maximum Storage

This example increases the physical I/O buffer's size

Any individual buffer may use almost the entire graphics memory, but you must reduce the other buffers' sizes to minimum or default values. For this example, let's assume you want to decrease the amount of time that your computer is involved in graphics data transmission and you don't need any of the other buffers. Use the following ESC.T instruction to maximize the size of your physical I/O buffer. Remember, when you increase the physical I/O buffer, you must also increase the size of logical I/O buffer with the ESC.@ command.

Drafting Plotter Instructions

In the following ESC.T instruction, the physical I/O buffer receives the largest storage area (18430 bytes), the polygon buffer is reduced to the minimum (0 bytes), and the downloadable character buffer is set at the default value (0 bytes). The ESC.@ instruction increases the logical I/O buffer to equal the physical I/O buffer's size.

```
ESC.T 18430;0;0:  
ESC.@ 18430:
```

Graphics Plotter Instructions

In the following ESC . T instruction, the physical I/O buffer receives the largest storage area (12752 bytes). All other buffers are reduced to minimum or default values. The ESC . L instruction outputs the size of the logical I/O buffer and causes your program to wait until all memory allocations are complete. The ESC . @ instruction increases the logical I/O buffer to equal the physical I/O buffer's size. (Remember to include a statement that tells the computer to read the ESC . L output.)

```
ESC . T 12752;4;0;0;44:
```

```
ESC . L
```

```
ESC . @ 12752:
```

Example 2 — Increase the Size of Two Buffers

This example increases the physical I/O and polygon buffers

Let's say you wanted to decrease the amount of time your computer is involved in plotting but also draw a complex polygon (in this case, the Olympic symbol). The ESC . T instruction should increase the size of the physical I/O buffer and the polygon buffer and set all other buffers at their minimum values.

The memory allocation instructions for each plotter model are given first, followed by a sample program that can be run on either type of plotter.

Drafting Plotter Instructions

The following ESC . T instruction allocates 8962 bytes to the physical I/O buffer, 9470 bytes to the polygon buffer and 0 bytes to the downloadable character buffer. The ESC . @ instruction increases the logical I/O buffer to equal the size of the physical I/O buffer.

```
ESC . T 8962;9470;0:
```

```
ESC . @ 8962:
```

Graphics Plotter Instructions

The following ESC . T instruction allocates 3286 bytes to the physical I/O buffer, 9470 bytes to the polygon buffer, and assigns minimum values to the remaining buffers. The ESC . L instruction confirms memory allocation. The ESC . @ instruction increases the logical I/O buffer to equal the size of the physical I/O buffer.

```
ESC . T 3286;9470;0;0;44:
```

```
ESC . L
```

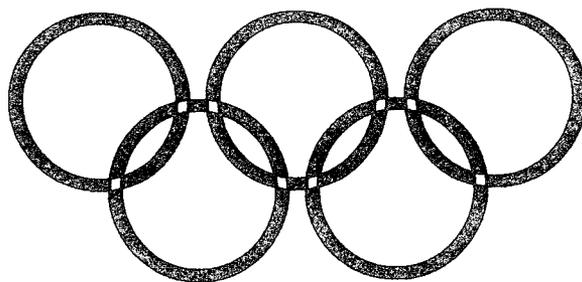
```
ESC . @ 3286:
```

The following BASIC program uses the PM, FP, and EP instructions to draw and fill the Olympic symbol. Note that lines 40 through 70 allocate memory for graphics plotters.

Modify this program for drafting plotters

If are using a drafting plotter, to account for the differences in memory allocation and scaling, modify the program as follows:

1. Replace the parameters in lines 40 and 70 with those given previously for drafting plotters.
2. Replaces lines 50 and 60 with a software delay.
3. Replace line 80 with `PRINT "IN;SP1;PA0,0;"`



```
10 PRINTER IS 705
20 PRINT CHR$(27)&".L"
30 ENTER 705;L
40 PRINT CHR$(27)&".T3286;9470;0;0;44:"
50 PRINT CHR$(27)&".L"
60 ENTER 705;L
70 PRINT CHR$(27)&".@3286:"
80 PRINT "IN;SP1;PA1500,2500;"
90 PRINT "PM0;CI500;CI435;"
100 PRINT "PR550,-500;CI500;CI435;"
110 PRINT "PR550,500;CI500;CI435;"
120 PRINT "PR550,-500;CI500;CI435;"
130 PRINT "PR550,500;CI500;CI435;PM2;"
140 PRINT "FP;EP;SPO;"
150 END
```

Example 3 — Increase the Size of Three Buffers

This example increases the physical I/O, polygon and downloadable character buffers

Let's optimize your computer time, draw the Olympic symbol, and label the symbol with Greek letters that spell the word "Olympiad". In this case, the ESC.T instruction should increase the storage area of the physical I/O, polygon, and downloadable character buffers. (The downloadable character buffer will store the Greek characters you create with the DL instruction.)

The memory allocation instructions for each plotter model are given first, followed by a sample program that can be run on either type of plotter.

Drafting Plotter Instructions

The following ESC.T instruction increases the physical I/O buffer (8312 bytes), polygon buffer (9470 bytes), and downloadable character buffer (650 bytes). The ESC.@ instruction increases the logical I/O buffer to equal the size of the physical I/O buffer.

ESC.T 8312;9470;650:
ESC.@ 8312:

Graphics Plotter Instructions

The following ESC.T instruction increases the physical I/O buffer (2636 bytes), polygon buffer (9470 bytes), and downloadable character buffer (650 bytes). It also allocates the minimum value to the replot and vector buffers. The ESC.L instruction confirms memory allocation. The ESC.@ instruction increases the logical I/O buffer to equal the size of the physical I/O buffer.

ESC . T 2636;9470;650;0;44:

ESC . L

ESC . @ 2636:

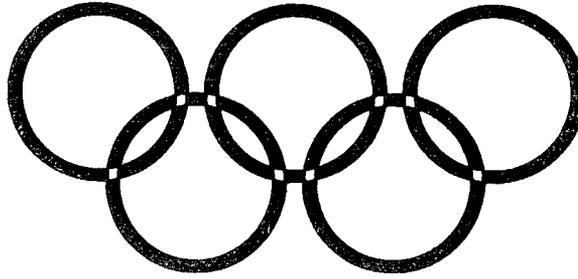
The following BASIC program uses the PM, FP, and EP instructions to draw and fill the Olympic symbol and the DL, CA, and LB instructions to define and draw Greek characters. Note that lines 40 through 70 allocate memory for graphics plotters.

Modify the program for drafting plotters

If you are using a drafting plotter, you need to modify the program to account for the differences in memory allocation and scaling instructions. Change the program as follows:

1. Replace the parameters in lines 40 and 70 with those given previously for drafting plotters.
2. Replace lines 50 and 60 with a software delay.
3. Replace line 80 with `PRINT "IN;SP1;PA0,0;"`
4. Replace line 220 with `PRINT "CA-1;SI.6,.8;PA-475,-1500;"`

NOTE: Although lines 150 through 200 are printed on two lines to fit on this page, you should send each line to the plotter as a continuous string. In line 230, the `␣` symbol terminates the label instruction, and the `␣` symbol invokes the downloadable character set, -1. In this program, `␣` represents the ASCII character **ETX** (decimal code 3), and `␣` represents ASCII character **SO** (decimal code 14). In a BASIC program, you can use a function such as `CHR$` to send these decimal codes. (Refer to your Interfacing and Programming Manual and your computer's documentation if you need more information.) ■



9 ΟΛΥΜΠΙΑΣ

```
10 PRINTER IS 705
20 PRINT CHR$(27)&" .L"
30 ENTER 705;L
40 PRINT CHR$(27)&" .T2636;9470;650;0;44:"
50 PRINT CHR$(27)&" .L"
60 ENTER 705;L
70 PRINT CHR$(27)&" .@2636:"
80 PRINT "IN;SP1;PA2000,3000;"
90 PRINT "PM0;CI500;CI435;"
100 PRINT "PR550,-500;CI500;CI435;"
110 PRINT "PR550,500;CI500;CI435;"
120 PRINT "PR550,-500;CI500;CI435;"
130 PRINT "PR550,500;CI500;CI435;PM2;"
140 PRINT "FP;EP;"
150 PRINT "DL76,0,0,4,0,16,17,-128,4,0,16,17,-128,4,32,
      8,32,28,0,32,0;"
160 PRINT "DL89,0,20,4,20,10,2,11,1,12,0,18,0,19,1,20,
      2,26,20,22,20;"
170 PRINT "DL77,-4,-4,-2,-2,8,20,10,6,11,4,12,3,13,1,
      19,1,20,3,21,4,22,6,27,20,32,0;"
180 PRINT "DL80,0,0,2,3,6,18,-128,2,20,6,18,16,18,18,
      19,21,22,-128,16,18,20,3,22,0,24,2;"
190 PRINT "DL73,17,0,15,2,15,20,14,19;"
200 PRINT "DL65,22,20,22,0,24,0,28,6,-128,22,16,18,20,
      6,20,2,16,2,4,6,0,18,0,22,4;"
210 PRINT "DL67,22,16,18,20,6,20,2,16,2,4,6,0,18,0,22,
      4,24,4,24,0,19,-4;"
220 PRINT "CA-1;SI.6,.8;PA1525,1500;"
230 PRINT "LB'0%LYM%;ES-.3,0;LB;PI%;ES;LB;AC%"
240 PRINT "SPO;"
250 END
```

Example 4 — Allocate Memory to Four Buffers (Graphics Plotter Only)

A complex plot requires forethought

In this example, let's assume you need a surface line chart labeled with variable-spaced, ANSI ASCII characters. You also want to replot two originals and increase the line quality of the characters. The ESC . T instruction should allocate memory to four buffers: the physical I/O buffer, polygon buffer, replot buffer, and vector buffer. For this complex plot, consider the following:

- The physical I/O buffer can be reduced and the extra bytes allocated to the replot buffer.

- The polygons in this plot are relatively simple; reduce the polygon buffer to accommodate the largest of the two polygons.
- The replot buffer must have enough bytes to store the entire plot.
- The vector buffer needs enough space to store the largest number of vectors a character is likely to have.

The following ESC . T instruction reduces the physical I/O buffer to 750 bytes, reduces the polygon buffer to 300 bytes, defaults the downloadable character buffer to 0 bytes, increases the replot buffer to 11 000 bytes, and increases the vector buffer to 720 bytes. The ESC . L instruction confirms the memory allocations.

ESC . T 750 ; 300 ; 0 ; 11 000 ; 720 :

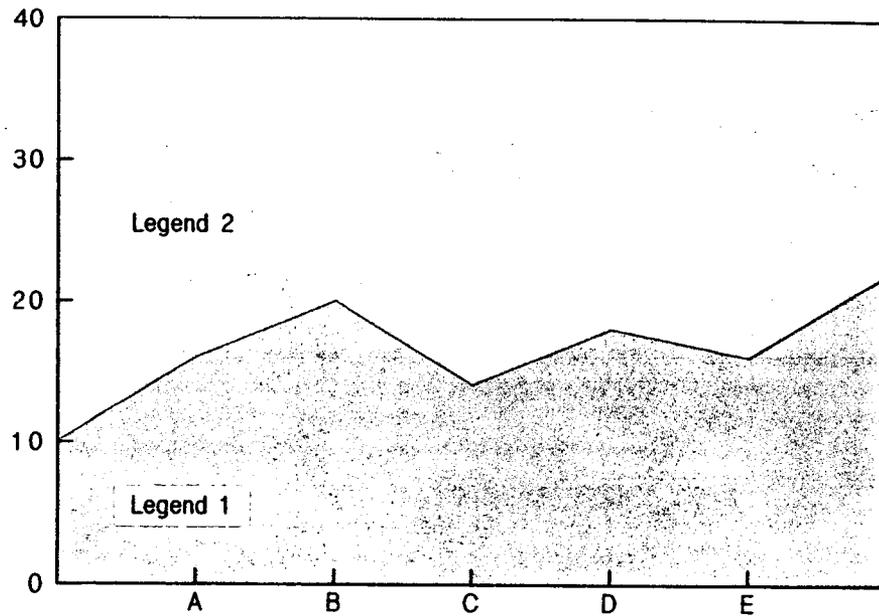
ESC . L

This program uses the physical I/O, polygon, replot, and vector buffers

The following BASIC program uses the PM, FP, and EP instructions to draw and fill two polygons. The TL, LO, and LB instructions mark plot axes and draw labels. The CV instruction increases the line quality of all labels. The BF instruction stores the plot in the replot buffer, and RP causes two replots to be drawn.

NOTE: Whenever you replot, you must first turn on the AUTO FEED mode and be sure that paper is loaded in the tray. Throughout this program, the ξ symbol terminates labeling instructions. This symbol represents the ASCII character **ETX** (decimal code 3). In a BASIC program, you can use a function such as CHR\$ to send the decimal code. (Refer to your Interfacing and Programming Manual and your computer's documentation if you need more information.) ■

Surface Line Chart



```

10  PRINTER IS 705
20  PRINT CHR$(27)&" .L"
30  ENTER 705;L
40  PRINT CHR$(27)&" .T750;300;0;11000;720:"
50  PRINT CHR$(27)&" .L"
60  ENTER 705;L
70  PRINT "BF;IN;SC-20,80,-10,60;"
80  PRINT "SP2;PA0,0;PM0;PDO,10,10,16;"
90  PRINT "PD20,20,30,14,40,18,50,16;"
100 PRINT "PD60,22,60,0,0,0,PM1;"
110 PRINT "PU4,4;PD4,7,14,7,14,4,4,4;PU;PM2;FP;EP;"
120 PRINT "SP3;PA0,10;PM0;PDO,30,10,36;"
130 PRINT "PD20,32,30,30,40,34,50,32,60,34,60,22;"
140 PRINT "PD50,16,40,18,30,14,20,20,10,16,0,10,PM1;"
150 PRINT "PU4,24;PD4,27,14,27,14,24,4,24;PU;PM2;FP;EP;"
160 PRINT "SP1;CS10;PA0,0;ER60,40;"
170 PRINT "CV1;SI.3,.6;PU16,42;LBSurface Line Chart%"
180 PRINT "SI.2,.3;PU5.5,5;LBLegend 1%"
190 PRINT "PU5.5,25;LBLegend 2%"
200 PRINT "TL1.5,0;"
210 FOR X=10 TO 50 STEP 10
220   PRINT "PA";X;"0;XT;"
230   READ A$
240   PRINT "L016;LB";A$;"%"
250 NEXT X
260 PRINT "TL1,0;"
270 FOR Y=0 TO 40 STEP 10
280   PRINT "PA0,";Y;"YT;"
290   PRINT "L08;LB";Y;"%"
300 NEXT Y
310 DATA "A","B","C","D","E"
320 PRINT "SPO;RP2;"
330 END

```