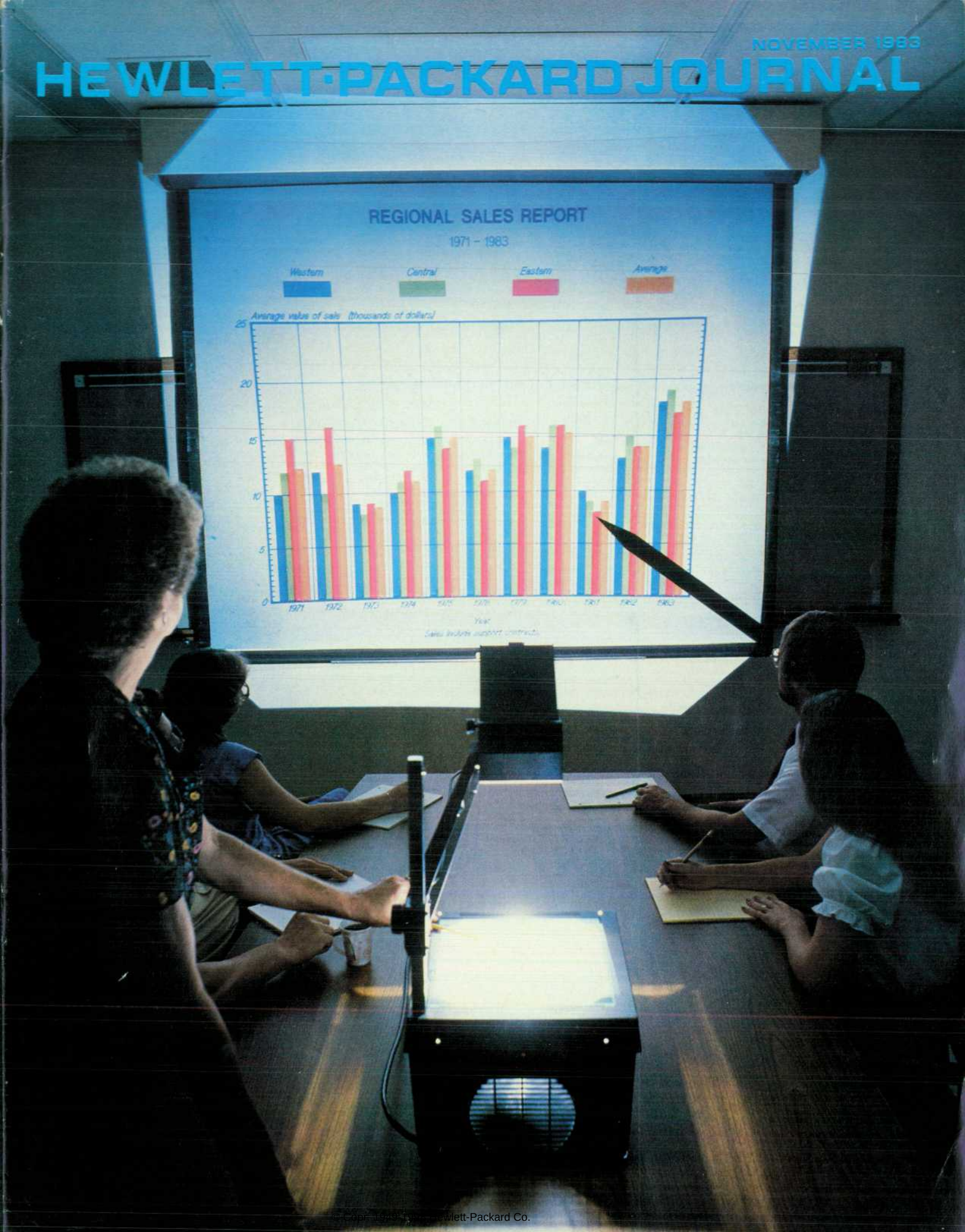
NOVEMBER 1983

HEWLETT·PACKARD JOURNAL



© Copyright 1983 Hewlett-Packard Co.

HEWLETT·PACKARD JOURNAL
Technical Information from the Laboratories of Hewlett-Packard Company

NOVEMBER 1983 Volume 34 • Number 11

Contents:

3 **Device-Independent Software for Business Graphics,** *by Yvonne Temple* *HP 3000 Computer users can create business graphics without regard to the type of output device they'll eventually use.*

5 **A Decision Support Chartmaker,** *by Janet Elich Morris and Richard J. Simms, Jr.* *Two user interfaces satisfy the needs of both the nonprogrammer and the sophisticated user.*

10 **An Easy-to-Use Chartmaker,** *by Martha Seaver, Robert W. Dea, and Richard J. Simms, Jr.* *It's the simplest way for a nonprogrammer to get a professional-looking chart.*

13 **Convenient Creation and Manipulation of Presentation Aids,** *by Chayaboon Purnaveja and Janet Swift* *Draw figures freehand or choose them from a library, then edit them by point-and-push methods.*

17 **Graphics Capabilities on a Laser Printer,** *by Tamara C. Baker, William J. Toms, James C. Bratnober, and Gerald T. Wade* *Printers do text documents and plotters produce graphics? This printer can do the whole job.*

23 **Special Report: The Center for Integrated Systems,** *by Frederick H. Gardner* *It's a new approach to industry support of university research.*

31 **Authors**

In this Issue:

Graphs, charts, and pictures are wonderful things to have when you're required to give a business presentation. They prompt the speaker, they help the audience understand (if they're well made), and they make it slightly less likely that the audience will fall asleep. In a report, a single graph can summarize pages of data. A picture can be worth a lot of words.

One of the major benefits of computers for business is that they've made it ridiculously easy to get transparencies and hard copy of charts, graphs, even relatively complicated pictures. You, the ultimate user of a chart, can produce it yourself. No longer do you have to take your idea to a graphic artist or illustrator, who applies special skills to generate artwork, which must then go to a printer or photo lab to be turned into the thing you want. Now you can sit down at a computer terminal, and if your needs are simple, like a pie, bar, or line chart, you can simply select items from a menu and type in some numbers. If you have a little more skill, you can get remarkably professional custom effects remarkably easily. With a graphics tablet, you can put freehand sketches into your computer-generated output. Color is no problem, either. All it takes is the right kind of plotter or camera hooked up to the computer—and, of course, the right kind of software.

It's the software that makes the machinery user-friendly, so that an ordinary person without programming skills can, in a relatively short time, learn to produce high-quality graphics. And that's what most of this issue is about—business graphics software for the HP 3000 Business Computer. In four of the articles, HP software designers describe a pair of chartmaker software packages called DSG/3000 and HP EASYCHART, a general picture creator called HPDRAW, and a software package that makes it possible to merge text and graphics on an HP 2680A Laser Printer. The article on page 3 starts off the issue by discussing some common features of these four packages.

The special report on page 23 is a first for the HP Journal—we commissioned it. Unlike most of our articles, it's not about the results of HP R&D efforts. HP is one of several companies supporting the Center for Integrated Systems at Stanford, an innovative approach to industry support of university research. The subject is taken seriously by HP management, and HP President John Young and Vice President for Research John Doyle were instrumental in getting CIS started. The article tells how questions of patent rights, academic freedom, and other tough issues were resolved in this pioneering effort.

-R. P. Dolan

Editor, Richard P. Dolan ■ Associate Editor, Kenneth A. Shaw ■ Assistant Editor, Charles J. Lisherness ■ Art Director, Photographer, Arvid A. Danielson ■ Illustrators, Nancy S. Vanderbloom, Susan E. Wright ■ Administrative Services, Typography, Anne S. LoPresti, Susan E. Wright ■ European Production Supervisor, Henk Van Lammeren

© Hewlett-Packard Company 1983 Printed in U.S.A.

© Copr. 1949-1998 Hewlett-Packard Co.

Device-Independent Software for Business Graphics

New programs fortify the electronic office with a choice of graphics interfaces.

by Yvonne Temple

TO BRING THE POWER of computer graphics to a broad range of office workers—from the novice programmer to the management information systems (MIS) specialist—Hewlett-Packard has developed a set of business graphics software packages for the HP 3000 Computer System. The new software products include DSG/3000, a decision support chartmaker with data file handling, HPEASYCHART, an interactive chartmaker designed for the nonprogrammer, HPDRAW, a figure design system that can be used with the HP 17623A Graphics Tablet for sophisticated data entry and digitizing, and the HP 2680A Graphics Package, which allows graphics to be printed on the HP 2680A Laser Printer at 45 pages per minute.

DSG/3000 (see article, page 5) is a sophisticated chartmaking package that can be used with either computer-stored or manually entered data to manipulate and visually convey information.

HPEASYCHART (see article, page 10) uses a follow-the-example data entry method to produce pie charts, bar charts, line charts, and scattergrams.

HPDRAW (see article, page 13) contains a library of frequently used graphic images which can be combined with freehand drawings to create customized visual aids.

The HP 2680A Graphics Package (see article, page 17) contains a set of programmatic intrinsics and an interactive interpreter so that graphics created with DSG/3000, HPEASYCHART, and HPDRAW can be converted to raster images and printed on the HP 2680A Laser Printer.

Device Independence

Not all graphics devices have similar operating capabilities or constraints. Information about physical device limits, device type (e.g. terminal, printer, or plotter), or color capabilities can vary greatly. Since the new HP business graphics software supports so many devices (seven different terminals, twenty plotters, and several printers), maximum operating efficiency is achieved by relegating all device-dependent activity to a low level in the program structure and by restricting device-dependent information in the code to as few locations as possible. A high degree of device independence is further realized by having all supported graphics devices identify themselves and respond to status requests for graphics input and/or output. The software uses a powerful set of query routines to determine the operating characteristics of an identified device. This variable information is then used to fill capability parameters in the business graphics software interface. The upshot of this device independence for users is that it allows them to access many different graphics devices without knowing the specifics of the devices' operating characteristics.

Device Support

Device-independent graphics relies on a set of utility routines that perform all graphical tasks and make all graphics devices appear the same to higher software. These routines are divided into three logical levels (Fig. 1):

- Utility level, a set of device-independent routines
- Universal level, an intermediate area where device-independent actions are performed and where the software text generator resides
- Device driver level, a set of device-dependent routines.

The utility level assumes all devices behave identically. Parameters are checked for errors at this level, and the universal level is called to perform requested actions.

The universal level consists of a main procedure called the universal driver and several action procedures. The utility level passes an action code and parameters to the universal driver, and the driver selects the appropriate routine to perform the action. As an example of activity at this level, all coordinates are converted from world coordinates to normalized device coordinates, and all global variables are updated to reflect the action performed.

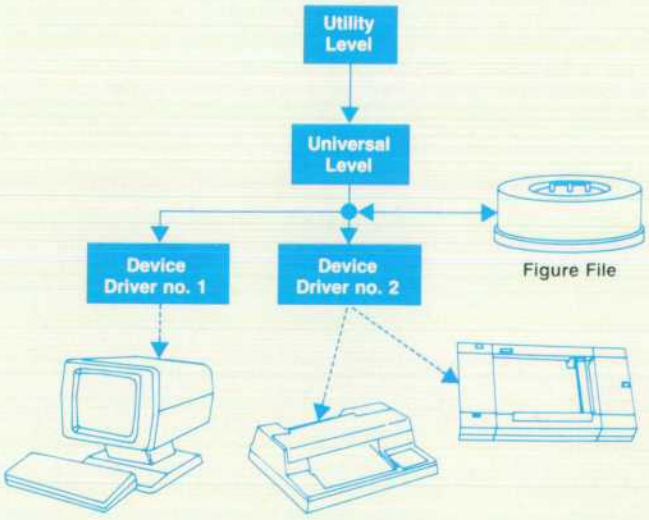


Fig. 1. *HP device-independent business graphics software uses a set of utility routines that is divided into three levels of activity.*

© Copr. 1949-1998 Hewlett-Packard Co.

Device Driver Feature Set

Move
Draw (8 line styles)
Digitize
Text (3 levels of quality)
Set pen speed
Select pen (0-64)
Marker (8 types)
Polygon fill (8 fill styles)

Fig. 2. *All graphics device driver routines have similar features.*

On the device driver level, there is a different driver for each class of devices. For example, one device driver generates HP-GL* commands for RS-232-C/V.24 plotters (HP 7220, HP 7580, etc.), while another generates escape sequences for a graphics terminal. All device drivers support a similar set of features (Fig. 2). The utility routines take advantage of device intelligence whenever possible, but if a device class does not support a necessary feature in resident firmware (e.g., polygon fill), then the task is performed with the aid of software helper routines (e.g., a polygon fill generator).

Figure Files

Device-independent graphics software packages also rely on three data structures—figure files, font files, and color palettes.

Initially developed for HPDRAW as a means of storing frequently needed graphic design images, figure files are keyed access files used by all HP business graphics software to store and transfer graphical data. They can hold several individual figures which can be drawn separately or included in other drawings.

A figure file consists of a figure file header and one or more figures. The figure file header contains the file's creation date and the version of graphic utility routines used. Each figure is composed of a figure header and figure data. The figure header holds a comment field, information on the size of the figure, and the date the figure was created. The figure data consists of action codes and the data associated with each action code. For example, a draw command consists of the draw action code and an (x,y) coordinate pair designating where the action will stop. Figure file action codes are read by a figure file translator and passed directly to the universal driver. The rotation or scaling of a figure file occurs in the universal level.

Drawing to a figure file is almost the same as drawing to any other graphics device. In fact, the figure file generation routines are implemented as a device driver since they logically fit the functional structure of other device drivers.

Text

Two software text generators provide HP business graphics products with a choice of several different fonts for high-quality text. The data used to produce the fonts is stored in a font file. Four software fonts are active at a given time, and any of the four can be redefined at any time during the graphics session. A new font is selected by specifying the font file name and the character set desired. To optimize performance, the data for the four active fonts is stored in separate extra data segments. When a new font is specified, the font information is read from the font file and stored in an extra data segment. When the text utility is called, the data for the current font is moved onto the user's stack and the appropriate text generator draws the text. New fonts are easy to support by creating a new font file.

*Hewlett-Packard Graphics Language

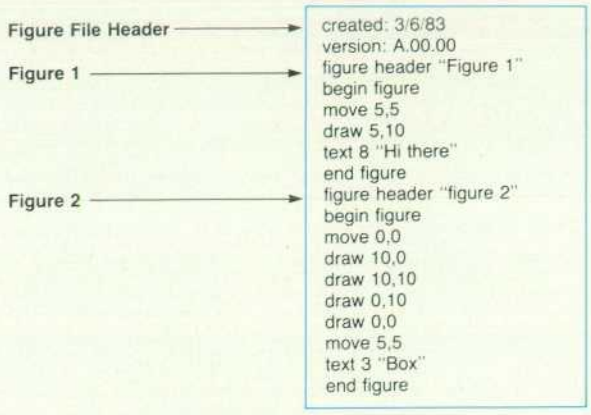


Fig. 3. *Example of a figure file.*

Color Palette

With the introduction of the HP color terminals (HP 2700 and HP 2627A) and the advent of programmatic color selection, it became evident that a more sophisticated method of color support was needed. Selecting a pen number just isn't sufficient when a device has the capability of defining a pen color.

HP device-independent graphics software uses color palettes that recognize a device's color specification capability and direct a user toward attractive color mixtures on plotters and terminals. The various palettes available are located in a palette file containing the definition of each palette pen in terms of its red, green, and blue content. When a color terminal is used, the palette is referenced and the pen colors are set accordingly. Since the 2627A does not support the definition of pen colors, the full range of palette values is supported only during solid fill by using the 2627A's dither patterns.

Each palette was designed with colors that appear aesthetically pleasing together. The default palette colors match HP plotter pen colors, so a drawing produced on a color terminal will look similar to one produced on a plotter.

© Copr. 1949-1998 Hewlett-Packard Co.

A Decision Support Chartmaker

by Janet Elich Morris and Richard J. Simms, Jr.

TRENDS ARE SPOTTED more easily on a chart than in a table of values. A large table of sales data for the past three years would be tedious to read, but a chart of the data might quickly show that sales at the beginning of each quarter are higher than at other times. With Decision Support Graphics/3000 (DSG/3000), data stored in any data file, including a file taken from a data base, can be charted to convey the stored information visually.

DSG/3000 is a comprehensive business graphics software package that produces bar, line, and pie charts, and slides. It is one of a family of graphics packages designed for the HP 3000 Computer System.

DSG/3000 allows the user to design, produce, and save charts generated from information kept in a data file. It is a versatile software tool used for creating several charts on a periodic basis or single charts for one-time applications. DSG/3000 features include: interactive and programmatic interfaces, data manipulation by projection, restriction, sorting, or transformation, chart layouts that conform to HP corporate design standards, integration with and access to other subsystems, and easy adaptation (localization) to a user's native language.

Interface and Intrinsics

The user has two ways of accessing the design features of DSG/3000. The first is through an interactive interface program called GRAPH, and the second is through a set of routines known as the DSG/3000 intrinsics. Each interface addresses a different type of user.

The interactive interface, GRAPH, appeals largely to nonprogrammers. GRAPH is a series of menus that guide the user through the chartmaking process. Menu and data display are handled through VPLUS/3000* intrinsics, while low-level plotting and device control are handled through the graphics utility routines (see article, page 3).

*VPLUS/3000 is a software system that implements and controls source data entry and provides an interface between terminals or files and any transaction-processing applications.

From the main menu (Fig. 1) the user can choose one of ten selections. Electing to create or edit a chart causes the chart design control menu to be displayed on the screen. Subsequent menus define the data file, choose the chart type, enhance the chart, and select the output device characteristics.

The menus are chained together in a logical sequence so the user may press PREVIOUS MENU and NEXT MENU function keys to step through the various menus needed to create a chart. A typical sequence is as follows:

- Enter main menu and select CREATE NEW CHART
- Select CREATE DATA FILE from chart design control menu
- Enter data and variable names, press NEXT MENU
- Define data and missing data values, press NEXT MENU
- Choose chart type, enter dependent and independent variables, press NEXT MENU
- Define axes scaling, ticks, and grids, press NEXT MENU
- Supply legends, colors, and textures, press NEXT MENU
- Enter titles, press NEXT MENU
- Add annotations to chart, press NEXT MENU
- Select graphics device and characteristics, press NEXT MENU
- Choose text characteristics, press NEXT MENU
- Define font characteristics
- Press PLOT to see the output of your chart.

Alternatively, the programmer may prefer to use the intrinsic interface. The intrinsics can be accessed from five programming languages: Pascal, COBOL, SPL, BASIC, and FORTRAN. The intrinsic interface provides the same capabilities as GRAPH and is especially useful when data files are created programmatically for periodic charts. The

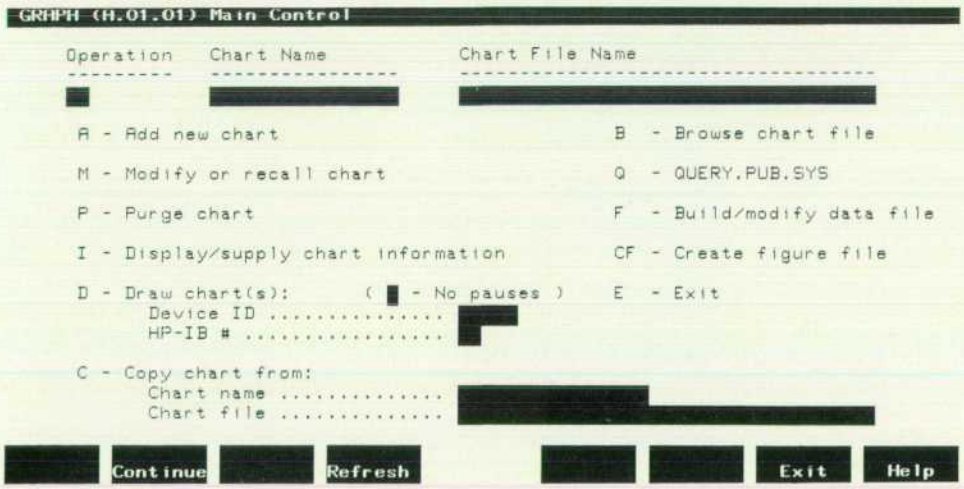


Fig. 1. *The main menu of DSG/3000.*

© Copr. 1949-1998 Hewlett-Packard Co.

chart file remains the same and only the data files are updated so the program can be run at any time without operator intervention. A typical sequence of intrinsic calls is shown in Table I.

Table I

GINITGRAPH	Initializes global storage area
GCREATECHART	Creates fundamental chart
GDATAFILE	Selects data file
GDEFN	Defines variable names
GCHARTTYPE	Chooses chart type
GINDVAR	Defines independent variable
GDEPVAR	Defines dependent variables
GLABVAR	Defines labels in data file
GBAR or GPIE	Selects chart attributes
GAXIS	Selects ticks, grids, axes scaling
GTEXTCONTROL	Sets fonts, colors, sizes
GLABEL	Defines labels
GFONT	Selects font characteristics
GTITLE	Adds titles
GLIMIT	Selects size and placement
GDEVICE	Chooses graphics device
GEXECDATA	Prepares data file
GEXECHART	Plots chart
GCLOSEFILE	Closes file
GTERMGRAF	Terminates graphics session.

Chart Files

A chart file can contain up to 50 charts. Each chart contains five records: chart, label, annotations, data, and font records. The actual data file is kept separate from the chart so the user can change chart types without having to retype data. Chart files are stored as keyed access files. The file directory is contained within the file and a binary tree algorithm is used for fast indexing. Fig. 2 shows the relation between the chart file and the stack. A paging mechanism minimizes the stack space required for large charts by dynamically moving the record that is needed into the GRAF extension area. The GRAF extension area is a global communications buffer located in the DL-DB area of the stack. The record previously in the extension area is updated if necessary and flushed from the stack. DSG/3000 also requires space on the stack for a static communications area and the VPLUS/3000 communications area.

Data Manipulations

DSG/3000 allows the user to select, subset, and sort data. Secondary data can even be computed. These functions are very useful to the user who is visually exploring data or preparing graphical reports.

The internal DSG/3000 data model uses simple relational data base concepts. Data is viewed as one or more tables and each variable represents a column of data. DSG/3000 uses self-describing files (SD files) for these tables. SD files have headers that contain information describing the file variables. This information includes the variable name, type, and position. Before manipulating the user's data, DSG/3000 converts the data file to an SD file. The data manipulation functions are a subset of common relational algebraic functions: projection, restriction, sorting, and transformation. Each function can be applied to a table, or in this case, an SD file.

Projection is the selection of variables. On the GRAPH chart menus, the user selects up to eight dependent variables to plot against one independent variable. Internally, a new SD file is created containing only the variables of interest. For example, consider a data file containing six variables A, B, C, D, E, and F. Plotting A and B versus F would result in the projection shown in Fig. 3 and the creation of a new SD file containing only the variables A, B, and F.

Restriction, or subsetting, is the selection of records of interest. If the user specifies a subset of the data to be plotted, a new internal SD file is produced containing only

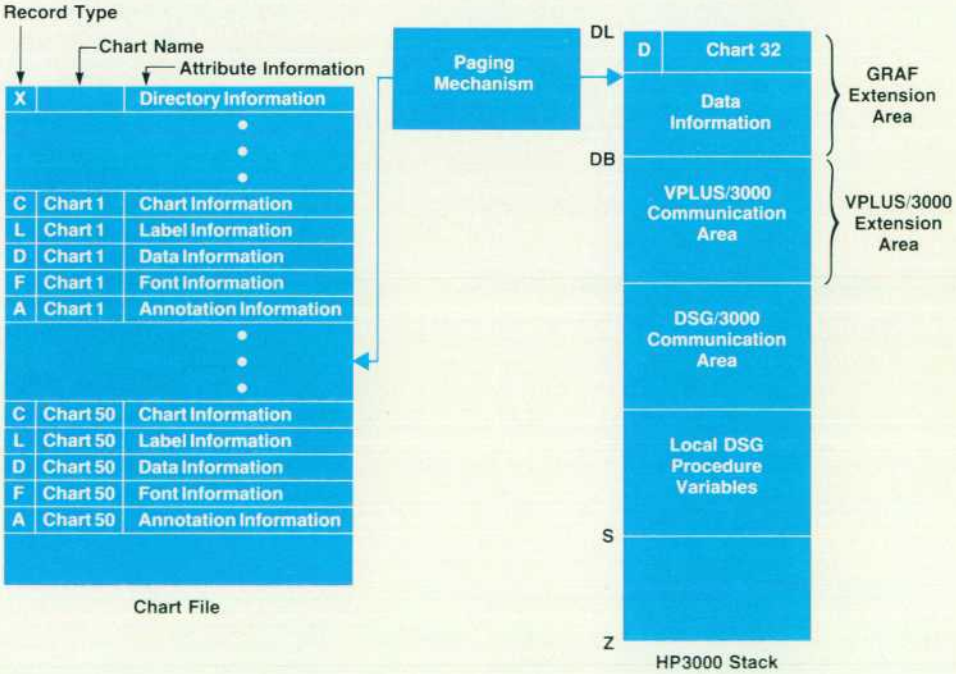


Fig. 2. *A paging mechanism brings the appropriate chart record into the HP 3000 stack and updates the chart file accordingly when a record is modified.*

© Copr. 1949-1998 Hewlett-Packard Co.

Datafile

A	B	C	D	E	F
1	x	73	x	x	a
2	x	73	x	x	b
3	x	73	x	x	c
4	x	73	x	x	d
5	x	73	x	x	e

Project A,B,F →

New Datafile

A	B	F
1	x	a
2	x	b
3	x	c
4	x	d
5	x	e

Fig. 3. *Selecting variables A and B to plot against F causes the projection of A, B, and F and creates a new file containing only those variables.*

Datafile

A	B
10	73
20	75
30	77
40	72

Sort on B →

New Datafile

A	B
40	72
10	73
20	75
30	77

Fig. 5. *Sorting rearranges the records of a data file based on a specified variable.*

the records of interest. For example, consider a data file containing the variables A, B, and C. The subset of A greater than 3 would result in the restriction shown in Fig. 4. This operation would create an SD file containing three variables and only two records.

Sorting is used mainly to arrange records in relation to an independent variable so that the data for a line chart can be plotted in positive X-axis increments (the independent variable). An example (Fig. 5) demonstrates the sorting of a table containing the variables A and B by increasing values of B.

Transformation allows new columns of data to be calculated from currently existing columns. An example of this function might take a data file with variables A and B and create a new variable C which is the sum of A and B (Fig. 6).

In DSG/3000, the restriction and transformation operations are optimized. This means that scanning and parsing are done only once per plot. The relational or arithmetic expression specified is parsed and converted to a small program which can be quickly executed when applied to a particular record. To visualize how the process works, consider the following example. The user has a data file that contains values for orders, shipments, and year. If the user wants to compute and chart a new variable, backlog, to display as a bar chart for the years 1975-1978, the actions shown in Fig. 7 would occur. Assume the user then wanted to change the subset expression and replot the chart looking only at the years 1979 and 1980. DSG/3000 would optimize data manipulation by performing only the necessary restriction.

Design Standards

Computer engineers are good at designing software, and graphic artists are good at making presentation quality charts. DSG/3000 makes use of both their talents. All of the DSG/3000 chart layouts were designed by graphic artists at HP Corporate Industrial Design. The layout guidelines include text size and positioning, line thickness, placement of bars, grids, and tick marks, cross-hatch patterns, and the positioning of multiple charts on the same page. For example, legends are positioned at the top of the chart for up to five variables, and on the left side of the chart for six to eight variables, so the chart will appear balanced. DSG/3000 also has default axis values that leave approximately 10% white space between the highest data point and the top of the axis frame. Default cross-hatch and line patterns are selected to yield maximum visual separation, making it easier for the eye to differentiate patterns. These design standards make it easy to produce nice looking charts consistently.

Integration

Last year HP released several new office products including graphics, word processing, and personal data base systems. Two separate file types enable these products to communicate with each other. Self-describing files (SD files) ease the transportation of data between DSG/3000 and other data bases. Figure files (see article, page 3) allow charts and drawings to be transported easily between HP software products.

SD files are created by QUERY/3000 and HPLIST, and are planned for future products as well. They can also be created by GRAPH in the data prompt menu. Once created, the information stored in the SD file header can be accessed by DSG/3000 to alleviate the task of typing in the variable descriptions again. The user just types in the SD file name and the rest of the information appears on the menu.

A figure file created with DSG/3000 can be included in an HPDRAW slide or merged with text and printed on the HP 2680A Laser Printer system or the HP 2608S Graphics Printer. Figure files allow the user to design charts with DSG/3000 and enhance them with figures or text using HPDRAW (see article, page 13). They also enable users to rotate and adjust their charts. The figure file is composed of relative moves and draws. No application-dependent information is stored in the figure file. Consequently, the figure file can easily be moved from one application pack-

Datafile

A	B	C
1	x	73
2	x	73
3	x	73
4	x	73
5	x	73

Restrict A>3 →

New Datafile

A	B	C
4	x	73
5	x	73

Fig. 4. *Data manipulation by restriction creates a new file with only the records that comply with the restricted variable.*

Datafile

A	B
10	20
20	20
30	20
40	30

Trans C=A+B →

New Datafile

A	B	C
10	20	30
20	20	40
30	20	50
40	30	70

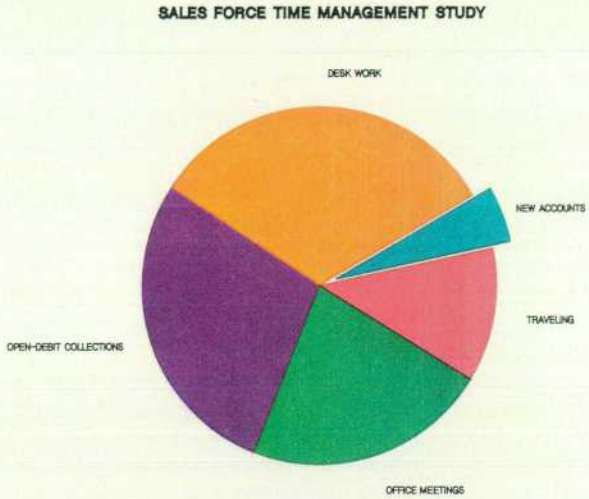
Fig. 6. *Transformation permits the creation of new data columns from currently existing columns.*

© Copr. 1949-1998 Hewlett-Packard Co.

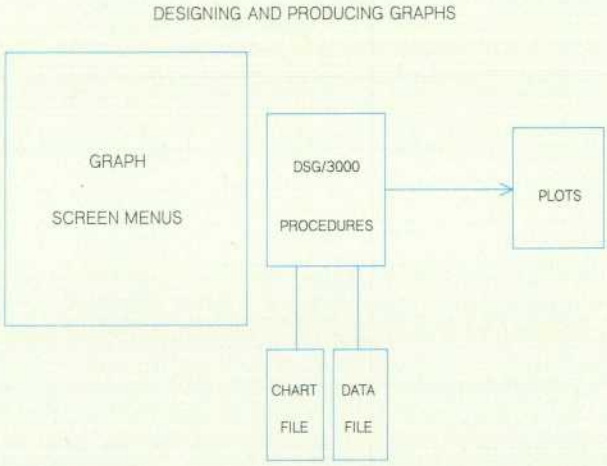
Business Presentation Aids

Most business presentation aids can be categorized into one of four generic types. Each category has a different purpose.

Bar charts are appropriate for charting discrete data.



Pie charts illustrate the percentage composition of a subject.



Line charts are used for charting continuous data.

Slides allow a user to display text, box structures, arrows, or circles.

age to another.

Another aspect of integration relies on a common set of interface standards for all office products. These standards outline menu conventions, function key placement and wording, message handling, and banner format. With all office product interfaces looking the same, the user isn't burdened with mastering several syntaxes and the learning time for new products is shortened.

Localization

The user interface of DSG/3000 can be adapted to another language by modifying the message catalog. The message catalog contains error and warning messages, language and paper size defaults, device coordinates, command strings, font file names, and general messages and prompts. By using EDIT/3000, a user can change the message catalog entries to reflect a native language.

Menus can also be translated to other languages. The FORMSPEC utility available in VPLUS/3000 makes it possible to change the wording on menus. The actual field lengths cannot be changed.

Although DSG/3000 is not fully localized, efforts have been made to simplify use of the product in countries other than the United States. For instance, all selection fields are two characters long, except those of a yes/no nature. This eliminates ambiguities that may arise when field labels are not easily translated.

Characters can be output in seven different languages. The characters are generated through a 7-bit International Standards Organization (ISO) substitution which maps the 7-bit character code sent by the terminal to the 8-bit Roman extension set. An 8-bit solution for foreign characters is currently being proposed to replace the temporary 7-bit

© Copr. 1949-1998 Hewlett-Packard Co.

Datafile

Orders	Shipments	Year
2018	1523	1974
1923	1445	1975
1899	1620	1976
1950	1690	1977
1875	1495	1978
1860	1498	1979

Trans Backlog Restrict Year

New Datafile

Orders	Shipments	Year	Backlog
1923	1445	1975	478
1899	1620	1976	279
1950	1690	1977	260
1875	1495	1978	380

Fig. 7. *A new file containing the variable backlog is created with records that comply to a year restriction.*

ISO substitution.

Flexibility

DSG/3000 has both data presentation (hard copy) and data display (terminal) capabilities. With the option to plot on transparencies, the user can produce high-quality charts for presentations. Best results are achieved by reducing the movement of the plotter pen to 10 cm/s from its top speed of 36 cm/s or higher. A new, quick-drying transparency medium developed by HP's San Diego Division eliminates the need for a pause between pen changes to prevent colors from running together.

Initially DSG/3000 supported only a stick character font. The second release added gothic, script, and roman fonts. The user was also given control over text size and color. The third release introduces three new fonts based on spline curves. These new fonts can be scaled to any size and will retain their smooth curves. They can also be filled to make solid characters. Each of the new fonts includes bold, italic, and outline faces in any combination. They correspond to the faces that typesetters use.

Every good chartmaking package should have a variety of basic capabilities, but it is flexibility that makes a package powerful. Sometimes our product engineers are surprised by an unusual chart created by DSG/3000. The chart in Fig. 8 is a frequently used type that is easily created by DSG/3000.

The Interactive Office

Hewlett-Packard developed the Interactive Office to meet the changing needs of business professionals. The Interactive Office consists of an integrated set of tools for document management, organizational communication, personal support, and decision support. As an element of this system, DSG/3000 was designed to aid office workers in decision making by accessing, interpreting, analyzing, and presenting the information they need to get their job done.

Acknowledgments

The authors would like to thank Sue Crane, Karen Chez, Robert Dea, and all of the other people involved in the development of the three releases of DSG/3000.

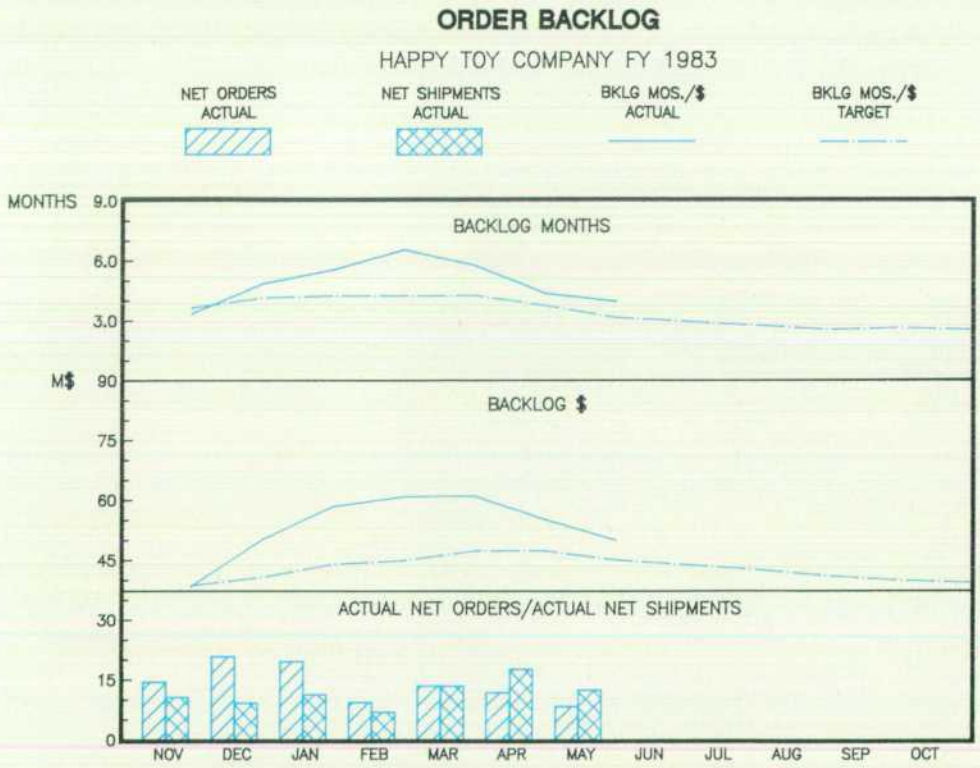


Fig. 8 *This chart combines several DSG/3000 design capabilities.*

© Copr. 1949-1998 Hewlett-Packard Co.

An Easy-to-Use Chartmaker

by Martha Seaver, Robert W. Dea, and Richard J. Simms, Jr.

HPEASYCHART is an interactive chartmaker designed to be used with the HP 3000 family of computers. HPEASYCHART is a simplified alternative to the more sophisticated DSG/3000 business graphics package (see article, page 5). It allows nonprogrammers to create visually pleasing charts without understanding all of the details behind design decisions or file control.

HPEASYCHART does not use a file for data entry. Instead, the user selects a chart type and then enters individual data values to construct the desired output. This data entry method makes HPEASYCHART well suited for one-time applications, such as charts needed as illustrations in a text document or as presentation aids. Periodic charts with unchanging design specifications are more efficiently prepared using data files and a chartmaker like DSG/3000.

HPEASYCHART began as an experimental prototype designed to put graphics capabilities at the fingertips of a variety of office workers. The prototype became such a popular internal tool that a decision was made to expand the program and offer it as a product. To appeal to the nontechnical user in particular, the typical menu-driven interface was modified by adding features that make certain default design decisions so a user can produce handsome charts by responding to just a few simplified menus.

Menu Design

The infrequent computer user may not know the names of chart types or specifications. HPEASYCHART eliminates this problem by displaying icons (pictures) on its main menu to provide the user with an easily identifiable summary of capabilities. Chart type names appear alongside the icons so HPEASYCHART can be used on both graphics and nongraphics terminals. There are no fields to fill in or boxes to check on the main menu (Fig. 1). Function keys control the selection of options and the flow of subsequent menus. Once the user selects a chart type from the main menu, HPEASYCHART displays a design menu containing the icon of the chart to be created. A standard design menu format for all chart types allows the user to create many types of charts without having to learn new concepts. Pie chart data is collected differently from other types but the general flavor of the standard design menu is still preserved.

Instead of presenting a confusing array of options, HPEASYCHART menus are uncluttered. A help menu is available with examples. Operating instructions are minimized by including sample data with each menu option. These samples not only instruct the user on correct data entry, but also serve as default values that allow the user to preview the output. New data overrides the sample values and the previewing function can illustrate how the chart has been altered. The technique of learning by example is used extensively throughout HPEASYCHART to simplify graphics for the inexperienced user.

Fig. 1. *HPEASYCHART's main menu with chart type icons.*

© Copr. 1949-1998 Hewlett-Packard Co.

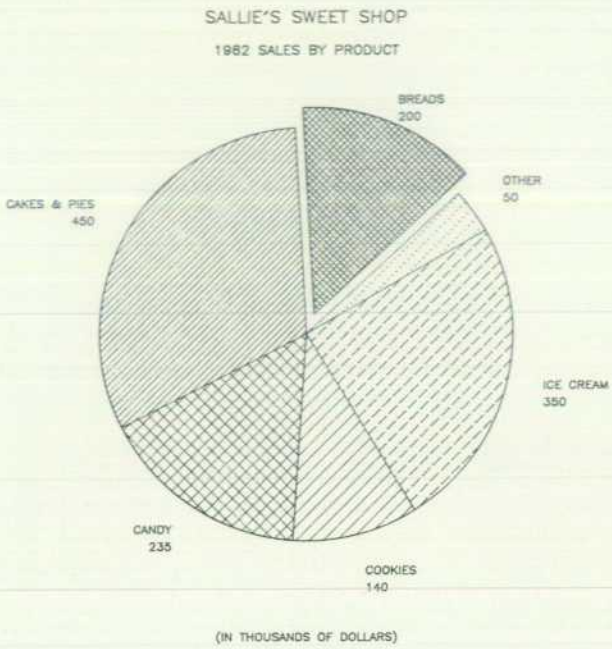


Fig. 2. *A sample chart prepared with HPEASYCHART.*

HPEASYCHART makes option selection simple. For example, on the pie chart design menu a combination of three options may be entered. A standard entry method would involve labeled boxes (fields) for each option. The user would make several keystrokes to reach the appropriate field position and enter an X to indicate an included option. To minimize the number of fields to be filled and the keystrokes to be made, HPEASYCHART designates a SELECT OPTIONS function key that consecutively highlights all eight conbinations. The user simply presses SELECT OPTIONS until the desired combination is highlighted.

HPEASYCHART uncomplicates business graphics by making certain design decisions, but it is also flexible enough to be useful to a broad range of office workers and professionals. It allows the user a choice of output medium, page size, chart positioning, and space for binding, without the need to specify a lot of details about plotting devices, paper dimensions, or exact chart position coordinates. HPEASYCHART menus include simplified parameter options that make it easy to tailor an output format. For instance, to plot a chart on the left side of a page, leaving space for ring binder holes, the user enters L (left) for both chart position and space for binding. This friendly method of design frees the user from remembering syntax and coordinates while encouraging creativity by anticipating many typically encountered format alterations.

The pie chart illustrated in Fig. 2 was produced with HPEASYCHART using the following sequence of steps:

- From the main menu (Fig. 1), press SELECT CHART until the PIE is highlighted
- Press CREATE CHART to arrive at the pie chart menu (Fig.3)
- Enter the necessary titles, labels, and values (the chart could then be previewed on a terminal screen by pressing PLOT TO SCREEN)
- To create the chart, press PLOT TO PLOTTER, which causes the plot menu (Fig. 4) to appear
- Enter the appropriate data for device, page size, etc.
- Press DONE.

Default Files and Parameters

Each time HPEASYCHART is executed a temporary copy of a default DSG/3000 chart file and a default sample data file are created.

The DSG chart file is made up of ten charts: one for each of the eight chart types provided by HPEASYCHART, one for the chart currently being designed, and one for extraneous information. When a user selects a chart type from the main menu, HPEASYCHART replaces the old current chart with a copy of the default chart of the selected type. The default chart is actually a DSG/3000 chart with default options set for such things as font, text size, and axis scaling. The values for these parameters have been chosen to produce a pleasing standard chart. As the user designs the current chart by selecting pen color, line or area texture, and titles, the associated chart in the chart file is modified via calls to DSG/3000 intrinsics. This current chart is where all the user-selected design information is stored. If the chart is saved, the current chart is simply copied to a new

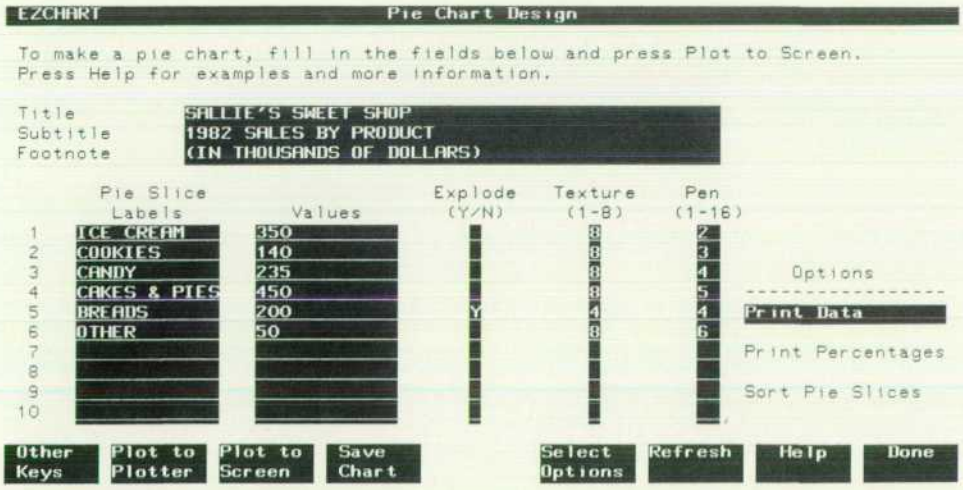


Fig. 3. *Pie chart menu containing the data used to create Fig. 2.*

© Copr. 1949-1998 Hewlett-Packard Co.

```
EZCHART                                  Plot

To plot a chart, fill in the fields below and press ENTER.  Press Help for
more information.

7221C Device Model Number

  M Title Boldness   L - Light            M - Medium

  T Page Medium      P - Paper            S - Scrollable Paper
                     T - Transparency    TP - Transparency (with pauses)

  A Page Size        A - 8.5x11 in        B - 11x17 in          D - 22x34 in
                    A4 - 210x297 mm      A3 - 280x420 mm   blank - 8.5x10.5 in

    Space For Holes  L - Left     R - Right     T - Top      B - Bottom

  W Chart Position  ---------     ---- ----     ---- ----     ---------
                   |         |   | TL | TR |   |    |    |   |    T    |
                   |    W    |    ---- ----    | L  | R  |   |---------|
                   |         |   | BL | BR |   |    |    |   |    B    |
                    ---------     ---- ----     ---- ----     ---------

                                                Refresh   Help   Done
```

Fig. 4. *Plot menu containing the data used to create Fig. 2.*

chart file. Although transparent to the user, DSG/3000 handles the internal structuring of the chart.

The default data file supplies the sample data displayed on chart design menus. This sample data and all user-supplied data are stored internally in a table. There are seven columns in the table: one for textual X-axis values, one for numeric X-axis values, and one for each of five available Y-axis values. When the user saves a chart, the data table is stored in a separate data file that is used in conjunction with the saved chart file to recreate the chart.

Use with Other Office Systems

HPEASYCHART is useful as a stand-alone graphics presentation software system. It is fully compatible with all HP graphics terminals, plotters, and graphics products. It can even be used on nongraphics HP terminals to design and plot charts on an attached plotter. But HPEASYCHART can also be considered as a component of the powerful electronic office system that includes DSG/3000, TDP/3000, HPDRAW, and HPWORD. A chart designed with HPEASYCHART can be enhanced using the expanded design capabilities of DSG/3000. Many experienced graphics designers prefer to make a quick, initial design with HPEASYCHART and then take the files into DSG/3000 to make special embellishments. HPEASYCHART's save feature can be used to create a figure file that can be incorporated into an HPDRAW drawing or an HPWORD document. Figure files allow text and graphics to be merged on the HP 2680A Laser Printer. HPEASYCHART permits interaction between sophisticated and novice users and it makes graphics and document preparation simpler, less time-consuming, and more cost-efficient.

Acknowledgments

The authors would like to thank everyone who helped make HPEASYCHART a released product. We thank Sandro Recchione for throughly testing the product, John Deck for writing an excellent manual under time pressure, Marilyn Johnson and Chris Kocher for handling the marketing and promotional efforts, Kim Sasko for her on-line support assistance, Ellen Brigham for training assistance and continued off-line support, and Ann Koehler for quality assurance finishing and testing.

© Copr. 1949-1998 Hewlett-Packard Co.

Convenient Creation and Manipulation of Presentation Aids

by Chayaboon Purnaveja and Janet Swift

THE USE OF COMPUTERS to assist in the production of visual aids has increased significantly in recent years. As a result, graphics users have begun to develop specialized needs that extend beyond simple tasks such as adding an arrow to a chart or modifying an annotation. They want products that can create customized drawings and simplify graphical data manipulations. HPDRAW and the HP 3000 Computer System fulfill those needs.

HPDRAW is a highly interactive graphics design system that allows data to be entered through direct graphics or menu interfaces. The editing and manipulation of freehand drawings, symbols, and signatures is accomplished by point and push operations using a graphics cursor instead of the typical coordinate point designation method. When HPDRAW is used in conjunction with the HP 17623A Graphics Tablet, figures can be created by moving a stylus across the tablet. Signatures can be digitized and entered into letters and documents. Figures generated with HPDRAW can be saved, enhanced with area fill, rotated or scaled, and even combined with a drawing, which can be merged with text.

Figure files contain graphic data (see article, page 3). They are a convenient means of interchanging graphical information between HP business graphics software products. During the initial development of HPDRAW there were no plans for figure files and much of the integrated text and graphics strategy had not been determined, so the interface was designed to allow for the addition of large new sets of capability at a later date. This design permitted the support of figure files and new capabilities that weren't originally foreseen. There were no immediate plans during the design phase for use of the 2680A Laser Printer as an output device or for use of a tablet like the 17623A in menu picking and object placement. The modular structure of the software made it easy to add new features in parallel while other parts of the product were still under development or being tested. The design (a basic functional outline is shown in Fig. 1) is a combination of top-down and bottom-up structures that allowed the design team to get the high-level portions of the user interface up and running while simultaneously developing the low-level screen and file handling capabilities. This methodology allowed the engineers to identify and respond to some serious performance and user interface issues early in the product development cycle.

Creating a Drawing

HPDRAW allows a user to create drawings directly by manipulating the display cursor with keyboard control buttons or with the stylus of the 17623A Graphics Tablet. Drawings may also be created from menus that provide access to a library of figure files containing commonly used graphic images such as arrows, flow chart symbols, devices, geometric figures, etc.

To create the drawing shown in Fig. 2, the user arrives at a figure menu (Fig. 3) and enters the name of an image from the HPDRAW figure library. After all the desired images have been edited into the drawing, annotations with a choice of font styles can be added from the text menu (Fig. 4).

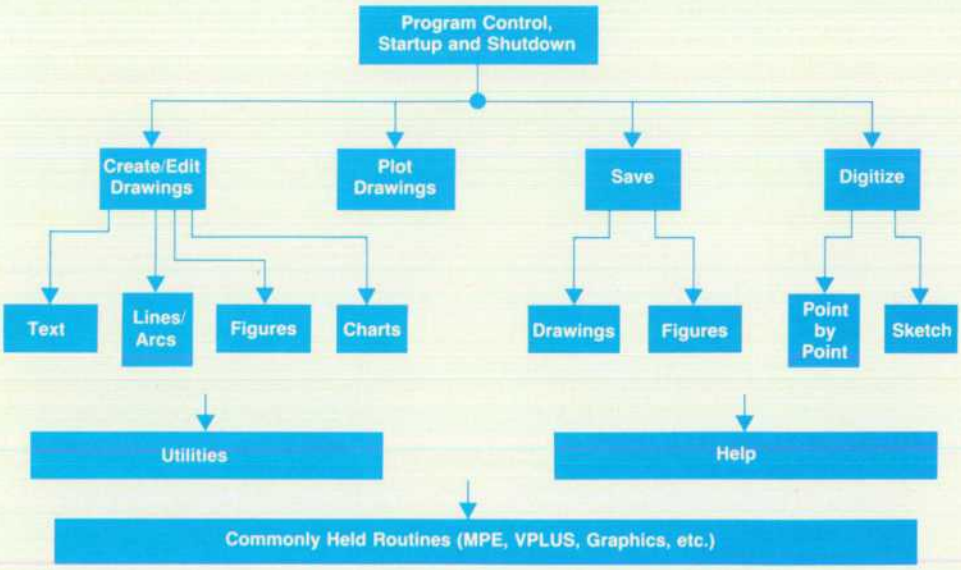


Fig. 1. *The structure of HPDRAW is designed to allow parallel development of independent capabilities and to accommodate new software modules as they become available.*

© Copr. 1949-1998 Hewlett-Packard Co.

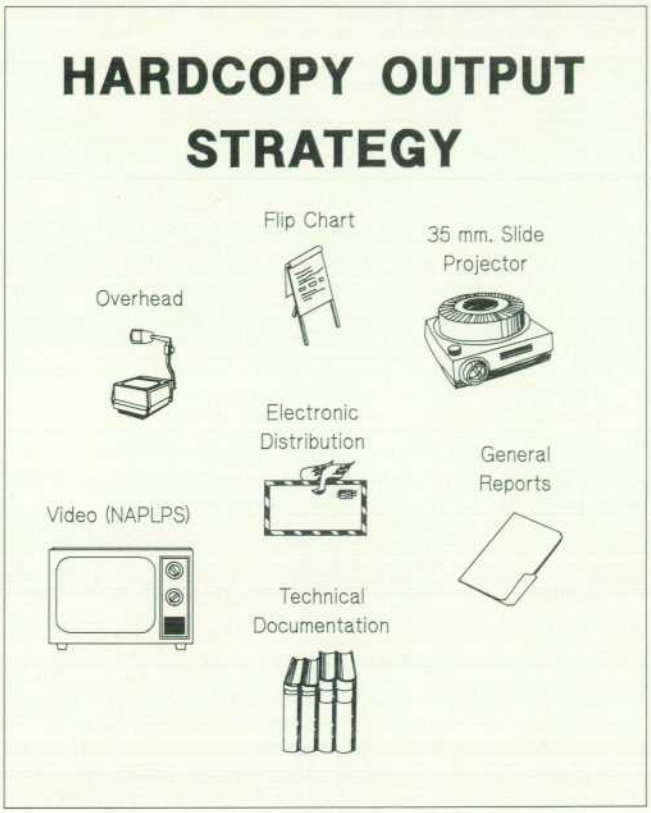


Fig. 2. *An example of an HPDRAW drawing created with images from a figure library.*

Figure Adjustments

Figure adjustment menus with interactive graphics are available to help the user edit and manipulate graphical data. These facilities use matrix transformation to move, scale, and rotate figures, which are composed of 1×3 data coordinate matrices.

$$\begin{bmatrix} x & y & 1 \end{bmatrix} \qquad \text{Data Coordinate Matrix}$$

A 3×3 matrix is used to represent a figure's transformation. When a figure is created, its default transformation is the identity matrix. When a subsequent transformation is applied to a figure, its coordinate matrices are multiplied by the appropriate transformation matrix. The overall transformation of the figure is then represented by the resulting matrices. The basic transformations that can be applied to a figure are translation, rotation, and scaling. The following matrices represent each transformation:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{The Identity Matrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x & y & 1 \end{bmatrix} \qquad \text{The Translation Matrix}$$

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{The Rotation Matrix}$$

$$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{The Scaling Matrix}$$

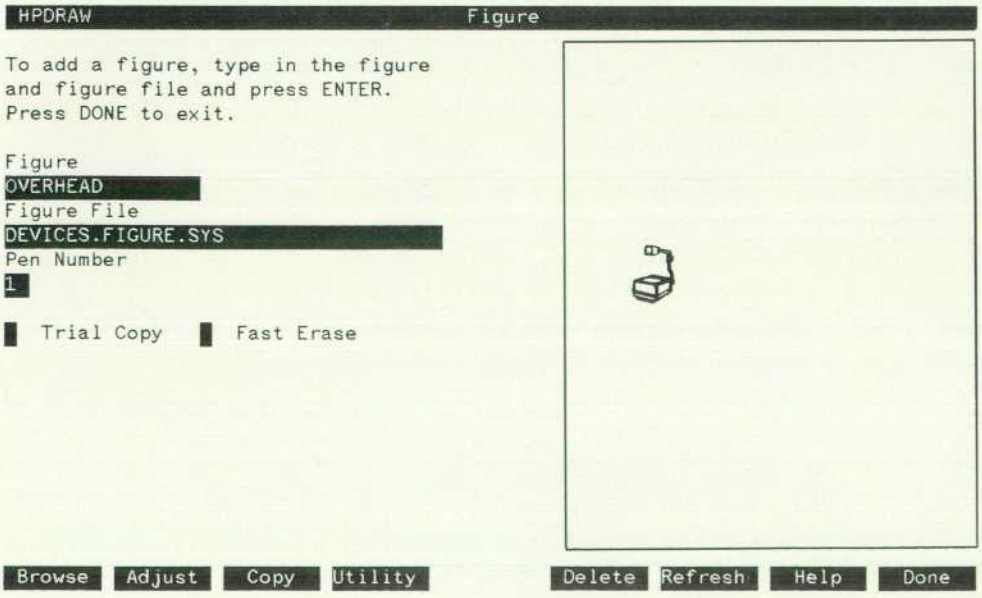


Fig. 3. *The figure menu displays an image requested from the figure file library.*

© Copr. 1949-1998 Hewlett-Packard Co.

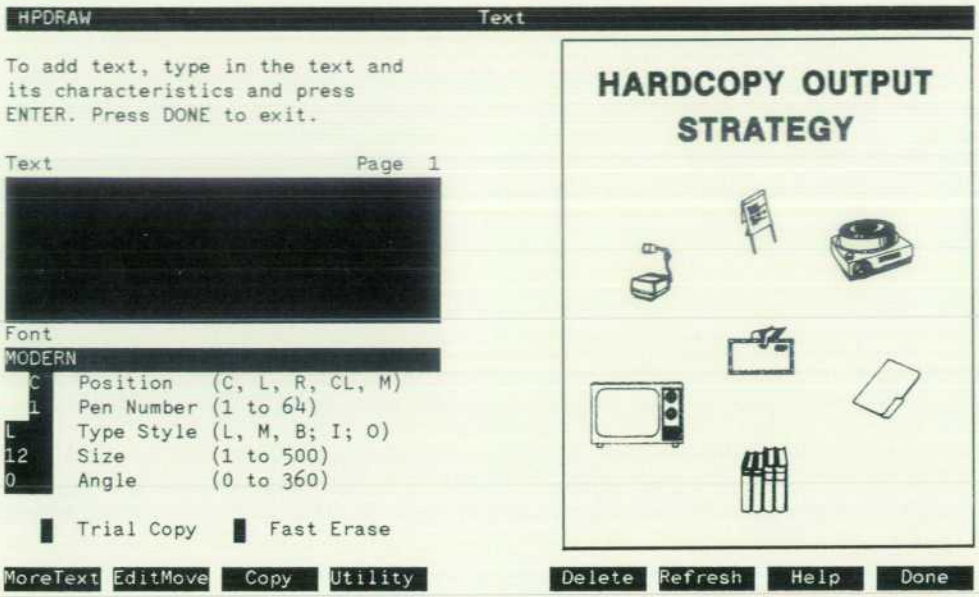


Fig. 4. *The text menu permits the user to add annotations to a drawing.*

The translation matrix can be used to move a figure x units along the X axis and y units along the Y axis. The rotation matrix represents a rotation of θ radians counterclockwise from the positive X axis. The scaling matrix scales a figure by the factors of S_x in the X direction and S_y in the Y direction.

When visual scaling or rotation is specified, a reference guide is drawn at the origin of the transformation. In the case of scaling, a reference rubberband arrow is drawn showing the size of the intermediate figure (Fig. 5a). The user moves the cursor along the direction of the arrow to indicate how much the figure should be expanded or shrunk. The length of the arrow is used as the scaling reference.

In the case of rotation (Fig. 5b), a reference circle appears around the intermediate figure and a reference arrow is drawn to indicate the current angle of rotation specified for the figure. The user moves the cursor along the circumference of the circle to indicate the angle through which the figure should be rotated.

Help Facility

A Help facility is available to give the user information and answer commonly asked questions. A novel aspect of this facility is the use of graphics to illustrate options and convey information. A user is presented with visual examples of font styles, line textures (Fig. 6), and other options so the computer graphics designer can actually see the capabilities of the medium. The Help facility is organized much like an index in a reference manual. Users can look at a list of topics (both global and local to the last used menu) and receive information about the topic by entering a keyword on the Help menu.

Graphics Tablet

One of the special capabilities of HPDRAW is support of the 17623A Graphics Tablet on the HP 2623A and HP 2627A Terminals. With this device, the user can track the cursor by moving the stylus across the tablet instead of using the four cursor movement keys on the keyboard. To use the tablet as a graphics input device, the user moves the cursor to the desired location and presses the stylus down. It is an easy and natural way for a person to select a location.

The tablet is also capable of transferring continuous coordinate data from a sketch (Fig. 7). A digitization facility allows the user to trace a sketch and convert it into a figure file. The figure can then be included on a drawing. This unusual capability gives users access to custom-designed figures, logos, and complex decorative features.

The tablet interface technology used in HPDRAW for cursor tracking and tracing and/or menu picking functions

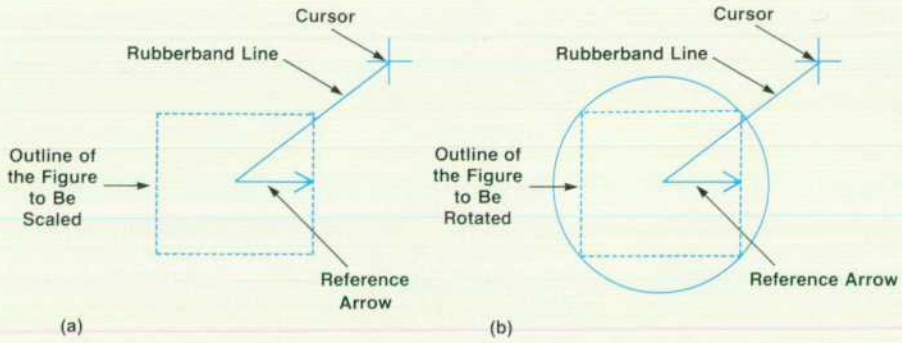


Fig. 5. *Examples of graphic manipulations on the figure scaling and rotation menus. a) Visual scaling of a figure is accomplished by expanding or contracting the rubberband line. b) The angular placement of the rubberband line from the reference arrow determines the rotation of the outlined figure.*

© Copr. 1949-1998 Hewlett-Packard Co.

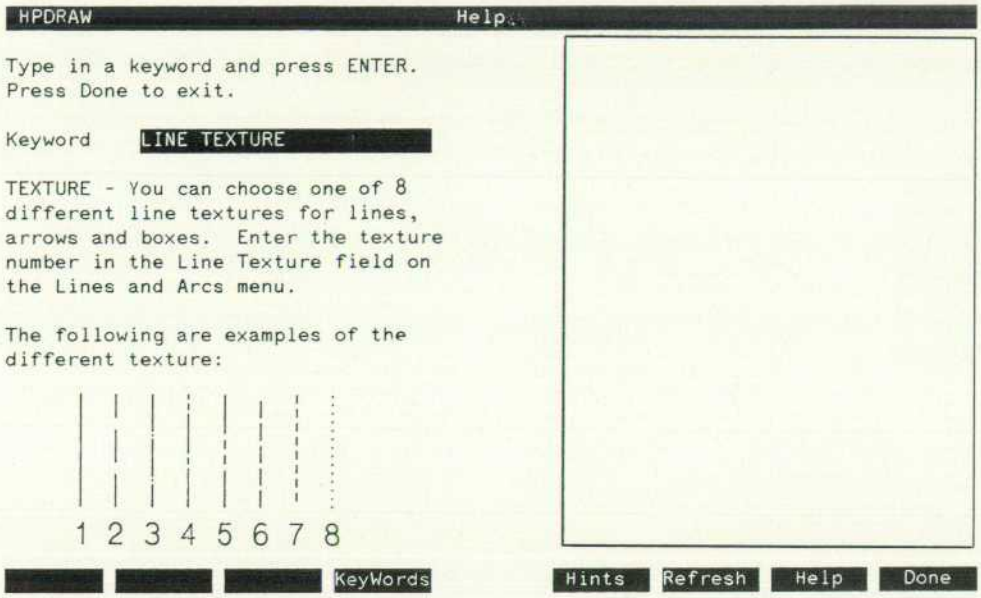


Fig. 6. *The Help menu responds to keywords with text and graphical information so the user can see design options.*

is an improvement over other interfaces, like a touchscreen or mouse, which are only suitable for one of the tasks but not the other. The touchscreen works well for menu picking but does not provide sufficient resolution to serve as a graphics input device. The mouse is good for menu picking and graphics input but a tablet is preferable for tracing applications.

HPDRAW uses VPLUS/3000* as a menu generator. Combining the two products was not a trivial task because VPLUS/3000 has no knowledge of graphical data and tries to interpret it as input from the keyboard. The HPDRAW tablet interface acts as a preprocessor to VPLUS/3000. It captures the data, processes it and lets VPLUS/3000 take over when necessary.

*VPLUS/3000 is a software system that implements and controls source data entry and also provides an interface between terminals or files and any transaction processing applications.

Debugging

When HPDRAW was being developed, the only debugging capability for the HP 3000 was a FORTRAN tool. It worked nicely when an engineer knew approximately where the code problem was happening and when the na-

Fig. 7. *When used with HPDRAW, the HP 17623A Graphics Tablet allows a user to create freehand drawings, digitize graphical data, and make menu selections.*

© Copr. 1949-1998 Hewlett-Packard Co.

ture of the problem was unrelated to program flow.

The HPDRAW design team needed a debugging tool with different capabilities. In particular, the engineers wanted to be able to examine an adjustable amount of information about program flow on a module-by-module basis. This was accomplished by creating a separate MPE** file to serve as a table for holding trace specifications.

Each module of HPDRAW contains checking code to determine if tracing is being requested. Information about routine names, parameters received, and values computed or accessed within the routine can be traced by changing the entries in the MPE file. Furthermore, tracing can be specified at the routine level as being on (trace me), off (don't trace me), directive (trace me and all routines called by me), or inquiry (trace me if my parent was traced.) By judiciously setting the options, it is possible to control the amount of tracing information that is displayed during execution. This saves time. It also allows engineers to focus on certain areas of the program flow and permits the tracing of specific variables. This debugging capability is now available to field service engineers for assistance in problem identification.

**MPE = Multiprogramming Executive, the HP 3000 Computer operating system

Acknowledgments

Phil Walden and Jim Long made HP 1000 software available to the design team. Their work was helpful in determining objectives during the development phase of HPDRAW.

Graphics Capabilities on a Laser Printer

by Tamara C. Baker, William J. Toms, James C. Bratnober, and Gerald T. Wade

TRADITIONALLY ALL TEXT DOCUMENTS have been prepared on printers and all graphics hard copy prepared on plotters. To understand why, consider that it would take 10 minutes for an HP 7221 Plotter running at full speed to plot the text of this paragraph. Plotters are designed to generate vectors (straight line segments). The HP 2680A Laser Printer, by contrast, is a raster image device similar in some ways to a black and white television CRT screen. A page of printer output is composed of small circular black dots analogous to the picture elements (pixels) on a CRT. There are 180 of these dots per inch in both the horizontal and vertical directions. Patterns of dots are grouped together into rectangular cells that form characters. A set of these characters is downloaded into the printer's memory and used to generate the text of the printed image on the page. To print a graphics figure the HP 2680A must be given a set of cells that collectively form the desired figure. This technique is fundamentally different from the vector generation approach to graphics, which more closely approximates the way an artist draws a figure. It is this difference between raster and vector generated output that has caused all documents to be printed and all graphics to be plotted until the recent release of the Interactive Formatting System/3000 (IFS/3000).

Formatting

IFS/3000 includes two major components. One is a set of intrinsics (routines) to allow programmatic control of document format. The other is IFS/2680, a program that allows interactive definition of the format of documents to be printed on the HP 2680A Laser Printer. The HP 2680A Graphics Package expands IFS/3000 by giving the user capabilities to print graphics output with programmatic

```
LPS Interpreter (A.00.00) HP36580(c) COPYRIGHT Hewlett-Packard Co. 1982

  IFS/3000 Intrinsics (A.01.00)

).help convertfigure

CONVERTFIGURE:
  Converts the specified figure into a raster image file.

  Syntax:
    .conv[ertfigure] ⟨figfilename⟩ ⟨figname⟩ ⟨rastfilename⟩ &
                     ⟨outputdev⟩ ⟨imageheight⟩ ⟨units⟩      &
                     ⟨imagerotation⟩

                     ⟨outputdev⟩      : 2680a
                     ⟨units⟩          : 0 = dots, 1 = inches, 2 = cm, 3 = mm
                     ⟨imagerotation⟩  : 0, 90, 180, 270

  Example:
  .convert figfile figure_name rastfile 2680a 3.25 1 90
```

Fig. 1. *The Help facility of the laser printing system interpreter for the HP 2680A Laser Printer provides the user with command descriptions and examples.*

© Copr. 1949-1998 Hewlett-Packard Co.

intrinsics or through the laser printing system interpreter.

To print graphics on the HP 2680A Laser Printer, graphical information from a figure file must be converted to a raster image and stored in a raster image file. Then the user loads a copy of the raster image into the memory of the HP 2680A and issues a print command to place the graphics on the page. The seven new graphics intrinsics that add this capability are:

PCONVERTFIGURE converts a figure in a figure file to a raster image. Since it is much faster to print a raster image than a figure, PCONVERTFIGURE can be used to ready the graphics, possibly during off hours, for printing later.

PLOADRASTER loads an existing raster image into the HP 2680A memory. Once a raster image has been created with PCONVERTFIGURE, it must be loaded into the printer before it can be printed.

PPRINTRASTER prints a raster image that is loaded in the HP 2680A memory on the current page.

PDELETERASTER deletes a raster image from the HP 2680A memory. The HP 2680A can hold no more than 32 raster images at one time in its memory. PDELETERASTER is used to make room for more images.

PFLASHRASTER loads an existing raster image into the HP 2680A memory, prints it on the current page, and deletes the raster image from the HP 2680A memory. To make one print of a raster image, the user can combine the three steps PLOADRASTER, PPRINTRASTER, and PDELETERASTER in this intrinsic.

PPRINTFIGURE takes a figure in a figure file, converts it if necessary, then loads it and prints it. The conversion decision is based on whether there is an existing raster image of the right size and orientation, and whether the figure has been modified since the raster image's creation. All the previous intrinsics are low-level; they provide direct access to printer features. PPRINTFIGURE, on the other hand, combines low-level features with some printer memory management and conversion optimization. This intrinsic is used by TDP/3000 and HPWORD.

PFIGUREINFO returns the size, creation date, and other information about figures and raster images.

These intrinsics provide a flexible way of printing figures and the ability to store several raster images in printer memory. The HP 2680A can be configured with up to two megabytes of memory. This large amount of storage can be used to eliminate the repeated loading time of a raster image if it will be printed more than once in a job.

The laser printing system (LPS) interpreter provides a nonprogrammatic interface to most of the IFS/3000 intrinsics and gives the user access to the features of the HP 2680A through commands embedded in a text file or entered interactively on a line-by-line basis. If the interactive mode is used, the user receives an error message whenever a command is entered incorrectly. The user also has the benefit of an interactive Help facility (Fig. 1), which lists each command with an example. A description of the command and its parameters appears with the examples. These commands allow the user to control such things as the character font or form being printed, the creation and placement of raster images, or the location and orientation of printing. The user can print data in a form using named fields and can also merge text and graphics at print time. The LPS interpreter sends the text file to the HP 2680A and executes formatting commands by calling IFS/3000 intrinsics.

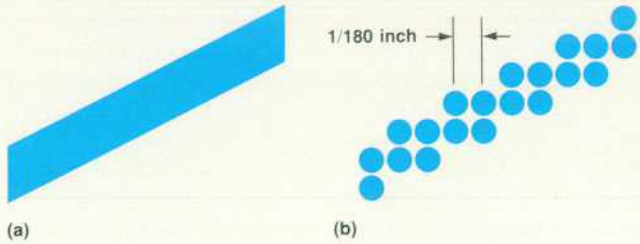


Fig. 2. *The HP 2680A Graphics Package uses a vector-to-raster conversion algorithm to simulate conventional graphic vectors (a) with printed raster images (b).*

Fig. 3. *This HPDRAW drawing contains 12,000 vectors, which would translate to 2.5 million dots if the raster image were printed on 8.5×11-inch paper.*

© Copr. 1949-1998 Hewlett-Packard Co.

Plotting to the HP 2680A

Business graphics on the HP 3000 are created with DSG/3000, HPEASYCHART, and HPDRAW through a set of graphics utility routines. These routines are responsible for providing a device-independent interface to the applications programs and for driving the graphics devices. The internal representation of graphics figures consists of vector commands like MOVE, DRAW, and TEXT. To be able to plot to the HP 2680A Laser Printer all the graphics commands must be converted into a raster format.

Plotting to the laser printer from any of the graphics software products is a multiple-step procedure. First, all graphical commands are reduced to either MOVE or DRAW commands forming simple line segments. For instance, the three commands:

```
MOVE 0,0
DRAW 0,10
DRAW 10,10
```

generate two line segments, one from (0,0) to (0,10) and the other from (0,10) to (10,10). After all the commands are processed, the vectors are then translated into a raster image (Fig. 2) by a vector-to-raster conversion algorithm.

For each vector, the conversion algorithm must first determine which dots should be turned on (black) to represent the vector and then store the information in the bit map array that is the raster image. To give an idea of the size of the task, the HPDRAW drawing, Fig. 3, is made up of about 12,000 vectors. If it were printed on 8.5-by-11-inch paper, the raster image would have more than 2.5 million dots (total of all dots, black and white).

After the vector-to-raster conversion is done, the raster image must be partitioned into HP 2680A-specific linked character cells and saved in an MPE* disc file.

Finally, since the HP 2680A is a spooled device, the raster image must be communicated to the printer through the MPE spooler. The spool file is created and two commands are written in the file to print the image. For example, the first command might be summarized as: store this partitioned raster image in your memory as image #1. A copy of the image would follow the command. The second command would then be: print image #1 at position (x,y), a coordinate pair indicating a point on the current page.

Vector-to-Raster Conversions

Devices like the HP 2680A Laser Printer are capable of turning on any addressable point on a page, but they cannot draw lines directly like a plotter. Some means of decomposing lines into individual dots must be provided to create graphics on raster devices. Interpolation, the most obvious way to perform such decomposition, is unacceptable because it tends to leave gaps in the line and is computationally slow, since it requires floating-point calculations to determine the position of each dot.

The algorithm first used for the HP 2680A Graphics Package was originally developed by J. E. Bresenham.[1] It has the advantage of requiring only integer additions, subtractions, and arithmetic shifts (multiplications by 2). The algorithm incrementally determines the position of a point p_i in a vector, based on p_{i-1} and an error term. The octant of the vector is determined by its orientation. Octants are 45-degree areas taken from a standard Cartesian coordinate system, where octant 1 contains vectors at angles of 0 to 45 degrees, and so on, counter-clockwise around the axes (Fig. 4). A line in the first octant is plotted by stepping through raster positions from the endpoint with the smaller coordinates up to the opposite endpoint.

When stepping along a vector there are three allowable direction choices. Point p_i may be reached by a step in the X direction, the Y direction, or both directions from point p_{i-1}. But once a vector's octant location is determined, the choices are limited to two directions. A vector in the first octant may be stepped out in only increasing X or increasing X and Y directions. The two possible directions change with respect to the octant in which the vector lies. They are labeled M1 and M2; in the first octant M1 is increasing X and M2 is increasing X and Y.

During vector conversion an error term is calculated at each incremental step. The conversion algorithm attempts to minimize this term at each point on the vector by stepping in the proper direction and adjusting the term depending on the direction. For vectors in the first octant, the iterative relationship for error term E_i is as follows:

$$E_1 = 2*\Delta y - \Delta x$$

$$E_{i+1} = E_i + 2*\Delta y - 2*\Delta x \quad \text{if } E_i \geq 0 \text{ Step in M2}$$

$$= E_i + 2*\Delta y \quad \text{if } E_i < 0 \text{ Step in M1}$$

The values used in the above equations, as well as the directions M1 and M2, will change for different octant orientations.

VECTRAST, the first vector-to-raster conversion program used in the HP 2680A Graphics Package, required some special considerations when implemented on the HP 3000 Computer. The maximum-size output page initially anticipated was 11 × 17 inches with 180 dots per inch. This translates into roughly 6 million bits of data space to store the raster image, or about 378K words on the host computer. Since the HP 3000 Computer has a data space (stack) limit of 31,223 words, such an image could not be supported. To overcome this limitation, the strip or raster band method was used.

The strip method of processing required dividing the

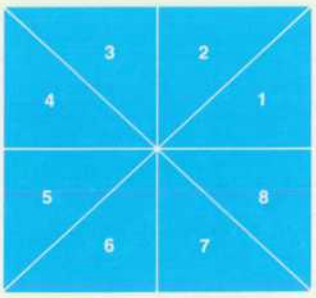


Fig. 4. *A vector can be categorized by octants. In each octant there are two allowable directions of movement between the endpoints.*

*MPE = Multiprogramming Executive, the HP 3000 Computer operating system.

© Copr. 1949-1998 Hewlett-Packard Co.

raster image into strips of a size that would fit into available memory. Each strip was processed only once, then all vector portions falling within the strip were plotted. Since the input vectors were not sequenced in the same order as the strip processing, some type of processing hierarchy had to be imposed. For the first release, a scheme based on sorting the input was developed.

Vectors received from the input file were sorted according to the smallest Y component of their endpoints. These vectors were then input and plotted in the current raster band. If a vector continued out of the strip, it was placed in a list for processing in the next raster band along with the coordinates of the last point plotted in the previous band. This scheme had the advantage of being useful for any image size. The complexity of the drawing was governed by how large the active vector list grew. As the drawing of each vector on the list was completed, the space it occupied on the vector list was removed and returned to the free area.

Data space limitations caused the VECTRAST program to be run as a separate process. On the HP 3000, each process is given its own maximum 31,223-word limit of stack space. If the subsystem is not spawned separately, its data area is decreased by the amount used by its callers.

Conversion Refinements

The Bucket Brigade is the name of the conversion technique used to improve the VECTRAST program; it decreases the time spent sorting vectors and simplifies Bresenham's conversion algorithm without significant loss of accuracy in the placement of dots.

To produce an arbitrary-size raster image with limited data space, a conversion method must sort the input vectors, divide the raster image into bands, and finally convert each band, one at a time. The Bucket Brigade technique reduces the time spent in vector sorting without increasing the time spent in conversion. The method is based on the observation that input vectors need to be sorted into separate bands, but not sorted within each band. The Bucket Brigade eliminates the full sort of input vectors, replacing it with a specialized two-phase bucket sort.

In phase 1 (Fig. 5a) main memory is allocated (it will be used differently for the two phases of the process) and divided into n buckets, where n is the total number of raster bands in the final image. As each vector is read from the input file, it is placed into the appropriate bucket. Long vectors are broken into subvectors that lie entirely in one raster band and the subvectors are then added to the appropriate buckets. When a bucket is filled with vectors, it is written to a temporary scratch file and its memory space is emptied so the bucket can be refilled. After all the vectors have been read, any partially filled buckets are flushed to the disc file.

Phase 2 redefines the use of main memory (Fig. 5b). It allocates room for only one bucket of vectors plus one raster band of dots. The raster band is first cleared and the bucket file is read into the bucket area. The vectors in any bucket for this raster band are then converted and recorded in the raster image area. There is no carryover of long vectors since they have already been broken into subvectors that fit entirely into this band. When all the buckets for this raster band have been processed the complete band is written to the output file. The raster band and bucket are cleared and the bucket file is scanned for a bucket of vectors in the next raster band. This process is repeated until all the raster bands have been processed.

This specialized bucket sort uses less CPU time than a full sort since vectors in a bucket can be in any order. Furthermore, the number of disc accesses are reduced by transferring complete buckets of vectors rather than the individual vectors themselves. If a bucket can hold 400 vectors then the number of disc accesses is reduced by up

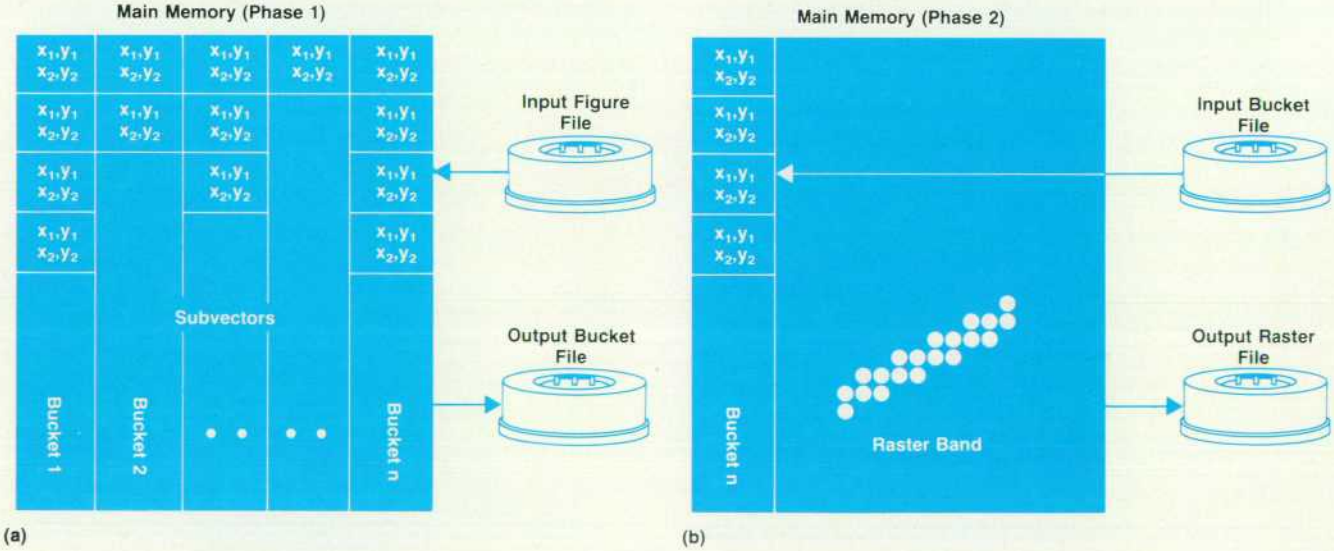


Fig. 5. *Memory use during Bucket Brigade vector-to-raster conversion. (a) Phase 1 fills memory buckets with subvectors that fall entirely with each raster band. (b) Phase 2 converts and records all the subvector buckets associated with a given raster band and then writes the completed band to an output file.*

© Copr. 1949-1998 Hewlett-Packard Co.

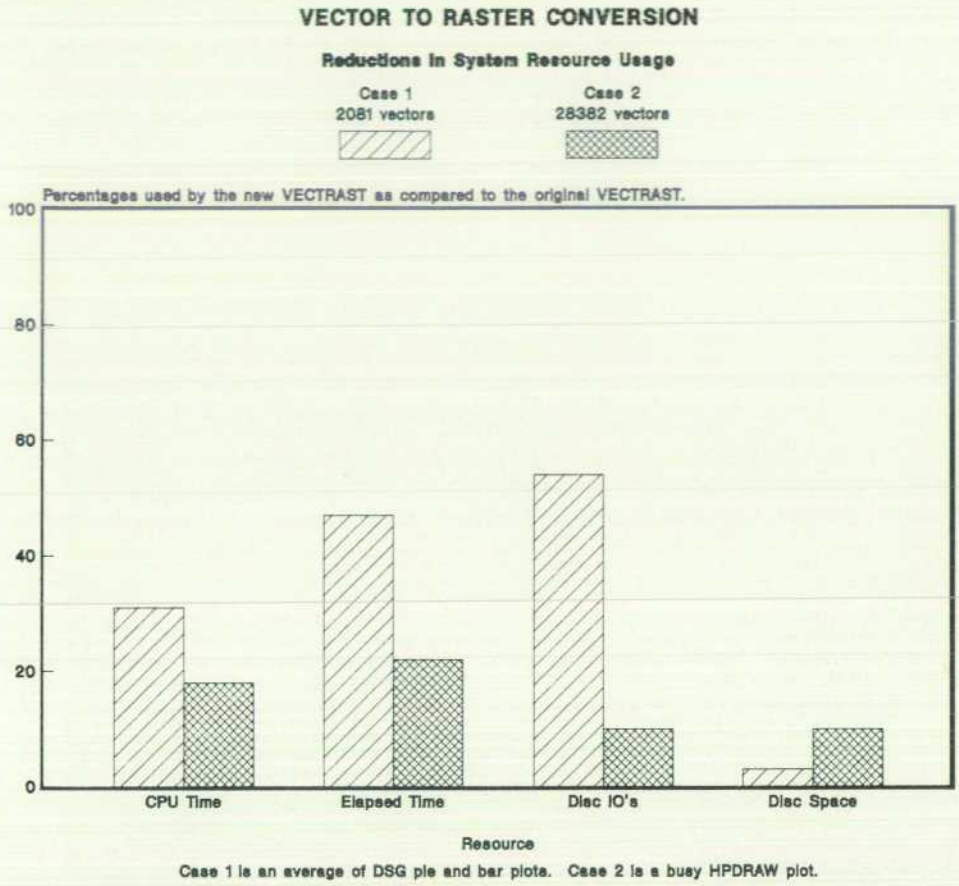


Fig. 6. *Benchmark comparisons illustrating the improved performance of the new VECTRAST using the Bucket Brigade technique over the early VECTRAST.*

to 400:1. Usually a full sort involves several stages so each vector is handled more than once. Consequently, the reduction can be even greater since buckets are written only once. A full sort requires large index files and several scratch files. Since the bucket file is the only necessary scratch file, disc space requirements are reduced. The process of subvectoring during the sort process frees memory space used for remembering long vectors that reside in more than one raster band. This main memory can then be used to make the buckets and raster bands larger. The overall result is a further reduction in disc I/O without increasing CPU time.

Two benchmark tests were run using the original VECTRAST program for vector-to-raster conversion as a standard. One was a simple bar chart which involved about 2,000 vectors. The second case was a large schematic which contained 28,000 vectors. The Bucket Brigade improvements in the use of CPU time, total elapsed time, disc I/Os and disc space are best summarized by Fig. 6. The amounts of each resource used by the original program are set to 100% for both tests and Bucket Brigade performance is expressed as a percentage of the original's. The original program uses the full presort of vectors technique. Note that as the number of vectors increases, the benefits of the Bucket Brigade technique become more pronounced.

The application program tool SAMPLER/3000 revealed that a cosine function was being called each time a diagonal vector's corresponding raster width was calculated. Development engineers realized that by substituting simple arithmetic statements for this trigonometric routine each vector could be processed faster.

The problem became how to determine which dots must be turned on to approximate a vector of an arbitrary width without using trigonometric functions. The solution involves a vector category scheme. A vector can be specified by its endpoints, (x_1,y_1) and (x_2,y_2), and its width. The width of the vector should be approximately half on each side of an imaginary line that connects the endpoints. Consider four cases:

1. The vector is horizontal $(y_1=y_2)$ and can be plotted as a series of horizontal lines from $y=y_1-(\text{width}/2)$ to $y=y_1+(\text{width}/2)$. Width is expressed as a purely vertical dimension spanning x_1 to x_2 (Fig. 7a).
2. The vector is vertical $(x_1=x_2)$ and can be plotted as a series of vertical lines from $x=x_1-(\text{width}/2)$ to $x=x_1+(\text{width}/2)$. Width is expressed as a purely horizontal dimension spanning y_1 to y_2 (Fig. 7b).
3. The vector is primarily horizontal $(|x_2-x_1|>|(y_2-y_1|)$. This is similiar to the purely horizontal case where width should be plotted vertically. To do this on a diagonal line, x is stepped from x_1 to x_2, one dot at a time. At each value of x, y is calculated based on the vector endpoints. Then dots at width/2 are plotted above and below this value (Fig. 7c). The equations used are:

© Copr. 1949-1998 Hewlett-Packard Co.

The slope of a line is $m=(y_2-y_1)/x_2-x_1)$
The Y intercept is $b=y_1-(m*x_1)$
For any x, $y=m*x+b$

4. The vector is primarily vertical ($|x_2-x_1|\leq|y_2-y_1|$). This is similiar to the purely vertical case where width should be plotted horizontally. Values of y are stepped from y_1 to y_2, one dot at a time. At each value of y, x is calculated, and the points at width/2 to the left and right of this value are plotted (Fig. 7d). The equations used are:

The slope of a line is $m=(y_2-y_1)/(x_2-x_1)$
The Y intercept is $b=y_1-(m*x_1)$
For any y, $x=(y-b)/m$

This technique eliminates all calls to trigonometric functions and their associated overhead. Only simple arithmetic calculations are used. Bresenham's original algorithm achieves greater accuracy, limited by the dot or pixel resolution. This simpler algorithm produces comparable results but uses significantly less CPU time.

The benchmark results in Fig. 6 were produced using the arithmetically simplified replacement of the Bresenham algorithm. The new VECTRAST on the HP 3000 Computer can handle conversion of approximately two billion vectors into a raster image of over 95 million pixels. On the HP 2680A Laser Printer this translates into a plot 11 inches wide and over 22 feet long. It does this using only 40,000 bytes of main memory.

Algorithm refinements and the Bucket Brigade technique were incorporated into a new VECTRAST program that extends a system's range of image processing and makes vector-to-raster conversions with a significant reduction in all system resource use with no loss of functionality.

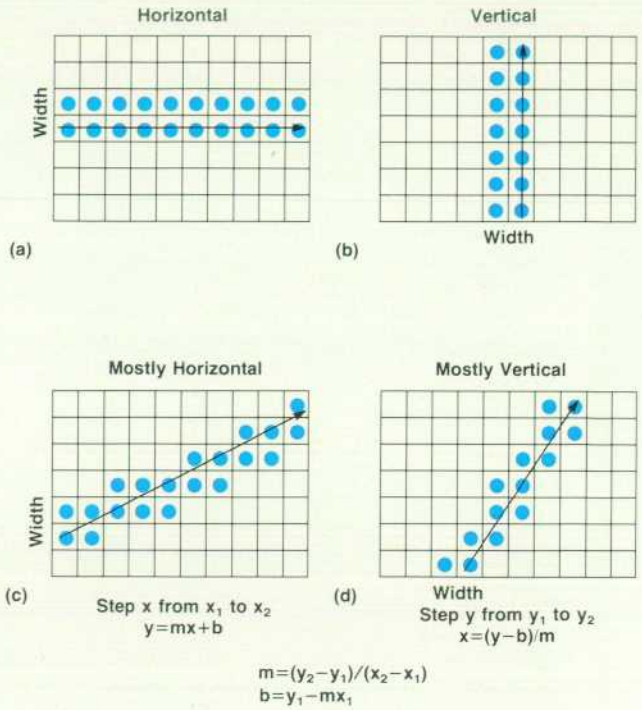


Fig. 7. *Vector width calculations are simplified by identifying directions: a) horizontal b) vertical c) primarily horizontal d) primarily vertical.*

Acknowledgments

Major contributions to the HP 2680A Graphics package were made by Jim Stratton. The project was a coordinated effort of several software project teams. We had almost daily contact with other groups from the Information Systems lab in Cupertino and the TDP/3000 project in Pinewood, England. By far the most outside communication occurred with our colleagues in the Boise Division where the HP 2680A Laser Printer is manufactured. New firmware was developed and tested in Boise to allow graphics to be printed on the laser printer. They also wrote the initial vector-to-raster conversion program based on Bresenham's algorithm.

Reference

1. J.E. Bresenham, "Algorithm for Computer Control of a Digital Plotter," IBM Systems Journal, Vol. 4, no. 1, 1965.

© Copr. 1949-1998 Hewlett-Packard Co.

The Center for Integrated Systems

Hewlett-Packard is helping to launch a new research facility at Stanford—and a new approach to industry-university relations in the U.S.A.

by Frederick H. Gardner

THERE WAS A GROUNDBREAKING at Stanford University last May 19. Its purpose was to celebrate more than the construction of a $14 million building that will house the Center for Integrated Systems (CIS). The speakers and guests were really saluting a new willingness on the part of major U.S. industrial companies and the academic community to conduct their basic research cooperatively. William Hewlett stressed the broadest implications in his keynote speech: "CIS is a clear and distinct answer to three of the major problems that face the United States—the failure of our national programs of basic research to keep pace with the needs of our universities and industries, the need to strengthen our system of education, and the challenge to U.S. trade and technology posed by foreign countries."

What answer does CIS provide and why does Hewlett-Packard consider it so important that President John Young and Vice-President for Research John Doyle have, during the last four years, devoted considerable time and energy to helping it get established?

CIS is a new teaching and research institution on the Stanford campus, funded in large part by a group of 19 high-technology companies, each of which will have a representative working there full time. As its name implies, the Center defines integrated systems as a primary field of study—encompassing computer science, integrated circuit engineering, solid state physics, and other disciplines—and brings together investigators who have traditionally operated in separate settings—the hardware and software people on the faculty, and scientists from competing industrial companies such as HP, IBM, and Texas Instruments. CIS will train 100 Masters and 30 PhD candidates a year, conduct research on every aspect of the semiconductor/computer systems relationship, and produce materials to serve the ongoing educational needs of engineers in private industry. The new building and some CIS research is being funded by the sponsoring companies. Various agencies of the United States Government will underwrite most of the equipment and research. The Commerce Department has recently revised its antitrust policies to sanction the collective basic research efforts of competing private-sector companies.

How it Began

The idea for such a center was being discussed as early as 1977 by a number of professors at the Stanford School of Engineering, notably John Linvill, James Meindl and James Gibbons. "We perceived," Linvill explains, "that integrated circuit design was by its nature interdisciplinary. You can't design custom integrated circuits without knowing a lot about computer systems—or having a co-worker who does. We also perceived that, given a diminution of Federal support for research, we would need a strong collaboration with industry to build a state-of-the-art facility for the design and fabrication of VLSI chips." A plan was developed whereby industrial companies, in return for sponsoring the center, would be offered "facilitated access" to its research program.

Next step for the initiative-taking professors was a meeting with William Kays, Dean of the School of Engineering. Linvill *et al* stressed that to stay at the forefront in integrated circuit research, Stanford would need a quick turnaround laboratory, a facility that would enable computer scientists to put their ideas directly into silicon and test them. Kays agreed and urged them to write a formal proposal to submit to the Adminstration.

President Donald Kennedy of Stanford (who was Provost at the time) foresaw certain complications. "Clearly VLSI was going to require a new facility," he reflects. "But since it would be too costly to fund in the mode in which we usually support university research, and since there hasn't been a line in a U.S. Government budget for that kind of research facility since 1968, the money almost had to come from private sources. And that would force us to work out, in very precise terms, the boundary line between philanthropy and corporate support of research. How would the hard questions of 'intellectual property rights' be handled? Would there be pressure on our faculty to do more applied and less basic research? Would the sponsors try to restrict access to the research for proprietary reasons? Could they accept the openness that characterizes the academic research enterprise? How much special access could we offer the sponsoring companies and still offer a fair degree of access to everyone? CIS has given us a forward look at these problems, which will inevitably arise as a consequence of consortium or single-corporation support of major programs. And it has given us a great research facility that Stanford simply could not have built by any other means."

By the fall of 1979, two groups still had to be convinced that CIS deserved their backing: the Stanford Board of Trustees and the hoped-for corporate sponsors. "It was very fortunate that we could turn to John Young," says Kennedy, who at this point asked Young to head the CIS Development

© Copr. 1949-1998 Hewlett-Packard Co.

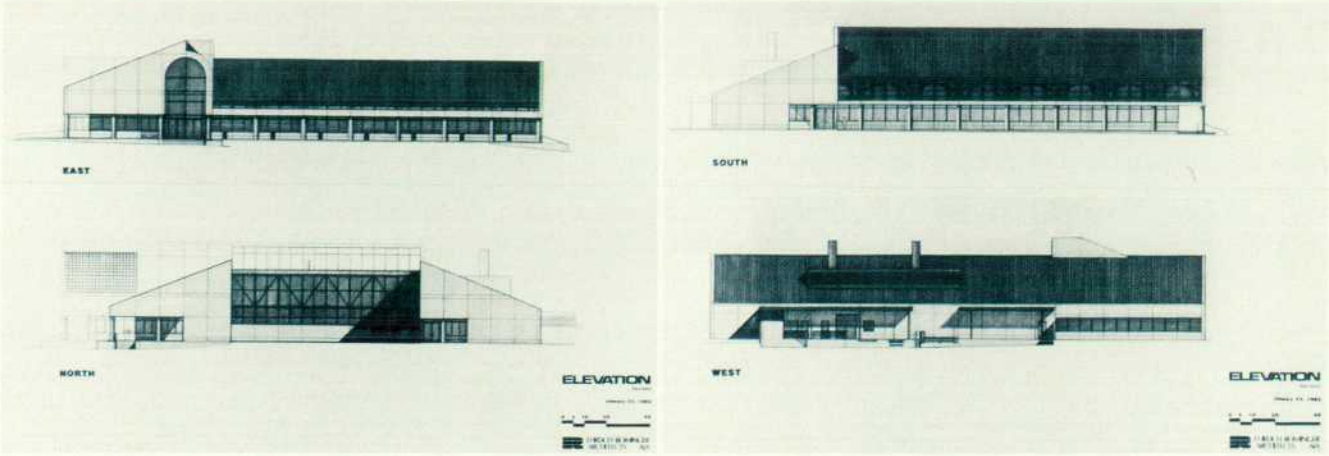


Fig. 1. *Elevation drawings of the new home of the Center for Integrated Systems at Stanford University. (Drawings provided by Erlich-Rominger, Los Altos, California, U.S.A., Architects.)*

Committee. "He had the foresight to see the significance of CIS and the thrust to convince both his corporate peers and the doubters on our Board."

To speed things along, Young, who is himself a Stanford trustee, assured the Board—before he had begun fund-raising in earnest—that corporate support would be forthcoming. On that basis they gave an official go-ahead, and in January 1980, the Center for Integrated Systems became an organizational entity with John Linvill as its director, reporting to an Executive Committee headed by James Gibbons, and to the Dean of the School of Engineering, William Kays. James Meindl, director of the Stanford Electronics Laboratory and the existing Integrated Circuits Laboratory, was subsequently appointed codirector of CIS.

John Young's View of CIS

"What appeals to me most about CIS," says John Young, "is that it represents a new era in possible relationships between the academic community and industrial companies in the pursuit of research. It's important in itself, and in the long run, it will be much more important as a model. What's at stake is the technological leadership to which we owe the very wealth and well-being of our nation.

"We've all seen the dramatic effect that the Japanese have achieved using cooperative research vehicles and planned technological development. I'm not suggesting that we emulate any Japanese models, but we should acknowledge that cooperation and planning in the basic phases of research is compatible with vigorous competition in the marketplace.

"Consider the Japanese approach to VLSI. Through their Ministry of International Trade and Industry they target microelectronics—just as they did with steel, autos, and shipbuilding—and agree to pay half or two-thirds of the basic R&D costs. To get things going they draft people from various companies—the leading researchers in all the relevant fields—and send them someplace to spend the time and money necessary to define the core elements of a VLSI research strategy. Then they all go home, taking this shared knowledge, which is a tremendous boost, and they proceed to complete vigorously in developing action plans to make products.

"CIS is a version, in our own culture, drawing on our own institutions, of that kind of cooperative-yet-competitive model. It will enable U.S. electronics companies to discuss and coordinate our research objectives and to gain leverage on our expenditures. Even companies as large as Hewlett-Packard, with half a billion dollars to spend on R&D every year—and IBM, with much more—sense that this kind of cooperative effort is going to be necessary for us to succeed in the competitive international environment between now and the end of the century. Support for education isn't altruism on our part; we recognize that without a substantial research base, we're out of business."

Young cites a number of other "strong selling points" for CIS:

- "It provides a new bricks-and-mortar facility that will undoubtedly have a substantial throughput in terms of good, basic research. Sponsoring companies, through their resident scientists, will have the earliest possible access to that research." All CIS findings will also be made available to the electronics industry through seminars and an open publication policy.
- "CIS creates a channel for spending money that didn't exist before. The U.S. Department of Defense has a lot of R&D money they intend to spend on integrated circuit development, but at one point there was no appropriate place to spend it.
- "It will give industrial companies a chance to influence and help direct fundamental research—whereas now the government gives grants, usually tied to Defense, that don't always help university researchers focus their objectives in a practical way.
- "It will train 30 PhDs and 100 MSs a year—a big increment in the total of available people. Graduate student recruitment is undoubtedly an important part of the *quid pro quo* for the sponsoring companies. Their researchers in residence are going to have a close look at who the students are and what they're doing, a chance to evaluate them and establish rapport. That has to be helpful in the recruiting process.
- "CIS will also produce training materials and videotapes

© Copr. 1949-1998 Hewlett-Packard Co.

that can be a great help to the sponsoring organizations. HP people have made extensive use of tutored videotape instruction (a technique developed by Gibbons) to take graduate courses and get advanced degrees from Stanford. With the delivery systems improving—satellites and electronic mail making two-way discussion easier—we're on the edge of some marvelous advances in off-campus education of the highest quality."

Raising the Money

The Development Committee that Young headed to raise funds for CIS included Richard DeLauer, Executive Vice President at TRW (and now Undersecretary of Defense for Research and Engineering), George Pake, Vice-President for Research at Xerox, and Robert Noyce, Vice Chairman of the Board at Intel. "We figured out what it would take to put the facility in place and came up with a budget," Young recounts. "We thought it would be best to have a small group of sponsors, 15 or 20 at the most. The question was: "How many people would it take putting up how much money to make it happen?" Young also had to make a tactical decision: how much could he ask the heads of other companies to commit on their own without triggering time-consuming internal debates? $750,000 spread out over three years was the formula he settled on.

Then came the process of putting together prospect lists, dividing the labor among members of the Development Committee, and picking up the phone. Young says that despite his list of selling points for CIS, "The project had to sell itself through the faculty, through visits, through the persuasion of an on-site coupling that says 'Okay, this thing can really work.' "

By March 1981, 10 corporations had agreed to spend $750,000 each towards the new CIS building: General Electric, Hewlett-Packard, TRW, Northrop, Xerox, Texas Instruments, Fairchild, Honeywell, IBM, and Tektronix, signing on in that order. They have since been joined by DEC, Intel, ITT, GTE, Motorola, United Technologies, Monsanto, Gould/AMI, and Philips/Signetics. The sponsors later agreed that each would contribute $100,000 annually for education, research, and adminstration.

The Question of Patent Rights

If there was a primary source of reluctance on the part of the potential sponsors, it centered around unresolved questions of patent ownership, licensing, and intellectual property rights. These issues were also of great concern to Stanford faculty members. (Recall that at the end of the '70s, patented research in the area of biotechnology had created several overnight stock market successes.) And in addition to the sponsoring companies and individual faculty members, the U.S. Government and the university adminstration had positions on these matters which the organizers of CIS had to take into account. They found themselves trying to work out a set of mutually acceptable policies in what one negotiator calls "a pressure-cooker atmosphere."

The two men who eventually took responsibility for hammering out the rules governing intellectual property rights were James Gibbons, head of the CIS Executive Committee, representing the interests of Stanford faculty members, and John Doyle, Vice-President for Research at Hewlett-Packard and Chairman of the CIS Sponsors Advisory Committee. (Before Gibbons and Doyle asserted their leadership the patent questions were being debated by lawyers. At one point there was a meeting at Stanford attended by corporate counsel from the sponsors. It ended, as one might expect, with the major issues unresolved.)

Gibbons observes: "It was a difficult problem because these companies don't have a lot of cross licenses with each other, and they've grown up in an environment where the belief is that it's your own proprietary technology that gives you the competitive edge. The idea that one of your people can invent something that will also be available to your competitors creates serious problems. So the companies' first posture was 'We want exclusive rights to anything that's invented in CIS.' And they wanted exclusive proprietary rights to anything that any of their own people invented while here, saying, with some justification I suppose, that these people had been working for the company for 10 or 15 years, and all the things they had learned and the problems they'd been thinking about when they made these inventions were related to the company, and the company is paying them. Stanford's point of view was, 'You're building a building, but almost everything that goes on in that building is paid for with Federal money, and we already have $12 million worth of Federal support, which is going to grow to $20 million. You put up 1/19th of the money and want rights to everything!'

"We finally resolved it by categorizing the patents in terms of who the inventor was—whether it was a corporate visiting scholar or a faculty member or a joint effort—and on the source of funding. In effect, if a Stanford faculty member invents something solely or jointly, Stanford patent policy applies."

Stanford's policy is unusual in that it entitles a faculty member or student who comes up with a patentable idea to own the patent if the work wasn't funded by the government. (It is common for universities to own the patents on inventions developed by their faculty members working at campus facilities.) "So theoretically," Gibbons notes, "a Stanford faculty member who invents something without government sponsorship could grant a license to somebody who wasn't a CIS sponsor—another company, say, even one in another country—for an invention that was developed at CIS." This was the worst-case scenario that haunted corporate counsel for the sponsoring companies.

On all government-funded research, however, Stanford has a legal obligation to make sure that any resulting patents are brought into the stream of commerce as rapidly as possible. This in effect means that any firm capable of commercially developing a patent has a right to bid on it and cannot be excluded. "Realistically," says Gibbons, summing up the situation, "since 98% of the work done here is under government sponsorship, and we're obligated to get it into the stream of commerce as rapidly as possible, all the sponsoring companies will have a chance to evaluate every patent and bid if they want a license to develop it. Nobody is going to get excluded. The sponsors finally realized that, given Stanford's patent policy and the law of the land governing grants for research, we hardly had any room to bargain."

© Copr. 1949-1998 Hewlett-Packard Co.

An Experiment in Mutual Interest

The negotiations may have been thorny, but they have apparently gone a long way towards instilling competing companies with a sense of their mutual interest. In addition to the agreement the CIS sponsors have worked out with Stanford, they have come up with a cross-licensing plan whereby they agree to share any inventions developed at CIS. "These aren't agreements that you could get very excited about," says John Doyle wryly, "except that they happen to make the whole thing possible. The final arrangement is that the faculty get most of what they want, but there will be a patent advisory council that tries to prevent an individual faculty member from doing something absurd, like licensing something exclusively to a foreign company. I really don't believe that is likely to happen, so we're not taking a large risk. In any case, because of the basic nature of the research at CIS, it's unlikely that a highly commercial mindbending application will come out of it directly. Whenever the research reaches a point where commercial application is foreseen, the ideas will be pursued back here" (at the sponsors' research labs).

"We've all had to cut them and shape them a little bit," says Donald Kennedy, "but I think we've come through with a workable set of policies. There may be respects in which they turn out in operation to be defective, and we'll have to fix them, but this is an experiment, after all."

Doyle uses similar terms: "CIS is an experiment—as much a social experiment as a scientific one, actually. What I'd like to see come out of it is a demonstration of effective industry-university cooperation that can break down some of the inefficiencies of research—the unnecessary duplication and secrecy— and that can bring together people from electrical engineering and integrated circuits and computer science into a center, to work together under optimal conditions and really get some new technology into the stream of commerce faster, to the benefit of the nation."

Speaking for the Government—and as a businessman in on the founding of CIS—Richard LeLauer said recently: "In industry we make a forecast, look at the market, figure out how much the investment is going to be, discount the cash flow, and if it doesn't match interest rates or what you can do with something else, forget it. That's terrific for investors but not very good for getting long-range risky projects into being. That's the reason we're so enthusiastic about the Center for Integrated Systems. It's the first time we really have a partnership."

Early Developments

Primary responsibility for overseeing the development of the new CIS facility has gone to codirector James Meindl, who in recent years has built Stanford's IC lab into a first-rate facility (it cannot be upgraded, however, to meet the requirement of VLSI research). Meindl and his coworkers are devising, testing, and building equipment for the new CIS lab in part under an $8 million, 3-year grant from the U.S. Defense Advanced Research Projects Agency (ARPA), to "Establish a facility for fabrication of large-scale integrated circuits with a turnaround interval of less than 10 working days between a user-generated mask design program and completion of functionally tested and packaged

CIS PATENT POLICY
Disposition of Rights

Inventor	Unsponsored	CIS Annual Gift	U.S. Government Sponsorship	Industrial Sponsorship
Stanford Faculty, Staff	Stanford patent policy existing at the time of the invention. Currently individual inventors retain rights.	Public domain (no rights shall be asserted)	Stanford patent policy	Negotiated with research contract
CIS Visiting Scholar			1 or 2 below	
Joint			Stanford policy for staff and 1 below for visiting scholars	
Others			Stanford patent policy	

1. Stanford takes title and inventor receives nonexclusive, fully paid license including right to sublicense.
2. Upon request of inventors, Stanford will authorize a petition by inventor's employers to U.S. Government for greater rights than 1.

© Copr. 1949-1998 Hewlett-Packard Co.

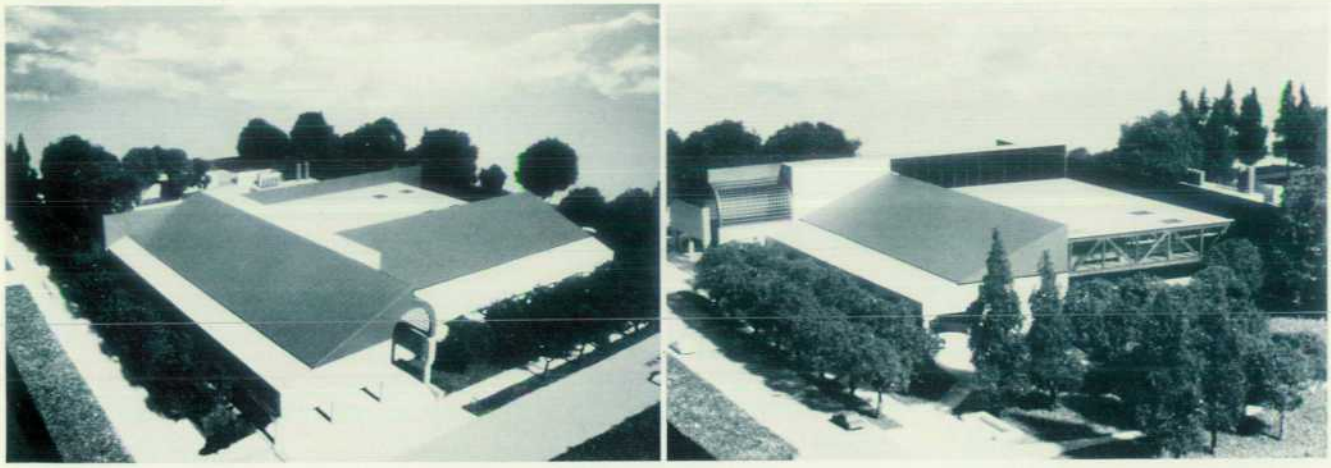


Fig. 2. *Photographs of the model of the new CIS building at Stanford. (Photos provided by Erlich-Rominger, Los Altos, California, U.S.A., Architects.)*

prototypes for evaluation by the user."

Over $1.5 million worth of equipment for the new lab has been donated by some 24 manufacturers (as distinct from the main CIS sponsors) who wish to advance the research cause and to place their hardware in a prestigious showcase.

Meindl's fast turnaround lab will occupy 10,500 square feet of the CIS building's main floor. It will provide all the necessary facilities for IC fabrication, from an electron-beam pattern generator to packaging and test equipment. Ultrasophisticated air-handling and waste-processing equipment will be installed above and below. Surrounding the IC lab will be general labs, conference rooms, and 65 offices for faculty, students, and technical representatives from the sponsoring companies. Much of the work now done in five different Stanford buildings will be headquartered at the CIS building, which is due for completion in early 1985.

The underlying premise of CIS is that physical proximity to the lab and to one another will enable researchers to do better work than they would in separate settings. Here's how interactions will take place if the organizers' dreams (as expressed in their brochure) are fully realized: "Scientists working in solid-state physics will investigate the fundamental principles of integrated circuits and pass their results to the integrated circuit engineers. These engineers will use that knowledge to design new devices and fabrication techniques, and give them to the application engineers. The applications people will define new systems and integrate the chips into complete functional systems. While the physicists and engineers are building and using chips, the computer and information scientists are developing the tools to design and test the integrated circuits. Each participant of CIS is expected to generate applications, which will drive the development of new systems, circuits, devices, or processes."

Faculty Participation

Recruitment of CIS affiliated faculty members has been chiefly the work of John Linvill. (Formal affiliation confers various advantages and responsibilities on a faculty member; above all, it indicates a willingness to conduct research in concert with scientists from sponsoring companies.)

When CIS was in its planning stages, Linvill and Dean William Kays had to reassure some of their colleagues at the School of Engineering—particularly those in materials science, radio science, and high-energy physics—that the new institution wouldn't overshadow or detract from their work. Kays recalls, "To those who worried that they weren't involved, our response was 'Get yourself involved, everything happens in a voluntary way around here.' Soon, Robert Sinclair began working with Meindl on an IC aspect of electron microscopy. Another original skeptic, Thomas Kailath, started to change his opinion when two people from the Information Systems Lab started doing outstanding work in computer design. There was some concern among the radio science people, but it now appears that whole field is going in the direction of computer implementation. In general, those who were originally suspicious of CIS are more supportive now, and they'll get very supportive when some project of theirs makes use of that lab."

Some 67 Stanford professors have joined the CIS affiliated faculty to date. Most are from electrical engineering and computer science; a few are from further afield: materials science, applied physics, statistics, radiology. Linvill estimates that 67 represents two-thirds of the faculty members with relevant research interests. He considers it a high level of participation, and expects it to keep increasing as the importance of integrated circuits becomes more pervasive.

Role of the Sponsors' Representatives

CIS research will be carried out in much the same way that scientific research is generally conducted at Stanford: by small teams that include faculty, graduate students, and technicians, with a faculty member in the role of principal investigator. It remains to be seen how researchers from the 19 sponsoring companies will fit into this scheme of things. Two have already arrived at Stanford: Bruce Delagi of DEC and Roy Russo of IBM. Delagi, who plans a two-year stay, is studying artificial intelligence (which DEC places

© Copr. 1949-1998 Hewlett-Packard Co.

considerable R&D emphasis on) in close association with Edward Feigenbaum. Russo, who has been involved with the IBM electronic design system, is resident in the Computer Systems Laboratory and working with John Hennessy. He has invited a number of IBM people to give talks to graduate students—exemplifying one of the payoffs that Stanford thinks the presence of scientists from industry will bring.

"Most of the sponsors will send people who know exactly what they want to do," John Linvill predicts. "In addition to their own contacts at Stanford, they receive a book with a list of projects being conducted by CIS faculty. Typically, they will contact me, because I'm in charge of industrial programs, and then I'll do a marriage-brokering operation. In the case of Roy Russo, he spent the first two months in the empty office next to mine because he wasn't sure with whom he wanted to work. That was fine. He attended seminars and talked to people and then one day he said 'I'm very interested in what John Hennesy is doing, and they have some space for me up there.' So off he went."

Hewlett-Packard plans to send a technical representative to CIS in 1984. "There are lots of candidates," according to John Doyle. "One criterion will be that they must want to do a particular piece of research that can be executed largely as an individual or in a small-team setting, and can best be done in an academic environment." Doyle says he won't place much weight on whether or not an individual would make a good recruiter: "Recruiting isn't our primary purpose; and in any event, it's best done indirectly, by winning the respect of the people you're working with. Since there will be Stanford resident faculty and visiting faculty and scientists from other companies, our representative has to be capable, first and foremost, of being an effective member of a community of scholars."

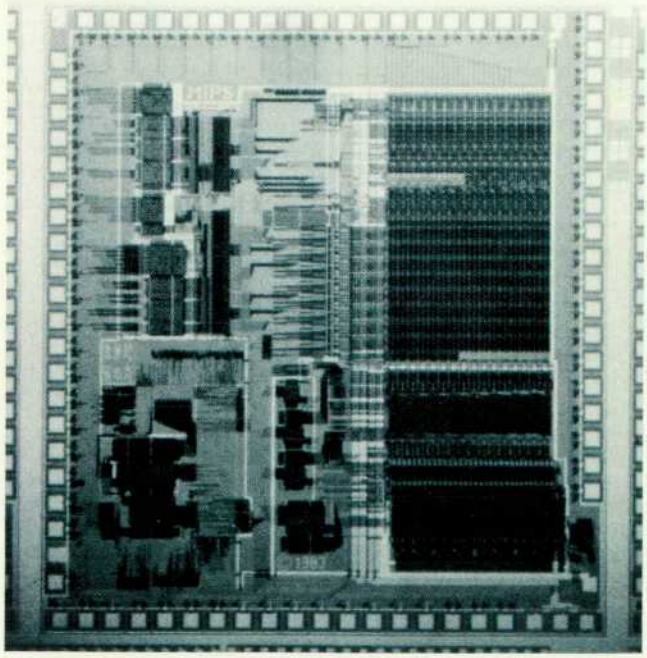


Fig. 3. *The product of one CIS project is this high-performance, streamlined-instruction-set, 32-bit microprocessor. It contains 24,383 transistors on a 5.4×6.1-mm die and executes two million instructions per second. (Photo provided by Professor J.L. Hennessy, Center for Integrated Systems, Stanford University.)*

George Pake says that Xerox is particularly looking forward to the intellectual stimulus provided by contact with graduate students. "Graduate students, because of their youth and energy, and because they haven't accumulated all kinds of external obligations and responsibilities, can focus on and be totally consumed by their scientific and technical interests. This provides an enormous stimulus. The blend of the experience and perspective of the established researcher with the drive, enthusiasm, and energy of the graduate student is enormously powerful. Ultimately, these students will be looking for employment. We'll certainly know them better and have a clear idea of whether a particular student would fit in well with our research program. But recruitment isn't the main reason we're in CIS. We're in it because we think we can get a better chance to understand new technologies as they're first being conceived, and for the interaction it will provide the people here at our own IC research lab." Pake, like Doyle of HP, expects to send a researcher with a specific project in mind.

What happens if 19 scientists arrive at CIS intending to pursue 19 different research subjects? "The same kind of process will take place that takes place now at the university or in any research environment," Pake responds. "Since you can't do everything, you have to reach a consesus about which should take priority, based on the scientific merits."

Too Much Industry Influence?

A related problem has been anticipated by critics of CIS: could the suggestion and funding of research projects by industry scientists distort the nature of what gets studied at Stanford, promoting commercially relevant investigations and subtly undermining academic freedom in the process?

James Gibbons feels that the chance of excessive industry influence on the direction of research has been overestimated. "Faculty members will still function as principal investigators," Gibbons points out, "and most of the time the initiative in selecting topics for study will be on the faculty side. In some cases we may want to abstract from specific problems that have arisen in industry research and approach them from the perspective of university research labs, asking basic questions that lend themselves to PhD thesis topics. For example, researchers may come here interested in silicides (metal-silicon contacts for ICs). TI and Bell Labs and IBM and Motorola are all working on different silicide combinations. But they know they can't ask me or Meindl to test their particular one without respect to the others. That's not the way we work in our labs. Our interest is in the fundamental mechanism, the formation of silicides. Representatives from any of those companies, or all of them, are welcome to work with us for as long as they choose. Then they can go back to their own labs and apply whatever new understanding we've developed to their own proprietary technology. And we'll keep going, trying to be imaginative and speculative, looking into new processes that may or may not be interesting, finally, to

© Copr. 1949-1998 Hewlett-Packard Co.

CIS Research Topics

Although its building won't be completed until early 1985, the Center for Integrated Systems is already coordinating a $12-million-a-year basic research program at Stanford, funded largely by the Federal government. Here's a sampler of recent and current topics investigated by CIS affiliated faculty in various existing labs:

Computers

- Research on streamlined-instruction-set microprocessor (featuring partnership of computer and IC engineers)
- Research in VLSI systems
- Knowledge-based VLSI project
- Image understanding
- Intelligent task automation
- Network graphics
- Partitionable computer systems
- Analysis and verification of high-order language programs
- Study of very high-speed integrated circuits (VHSIC-Phase 3)
- Real-time communications systems: design, analysis, and implementation
- Structured design methodology for VLSI systems
- Logical methods for program analysis
- Ultra-concurrent computer systems
- Silicon compilation
- Data base theory
- Computer languages for VLSI fabrication

Information Systems

- Multiple user channels and information theory
- Computational complexity, efficiency, and accountability in large-scale teleprocessing systems
- Multiplexed holographic reconstruction methods for 3D structures
- Information theory and data compression
- Signal processing and compression
- Statistical data processing, system modeling, and reliability
- Real-time statistical data processing
- Algorithms for locating and identifying multiple sources by a distributed sensor network
- Fast algorithms for improved speech coding and recognition
- Dual-energy digital subtraction radiography for noninvasive arteriography

Integrated Circuits

- Computer modeling of complete IC fabrication process
- Integrated electromechanical and optical sensor arrays for Optacon II (a reading aid for the blind)
- BME Center for Integrated Electronics in Medicine (to produce implantable telemetry systems for biomedical research)
- Computer-aided design of IC fabrication processes for VLSI devices
- Submicron device physics and technology
- Development of multichannel electrodes for an auditory prosthesis
- Study of transdermal electronics for an auditory prosthesis
- Multilevel-metal interconnection technology
- Biomedical silicon sensors
- Fast turnaround laboratory for VLSI

Solid State

- Ion implantation and laser annealing in semiconductors and related materials
- Laser and electron beam processing of semiconductors
- Characterization of high-speed semiconductor device materials using advanced analytical techniques
- Ion implantation and laser processing of 3-5 compound conductors
- Defects at electrode-oxide and electrode-silicon interfaces in submicron device structures
- Microstructure fabrication using electron beams of conventional and very low energies
- Advanced concepts in VLSI metallization
- Advanced packaging concepts for VLSI
- Studies of surfaces and interfaces of 3-5 compounds and Si:silicides
- Silicon photocells in thermophotovoltaic energy conversion
- Investigation of metallic impurities introduced into SiO_2 and Si by various candidate VLSI metallization systems
- Modeling of emitters
- Structural and bonding studies of practical semiconductor layers

Space Telecommunications and Radioscience

- Establishment of a Center for Aeronautics and Space Information Sciences at Stanford University (Funded by the U.S. National Aeronautics and Space Administration. The focus will be on applying VLSI techniques to develop new hardware and firmware for space instrumentation and command and control.)
- Communication satellite planning center

Material Science and Engineering

- Atomic-level physics modeling of the thermal oxidation process
- Fabrication and properties of multilayer structures
- Computer simulation of surface and film processes
- Photoelectronic properties of 2-4 heterojunctions
- Photoelectronic properties of zinc phosphide crystals, films, and heterojunctions
- Photovoltaic heterodiodes based on indium phosphide
- Preparation and properties of CdTe evaporated films compared with single-crystal CdTe

Ginzton Laboratory

- Superconducting thin films, composites, and junctions
- Acoustical scanning of optical images
- Research on nondestructive evaluation
- Evaluation of machining damage in brittle materials
- Optical and acoustic wave research
- High-frequency transducers
- Research on acoustic microscopy with superior resolution
- Study of properties of material by channeling radiation
- Surface acoustic wave MOSFET signal processor

industry."

John Young thinks that industrial influence on university research should be welcomed, not feared: "Engineers tend to be pragmatic, reasonable people. I think there's going to be a synchronizing of interests through CIS, not a conflict. The sponsors advisory committee will meet and discuss in large terms what kinds of things we're all really thinking about—photolithography, CMOS processing, silicon compilers, and other fundamental areas of current interest. That should be enlightening to faculty members. It certainly won't affect their academic freedom in a negative way.

"There's a false image of academic freedom," Young adds pointedly, "—professors just pursuing whatever they want to. In actuality, many times they do what they think they can get money for. CIS may prove that it's a lot more efficient for a university researcher to have a common view

© Copr. 1949-1998 Hewlett-Packard Co.

of what's important with some working company researchers who are doing this business day in, day out, rather than government or other people who only know about handing out money. We'll see."

Some Hard Questions

Donald Kennedy, President of Stanford, has ultimately had to deal with all the hard questions raised in connection with the founding of CIS. Here are some of the major ones, along with Kennedy's answers:

- Could industrial sponsorship distort the nature of academic research? "Faculty will work on the things they want to work on. If the problems brought in by industry are good problems, the faculty will be receptive."
- Will industrial representatives exert undue pressure on graduate students to join their companies? "The companies are already recruiting pretty intensely. The graduate students at CIS will probably find it more comfortable because it will be less frantic and they'll get a better, longer look at prospective employers."
- Might there be pressure to conduct projects in an atmosphere of secrecy? "All research carried on at CIS will come under the university's research policy, which guarantees open publication and countenances only brief delays to permit certain types of review. We didn't make any exceptions for CIS."
- Will CIS sponsors gain an unfair advantage over less affluent companies in the microelectronics field? "All companies, small and big, have access to the graduate students and access to the research through meetings on campus where we describe what's being done, through publication, through their network of contacts and connections. And if a CIS faculty member wants to invite a postdoctoral fellow from a nonparticipating company to join his research team, he can do it. We had to fight that battle hard. Some of the sponsors didn't like the idea of 'free riders,' but we persuaded them that Stanford couldn't go into an agreement that would shut others out. 'We can help you, but we can't do it by hurting somebody else.' That's the principle we've operated on in establishing CIS."

Similar Centers

Kennedy calls the founding of CIS "a watershed" and the difficulties he has encountered "prototype problems." He expects analogous centers to be created at Stanford and other universities, in various disciplines, as private-sector companies increase their support for academic research and training. In the microelectronics field somewhat similar institutional models are now being developed at MIT, the University of Minnesota, the University of California at Berkeley, and elsewhere. The University of Texas in Austin and Texas A&M University in College Station are upgrading their integrated circuits research programs and facilities in conjunction with a consortium of 12 U.S. firms known as the Microelectronics and Computer Technology Corporation. In North Carolina researchers from private industry and three universities are sharing a new facility, the Microelectronics Center at Research Triangle Park near Durham, supported largely by a $24.4 million appropriation from the state legislature. The Semiconductor Industry Association, made up of 20 major computer and integrated circuit companies, including Hewlett-Packard, has formed the Semiconductor Research Cooperative and given it a substantial budget for the sponsorship of university-based research projects in IC-related fields.

All these undertakings are viewed with enthusiasm by John Young, and Hewlett-Packard is directly involved in two of them—the Semiconductor Research Cooperative and the Microsystems Program at MIT. "We're putting more of our money into CIS," Young explains, "because physical proximity is a factor in how effectively you can relate to an institution. We've been able to play a leadership role in CIS. Another factor, of course, is the close relationship Hewlett-Packard has had with Stanford throughout our 44 years. There's a long history of successful contacts and contracts, progressive escalations of closeness. CIS is the latest. Which is one of the reasons I'm confident it's going to work."

© Copr. 1949-1998 Hewlett-Packard Co.

Authors
November 1983

3

Yvonne Temple

Yvonne Temple received a BS degree in computer science from the University of Minnesota Technical Institute in 1981. She joined HP after graduation and has been involved in business graphics software development for the HP 3000 Computer. A native of St. Paul, Minnesota, she is married and now lives in San Jose, California. Her hobbies are reading and sewing.

5

Richard J. Simms, Jr.

Rich Simms attended the University of Michigan where he earned a BS degree in computer science and an MS in computer engineering. A member of the Association for Computing Machinery(ACM), he has taught classes in computer science and authored papers on stratigraphic analysis and visual information display systems. With HP since 1978, Rich is a project manager for HP 3000 graphics software. He is single, lives in Santa Cruz, California, and enjoys running and motorcycling.

Janet Elich Morris

Janet Morris received a BS degree in computer science from California Polytechnic State University at San Luis Obispo in 1982. She participated in HP's SEED student employment program and joined HP as a software development engineer after graduation. Recently married, she is a member of ACM and is organizing a Cal Poly computer science alumni association. Born in Duarte, California, she now lives in Santa Clara and is working on an MS in computer science.

10

Robert W. Dea

Bob Dea was involved in the design of DSG/3000 and HPEASYCHART. He is currently at HP Laboratories working on strategies for the development of software tools, a subject on which he has published an article. He graduated from the University of California at Berkeley in 1973 with a BS degree in EE/CS and came to HP in 1979 with experience as a software engineer. He is married, lives in Fremont, California, and is interested in microcomputers and martial arts.

Martha Seaver

A native of Los Angeles, California, Martha Seaver earned a BS degree in physical science from Stanford University in 1976 and an MS in computer science from the University of North Carolina at Chapel Hill in 1981. She was a systems engineer before joining HP in 1981 as a software development engineer. She is engaged, lives in Cupertino, California, makes stained glass windows, and participates in long-distance swimming. She is also a tap dancer with Broadway aspirations.

13

Janet Swift

Janet Swift participated in the design of HPDRAW and is currently a project leader for new HP 3000 software products. The author of a paper published in the National Computer Graphics Association Proceedings, she graduated from the University of Oregon in 1977 with a BS degree in mathematics and computer science. Janet is a member of the San Jose Junior Women's Club. She is engaged to another HP employee, and will be moving to the Boise Division soon. She enjoys running.

Chayaboon Purnaveja

Audi Purnaveja was born in Bangkok, Thailand and attended Rensselaer Polytechnic Institute, where he received an MS degree in computer science in 1978. He later attended the University of Michigan for training in computer graphics, completing that program in 1981. He is a software development engineer at HP and is a member of the ACM, SIGGRAPH, IEEE, and NCGA. He lives in Sunnyvale, California and his outside interests are ice skating, volleyball, skiing, and ballet.

17

Gerald T. Wade

Gerry Wade received a BS degree in physics from New Mexico State University in 1971. He was a physicist for the U.S. Atmospheric Science Laboratory before joining HP in 1974. Gerry contributed the new vector-to-raster program for the 2680A Graphics Package, including the Bucket Brigade and algorithm refinements. He has also worked as a product specialist and as a systems engineer for the HP 3000 Computer. He has coauthored three published papers on laser technology.

William J. Toms

Bill Toms was a computer specialist for the U.S. Army Corps of Engineers in Frankfurt, West Germany before coming to HP as a development engineer in 1981. He holds a BS degree in mathematics awarded by the University of Washington, where he has also done graduate work in computer science. Bill lives in Santa Clara, California, and enjoys international travel, skiing, scuba diving, and collecting Steiff bears.

James C. Bratnober

A graduate of Stanford University with a BS degree in mathematical sciences awarded in 1979, Jim Bratnober came to HP in 1981 as a development engineer. Born in Cottonwood, Arizona, he is married, has two children, and now lives in Boise, Idaho, where he enjoys skiing and French intensive gardening.

Tamara C. Baker

Tammy Baker graduated from Cornell University in 1981 with a BA degree in computer science. She joined HP shortly thereafter as a development engineer and contributed to the design of the 2680A Graphics Package. Born in Jacksonville, North Carolina, she now lives in Mountain View, California, and enjoys boardsailing, cycling, and skiing.

23

Frederick H. Gardner

Fred Gardner is director of Redwood Publication Specialists, a technical writing service based in Glen Ellen, California. A former member of the *Scientific American* Magazine Board of Editors, he has written professionally in almost every medium, including fiction, journalism, screenplays, children's books, songs, and commercial literature. He is married and has four children. His leisure time (Fred scoffs at the term) is devoted to putting part of a rocky Sonoma County mountainside into agricultural production.

© Copr. 1949-1998 Hewlett-Packard Co.

Hewlett-Packard Journal Book Now Available

The book we told you about in our July 1983 issue, *Inventions of Opportunity: Matching Technology with Market Needs*, is now available through HP's Computer Supplies Operation.

The 364-page book covers 33 years of engineering at HP through articles reprinted from the Hewlett-Packard Journal.

Highlighting the book is commentary by William R. Hewlett, company cofounder and vice chairman of its board of directors. In introductory remarks and chapter prefaces, Hewlett sets the scene for the market-driven technology developments detailed in the book's 31 reprinted articles.

Selected by a panel of 20 senior HP engineers, the featured articles were chosen for their description of products that contributed to the state of the art in electronics technology at the time of development.

In the early days, when the company had a modest program for research and development, HP engineers followed what they called "engineering of opportunity." This approach sought to match a market need with a technology that would produce an appropriate and commercially successful product.

Hewlett provides glimpses into the growing organization that continued to produce technological innovations through more than two decades, leading to the formation in 1966 of the centralized Hewlett-Packard Laboratories.

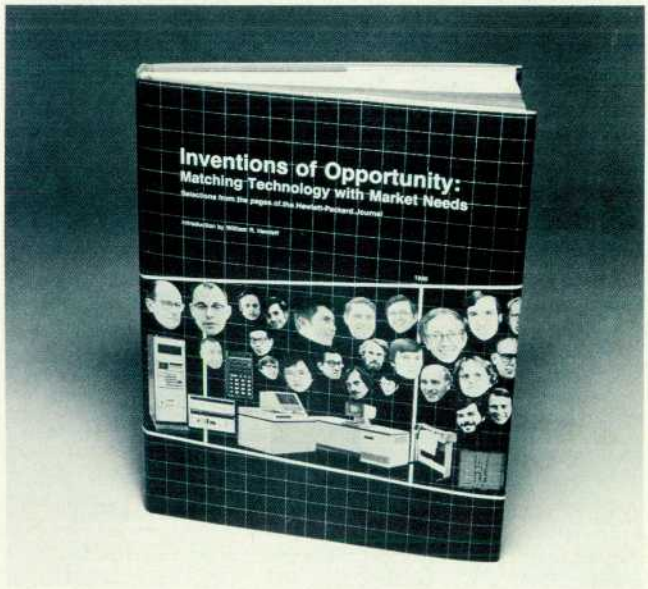
Among the articles are those on the first high-speed frequency counter, the HP-35 Calculator that replaced the slide rule in engineers' pockets, the beginnings of computer-controlled instrumentation systems, and many others. The introduction comments on each article, providing insights into the dynamics of innovation and showing how a need, a technology, and creative people come together to produce a successful invention.

Cloth-bound, with dust cover, *Inventions of Opportunity: Matching Technology with Market Needs* is available exclusively from Hewlett-Packard. The HP part number is 92233B.

The easiest way to order *Inventions of Opportunity* is to call HP's Computer Supplies Operation at one of the special telephone numbers listed here.

Location	Telephone Number
United States	800-538-8787
California	408-738-4133
United Kingdom	0734-792868
	0734-792959
France	(6)928 32 64
Belgium/Luxembourg	(02) 762 32 00
Switzerland	(057) 31 22 54
	or 31 22 59
Canada	(416) 678 94 30
West Germany	07031-142829
	07031-223133
The Netherlands	020-470639
South Africa	802-5111
	53-7954
	28-4178

Of course, orders may also be placed in the regular way through local HP sales offices, whether or not your area is served by one of these special telephone lines.



Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, California 94304

Bulk Rate
U.S. Postage
Paid
Hewlett-Packard
Company

HEWLETT·PACKARD JOURNAL

NOVEMBER 1983 Volume 34 • Number 11

Technical Information from the Laboratories of
Hewlett-Packard Company
Hewlett-Packard Company, 3000 Hanover Street
Palo Alto, California 94304 U.S.A.
Hewlett-Packard Central Mailing Department
Van Heuven Goedhartlaan 121
1181 KK Amstelveen, The Netherlands
Yokogawa-Hewlett-Packard Ltd., Suginami-Ku Tokyo 168 Japan
Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive, Mississauga, Ontario L4V 1M8 Canada

02000207 07&&&BLAC&CA00
MR C A BLACKBURN
JOHN HOPKINS UNIV
APPLIED PHYSICS LAB
JOHNS HOPKINS RD
LAUREL MD 20707

CHANGE OF ADDRESS: To change your address or delete your name from our mailing list please send us your old address label. Send changes to Hewlett-Packard Journal, 3000 Hanover Street, Palo Alto, California 94304 U.S.A. Allow 60 days.

5953-8517

© Copr. 1949-1998 Hewlett-Packard Co.