

MARCH 1977

# HEWLETT-PACKARD JOURNAL

0:30: 5:150

PT SZ PRGRM,T,PRIOR\*DRMT\*SCHD\*I/O \*WAIT\*MEMY\*DISC\*OPER \* NEXT TIME

0 \*\* SPOUT\*1\*00011 \*\*\*\*\* 3, CL 030

2 2 WHZAT\*2\*00030 \*\*\*\*\* 1

10 7 GASP \*3\*00080 \*\*\*\*\* 2, 12(2(000000000)) \*

10 14 RT3GN\*6\*00099 \*\*\*\*\* 6

4 5 TVST \*0\*\*\*\*\* 0 \*\*\*\*\*16:30: 5:880

9 6 FMG12\*\*\*\*\* 3, COMPL

11 6 FMG63\*\*\*\*\* 3, RQL31

5 6 FMG31\*\*\*\*\* 3, ED131

8 6 FMG16\*90\*\*\*\*\* 3, BL,EQT16

1 2 R\$PH\*\*\*\*\* CL 032

11 12 COM\*\*\*\*\* B

6 7 EDI\*\*\*\*\* 00101) \*

12 12 ASM\*\*\*\*\*

5 2 RQL\*\*\*\*\*

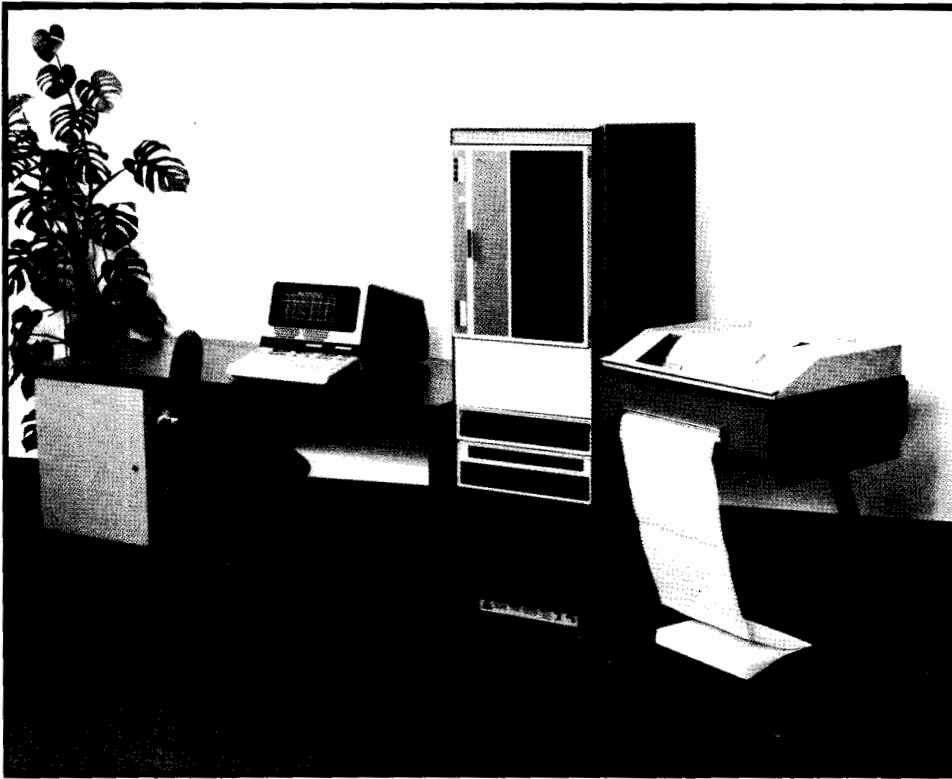
029 LKPRG\*GL DBI

DOW

DOW

16:





**Fig. 2.** HP 1000 Model 80 is a larger system suitable for data base management, operations management, and service as a central computer in a distributed system. Model 81 is the same system in two upright cabinets.

system disc storage device, an operator's console and a system cabinet, either desk or upright. Each model is designed to provide the correct performance and capability level for expected application needs, the goal being a better match between computing power and applications.

The HP 1000 product line was conceived as a family of systems that would eventually extend from low-cost memory-based configurations up to high-capacity disc-based configurations. Models 30/31 and 80/81 are the first of this family. A model 30/31 can be easily extended to a model 80/81 by adding 64K bytes of memory and dynamic mapping system hardware, upgrading to the RTE-III operating system, and adding the IMAGE data base package, a magnetic tape unit, and a line printer. No extensions to the processor and console are required, and user program compatibility is assured within the RTE software.

The goal of product simplicity results in well-defined functional systems with options defined only where factory modifications are required. The user benefit is a better understanding of the basic product and the extended capabilities provided by factory options. Accessories, interfaces, and peripherals are specified separately and supplied with the system as a coordinated shipment. This is in marked contrast to earlier (9600) systems, for which there were hundreds of options. The one advantage of options on earlier systems was the control of compatible products that could be integrated with the system. This advantage

is retained in HP 1000 Systems, not through options, but by a compatibility matrix that specifies exactly which products have been tested and determined to be operable within a certain system configuration.

The compatibility matrix is part of the manufacturing specifications documentation package and is managed by engineering change control procedures. The matrix is the official source for the HP 1000 Configuration Guide, a document that field engineers and customers use to construct a complete system to meet specific needs. Only the products listed in the Configuration Guide are supported by HP as part of HP 1000 Systems. Before a new accessory, peripheral, or software product can be added to the set of 1000-compatible products, it must undergo extensive testing by the engineering laboratory and then be certified as compatible by the quality assurance and systems integration groups. This degree of configuration control is aimed at increasing customer satisfaction by providing a clear statement of what works with what and in which combinations, in the initial system configuration as well as in field upgrades to installed systems.

#### **Multi-Media Software Distribution**

A major objective of the enhanced RTE operating system was to eliminate the dependency on paper tape as the medium for software distribution and as the required input medium for system generation. This was achieved by providing system and diagnos-

## HP 1000 Computer System Applications

Since HP's first "instrumentation computer", Model 2116A, was introduced in 1966, the majority of over 16,000 2116's, 2100's, and 21MX's installed have been for scientific and engineering computation, data acquisition and control, factory automation, process control, product and production testing, and computer-aided design.

HP 1000 Computer Systems are designed principally for dedicated applications in engineering and manufacturing. The performance and capability levels of these systems have been focused on computation, instrumentation, and operations management applications that demand high performance. These systems can best be applied by experienced end-users, OEM's (original equipment manufacturers), and system houses.

The significant performance improvements of the E-Series Computer, coupled with technical languages, microprogramming support, high-performance peripherals, and specialized microcode such as the Fast FORTRAN Processor, form a powerful combination for solving computation problems such as

- Scientific and engineering computation
- Product development testing
- Simulation and modeling
- Computer-aided design
- Statistical analysis
- Project control

Instrumentation problems can be solved either through the use of HP-IB instrument clusters, with multiple HP-IB-compatible instruments and devices connected to an HP 1000 System, or by measurement and control stations for sensor-based applications requiring medium-speed analog and

digital input and output capabilities. The stations are based on plug-in interface cards and instrumentation subsystems from the 9600 family of measurement and control systems. Typical applications are

- Machine monitoring and control
- Product quality control testing
- Work station reporting
- Continuous and batch process control
- Shop floor monitoring and control
- Automated materials handling
- Automatic testing
- Facilities monitoring

Operations management applications can best be solved with the aid of a data base management system (DBMS) so that all relevant data can be interrelated, easily updated, and available for on-line access. IMAGE/1000 and QUERY/1000 combine to meet this need. Factory data collection is facilitated by the compatible 3070A Data Entry Terminal. An HP 1000 System can also serve to control a network of HP satellite computer systems with the HP distributed systems capability. A link to a large data processing computer system can be made with the RJE/1000 package. Typical operations management applications include

- Material requirements planning (MRP)
- Capacity requirements planning
- Factory data collection
- Master scheduling
- Purchase order and work order control
- Stores control
- Production status monitoring and control
- Manufacturing and engineering data control

tic software on multiple media and by the new on-line system generator in RTE. The distribution media include paper tape, magnetic tape (800 or 1600 cpi density), disc cartridge (HP 7900A or 7905A) and mini-cartridge tape. Specification of the desired medium is made by options to the software (e.g., 020 for mini-cartridge), and by a specific product number for the diagnostic library. The diagnostic library is a collection of tests for the processor, memory, accessories, interfaces, and peripherals available with 21MX Computers. A new diagnostic configurator was developed to accommodate the diagnostic library on each medium; the configurator is a monitor or control program that manages the loading and execution of each individual diagnostic test and provides a common interactive dialogue and reporting interface to the user, typically an HP customer engineer. It also configures each diagnostic for the appropriate I/O channel and provides for timing loop control depending on the processor and memory speeds.

The standard software distribution media for HP 1000 Systems are mini-cartridges for diagnostics and supplemental software and disc cartridges for the

operating system. The standard disc provides a minimum operating system supporting a wide range of peripherals plus all the relocatable binary programs needed to distribute and regenerate a system. This disc cartridge also contains all drivers for standard peripherals as well as standard languages and utilities. A specially configured disc cartridge, prepared during the system integration operation, represents the operating system and drivers for the user's HP 1000 system, which may include various peripherals and additional software such as real-time multi-user BASIC, IMAGE, and/or the measurement and control software library. This configured disc is what the user will begin operation with after installation, while the original disc serves as a backup standard system and can be easily updated to reflect the latest software changes.

The multi-media distribution and grandfather disc for RTE have definite advantages for system flexibility and improved user satisfaction. A secondary but significant benefit of these new features is improved efficiency in the systems integration process during manufacture. This process involves integrating all

the hardware products, including racking, cabling, and running diagnostics. Loading and running diagnostic routines from mini-cartridges in the 2645A Terminal is faster and much more convenient than with paper tape. Once the hardware integration is completed, the process of generating the configured operating system according to the user's order begins. Since all drivers, device subroutines, and libraries for the standard systems and compatible peripherals are contained on the grandfather disc, the amount of additional software that must be added before generation is minimized. The on-line RTE generator is designed to run with minimal operator intervention, with most configuration parameters contained in an answer file on disc. Most new generations require only minor modifications to an existing answer file from previous generations. Of course, the faster processor and console terminal also reduce the generation and system test execution times.

HP 1000 Computer Systems represent an integration of many hardware and software products. The following articles describe the technical details of the more significant new products which contribute most to these systems and are significant contributions as individual products.

#### Acknowledgments

The development of the HP 1000 Systems and the component products required the dedication and commitment of many people in Data Systems Division.

Dick Anderson, general manager, provided the overall leadership. Contributing to the success of the system-level products were Neal Walko, Arlan Saunders, Bob Daniel, Bill Becker, Ken Fox, Dave Borton, Chris Lehner and Phil Williams. Hugo Schaerli and Mike Winters developed the new multi-media diagnostic software.



#### Lee Johnson

Lee Johnson joined HP's computer program in 1966. He was responsible for developing the basic control system (BCS) for the 2116A Computer, and then designed and managed the development of the RTE operating system. He managed the development of the IMAGE Data Base Management System, and was responsible for the system development of the HP 1000. He's now the engineering manager of the 21MX/RTE laboratory at HP's Data Systems Division. Lee was born in New Mexico and grew up in Colorado Springs, Colorado. He graduated from the University of Colorado in 1963 with a degree in mathematics and in 1970 received an MBA degree from the University of Santa Clara. He's married and has two children.

## HP 1000 Computer Systems

### Model 30/31 Features

#### CABINETS

MODEL 30: Desk-style with matching mini-rack for disc subsystem.  
MODEL 31: Single 56-inch high upright cabinet. (System console must be placed on separate stand or table.)

**CPU:** 2113A 21MX E-Series Computer with Time Base Generator, Memory Controller, Memory Protect, Dual-Channel Port Controller, Power Fail Recovery System, Loader ROMs for disc subsystem and CRT terminal, ROM diagnostics for CPU and memory.

#### MAIN MEMORY

MODEL 30: 64K bytes standard. Expandable to 304K bytes when using optional RTE-III operating system.

MODEL 31: 64K bytes standard. Expandable to 608K bytes using optional RTE-III operating system and a memory extender.

**OPERATING SYSTEM:** RTE-II Real-Time Executive is standard. RTE-III, including Dynamic Mapping System and +64K bytes of memory, is available as an option. RTE-III is required in systems using more than 64K bytes of main memory.

#### DISC SUBSYSTEMS

MODEL 30: 12962D Subsystem with 14.7M bytes of storage, expandable to 117.9M bytes using 7 add-on drives.

MODEL 31: 12960A Subsystem with 4.9M bytes of storage, expandable to 19.6M bytes using 3 add-on drives, or 12962C Subsystem with 14.7M bytes of storage, expandable to 117.9M bytes using 7 add-on drives.

**SYSTEM CONSOLE:** 2645A Display Station with Option 007 dual mini-cartridge tape transport.

### Model 80/81 Features

#### CABINETS

MODEL 80: Desk-style cabinet for CPU and system console; single 56-inch upright cabinet for disc and magnetic tape subsystems.

MODEL 81: Two 56-inch upright cabinets. (System console must be placed on

separate stand or table.)

**CPU:** 2113A 21MX-E Series Computer with Time Base Generator, Memory Controller, Memory Protect, Dual-Channel Port Controller, Power Fail Recovery System, Loader ROMs for disc subsystem and CRT terminal, ROM diagnostics for CPU and memory.

#### MAIN MEMORY

MODEL 80: 128K bytes standard. Expandable to 304K bytes.

MODEL 81: 128K bytes standard. Expandable to 608K bytes when using a memory extender.

**OPERATING SYSTEM:** RTE-III, including Dynamic Mapping System, and IMAGE/1000 Data Base Management System.

**DISC SUBSYSTEM:** 12962C Subsystem with 14.7M bytes of storage, expandable to 117.9M bytes using 7 add-on drives.

**SYSTEM CONSOLE:** 2645A Display Station with Option 007 dual mini-cartridge tape transport.

**REQUIRED PERIPHERALS:** A Line Printer Subsystem from following selection:

Model	LPM	Col
12975A	300	136
12983A	1250	132
12987A	200	132
13053A	600	136

B Magnetic Tape Subsystem from following selection:

Model	Type	bpi
12970A	NRZI	800
12972A	Phase-	1600

Encoded

**PRICES IN U.S.A.:** Minimum prices for HP 1000 Systems without options are as follows: Model 30, \$36,500; Model 31, \$31,500; Model 80, \$62,000; Model 81, \$62,000.

**MANUFACTURING DIVISION:** DATA SYSTEMS DIVISION

11000 Wolfe Road  
Cupertino, California 95014 U.S.A.

# HP 1000 Operating System Is Enhanced Real-Time Executive

*New RTE-II and RTE-III software provides for on-line system generation and switching, disc cartridge backup, disc and mini-cartridge distribution of software, new system string communication, and improved I/O error management.*

by David L. Snow and Kathleen F. Hahn

**T**HE OPERATING SYSTEM for HP 1000 Computer Systems is an enhanced version of Hewlett-Packard's disc-based, multi-user, multiprogramming real-time executive (RTE-II/III) operating system.<sup>1,2</sup> Major features include priority scheduling of concurrent programs, separation of real-time and non-real-time tasks into foreground and background partitions, and a powerful file management package (FMP). RTE provides program partition swapping, buffered output, resource locking, class input/output, and a batch entry processor featuring input and output spooling of jobs for maximum throughput. RTE-II (HP 92001B) provides, in addition to memory-resident partitions, two disc swapping partitions (one real-time and one background) using either 48K or 64K bytes of main memory. RTE-III (92060B) adds a memory management scheme that handles up to 64 real-time and background partitions and 2M bytes of main memory (hardware limited to 608K bytes in the System 1000). Combined with the distributed system central software package (HP 91700A),<sup>3</sup> RTE becomes a powerful central station controlling distributed processes at several types of satellite computer systems. HP 1000 models 80 and 81 combine the RTE-III operating system with Hewlett-Packard's IMAGE/1000 data base management package (HP 92063A), providing multi-user, multiprogramming access to a user-tailored data base.

RTE-II/III operating systems have been enhanced to include an on-line system generator, disc-to-disc and disc-to-magnetic-tape backup utilities, expanded user-to-program and program-to-program communications, restructuring of I/O management for device errors, and an enhanced batch/spool monitor. For the HP 1000, disc cartridge distribution of operating system software has been added, along with HP mini-cartridge distribution of all RTE software subsystem products (IMAGE/1000, microprogramming package, device diagnostics, distributed systems software, and others).

The hardware environment for the enhanced RTE-II/III operating systems is the HP 1000 models 30, 31,

80, and 81, using the 21MX E-Series Computer as a control processor, a 7905A or 7900A Disc Drive as a mass storage device, and a 2645A Terminal as system console. Previous RTE-II/III hardware environments that have at least 48K bytes of memory can also accommodate the enhanced operating systems.

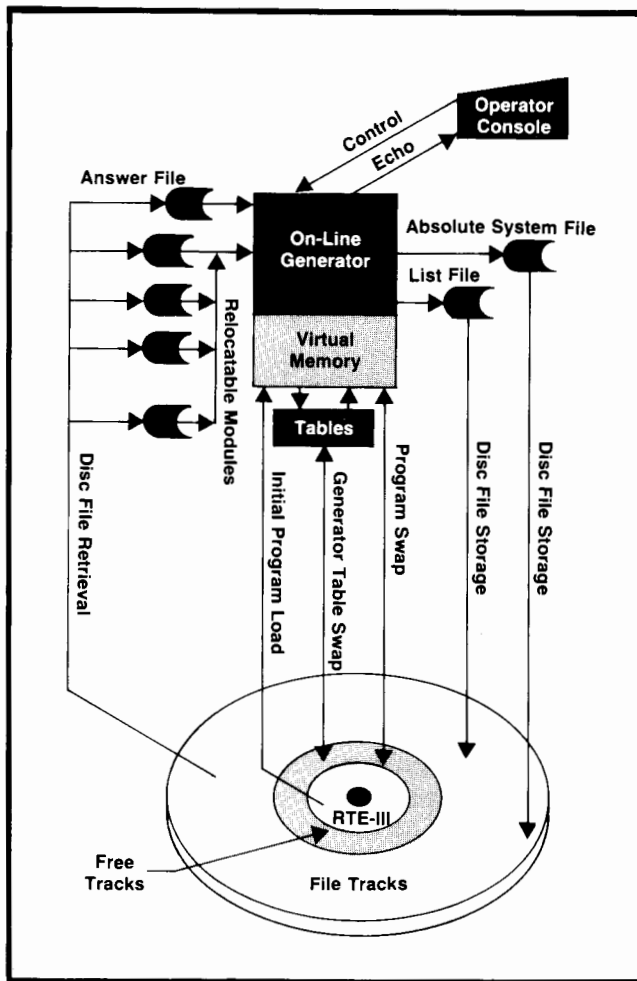
## On-Line Generator

The modularity of the RTE software makes it easy to configure a real-time operating system tailored to particular application requirements for input/output peripherals, instrumentation, program development, and user software. With the on-line generator a new configuration can be achieved under control of the present configuration, concurrent with other system activities, such as interactive access to an IMAGE/1000 data base or real-time test monitoring and process control.

The on-line generator, a background program, exists in two forms, RT2GN and RT3GN, for configuring RTE-II and RTE-III operating systems, respectively. A special utility program, SWTCH, performs the switch-over from the present operating system configuration to that of the new system, with a minimal amount of shut-down time (15-30 seconds). Of greatest impact is SWTCH's capability to preserve the file structure of the system it is replacing.

The on-line generation process uses the file management features of the batch/spool monitor (BSM) for retrieval of the generation parameters and software modules, and for storage of the absolute system code and its associated generation map (see Fig. 1). Storage of the relocatable software modules in files allows easy updates when new versions are implemented, as well as the convenience of referencing and identifying the modules by their descriptive file names during generation. When the generation parameters (a set of commands and responses) also exist in a file, this ANSWER file can be modified to create answer files for new system generations.

One of the most useful features of on-line generation is the storage of the generated system in a core-



**Fig. 1.** The on-line generator makes it possible to develop a new RTE-II/III operating system configuration under control of the present configuration, concurrently with other system activities. The entire generation process can be directed from a disc answer file. All program modules to be included in the new operating system must exist in relocatable disc files. The new system is stored in an absolute system disc file after generation.

image absolute output file, making it possible to

- Store system files indefinitely, even after the system is installed by SWITCH
- Have several versions of RTE systems and easily SWITCH between them
- Distribute a particular RTE system configuration by storing its core-image file on portable media
- Store the file on a peripheral disc subchannel where it can be accessed by other RTE systems.

#### Generation Process

The generation process can be directed from a disc answer file, from a logical unit (representing a peripheral device), interactively from the operator console, or from a combination of these. The desired answer file name or logical unit can be specified in the RU command parameters when the generator is

scheduled for execution, otherwise the default source of command input will be from the operator console. A TR command to transfer to another command input source can be entered any time the generator is waiting for a response from the current input source. The current input source is pushed down on a command input stack and the new source specified in the TR command is placed at the top. Therefore, an answer file can transfer to another answer file, to a logical input unit, or to the operator console for the next response to a generator query. A TR command with no parameters will pop the stack and the input source will revert to the previous answer file or logical unit.

When an error occurs during the generation, either an invalid response to a query or an error in the system being generated, error recovery mode is entered. In this case, the generator issues itself a TR command transferring the command input source to the logical unit of the operator console (unless it was already there). The operator is notified of the type of error condition and is given the opportunity to rectify the situation by re-entering a response, or by entering a !! command to abort the generation. After entering a correct response, the operator can issue a TR command and the generator will return to the answer file, picking up where it left off.

The generation's listed output, or generation map, can be sent to a disc file or output device. A disc file provides more permanent storage and eases the problem of duplication since a hard copy can be listed on the line printer at will. It also allows easy identification of the RTE configuration of a particular system file when planning to run SWITCH. While the generation map is being sent to the list file, it may also be echoed on the operator console. The operator can then follow the generation as it proceeds and make better decisions when error conditions occur, when memory bounds need to be set, and so on.

The generator uses a virtual memory scheme to maintain three internal tables of dynamic size. The available memory area beyond that occupied by the generator code and up to the last accessible word of background memory space is divided between these tables. When the memory space for a table becomes full, that block of memory is written onto the disc. An indexing scheme is used to determine whether or not a particular block is resident in memory. If not, a swap must be done: the current memory block is written to the disc, and the referenced block is read in to the same memory partition space.

The performance of the generator can be improved by increasing the size of the background area in which the generator is executing. The available memory space assigned to each table is thus increased, the resulting block size is larger, and the number of block swaps decreases. The user can also



order the specification of software modules to be included in the system and thereby determine the order of entries in the tables. Thus the likelihood that intermodule references will occur in the same table block is increased.

## SWTCH

When a generation has been completed, the new RTE operating system is stored in a disc file, and the SWTCH program is run to install it. SWTCH, a background program with embedded 7905A and 7900A disc drivers, assumes that any interfering real-time activity will be terminated during the short time it takes for the system transfer operation. Either a 7905A or 7900A disc-based RTE-II or RTE-III system may be installed, regardless of the disc type and version of RTE system currently operating. Since SWTCH's disc drivers are configured to a user-specified channel, SWTCH can transfer an RTE system to a disc I/O configuration that differs from the current disc I/O configuration, or to a temporary configuration different from the generation-defined configuration. Some useful SWTCH transfer modes are:

- Transferring the new system to the configuration of the currently running HOST system, overlaying it
- Transferring the new system to the generation-defined DESTINATION configuration
- Transferring the new system to a temporary TARGET disc configuration, defined by a user-specified disc controller channel and/or system disc subchannel/unit
- Transferring the new system to another disc drive (by specifying a TARGET subchannel or unit) to facilitate system distribution, as in a manufacturing environment
- Transferring the new system to the system sub-

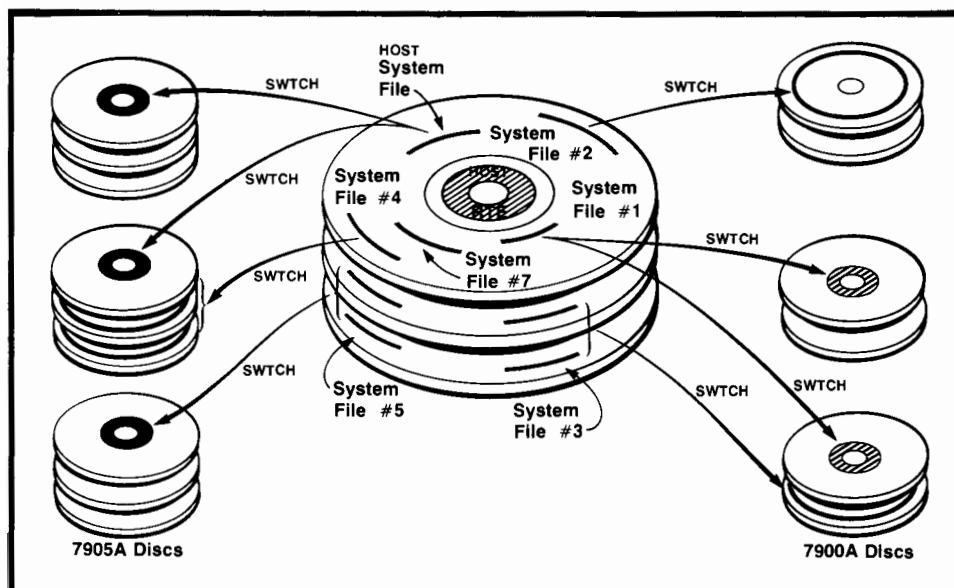
channel definition of the HOST, and then when given permission by SWTCH, replacing the host's removable disc cartridge with the disc cartridge destined for the storage of the new system.

SWTCH makes use of RTE's new, extended string communication mechanism (see next section) for the specification of up to six RU command parameters when being scheduled. In addition to the file name parameter of the RTE system to be transferred, the user may also specify the disc controller channel and the system disc subchannel/unit, and whether automatic boot-up (start-up) of the new system is desired, the target file structure is to be saved, or memory-image program files belonging to that target file structure are to be purged. Any missing or erroneous parameters are requested by SWTCH at the proper time during execution. If all parameters are correctly entered, SWTCH operates in automatic mode, requiring no operator intervention unless a requested option cannot be satisfied. A typical RU command scheduling SWTCH would be:

```
RU,SWTCH,SYSFIL:KH:10,21B,1,Y,Y,Y
```

which SWTCH would interpret as transferring the system contained in file SYSFIL:KH:10 via the disc controller in channel 21, with system subchannel/unit 1, whose file structure is to be saved, memory-image program files purged, and automatic boot-up performed when the transfer is completed. SWTCH determines the destination disc type from the absolute file, which indicates whether the 1 applies to a 7905A disc unit or a 7900A disc subchannel.

After retrieving and opening the new RTE system file, SWTCH displays the destination I/O configuration and the system disc subchannel definition. This gives



**Fig. 2.** The actual changeover from one RTE-II/III operating system configuration to another is performed by the utility program, SWTCH. Shown here are several types of SWTCH transfers of various RTE-II/III system configurations from their respective disc files to their target disc areas. An important feature of SWTCH is its ability to preserve the file structure of the system being replaced.

the user some additional information about the new system. It may be used for identification purposes and in answering the next set of questions SWITCH may ask concerning the target channel, subchannel/unit, and so on.

The greatest asset of a SWITCH transfer is the ability to save the file structure of the RTE system when updating the operating system configuration. To preserve a file structure, SWITCH must first verify that there exists a target file structure that conforms to the new system being transferred. Using the system subchannel definition (i.e., first track, number of tracks, surface number, etc.) obtained from the absolute file, SWITCH checks for a precise match between the physical boundaries of the new and target systems to insure the same logical track addresses of the file structure. At what will be the last track of the system subchannel, SWITCH must verify the existence of a cartridge directory and a file directory. At the same time, SWITCH retrieves some relative (logical) track addresses, which it then compares against similar information picked out of the disc-resident bootstrap loader located at logical track 0 of the system (provided its existence was also verified). If the comparison indicates that these two tracks reside in the same system, then the boundary tracks for the new and target systems are the same, indicating a precise match of system subchannel definitions. SWITCH is thus capable of saving the file structure of either the host system, a system located on a different disc subchannel/unit of the present hardware configuration, or an RTE-II/III system satellite in a distributed system network. Fig. 2 shows an example of a SWITCH operating environment.

SWITCH next compares the first logical track of the file structure with the track size of the new system plus the minimum 8-track free area needed by RTE. If the new system area is going to overlay part of the file space, the user is warned of the situation and given the opportunity to terminate SWITCH before the transfer is begun. If allowed to proceed, SWITCH will purge the overlaid files from the file directory, displaying their file names on the operator console. If the user chose to have memory image program files purged, SWITCH will display their file names as their file directory entries are purged.

During the transfer SWITCH initializes the tracks of the system subchannel as they are being written (when tracks are initialized, the physical track and sector addresses are written in the preamble of each sector). When the system area tracks are written, their preambles are set to indicate write-protected tracks. If a file structure is to be preserved, only the area up to its first track is reinitialized, otherwise the entire system subchannel is done. When a defective track is encountered during the initialization of a 7905A disc

subchannel, a spare track is assigned to it. The preamble of a defective track indicates that it is defective and gives the address of the spare track that is replacing it so the disc controller will automatically switch to that track on future references. For 7900A discs, any bad tracks encountered outside the system area are flagged defective; bad tracks within the absolute code of the system are not allowed.

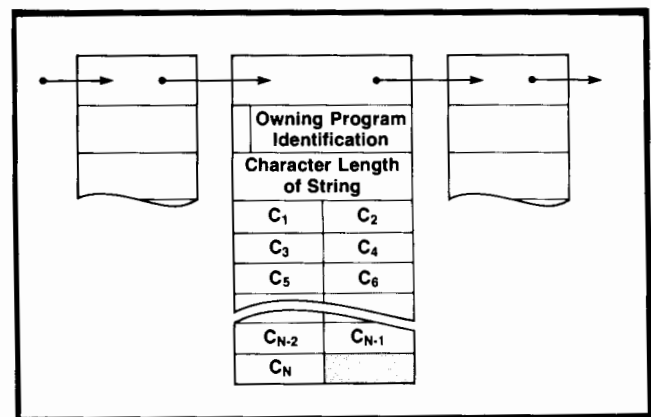
Automatic boot-up of the new system may occur following the transfer operation if all of the following conditions are determined to be true:

- the TARGET disc channel = the DESTINATION disc channel
- the TARGET disc subchannel/unit = the DESTINATION disc subchannel/unit
- the HOST time base generator channel = the DESTINATION time base generator channel (provided both are present)
- the HOST privileged interrupt channel = the DESTINATION privileged interrupt channel (provided both are present)
- the HOST system console channel = the DESTINATION system console channel.

If the above conditions are not met, SWITCH informs the user of its exit mode: either a return to the host system, or a halt if all or part of the host system was overlaid.

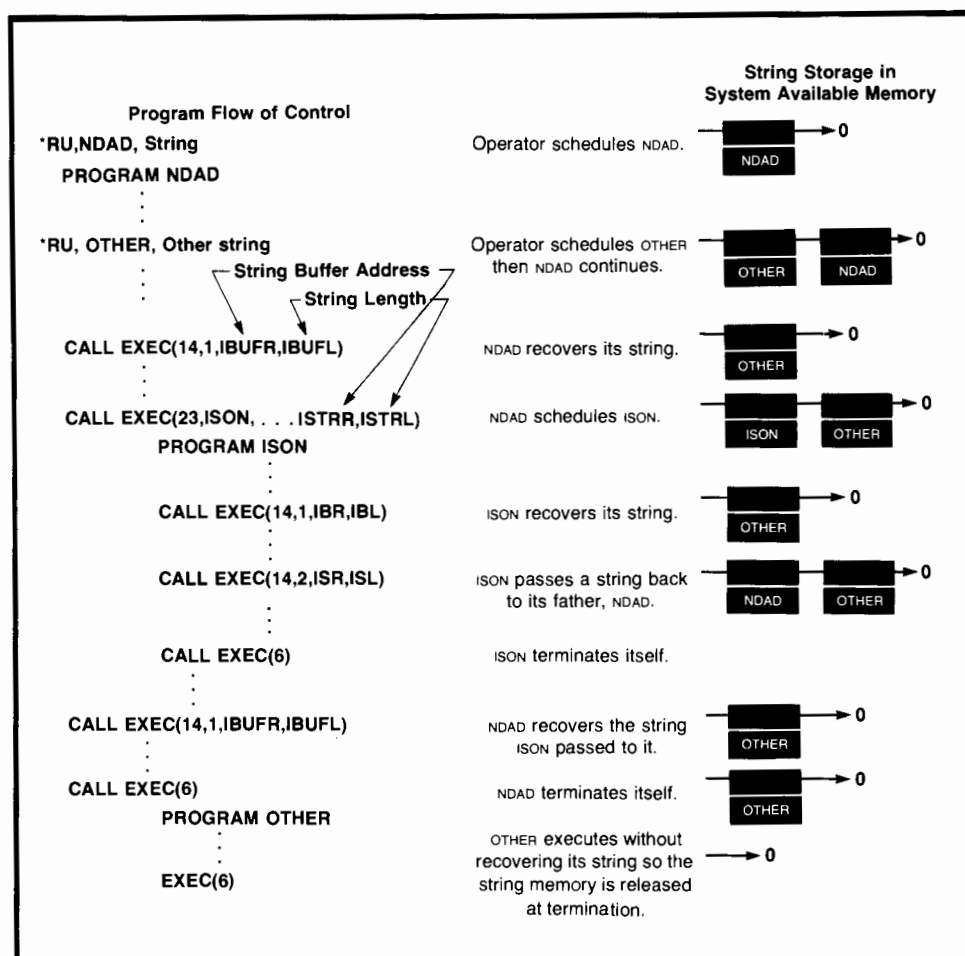
### System String Communication

Earlier versions of RTE limited the operator or scheduling program to ten bytes of information that could be passed to the scheduled program. To increase this communication area, HP 1000 RTE-II/III operating systems provide temporary storage of an operator's or a father (scheduling) program's scheduling string of information in system available memory, so it can be retrieved by the scheduled son program (see Fig. 3). Whenever an operator schedules a pro-



**Fig. 3.** HP 1000 RTE-II/III operating systems provide temporary storage of an operator's or a father program's scheduling string of information in system available memory, so it can be retrieved by the scheduled program. Shown here is the format of string communication storage in system available memory.





**Fig. 4.** An example of system string communication. Programs NDAD and OTHER are scheduled by the operator while program ISON is scheduled by NDAD. Each time a program is scheduled an optional buffer is saved for it in system available memory. Each time a program recovers a string this memory is released. When OTHER terminates, its unrecovered string is released.

gram via a "RU,PROGA, parameter string", the entire command string is saved. Whenever a father program schedules a son via an EXEC 9, 10, 23, or 24 call, two optional parameters are passed, pointing to a string buffer of information that will be saved. When the son executes, it may recover this saved string via an EXEC 14 call. In the case where the son was scheduled by another program, the son may also use the EXEC 14 call to return a new string of information to the father. String storage in system available memory is released when the program retrieves the string, when a son returns a new string to its father, or when the owning program is set dormant (normally or abnormally). String communication is used by the SWTCH program to recover up to six scheduling parameters.

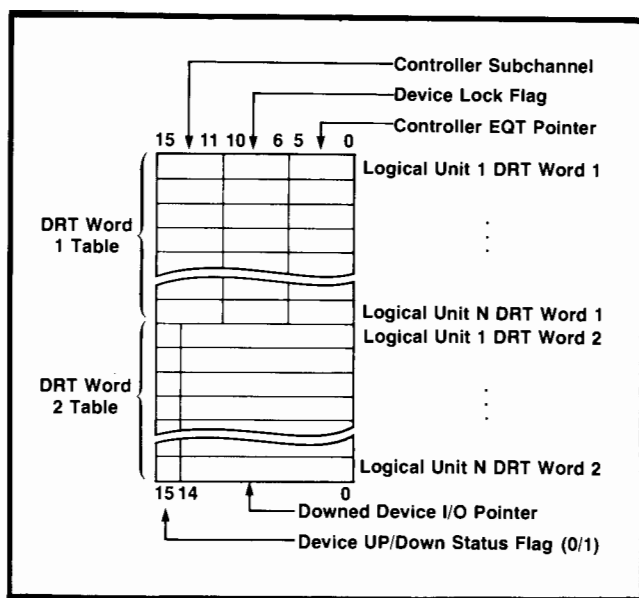
Fig. 4 shows two programs being scheduled by the operator, with one of these programs scheduling a son. System available memory is assigned for each string when requested and is released each time a program recovers its string. Note that since program OTHER made no effort to recover its own string, the memory assigned to the program is released when the program is set dormant.

If no system available memory is available when a user schedules a program, an error message (CMD

IGNORED-NO MEM) will be printed at the system console. Inhibit forms of the effected commands (RUIH,ONIH,GOIH) are provided to disallow scheduling string passage. If no system available memory is available when a father schedules a son (or when a son returns information to its father), then the father (son) will be temporarily suspended until sufficient memory is returned to system available memory.

### I/O Device Error Management

To accommodate the 2644A/45A Terminals and the HP-IB interface card, the RTE-II/III operating systems have been modified to better support multiple, dissimilar devices on one input/output controller. Previously when any device on a controller encountered an error condition (e.g., not ready, parity error, etc.), the controller and all associated devices were set into the down state by setting a flag in the controller's equipment table (EQT). This condition continued to exist, blocking all other devices' I/O on this controller, until the malfunctioning device was either fixed or disconnected and the controller was set into the up state. Besides needlessly delaying other devices' I/O, this caused problems when the 2645A Terminal was the system console. A malfunction on a mini-cartridge



**Fig. 5.** To improve the management of I/O device errors, the device reference table (DRT) was doubled in size. Each I/O device has associated with it one or more logical units, each of which has a DRT entry. The new DRT word 2 for each entry contains a device up/down status flag and a pointer. Formerly, device status flags were in the equipment table (EQT) of the corresponding I/O controller.

drive would never have set the 2645A controller down, since in doing so, it would also be setting the system console controller down, which is not allowed in RTE.

To alleviate this situation, the device up/down status was put where it belongs—with the device. Each device defined within RTE has associated with it one or more logical units (LU), each of which has a device reference table (DRT) entry. Each DRT entry was expanded to two words (see Fig. 5), with the second words collected into a second table following the original table. DRT word two contains a device up/down status flag and a pointer. Whenever a device is up, all I/O directed toward the device by the system is queued as before on the device's controller in order of program priority. If a malfunction occurs on a device (see Fig. 6), all LU's pointing to the device are flagged in their DRT word two as being down, and the condition is reported to the system console. The pointer in each LU's DRT word two is set to the LU number of the first LU for this device in the DRT (the device's major LU). The pointer for the device's major LU is set to point to the requeued I/O requests for the downed device, which were originally queued on the controller. This allows I/O requests for other devices using this controller to be processed while the malfunctioning device is being fixed. When the device is again operational, it can be set into the up state by setting the controller up. Alternatively, the downed device's I/O may be redirected to another device by

redefining the logical unit.

### Disc Backup Utilities

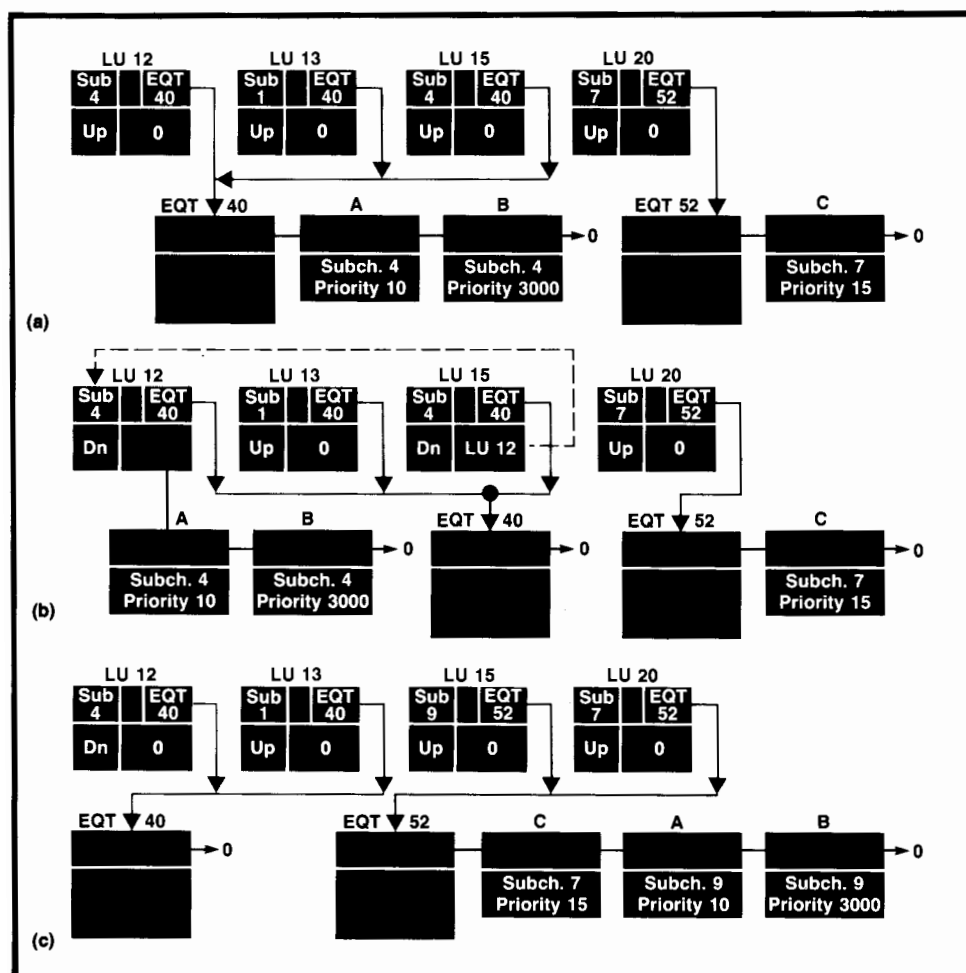
RTE-II/III disc backup utility programs have been created that SAVE information from disc to magnetic tape, RESTORE information from magnetic tape to disc, COPY information from one disc to another and VERIFY data transferred during any of the above operations. These operations may be done either on-line under control of the RTE-II/III operating systems or off-line. On-line transfers occur in logical mode using subchannels defined by the RTE-II/III system. Off-line transfers use physical address spaces defined by the user. These utilities support 7905A and 7900A Disc Drives, 9-track 7970B Magnetic Tape Drives and all RTE supported terminals.

Three modes of data transfers are supported. LU mode transfers data on one RTE-II/III-defined subchannel (on-line only). UNIT mode transfers data on an entire disc drive (both on-line and off-line). FROM-TO mode transfers data on an area of the disc defined by the user (off-line only).

### Batch/Spool Monitor and Other Changes

Several modifications were made to the RTE II/III batch/spool monitor (BSM). BSM includes RTE's file management package, a job entry processor, and input/output spooling to disc files. Two new commands were added to the FMGR program. First, all system-level commands were implemented within FMGR using a "SYxx, parameter string" format, where xx is the command. This now makes it possible to execute most system functions from the FMGR program either interactively or through user-defined procedure files. Second, FMGR comments were implemented by ignoring any command starting with \*. The BSMRU command was modified so the command string can be passed to the scheduled program in a manner similar to the system RU command. Error reporting within the BSM subsystem was improved. All abort conditions resulting from EXEC calls that can be processed by the caller are now handled by the BSM subsystem, resulting in BSM error messages and the non-abortion of BSM processes. Spooling errors that would result in illegal driver request errors were given new system spooling error messages. When the FMGR program is scheduled by a program with a non-interactive input device, any optional scheduling string in the EXEC call that begins with a colon is inserted as the first FMGR command. Finally, whenever the system is booted up, the FMGR program is scheduled after any user-defined start-up program, with control passed to a file named WELCOM.

Several other modifications were made to RTE-II/III modules. DVR05, the software driver for HP 264x terminals, was modified to accommodate the 2645A



**Fig. 6.** An example of I/O device error management. In (a), three I/O requests, A through C, are queued on two controllers (EQT 40 and 52). All four logical units shown are in the up state. In (b), a "not ready" error occurs on the device associated with logical unit (LU) number 15. This causes LUs 12 and 15 to be set into the down state. I/O requests A and B are dequeued from EQT 40 and requeued on LU 12. The pointer field in LU 15 is set to point to LU 12, the device's major LU. All programs sending input/output to the downed LUs are suspended. In (c), the operator reassigns LU 15 to an active device, EQT 52 subchannel 9. This causes requests A and B to be queued on EQT 52, LU 15 to be set into the up state, and all suspended programs using this LU to be restarted. LU 12, pointing to the downed device, is left unchanged.

Terminal. The software driver for the 7900A Disc Drive was modified to work with the 21MX E-Series Computer. The WHZAT system status program was modified and also released with RTE-II. WHZAT reports the status of all non-dormant programs, all partitions, and any I/O devices or controllers that are in the down state. Finally, to implement the string communication and I/O device error management handling capabilities, the RTE-II/III operating systems grew in size. To reduce this growth, a foreground disc-resident program, \$SCMD, was added that implements the system LU (logical unit switching), EQ (EQT status), and TO (change timeout) commands.

### Disc Distribution of Software

With the addition of the on-line system generator, the RTE-II/III operating system software products can be distributed on 7905A or 7900A removable "grandfather" disc cartridges. Each grandfather cartridge contains a minimum system supporting a wide variety of peripheral devices that can be booted up on any I/O configuration. A file structure containing all software distributed with RTE-II or III provides a convenient storage medium from which the on-line

generator can configure one or more systems customized to the user's needs. The user can easily create archive copies of software or master software libraries by copying the grandfather cartridge to magnetic tape or another cartridge using FMGR commands or the disc backup utilities.


For subscribers to Hewlett-Packard's software subscription service, update software will be distributed on either HP mini-cartridge or paper tape. Easy-to-use update utilities provide a simple means of updating grandfather or master software library cartridges from HP mini-cartridges.

Other RTE software products (e.g., IMAGE/1000, microprogramming package, multi-user real-time BASIC) are now available on HP mini-cartridges. Each mini-cartridge with the exception of off-line device diagnostics contains an ASCII source directory of the items on the mini-cartridge followed by from one to ten software parts.

### Acknowledgments

Many people contributed their efforts and guidance to enhancing RTE-II/III. We wish to acknowledge the extensive efforts of George Anzinger and Adele Gadol

in enhancing the batch/spool monitor, Shaila Kapoor for developing the disc backup utilities, and Bill Bowman for his work with the 2645A Terminal. Also, Nancy Rohlf, Alan Sanderson, and Larry Jones for their efforts in making these products manufacturable; Eugene Wong, Doug Baskins, Marlin Schell, Jim Hartsell, and Fred Warren for their pre-release testing; Van Diehl for his marketing expertise; Carl Davidson and Nino Mateos, quality assurance; Peter Baker, John Gowan and Dave Tribby, our technical writers; and the efforts of Joe Bailey, Linda Averett

and Mike Kaessner in giving our efforts a critical review. 

## References

1. G.A. Anzinger and A.M. Gadol, "A Real-Time Operating System with Multi-Terminal and Batch/Spool Capabilities," Hewlett-Packard Journal, December 1975.
2. L.W. Averett, "Real-Time Executive System Manages Large Memories," Hewlett-Packard Journal, December 1975.
3. S. Dickey, "Distributed Computer Systems," Hewlett-Packard Journal, November 1974.



**Kathleen F. Hahn**

Kathy Hahn was responsible for the on-line generator and switch portions of the RTE enhancements project. A native of Sioux City, Iowa, Kathy attended Iowa State University, graduating in 1973 with a BS degree in computer science and mathematics. She started with HP the same year, working on the ATLAS compiler project prior to joining the RTE software group. She's now continuing her studies toward an MSEE in computer engineering at Stanford University. Kathy enjoys sewing, tennis,

jogging, and gardening. She and her husband, a CPA who also works for HP, live in Sunnyvale, California.



**David L. Snow**

As project manager for the RTE-II/III enhancements project, Dave Snow had responsibility for all the operating system enhancements. A native of Austin, Texas, Dave joined HP in 1972 after graduating from the University of Texas with a BS degree in electrical engineering. He is currently working on his MSEE degree at Stanford University. Before joining the RTE software group, Dave was production engineer for the RTE-II and TODS-C operating systems and the 9500 Automatic Test System.

He enjoys skiing and tennis, is a member of the Air Force Reserve, and just recently bought a home in San Jose, California.

## HP 92001B Real Time Executive II Operating System (RTE-II)

## HP 92060B Real Time Executive III Operating System (RTE-III)

### FEATURES

- Real-time and background multi-user swapping partitions in RTE-II, 64 partitions in RTE-III
- Up to 64K bytes of memory in RTE-II, 2M bytes in RTE-III (hardware limited to 608K bytes)
- Ample partition space for user programs, typically 37K bytes
- Support for choice of 4.9M-byte or 14.7M-byte cartridge disc subsystem, the latter expandable to 117.9M-byte capacity
- Time, event, and program-to-program scheduling for real-time measurement, control, and/or automatic test applications
- Batch-spool monitor for concurrent disc file management and batch processing
- Input/output spooling to disc to speed throughput without excessive use of main memory for buffering
- Interactive text editor to aid program development
- Concurrent processing and program development in FORTRAN II/IV, conversational Multi-User Real-Time BASIC (optional), HP ALGOL, and HP assembly language
- Optional RTE microprogramming package for on-line development of new user-microprogrammed instructions for system computer
- Multi-terminal access to all system resources, serving multiple users concurrently
- Includes RTE drivers and device subroutines for supported peripherals
- Choice of on-line or off-line system generation
- Support of multiple instrument clusters connected via the Hewlett-Packard interface bus (HP-IB). The Hewlett-Packard interface bus is Hewlett-Packard's implementation of IEEE Standard 488-1975, "Digital Interface for Pro-

grammable Instrumentation."

Support of IMAGE/1000 data base management system for more efficient use of data files, easier access to data

Supports operation as distributed system network central

Distribution on disc cartridge

### ORDERING INFORMATION

92001B-010 RTE-II distributed on paper tape

92001B-030 RTE-II distributed on 7900A disc cartridge

92001B-031 RTE-II distributed on 7905A disc cartridge

The 92001B-030 or 031 RTE-II system is included in the 2170A, 2171A, and 2172A Computer System building blocks, which form the basis of the HP 1000 Model 30 and Model 31 Computer Systems, as the standard operating system.

92060B-010 RTE-III distributed on paper tape

92060B-030 RTE-III distributed on 7900A disc cartridge

92060B-031 RTE-III distributed on 7905A disc cartridge

The 92060B-030 or 031 RTE-III system is included in the 2170A, 2171A, and 2172A Computer System building blocks with Option 001, which form the basis of the HP 1000 Model 80 and Model 81 Computer Systems, as the standard operating system. The 92060B-030 or 031 is also available in the HP 1000 Model 30 or Model 31 system by ordering 2170A, 2171A Option 001, or by later substituting 92060B-030 or 031 for the 92001B-030 or 031 along with 13304A Firmware Accessory Board, 12731A Memory Expansion Module, 13307A Dynamic Mapping Instructions, and two 13187A 32K-byte Memory Modules.

### PRICES IN U.S.A.:

92001B-010, \$5,000	92060B-010, \$6,000
92001B-030, \$5,000	92060B-030, \$6,000
92001B-031, \$5,000	92060B-031, \$6,000.

### MANUFACTURING DIVISION: DATA SYSTEMS DIVISION

11000 Wolfe Road  
Cupertino, California 95014 U.S.A.

# Development and Application of Microprograms in a Real-Time Environment

by Harris Dean Drake

AS THE USE OF MICROPROGRAMMABLE processors and microprogramming proliferates, more and more users are addressing themselves to the application of microprograms in a system environment. To facilitate microprogram development, Hewlett-Packard offers a combination of hardware and software that includes the 1K writable control store (WCS) I/O board and the RTE microprogramming package consisting of a microassembler, cross-reference generator, WCS driver, WCS load utility, microdebug editor, and PROM tape generator.

At first glance the difficulty of developing a microprogrammed product may seem somewhat intimidating. The application must be defined, the source program written, the logic flow debugged and possibly PROM or ROM parts created. In the case of a dedicated processor performing a specific set of tasks, the application is dictated directly, but deciding what or how much is to be microcoded is not so straightforward. The most economical method is to use the system itself to produce a profile of system activity, then use the profile to pinpoint the areas most likely to increase system performance and define the microprogram application accordingly.<sup>1</sup>

## Microprogram Development

Once the microprogramming requirements are established, the microcode itself must be developed. The initial source code is translated into micro-object code using the microassembler with optional cross-reference generation. The object code may then be loaded into WCS using a WCS load utility and the RTE driver for WCS.

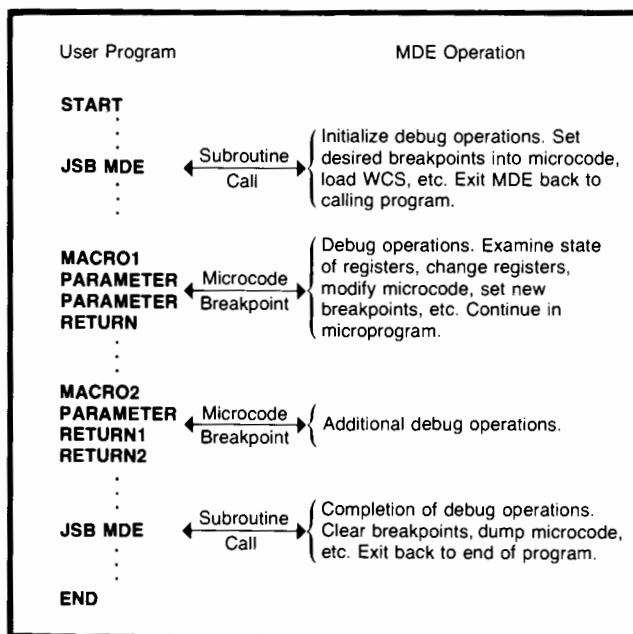
WCS provides a number of development advantages over simulators. One is that all execution and debug operations are on-line. Also, the user doesn't have to contend with the typical shortcomings of simulators. I/O operations, for example, are almost impossible to simulate faithfully.

Another advantage of WCS when used with RTE is improved interaction in microprogram software generation using the standard RTE capabilities of interactive editor, disc file manipulation, and so on. Most computer systems used for the development of microcode must be shared with other system operations

on a mutually exclusive basis, while RTE allows normal operations and microprogram development together.

The microdebug editor (MDE) program with its RTE-type operator interface facilitates debugging of a user microprogram by combining debug functions (set breakpoints, initialize microprogram calling parameters, etc.), edit functions (replace microinstruction, delete microinstruction, etc.), and some system functions (disc file dump, WCS logical unit manipulation, etc.). Edit operations are performed using the symbolic form of the various microfield mnemonics rather than numeric object codes.

In a system a microprogram may be subjected to a wide range of input parameters and execution conditions. To further assist a user in debugging his mi-



**Fig. 1.** Part of the HP 1000 microprogramming package, the microdebug editor (MDE), facilitates debugging of user microprograms. MDE is callable as a utility subroutine that can be appended to a user program for interactive debugging operations, as shown here. Other elements of the RTE microprogramming package are a microassembler, a cross-reference generator, a WCS driver, a WCS load utility, and a PROM tape generator.

croprogram in the environment of his program, MDE is also callable as a utility subroutine. Fig. 1 demonstrates a typical use of MDE as a subroutine appended to the user's program for debugging.

After the microprogram is debugged it must be implemented in some fashion. The microcode may reside permanently in ROM or dynamically in RAM (WCS) or a combination of both. When a microprogram is programming a "processor on a board" such as the 21MX K-Series processor, ROM is almost always used. In the E-Series processor up to 3.5K words of firmware may be added to the CPU with no extra power or I/O requirements. In a system, power fail considerations might be the dominant factor in a decision to use ROM to hold a microprogram; the contents of WCS are lost in the event of a loss of power. A PROM mask tape generator is provided for the purpose of burning PROM devices. The generator produces a variety of PROM mask tape formats to enable a number of vendors to burn parts for the user.

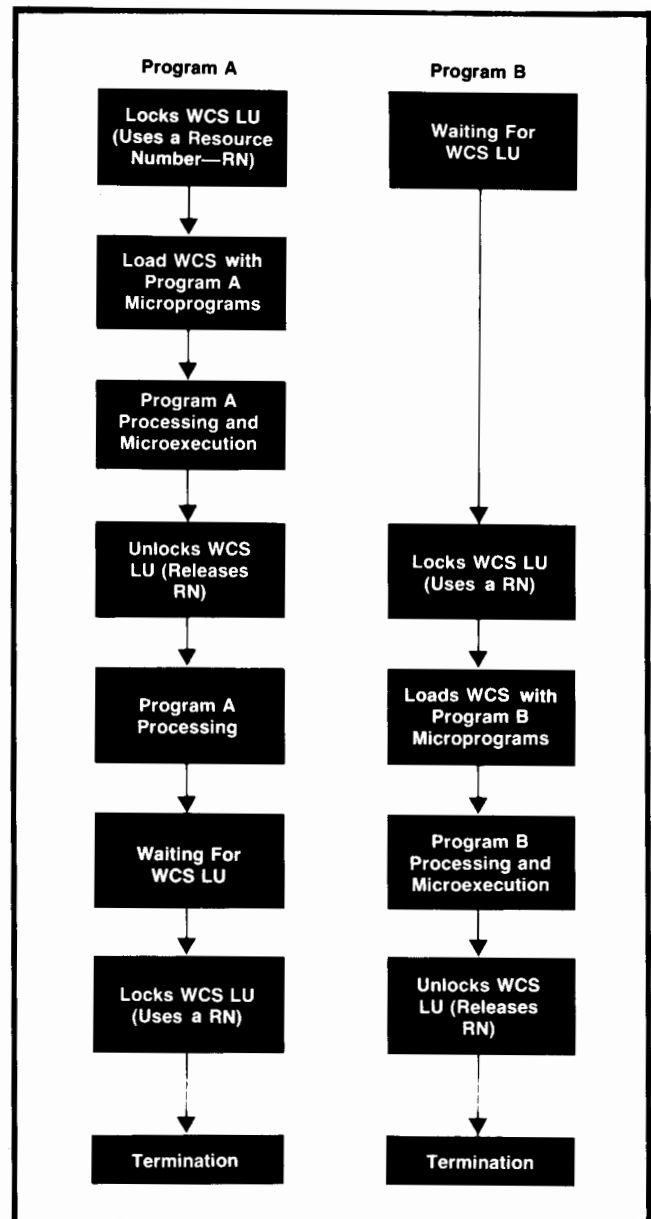
#### WCS As a System Resource

Can WCS be used as a system resource? In a real-time, multiprogramming operating system environment it can, with certain constraints. Several modes of microprogram use are possible.

Multiple programs may use a single set of WCS-resident microprograms. In this case the microcode is loaded once and the programs link to the micro-routines as needed. The only possible constraint might be that if microprogram reentrancy is desired, it will have to be provided for in the structure of the microprograms themselves.

The same WCS area may be used by multiple programs with different microprograms. Since each program requires that program's microcode to be in control store, a new constraint is added. A method must be determined by which each program has exclusive use of WCS when needed. This is accomplished in RTE through the sharing of resource numbers. Fig. 2 shows an example of two programs using the same WCS space for their microprograms. Program A obtains the first lock on a logical unit (LU) associated with WCS while program B must wait for the LU to become available. Program A then loads and executes its microprograms. When the microcode is no longer required, the WCS LU is unlocked and program A continues processing. Program B now gets the WCS LU and locks it. Program B repeats the sequence of operations performed by program A, which is now waiting for the WCS LU to be available. The two programs cooperate in this manner to termination.

Several WCS areas may be used by multiple programs with a combination of separate and identical microprograms. This may require more WCS boards, but makes the best use of WCS as a system resource. It




**Fig. 2.** Writable control store (WCS) can be used as a system resource in HP 1000 Systems. Here two user programs use the same WCS space for different microprograms.

creates no additional restrictions, but requires closer attention to system planning.

#### Acknowledgments

The RTE Microprogramming Package is the result of the efforts put forth by many people. Development of the microassembler, cross-reference generator, WCS driver, WCS load utility, and PROM tape generator was done by Ken Mintz. Jack Howard contributed toward the design of MDE and prepared the training course. Special thanks to Don Reid and Art Purdy for their work on the manuals. Thanks also to Bonnie Lundberg, production engineer, Van Diehl, product manager, Gary Gubitz, support engineer and



Bob Nicholson and Tom Engleman, quality assurance. 

## Reference

1. D.C. Snyder, "Computer Performance Improvement by Measurement and Microprogramming," Hewlett-Packard Journal, February 1975.

## Glossary

DCPC: Dual Channel Port Controller  
 RTE (or RTE-II or RTE-III): Real-Time Executive, the System 1000 operating system.  
 ROM: Read-Only Memory  
 PROM: Programmable Read-Only Memory  
 RAM: Random-Access (read/write) Memory  
 WCS: Writable Control Store, a random-access memory board that can be plugged into the 21MX Computer like an input/output (I/O) board to hold user microprograms  
 MDE: Microdebug Editor, a program supplied with the HP microprogram development package



### Harris Dean Drake

Dean Drake graduated from the University of California at Davis with a BSEE degree in 1969. After a stretch in the U.S. Army and some programming experience writing modules for an on-line test system, he joined HP in 1973. Among his projects are the CPU, memory, and WCS diagnostics for the 21MX, some modifications to the RTE-C FORTRAN compiler, and the RTE microprogramming package. Dean was born in Tulsa, Oklahoma. He's married, has a son, and lives in Santa Clara, California. His principal recreational activities are handball—he's a member of a local club—and backpacking.

## HP Model 92061A RTE Microprogramming Package

The 92061A is a support package for on-line development by the user of special microprogrammed instructions for HP 21MX and 21MX E-Series Computers.

### FEATURES

- On-line operation in RTE-II/III system
- Simple assembly language for microprogramming
- Cross-reference generator for simplified program development
- Microdebug editor for interactive program editing and checkout
- Operator-entered microprogram breakpoints
- Full WCS support, including driver, load utilities, and load verification routines
- Dynamic WCS overlay utilities
- Up to 3072 instructions in WCS
- PROM tape generator for outputting production microcode on (punched) PROM "burn" tapes in user-specified format

### FUNCTIONAL SPECIFICATIONS

#### ENVIRONMENT

92001A RTE-II system with 92002A Batch-Spool Monitor or 92001B RTE-II system or 92060A/B RTE-III system.

#### MEMORY USAGE

The WCS driver requires 1080 words of resident memory. Other programs in the RTE Microprogramming Package require an 8K-word background partition in RTE-II or a 9K-word partition in RTE-III, including the 1K words re-

quired for base page in each RTE-III disc-resident partition.

### MICROPROGRAM CAPACITY

The WCS Load Utility and Driver programs work with up to three 13197A WCS boards (3072 user instructions) in the computer.

### MINIMUM REQUIREMENTS

92001A/B RTE-II or 92060A/B RTE-III operating system.  
 12960A (4.9M byte) or 12962A/B/C/D (14.7M-byte) cartridge disc subsystem  
 HP 2108A, 2109A, 2112A, or 2113A Computer.  
 Batch-Spool Monitor (included in 92001B and 92060A/B).  
 At least 24K words of memory in RTE-II system, 32K words of memory in RTE-III system.

### OPTION 020

Replaces the punched tape software modules listed under items 1 through 7, above, with software on one 9162-0061 HP mini-cartridge (92061-13301) for read-in by 2645A-007 or 2644A CRT Terminal interfaced to the computer via the 12966A-001 interface and to the operating system via RTE driver DVR05.

### PRICES IN U.S.A.:

92061A, \$1000.

Option 020, N.C.

### MANUFACTURING DIVISION: DATA SYSTEMS DIVISION

11000 Wolfe Road  
 Cupertino, California 95014 U.S.A.

(continued from back page)

Errors from mechanical tolerances can be eliminated by e-beam exposure of the mask patterns directly on the etch resist on a wafer. This technique avoids errors associated with mask aligners and it compensates for the differing thermal expansion of the masks and the silicon wafer. Dimensions of less than  $1\mu\text{m}$  seem achievable with e-beam but, unfortunately, the dimensional stability of a wet-etched pattern is not adequate for this degree of resolution. Dry etching is needed to take advantage of the resolution and accuracy possible with direct e-beam exposure.

Even if the necessary etching precision were attainable, at the present time the e-beam approach to direct pattern formation is not practical from an economical point of view. The high capital investment in an e-beam machine, which can exceed one million dollars, and the relatively slow exposure rate—typically 30 minutes per exposure—means that during the exposure time required by the six

to eight levels typical of most processes, the depreciation alone would cost several hundred dollars per wafer.

Nevertheless, as various dry etch steps become feasible, process refinements using present lithography will enable us to use finer lines. Furthermore, the e-beam technology is immediately economical for making more accurate masks. Possibly not-yet-perceived mechanical and optical inventions will give better mask-to-mask registration, continuing development of higher density processes for large-scale integrated circuits.



Well-known to a whole generation of circuit designers for the Ebers-Moll model of the large-signal behavior of transistors, John Moll's affiliation with HP began over twenty years ago as a consultant to HP's first semiconductor operation. John has been on the faculties of Ohio State and Stanford Universities, and he has worked for RCA (magnetrons), Bell Labs (transistors), and Fairchild (microwave and optoelectronics). He is now Technical Director of the IC lab within HP Labs, HP's advanced research facility.

# E-Series Doubles 21MX Computer Performance

*Faster logic, improvements to the architecture and firmware, and new microprogrammed features greatly increase performance without significantly increasing cost.*

by Cleaborn C. Riggins

**T**HE 21MX E-SERIES (Fig. 1) is an enhanced member of the 21MX Computer family introduced in 1974.<sup>1</sup> The E-Series is designed to increase processor speed without significantly increasing its cost. The performance range of the 21MX family is expanded by a factor of approximately two by the E-Series.

The 21MX E-Series Computer is the processor in the HP System 1000, a cost-effective, real-time, multi-programming computer system (see article, page 2). The E-Series also provides a cost-effective solution for the original equipment manufacturer (OEM) who adds value through hardware and/or software.

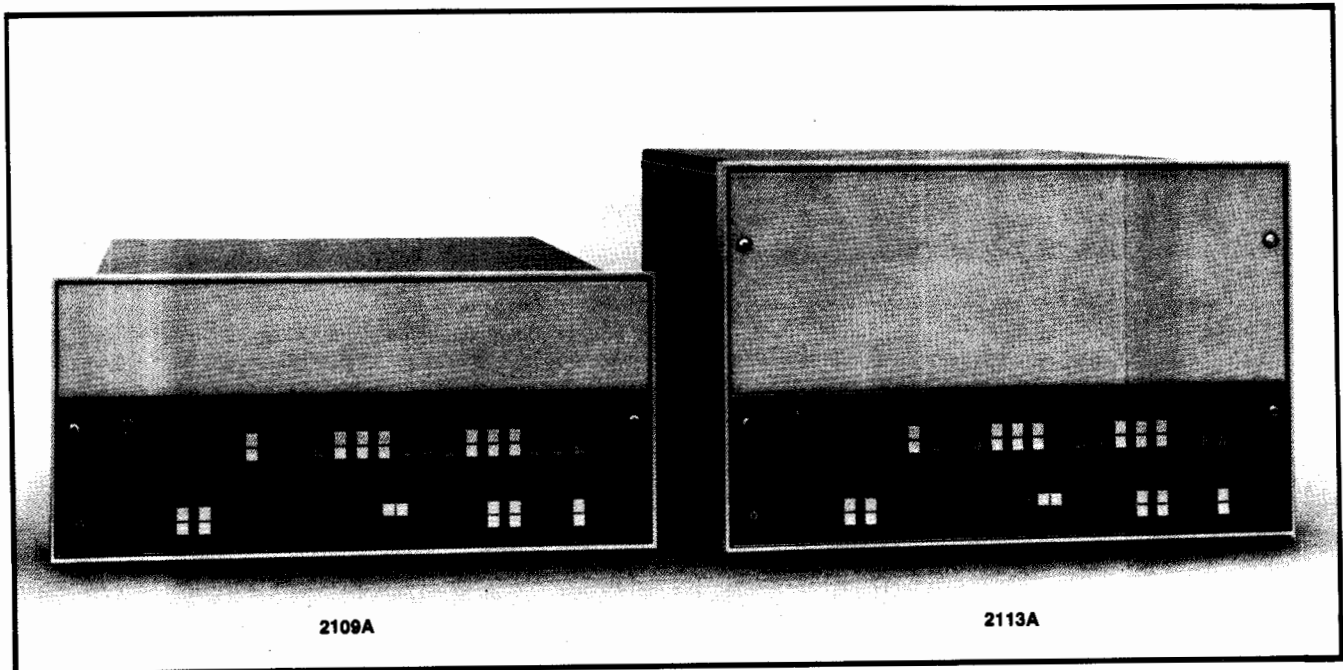
## What Was Enhanced

Greater performance within the 21MX family was the major goal of the designers for the E-Series. Of equal importance was that the user be able to add to

the machine's capability and that it be possible to add future products without redesign of the computer.

One of the first questions to be answered was the technology to be used. Speed improvements within the T<sup>2</sup>L bipolar logic family provided the designers with the necessary performance. This choice of technology had the added benefit that much existing hardware design could be used.

Although faster technology contributes to the performance (approximately 15-20%), it alone was insufficient to meet the design goals. To provide additional performance improvements the computer architecture was examined by the design team. The logic structure, microcycle timing and memory system were all investigated for possible design improvements. In almost every area significant speed improvements were accomplished. The following articles explain what was done.



**Fig. 1.** 21MX E-Series Computers have approximately twice the performance of earlier 21MX Computers (the M-Series), at about the same cost. They are used as the processors in HP 1000 Systems and are suitable for OEM applications as well.

To provide growth power for this new 21MX family member, the user's ability to microprogram the computer was increased in the E-Series.

- The control store address space was increased four times to 16K words, of which up to 8.5K are available to the user.
- A new 1K WCS (writable control store) was designed. It increases the program space in WCS while reducing the power requirements.
- New software was designed to assemble, edit, and dynamically use microprograms in a real-time operating system (see article, page 15).
- Many microinstructions were enhanced to perform more work within a single microcycle instruction time.
- A new firmware accessory board was designed to house the firmware (microcode in ROM) offered by Hewlett-Packard; it also has space for the user to add his own firmware.
- A new feature, the microprogrammable processor port, allows external processors (such as another E-Series, or a special arithmetic processor) to be connected to the fast internal data bus and be controlled by microprogram.
- Another new feature, microprogrammed block I/O, allows blocks of data to be transferred through the I/O structure under microprogram control.
- Remote program load was added.
- Standard self-test firmware automatically tests the memory system and the internal data paths and registers.

#### How the User Benefits

The most obvious benefit to the user is the additional performance the E-Series provides. This performance is attained with only a small incremental cost, so the performance/price ratio is much greater. The 21MX instruction set is retained, so previously developed software is still usable. The E-Series also provides the user a complete spectrum of software and hardware products, one of the benefits of being a family member.

User microprogrammability, a powerful feature of the 21MX family, is included and enhanced in the E-Series. A user's initial plans may not involve this feature, but when the computer approaches 100% loading, the power of microprogramming can improve its performance so the same computer can do more work. The built-in self-test feature helps by notifying the operator of malfunctions and by building confidence that the computer is healthy when no problem exists.

As a 21MX family member, extensive testing with existing systems and test software was possible at very early stages. These tests have been on-going since February 1976, helping to assure the solidarity

of the product.

#### Acknowledgments

The original concept of enhancing the 21MX family was formulated by John Stedman, Phil Gordon, and Greg Hansen. Phil Gordon was the chief designer of the 21MX E-Series computer; his ingenious ideas, hard work, and application of concepts into working circuits made the E-Series the performer it is. Don Cross designed the memory protect and memory controller and assisted in the DCPC design. Scott Stallard designed the firmware accessory board and with Tom Lane contributed to the testing of the prototypes and provided design improvements during the pilot production phase. Scott also accompanied the E-Series into production to assure a smooth pilot production phase. Earl Stutes provided the microprogram for FFP and DMS and assisted heavily in system testing with George Anzinger and Linda Averett. Jim McClure provided fresh insight by reviewing the design and discovering potential problems. Frank White, Bill Thormahlen and Tom Engleman performed the quality assurance, verifying that the E-Series met all the specifications at prototype and pilot phases. Steve Adams designed a guided probe test system that is invaluable to production testing. Chuck Habib's PC department did an outstanding job of laying out the printed circuit boards in a timely manner. Bill Ribble and his production engineering staff were involved early in the cycle to assure producibility. Bill Senske and staff provided the supportability and manual writing. Bob Frankenberg, Dave Carver, and Orin Mahoney contributed the marketing plan, sales literature and manual content. ☐

#### Reference:

1. J.M. Stedman, "A User-Oriented Family of Minicomputers," Hewlett-Packard Journal, October 1974.



#### Cleaborn C. Riggins

Cle Riggins is project manager for the 21MX E-Series. With HP since 1960, he's served as design engineer, test methods supervisor, production engineering manager, and project manager. A member of IEEE, he organized and chaired a WESCON 1976 session on computer power supplies. Cle was born in Dill, Oklahoma. Returning to school after four years in the U.S. Air Force, he received his BSEE degree from Oklahoma State University in 1960. In 1970, he received his MSEE degree from Santa Clara University. He's active in church work and enjoys fishing, bowling, bridge, and woodworking. He and his wife and three children live in San Jose, California.

# How the E-Series Performance Was Attained

by Scott J. Stallard

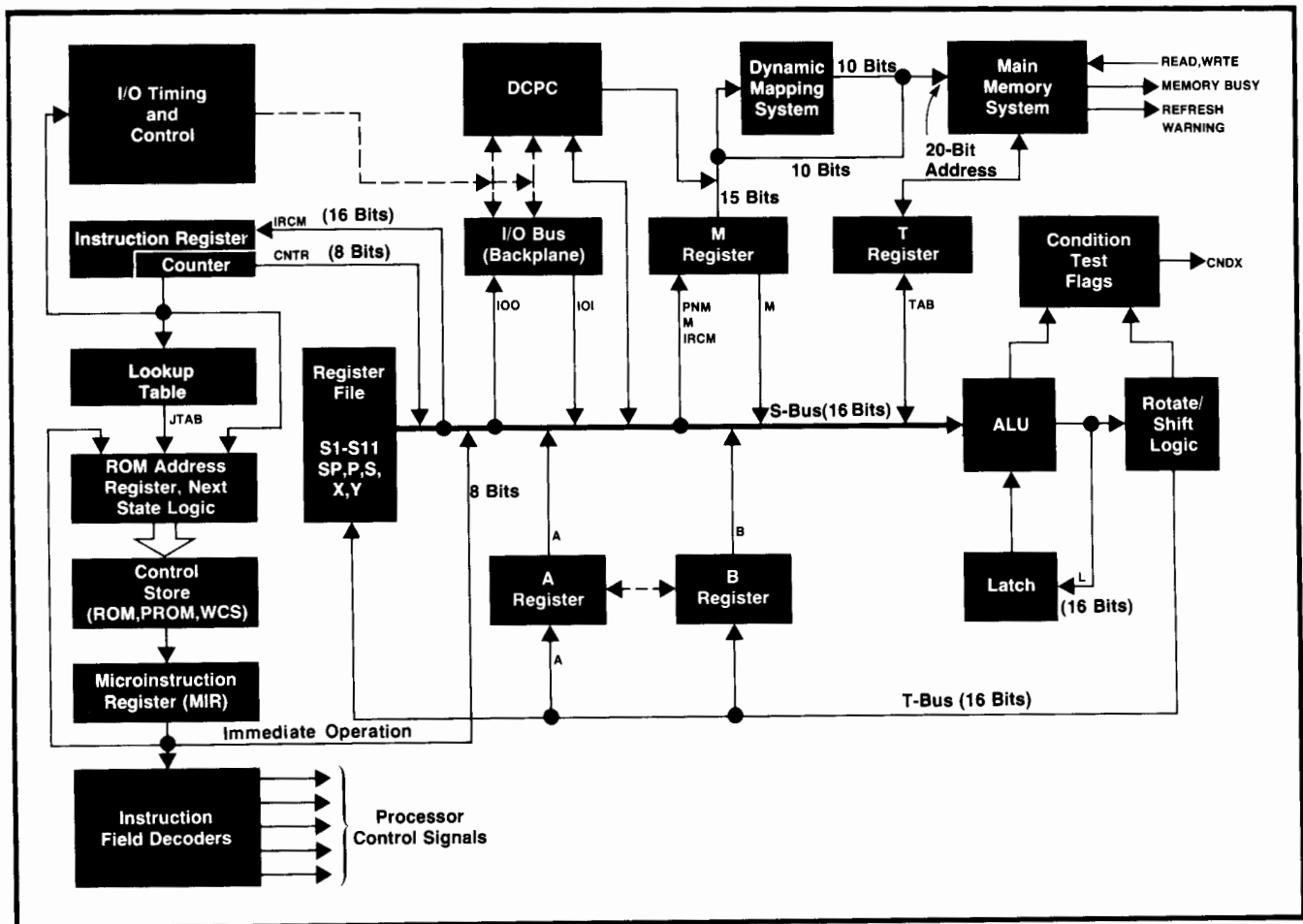
**T**HE DESIGN GOAL FOR THE 21MX E-Series Computer project was to increase the performance of the 21MX family. Our investigation showed that the dual objective of increased speed with minimal hardware changes could best be realized by concentrating the design efforts on a new CPU (central processing unit) board. We found that major performance increases could be realized using the existing memory system, input/output system, dual channel port controller (DCPC), and dynamic mapping system (DMS) designs of the 21MX, thereby lowering both development time and production costs.

First, critical areas of the data path and control

processor were made faster while increasing board density. Because technology had progressed since the 21MX was designed, we were able to use TTL Schottky and low-power Schottky (74S-, 74LS-) devices.

## Variable Microcycle Timing

Second, certain modifications to the microprogrammable control processor were made to optimize the control sequence according to the microinstruction to be performed during a given microcycle. The 21MX family is based on a microprogrammed bus-oriented processor, as shown in Fig. 1. During one



**Fig. 1.** 21MX E-Series Computer is based on this microprogrammed, bus-oriented processor. Performance has been improved over earlier 21MX models by using faster circuits and optimizing microprocessor operation.

microcycle, the contents of any register may be placed on the data (S) bus. The ALU (arithmetic/logic unit) performs some operation involving the data and the latch, after which the result may be shifted and then stored in the destination register.

The microinstruction word types of the 21MX have been retained.<sup>1</sup> In a simple word type 1 operation, such as

INC A S2

which stores into the A register the incremented value of S2, the sequence of control states is divided into five periods, numbered P1 through P5 (Fig. 2). During P1, the microinstruction is loaded into the control processor's microinstruction register (MIR), and decoding is initiated. During P2 and P3, the S2 register is placed on the data (S) bus. By the end of P4, the increment operation is complete and the valid result may be clocked into the A register at P5. Since this type of operation occurs frequently, the state (P) assignments and timing were optimized to perform this microinstruction type in the minimum time as dictated by the data path device delays. Each P period is 35 nanoseconds, so the microinstruction execution time is  $5 \times 35 = 175$  nanoseconds.

In parallel with the execution of the word type 1 microinstruction the control processor reads the next sequential microinstruction from control store, so that delays through control store, whether it is ROM, PROM, or RAM (writable control store), are transparent. However, if a branch instruction is encountered, a different sequence of events must take place. For example, a word type 3 instruction, such as

JSB CNDX OVFL RJS 157B

will conditionally jump to the subroutine beginning at control store location 157<sub>8</sub> if the overflow bit is not

Period	Microprocessor Control State
Previous P5	
P1	Load Microinstruction Register, Start to Read Next Sequential Microinstruction
P2, P3	Decode Word Type, Decode Source, ALU Function, and Destination Register as Required
P4	Perform ALU Function
P5	Store Result into Destination Register, Perform Special Tasks
Next P1	

**Fig. 2.** Each microinstruction executes in one microcycle, which is divided into P-periods, each 35 ns long. For a word type 1 microinstruction there are five P-periods and the microcycle is 175 ns long.

Period	Microprocessor Control State
Previous P5	
P1	Load Microinstruction Register, Start to Read Next Sequential Microinstruction
P2, P3	Decode Word Type, Check to See if CNDX Is Met.
E1	Read Branch Target
E2, E3, P4, P5	Access New Microinstruction
Next P1	Execute Target Microinstruction(157)

**Fig. 3.** Microcycle for a word type 3 instruction contains three additional 35-ns periods to allow time for branching. Thus the E-series timing is adjusted according to the requirements of each microinstruction instead of being fixed according to worst-case conditions.

set (RJS); otherwise the processor will execute the next sequential microinstruction. The control processor sequence is shown in Fig. 3. P1 is similar to that in word type 1. During the decoding events in P2 and P3, the JSB operation is detected, and the condition code is evaluated to determine whether a branch is to be performed. If it is, three additional periods, E1, E2, and E3, are inserted into the control state sequence to allow the required time to reload the microaddress register and access the contents of target address 157B from control store. The execution time, therefore, for unconditional branches or conditional branches where the condition is met, is  $(5 \times 35 \text{ ns}) + (3 \times 35 \text{ ns}) = 280 \text{ ns}$ . This variable microcycle timing, a dynamic altering of the state sequence in a microprogrammed control processor, provides a real performance advantage and represents a departure from the simpler philosophy of defining the microinstruction states to accommodate the worst-case event.

### Asynchronous Memory Operation

A third major contribution to E-Series performance is improved memory/CPU efficiency. Memory is treated asynchronously.

The basic mechanism for accessing main memory requires that the address be set and a READ command be issued to the memory system. Then, in a subsequent microinstruction, the results are retrieved using a TAB (T-register to A or B) command. A typical sequence to read a word might be:

```
(1) READ COV PASS M S1 (SET ADDR, ISSUE READ)
(2) INC S1 S1
(3) JMP CNDX OVFL HALT (CNDX NOT MET)
(4) PASS L TAB (RETRIEVE MEMORY DATA)
```

The READ operation is decoded during the execution

Period	Microprocessor Control State
Previous P5	
P1	Load Microinstruction Register
P2, P3	Decode Word Type, Detect T-Register Request
P3 : P3	Wait for Memory Operation to Complete
P4	Pass T-Register Data To L-Register (Latch)
P5	Store In L-Register
Next P1	

**Fig. 4.** Main memory is treated asynchronously in the E-Series. During a memory operation the microprocessor may pause in period P3 if necessary to wait for the memory operation to complete.

of line 1 and is issued to the memory controller at the beginning of the execution of line 2. Since neither line 2 nor line 3 makes another request to memory, and memory operations require an amount of time equivalent to that of two or three microinstructions, these lines are allowed to execute normally and without interruption. Line 4 requests that the transfer register (TAB), which contains the result of the READ operation, be passed to the latch register (L). If the memory system has not completed the read operation by the time line 4 is decoded, the control state sequence will pause in period P3 until the T-register becomes valid, as shown in Fig. 4.

The example could be changed so that lines 2 and 3 are omitted. Then the control sequence will pause for

Operation	Time Required		Percent Improvement
	E-Series	M-Series	
Integer Arithmetic	10 s	15 s	50%
Floating Point Arithmetic	12 s	21 s	75%
RTE Operation			
Initiate Request	0.122 s	0.189 s	70%
Interrupt Processing	0.224 s	0.341 s	52%
Task Scheduling	0.225 s	0.370 s	64%
Sine Calculation	228 $\mu$ s	516 $\mu$ s	126%

**Fig. 5.** Benchmark results show performance improvement of the 21MX E-Series Computers over the M-Series. In another benchmark study involving task scheduling, analog and digital data acquisition, floating point calculations, and swapping of disc- and memory-resident programs, E-Series performance/price was calculated to be 8.7 versus the M-Series' 5.4, using operations completed in a given time as a measure of performance.

several more periods waiting for the memory operation to complete. This is essentially wasted time if other processing could be done, so efficient E-series microprogramming means performing as much processing as possible in parallel with a memory operation. In either case, the memory operation is transparent to the microprogrammer. The read request may precede the T-register request by up to three microinstructions (as in the example) and the data is guaranteed to be valid even when the DCPC is active.

Implicit here is that the E-Series Computer can accommodate any speed memory system, leaving room for future performance or cost enhancements. Currently, using the standard memory system, memory is busy approximately 90% of the time, while the control processor is busy (i.e., not paused) more than 75% of the time in a typical machine instruction stream. Higher-speed memory could increase processor performance, and therefore overall performance, by over 20% and no modifications would be required to the CPU.

#### Overlapped Instruction Fetches

A fourth area where the performance was improved over the 21MX was in enhanced microprogramming features. Many of the special operations required to emulate the 21MX instructions have been supplemented with additional hardware to add parallelism and thereby increase the speed while reducing the amount of code required. For example, the four-instruction fetch routine of the 21MX,

Fields:	LABEL	OP	SPECIAL	ALU	STORE	S-BUS ENABLE
000	FETCH	READ	FTCH	INC	PNM	P
001			ION			
002			CLFL	PASS	IR	TAB
003		READ	JTAB	INC	CM	ADR

has been compressed to two lines:

Fields:	LABEL	OP	SPECIAL	ALU	STORE	S-BUS ENABLE
000	FETCH	READ	FTCH	PASS	IRCM	TAB
001			JTAB	INC	PNM	P

This routine takes the result (TAB) of a previously issued (by the previous instruction) read operation, places it in the instruction register (IR), and in the memory address register (CM), sets the base or current page register, initializes memory protect and several internal registers (FTCH), and starts a read on the operand address in M. Line 001 conditionally increments the program counter and the address register (except in special cases) and performs a jump (JTAB) to the control memory address specified by the IR. If an interrupt is pending, JTAB branches to the interrupt handling routine. Instructions that occur frequently in typical programs (memory reference, shift-rotate,




alter-skip) are mapped directly with JTAB. Others of the 166 supported instructions that occur less often undergo multiple decode steps (actually indexed control branches) to find the entry point of the routine, but this added time is a very small fraction of the time required to execute a typical job stream, providing a very inexpensive decode system.

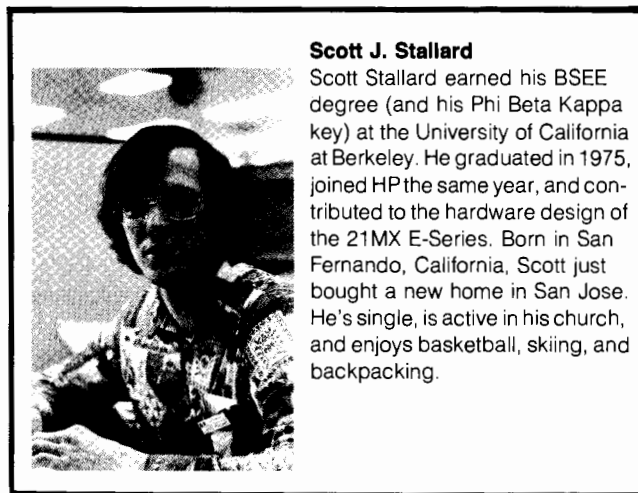
Floating point instructions have been augmented by a 48-bit shift normalization operation that repeats a single shift microinstruction until the combined A, B, and specified registers have been normalized. This operates much faster than the previous algorithm in which several conditional tests must be done in microcode.

Shown in Fig. 5 are the results of some benchmark studies that demonstrate the performance improvement of 21MX E-Series Computers over their predecessors, the M-Series.

### Multilevel Subroutines for the Control Processor

A subroutine save stack has been added to allow the microprogrammer to nest up to three levels of subroutines (compared to one in the 21MX). This provides for more modular and structured microprogramming than was previously possible. Utilities that are standard in the base set (indirect handler, interrupt processor, loader, etc.) may be invoked from

user-defined subroutines, making a user's routine easier to write and debug. 



### Scott J. Stallard

Scott Stallard earned his BSEE degree (and his Phi Beta Kappa key) at the University of California at Berkeley. He graduated in 1975, joined HP the same year, and contributed to the hardware design of the 21MX E-Series. Born in San Fernando, California, Scott just bought a new home in San Jose. He's single, is active in his church, and enjoys basketball, skiing, and backpacking.

### Reference:

1. P. Gordon and J.R. Jacobs, "Microprogrammable Central Processor Adapts Easily to Special User Needs," Hewlett-Packard Journal, October 1974.

## SPECIFICATIONS

### HP 21MX E-Series Computers

**CENTRAL PROCESSOR:** The central processor is microprogram controlled and is also microprogrammable. Microprogrammability fully software supported.

ADDRESS SPACE: 32,768 words; 1,048,567 words with DMS (optional).

WORD SIZE: 16 bits.

SYSTEM CYCLE TIME: 560 ± 35ns with HP 2102B memory system.

BASE SET INSTRUCTIONS: 128 standard instructions including index register instructions, bit, byte and word manipulation instructions, extended arithmetic instructions and high-speed floating point.

DATA REGISTERS: 2 accumulators, 2 index registers.

SELF TEST: Automatic tests of CPU and memory operating condition. Executed on cold power up and IBL/Test Switch.

INITIAL BINARY LOADERS: HP disc loaders and paper tape loader standard.

**CONTROL PROCESSOR:** Provides complete control of the central processor via microprograms. (HP supplied or user generated.)

ADDRESS SPACE: 16,384 words

WORD SIZE: 24 bits

CONTROL STORE CYCLE TIME: Variable (175ns or 280ns)

CONTROL PROCESSOR INSTRUCTIONS: 211

### INPUT/OUTPUT

INTERRUPT STRUCTURE: Multilevel vectored priority interrupt; priority determined by interrupt location.

I/O System Size	2109A	2113A
Standard I/O Channels	9	14
with one extender	25	30
with two extenders	41	46

### MEMORY SYSTEM HP 2102B

TYPE: 4K N-channel MOS semiconductor RAM

WORD SIZE: 16 bits plus parity bit

CONFIGURATION: Controller plus plug-in memory modules available in 8192 word and 16,384 word modules.

PAGE SIZE: 1024 words

**DIRECT MEMORY ACCESS (DCPC ACCESSORY):** Assignable to any two I/O channels. MAXIMUM TRANSFER BLOCK SIZE: 32,768 words.

TRANSFER RATE (in 10<sup>6</sup> words/s):

	Input	Output
without DMS	1	.92
with DMS	1	.86

### ENVIRONMENTAL

OPERATING TEMPERATURE: 0° to 55°C

STORAGE TEMPERATURE: -40° to 75°C

RELATIVE HUMIDITY (non-condensing): 20% to 95% at 40°C

ALTITUDE OPERATING: 4500 m (15,000 ft)

NON-OPERATING: 15,300 m (50,000 ft)

SHOCK: 30g for 11ms, 1/2 sine wave shape

VIBRATION: 0.30 mm (0.012 in) p-p. 10-55 Hz, 3-axis

### PHYSICAL

	2109A	2113A
HEIGHT:	222 mm (8 3/4 in)	317.5 mm (12 1/2 in)
WIDTH:	483 mm (19 in)	483 mm (19 in)
DEPTH:	597 mm (23 1/2 in)	597 mm (23 1/2 in)
WEIGHT:	20.5kg (45 lb)	29 kg (64 lb)

### POWER SUPPLY

LINE VOLTAGE: 110/220V ac (±20%)

FREQUENCY: 47.5 to 66 Hz

INPUT POWER MAX: 2109A, 525W; 2113A, 800W

### PRICES IN U.S.A.

2109A, \$5850.

2113A, \$6850.

2102B Controller, \$600; 8K, \$750; 16K, \$2100.

1K Writable Control Store, \$2000.

Dual Channel Port Controller, \$750.

Dynamic Mapping System, \$1950.

### MANUFACTURING DIVISION: DATA SYSTEMS DIVISION

11000 Wolfe Road  
Cupertino, California 94015 U.S.A.

# Microprogrammed Features of the 21MX E-Series

by Thomas A. Lane

**M**ICROPROGRAMMING IN THE 21MX E-Series processor makes it possible for the processor to emulate the 21MX instruction set, thereby maintaining instruction set compatibility with the 21MX family. Microprogramming is also used to implement standard features and user options that enhance user capability and convenience for minimum incremental cost.

The E-Series also supports user microprogramming, which allows the user to enhance performance by defining additional instructions tailored to his needs. HP offers both hardware and software support for user microprogramming (see reference 1 and article, page 15).

New microprogrammed features that are standard in the E-Series are a self-test firmware diagnostic, remote program load, a microprogrammable processor port, and microprogrammable block I/O.

## Self-Test Firmware Diagnostic

Vital to the operation of all computer systems is maintaining system integrity, or freedom from hardware failures. System integrity is verified by periodically running diagnostic programs on the system. A memory diagnostic is an essential element in this process.

In the E-Series, main memory is semiconductor

dynamic RAM which, although more reliable than magnetic core, still requires periodic testing and verification. Memory testability becomes progressively more important as memory size increases and represents a greater proportion of the system hardware. Thus there is a need for stand-alone diagnostics that can be easily used to detect and locate failures.

In the E-Series a set of memory diagnostics is resident in the base set microcode. The advantage of using microcoded diagnostics is the short run time resulting from the fast execution rate of microinstructions. This is important for memory diagnostics, whose execution times are often prohibitively long. Another advantage of microcoded diagnostics is that they can be ROM-resident. Therefore they do not require loading or configuring. Executing ROM-resident memory diagnostics also avoids the hazards of testing memory with a diagnostic resident in that memory.

The E-Series base set contains three separate diagnostic programs, Test 1, Test 2, and Test 3 (Fig. 1). Each test is initiated by a specific user action. The processor and memory are tested automatically during cold power-up by Test 1 and Test 3. Since the E-Series has standard parity error detection, Test 3 performs the additional function of initializing all present memory with the correct parity.

Test	Test Description	Event That Initiates Test Execution	Error Reporting
Test 1	Tests CPU Data Paths	Press IBL Switch or Cold Power-Up or INSTP STEP Machine Instruction 100000 <sub>6</sub>	All Operator Panel Display Register Bits and Display Indicator Bits On. OVFL Bit Is On.
Test 2	Tests All Present Memory (Up to 32K) with Read/Complement/Write Algorithm. Test is Non-Destructive to Memory Contents.	Press IBL Switch	All Operator Panel Display Register Bits and Display Indicator Bits On. OVFL Bit is Off. A = Good Data B = Bad Data M = Failure Address
Test 3	Tests All Installed Memory. The Test Algorithm Writes Data Patterns in a Known Background and then Verifies the Written Data. It then Restores the Background and Verifies It. The Test Is Destructive to Memory and Fills Memory with Contents of the S-Register.	Cold Power-Up or INSTP STEP Machine Instruction 100000 <sub>6</sub>	All Operator Panel Display Register Bits and Display Indicator Bits On. OVFL Bit Is Off. A = Good Data B = Bad Data M = Low-Order 15 Bits of Failing Address S(Low-Order 5 Bits) = High-Order 5 Bits of Failing Address

**Fig. 1.** 21MX E-Series base set microcode contains three self-test diagnostic programs. If a test reveals a failure the front-panel display indicates what happened.

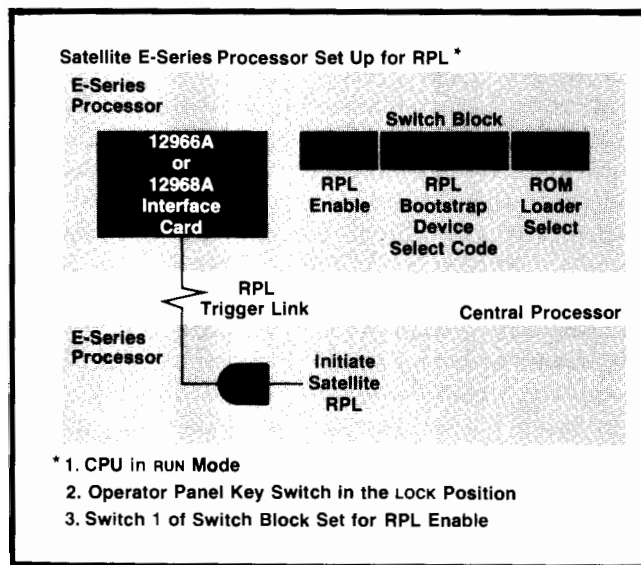
Pressing the IBL-TEST switch on the operator panel causes the processor and memory to be tested by Test 1 and Test 2. Thus a nondestructive memory test is done every time IBL is invoked.

The user can run Test 1 and Test 3 by single-cycling machine instruction 100000s with the INSTP STEP switch, and the key in OPERATE or LOCK. In the OPERATE position each test runs once. In LOCK, the tests loop until the key is returned to OPERATE or a failure is detected. Since Test 3 is a destructive memory test, it is imperative that its execution be prohibited in the RUN mode, because if 100000s is accidentally executed in a running program, it will destroy the memory contents and the program. For this reason, Test 3 will run only in the HALT mode and INSTP STEP must be used.

Whenever a set of tests is terminated the processor returns to the normal front-panel mode. If a failure has occurred, the front-panel display indicates the nature of the failure, and in the case of memory failures, helps the user locate and replace the failing part.

#### Remote Program Load

Remote program load (RPL) is a feature of the E-Series base set microcode that allows users to initiate an automatic bootstrap and run operation from either a local or a remote site. This operation consists of a complete bootstrap load operation from disc, communication line, or other specified device, followed automatically by its execution. Thus, RPL is useful in distributed processing systems where automatic startup must be initiated from a remote or unattended location.



**Fig. 2.** Remote program load is a feature of the E-Series base set microcode that makes it possible to initiate an automatic bootstrap and run operation from a local or remote site. Here a distributed-system central station triggers a satellite E-Series processor.

The normal method of bootstrapping an E-Series processor is to set specified information in the S-register and invoke the IBL function from the operator panel. The user sets the S-register to select one of four ROM loader programs resident in the E-Series processor and the select code of the I/O device from which the computer will be loaded. Pressing IBL causes the selected loader program to be read into memory and configured to the specified I/O device. Pressing RUN causes the loader program to be executed, resulting in a system load from the I/O device. Pressing RUN again initiates system operation. RPL can accomplish this task automatically and unattended.

The RPL process can be triggered in any of three ways: an I/O interface manipulating the processor RUN flip-flop, cold power-up of the processor, or execution of a HALT instruction. Any of these events can trigger RPL provided other conditions are met. The operator panel key must be in the LOCK position, and the configuration switch block located on the processor board must be properly set. Eight switches there are used to provide the I/O device select code and ROM loader selection previously provided by the operator. In addition, a switch that enables RPL must be in the enable position.

The RPL operation is controlled in the HALT code portion of the base set microcode. It should be noted that although the target machine being emulated is halted after executing a HALT instruction, the control processor, which executes microcode, is never halted so long as power is applied. The microcode determines whether RPL is desired, calls the IBL routine indicated by the configuration switches, and issues the run command to the processor. Special care must be used in systems that run with RPL enabled because the RPL process will be initiated by every HALT encountered.

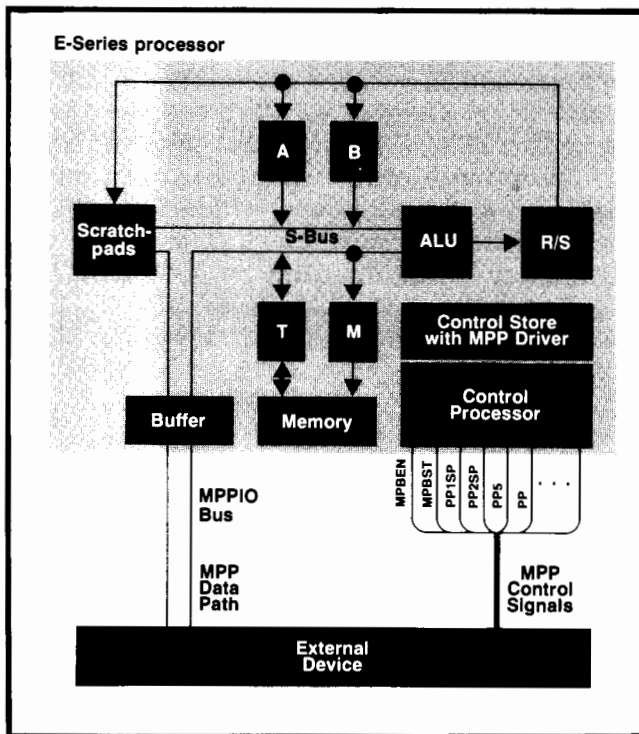
RPL can be used to create a system that starts operating automatically during power-on. This can be a convenience when a complex startup procedure is needed or when startup must be done by inexperienced people.

RPL can be used to control a remote satellite in distributed systems. The satellite is configured with RPL enabled and is triggered over a remote link through an I/O interface card in the satellite. A typical system is shown in Fig. 2.

#### Microprogrammable Processor Port

Standard in the E-Series processor is a new microcoded I/O method called the microprogrammable processor port (MPP). The MPP provides a high-bandwidth, direct data path between the E-Series processor and external devices.

In a bus-oriented, microprogrammed processor like



**Fig. 3.** The microprogrammable processor port (MPP) is a new high-bandwidth direct data path between the E-Series processor and an external device, such as a front-end processor, a special arithmetic processor, or another E-Series processor.

the E-Series, each data processing operation is done in a unit of time called a microcycle. During each microcycle, one microinstruction is executed. The microinstruction contains fields that specify the source of the data to be gated onto the S-bus, the ALU operation to be performed, and the data destination. Any device connected to the appropriate buses can be a data source or destination.

The MPP is based on an extension of this concept. The S-bus and appropriate control signals are made available to external devices to allow the external device to function as a source and/or destination in any microcycle. Thus the MPP gives the user the power to define generalized source and destination microorders for devices external to the processor.

The MPP consists of a buffered version of the main processor data bus (S-bus) called the MPPIO bus, plus additional signals needed to control the transfer. Fig. 3 shows how the MPP links the E-Series processor with an external device. Typical external devices are a front-end processor, a special arithmetic processor, or another E-Series processor.

A transfer from the external processor to the E-Series is done by the external processor's using MPBEN to gate data onto the MPPIO bus (and S-bus) for the entire microcycle. The processor writes the S-bus data into the specified destination at the end of the

cycle. For example, the following microinstruction would perform a transfer from the external processor into the A-register.

OP	SP	ALU	ST	S BUS
		PASS	A	MPBEN

A transfer from the E-Series to the external device is accomplished by the external device's using MPBST·P5 to strobe the data on the S-bus (and MPPIO bus) into a register. For example, the microinstruction for a transfer from the A register to the external device would be:

OP	SP	ALU	ST	S BUS
		PASS	MPBST	A

The two special field signals (PP1SP and PP2SP) are control lines that are activated by placing the corresponding microorder in the special field. The function of these signals is user-definable. Another signal, PP, is an input to the E-Series processor; it is a microcode branch condition and can be used to communicate the status of the external device.

The use of a microprogrammed I/O driver produces very high data transfer bandwidths. Shown below are typical MPP transfer rates obtainable in various applications.

Type of Transfer	Data Rate
Burst Mode	5.7M words/s
Synchronous	1.5M words/s
Asynchronous, Interruptible	0.75M words/s

### Microprogrammable Block I/O

Microprogrammable block input/output (MBIO) is another new feature of the E-Series processor. It provides a microprogrammable data path between the processor and the I/O bus, and accomplishes data transfers between the I/O bus and the processor in a single microcycle. MBIO can provide a high-bandwidth data path between the processor and an external device or between two processors. MBIO is implemented by means of a microcoded driver in conjunction with the appropriate I/O interface.

Standard I/O transfers are initiated by a microorder, IOG special, that causes the processor to synchronize to input/output period T2 and then perform the indicated I/O operation during T3, T4, and T5.\* The MBIO feature eliminates the use of IOG and therefore allows I/O transfers to occur in any T-period, totally asynchronous with respect to T-period timing.

MBIO requires three new signals in the I/O

\*Each I/O operation takes one I/O cycle, which consists of five T-periods, T2 through T6. Each T-period is equal to one microcycle, which consists of a number of P-periods ( $\geq 5$ ).

backplane of the E-Series processor. BIOI is activated by IOI in the SBUS field of a microinstruction, providing there has been no IOG special in the previous three microcycles. The signal is active for the entire microcycle and can be used by the MBIO interface to get data onto the I/O bus. In addition, IOI creates a data path from the I/O bus to the S-bus. BIOO is activated by IOO in the STORE field of a microinstruction, providing there has been no IOG special in the previous three microcycles. The signal is active for the entire microcycle and can be used by the MBIO interface as a qualifier to store data from the I/O bus. In addition, BIOO creates a data path from the S-bus to the I/O bus. BIOS is a timing signal that is used to synchronize the processor and the MBIO interface during a transfer. BIOS is active during processor period P3.

MBIO transfers are accomplished on the I/O bus by manipulating these three control signals. The interconnection between the E-Series processor and an MBIO interface is illustrated in Fig. 4.

The following microinstructions perform transfers between the MBIO interface and the E-Series processor, providing IOG special has not occurred in the

previous three microcycles.


OP	SP	ALU	JT	J BUS
		PASS	A	IOI

The following example transfers data from the MBIO interface to the A-register.

OP	SP	ALU	ST	S BUS
		PASS	IOO	A

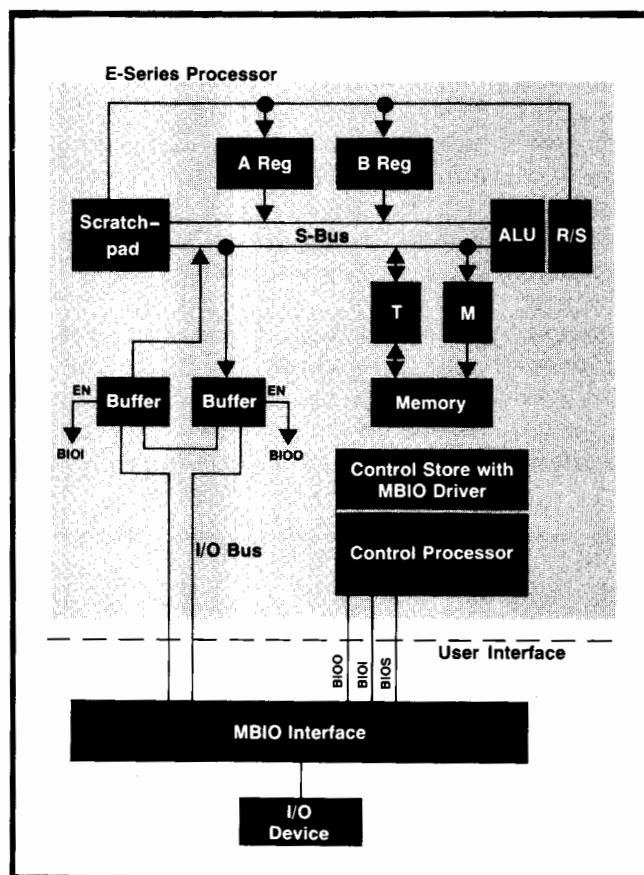
When many MBIO interfaces are connected to the I/O bus, a specific interface can be addressed through its select code. This is done simply by loading the select code of the desired interface into the instruction register.

There are many additional signals in the I/O backplane that can be given user-definable functions on the MBIO interface card and can be manipulated by simulating the corresponding I/O instruction in microcode. SKPF, another signal in the I/O backplane, is wire-ORed to every I/O card slot. When a card is selected by having its select code in the instruction register, it can enable the status of its flag onto the SKPF line. Thus SKPF can be used to examine the status of the selected card. SKPF is very useful in MBIO applications because it is a microcode branch condition and can be used to test status.

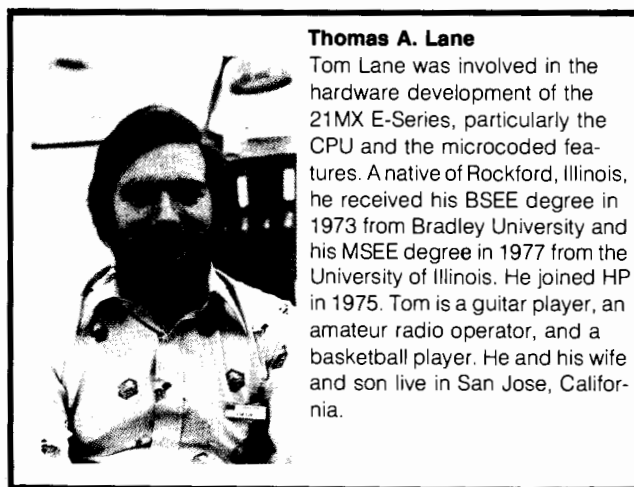
An MBIO interface resides in the I/O backplane, enabling it to use the powerful interrupt system of the 21MX family. This gives MBIO devices the ability to communicate through the interrupt system. 

#### Reference

1. F.F. Coury, "Microprogramming and Writable Control Store," Hewlett-Packard Journal, July 1972.



**Fig. 4.** Microprogrammable block I/O is a new feature that provides a microprogrammable data path between the E-Series processor and the I/O bus. Data transfers between I/O bus and processor take only a single microcycle.



# OPNODE: Interactive Linear Circuit Design and Optimization

*OPNODE is a powerful software package for computer-aided circuit design with an interactive graphics console in a minicomputer environment.*

by William A. Rytand

ONE OF THE ELECTRICAL engineer's dreams has been a design system whereby he can quickly sketch a circuit and instantaneously view its performance on a large, accurate display. Although this dream is not yet totally a practical reality, the system described in this article comes close to it. The system is called OPNODE (the name emphasizes the key words "Optimization" and "NODE-oriented circuit description"). For engineers designing linear circuits using frequency-domain techniques, OPNODE offers two major advantages that previously have not been generally available: rapid user interaction and the flexibility of the BASIC language.

User interaction is rapid because OPNODE runs on a dedicated minicomputer system, taking full advantage of the interactive graphics features of the HP 8500A System Console.<sup>1</sup> This console is the user interface for either the 8542B Automatic Network Analyzer<sup>2</sup> or the 8580B Automatic Spectrum Analyzer.<sup>3</sup> It is capable of generating high-quality graphic displays in real time, allowing the user to gain insight into circuit performance much more rapidly than with tabular data. Another level of interaction is achieved when using OPNODE with the Automatic Network Analyzer; accurate measurements may be made on microwave devices and automatically analyzed in circuit models to predict overall performance.

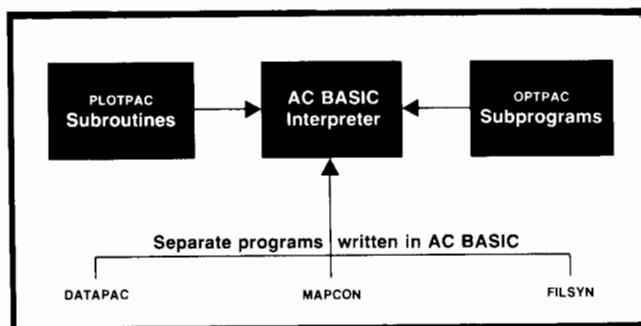
The flexibility of the OPNODE package is due largely to AC BASIC, a superset of the BASIC language with over 100 additional statements and commands that make it possible to model, analyze, and optimize linear networks as large as 40 nodes and 200 components. Using AC BASIC as a foundation, additional circuit-design building blocks are also provided (Fig. 1): PLOTAC for plotting data in various formats (including semilog and Smith charts), OPTAC for optimizing microwave circuits, DATAPAC containing data on HP microwave transistors, MAPCON for converting measured data from the 8542B Automatic Network Analyzer, and FILSYN for synthesizing various filter structures. The user is free to expand the OPNODE package to suit his or her unique application, since all of the building blocks are written in

AC BASIC and may easily be modified or expanded. Also included in the OPNODE package is a tutorial manual with ten examples, a pocket guide, and five hours of videotapes for training.

## Analysis Technique

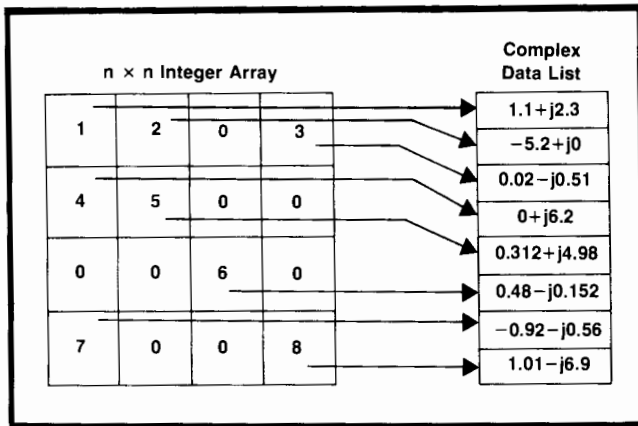
The analysis technique used in OPNODE is a sparse Y matrix algorithm, in which the zeros in the Y matrix are not stored as data. This technique offers several advantages over other methods. It lends itself easily to a simple component-by-component nodal description of the circuit, it is computationally efficient, it requires less data storage, and the algorithm is compact, occupying less than 800 words of memory.

At any particular analysis frequency, the algorithm begins by initializing an  $n \times n$  integer array to zero, where  $n$  is the number of nodes in the circuit. The admittance of each component in the circuit is then calculated and added to the overall Y matrix of the network. There are two possibilities for a new admittance being added to the  $(i,j)$ th element: either this is the first entry in this position, or it is not. If it is the first entry, the complex admittance (requiring four 16-bit words) is placed in the next available position in a one-dimensional data list, and its location in this data list is placed in the  $(i,j)$ th position of the  $n \times n$



**Fig. 1.** OPNODE is a software package, written in AC BASIC, that runs on HP automatic network and spectrum analyzer systems. It can model, analyze, and optimize linear networks that have up to 40 nodes and 200 components. The OPNODE package includes all of the software modules shown here.





**Fig. 2.** OPNODE uses a sparse Y matrix algorithm, in which the zeros in the Y (admittance) matrix are not stored as data. The technique is efficient and fast. A 40-node ladder network is solved at a rate of about one frequency per second.

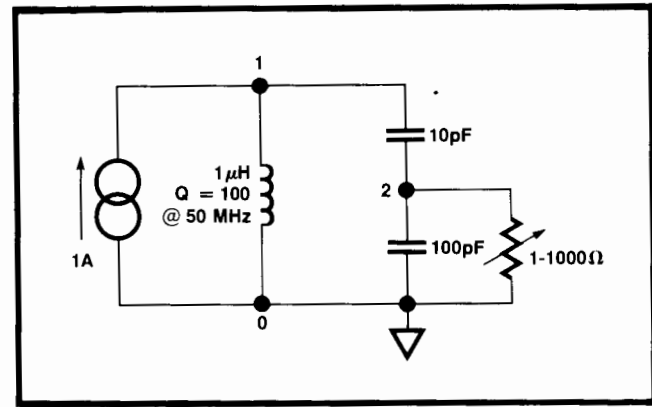
integer array (which was previously zero). For subsequent additions to this same (i,j)th element, the location of the data is determined by the  $n \times n$  integer array and the new admittance is then added to the previous data contained in the data list. Fig. 2 illustrates this relationship. Once the complete Y matrix has been formed, a conventional lower/upper decomposition technique<sup>4</sup> is used to solve for all of the node voltages in the circuit.

This sparse Y matrix approach yields two advantages: it requires less memory to store the matrix, and it is faster than a non-sparse approach. For example, a 30-node ladder network would require  $4 \times 30^2$ , or 3600 words, if its complete Y matrix were stored. Using the sparse matrix approach, only  $30^2 + 4 \times 3 \times 30$ , or 1260 words, are required.

The speed of the sparse matrix approach is also superior: a 40-node ladder network is solved at a rate of about one frequency per second, and the time is approximately a linear function of network size. Algorithms that do not take advantage of matrix sparsity require times that increase as the cube of the number of nodes.

### Optimization Technique

OPNODE can automatically adjust up to ten component values to improve the circuit's performance. The technique used is a random adaptive search. The first step in the optimization process is a pair of trials, the first a random move from the starting point, and the second in the opposite direction. Depending upon the success or failure of these two trials, a steering matrix is updated. This matrix is then used to steer future random trials away from failures and towards successes. The user can observe the progress of the optimization and even modify the rate of learning dynamically by the use of switch options. This interaction with the optimization process allows the



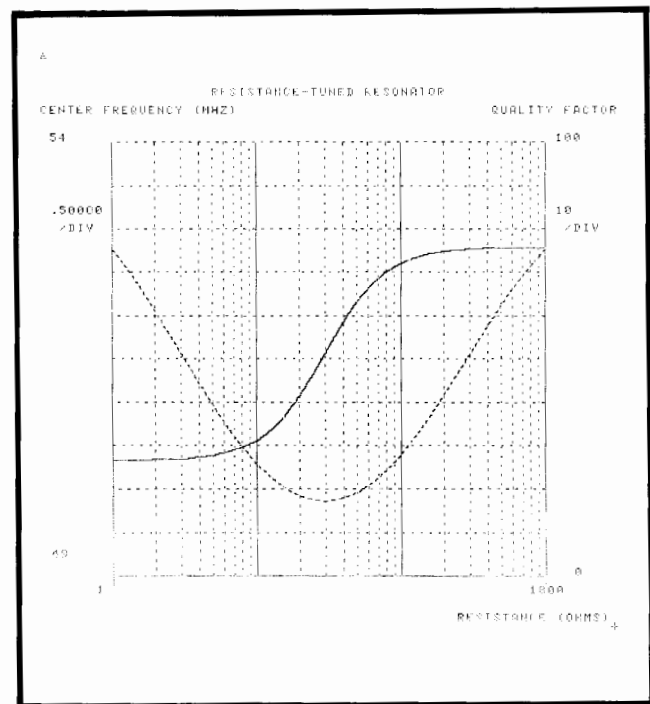
**Fig. 3.** A simple tuned resonator circuit. OPNODE took about ten seconds to plot the resonant frequency and Q of this circuit as functions of the tuning resistor (see Fig. 4).

designer to develop a feeling for how well it is doing, thus allowing him to make topological changes if he perceives that the optimization is in some sense "stuck".

Since the computer system is a dedicated one, users frequently allow optimization runs to go all night. A power-fail feature automatically continues a program when power is restored after a failure.

### Example of Flexibility

As an example of the flexibility of OPNODE, consider the simple tuned resonator circuit shown in Fig. 3.



**Fig. 4.** Hardcopy version of the graphics-console display showing the resonant frequency and Q of the resistance-tuned resonator of Fig. 3.

```

1000 KPHLSET ..... Initialize Units
2000 ICS(V,1,1,A)
2010 L(1,0,1)
2020 U(100,50)
2030 C(1,2,10)
2040 C(2,0,100)
2050 R(2,0,1)
3000 DIM F(31),Q(31),H(31)
3010 XLOG
3020 LABEL VTAB(2);TAB(19);"RESISTANCE-TUNED RESONATOR"
3030 LABEL VTAB(3);"CENTER FREQUENCY (MHZ)"
3040 LABEL VTAB(31);TAB(46);"RESISTANCE (OHMS)"
3050 HGH10(1,1000,50,55) ..... Draw Semilog Grid
3060 FOR J=1 TO 31
3070 LOGSTP(1,1000,31,R1)
3080 CHANGE(5,1,R1) ..... Modify Resistor
3090 LET V0=0
3100 FOR J=1 TO 500
3110 LINSTP(50,55,500,J,F)
3120 SOLVE(F)
3130 IF VMAG(1) <= V0 THEN 3170 } ..... Loop to Find Resonance
3140 LET V0=VMAG(1)
3150 LET F1=F
3160 NEXT J
3170 PLOT(R1,F1,3) ..... Plot New Resonance
3180 LET H(1)=R1
3190 LET F(1)=F1
3200 LET Q(1)=VREA(1)/(P(2)*F) } ..... Save Results
3210 LET F0=F1
3220 NEXT I
3230 MVTPT(1,1)
3240 CLFAP(3) ..... Clear Old Plot
3250 LABEL VTAB(3);TAB(50);"QUALITY FACTOR"
3260 WDRAW(H(1),F(1),Q(1),-N) ..... Autoscaled Plot

```

Fig. 5. The complete AC BASIC program to calculate and plot the curves of Fig. 4.

The analysis problem is to plot the resonant frequency and Q of this circuit as the tuning resistor is varied from 1 to 1000 ohms. Most computer-aided design programs would require the user to input the resistor value, compute the frequency response of the network, and manually pick out the resonant frequency, all this to get one point on the desired output plot. An AC BASIC program to perform this analysis contains two loops: an inner loop to search for the resonant frequency, and an outer loop to change the tuning resistor. Fig. 4 shows the results of this analysis. This plot was generated in approximately ten seconds.

A complete AC BASIC program to calculate and

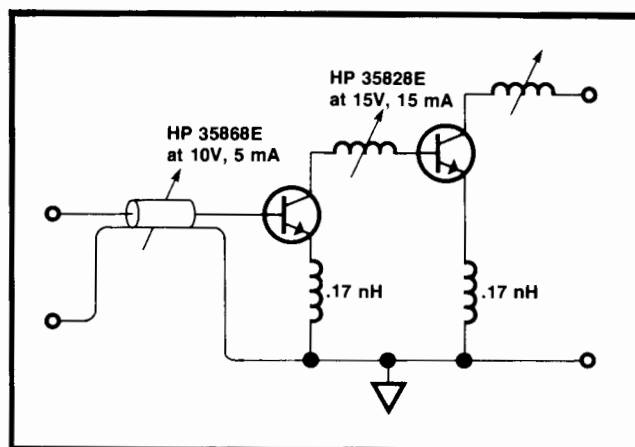


Fig. 6. A typical two-stage microwave amplifier. The design is to be optimized to meet certain gain, bandwidth, and VSWR specifications.

plot these curves required less than 35 statements, including labeling the axes (see Fig. 5).

### Amplifier Optimization Example

Fig. 6 shows a typical two-stage microwave amplifier. The design goal is to achieve a flat gain of  $23 \pm 1$  dB from 2.0 to 2.3 GHz, while simultaneously maintaining VSWRs better than 1.22 at the input and output ports. The initial circuit performance is shown in Fig. 7, along with the improved performance after about ten minutes of optimization. The rate of improvement suggests that the topology is not yet correct, so the circuit is quickly modified to that shown in Fig. 8, and the optimization is repeated. This time, after ten minutes of optimization the design goals are achieved as shown in Fig. 9.

### Acknowledgments

Special thanks go to Luiz Peregrino, who worked

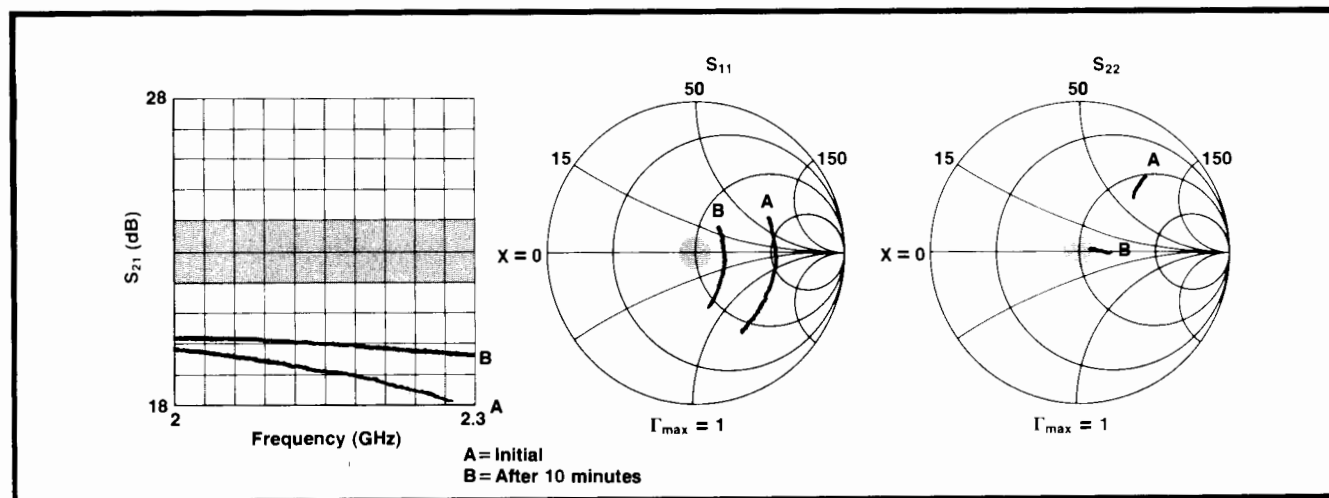
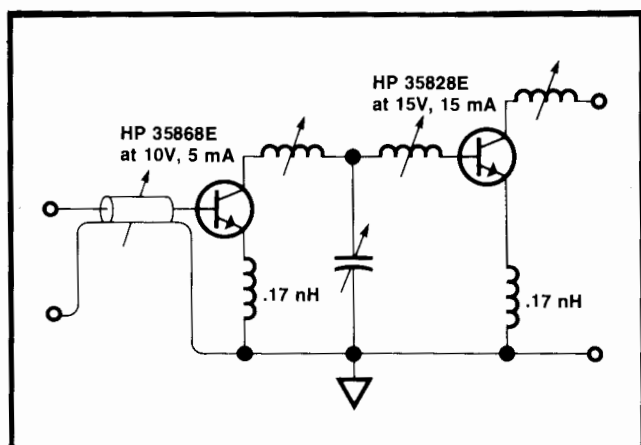


Fig. 7. Interactive output displays of the performance of the original design of the circuit of Fig. 6 and the performance after ten minutes of optimization. Design goals are shown by shaded areas.



**Fig. 8.** Rate of improvement in Fig. 7 suggests that the circuit of Fig. 6 should be modified. The modified design is shown here.

long and hard on several of the programs that preceded OPNODE. The design engineers who used the early versions of OPNODE are also due thanks for their patience. Diane Gonzalez-Donahue gave invaluable help in preparing the video tape, and Joe Williams contributed much to the structure and content of the user's manual. Ron Carelli contributed much guidance throughout the project.

## References

1. K.A. Fox, M.P. Pasturel, and P.S. Showman, "A Human



## William A. Rytand

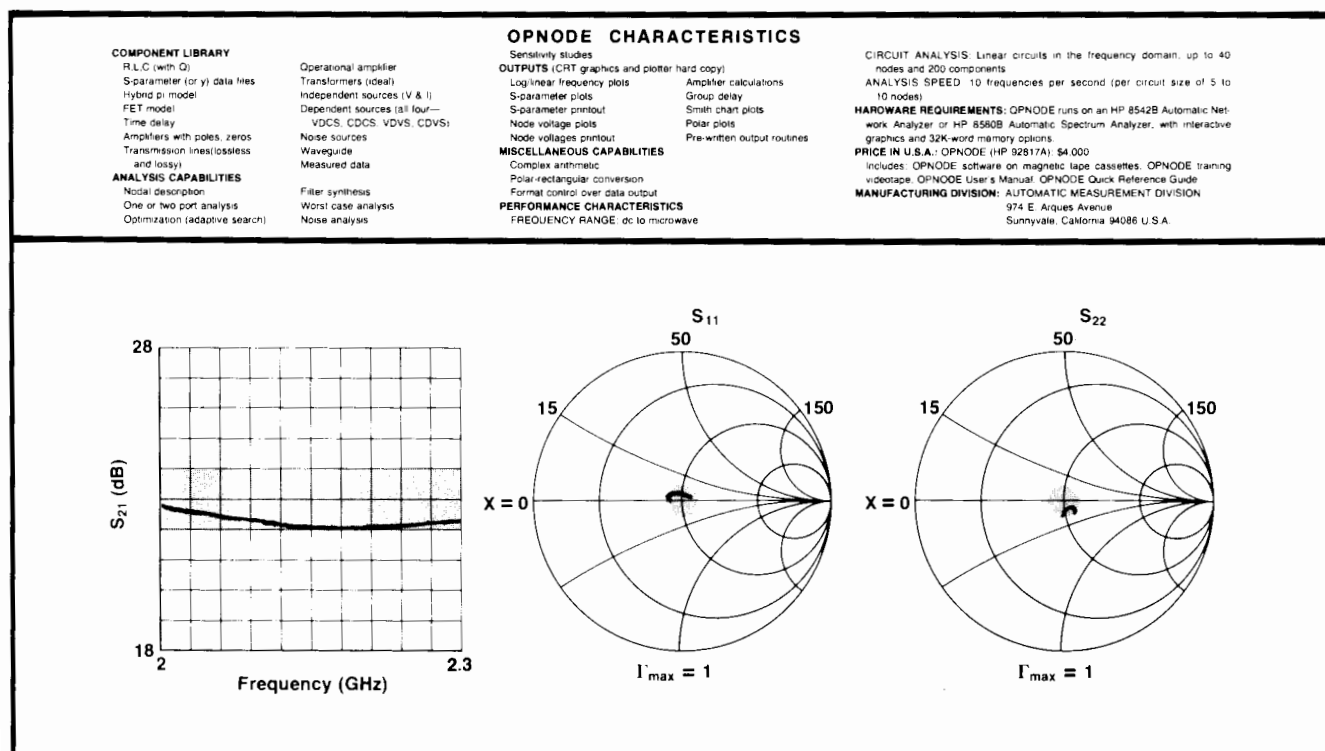
Bill Rytand is project engineer for digital test systems at HP's Automatic Measurement Division. With HP since 1965, he's served as project engineer for the 8414A Polar Display, project manager for the 8407A Network Analyzer, and section manager for automatic network analyzers. He's a member of IEEE. He received his BSEE degree from Massachusetts Institute of Technology in 1965 and his MSEE degree from Stanford University in 1967. Born in San Francisco, Bill now lives in Menlo Park, California. His favorite sports are handball and squash, but he also plays racketball and tennis, learned to ski last winter, and owns part of a Santana 22 sailboat. Other major interests are his two daughters, a van that he's converting into a camper, and playing backgammon.

Interface for Automatic Measurement Systems," Hewlett-Packard Journal, April 1972.

2. D.K. Rytting and S.N. Sanders, "A System for Automatic Network Analysis," Hewlett-Packard Journal, February 1970.

3. M. Cunningham and L. Wheelwright, "Introducing the Automatic Spectrum Analyzer," Hewlett-Packard Journal, February 1972.

4. G.E. Forsythe and C.B. Moler, "Computer Solution of Linear Algebraic Systems," Prentice-Hall, 1967, pp. 58-60.



**Fig. 9.** After ten minutes of optimization, the circuit of Fig. 8 meets the specifications. Final component values can also be displayed.

# Viewpoints

## John Moll on HP's Integrated Circuit Technology

As integrated circuits are adapted to a rapidly growing range of applications, equipment manufacturers need to be concerned with the technology involved in building the circuits they use. At Hewlett-Packard, silicon has become—and seems destined to remain—the basis for building 90 to 95% of the circuits used here. Thus, the major emphasis in our IC labs is on circuit design and process technology.

Within silicon circuit technology there exists a large number of possible approaches to any given electronic function. Possible choices include CMOS in bulk substrate, CMOS on sapphire, I<sup>2</sup>L (bipolar injection logic), n-channel MOS, p-channel MOS, CCD (charge-coupled devices) for certain signal and memory application, and ECL (emitter-coupled logic) or EFL (emitter-follower logic) bipolar circuits for speed. Within each technology there are further choices, such as silicon gate or metal gate in the MOS process, and junction isolation or oxide isolation in the bipolar circuits. Some technological limitations as well as the basic circuit requirements and performance goals help determine appropriate choices. Because of the wide range of products manufactured by HP, a performance goal may be very low power consumption at moderate frequencies or it may be highest possible frequency at whatever power it takes. Each of our labs makes choices according to the kinds of products that will use their devices.

A close link between the engineering lab and the IC lab is proving to be a successful mode of operation at Hewlett-Packard. The systems designers in the engineering lab supply the product goals and the circuit designs, and the IC lab provides the new device and circuit technology. This has resulted in some significant devices.

Some speculation on future directions may be of interest. We presently build digital circuits containing 20,000 to 30,000 active devices within a 5×5-mm area with relative ease. We would like to put many more devices—perhaps a few hundred thousand—on a single chip because this would result in fewer packages for any given system, and fewer packages means higher reliability, as well as lower cost.

The most serious limitations on extending size and complexity are the processes related to photolithography. The accurate reproduction of photomasks, the mechanical process of positioning succes-

sive photomasks, the precision of optical lithography, and the etching process are all being tested throughout the industry at the present time. Advances in any one technology can bring some small benefits but significant benefits can come only from an across-the-board advance.

Wet-acid etching, until now the favored etching technique, undercuts the masked area by 0.5 to 1.5  $\mu\text{m}$ . This has contributed towards restricting the minimum line dimension to 5  $\mu\text{m}$  for the larger MOS circuits, though some bipolar circuits have emitters 2 or 3  $\mu\text{m}$  wide and some discrete devices achieve 1- $\mu\text{m}$  minimum dimensions using special tricks.

Dry etching technologies that have minimal and well-controlled undercutting have been demonstrated. These involve plasma or sputter etching, or molecular-beam etching. Unfortunately the selective-etching capabilities of those technologies have not been well developed. A major advantage of wet-etching technology is that mixtures are available to etch silicon dioxide without etching elemental silicon, silicon nitride or aluminum, or to etch aluminum without etching silicon, silicon oxide or silicon nitride, and so on. Proper choices of materials and etching sequences allow patterns to be formed in any of these commonly-used conductors and dielectrics without disturbing the other three. To take advantage of the dimensional-control potential of dry etching, gas mixtures that etch silicon oxide and aluminum selectively need to be found. Presently a carbon-tetrafluoride plasma, easily masked by the commonly used photoresists, etches silicon and silicon nitride much faster than either silicon dioxide or aluminum but it requires care because it may damage the underlying silicon. Although progress in dry etching is being made, there is still much to be done.

Another discernible direction of change involves the use of electron beams (e-beams) for mask generation. A number of companies are beginning to use the "fine-cut" capabilities of e-beam lithography to directly fabricate photo masks that are then used in optical lithography. One source of error—i.e., the optical reduction of the masks—is eliminated but the mechanical and other processing tolerances associated with lithography remain.

(continued on page 17)

Hewlett-Packard Company, 1501 Page Mill  
Road, Palo Alto, California 94304

HEWLETT-PACKARD JOURNAL

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

CHANGE OF ADDRESS

To change your address or delete your name from our mailing list please send us your old address label (it peels off). Send changes to Hewlett-Packard Journal, 1501 Page Mill Road, Palo Alto, California 94304 U.S.A. Allow 60 days.