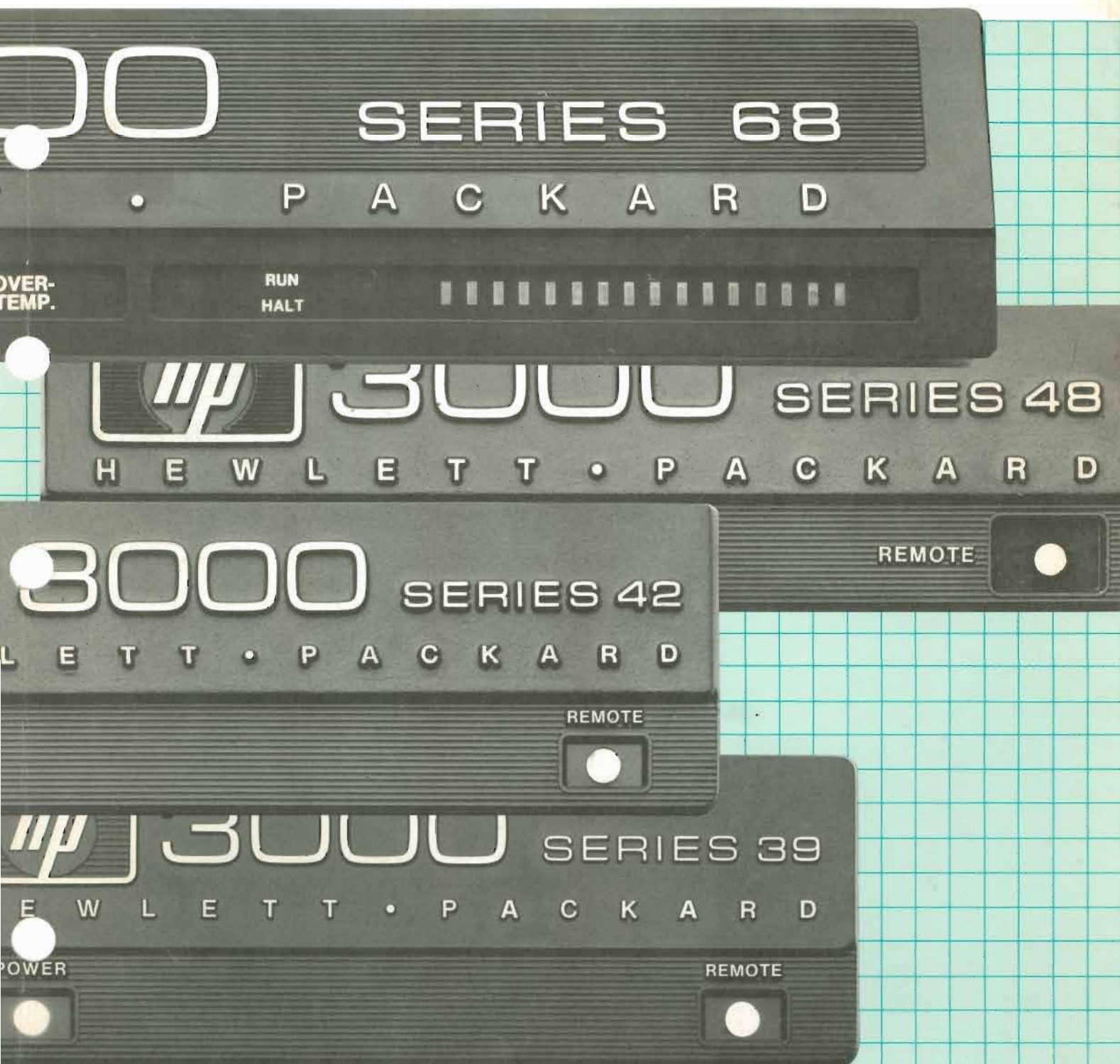


HP 3000 Computer Systems



General Information Manual

Effective August, 1983



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

HP 3000 Computer Systems General Information Manual



19447 Pruneridge Avenue
Cupertino, California 95014

Notice

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

Copyright © 1983 by HEWLETT-PACKARD COMPANY

Preface

Today's business decisions cannot be based on yesterday's data. Immediate access to the most up-to-date information is a necessity for the financial planning, sales forecasting, production scheduling, and other complex tasks which are part of each business day. The HP 3000 interactive business computer systems are uniquely qualified to meet these demands for accurate, timely information. Designed specifically for terminal-oriented business data processing, the HP 3000 systems are based on an integrated hardware and software concept. HP 3000 computer systems are delivered ready to run application programs. Complete data base management inquiry facilities are included to make the basic HP 3000 system a fully capable and operational system.

Several office systems software packages are available to HP 3000 users. For general office correspondence and reports a versatile interactive, secretarial word processor is available. Two other packages give basic word processing features for casual users, and extensive formatting capabilities for documentation and lengthier documents. A desk management function combines electronic mail, an electronic calendar/diary, personal filing, basic word processing and personal computing. Business graphics packages allow users to interactively design and produce text slides and business graphs and charts, which may be integrated into the above mentioned word/text processors. Merging text and graphics on the same printed page, the HP 2680 Laser Printing System may also be used for electronic forms management with logos, forms, and even signatures at a CRT, utilizing the interactive design and formatting systems.

For manufacturers, Hewlett-Packard offers user-customizable materials planning and control software. Ten functionally integrated modules are available to address all of the important phases of materials planning and control. Another product consisting of six application modules, provides a user customizable, interactive system for managing the production planning and control function of a manufacturing operation. These inter-

active, data base oriented HP products can help you improve inventory management, control costs, and obtain more timely and accurate information on which to base purchasing and manufacturing decisions.

In the area of wholesale distribution, HP provides complete solutions with two application software products. One of these application packages consists of seven integrated modules designed to handle order processing, inventory management, and associated accounting functions in the whole distribution environment. The other application package contains four modules and is geared specifically toward the sales function.

HP offers a full range of financial management application software that is flexible enough to be used in virtually any environment. HP Financial Accounting is an on-line, interactive, totally integrated software package comprised of eight integrated modules.

For in-house software development, Hewlett-Packard provides tools for efficient interactive program development. With six programming languages, debugging aids, programming utilities, and high level procedures for data base management, data entry, data inquiry and other programming tasks, programmer productivity on the HP 3000 is well above the industry average.

Advanced distributed processing capabilities through a wide variety of data communications products allow HP 3000 systems to bring information directly to the people who need it, when they need it.

- This manual contains a thorough discussion of the HP 3000 operating system and system architecture. The appendices outline the various operating system and machine level commands and intrinsics, and expand on the hardware features of the computer systems. Other HP 3000 General Information Manuals cover the HP 3000 transaction processing environment, office systems software, data communications, and manufacturing applications systems.

Contents

Chapter 1. System Introduction

Management Overview	1-1
System Components	1-3
Fundamental Operating Software	
MPE Operating System	1-4
Data Base Management	1-5
Utilities	1-5
Additional Software Subsystems	
Languages	1-5
Productivity Aids	1-6
Performance Measurement Software	1-6
HP Distributed Systems Network (DSN) Software ..	1-6
Manufacturing Applications Software	1-7
Financial Application Software	1-11
Wholesale Distribution Software	1-13
Office Systems Software	1-13
Transaction Processing Tools	1-15
HP PLUS	1-15
Compatibility	1-15
Documentation	1-16
Training	1-17
Software Support Services	1-17
Hardware Support Services	1-17
Consulting Services	1-18
HP 3000 International Users Group	1-19

Chapter 2. System Software

MPE Operating System	2-1
MPE V Features	2-2
Efficient Versatility	2-3
A User Oriented Operating System	2-5
User Interface	2-6
Program Development	2-9
A Dynamic Environment	2-13
System Operation	2-15
Performance Measurement	
On-line Performance Tool/3000	2-19
Application Program Sampler (APS)/3000	2-20
Flexible Disccopy	
Flexible Disccopy/3000	2-21

Chapter 3. System Architecture

Stack Architecture	3-1
Separation of Code and Data	3-1
Processes	3-2
Variable-length Segmentation	3-2
Code Segmentation	3-3
Data Stack	3-3
Registers	3-5
Virtual Memory	3-5
Microprocessor	3-5
Microcode	3-5
Instructions	3-6

Appendices

A. Documentation	A-1
B. MPE Commands	B-1
C. MPE Intrinsics	C-1
D. HP 3000 Series 68 Hardware Features	D-1
E. HP 3000 Series 68 Machine Instructions	E-1
F. HP 7000 Series 48 Hardware Features	F-1
G. HP 3000 Series 39 and 42 Hardware Features	G-1
H. HP 3000 Series 39, 42 and 48 Machine Instructions	H-1

System Introduction

Management Overview

System Components

**Fundamental Operating
Software**

MPE Operating System

Data Base Management

Utilities

Languages

Productivity Aids

Performance Measurement Software

Data Communications

Manufacturing Application Software

Financial Application Software

Wholesale Distribution Software

Office Systems Software

Transaction Processing Tools

HP Plus

Compatibility

Documentation

Training

Software Support Services

Hardware Support Services

Consulting Services

HP 3000 International Users Group

Management Overview

HP 3000 computer systems are a compatible family of interactive, data-base oriented business processing systems. They have an integrated hardware and software design ideally suited to multi-purpose business data processing and dedicated applications. Compatibility across the HP 3000 product line insures upgradability and an easy growth path to meet expanding EDP requirements. With a variety of programming productivity tools and many ready-to-run application software packages, the HP 3000 is very easy to use. The advanced design of the HP 3000 provides a wide range of user-oriented capabilities that can be utilized with equal ease on both stand-alone systems and those in distributed processing networks.

HP 3000 systems can be configured in three different ways: as a ready-to-run production system, as a program development system, and as an integrated element of a distributed data processing network (see Figure 1-1).

Ready-To-Run Production Systems

HP 3000 computers are delivered as complete systems with all the hardware and software necessary to run and operate applications programs immediately. In addition to hardware, every HP 3000 computer system includes Fundamental Operating Software (FOS) which makes it a completely functional production system. Included in FOS is the Multi-Programming Executive operating system (MPE) that schedules, monitors, and controls processing, and a complete set of software subsystems for data management, data entry and other capabilities needed in business processing applications. The Fundamental Operating Software enables you to immediately run application programs developed and compiled on any HP 3000 without installing any additional software on the executing system other than the program itself. One HP 3000 can satisfy the program development needs of an entire network of HP 3000 Computer Systems.

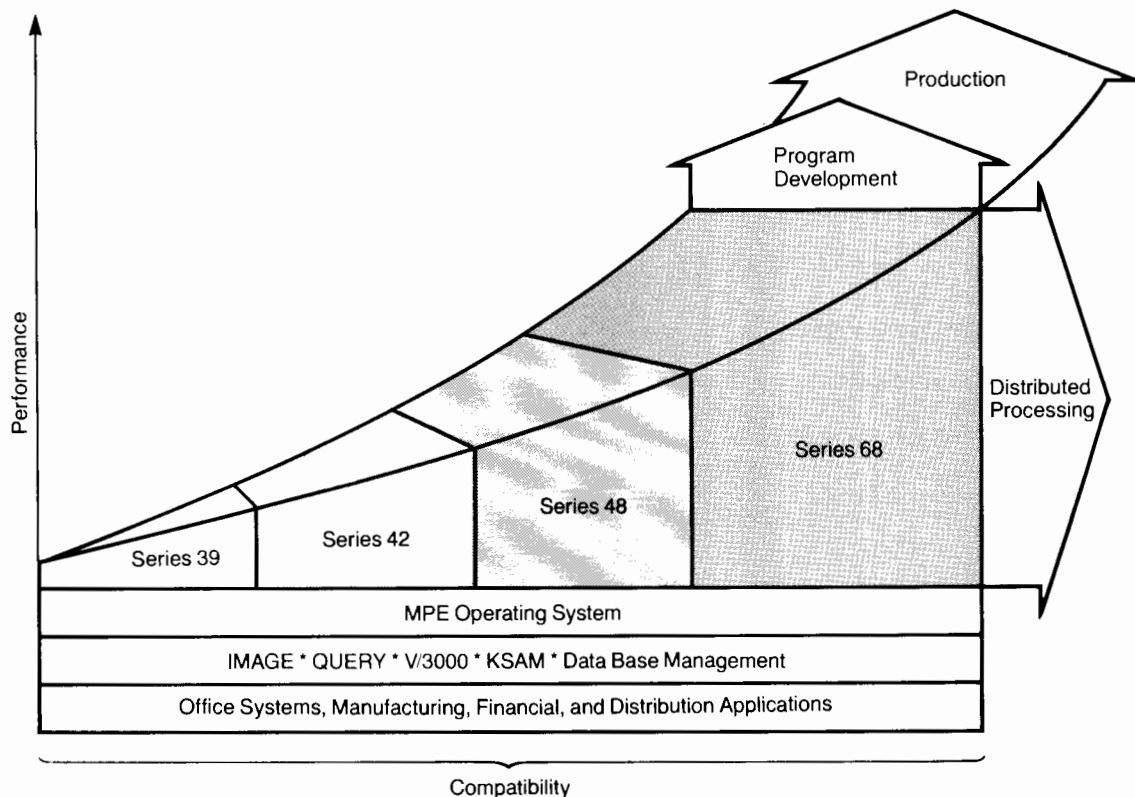


Figure 1-1 Each HP 3000 computer system can be configured to provide the exact mix of capabilities you require along three functional dimensions: Production, Distributed Data Processing, and Program Development.

Program Development Systems

Additional software subsystems such as language compilers and program development tools, complement the Fundamental Operating Software. A wide range of ANSI standard programming languages are available. The HPToolset program development system brings together an integrated set of programming tools, including a full-screen editor, source version manager and symbolic debug for COBOL program development. The subsystems of FOS include a wide range of high-level procedures for data management and data entry that are callable from any of the HP 3000 programming languages. Together these subsystems give programmers comprehensive capabilities and makes program development on the HP 3000 quick and convenient.

Distributed Data Processing Systems

Data communication subsystems are also available to enable you to set up distributed processing networks that conform to the way you want to do business. A variety of terminal communications products allow you to place terminals wherever you wish without regard to distance or location. HP also offers software that allows you to connect workstations to your HP 3000 computer. HP Distributed Systems software permits sharing of resources between HP 3000 computer systems and other Hewlett-Packard computers. HP 3000 systems can also provide batch and interactive access to IBM mainframe computers. These capabilities are supplied as high level services which free the programmer from complex tasks and allow you to place the processing power where you need it—regardless of the level of expertise at the remote site.

Combined Function Systems

The above three HP 3000 configurations are by no means mutually exclusive. All HP 3000 computer systems can simultaneously perform transaction processing operations, interactive program development, word processing, batch processing, and data communications.

You can distribute your processing according to functional needs or geographical location while retaining the degree of control and security you desire. At each location, you can specify the function and capability you want your HP 3000 to have: a single dedicated application, multiple applications, program development or a combined general use system.

Applications

The HP 3000 family meets the EDP requirements for a broad range of applications. The Series 68 with its high-speed CPU, extended I/O bays, and extensive data communication capabilities has the capacity to support either centralized EDP operations for a division or distributed EDP across a whole company. Given Hewlett-Packard's complete offering in manufacturing applications software, the Series 68 is also a strong system for manufacturing or quality assurance departments. In the office systems arena where department centralized EDP must be combined with an integrated family of intelligent workstations running office automation software, the HP 3000 family provides three choices, the Series 48, 42, and 39. For the small business or branch office, the low cost of the Series 39 or 42 makes these systems very attractive. The HP 3000 family is also purchased by educational institutions, OEM accounts, and system houses developing applications software.

A Commitment to Your Success

Hewlett-Packard is committed to making our products easy to use by non-data processing professionals, whether through HP supplied application programs or the interactive data access facilities for terminal users. Also, the wide range of program development tools makes it easy for programmers to create user-oriented programs.

Superior hardware and software is not enough. Comprehensive support services are available to assist you in maintaining all phases of your operation. The Series 39, 42, 48, and 68 can all be diagnosed remotely from an HP office by system experts. Another innovative service is the Phone-In Consulting Service (PICS) that enables your System Manager to receive immediate assistance via telephone to resolve software related problems. Personal attention from a well trained Systems Engineer is available whenever you need it. A full range of documentation, training, consulting, and support programs are offered to insure that you have the assistance you need to be successful. Hewlett-Packard supplies a full range of support services and gives you the flexibility to tailor them to your needs.

Hewlett-Packard documentation, training and support services are briefly discussed at the end of this chapter and in more detail in this and other general information manuals.

System Components

An HP 3000 computer system consists of the system hardware, the fundamental operating software, and additional software subsystems, all of which contribute to the easy development and operation of user applications. Hewlett-Packard also provides a set of ready-to-use applications developed for manufacturing, financial, wholesale distribution, and office automation environments. These components and their relationships are illustrated in Figure 1-2.

Broad Family of Processors

The Series 68 distributed mainframe is the high end of the HP 3000 family. The Series 68 supports 3 to 8 Mb of main memory, 3 I/O channels and up to 400 terminals. As with every HP 3000 the Series 68 concurrently performs on-line transaction processing, program development, batch processing, word processing, and data communications.

The Series 48 is a mid-range system supporting 2 to 4 Mb of main memory and up to 152 terminals. The Series 42 is a mid-range system similar in performance to the Series 48 but supports a smaller configuration. This system supports 1 to 3 Mb of main memory and 92 terminals.

The low-cost, entry level member of the HP 3000 family is the Series 39. Main memory on this system runs from 512 Kb to 3 Mb. The Series 39 comes standard with an integral cartridge tape and a 28, 65 or 132 Mb Winchester disc and can support up to 92 terminals.

An Integrated System

The fundamental HP 3000 computer system consists of the system hardware, and the Fundamental Operating Software that includes the MPE operating system, utility programs, and data management, and data entry subsystems.

The HP 3000 software is integrated with an advanced hardware design that includes stack architecture, variable length code segmentation, a hardware-assisted virtual memory scheme, user protection, and dynamic storage allocation. Hardware and software work together, with hardware performing many of the operations conventionally performed by software, such as interrupts or subroutine calls.

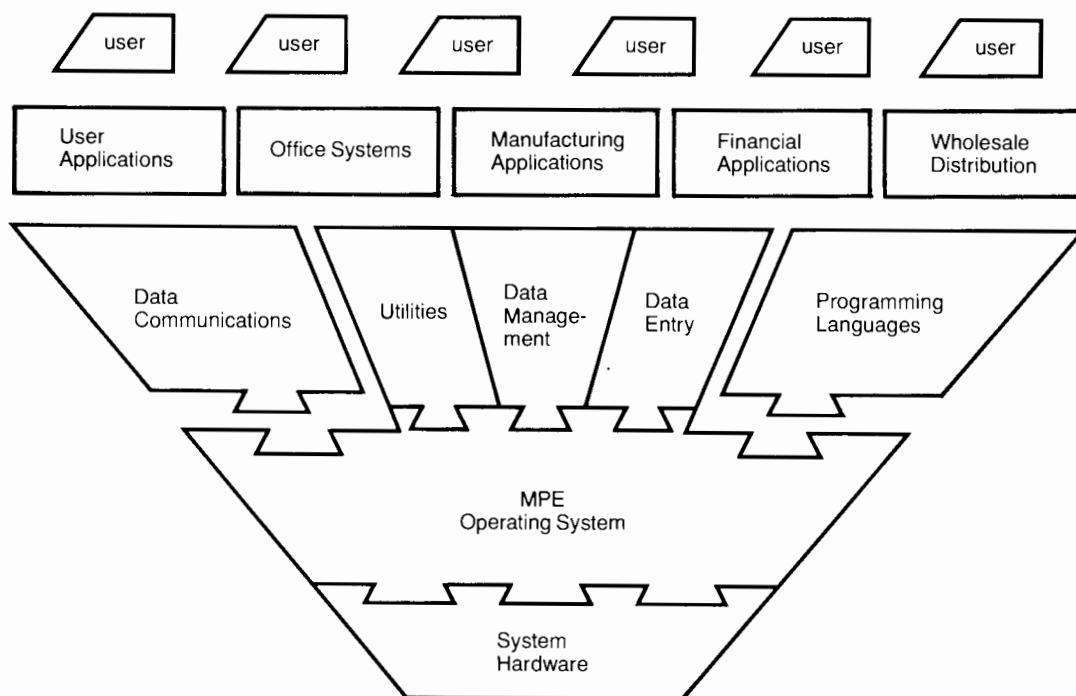


Figure 1-2 HP 3000 Computer System: System Components

System Hardware

The HP 3000 incorporates many features otherwise found only on very large computer systems, including:

- **Stack Architecture**—provides private, hardware protected data storage for each user plus automatic movement of this data to and from the central processor.
- **Virtual Memory**—consists of main memory plus an extensive storage area on magnetic disc to provide a total memory far exceeding the main memory size.
- **Separation of Code and Data**—strictly separate domains for code and data permits code sharing and re-entrant execution while maintaining data privacy.
- **Fault Control Memory**—high speed semiconductor memory modules provide automatic fault detection and single-bit correction.
- **Microcode**—more than 200 instructions are micro-coded operations within the system, plus a full set of microcoded system operations.
- **Concurrent I/O and CPU operation**—allows input/output to be performed concurrently with CPU operation.

Although the same software is used by all HP 3000 computer systems, specific system hardware differs for the Series 39, 42, 48, and 68. The Series 68 is described in Appendix D, the Series 48 is described in Appendix F, and the Series 39 and 42 are described in Appendix G. Generally, HP 3000 system hardware includes the central processor unit (CPU), main memory, and various peripheral devices available on each system series. Refer to Chapter 3 for a full discussion of the system architecture, and the HP 3000 Configuration Guide for configuration details.

Fundamental Operating Software

MPE Operating System

The Multiprogramming Executive operating system (MPE) is the disc-based software system which supervises the processing of all programs that run on the HP 3000. MPE dynamically allocates such system resources as main memory, the central processor, and peripheral devices to each program as needed. In addition, MPE coordinates all user interaction with the system, providing an easy to use command language interface and a powerful programmatic interface in the form of intrinsics and a versatile file system.

MPE monitors and controls program input, compilation, run preparation, loading, execution and output. It also controls the order in which programs are executed and allocates and maintains usage records of the hardware and software resources they require. By relieving you of many program controls, input/output, and other housekeeping responsibilities, MPE makes the HP 3000 extremely easy to use.

The major features of the operating system are:

- **Multiprogramming:** Concurrent transaction processing, data communications, on-line program development and batch processing
- **Virtual memory**
- **Disc Caching**
- **Stack architecture:** Separation of code and data, variable length segmentation, and data stacks
- **Concurrent multilingual capability:** COBOL, RPG, FORTRAN, BASIC, PASCAL and SPL
- **File system with file backup, user logging, security and Interprocess Communications**
- **Access security and complete accounting of resources**
- **Friendly, powerful command language,** including user-defined commands, conditional job control, on-line HELP facility, and meaningful error messages
- **Device and file independence**
- **Input/output conveniences:** Spooling of input and output, private disc volumes, and tape labels
- **Complete, automatic terminal management,** local and remote
- **Power fail/auto restart**

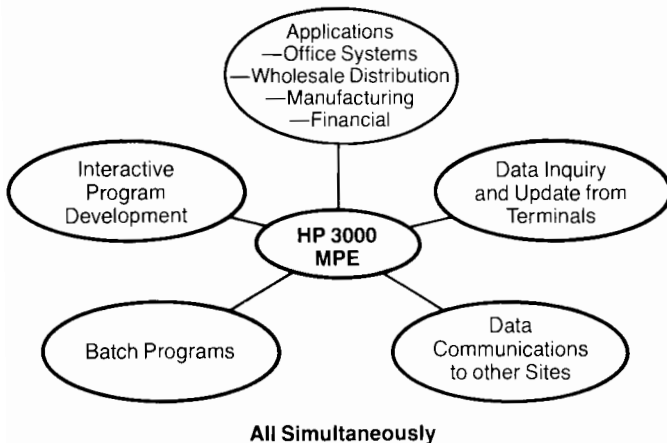


Figure 1-3 HP 3000 MPE Operating System controls and supervises all processing.

Data Base Management

A wide choice of data base management facilities are available to HP 3000 users.

IMAGE/3000—The award winning IMAGE/3000 is a network data base management system which allows information to be logically related between data sets (files), minimizing data redundancy and providing fast information retrieval. IMAGE provides comprehensive password security, predicate locking and roll forward recovery mechanisms. IMAGE can be called from any of the HP programming languages and multiple programs can access the data concurrently. IMAGE also provides remote data base access in QUERY/3000 distributed networks.

QUERY/3000—Developed primarily for programmers, this QUERY facility allows easy access to IMAGE data without writing a program. Simple commands make it easy to locate, update and report data. Programmers can easily check to make sure the files they create are set up correctly and check the accuracy of the data. QUERY procedures can also be saved to run at a later time.

KSAM/3000—An indexed file subsystem that speeds entry and retrieval of data by using the Keyed Sequential Access Method. KSAM/3000 provides a file organization based on values of key fields within the data records in the field. Records may be accessed randomly or sequentially. Generic searching is also handled very efficiently in KSAM. All the HP languages can access these files.

VPLUS/3000—A data entry software product which provides a friendly fill in the blank form to end users.

VPLUS features a form design facility that lets the programmer interactively create the data entry screen and define the associated editing criteria by answering simple questions. Advanced screen management and sequencing is command driven.

Also provided is a source data entry facility which allows on line entry and modification of data into a file without any programming. And a data reformatting facility that makes it easy to change the format of previously entered data to meet the input requirements of existing application programs. VPLUS also includes high level procedures for programmatic interface from any of HP's languages.

VPLUS requires HP block mode terminals and supports many advanced HP terminal features. These include labeled function keys, display enhancements, line drawing, local forms storage and local editing.

Utilities

A set of utility programs, standard on each HP 3000, eases program development and file manipulation and aids in system administration. The utilities included in the Fundamental Operating Software are:

- **EDIT/3000**—A powerful and easy to use text editor
- **FCOPY/3000**—A program for general file copying
- **SORT-MERGE/3000**—A facility for ordering records in a file and merging sorted files
- **System Utilities**—provide administrative controls, reports on system resources, and other special purpose capabilities.
- Facility to execute compiled programs without the source language compiler on the system.

The utilities, data entry and data management software subsystems are described more fully in the HP 3000 Software General Information Manual.

Additional Software Subsystems

In addition to the Fundamental Operating Software, a full set of programming languages, extensive data communication and data entry facilities are offered for use on HP 3000 computer systems.

Languages

Designed to provide you with language flexibility, the HP 3000 offers seven high-level programming languages which let you select the language best suited to the task. Programs can be written in:

- **COBOL II**
- **RPG**
- **FORTRAN**
- **BASIC**
- **PASCAL**
- **SPL** (Systems Programming Language—a high level machine dependent language that takes full advantage of HP 3000 design features)
- **TRANSACTION/3000** A high-level language designed for transaction processing

Applications written in different languages, or a combination of languages, can be run simultaneously. Data files and peripheral devices can be used in common by programs in any language without program changes.

Within MPE and the other subsystems of the Fundamental Operating Software reside high level procedures for terminal handling, data entry, and data management. These procedures, or intrinsics as HP refers to them, can easily be called from HP 3000 programming languages. Thus, the addition of a compiler is all that is needed to equip the HP 3000 as a program development machine. Because FOS is standard on all HP 3000's, compiled programs run on all HP 3000 computer systems and can take full advantage of the data entry, data management, and other high level program calls.

Productivity Aids

HPToolset is a productivity aid which includes a workspace manager, a full-screen editor, and HP Cobol II symbolic debugging. This combination eliminates the need to manage files while promoting an information sharing environment for programmers. The workspace manager manages all source files, versions of source files, INCLUDE files, USL files, and program files and listings. Since source files may be shared, individual and team programming efforts are simplified. The full-screen editor provides direct editing of text to simplify source-code entry and modifications. The Cobol II interface program key set consists of function keys which permit the user to compile, prepare, and run programs. The Cobol II symbolic debug locates run-time errors by using actual program variable and paragraph names rather than primitive-level memory locations and code addresses. The programmer can set breakpoints, trace retrace execution, and display and modify data-item values.

Performance Measurement Software

As applications grow, demands on HP 3000's grow. HP recognizes the desire to manage this growth to ensure maximum system performance. For the system analyst, the On-line Performance Tool/3000 (OPT/3000) performance measurement package is designed to help HP 3000 users gather and utilize performance data for system management, capacity planning, and application development activities.

Another performance measurement tool is APS/3000. Application Program Sampler/3000 (APS/3000) is a friendly interactive performance measurement software product for tuning application programs on the HP 3000, with the goal of increasing the overall system performance. APS/3000 helps the programmer to identify CPU bottlenecks in the source code of application programs.

The OPT/3000 and APS/3000 packages are discussed in detail in Chapter 2.

HP Distributed Systems Network (DSN) Software

The Distributed Systems Network (DSN) is a comprehensive set of data communications capabilities that cover three broad areas: Terminal/Workstation to HP system communications, HP system to HP system communications and HP system to IBM mainframe communications.

These sets of capabilities provide the customer with a powerful set of tools to distribute his processing both functionally and geographically, thus making information and computer resources available at locations they are needed.

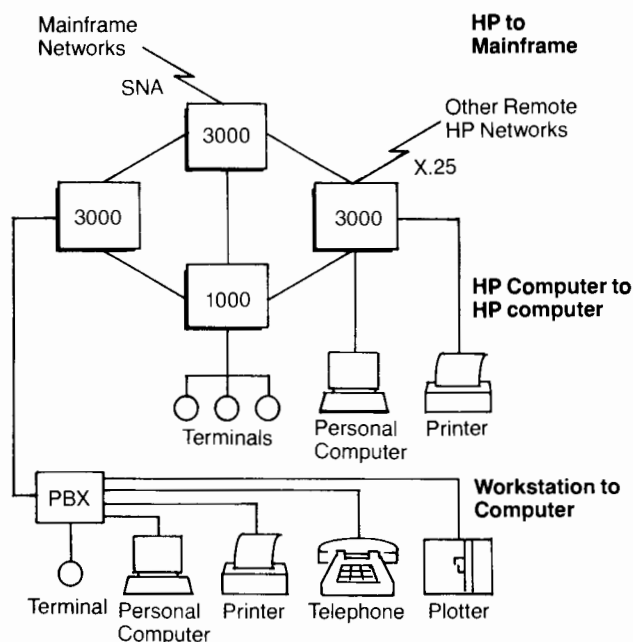


Figure 1-4 HP 3000 Distributed Systems Networking.

Terminal/Workstation to HP Systems Communications

In an environment where a large number of users need quick and easy access to the system it is desirable to be able to connect a number of terminals to the system. The point-to-point terminal capability provides a means of accomplishing this; this capability allows multiple terminals to be attached locally to an asynchronous port through either an EIA RS-232-C or RS-422 interface. This capability is also available remotely over a supported modem by using a EIA RS-232-C or RS-449 interface. The number of terminals that can be so connected depends on the specific model of HP 3000 being used.



In locations where terminal cabling is difficult, such as an office setting, or when users need to access multiple systems; HP 3000 users can utilize their digital Private Branch Exchange (PBX) for integrated voice/data communications. Both telephone operation and terminal communications can take place simultaneously. HP has certified several PBX manufacturers' systems for use with HP 3000 computers.

When the number of terminals that have to be connected to the system and the distance become large, the cost of individual lines becomes prohibitive. The DSN/Multipoint Terminal Software (DSN/MTS) capability provides an economical way of connecting multiple terminals to the HP 3000 over a single data communications line either locally using the DSN/Data Link, or remotely over supported modems. If the terminals are in clusters, the HP 2333A cluster controller can also be used with DSN/MTS in a local or remote environment. DSN/MTS also supports remote spooled printers which provides users with the ability to have remote data processing centers.

Applications which serve multiple remote locations can best utilize X.25 packet switched networks (PSN). The HP 3000 supports remote terminals or printers via the X.25 PSN either through network supplied interfaces or with the HP 2334A X.25 cluster controller.

HP Systems to HP Systems Communication

As companies grow, the need for data processing grows. Increased data processing needs are met through an increase in the number of systems that must be used. This results in an increased need for exchanging data between systems and a need for sharing system resources.

DSN provides a means for distributing data and resources among HP systems through the DSN/Distributed Systems (DSN/DS) capability. DSN/DS includes the following features:

- Direct Access of data bases on remote systems
- Access to files on remote systems
- Transfer of files between systems
- Access of remote resources such as peripherals and application programs
- Ability of programs on two systems to communicate
- Point-to-point local and remote connections
- X.25 Packet Switched Network connections

HP to IBM Communications

Large IBM mainframe computers often process data that is collected and transmitted by geographically/functionally distributed systems. The HP to IBM communication products provide the capability to transmit information to these IBM (and IBM plug compatible) mainframes, in either batch or interactive modes.

HP to IBM Batch Communication Products

DSN/Remote Job Entry (DSN/RJE) and DSN/Multileaving Remote Job Entry (DSN/MRJE) provide the capability to submit large batch jobs or files to the mainframe for further processing, using BSC protocol. They also turn an HP 3000 system and HP 2680 Laser Printer into a very efficient laser printer station.

HP to IBM Interactive Communication Products

DSN/Interactive Mainframe Facility (DSN/IMF) provides the capability to access data bases on the mainframe (such as IMS/VS and CICS/VS) or to do program development on the mainframe (using TSO). It also allows programs on the distributed HP 3000 systems to exchange data with programs on the mainframe through a set of easy to use intrinsics (callable routines). DSN/IMF uses either BSC or SDLC protocol.

The capabilities of these communications products are covered in greater detail in the HP 3000 Data Communications Products Specifications Guide (HP Pub. No. 5953-7444).

Manufacturing Application Software

HP Materials Management/3000

HP Materials Management/3000 is a customizable, interactive system for managing the materials planning and control functions of a manufacturing operation.

HP Materials Management/3000 consists of eleven application software modules:

- Parts and Bills of Material
- Routings and Workcenters
- Material Issues and Receipts.
- Inventory Balance Management
- Work Order Control
- Purchase Order Tracking
- Factory Order Entry
- Master Production Scheduling
- Rough Cut Resource Planning
- Material Requirements Planning
- Standard Product Costing

A discrete manufacturer who fabricates/assembles standard, multi-piece products represents the ideal candidate for HP Materials Management/3000. However, the software is applicable to most manufacturing operations.

Features of HP Materials Management/3000

- Use of proven materials management techniques
- Easy-to-use, on-line transaction menus
- On-line assistance via "Help" screens
- On-line IMAGE/3000 data base update
- Customizable user interface and data base
- On-line terminal data entry, field editing, and error correction using the capabilities of HP's data entry and forms management system, VPLUS/3000
- Automatic transaction logging
- Tailorable data entry screens and retrievals
- Use of "intelligent" HP CRT terminals
- Predefined materials data base
- Advanced security capabilities
- Automated operator functions

An important feature of HP Materials Management/3000 is the ease and speed with which the user can enter, retrieve, and modify data via an interactive terminal. The user has the capability to select a variety of "menu-like" screens on the terminal to perform the tasks associated with materials planning and control.

Overview

HP Materials Management/3000 is available in a form which meets the input, output, and processing requirements of most manufacturing companies. Manufacturing personnel may, however, modify data entry screens, data edits, and information retrieval screens to suit their specific needs. They may also add, delete, and/or modify data items in the HP Materials Management/3000 data base. All of these changes can be accomplished easily and without the need for computer programming personnel.

Three levels of customization capabilities are offered:

Operational Customization

A customer using Operational Customization will be able to tailor the environment in which the application executes.

Data Customization

A customer using Data Customization has, without the need of any programming, expanded tailoring capabilities as follows:

- Items can be added to or deleted from the data base and the characteristics of existing items can be changed.

- Menu and transaction screens can be added, deleted, or resequenced.
- Fields can be added, changed, rearranged, or deleted on existing screens.
- Help screens can be modified.
- Arithmetic relationships between data items can be defined.
- Reports can be customized by rearranging columns, changing column headings, and changing the length of data items.
- Automatic update of DICTIONARY/3000.

Advanced Customization

Advanced Customization provides additional tailoring capabilities for those customers who need specialized processing to satisfy unique business requirements. This level of customization is intended for those companies whose requirements can be met only through extension of the standard processing of HP Materials Management/3000 transactions.

Standard Application Products

HP Materials Management/3000 is a standard (object code) application product which is fully supported by Hewlett-Packard. A comprehensive support package is available which includes:

- User reference and installation manuals
- Customer training courses
- Planning, implementation, and customization consulting by Hewlett-Packard Manufacturing Application Specialists
- A wide range of Software Support Services

Each of the eleven HP Materials Management/3000 modules is related and integrated with the others. By maintaining bills of material and current information about parts, current inventory, order information, and time-phased master schedule information, the customer can take advantage of Material Requirements Planning to meet the company's production plan. Additionally, by storing labor, material and overhead costs in the bills of material and bills of labor, the user can calculate standard costs for each item in inventory.

Interfaces exist which allow HP Production Management/3000 to complement the functions of HP Materials Management/3000 and provide closed-loop manufacturing control. HP Materials Management/3000 can also be integrated with other application software including HP Financial Accounting and a sales order entry system to provide a complete manufacturing system as part of the HP Manufacturers' Productivity Network (HP-MPN).

Local Language Versions

HP Materials Management/3000 is available in a number of foreign language versions including German, Spanish, French, Dutch, Katakana, Australian, Norwegian, Swedish, British, Finnish, and Italian.

HP Production Management/3000

HP Production Management/3000 is a customizable, interactive system for managing the production planning and control function of a manufacturing operation.

HP Production Management/3000 consists of six software application modules:

- Routings and Workcenters
- Work-In-Process Control
- Work Order Scheduling
- Shop Floor Dispatching
- Work Order Tracking
- Capacity Requirements Planning

A discrete manufacturer who assembles standard multi-piece products in lots represents the ideal candidate for HP Production Management/3000. However, the software is applicable to most manufacturing operations. Hewlett-Packard Sales Representatives and Manufacturing Application Specialists are available for consultation regarding the suitability of HP Production Management/3000 for a specific application.

Features of HP Production Management/3000

- Use of proven production management techniques
- Easy-to-use, on-line transaction menus
- On-line assistance via "Help" screens
- On-line data base update
- Customizable user environment
- On-line terminal data entry, field editing, and error correction using the capabilities of HP's Data Entry and Forms Management System, VPLUS/3000
- Automatic transaction logging
- User manuals tailored to each job function
- "Intelligent" HP CRT's and factory data capture terminals
- Advanced security capabilities
- Automated operator functions
- Tailorable data entry screens, retrievals and messages
- Customizable data base and reports
- Local language versions

An important feature of HP Production Management/3000 is the ease and speed with which the user can enter, retrieve, and modify data via an interactive terminal. The user is presented with a series of "menu-like" screens from which any desired function can be easily selected using "soft keys."

Overview

HP Production Management/3000 is designed to meet the requirements of most manufacturing companies. Manufacturing personnel may, however, modify data entry screens, data edits, and information retrieval screens to suit their specific needs. They may also add, delete, and/or modify data items in the HP Production Management/3000 data base. All of these changes can be accomplished easily and without the need for computer programming.

System configuration, security, and job scheduling are also defined and modified without the need for programming personnel.

Three levels of customization capabilities are offered:

Operational Customization

A customer using Operational Customization will be able to tailor the environment in which the application executes.

Data Customization

- Items can be added or deleted from the data base and the characteristics of existing items can be changed
- Menu and transaction screens can be added, deleted, or resequenced
- Fields can be added, changed, rearranged, or deleted on existing screens
- Help screens can be modified
- Arithmetic relationships between data items can be defined
- Reports can be customized by rearranging columns, changing column headings, and changing the length of data items
- Automatic update of DICTIONARY/3000

Advanced Customization

Advanced Customization provides additional tailoring capabilities for those customers who need specialized processing to satisfy unique business requirements. This level of customization is intended for those companies whose requirements can be met only through extension of the standard processing of HP Production Management/3000 transactions.

Standard Application Products

HP Production Management/3000 is a standard (object code) application product which is fully supported by Hewlett-Packard. A comprehensive support package is available which includes:

- User reference and training manuals
- Customer training courses
- Project planning and implementation service products
- Planning, implementation, and customization consulting by Hewlett-Packard Manufacturing Application Specialists
- A wide range of Software Support Services

The six functional modules of HP Production Management/3000 are integrated with one another to provide a comprehensive set of production management functions. In addition, interfaces exist which allow HP's Materials Management/3000 (or some other materials system) to complement the functions of HP Production Management/3000 and provide closed-loop manufacturing control. HP Production Management/3000 can also be integrated with other HP application software including HP Financial Accounting to provide a complete manufacturing system as part of the HP Manufacturers' Productivity Network (HP-MPN).

Local Language Versions

HP Production Management/3000 is available in a number of foreign language versions including German, French, Katakana, Australian, British, Finnish, and Italian.

HP Lot Control and Traceability/3000

HP Lot Control and Traceability/3000 is designed to help manufacturers satisfy the lot tracking and reporting requirements of manufacturing environments using controlled substances or materials. It is a functional addition to HP Materials Management/3000 which is an interactive application software system for managing the materials planning and control functions of a manufacturing facility. Based upon the proven manufacturing principles of MRP, HP Materials Management/3000 tracks inventory throughout the manufacturing cycle at the part number level. HP Lot Control and Traceability/3000 extends the HP Materials Management/3000 capabilities to include the tracking and reporting of inventory at a lower level of detail—at the lot or batch level.

HP Lot Control and Traceability/3000 is designed to comply with government standards and regulations. These regulations require an historical record of the consumption of controlled substances. HP Lot Control and Traceability/3000 provides this control through the identification and tracking of controlled substances by lot number. This allows the content and genealogy of blends, assemblies, and end products to be examined quickly and easily.

HP Lot Control and Traceability/3000 consists of two modules:

- Lot Tracking and Control
- Genealogy Reporting

Lot Tracking and Control Features

- Lot identification specified on a part by part basis
- Comprehensive lot traceability for controlled components
- Potency adjusted inventory reporting
- Lot/location status control
- Pull quantity adjusted for actual potency at work order release
- Mix control at work order release
- Component expiration dating and control

The Lot Tracking and Control module facilitates the identification of batches of materials at all stages of manufacturing. A complete record of material movement is maintained throughout the manufacturing cycle for all parts designated as controlled components or substances. This information is automatically stored in a separate Lot Information Data Base for genealogy reporting and eventual tape archiving.

Genealogy Reporting Features

- On-line single level and indented Actual Where-Used
- On-line single level Actual Bill of Material
- Indented Actual Bill of Material and Actual Where-Used Reports
- Selection criteria which focus off-line genealogy reporting
- Tape archiving and archive management

Genealogy Reporting allows complete usage visibility from both the top down (explosion) and bottom up (where-used) points of view. A lot traceability explosion is the reporting, by lot number, of all controlled items used in a product; bottom up lot traceability indicates all uses of a given lot. Together, these reports detail the history and origin of lot controlled parts.

Documentation

A set of manuals is supplied with the software to provide reference and training material for HP Lot Control and Traceability/3000. These manuals address specific users by grouping logical system functions together. Three manuals are available:

- Parts and Bills of Materials with Lot Control
- Managing Inventory and Orders with Lot Control
- Genealogy and Archive Management

Training

A one-day training course for HP Lot Control and Traceability/3000, available either on-site or at an HP training facility, is offered for the System Administrator/User Trainer. The System Administrator is responsible for the day-to-day operation of HP Materials Management/3000. The User Trainer is responsible for training end users. These functions could be performed by the same person.

Financial Application Software

HP Financial Accounting is a series of eight integrated financial software products for the HP 3000 family of computers. They provide a complete range of solutions which can be tailored to fit your accounting environment. All the HP Financial Accounting products share a number of features:

- Application modules are integrated for one-time data entry
- Data processing can be on-line or batch
- Multiple companies and divisions are supported
- Advanced security throughout the system
- User friendly screens
- On-line management inquiry
- Interactive job scheduling
- Automatic voucher numbering
- On-line documentation
- Multiple currency handling with automated gain/loss booking
- Screen customization (except HP General Accounting)
- Data base customization (except HP General Accounting)
- Flexible report customization
- Interfacing customization
- Processing customization
- One vendor solution
- On-line "Help" screens

HP Financial Accounting is a standard (object code) product which comes with complete documentation, training, implementation and support services.

HP General Ledger

HP General Ledger significantly reduces closing cycles and evens out the flow of work to reduce closing activities. Product features which make this possible include:

- Total flexibility in defining companies, cost centers, charts of accounts, and fiscal years

- Up to 13 periods kept for both last years and current year's data
- Three budget amounts kept at account, cost center or account within cost center levels
- Standard vouchers automatically booked
- Accruals provide automatic reversal bookings
- Automated period and year-end closing procedures

HP Accounts Payable

HP Accounts Payable helps minimize the amount of cash needed to support regular business operations and optimizes discounts taken, reducing your cost of money. Product features include:

- Cash requirements forecasting
- Automatic tax, discount and due date calculations based on company policies
- Automatic payment proposals including advanced and partial payments with on-line maintenance and approval
- Multiple distributions per vendor invoice
- Hold capabilities for vendors and individual invoices
- Automated check writing and bank tape generation
- Recurring payments automatically created for review and posting
- Automatic and manual payment matching techniques
- On-line vendor analysis

HP Accounts Receivable

Accelerating cash inflows and improved cash management are significant benefits of HP Accounts Receivable. Features include:

- On-line customer analysis
- Automated statement generation
- Flexible aging by due date or estimated payment date, based on a weighted average of customer's payment history
- Automatic discount calculations and write-offs
- Multiple billing and aging cycles
- Automated delinquency notices printed in any language with five levels of severity and special exception capabilities
- Automatic or manual cash application features
- Recurring open items automatically created for review and release
- Cash receipts forecast

HP Dual Ledger

HP Dual Ledger expands the capabilities of HP General Ledger by adding a second set of books. HP Dual Ledger's features include:

- Entries are kept in both ledgers with one transaction
- Simulation capabilities to analyze future financial positions based on fluctuating currency rates
- Historical currency rates are maintained for analysis
- Two unique charts of accounts per company
- Automatic currency revaluation based on individual accounts
- Automatic posting of currency gains or losses

HP Allocator

HP Allocator is a sophisticated cost allocation application for use with HP General Ledger. Allocation criteria and rules are flexible, easily defined and carefully controlled by you. Product features include:

- Allocates both actual and budgeted amounts
- Amounts may be allocated within either the primary or secondary ledger if HP Dual Ledger is installed
- Unlimited user defined allocation criteria
- Allocation sources may be accounts, cost centers or any organizational unit reflected in the ledger
- Both multi-pass and sequential methods are supported
- Simulation techniques provided for control and auditability
- Allocations may be based on fixed or variable percentage or absolute amounts
- Zero-balance cost centers
- Audit reports

HP Report Facility

HP Report Facility is a powerful reporting tool which can be used to either modify the standard reports included with HP Financial Accounting, or define completely new reports, interactively, without programming. Features include:

- Easy creation of ad-hoc reports with no data base knowledge
- Same report formats may be used to report different sets of data
- Interactive page and line formatting
- Unique company forms (i.e. checks, statements) may be defined
- Automated, consolidated financial statements with intercompany eliminations
- Reports available both on-line and in hard copy format

HP Interface Facility

HP Interface Facility allows your analysts to interactively define interfaces between HP Financial Accounting products and your other in-house systems. This means a substantial reduction in systems implementation time. Product features include:

- Interactive data specification for files going to or coming from HP Financial Accounting applications
- Data conversion (ASCII to EBCDIC or vice versa)
- Sorting and summarization capabilities on data within the files for better audit control
- Automated restructuring of data within the files
- Standard interfaces pre-defined by HP

HP General Accounting

HP General Accounting is an integrated general accounting solution consisting of the comprehensive feature set and benefits found in HP General Ledger, HP Accounts Payable and HP Accounts Receivable. HP General Accounting meets the needs of customers who want multi-currency capabilities or who have unique reporting and interfacing requirements but do not need customization of screens and data base structures. HP General Accounting's features include:

- Feature sets identical to those described for HP General Ledger, HP Accounts Payable and HP Accounts Receivable
- Low-cost, integrated general accounting system
- Increased central control for distributed accounting systems
- A standardized system resulting in reduced maintenance and support costs

Because HP General Accounting is offered by Hewlett-Packard, HP General Accounting can be easily upgraded to provide more sophisticated accounting capabilities when your needs change.

Wholesale Distribution Software

System For Distributors/3000

SFD/3000 is a fully integrated operational application software system for the HP 3000 and is designed to improve productivity, control and cash flow in the distribution environment.

SFD/3000 provides a completely integrated computer system designed to meet the needs of a distribution organization dealing with inventory control and the processing of customer orders, shipping, billing, accounts receivable and the recording of payments made for goods and services provided. It also provides for the procurement of goods and services, receipt of goods, recording of vendor invoices and the production of checks to reimburse vendors for goods and services received. To maintain a record of activities affecting the assets and liabilities of the company, the general ledger is automatically updated and reports are generated to record these changes along with the company's financial position.

SFD/3000 software consists of a number of subsystems that access and update an IMAGE data base. The system is menu driven to provide ease of use and user friendliness. SFD/3000 is subdivided into the following basic subsystems:

- Sales Order Processing
- Inventory Management
- Purchase Order Entry
- General Ledger
- Accounts Receivable
- Accounts Payable

Features of SFD/3000

The SFD/3000 application system provides a comprehensive feature set designed to allow distribution companies to increase their profitability through an on-line operational tool designed by distributors for distributors. The following is a brief list of the key features of SFD/3000:

- Operational tools for distribution environments
- Fully integrated applications
- Real-time data base access and maintenance
- Numerous methods of order taking
- On-line editing and data entry
- Comprehensive distribution pricing
- Extensive installation support

Order Management/3000

OM/3000 is a fully integrated operational software system for the HP 3000 and is designed to improve productivity, control, and cash flow in the wholesale distribution environment. The system focuses on three vital areas: sales order processing, inventory management, and accounts receivable.

The OM/3000 software consists of a group of programs which update an IMAGE data base. These programs, executed under menu control, are subdivided into the following categories:

- Sales order Processing
- Inventory Management
- Accounts Receivable
- Sales Analysis

Features of OM/3000

- Operational tool for distribution environments
- Fully integrated applications
- Extensive installation support
- Real time data base access and maintenance
- Flexibility through business parameters
- Numerous methods of order taking
- On-line editing of data entry
- Comprehensive distribution pricing

Office Systems Software

Using the HP 3000 as the foundation, Hewlett-Packard's Office Systems products can help to bring your office into the computer age. We call it The Interactive Office. Hewlett-Packard's commitment to increase productivity in your office begins with tools for document management, decision support, personal support and organizational communication. Best of all, The Interactive Office products operate on all HP 3000 systems simultaneously with your data processing activities.

Document Management

An important step towards managing office information starts with controlling the voluminous amounts of text that must be handled every day. HP's document management products offer flexibility for the different types of document needs throughout your office.

The HPWORD word processing software teams with the HP 2626W Word Processing Station and HP2601/2602 Daisywheel Printers to provide a powerful secretarial system. HPWORD is a complete, powerful word processor that new users can easily learn through self-paced training materials, and experienced users will ap-

preciate because of its advanced operator-oriented features. HPWORD provides a full complement of document creating, editing, storing, printing, and recalling capabilities for quick memo, letter, and report generation. Graphics from HP business graphics products can also be merged with the HPWORD document and printed on the HP 2680 Laser Printing System. In addition, the HP 2626W can operate as a full data processing terminal on the HP 3000.

Text and Document Processor/3000 is a full capability text editing and document formatting system for the HP 3000. Using a command environment, TDP/3000 has extensive formatting features that can be used to create manuals, contracts, and lengthy proposals and also merge text and graphics. TDP/3000 operates with a variety of HP terminals and printers.

HPSLATE is a commandless, text processor that has a menu driven set of functions for entering, formatting, revising, printing, and saving shorter documents. Operating on a variety of terminals, HPSLATE utilizes screen-labeled function keys to perform the various editing activities.

The HP 2680A Graphics Package lets users merge text from our document management tools with charts and graphs created from our business graphics packages and print the result on the HP 2680 Laser Printing System.

The HP 2680A is a shared office network resource of high-quality, high-volume printing for both office and data processing needs. It's ideal for printing manuals, reports and other lengthy documents as needed, eliminating costly inventories.

Its special software allows users to create and store forms, logos, symbols and letterheads, then merge these with text and data to create final documents for distribution inside and outside your organization. The HP 2680 delivers clear, easy-to-read, printed output at 45 pages per minute on letter-sized paper.

Organizational Communication

Effective and timely communication is essential to your organization's day-to-day business. Information communication is critical to maintain accurate up-to-date records on accounts and documents which are distributed throughout your firm. With the Interactive Office individuals can electronically send and receive documents, messages and business charts and graphs.

HPDeskManager is a comprehensive electronic management software system for the HP 3000. It links electronic mail with a range of other desktop services. These include personal filing, time management and basic word processing. Each is accessed via a simple screen menu and the user can move freely and directly from one ser-

vice to the other using conversational commands. Electronic mail provides the ability to send and receive messages, documents, files and graphics, to and from any other users in the system. This may be implemented locally on a single system or a network of HP 3000 systems using public, private wire or dial-up circuits. Time management is the equivalent of the personal calendar. Daily appointments, reminders and action points can be listed and a day's entries can be as long as necessary. Personal electronic filing allows the user to file mail and documents in a set of "folders" which can be organized and structured to suit the individual's work routines. HPSLATE is provided with HPDeskManager for basic word processing functions, including the means to compose quick electronic mail messages or to draft lengthier reports. Once the rough draft has been entered, it can be quickly edited and formatted and merged with other text or files if required.

HPDRAW is an easy-to-use visual aid system for creating high-quality, personalized presentations combining words and graphics. A number of character styles and commonly used symbols are available. Figures, schematics and diagrams may be created and saved for later use. With HPDRAW, a user can easily revise presentation aids to keep them current. Final output can be to either overhead or hardcopy slides using HP plotters, or the HP2680 Laser Printing System.

The HP 3000 Distributed Systems Network (DSN) ties together your systems for distributed information processing. With DSN products, reports and documents can be sent among HP 3000 systems, between HP 3000s and other HP systems, and between HP 3000s and IBM or IBM-compatible mainframes.

Decision Support

Having timely access to accurate, more detailed information can help you make better decisions. The problem, however, is to quickly identify what information is relevant to each business decision. With HP Decision Support products, users have a more effective way to interpret information.

Decision Support Graphics/3000 is a fully interactive graphics software package that lets users design business charts without programming. It is a powerful management tool that offers many capabilities for efficient and timely management decision making. Requiring no special programming knowledge or graphics design knowledge, DSG/3000 is truly for the non-computer professional who needs the convenience of presentation graphics output at the press of a button.

HPEASYCHART is a "no experience necessary" chart-maker that puts graphics at the fingertips of a variety of users. It allows users to create line, bar and pie charts and scattergrams in minutes by simply entering the data

and pressing a button. A quick and easy solution for those one-time charts that don't require access to information stored in your HP 3000 system.

HP Inform/3000 allows users to produce reports in minutes, without programming. Just select report items from menus and the report is automatically formatted from data stored on your HP 3000 System. Report formats can be stored for later report generation and periodic checking.

Transaction Processing Tools

Several new productivity tools are offered to complement HP 3000 users in information management and retrieval. These new products offer a new and revolutionary concept in data management and application development. HP Dictionary/3000 provides for centralized control and standardization of information resources and creates logically related groups of information. Using HP Inform/3000 allows the user to quickly generate meaningful reports. HP Report/3000 is a non-procedural report-writer that can generate and format simple or complex reports from IMAGE data bases, KSAM, or MPE sequential files. HP Transact/3000 is a high-level fourth generation language for fast application development.

To simplify data entry, HP VPLUS/3000, a comprehensive data entry and forms management subsystem provides both a ready to use data entry program, ENTRY and a programmatic interface for your terminal-oriented application programs. It includes facilities for immediate on-line entry and modification of data, a wide range of data editing and validation, record reformatting, and interactive forms design and forms management.

HP PLUS— A Program For Locating User Software

HP PLUS provides HP customers with more software solutions. Many third-party software suppliers have written applications and utilities for the HP 3000 computer systems. The HP PLUS program locates this software and matches customers with the most appropriate package in one of two ways: by enabling customers to order these application packages directly through HP sales representatives, or by referring customers to the third-party software supplier.

Compatibility

The HP 3000 family of business systems is comprised of the Series 39, 42, 48 and 68 systems, each of which offers a full range of peripheral options and expandable hardware configurations. These HP 3000 systems feature compatible system software and application programs. Applications developed on one of these HP 3000 systems can be executed on any other of these systems. Each HP 3000 is supplied with the facility to execute compiled programs without the source language compiler on the system. The systems differ only in performance and expandability.

Features of to the HP 3000 family

- Patented stack architecture
- Virtual memory
- Automatic memory fault detection and fault correction
- Microprogrammed CPU
- Multiprogramming operating system
- Main memory cache for disc I/O (optional for the series 39)
- Complete system security and automatic accounting of resources
- Concurrent CPU and I/O operations
- Integrated terminal access with modem support
- Rechargeable battery packs to maintain memory data during power failure
- Automatic restart after power failure
- Microprocessor based diagnostic and control unit
- Fundamental Operating Software
- Software compatibility between all current systems.

Availability of software and hardware products is subject to change. Refer to the HP 3000 Configuration Guide for the most current product availability information.

Users

The final element in an HP 3000 computer system is the user. You interact directly with the computer system through the software subsystems and the operating system which in turn uses the hardware and the system peripherals. You may also interact indirectly through an application program. The tasks you perform usually fall into one of the following three categories:

- On-line interaction with application programs
- Program development
- System management

All of the system resources previously described are at your fingertips when you sit down at a terminal and access an HP 3000 computer system. Hewlett-Packard is committed to making the time you spend at the terminal friendly and highly productive.

Documentation

A comprehensive set of user manuals is included with the HP 3000. Complete documentation is provided for the operating system, for special tasks such as system installation and program conversion, and for the individual subsystems installed on the computer.

Figure 1-5 shows the four levels of MPE documentation. A complete listing of HP 3000 manuals is presented in Appendix A.

Introductory Level Manuals

These manuals introduce all of the major concepts of the HP 3000 computer system. The Guide for the New User, available late 1983, provides an introduction to the computer in a friendly, convenient format.

Standard User Level Manuals

These manuals provide complete reference information for the MPE operating system. The MPE V Commands manual covers operating system commands entered during an interactive or batch session. The MPE V Intrinsic manual details system procedures that may be called by the applications programmer. The MPE V Utilities manual explains the utility programs that may be used to determine system and session status and other information. The Segmenter and Debug/Stack Dump manuals are used by the applications programmer to optimize the debug code. The File System manual is a tutorial document which explains the details of the MPE file system.

Administrative Level Manual

The MPE V System Operation and Resource Management manual provides full reference and tutorial information for those responsible for administering the HP 3000 system. Account management, security, configuration, backup, and system operation are all detailed.

Summary Level Manual

The Quick Reference Guide, available late 1983, summarizes all MPE commands and functions. It is intended as a memory aid and as a convenient central reference for those who are already familiar with the use of the HP 3000.

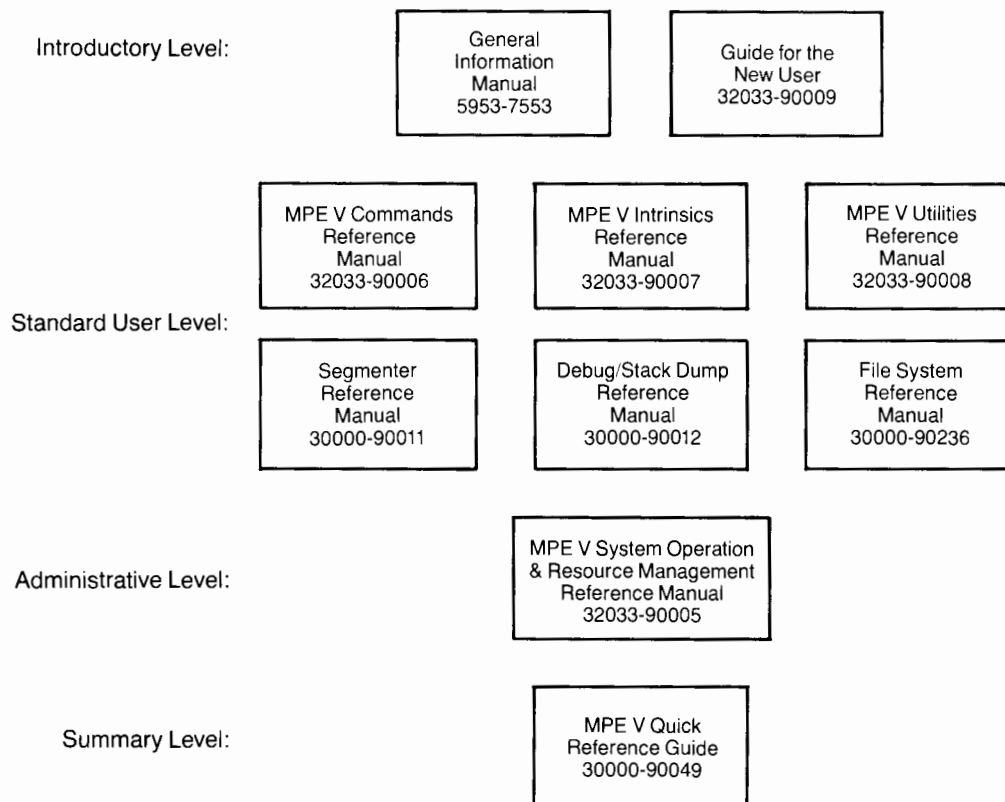


Figure 1-5



Training

Hewlett-Packard offers a variety of training courses that enable you to derive maximum benefit from your system's capabilities. A full curriculum of training courses spans the needs of a variety of users—from the system administrator, to the program developer, to the non-technical user of a software application.

For programmers, analysts and technical specialists:

- A Programmer's Introduction
- IMAGE/DMBS 3000
- HP VPLUS/3000
- System Management
- System Operator
- Application Design
- Transact
- Dictionary-Report-Inform
- SPL/File system Introduction
- HP 3000 Special Capabilities
- System Performance Training for OPT/3000
- A Comprehensive Introduction to Data Communications and Networking
- HP 2680 Laser Printing System

For applications specialists:

- Using Materials Management/3000
- Using Production Management/3000
- HP Manufacturing Systems Customization and Operations
- HP General Accounting/3000
- HP Financial Accounting
- DSG/3000 Programmatic Use
- TDP/3000
- Office Systems Coordination

Self-Paced Training

In addition to classroom instruction, Hewlett-Packard offers self-paced courses which allow users to learn, unassisted, at their own facility according to their own schedule. Among the courses offered are:

- A Guided Tour to the HP 3000
- Programming in Transact/3000
- Using Dictionary/3000
- Learning DSG/3000
- Learning COBOL II/3000
- Using HPWORD
- Using HPDRAW

Software Support Services

A well-defined set of software support services is offered with HP 3000 computer systems. Since customer support needs can differ considerably, these support services are available to provide a flexible range of support. The software services are divided into two main categories:

- **CSS—Customer Support Service**, for customers who choose a close support relationship with Hewlett-Packard.
- **SSS—Software Subscription Service**, for customers who prefer to rely on their own resources for software support, but still want to receive documentation and updates.

Documentation Distribution Services

In addition to the software support services described above, Hewlett-Packard offers two types of documentation distribution services that are appropriate for customers with a large programming staff who wish to be individually informed of software problems or keep their documentation up to date. These services are:

- **SNS—Software Notification Service**, provides a subscription to the Software Status Bulletin and the Communicator.
- **MUS—Manual Update Service**, provides one copy of updates or new editions to manuals automatically whenever they are issued. The various sets of manuals that can receive this support are specified in the Configuration Guide.

The reference sheets in the HP 3000 Specification Guide provide a full description of the Software Support Services outlined above.

Hardware Support Services

Hewlett-Packard offers a full range of system and product maintenance services for your selection. Your Sales Representative and your Customer Engineer will help you best plan your support program.

All Services

Hardware services are purchased under the Customer Support Services Agreement (CSSA). Agreements provide a known monthly cost that can be budgeted. All hardware services include parts and labor for remedial maintenance. Engineering improvements are installed to assure that your system performs to Hewlett-Packard's high standards. On-site warranty for 90 days is included for all HP computer products purchased with your HP 3000. Warranty services are extended to match the coverage available under the maintenance service provided the service agreement is signed at the time of system order. A short-term maintenance option, from 90 days to one year, is available for OEM's.

System Maintenance Services

These services are designed for system CPU's and their associated major peripherals. The system support services include valuable account management services such as an account-assigned Customer Engineer, Preventive Maintenance, Add-on Installation and Problem Escalation. For eligible products, a Remote Support Option is available. Under this option, service is initiated from an HP site via a telephone connection to the system. Remote Support significantly reduces maintenance costs and is strongly recommended.

Guaranteed Uptime Service (GUS). This service provides the highest level of maintenance service for Hewlett-Packard's most advanced computer systems. The service is designed for the user who requires maximum system availability. Features include:

- System uptime to exceed 99% during any three consecutive month coverage period
- Service credit of one month's charges for all products covered by GUS whenever the uptime percentage is reported below 99%
- Monthly activity report, listing all maintenance activity during the reporting period
- Round-the-clock continuous coverage; four hour response to all service requests

Standard System Maintenance Service. This service provides a high level of service with the quickest response. A choice of nine different coverage periods—13, 16, and 24 hours per day, 5, 6 or 7 days per week—are offered to meet the specific needs of your operation. Features include:

- 4 hour response to service requests placed between 8AM and 5PM, at sites within 100 miles of a Primary Service Responsible Office (SRO)
- Remote Support Option available to reduce service cost
- Site Environmental Surveys performed periodically to assure a sound operating environment
- Scheduled preventive maintenance to identify potential problems before they occur
- Work will continue even after your coverage period has ended once on-site work on a remedial repair has commenced and progress is being made
- Installation for new HP computer system products added to your system already covered by this maintenance service

Basic System Maintenance Service. This service offers all the features of the Standard System Maintenance Service, but with a slower response for a reduced cost. Next-day response, Monday through Friday, to service requests for sites within 100 miles of an HP Primary SRO. Other response times are specified for service beyond 100 miles.

Product Support Services

Product Support services are provided exclusively for HP workstation products—personal computers, flexible disc drives, terminals, small printers and plotters. Augmenting these services with spare units will yield a very high degree of equipment availability at the lowest possible prices. The following services are available:

On-Site Product Maintenance Service. This service provides next day, on-site response to customers within 100 miles of a Primary SRO. Coverage hours are from 8AM to 5PM, Monday through Friday. (Scheduled preventive maintenance for eligible products is typically unnecessary or easily performed by the user.)

Volume Repair Center Maintenance (VRC). This service offers the lowest on-site support costs. It is available for most workstation products; customers must place a minimum of 25 eligible units on this agreement. HP will make scheduled, weekly visits to a single, central site chosen by the customer. VRC is available to customers located within 100 miles of a Primary SRO.

Field Repair Center Service. This service offers the lowest hardware support costs for selected HP workstation products. An approximate 50% cost savings over the next-day on-site program can be realized through this program. In the event repairs are necessary, you ship the unit to the nearest HP Field Repair Facility. HP will repair the unit and ship it back to you within 3 days of receipt at the Field Repair Center.

Consult your account representative for more details about these services.

Consulting Services

To help you improve your productivity with your HP 3000, Hewlett-Packard offers on-site Systems Engineering consulting, both as standardized services and on a time and materials basis. Standardized consulting services provide assistance for clearly defined and commonly encountered areas of need. Time and materials services are also available for problems which do not fall into the standard categories.

The standardized consulting products presently available for the HP 3000 include:

- **Installation Management**—This service is designed to help the System Manager of a new installation become immediately productive by reducing start-up problems, customizing procedures to your objectives, and planning for long-term operational success. This includes discussion and recommendations on performance variables, accounting structure design, back-up strategy, system start-up methods, problem management, disaster recovery, job scheduling and load management.

- **System Performance Evaluation**—This service can help you optimize system performance by identifying bottlenecks and their causes, and by recommending a strategy for corrective action. A specially trained HP Systems Engineer uses HP software to analyze the system's job mix and workload, and recommends changes in areas such as job mix, memory management, utilization of I/O devices and files, and subsystem configuration.

The scope of these services is described in individual data sheets, available from your local HP sales office. For more information on Consulting Services, please contact your HP Sales Representative or Systems Engineer.

HP 3000 International Users Group

The Users Group is an independent world-wide organization for the purpose of exchanging techniques and ideas among HP 3000 users. Hewlett-Packard's liaison works closely with the Users Group to promote communication between Hewlett-Packard and HP 3000 users. Membership is open to any interested individual using an HP 3000. You will find that meeting with other users and exchanging ideas provides a stimulating environment in which to sharpen your programming and operational techniques. For further membership information contact:

HP 3000 International Users Group
289 S. San Antonio Rd.
Los Altos, CA 94022 U.S.A.

2

System Software

MPE Operating System

Performance Measurement

Flexible Disccopy

MPE Operating System

The functional heart of the HP 3000 is the Multiprogramming Executive, MPE. This general purpose, disc-based operating system supervises all processing and maintains all user interface with the HP 3000. Two major attributes of MPE are its versatility and ease of use.

Designed to take full advantage of the computer's hardware features such as virtual memory and stack architecture, MPE demonstrates its versatility by enabling the HP 3000 to perform transaction processing, on-line program development, data communications, and batch processing concurrently. In addition, MPE permits system resources to be accessed simultaneously by multiple users, each of whom interfaces with the system independently.

MPE demonstrates its ease of use with its many user assistance features such as a powerful, straightforward command language and an on-line HELP facility which guides you in using MPE commands. In addition, MPE simplifies the programming task by monitoring and controlling program input, compilation, execution, and output. MPE regulates the order in which programs are executed, and dynamically allocates any hardware and software resources the programs require.

A complete account structure and automatic resource accounting are standard features of MPE. Easy to use MPE commands allow the system manager to set up a hierarchical accounting structure on the system in a style similar to a company organization chart. MPE then automatically keeps track of the system resources used

by the various groups in the account structure. This resource usage information can then be used for billing, accounting, or any other application that requires such data.

MPE provides complete security, enabling you to operate in an environment protected from interference or illegal access by other users. This security is accomplished by means of multiple logon passwords, file lockwords, hierarchical access restrictions, and user capability sets.

MPE's interprocess communication facility allows processes to communicate efficiently with each other via disc files. All processes which have access to a message file via the security system may communicate with each other using standard file system intrinsics.

MPE handles all input/output to peripheral devices, receiving the I/O requests, queueing them if necessary, and performing the actual data transfer. Because MPE treats I/O devices as files, you can write programs without concern for the physical source or destination of the data, and you can run them in either batch or interactive mode without changing the names of the files they reference.

MPE also handles asynchronous terminal communications. Synchronous data communication to terminals, other HP computers, or even non-HP computers, is provided by optional data communications subsystems.

Figure 2-1 illustrates the major components of the Multiprogramming Executive Operating System.

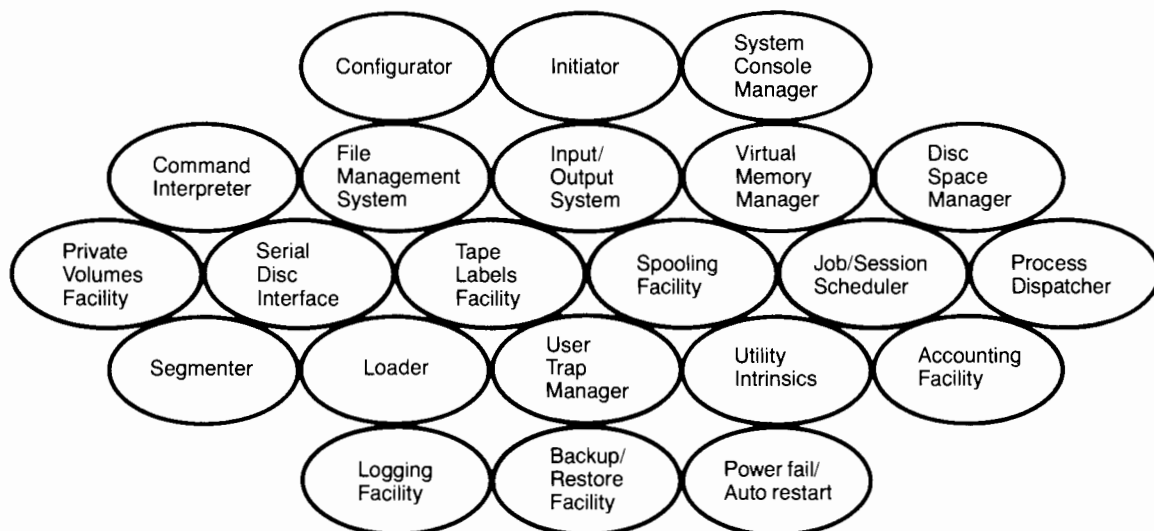


Figure 2-1 Components of the MPE operating system.

MPE V Features

Expanded System Tables

Through a major expansion of system tables, MPE V provides the software support for the expanded configurations available with the newest members of the HP 3000 family. MPE V makes it possible for you to configure three times as many terminals on our high-end HP 3000 system. As a result, you can distribute computing power even further throughout your organization. The larger MPE V tables also allow the support of significantly more HP 3000 peripherals. With the expanded systems software, the Series 68 will support a third IMB, thereby increasing by 50% the number of disc drives supported on the system.

As the size of your configurations continues to grow, an increasing number of users will want access to the system. MPE V makes the HP 3000 simultaneously accessible to more users than ever before. Up to 400 users can now log on concurrently, with about 200 of these users developing programs or running applications. An important change in table structure and organization will allow these users' programs to be segmented into smaller segments for more efficient memory management, and it will increase the maximum program size from 2 to 8 Mb.

The enlarged tables give your HP 3000 the room for future expansion. As you place continually heavier workloads on your system, you will require higher performance to satisfy all of your users' needs. MPE V-based disc caching addresses this need by significantly improving system response and throughput.

Disc Caching

Disc caching is an innovative technology available with MPE V and subsequent releases that significantly improves I/O performance on the HP 3000. Designed particularly for I/O-intensive and database-intensive environments, disc caching management routines work to eliminate many of the time-consuming disc accesses that have traditionally limited I/O and system throughput.

How does disc caching work to enhance system performance? The technology uses main memory as a high speed buffer for disc. Disc caching places those most frequently accessed portions of files and directories in available portions of main memory, where they can be accessed repeatedly by an executing program. Thus, instead of going immediately to the disc to get requested information, disc caching first checks to see if the file information is already in memory, i.e. in the disc cache. When the information is located in the disc cache, I/O performance is improved in two principal ways. First, several time-consuming disc accesses are eliminated. Second, since information may be accessed in main memory 10—100 times faster than it may be accessed from disc, disc caching greatly reduces the time required for an I/O to complete. This two fold improvement means better response and higher throughput for I/O-intensive applications.

Disc caching is most effective in applications which have a high hit rate. The disc cache hit rate is the frequency with which the desired information is found in main memory, and an access to disc eliminated. For a typical application, the disc cache hit rate varies from 50—70%. The hit rate is particularly high and the performance improvement from disc caching greatest with applications where multiple users share the same information. Programs which share IMAGE databases, for example, benefit greatly. Disc caching not only reduces the contention for these data bases; it also reduces the lock service times of the data base control blocks. Because I/O requests can be serviced more quickly with disc caching, the locking delays are reduced, and the throughput of applications like IMAGE is increased.

The MPE memory manager controls the operation of disc caching, and handles file information in the cache dynamically just like a data segment. As a result, special areas of main memory need not be dedicated for the disc cache. However, users who wish to gain significant performance benefits from disc caching must configure an adequate amount of main memory on their systems. Your local SE or performance specialist can help you decide how much, if any, additional memory will be required.

Though the operation of disc caching is fully transparent to the user, the system manager or console operator can enable, disable, and control caching through designated operator commands. Since disc caching can be enabled on a device-by-device basis, only those disc drives which will benefit from caching need to have caching enabled. Full guidelines on the use of all the disc caching commands are provided in the MPE V System Operations and Resource Management Manual.

Efficient Versatility

All HP 3000 computer systems operate under a single operating environment—MPE. This means that you can move from one HP 3000 to another without undergoing additional training for a new environment.

Multiprogramming

One of the major ways in which operating efficiency is achieved in an HP 3000, is by multiprogramming—the concurrent execution of multiple programs. Multiprogramming allows system resources to be allocated among several competing programs. While one program is awaiting an I/O operation, for instance, control of the central processor is shifted to the next highest priority program waiting for the CPU. MPE is designed to allocate, schedule, and dispatch control of the central processor, storage, and input/output devices among the competing programs. This controlled competition for system resources reduces turnaround time, and significantly increases system throughput. Operating in conjunction with the architecture of the central processor, MPE provides complete protection against one program interfering with another.

The number of programs that can be processed concurrently depends upon such factors as hardware configuration, program operating modes, and the application programs involved. MPE is designed so that the maximum number of concurrently running programs can be increased or decreased by changing a single system configuration parameter.

MPE allows the concurrent execution of programs from two types of input media—traditional batch input devices and interactive terminals. Programs are independent of their input mode and the same system code is used to perform particular functions in either mode. This results in storage economy and reduced overhead.

Interactive Processing

When using the interactive processing mode, you enter commands and data through a keyboard terminal and receive immediate responses to your input. This type of interaction is called a session and is especially useful for program development, text editing, data entry, information retrieval, computer-assisted instruction, and other applications where a direct dialogue with the system is preferred. Sessions can be used to access:

- Operating system commands and subsystems
- Language and utility programs
- Data base management programs
- Data communications programs
- Application programs
- Office System programs

A session begins when you enter the HELLO command from an on-line terminal and MPE connects you to the command interpreter. You may then enter commands to use language compilers or other subsystems such as the text editor, to run programs, or to modify your files. The session continues until you enter a :BYE command, a new :HELLO command, or the system operator intervenes to abort the session.

As an example let's assume you want to create a COBOL program and then compile, prepare, and run it during an interactive session. Figure 2-2 shows the various commands entered during such a session. You initiate the session by pressing the RETURN key on a terminal that is connected on-line to the system. MPE responds by displaying a colon prompt character. You log on to MPE by entering a HELLO command containing your assigned user and account names, then call the HP 3000 text editor and enter two editor commands followed by the COBOL statements that constitute your program. When the entire source program has been entered, you save it on disc under a file named YOURFILE by entering a KEEP editor command, and then terminate the editor subsystem. Now the source program exists as a disc file in the system. To compile, prepare, and execute the program, you simply enter a COBOLGO command specifying the file YOURFILE. This one command first invokes the COBOL compiler which compiles the program, then invokes the MPE segmenter which prepares the compiled program into an executable form, and finally executes the prepared program. When the program has finished executing, MPE displays the message END OF PROGRAM followed by another colon prompt character. You terminate the session by entering the BYE command.

```

:HELLO YOURNAME.YOURACCT _____ Initiates the session.

:EDITOR _____ Invokes the HP 3000 text editor.

/SET FORMAT=COBOL _____ Specifies that you will be entering COBOL source statements.

/ADD _____ Specifies that you wish to enter source code.
    (COBOL source statements)

/KEEP YOURFILE _____ Saves the text file on disc under the name YOURFILE.

/EXIT _____ Terminates the text editor.

:COBOLGO YOURFILE _____ Causes the COBOL source program contained in YOURFILE to be compiled, pre-
    pared, and executed.

:BYE _____ Terminates the session.
_____ Prompt characters issued by MPE and the text editor.

```

Figure 2-2 Sample session

The example above is somewhat simplified since it does not include the various informational messages, compilation output, and program output generated by MPE, the text editor, the COBOL compiler, and the program itself. The fact remains, however, that if the source program (entered by way of the editor) contains no errors, the entire session can be performed by entering just eight MPE and text editor commands in addition to the COBOL statements which constitute the program.

Batch Processing

Batch processing is a logical extension of the interactive functions available through MPE. Any capability, with the exception of BREAK, that is available in one mode, is available in the other and employs the same MPE commands. Languages, utilities, and applications development software can be run in either batch or interactive mode without changes. The standard input and output devices are automatically redefined.

The batch processing mode lets you submit to the computer, as a single unit, commands that request various MPE operations such as program compilation and execution, file manipulation, or utility functions. Such a unit is called a job. Jobs contain all necessary instructions to MPE and all references to programs and data required for their execution. Once a job is running you need to supply no further information.

Jobs are often read through batch input devices such as card readers or tape units. A unique feature of MPE, however, allows you to enter batch job streams through the terminal during the course of an interactive session.

Several jobs can be submitted to the system from multiple devices concurrently. Batch job input is spooled on disc and MPE schedules each job for execution according to its job input priority specified in the JOB command. Additional commands are provided for monitoring job selection. The system operator specifies the maximum number of jobs that can be executed concurrently and can dynamically adjust the job selection criteria.

When a job enters execution, the commands within it are executed sequentially on a multiprogramming basis. MPE generates the job output on a local device such as a line printer, tape unit, or disc unit, or on a local or remote terminal. When one job is temporarily suspended, perhaps to await input of data, another job or session (if available) immediately enters execution. Spooled output on disc is selected for output processing according to the output priority specified in the JOB command.

MPE executes many sessions and batch jobs simultaneously. The only significant difference between a session and a batch job is that during a session you can interactively alter the course of processing, whereas in a job, the command stream is fixed and the job will be executed in its entirety, as pre-defined in the job control statements, without active intervention.

A User-oriented Operating System

The many features and capabilities of the Multiprogramming Executive operating system are designed around the concept of the user. There are two distinct types of users:

- the end user, such as a data entry clerk, whose only concern is running a compiled application program, and
- system administrators, such as application programmers, who are responsible for the creation and maintenance of application programs as well as the day-to-day operation of the system.

Each type of user is associated with a particular set of capabilities and responsibilities, and each has access to MPE features which assist him with his specific tasks.

User Classification

Programmers are users who create application programs which run on the system. MPE provides two major areas of system interface for these users: an interactive interface which includes a command language, an on-line HELP facility, and job control facilities; and a programmatic interface which includes programming intrinsics and the MPE file system.

The end user, who can range from an order entry clerk to a functional manager, takes advantage of all the capabilities of the operating system through an application program which he can run without any knowledge of MPE itself.

System administration is performed by a hierarchy of users whose overall responsibility is the successful administration of the computer system. The various levels of MPE administration are defined in Table 2-1. In a small installation, a single user may perform the functions associated with all levels of administration. In a larger installation, the capabilities may be divided among several individuals at each administrative level. Thus it is more appropriate to think of "a user with system manager capabilities" rather than a formally titled "system manager."

Table 2-1 System Administrators

System Manager:

Manages the overall system by creating accounts (basic structures for user access) and defining resource-use limits.

System Supervisor:

Manages the system on a day-to-day basis, controls scheduling queues, alters system configuration, and maintains the system library.

Account Managers:

Maintain accounts by defining the valid users and file groups for the accounts and specifying resource-use limits for them.

The system operator is the user who operates the system console and is responsible for responding to all system requests. MPE provides a range of operational capabilities which augment the performance of day-to-day operations such as system start-up, back-up, maintenance and recovery, as well as helping the system operator keep the system operating as smoothly and efficiently as possible.

User Capabilities

Capability sets are assigned to each system user based on the kinds of tasks each user needs to perform. These capability sets are divided into three categories:

- User attributes
- File access attributes
- Capability-class attributes

User attributes include system manager, system supervisor, and account manager capabilities. Additional capabilities such as account librarian with special file access capabilities for maintenance of account files, group librarian with special file access capabilities for maintenance of group files, and diagnostician with the ability to run diagnostic programs under MPE for on-line checkout of HP 3000 hardware components, may also be assigned.

There are two file access attributes: *save*, which permits the user to save files by declaring them permanent, and *non-shareable devices*, which allows a user to use non-shareable devices, such as a magnetic tape unit.

Capability-class attributes refer to the ability to access special MPE facilities. Included here are interactive access, local batch access, process handling, extra data segment acquisition, multiple resource identification numbers, privileged mode, user logging, private volumes, and data communications.

The majority of users in a typical HP 3000 installation will simply have capabilities for interactive access and local batch access. MPE simplifies the assignment of user capabilities by establishing a set of default capabilities. If a user needs additional capabilities later, these can easily be added.

The presence of capability sets greatly simplifies the use of the system from the standpoint of each individual user by defining the extent to which he must understand MPE, and permitting him to ignore those aspects of the system that do not apply to him.

User Interface

Three user-oriented software facilities provide a comprehensive interface between the system/application programmer and MPE. These tools are: the MPE command language, the on-line HELP facility, and job control facilities.

Command Language

The simplicity of the MPE command language greatly enhances the system's usability. MPE commands enable you to initiate a session and specify the various MPE operations you wish to have performed. When you specify a command, a portion of MPE called the command interpreter reads the command, checks its validity, and then causes the appropriate action to be taken. After the requested action is successfully completed, the interpreter processes your next command. You may enter commands interactively during a session or through a batch input device. Commands may also be issued programmatically from a running program.

MPE uses a colon (:) to prompt you for a command during an interactive session. When a batch job is submitted, MPE commands within the job are designated by a colon in column one.

The MPE command language is composed of many commands. Each command enables you to request a specific action of MPE. Collectively they provide a powerful system-usage tool. The full range of MPE commands is presented in Appendix B of this manual. The following is a list of common command uses:

- Initiating and terminating jobs and sessions.
- Running system programs or compilers.
- Running programs.
- Running system utilities.
- Creating, managing, or deleting files.
- Displaying file information.
- Displaying job, session, or device status.
- Transmitting messages.
- Assisting in program debugging.
- Establishing communication between a local and a remote computer.

If the command interpreter detects an error during a session, MPE informs you with an error message which specifies the erroneous parameter. MPE then requests that a new command be entered. You can instantly correct command errors by retyping the command or using the REDO command, which allows you to edit the erroneous command, and the session can continue.

During a batch job, MPE lists the error on the listing device. Input from that point through the next EOJ, DATA, or JOB command are usually ignored. You can, however, use the CONTINUE command to request that the job be continued despite the error.

User-Defined Commands

MPE allows you to define your own commands by combining several MPE commands into a command procedure and assigning the procedure a name. The name can then be used as a command. Thus it is possible to enter a single command name which you have defined and cause several commands to be executed. These user-defined command sets can be created by each individual user as well as being made available to entire accounts and all accounts systemwide. It is also possible to redefine existing MPE commands and messages to suit your particular situation.

On-line HELP Facility

Whenever you need assistance with command language syntax or even the name of a particular command, you can invoke the on-line HELP facility. This facility provides graduated information on any MPE command or set of commands. The HELP messages displayed coincide with the information contained in *HP3000 Command Reference Manual*.

Figure 2-3 demonstrates the two ways in which the HELP facility can be used. In the "immediate" mode, you merely enter the HELP command followed by a parameter. Information detailing that parameter is displayed immediately. In the "subsystem" mode you enter the HELP command without any parameters. The system then displays a menu of valid parameters, and prompts you with a greater than sign (>) for the parameter you wish explained.

:HELP REMOTE,EXAMPLE _____ Example of HELP being used in the "immediate" mode.

EXAMPLE

To establish the communications link HDS2 and log on to System B from System A, you could enter:

```
:HELLO USER.X
:REMOTE HELLO USER.X;DSL=HDS2
KEYWORDS:PARMS,OPERATION,EXAMPLE
:
```

MPE Prompt. You are given a new prompt automatically following the display of information.

:HELP _____ Example of HELP being used in "subsystem" mode.

Information is available on the following classes of commands:

```
Running Sessions
Running Jobs
Managing Files
Running Subsystems and Programs
System Management, Status, and Accounting
Utility Functions
```

For more information enter a KEYWORD. You can also enter any command name as a keyword. Enter "help" for information on help. Enter "exit" to leave help.

KEYWORDS: SESSIONS,JOBS,PROGRAMS,FILES,MANAGE,UTILITY

>SESSIONS

Running Sessions. Following are the commands used:

```
( ) COMMAND LOG ON
ABORT
BYE
DSL=
EOD
EOF
HELLO
HELP
REMOTE HELLO
RESUME
```

You can use any command as a keyword.

KEYWORDS:SESSIONS,JOBS,PROGRAMS,FILES,MANAGE,UTILITY

>EXIT _____

You EXIT the HELP Subsystem

:

MPE Prompt

Figure 2-3 Using the on-line HELP facility.

Job Control Facilities

MPE contains job control words (JCW) and conditional execution functions which permit the user to design job streams whose execution can be dynamically altered based on the results of previous job steps.

You can use both system defined and your own job control words to store job status information and to pass such information between programs and between a program and the MPE command interpreter. JCWs are defined and accessed by commands from the command interpreter and by intrinsics from your programs.

You can also use JCWs in conjunction with conditional execution function statements. These statements specify

a logical expression (TRUE or FALSE), and are evaluated during program execution. If the value found is TRUE, the remaining statements related to that condition are executed. If the value is FALSE, any existing alternative statements are executed instead.

The following example illustrates the use of JCWs and a conditional execution function. The sample job runs a program which edits and verifies transaction cards and counts valid transactions. If no fatal errors are encountered, the job schedules shipments (either all shipments or only high priority shipments depending on the value of JCW) and produces a final report. If fatal errors do occur, the job does no shipment scheduling. Instead, it produces only an error report and a final report.

```

:RUN P108X1_____ (Edit and verify transaction cards)
:RUN P108X2_____ (Count valid transactions)
:IF (JCW < FATAL) THEN_____ (If no fatal errors, schedule
                                shipments)
:  IF (JCW < 5000) THEN_____ (Number of shipments to schedule)
:    RUN P108X3_____ (Schedule low priority shipments)
:  ENDIF
:  RUN P108X4_____ (Schedule high priority shipments)
:ELSE
:  RUN P108X5_____ (Produce error report and fix JCW)
:ENDIF
:RUN P108X6_____ (Produce final report)

```


Program Development

The Multiprogramming Executive provides meaningful assistance with the task of generating application programs. MPE programming assistance includes:

- Consistent command language interface to all compilers.
- Program preparation performed by the MPE segmenter.
- Procedure libraries for external references.
- A device-independent file system.
- Flexible file security.
- Subroutines callable across languages.
- Access to all system intrinsics.

The use of these software tools during program generation is described below.

Creating Programs

Three steps are required to take a program from source form to an executable state. The first step is to compile the source program into relocatable binary modules, called RBMs. This is done by the various MPE language compilers which automatically store the RBMs in a specially formatted file called the user subprogram library, or USL.

The second step is to take the USL file and prepare it into a program file. Program preparation resolves external references and results in loadable code segments. This step is done by the MPE segmenter.

The third and final step is to have the MPE loader allocate entries in the HP 3000 code segment table for all the segments in the program file, and to allocate an entry in the HP 3000 data segment table for this process data stack.

Often all of these steps are initiated by a single MPE command. (This was illustrated in Figure 2-2.) When necessary, however, you can initiate each step individually, thereby controlling what happens along the way.

Accessing Compilers

Program compilation is the first step in converting a source program to an executable state. The format of the commands used to access a language compiler is consistent for all the MPE language compilers. Thus you do not have to learn a new method of program preparation for each programming language you employ.

Three commands are used in the process of program preparation. The first command compiles the program only and stores the resulting RBMs in a USL file for later use. The second command compiles and prepares the program, creating a program file for use in program execution. The third command compiles, prepares and executes the program.

The compiler name is used in the format of the commands. To illustrate, the commands for accessing the COBOL compiler are:

- COBOL, which compiles the source program
- COBOLPREP, which compiles and prepares the program
- COBOLGO, which compiles, prepares, and executes the program

Access to the other compilers is identical, except that the name "SPL," "BASIC," "RPG," "PASCAL," and "FORTRAN" replace "COBOL" in each of the three commands. The command BASIC invokes the BASIC interpreter, whereas the command BASICOMP invokes the BASIC compiler.

It is important to note that the data files created by these languages are all generated by the same MPE file system. Thus these data files are shareable among the various languages. For instance, a file created by a BASIC program can be read by a FORTRAN program. This file-sharing characteristic also carries down to many HP 3000 subsystems such as KSAM and IMAGE.

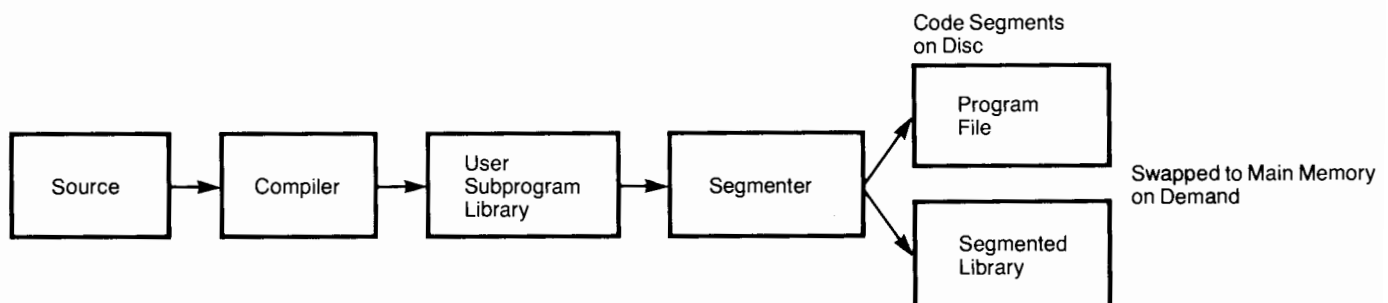


Figure 2-4 Code segment evolution

Segmenter

Program preparation is actually performed by the MPE segmenter in response to a PREP command or one of the combining forms which include PREP or GO. Segmenter commands may be used to manage USL files by adding, deleting, activating, or deactivating libraries (SLs) which are used to resolve external references from user programs.

Occasionally, you may wish to alter the segmentation of a program to improve its run time efficiency. In many other systems the program would have to be recompiled. With MPE, however, the segmentation can be modified by using the segmenter to rearrange RBMs and then preparing the USL file into a new program file.

Another feature of the segmenter allows different versions of the same subprogram to be stored within a single USL file. An optional index capability of the segmenter lets you activate and deactivate entry points within various versions of the subprograms.

Procedure Libraries

When a program is allocated and scheduled for execution, MPE searches the following libraries for unresolved external references:

- The user's log-on group library
- The public group library of the user's log-on account
- The public group library of the system account.

Each library can possess two types of library files: segmented library files, and relocatable library files. Segmented library or SL files contain procedures in segmented form which may be shared between programs. Relocatable library or RL files contain procedures in RBM form which must be prepared before they can be loaded with your program.

Procedures contained in the SL file are in prepared form, that is, they are segmented. When a particular procedure is needed, the segment containing the procedure is loaded, as are all external references from that segment. Because the segmentation has been pre-defined in this manner, these procedures may be shared between programs, and only one copy will exist at any time in virtual memory even though several users may require a particular procedure concurrently.

The combination of segmented libraries and relocatable libraries gives great flexibility for storing often used subroutines and procedures. Procedures used system wide are normally stored in the segmented library at the system level. Procedures used by only a few users or a single group are stored in relocatable or segmented libraries at the group, account, or system level.

Special segmenter commands are provided by MPE which enable you to build an SL or RL within a particular group. In addition, commands are available to add or purge routines, and list the procedures contained in either library.

Code Segmentation

To fully appreciate the programming assistance provided by the segmenter, it is necessary to understand the logic behind program code segmentation. In the HP 3000 computer system, program code is grouped into topical entities which consist of one or more procedures or subroutines. Each code segment may be up to 32 Kbytes in length (where K = 1024). Programs may be broken into multiple segments with procedures or subroutines fully contained within one segment.

A code segment consists entirely of information that is not subject to change during program execution. This includes program instructions and constants. No modifiable data may be interspersed with the instructions in a code segment, and it is only possible to change a code segment by recompiling the modified source. This feature ensures that all code is re-entrant, meaning that any sequence of instructions can be in simultaneous execution by multiple users. All HP 3000 computer system procedures are potentially recursive.

The fact that code segments are not modified during execution has specific advantages for the memory management system. Since all code segments are re-entrant, all are potentially shared by more than one user when present in main memory. For example, operating system services are provided in part by segments contained in a system segmented library shared by all programs which request those services. Similarly, all users executing the same program file share the program's segments. Only one copy of a segment needs to be in main memory, no matter how many users may be executing it concurrently. Thus the system is able to handle more users in a given memory capacity.

Another advantage of re-entrant code segments is that code segments are read-only. Thus they never need to be swapped from main memory back to disc, even when overlaid, because there is always an identical copy of the segment in the program file or library. Code is swapped only into main memory, never out. The resulting reduction in swap traffic leads to more efficient memory management.

File System

One of the main uses of a computer system is information management, i.e., the input, processing, and output of data. MPE manages information by means of its file system. With MPE a file is a body of information or data identified by a user-assigned name. A file may contain commands and/or programs, as well as information, and may be stored on disc, tape, or cards.

MPE also treats peripheral devices as files. Access to such files is device-independent, meaning that a program can read data from a card reader, terminal, magnetic tape, or disc by means of the same request. MPE automatically locates, buffers, transfers, and deblocks the data.

When you ask to read a named disc file, you are only implicitly specifying the disc address of the file; the MPE system determines the explicit address and performs the read. In the same manner, when you ask for a certain type of device by specifying a device class name (disc, line printer, etc.), the file system allocates the actual device for you.

The MPE file system permits sequential access to all files. Disc files with fixed- or user-defined length records may also be accessed randomly. Extensive disc file backup facilities are provided for all types of disc files. The STORE command copies files to a serial storage device; the RESTORE command restores the files to disc.

Files can be accessed from any programming language by means of standard MPE file system intrinsics. A file can be accessed simultaneously by multiple programs; MPE automatically resolves any contention problems which might occur.

MPE file system commands enable programs to reference files without specifying their actual names, addresses, or characteristics. A file can be redefined without a major change to the program. For example, a program's input file named CARDIN designated as a card reader could be changed to a disc file through the use of a FILE command. File specifications can also be altered at run time by means of commands. To illustrate, a program could be coded to open a file with a record length of 128 characters. If at run time it is determined that the file has only 64 characters, you can override the file opening with a FILE command that designates the file to be 64 characters. Reprogramming is not required.

MPE file system commands can be used to build, purge, rename, and display file characteristics. You may specify how disc space is to be dynamically acquired, whether to deal with logical or physical records, whether to include special characters, and so forth.

The user logging facility of MPE allows users and subsystems to record additions and modifications to files. In the event of a loss of a file, the user logging record can be used to recover the data in the file. In addition, the logging file can be used as a record of the activity in that file. The entire user logging facility is implemented through the use of MPE commands and intrinsics designed for this purpose.

The MPE file system is actually a collection of routines which reside in the system segmented library (SL). These routines enable you to open a file, obtain status information, read or write data, perform control functions, and close the file. When a program contains statements or constructions that input or output data, these procedures are brought into play automatically by MPE. The loading operations done by MPE to run your program search the library and establish linkages to allow these routines to be referenced during program execution. The code segments containing these file system procedures are shareable, as are all code segments under MPE, and may be used by several programs at the same time.

Subroutine Compatibility

MPE allows programs written in one language to call subroutines written in another language. Once the subroutine is written in the chosen language, it is stored in the system SL (segmented library). Then, user programs written in FORTRAN, BASIC, COBOL, PASCAL, or SPL can access this subroutine with a standard subroutine call. RPG programs can also execute these SL subroutines via an SPL subroutine call.

This ability to intermingle different programming languages significantly expands programmer productivity and application efficiency. Your programmers are free to program in the languages that they are most familiar with. The different language blocks can then be linked by a master program for execution. This way, segments of programs can be written in the language that is most efficient for the operation being performed. This MPE ability allows you to expand the efficiency and capabilities of your application programs.

File Security

MPE provides two general methods of file security. The first is the use of passwords. The creator of a file can establish passwords (also referred to as lockwords) which must be correctly supplied when anyone makes reference to that file. The second method of file security is the use of file access mode and user type restrictions as outlined in Table 2-2.

The system manager specifies the file access modes allowed for an account and the types of users to whom they are available. The account manager specifies the access modes allowed for a group and the types of users to whom they are available. Finally, the creator of a file specifies the file access modes allowed for the file and the types of users to whom the file is available.

In this manner, access to files can be controlled at several levels which range from unrestricted access (making the file available to anyone), to controlled access (making the file available to its creator only). For example, you can make your data file available to any other user in a "read-only" mode, while only members of your account can append data to the file.

Often a need exists to save general purpose utility programs in public groups or accounts which may then be accessed by all system or account users. MPE provides a special system account named "SYS", and a public group named "PUB" which exists under any account with less-restrictive default security provisions.

Intrinsics

A multitude of additional system functions are available in the form of special MPE procedures, called intrinsics, which may be invoked by calls from your program. Intrinsic calls are acted upon when the segmenter prepares the program containing the intrinsic calls for execution. The segmenter establishes a link between the executing program and the MPE procedure specified by the intrinsic call.

System intrinsics are written in the HP 3000 Systems Programming Language (SPL) and follow the rules and constraints of that language. They may be called from COBOL, BASIC, FORTRAN, PASCAL, or SPL programs, and from RPG programs by way of an SPL subroutine.

There are MPE intrinsics for:

- Opening and closing files
- Reading from, writing to, and managing files
- Controlling devices (such as rewinding magnetic tapes)
- Obtaining file information
- Obtaining user information
- Obtaining detailed error information
- Performing data translation
- Obtaining date and time
- Process handling
- Resource handling
- Data segment handling
- User logging
- Handling software interrupts

Table 2-2 MPE file access modes and user types

Access Modes	
Reading	Allows the user to read files.
Appending	Allows the user to add information and disc extents to the existing files.
Writing	Allows the user to delete or change information already present in existing files
Executing	Allows the user to run programs stored in existing files.
Locking	Provides a logical lock which gives the user exclusive access to a file if desired.
Save Files	Allows the user to save permanent disc files within the user's group.
User Types	
Any User	Makes the specified access modes available to any user of the system.
Account Member	Restricts the specified access modes only to users of the account in which the file resides.
Account Librarian	Restricts the specified access modes only to the account librarian of the account in which the file resides.
Group User	Restricts the specified access modes only to users of the group in which the file resides.
Group Librarian	Restricts the specified access modes only to the group librarian of the group in which the file resides.
Creator	Restricts the specified access modes only to the creator of the file.

Appendix C gives a complete list of all MPE intrinsics with a brief description of each.

When a system intrinsic is invoked to perform a system function, two types of error conditions may occur. MPE informs the calling program of a recoverable error by setting the condition code bits of the HP 3000 status register when the intrinsic is exited. The condition code indicates whether or not the request was granted and what conditions existed pertinent to the request. A request to an intrinsic which requires a special capability class not possessed by the calling program, or which passes illegal parameters to an intrinsic, is considered an irrecoverable error and causes the system to abort the program. In such a case, if you have not specified an appropriate "trap procedure," a batch job is usually re-

moved from the system; an interactive session resumes with a message and a prompt for another command. The CONTINUE command can be used to continue execution of a batch job despite an irrecoverable intrinsic error. Also, by initially calling system trap intrinsics, you may specify special action to be taken in the event of an irrecoverable error.

A Dynamic Environment

The Multiprogramming Executive environment is a dynamic one where programs are run on the basis of processes. A process is the basic executable entity of MPE. It is not a program, but the unique execution of a program by a particular user at a particular time. MPE automatically creates, manages, and deletes processes.

When you execute a program, a private hardware protected data segment called a stack is created for that particular execution. The stack and the program's code segments together constitute the process. To illustrate, when multiple users access the BASIC interpreter, a separate process is created for each of them. They all use the same code, there is only one BASIC interpreter; but each has his own data stack (environment) created by MPE.

The creation, maintenance and deletion of processes is accomplished by means of three MPE components: the virtual memory manager, the job/session scheduler, and the process dispatcher.

Virtual Memory

MPE's virtual memory manager uses both main memory and disc storage to greatly expand the total amount of memory space available. In fact, virtual memory allows programs up to 2 Megabytes in length to be executed in minimal memory systems. MPE logically divides programs into variable length segments of code and data. These segments reside in disc memory and are brought into main memory only when required for program execution, as shown in Figure 2-5. When a code segment is no longer needed, it is overwritten by a new segment. If the code segment is needed again later, it is simply copied again from the disc on which it resides.

Data segments are dynamic and are handled somewhat differently. Since the content of a data segment may change during program execution, a data segment is copied automatically back to the system disc when it is no longer needed, thereby replacing the previous version of that segment.

This approach of segmenting code and data and transferring the segments back and forth between main memory and disc memory, results in the allocation of local storage only as needed. In addition, since program code segments are not modified during execution, multiple users are able to share a single copy of the program code.

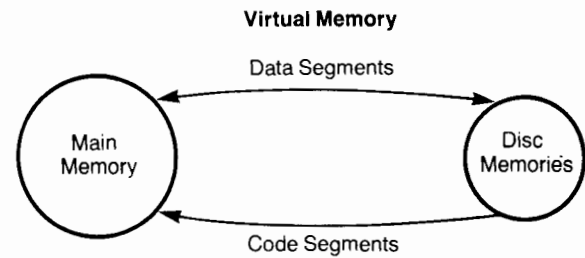


Figure 2-5 MPE virtual memory

A closer look at memory management identifies additional advantages. MPE divides main memory into two areas. The first, fixed memory, contains only those items required to be memory resident such as interrupt handlers, the memory manager, and the scheduler. The remainder of memory, linked memory, contains all other code and data. User and operating system segments are brought into this area by the memory manager as they are required. The architecture allows the operating system, including the file system, the command interpreter, the spooler, and even much of the I/O system, to be shared by all users even though they are brought into main memory only when needed.

The MPE memory manager is responsible for the allocation of main memory to the executing processes. Program and library segments which are needed for execution are automatically swapped into main memory from disc. An attempt by an executing process to access code or data not present in main memory causes the memory manager to allocate main memory space for the missing segment. Other processes may execute while the missing segment is being brought into memory. When the absent segment is swapped from the disc memory to the main memory, the executing process is again eligible for execution. Frequently used segments remain in main memory, and may never be swapped, while rarely used segments are in disc memory most of the time. This results in high efficiency and faster overall execution time. It also creates a dynamic situation in which segments are being swapped rapidly between main memory and disc memory, according to the demands of the executing programs.

Automatic Scheduling

The MPE job/session scheduler schedules jobs and sessions according to their assigned priorities. When the execution of one process is interrupted for any reason, such as I/O, an internal interrupt, or an interrupt from the Scheduler itself, control is passed to the process with the next highest priority which is awaiting CPU resources. When two or more programs have the same priority, the oldest process is selected first.

Jobs and sessions are scheduled by means of a master queue which is ordered by priority as shown in Figure 2-6. This master queue is divided into areas called priority classes. Each area is bounded by two priority numbers established by the system manager.

MPE automatically assigns priority classes to each process executing on the system. The user may, however, specify priority classes by selecting a general category of process dispatching priority for the program. This is done by including the `PRI =` parameter in your `JOB` or `HELLO` command. The five process dispatching priority types (queues) available are:

- AS—system processing only
- BS—very high priority
- CS—interactive
- DS—batch
- ES—very low priority (background)

MPE actually translates priority types into numerical ranges which are ordered in a master queue (see Figure 2-6). The numerical range of each priority type can be changed at any time to ensure that an optimal balance of services is maintained among the processes on the system.

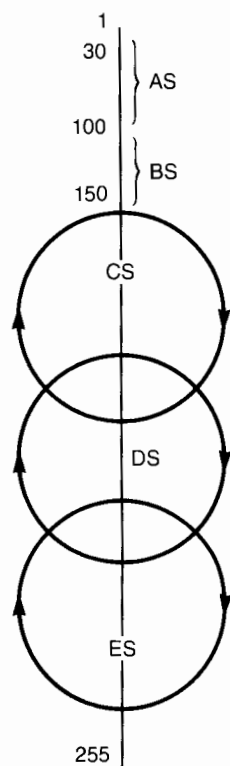


Figure 2-6 MPE master queue structure

Process Execution

The MPE process dispatcher schedules processes for execution. Each process has a dynamically changing priority number, and the dispatcher keeps a list of active processes (those requesting execution) ordered by priority. The dispatcher attempts to execute the highest priority process first. If that process is not in memory, the dispatcher instructs the memory manager to make enough of the process's segments present in memory to allow it to run. The next highest priority process is then chosen for execution while the segments are coming into memory.

As a process runs it may require another code or data segment. If the segment is not present in main memory, the memory manager is instructed to retrieve the segment before the process is allowed to continue executing. While the process waits for the needed segment to be transferred, MPE transfers control to the next process ready to be executed.

In order to bring a process segment into memory, the memory manager first looks to see if there is an available area of memory big enough for the segment. If so, the best fit available region is reserved for the segment and the I/O system will handle the swap. Otherwise, space is made by scanning main memory and making relatively unused portions available. This is done until a space big enough for the segment is found and reserved.

The objective of the process dispatcher and the memory manager is to provide for optimum efficiency in the use of system resources while satisfying the requirements of executing processes. This is done automatically by MPE without assistance from the system users.

System Operation

This section deals with the day-to-day operational aspects of running an HP 3000 installation. For all of its power and features, MPE is surprisingly easy to maintain. General operation of the system is primarily the responsibility of three users: the system manager, system supervisor, and the console operator. The console operator is responsible for routine day-to-day system operations. It is his task to respond to system messages and to keep the system and the peripheral devices functioning smoothly and efficiently. The system supervisor, who is appointed by the system manager, has day-to-day responsibilities in several areas. Included in his duties are the maintenance of the system logging and resource accounting facilities. He also has the capability to retrieve information and change parameters relating to the master scheduling queue. The system manager implements MPE's unique account/group/user organization and appoints account managers to monitor account usage.

System configuration, the first operation undertaken with a new HP 3000, establishes the boundaries under which MPE will function in terms of memory and peripheral devices. An interactive process that takes place with the operator at the system console, system configuration requires intervention only when options other than the system defaults provided are specified, and can be accomplished in a matter of minutes. Once configured, the system is ready for use.

The next operation involves the building of various accounts and the identification of system users. Usually, the first user identified is the system manager who will have overall control of the system. Among his responsibilities is the task of allocating various accounts within the system and identifying an account manager for each. The account managers in turn identify the users who may access the system through their respective accounts.

Once the system is configured and the account and user structure has been defined, the operation of program development begins. Programming may be done interactively on the HP 3000 by means of terminals and the many programming aids provided by MPE for the development of source programs; or programs can be developed in batch processing mode. Once developed, programs are executed as frequently as required.

Occasionally the account manager may need to add new users to their accounts, or it may be necessary to reconfigure the system as more memory or new peripherals are added. But the primary continuing operations are the development of programs and their execution. Figure 2-7 illustrates the interrelation of these general HP 3000 system operations.

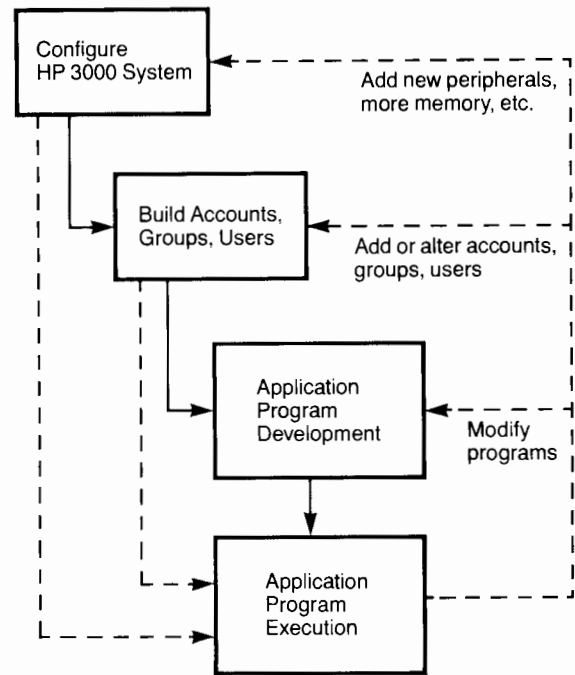


Figure 2-7 HP 3000 operations

Console Operator Function

The console operator function is available to any one user logged onto a non-REMOTE interactive terminal on the system. Operator commands are entered in SESSION mode just as are MPE commands. The console operator has the ability to move the console to other terminal devices, and may selectively allow users to execute specific operator commands. The operator can also allow individual users to have operator control over specific devices. Operator commands deal primarily with peripheral and other hardware system requirements, as well as with system start-up, back-up and recovery procedures.

Start-up and Modification

MPE is initially brought up on an HP 3000 by the operator at the system console, through one of several restart operations:

- **WARMSTART**—The system is restarted from the disc. Spoolfiles are recovered, and incompletely processed spooled jobs are automatically restarted.
- **COOLSTART**—The system is restarted from the disc. The spoolfiles in existence when the system was shut down are not saved, but all resident user files are saved.
- **COLDSTART**—The system is read from a serial storage device (magnetic tape or serial disc). The I/O configuration and system configuration from the serial storage device used to define the system are merged with the directory and files present on the disc.

- **UPDATE**—Similar to cold start, except that the I/O configuration on the disc is used to define the system. This start-up mode is used to define the system. This start-up mode is used when starting the system from an updated version of MPE supplied by Hewlett-Packard.
- **RELOAD**—The entire system, directory, files, and configuration information are read from a serial storage device.

The restart operation can include an interactive dialogue between the console operator and the MPE initiator program. This optional dialogue permits the operator to change the system configuration. Upon completion, MPE is operational.

System Backup and Recovery

Periodically, the MPE system and user files are copied on a serial storage device for back-up purposes. The serial storage device is then available for reloading the system in the event of a hardware or software failure, or to transfer the system to another hardware installation.

In the latter case, a new I/O and system table configuration may be specified during an interactive dialogue. Areas which may be changed include:

- I/O devices
- System table sizes
- System disc allocation
- Logging facility
- Scheduling changes
- Segment size limits
- System modules to be allocated or made memory resident

The SYSDUMP command (which requires System Supervisor capability) may be used to create media for reloading on a different or changed HP 3000. A new I/O configuration and system table configuration may be specified during the interactive dialogue which occurs. A back-up date may also be specified which enables the operator to back-up only those disc files which change each day.

Power Fail/Auto Restart

When an AC power failure is detected by the HP 3000, a power fail/auto restart routine is automatically invoked. This routine preserves the operating environment prior to complete loss of power. Normal system operation resumes as soon as power is restored. Jobs and sessions in progress on the system continue where they were interrupted, with programs unaware of the interruption (except for magnetic tape units and dial-up terminals in use when the failure occurred, or if the power outage lasts longer than the built-in system memory battery backup).

In addition to the system configuration aids mentioned above, MPE provides a range of software aids specifically designed for use by the system operator. Among these are:

- Spooling facility
- Tape label facility
- Disc options.

Spooling Facility

This MPE facility permits the concurrent usage of devices which would otherwise be non-shareable, such as card readers, magnetic tape drives, or line printers.

This is accomplished by copying the input or output to disc where it is processed later. This procedure is called spooling. (SPOOL is an acronym for Simultaneous Peripheral Operations On-line.)

To illustrate, if six users need to produce output on a line printer at approximately the same time, their output is directed to spoolfiles on disc from which the output is printed on a priority basis as the line printer becomes free. In this way each user can immediately proceed with other processing activities without having to wait for the line printer. Similarly, if there are ten jobs to be read from a card reader or magnetic tape unit, they are all read immediately and are directed to spoolfiles on disc where they wait to be executed. Thus, the card reader or magnetic tape unit is not tied up by one job which must be executed before the others can be read.

The spooling of batch job input can be initiated not only from a card reader or magnetic tape unit, but also from within an interactive session as described earlier.

Tape Labeling Facility

MPE provides a tape labeling facility for use in reading and writing labels on magnetic tapes. The facility can be used to:

- Identify magnetic tape volumes (reels).
- Protect tape volumes from being inadvertently written over.
- Protect private information.
- Facilitate information interchange between computer systems.

The facility can be used to read, but not write, IBM-standard tape labels; read and write ANSI-standard labels; and read and write user-defined labels on previously labeled magnetic tapes.

Disc Options

The MPE operating system includes a serial disc interface which allows non-system domain drives to be used as non-shareable serial devices. To MPE, the discs appear to be magnetic tape drives. They provide a fast system backup and recovery capability when used as an alternative to magnetic tape in backing up the MPE system and storing and restoring files and/or data bases.

MPE also provides a private disc volume facility which allows you to create and access files on removable disc volumes. Private volumes consist of removable disc packs which, when mounted on a disc drive, can be accessed by MPE through the file system. Under private volumes, the disc packs mounted on the drives during a cold load are dynamically allocated to the system domain for normal use or to the non-system domain for private use. Non-system domain packs can be both physically and logically mounted and dismounted during normal system operation.

System Account Structure

The primary responsibility of the system manager is to create and maintain the organizational structure of accounts, groups, and users under which access to MPE occurs. The account structure provides maximum security and control by permitting the system manager to assign specific access and system usage capabilities to each user.

"Accounts" are collections of users and groups. Each account has a unique name and an optional password assigned to it when the system manager creates the account. Each account also has its own file domain or unique set of files. The system manager may define resource-use limits for an account. MPE maintains a running count of each resource that the account uses. MPE also stores a list of user name and group names recognized by the account, the maximum job priority at which jobs in the account may be scheduled, and limits established on the account's usage of disc file space, CPU time, and connect time.

"Groups" are used to partition the file domain of an account. Files must be assigned to a group, and each group has a unique name (within the account) and optional password. Limits may be established on the permanent disc space, CPU time, and connect time used by a group. MPE maintains running counts of resource usage for each group and the sum of these group counts always equals that of the account in total.

"Users" are individuals who access the HP 3000. Each user is assigned a unique name and optional password, and is assigned to a specific account. Each user may have a specified home group of files, and may access any other file groups in the account. A maximum job priority may be assigned to each user.

Each account "owns" a unique set of files separate and distinct from every other account. This ownership is indirect in that only groups may own files directly. Thus, every file belongs to a group, every group belongs to an account, and every account belongs to the system.

To illustrate how accounts, groups, and users interrelate, consider the following example. Figure 2-8 represents a system which includes interactive terminals dispersed throughout a company. The system manager has assigned three accounts: Marketing, Engineering, and Finance. The marketing account manager has defined two users who can access the system: Bill and Dave. Each user has his private group (assigned as his home group) where he stores his private programs and files. Bill and Dave can also access programs and files stored in the other groups in the account. A group named PROJ1 was created to contain programs and data files related to current projects. An administrative group was also created to contain administrative work such as schedules and budgets. The public group, to which no password was assigned, contains general purpose utility programs for use by all.

Bill can log on to the HP 3000 from a terminal with the command:

```
:HELLO BILL.MKTG
```

By default, Bill now has access to all programs and data files in his home group: BILLSGRP. Bill can gain access to a file in the PROJ1 group by using the fully qualified file name which specifies both the account and group under which the file was created. The file Bill wants is data file 1 so he enters:

```
FORECAST.PROJ1.MKTG
```

Alternatively, Bill could have logged on to the HP 3000 and requested access to all programs and files in the PROJ1 group by appending the group name to his log on request, as follows:

```
:HELLO BILL.MKTG,PROJ1
```

Bill now has access to all files in the PROJ1 group.

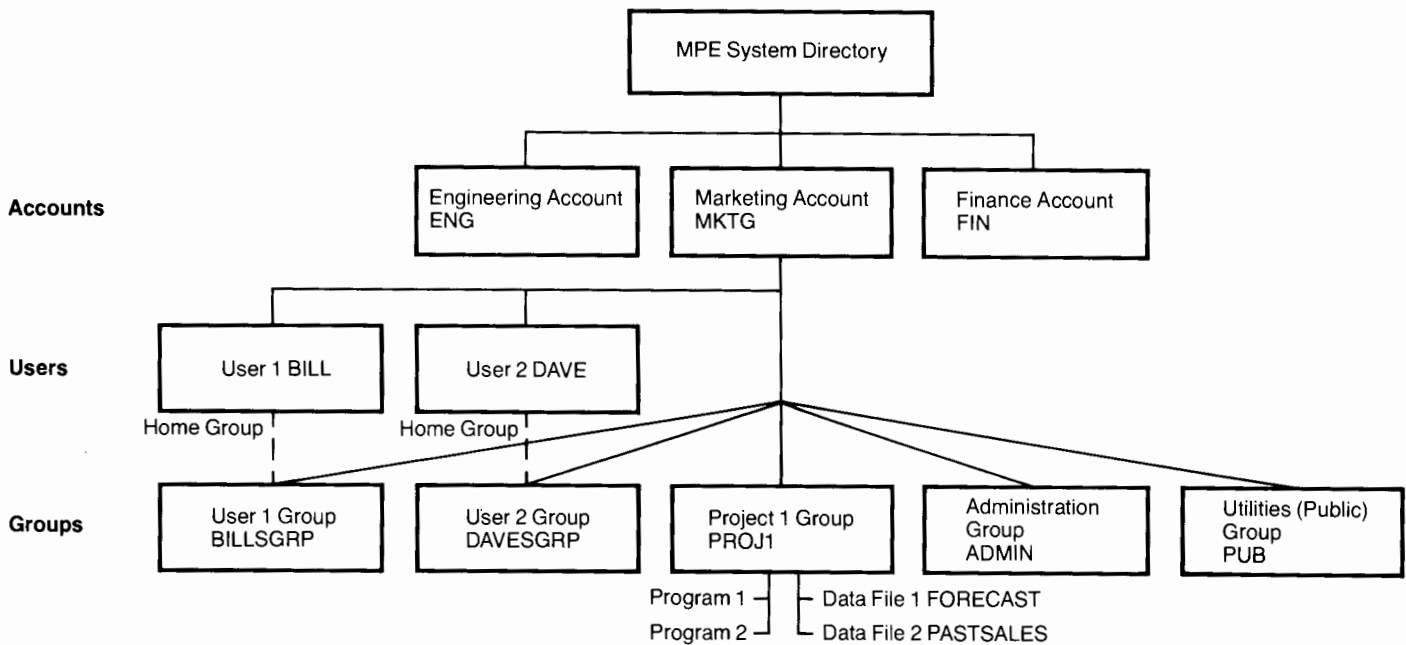


Figure 2-8 MPE account structure

To summarize, you can log on the system using only your assigned user and account names, in which case you are automatically given access to your home group. Or, you can log on specifying your user name, account name, and a group name which gives you access to the group you specify whether or not it is your home group.

To access a file in a group other than the one specified when you logged on, you use the fully qualified name of the file which can consist of the file name, the group

name, the account name, and the appropriate passwords when required.

As you can see, the account structure provides both control and security over file use. Access to the system is granted only to individuals with a valid log-on identification consisting of account, group, and user names, each of which may require a password. Figure 2-9 illustrates both an unsuccessful and successful log-on procedure where passwords are required.

```

:HELLO MANAGER.FINANCE,SALES
ACCOUNT PASSWORD (PASS)?

```

```

USER PASSWORD (PASS)?

```

```

GROUP PASSWORD (PASS)?

```

```

INCORRECT PASSWORD. (CIERR 1441)

```

```

:HELLO MANAGER.FINANCE,SALES
ACCOUNT PASSWORD (PASS)?

```

```

USER PASSWORD (PASS)?

```

```

GROUP PASSWORD (PASS)?

```

```

HP3000 / MPE IV C.00.00. TUE, MAY 5, 1981, 4:02 PM

```

The password is typed but not displayed to ensure privacy.

Access denied after incorrect password is entered. To log on you must reenter the HELLO command and passwords, if prompted.

The password is typed but not displayed to ensure privacy.

The user now has access to all HP 3000 resources as defined by his user capability set.

Figure 2-9 MPE system access security.

System Supervisor

The user with system supervisor capabilities can use commands which enable him to control the master scheduling queue, permanently allocate or deallocate a procedure or program to virtual memory, and manage the logging facility.

He can control the master scheduling queue in two ways: first by entering the SHOWQ command which displays information about the scheduling of processes and the contents of the master queue and various subqueues; and second, by entering the TUNE command which allows changes to the time quantum or limits defining the bounds of any subqueue.

Two important commands, ALLOCATE and DEALLOCATE, allow him to tune the system by permanently allocating a procedure or program in the HP 3000 to virtual memory. This significantly decreases the time needed for an operation with large, frequently used routines.

System Logging Facility

The MPE logging facility records details of system resource requests in a series of log files on disc and can be used to monitor system resource usage. The system supervisor selects those system and user events that are to be recorded. Log records are provided for job and session initiation or termination, program termination, file closing, file spooling completion, and system shutdown. Log files can be used in the generation of precise billings based on accurate system usage records.

I/O device failures on any device are also recorded in log files which can then be used to detect problems before they begin to interfere significantly with overall system operation.

Accounting Facility

The MPE accounting facility provides a flexible and powerful means of coordinating access to the system and disc file usage. To coordinate system access, system administrators can devise a structure of accounts and users which reflects the functional organization of the people who use the system. The accounting facility maintains running totals on the amounts of system resources that each account consumes, including disc space used, cumulative CPU time consumed, and cumulative terminal connect time for sessions. The current totals can be displayed at any time and can be used for billing purposes.

File usage can be coordinated also because the overall permanent disc file domain of the HP 3000 is partitioned among the various accounts. Each account's file domain is further partitioned into groups. If a request to save a file would result in exceeding the permanent file space limit at either the account or group level, the request is denied.

Users may create files only in the account and group under which they are currently running. By using fully-qualified file names users also have access to any file present in the system, provided that existing security provisions allow them access.

Interprocess Communication Facility

The MPE interprocess communication facility allows independent executing processes to pass information back and forth efficiently via disc files. The facility uses standard file system intrinsics as well as additional intrinsics to handle software interrupts. The facility provides security for the passed data via the MPE security system.

Performance Measurement

On-Line Performance Tool/3000 (OPT/3000)

The On-line Performance Tool/3000 (OPT/3000) is a performance measurement package which consists of two interrelated products: OPT/3000 Performance Measurement Software and the OPT/3000 System Performance Training Course. This Package is designed to help HP users gather and utilize performance data for system management, capacity planning, and application development activities.

On-Line Performance Software

On-line Performance Tool/3000 is an interactive performance measurement software product for the HP 3000 that provides information to the system analyst. OPT/3000 can be used to characterize the current system workload, CPU utilization, memory management activity, I/O traffic, program and process activity, and system table usage to help the user isolate bottlenecks and improve system performance. Performance data provided by OPT/3000 is continuously updated at regular intervals and can be presented in charts, graphic displays, or summary reports. Information can be displayed on an HP terminal and a hard copy of any display can be generated on a line printer with a single keystroke. Although OPT/3000 is primarily designed for interactive use, it can be executed in batch mode to collect snapshots of system activity over a period of time.

Display Contexts

OPT/3000 can generate 24 unique displays containing system performance information. These displays are grouped into six categories called display contexts. Each context is associated with a different type of system resource.

These six display contexts are:

- Global
- Memory
- CPU-Memory Manager
- I/O
- Process
- System Tables

Within each context, displays are available at successively greater levels of detail. This structure allows the user to progress from summary level information to more detailed information as required.

Features

- **Interactive Terminal Reporting of Performance-Related Data:** OPT/3000 provides summary and detailed CPU, memory, I/O, and process information in dynamically updated terminal displays. The displays are updated under the control of the user and may be updated automatically or by simply pressing the return key.
- **Graphical Presentation of Information:** OPT/3000 utilizes the features of most HP 26xx series of terminals to generate displays with a graphical format. The terminal video enhancements used include blinking, inverse video, underlining, and half-bright. The line drawing character set and the cursor addressing capabilities are also used.
- **Multiple Display Levels:** The displays associated with a particular context are layered such that each level provides the user with more detailed information than the previous level.
- **Summary Reporting of Resource Usage:** Although OPT/3000 is primarily designed for interactive use, it can be executed in batch mode to collect summary information about system activity. These reports can also be generated interactively and can be used to provide data for capacity planning activities.
- **Hard Copy Capability:** A hard copy of any display can be generated on the line printer with a single keystroke.
- **Logging Capability:** The information used to generate the summary reports can also be logged into a disc file. This data can be accessed by user programs for reformatting and reporting at a later time.
- **Low Overhead:** The performance impact of OPT/3000 running on an HP 3000 is low, varying from two to seven percent of available CPU time. As a result, multiple users of OPT/3000 on a system can be executing simultaneously with little impact on system performance.
- **Extensive On-Line Help Facility:** With this integrated facility, documentation explaining any command or context display can be quickly and easily displayed. In many cases, interpretation guidelines are provided to aid in the identification of performance problems.

System Performance Training Course

The OPT/3000 System Performance Training Course teaches users how to use OPT/3000 software to generate performance related data and interpret the results. The course shows system analysts how to identify performance bottlenecks, characterize workloads, collect information for capacity planning, analyze system table configurations, and tune the performance of individual programs. This training is required with every initial installation of OPT/3000.

Application Program Sampler (APS)/3000

Application Program Sampler/3000 (APS/3000) is a friendly interactive performance measurement software product for tuning application programs on the HP 3000, with the goal of increasing the overall system performance.

APS/3000 helps the programmer to identify CPU bottlenecks in the source code of application programs. APS/3000 does this by monitoring the execution of application programs using a status sampling technique, and producing histograms showing the relative CPU time spent by various portions of the monitored program. Since an application program frequently calls for the services of the operating system, such as intrinsics and library calls, APS/3000 reports the CPU time spent in system services as well as the CPU time spent directly in the user code. APS/3000 can monitor the single execution of a program or the multiple execution of shared programs.

APS/3000 can be used either during program development or on existing programs. In a typical application, the user monitors a particular program, studies APS/3000's histograms, and optimizes the code which consumes the most CPU time. This process is repeated until the programmer believes that the software performance is acceptable. APS/3000 can thus simulate the judgement of an experienced programmer in identifying CPU software bottlenecks, resulting in increased programmer productivity. Although APS/3000 is intended primarily for interactive use, it can also be executed in batch mode. The histograms can be displayed on any HP terminal and a hard copy of the display generated on a line printer for future study.

Features

- **Interactive Menu Screens** that make the user interface very friendly. The user selects commands from a menu and is then prompted for special parameters. Depending on the command, APS/3000 generates a display, moves to a new menu, or displays appropriate messages.

- **Hierarchical Display Levels** that permit the user to analyze programs at different levels of detail, with each level providing the user with more detailed information than the previous level. The histograms are produced in terms of the logical structures of the source program, such as segments, procedures, and address regions relative to procedures or segments. Such an organization also allows for easy interpretation of the histograms.
- **Direct and Indirect CPU Utilization Histograms** that allow for better optimization of application programs. The histograms report the relative distribution of the CPU time, spent either directly in the user code or indirectly in exercising the system services. The user can also obtain the estimated distribution of wait times.
- **On-line Display of Measurement in Progress.** As a measurement progresses, APS/3000 can provide histograms of data currently being collected.
- **Display Enhancements.** The "inverse video," "inverse half-bright," and "underline video" capabilities of HP terminals are utilized by APS/3000 but are not required for accurate interpretation of results.
- **Logging Capability.** Samples taken by APS/3000 during the measurement process can be logged into a disc file for later detailed analysis.
- **Hard Copy Capability.** A hard copy of any display can be generated on a line printer.
- **Low Overhead.** APS/3000 has a low performance overhead, varying from 2—3% of available CPU time with the default sampling rate. The sampling rate can be adjusted by the user.
- **On-line Help Facility.** Documentation explaining currently accessible commands is quickly and easily displayed. Dialogue verbosity can be set to three different levels.

Flexible Disccopy

Flexible Disccopy/3000

Flexible Disccopy/3000 provides a method of converting IBM 3741 format flexible disc data sets (files) to HP 3000 disc files and translating EBCDIC character code to ASCII. It can operate in either an interactive environment or in batch mode, and can convert either single or multiple data sets and volumes. (Each IBM 3741 flexible disc is called a volume.) A complete error, warning, and status message file is included, and with the MPE Operating System provides the user and console operator with messages about program status, user prompts, and error conditions.

System Architecture

Stack Architecture

Separation of Code and Data

Processes

Variable-Length Segmentation

Code Segmentation

Data Stack

Registers

Virtual Memory

Microprocessor

Microcode

Instructions

Stack Architecture

The HP 3000 with its hardware stack implementation is referred to as a stack machine. A stack is a linear storage area for data. It is so named because data items are placed on the "top," "pushing down" the data items already present. Data items are removed from the top, "popping up" those data items remaining. If you have ever worked with a Hewlett-Packard calculator that has an "ENTER" key, then you have worked with a stack. Consider the following calculation:

$$\frac{6 * 10}{2 * (3 + 12)}$$

Using a stack, no temporary intermediate values need to be named or stored in registers until required later in the computation. An example of how this problem would be evaluated in a stack is presented in Figure 3-1. Note that the top of stack (TOS) moves downward in keeping with the HP 3000 conventional representation. All operations can be performed without naming specific operands, as the top two stack data values are implicitly used.

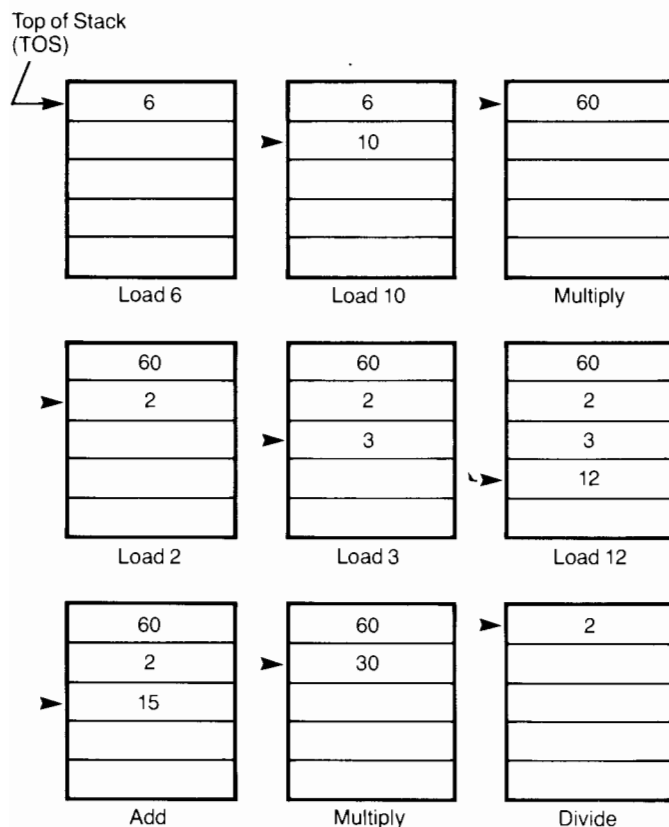


Figure 3-1 Stack Evaluation of $(6 * 10) \div [2 * (3 + 12)]$

The benefits of a stack architecture are numerous. First, as seen above, the storage allocation is dynamic. Local storage is allocated only upon entry of a procedure and is automatically freed upon exit so that areas of memory are not tied-up and can be re-used by other parts of the program. Temporary storage of intermediate values is automatically provided. Thus compilers do not have to be concerned with saving and restoring registers for intermediate results.

Code compression is made possible by the omission of operands in many of the instructions on a stack machine. No extra registers are required for subroutine parameters and temporary variables. A register machine requires 50 to 100 percent more bits for code than a stack machine.

The subroutine is one of the most important concepts in software. Modular, structured programming has as its principal idea the partitioning of a large program into many small, understandable modules which can be called as subroutines. The best mechanism for subroutine calls and execution is the stack, because all subroutine return addresses, I/O parameters, and local variables can be pushed onto the stack. This leads to easy parameter passing and provides highly efficient subroutine linkage.

Fast execution on the HP 3000 is a benefit of having several CPU registers to hold the top part of the stack, rather than leaving it all in main memory. The HP 3000 provides each user with hardware protection of his data stack. Rapid interruption and restoration of user environment is made possible by storing the operating environment in a special block as an extension to the user's stack.

Separation of code and data

In most computer systems, programs consist of an intermixing of instructions and data. For example, within a subroutine there are program locations reserved by the compiler for return addresses of other subroutines and space set aside for the storage of local variables.

The HP 3000 approach separates a program into those elements that do not need to be altered and those that do. The result is that an HP 3000 program consists of a separate code area and data area (data stack), as illustrated in Figure 3-2.

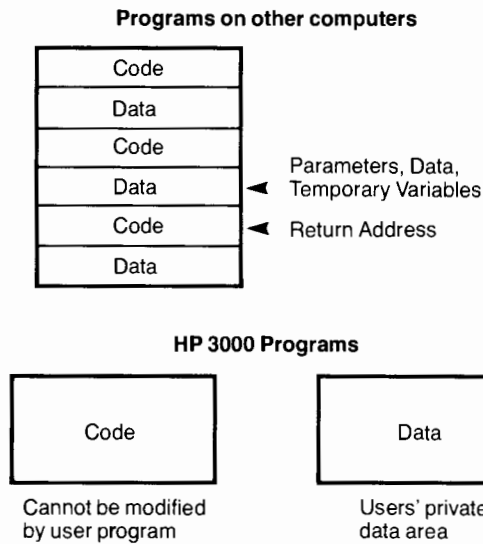


Figure 3-2 Separation of Code and Data

Code consists of the executable instructions that make up a program or subprogram. The values and arrays used by the program or subprogram are referred to as data. Code cannot be altered by your program as there is no instruction available in the HP 3000 instruction set to allow such a modification. However, since you, and you alone, must be able to manipulate your data, the system provides you with a unique, private, modifiable data area (data stack).

In the HP 3000, code and data are maintained in strictly separate domains and cannot be intermixed (with the exception, however, that program constants may be present in code segments). This fact, plus the fact that code is nonmodifiable while active in the system, permits code to be shareable among several users. HP 3000 code is also re-entrant. Re-entrant means that when a program is interrupted during execution of a code segment and another user's execution needs the same segment, that segment can be used, is completely protected against modification, and will be returned intact to the previous user's execution. Since re-entrant code allows one copy of heavily used programs to be shared by many users concurrently, main memory can be used with optimum efficiency. Re-entrant code and stack-structured data together make possible subprogram recursion (a subprogram calling itself), a capability which is essential for efficient compilers and system software. Also, since code is non-modifiable, exact copies of all active code can be retained on disc, thus allowing code to be overlayed without having to first write it back out to the disc.

Processes

Programs are run on the basis of processes created and handled by the operating system. The process is not a program itself, but the unique execution of a program by a particular user at a particular time. If the same program is run by several users, or more than once by the same user, it forms part of several distinct processes.

The process is the basic executable entity. It consists of a process control block (PCB) that defines and monitors the state of the process, a dynamically-changing set of code segments, and a data area (stack) upon which these segments operate. Thus, while a program consists of data in a file and instructions not yet executable, a process is an executing program with the data stack assigned. The code segments used by a process can be shared with other processes, but its data stack is private.

For example, each user working on-line through the BASIC language is running his program under a separate process; all use the same code (the only copy of the BASIC interpreter in the system) but each has his own stack.

Processes are invisible to the programmer. In normal operation you have no control over processes or their structure. However, optional capabilities are available to permit you to create and manipulate processes directly.

Variable-length segmentation

Variable-length segmentation of code and data is used to facilitate multiprogramming. This method, in comparison with paging schemes, minimizes "checkerboard" waste of memory resources due to internal fragmentation. It also makes it possible for the operating system to deal with logical instead of physical entities. This means, for example, that a particular subprogram can always be contained within one segment rather than arbitrarily divided between two physical pages, thus minimizing the amount of swapping that needs to be done while executing that subprogram. The location and size of all executing code segments is maintained by MPE in a code segment table while the location and size of all associated data segments is maintained by MPE in a data segment table. These tables are known to both hardware and software. Software uses them for dynamic memory management by the operating system. Hardware uses them to perform references and transfers between segments and to make sure that all seg-

ments required for current execution are present in main memory. Code segments may be up to 32,760 bytes in length. Data segments may be up to 65,528 bytes in length.

Segments are stored on disc and are brought into main memory only when needed. This design results in a virtual memory environment which appears to be many times larger than the maximum size of the physical main memory. It should be noted that virtual memory in MPE does not contain code segments—only the data segments which must be swapped. The code segments stay in their original positions, anywhere on the discs.

Code segmentation

Code segmentation allows you to divide your program into several segments using control statements to the compiler at compilation time or commands to the MPE operating system. Then the operating system takes over the management of these segments. That is, MPE determines whether or not a code segment should be in main memory or in disc memory. Thus you can write programs much larger than the available main memory of your HP 3000. In the example in Figure 3-3, a program has been divided into five code segments. Code segments 1, 4 and 5 are in main memory while code segments 2 and 3 are in disc memory. The code segment table (CST), which is resident in main memory, points to the active code segments. If the code segment is in main memory, the code segment table points to the beginning of that code segment. If the code segment is on disc, the code segment table points to its disc address. The management of code segments and the code segment table is completely transparent to your program. When a subroutine call is made from one code segment to the next, the instruction that makes the call examines the code segment table to determine whether or not the new code segment is in main memory. If it is, control is transferred immediately to that code segment. If the code segment table indicates that the required segment is on disc, then the instruction interrupts the operating system and informs it that a required code segment is not in memory. It is then up to the operating system to make space for that code segment in memory so execution of the process can continue.

You have control over segmentation; that is, you determine where the program is divided. The result is that segmentation can be tailored to your program's logic. The size of segments can be optimized to memory size and you can avoid thrashing (unnecessary swapping of code segments due to poor segmentation—across a DO loop, for example).

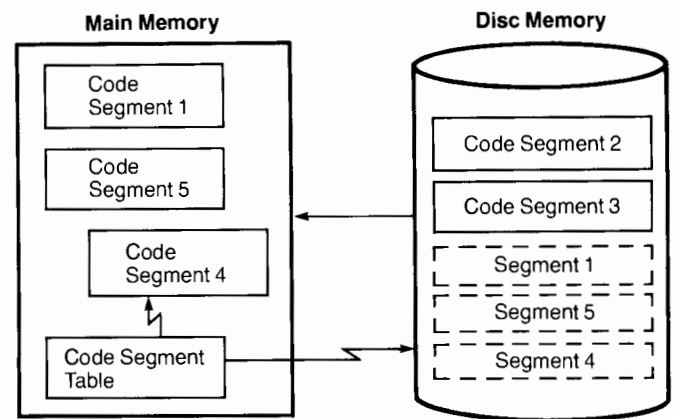


Figure 3-3 Code Segmentation

Data stack

The data area of an HP 3000 program consists of a data stack, an area of main memory that expands and contracts as the program requires. In all programs there is a certain amount of global or common data that is required throughout the life of the program. This global area represents the absolute minimum size of the data stack. Beyond this, the data area grows to meet the needs of subroutines as they allocate storage for their own working areas. When a subroutine has finished its job, this area can be cut back to make use of the same memory space for the next subroutine. The end result is that less memory is required for program execution.

In general, a stack is a storage area in which the last item entered is usually the first item removed. In actual use, however, programs have direct access to all elements in the stack by specifying addresses relative to several CPU registers (the DB, S and Q registers). The stack structure provides an efficient mechanism for parameter passing, dynamic allocation of temporary storage, efficient evaluation of arithmetic expressions, and recursive subprogram calls. In addition, it enables rapid context switching to establish a new environment on subprogram calls and interrupts. In the HP 3000, all features of the stack (including the automatic transferring of data to and from the CPU registers and checking for stack overflow and stack underflow) are implemented in the hardware.

When programming in a high-level language such as COBOL or RPG, all manipulation of the stack is automatically done for you by the language processor. You can, however, manipulate the stack directly by writing subprograms in SPL (Systems Programming Language for the HP 3000).

Figure 3-4 illustrates the general structure of a data stack as viewed from a subprogram. The white area represents filled locations, all containing valid data, while the shaded area represents available unfilled locations.

You can see that the contents of the data stack fall into four general areas. They are the global data required by all subroutines during the execution of a program, parameters that are passed to subroutines, the local data area required by the currently active subroutines, and the temporary storage required for the evaluation of expressions and intermediate results. The remaining area of the stack is unused and represents the amount of expansion possible in the stack without operating system intervention.

The stack area is delimited by the locations defined as DB (data base) and S (stack pointer). The addresses DB and S are retained in dedicated CPU registers. The Q-minus relative addressing area contains the parameters passed by the calling program. The area between Q and S contains the subprogram's local and temporary variables and intermediate results.

The data in the DB location is the oldest element on the stack. The data in the S location is the most current element. The location S is also referred to as the top of stack or TOS. Conventionally, the top is shown in diagrams downward from DB; this corresponds to the nor-

mal progression of writing software programs where the most recently written statement is farther down the page than previous (older) ones.

The area from S + 1 to Z (the shaded area) is available for adding more elements to the stack. When a data word is added to the stack, it is stored in the next available location and the S pointer is automatically incremented by one to reflect the new TOS. This process is said to push a word onto the stack. To delete a word from the stack, the S pointer is simply decremented by one, thus putting the word in the undefined area.

To refer to recently stacked elements of data, S-minus relative addressing is used. Under this condition, S - 1 is the second element on the stack, S - 2 is the third, and so on. S-minus relative addressing is one of the standard addressing conventions. The others are DB-plus relative addressing and Q-minus and Q-plus relative addressing (the Q-register separates the data of a calling program or subprogram from the data of a called subprogram).

Since the top elements of the stack are the most frequently used, there are several CPU registers which may at various times contain the topmost stack elements. The use of CPU registers in this way increases the execution speed of stack operations by reducing the number of memory references needed when manipulat-

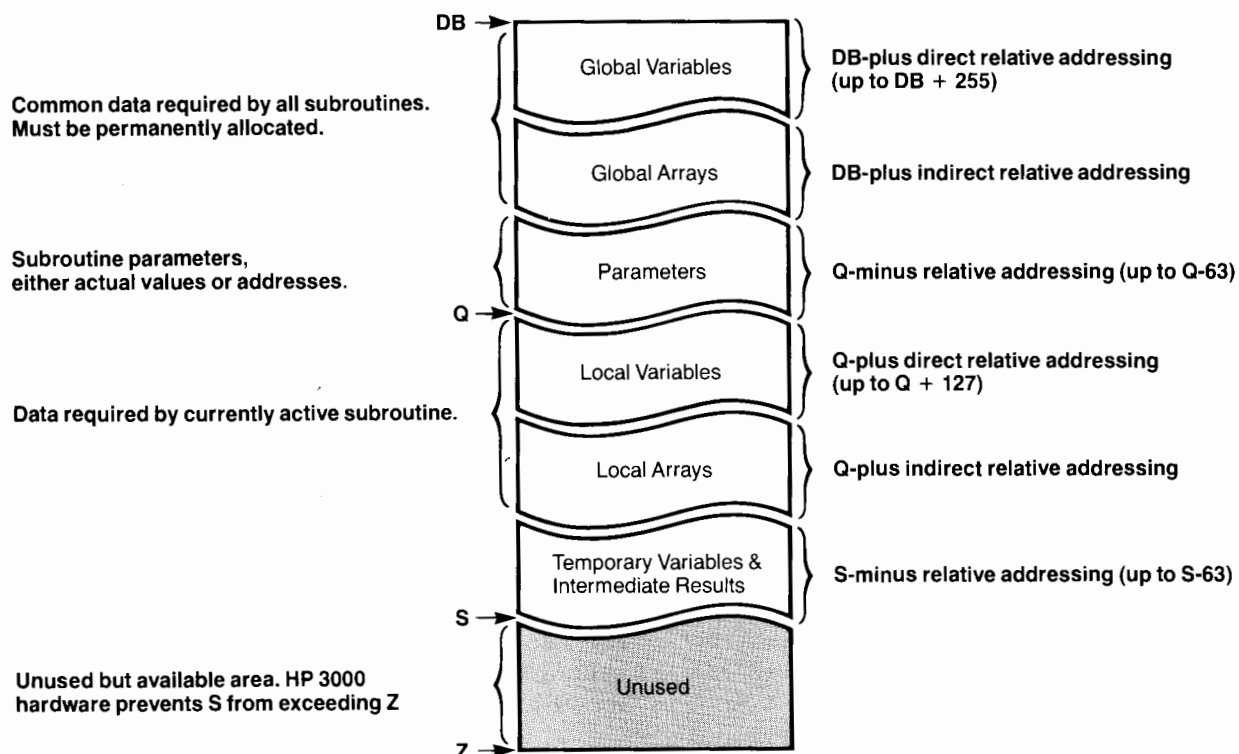


Figure 3-4 Data Stack Contents

ing data at or near the top of the stack. These registers are implicitly accessed by many of the machine instructions and whenever the top stack locations are specifically referenced. Data stacks are automatically expanded by the operating system during execution up to a maximum size of 64 Kbytes ($K = 1024$).

Registers

The architecture of the HP 3000 employs a set of specific purpose registers rather than a set of general purpose registers. Each register is included in the system to efficiently perform a single specific function.

All addressing of code and data is done relative to hardware address registers. Thus by simply changing the base addresses in these registers, segments are dynamically relocatable in memory. The few instances where absolute addresses are required are privileged operations handled by the operating system.

Approximately one-half of the HP 3000 registers are accessible to user programs and/or the operating system and its related software. The remaining registers are used, for example, by the interrupt system and for microprogram processing. The registers for the Series 68 are summarized in Appendix D, the registers for the Series 48 are summarized in Appendix F, and the registers for Series 39 and 42 are summarized in Appendix G of this General Information Manual.

Virtual memory

Virtual memory is a very efficient memory management scheme which, in addition to main (semiconductor) memory, uses disc storage as secondary memory. Users' program code and data are divided into variable-length segments which reside in secondary memory. As a program is being executed, only those segments of code and data which are required at a particular time actually reside in main memory; the other related segments remain on disc until they in turn are required. When a particular code segment is no longer needed, it is simply overlaid by another segment.

This can be done because in the HP 3000, code segments are nonmodifiable and re-entrant. When the segment is needed again, it is simply copied from the disc on which the program resides. Since programs are copied into main memory directly from disc memory, they need not be copied prior to execution to a special "swapping disc." Data segments, however, are dynamic, and their contents can change during execution. Therefore, when a particular segment is no longer needed, it is automatically copied back to the system disc (replacing the previous version of that segment on the disc), and the main

memory space of that segment is then available for other segments. The process of transferring segments between secondary memory and main memory is referred to as swapping. Whenever a segment is referenced, the hardware checks to see whether it is in main memory; if it is not, the operating system is invoked to bring it in. Thus the management of the virtual memory is totally automatic and transparent, and the system can reference a virtual memory space far larger than the real memory available.

Microcoded Processor Architecture

The HP 3000 Central Processing Unit is microcoded. This means the HP 3000 instruction set is broken down further into a series of microinstructions a group of which represents each macroinstruction. This microcode is stored in the Control Store of each CPU. On the Series 39, 42 and 48 this Control Store is implemented in permanent Read Only Memory (ROM) which is part of the CPU hardware. On the Series 68 the Control Store is implemented as Writeable Control Store (WCS) and is loaded from the Cold Load Storage device. Control Store never changes during the operation of the machine.

Microcode

On the Series 39, 42 and 48 the microcode is organized as a series of 48-bit microinstructions. On the Series 68 the microcode width is 64-bits which controls two closely-coupled processors.

In addition to the instruction set, many system operations that in the past were programmed in software have been microcoded. These operations are requested by machine instructions that each, in turn, execute multiple microinstructions built into the central processor hardware. Some of the standard system functions which have been microprogrammed include the interrupt handler, a cold-start loader, the saving of critical environment information on power failure, automatic restart upon restoration of power, and a set of microdiagnostics that can be invoked from the front panel of the system.

The microprogrammed instructions routinely check for addressing bounds violations during execution and automatically interrupt to error handling routines if violations occur. These memory protection checks are usually overlapped with the operand fetch and therefore do not slow execution.

Instructions

There are over 200 unique HP 3000 instructions. Many of these instructions have multiple actions, several addressing modes, indirect addressing, and/or indexing which give a high complexity-to-instruction ratio. Code compression is achieved through the use of no-address (stack) instructions which implicitly use the contents of the stack registers as operands. Machine instructions are in an 8, 16, and 32-bit format. Stack operation instructions are 8-bit and may be packed two per 16-bit word to further enhance code density.

A complete set of arithmetic instructions provides integer (16-bit two's complement), double integer (32-bit two's complement), logical (16-bit positive integer), 28-digit packed decimal (BCD coded digits packed two per byte), floating-point (32 bits including a 23-bit precision mantissa), and extended precision floating-point (64 bits including a 55-bit precision mantissa) arithmetic.

Other instructions are designed to facilitate string processing, subprogram linkage, and loop control. Certain special instructions are designated as privileged, meaning that they were designed specifically for use by the operating system. They may, however, be used by programs which the installation permits to run in privileged mode. Some of these special instructions, such as the DISP instruction for entry to the MPE dispatcher, instructions for enabling/disabling process switching, and instructions for data transfers between data segments, contribute greatly to the efficiency of the operating system. Complete machine instructions are provided in Appendix E for the Series 68 and Appendix H for the Series 39, 42, and 48.

A

Appendix A: System Documentation

**Overview**

General Information Manual 5953-7553	Guide for the New User 32033-90009
---	---------------------------------------

HP 3000 Hardware

Pre-Series 64 Site Preparation Set 30000-60029	Pre-Series 64 Site Planning and Preparation Guide 30000-90206	Pre-Series 64 Site Planning Workbook 30000-90207	HP Computer Supplies 5953-2450
---	--	---	-----------------------------------

Series 64/68 Site Preparation Set 30140-60085	Series 64/68 Site Planning and Preparation Guide 30140-90011	Series 64/68 Site Planning Workbook 30140-90012	Machine Instruction Set 30000-90022
--	---	--	--

System Usage

MPE V Commands Reference Manual 32033-90006	MPE V Intrinsic Reference Manual 32033-90007	Segmenter Reference Manual 30000-90011	MPE V Utilities Reference Manual 32033-90008
--	---	---	---

EDIT/3000 03000-90012	FCOPY/3000 03000-90064	SORT/3000 32214-90001	Debug/Stack Dump Reference Manual 30000-90012
--------------------------	---------------------------	--------------------------	--

MPE V Quick Reference Guide 30000-90049	File System Reference Manual 30000-90236
--	---

System Administration

MPE V System Operation and Resource Management Reference Manual 32033-90005	Communications Handbook 30000-90105
--	--

Additional System Software

OPT/3000 32238-90001	Flexible Discopy/3000 32199-90001	APS/3000 32180-90001
-------------------------	--------------------------------------	-------------------------

B

Appendix B: MPE Commands

All MPE commands are summarized below, grouped alphabetically within capability. The last column denotes when the command can be issued: during a batch job (J), during a session (S), during a break (B), or programmatically (P), through the COMMAND intrinsic.

Standard Capability Commands

Command	Function	When issued
:() command log on	Begins a session, executes the enclosed MPE command, and ends the session upon completion of the command.	S
:ABORT	Aborts the current program.	B
:ALTLOG	Alters the attributes of an existing logging identifier.	J,S,B,P
:ALTSEC	Changes security provisions for a file	J,S,B,P
:APL	Accesses the APL subsystem.	J,S
:ASSOCIATE	Gives a user operator control of a device	J,S,B,P
:BASIC	Calls BASIC/3000 interpreter.	J,S
:BASICGO	Compiles, prepares, and executes a BASIC/3000 program.	J,S
:BASICOMP	Compiles a BASIC/3000 program.	J,S
:BASICPREP	Compiles and prepares a BASIC/3000 program.	J,S
:BUILD	Creates a new file.	J,S,B,P
:BYE	Terminates a session.	S
:COBOL	Compiles a COBOL/3000 program.	J,S
:COBOLGO	Compiles, prepares, and executes a COBOL/3000 program.	J,S
:COBOLPREP	Compiles and prepares a COBOL/3000 program.	J,S
:COMMENT	Inserts comment into command stream.	J,S,B,P
:CONTINUE	Disregards job-error condition.	J
:DATA	Defines data from outside standard input stream. Cannot be read on \$STDINX file. Acceptable for device recognition.	J,S
:DEBUG	Invokes the MPE debug facility.	S,P
:DISASSOCIATE	Removes the control of a device from a user	J,S,B,P
:DISMOUNT	Causes a volume set that was mounted by the user to be dismounted.	J,S,B
:DSLIN	Opens or closes communication line with DSN/DS.	J,S
:DSTAT	Displays the current status of the disc drives on the system.	J,S,B,P
:EDITOR	Calls the EDITOR.	J,S
:ELSE	Provides an alternate execution sequence for an IF statement.	J,S,B
:ENDIF	Terminates an IF block.	J,S,B
:EOD	Denotes end of data. Cannot be read on \$STDINX file.	J,S
:EOF	Simulates hardware end-of-file on input stream from any device	J,S
:EOJ	Denotes end of batch job. Cannot be read on \$STDINX file.	J
:FCOPY	Invokes the FCOPY facility.	J,S
:FILE	Defines or redefines a file's characteristics.	J,S,B,P
:FORTGO	Compiles, prepares, and executes a FORTRAN/3000 program.	J,S
:FORTPREP	Compiles and prepares FORTRAN/3000 program.	J,S
:FORTRAN	Compiles a FORTRAN program.	J,S

Command	Function	When issued	Command	Function	When issued
:FREERIN	Deallocates a global RIN, and returns it to RIN pool.	J,S	:RELEASE	Temporarily suspends all security provisions for a file.	J,S,B,P
:GETLOG	Establishes a logging identifier on the system	J,S,B,P	:RELLOG	Removes a logging identifier from the system	J,S,B,P
:GETRIN	Acquires a global RIN.	J,S,B,P	:REMOTE	Establishes communication between a local computer and a remote computer.	S
:HELLO	Initiates a session. Acceptable for device recognition. Requires 1A capability class.	S	:RENAME	Renames a file.	J,S,B,P
:HELP	Access the HELP subsystem.	J,S,B	:REPORT	Displays total accounting information for a log-on group.	J,S,B,P
:IF	Used to control the execution sequence of a job.	J,S,B	:RESET	Resets a formal file designator.	J,S,B,P
:IML	Initiate DSN/IMF Pass-Through Mode		:RESETDUMP	Disables the MPE stackdump facility.	J,S,B,P
:JOB	Initiates a batch job. Requires BA capability class.	J,S	:RESTORE	Restores a complete fileset, stored off-file.	J,S,B,P
:LISTF	Lists descriptions of files	J,S,B,P	:RESUME	Resumes an interrupted program.	B,S
:LISTLOG	Lists active logging identifiers	J,S,B,P	:RJE	Calls the DSN/RJE Subsystem.	J,S
:LISTVS	Produces a formatted listing of volume set definition information.	J,S,B,P	:RPG	Compiles an RPG/3000 program.	J,S
:MOUNT	Requests the console operator to mount a volume set.	J,S,B	:RPGGO	Compiles, prepares, and executes an RPG/3000 program.	J,S
:MRJE	Initiates execution of the DSN/Multi-leaving Remote Job Entry (MRJE) facility.		:RPGPREP	Compiles and prepares an RPG/3000 program.	J,S
:PREP	Prepares a compiled program into segmented form.	J,S	:RUN	Loads and executes a program.	J,S
:PREPRUN	Prepares and executes a program.	J,S	:SAVE	Changes a file to permanent status. Requires SF capability for saving files.	J,S,B,P
:PTAPE	Reads a paper tape without X-OFF control.	S,B,P	:SECURE	Restores suspended security provisions for a file.	J,S,B,P
:PURGE	Deletes a file from the system.	J,S,B,P	:SEGMENTER	Calls MPE Segmenter.	J,S
:RECALL	Displays all pending console REPLY messages.	J,S,B,P	:SETCATALOG	Causes the command interpreter to search a catalog of user-defined commands and to establish a directory entry for each command in the catalog.	J,S,B
:REDO	Allows the user to edit a command entry.	S,B			

Command	Function	When issued	Command	Function	When issued
:SETDUMP	Enables the MPE stackdump facility on abort.	J,S,B,P	:STREAM	Spools batch jobs or data in session or job mode.	J,S,B,P
:SETJCW	Scans the JCW table for a specified JCW name and updates the value of this JCW.	J,S,B,P	:TELL	Transmits a message.	J,S,B,P
:SETMSG	Disables or enables receipt of user or operator messages at standard list device.	J,S,B,P	:TELLOP	Transmits a message from the user to the computer operator.	J,S,B,P
:SHOWCACHE	Displays statistics on the behavior of MPE disc caching.	J,S,B,P	:VINIT	Accesses the VINIT subsystem to perform on-line conditioning and formatting of serial discs and private volumes.	J,S
:SHOWCATALOG	Lists user-defined command (UDC) files.	J,S,B	:VSUSER	Prints a listing of all users of a currently mounted volume set.	J,S,B
:SHOWDEV	Reports status of input/output devices.	J,S,B,P	Standard Capability Commands (Segmenter Commands)		
:SHOWIN	Reports status of input device files.	J,S,B,P			
:SHOWJCW	Displays the current state of a job control word.	J,S,B,P	Command	Function	When issued
:SHOWJOB	Displays job/session status.	J,S,B,P	-ADDRL	Adds a procedure to an RL.	J,S
:SHOWLOG-STATUS	Displays status information about currently opened log files.	J,S,B,P	-ADDSL	Adds a segment to an SL.	J,S
:SHOWME	Reports job/session status.	J,S,B,P	-ADJUSTUSLF	Adjusts directory space in a user subprogram library (USL) file.	J,S
:SHOWOUT	Reports status of output device files.	J,S,B,P	-AUXUSL	Designates a source of RBM input for COPY command	J,S
:SHOWTIME	Displays current date and time-of-day.	J,S,B,P	-BUILDRL	Creates a permanent, formatted RL file.	J,S
:SPEED	Changes input speed or output speed of terminal.	S,B,P	-BUILDSL	Creates a permanent, formatted SL file.	J,S
:SPL	Compiles an SPL/3000 program.	J,S	-BILDUSL	Creates a temporary, formatted USL file.	J,S
:SPLGO	Compiles, prepares, and executes an SPL/3000 program.	J,S	-CEASE	Deactivates one or more entrypoints in a USL.	J,S
:SPLPREP	Compiles and prepares an SPL/3000 program.	J,S	-CLEANSL	Copies the currently managed SL to a new SL file, removing inactive segments	J,S
:STORE	Stores a set of files off-line.	J,S,B,P	-CLEANUSL	Copies the currently managed USL to a new USL file, removing inactive segments	J,S
			-COPY	Copies an RBM or segment from one USL to another.	J,S

Command	Function	When issued
-COPYSL	Same as CLEANSL except allows user to expand SL space by a given percentage	J,S
-COPYUSL	Same as CLEANUSL except allows user to expand USL space by a given percentage	J,S
-EXIT	Exits from Segmenter, returning control to MPE command interpreter.	J,S
-EXPANDUSLF	Changes length of a USL file.	J,S
-HIDE	Sets an RBM internal flag on.	J,S
-INITUSLF	Initializes buffer for a USL file to the empty state.	J,S
-LISTRL	Lists the procedures in an RL.	J,S
-LISTSL	List specific segments in the SL.	J,S
-LISTUSL	List specific segments in USL.	J,S
-NEWSEG	Changes the segment name of an RBM.	J,S
-PREPARE	Prepares RBMS from a USL into a program file.	J,S
-PURGERBM	Deletes one or more RBMs from a USL.	J,S
-PURGERL	Deletes an entrypoint or a procedure from an RL.	J,S
-PURGESL	Deletes an entrypoint or a segment from an SL.	J,S
-REVEAL	Sets an RBM internal flag off.	J,S
-RL	Designates an RL for management.	J,S
-SL	Designates an SL for management.	J,S
-USE	Activates one or more RBM entrypoints.	J,S
-USL	Designates a USL for management.	J,S

System Manager Capability Commands

Command	Function	When issued
:ALTACCT	Changes an account's characteristics.	J,S,B,P
:ALTVSET	Modifies volume set definitions.	J,S,B,P
:LISTACCT	Lists attributes of an account.	J,S,B,P
:NEWACCT	Creates a new account.	J,S,B,P
:NEWVSET	Defines private volume sets and classes.	J,S,B,P
:PURGEACCT	Removes an account and users from the system's or the volume set's directory.	J,S,B,P
:PURGEVSET	Deletes an existing volume set.	J,S,B,P
:REPORT	Displays an account's resource usage.	J,S,B,P
:RESETACCT	Resets resource-use counters for an account and its groups.	J,S,B,P

Account Manager Capability Commands

Command	Function	When issued
:ALTGROUP	Changes a group's attributes.	J,S,B,P
:ALTUSER	Changes a user's attributes.	J,S,B,P
:LISTACCT	Lists attributes of user's log-on account.	J,S,B,P
:LISTGROUP	Lists attributes of a group in user's log-on account.	J,S,B,P
:LISTUSER	Lists attributes of a user in log-on account.	J,S,B,P
:NEWGROUP	Creates a new group in log-on account.	J,S,B,P
:NEWUSER	Creates a new user in log-on account.	J,S,B,P
:PURGEGROUP	Removes a group from the system's or the volume set's directory.	J,S,B,P
:PURGEUSER	Deletes a user from log-on account.	J,S,B,P
:REPORT	Displays resource-usage counts for log-on account and its groups.	J,S,B,P

System Supervisor Capability Commands

Command	Function	When issued
:ALLOCATE	Loads a program or procedure.	J,S
:CACHECONTROL	Changes global caching parameters.	J,S,B,P
:DEALLOCATE	Removes a program or procedure from virtual memory.	J,S,P
:IMLMGR	Starts interactive IML Managers' program	J,S
:JOBPRI	Sets or changes the priority for batch jobs or sessions.	J,S,B,P
:RESTORE	Returns files to the system.	J,S,B,P
:RESUMELOG	Resumes logging following suspension caused by an error.	J,S,B,P
:SHOWLOG	Displays log file status.	J,S,B,P
:SHOWQ	Displays scheduling subqueue information.	J,S,B,P
:STARTCACHE	Initiates operation of disc caching.	J,S,B,P
:STOPCACHE	Terminates operation of disc caching.	J,S,B,P
:STORE	Stores disc files into magnetic tape or serial disc.	J,S,B,P
:SWITCHLOG	Closes the current log file, and creates and opens a new log file.	J,S,B,P
:SYSDUMP	Starts configurator dialog and copies MPE to magnetic tape or serial disc.	J,S
:TUNE	Changes scheduling parameters for processes	J,S,B,P

Master Operator Commands

Command	Function
:ABORTIO	Aborts pending I/O requests for a device.
:ABORTJOB	Aborts a job or session.
:ACCEPT	Permits a device to accept job/sessions and/or data.
:ALLOW	Grants a user access to a specific operator command

Master Operator Commands

Command	Function
:ALTJOB	Alters attributes of waiting job or session.
:ALTPOOLFILE	Alters attributes of output spooling files.
:BREAKJOB	Suspends an executing job.
:CONSOLE	Changes the system console from its current device to another job-accepting (non-DS) terminal
:DELETESPOOL-FILE	Deletes a spooled device file.
:DISALLOW	Prohibits a user access to a specified operator command.
:DISCRPS	Enables or disables rotational position sensing (RPS) on CS 80 disc drives.
:DOWN	Removes a device from normal system use.
:DOWNLOAD	Downloads information to an output device.
= DSLINE	Enables or disables the data communications link under control of the DS/3000 subsystem.
:GIVE	Assigns a DOWNed device to the diagnostics.
:HEADOFF	Stops header/trailer output to a device.
:HEADON	Resumes header/trailer output to a device.
:JOBFENCE	Defines input priorities.
:JOBSECURITY	Controls the availability of certain job commands to a user.
:LDISMOUNT	Logically dismounts a private volume set/class (UV capability required).
:LIMIT	Limits the number of concurrently running jobs/sessions.
:LMOUNT	Logically mounts a private volume/class on a non-system domain disc drive.
:LOG	Starts, restarts, stops User Logging (LG capability required).
= LOGOFF	Aborts all jobs/sessions and prevents further log-ons by all except HIPRI jobs/sessions.

Command	Function
= LOGON	Enables job/session processing following a LOGOFF command.
= MPLINE	Enables or disables the data communications link under control of the DSN/MTS subsystem.
= MRJE	Enables or disables the data communications link under control of the DSN/MRJE subsystem.
: OPENQ	Opens the spool queue for the specified logical device or device class.
: OUTFENCE	Defines priorities for output spooled files.
: REFUSE	Disallows jobs/sessions and/or data on a designated device.
: REPLY	Replies to a pending console request.
= REPLY	Same as :REPLY
: RESUMEJOB	Resumes a suspended job.
: RESUMESPOOL	Resumes a spooled device.

Command	Function
= SHUTDOWN	Closes down the operating system.
: SHUTQ	Closes the spool queue for the specified logical device or device class.
: STARTSPOOL	Initiates spooling of a device.
: STOPSPPOOL	Terminates spooling of a device.
: STREAMS	Enables or disables the users' ability to submit job/session and/or data streams.
: SUSPENDSPOOL	Causes a spooled device to stop operation.
: TAKE	De-assigns a device that was GIVEN to the diagnostics.
: UP	Allows a DOWNed device to function again.
: VMOUNT	Enables or disables the private volumes facility.
: WARN	Sends an urgent message to jobs and sessions.
: WELCOME	Defines the message users receive when they log on the system.



Appendix C: MPE Intrinsic



All MPE intrinsics (system procedures) are summarized below, listed alphabetically. Any special capabilities required to call a particular intrinsic are noted.

Intrinsic	Function
ACCEPT	Accepts (and completes) the request received by the preceding GET intrinsic call.
ACTIVATE	Activates a process. (Requires PH capability.)
ADJUSTUSLF	Adjust directory space in a USL file.
ALTDSEG	Changes the size of an extra data segment. (Requires DS capability.)
ARITRAP	Enables or disables internal interrupt signals from all hardware arithmetic traps.
ASCII	Converts a number from binary to ASCII code.
BINARY	Converts a number from ASCII to binary code.
CALENDAR	Returns the calendar date.
CAUSEBREAK	Requests a session break.
CLOCK	Returns the time of day.
CLOSELOG	Closes access to the logging facility.
COMMAND	Executes an MPE command programmatically.
CREATE	Creates a process. (Requires PH capability.)
CTranslate	Converts a string of characters from EBCDIC to ASCII or from ASCII to EBCDIC.
DASCII	Converts a value from double-word binary to ASCII code.
DBINARY	Converts a number from ASCII to double-word binary value.
DEBUG	Sets breakpoints and modifies or displays stack or register contents.
DLSIZE	Changes size of DL-DB area.
DMOVIN	Copies block from data segment to stack. (Requires DS capability.)
DMOVOUT	Copies block from stack to data segment. (Requires DS capability.)
EXPANDUSLF	Changes length of a USL file.

Intrinsic	Function
FATHER	Requests PIN of father process. (Requires PH capability.)
FCARD	Drives the HP 7260A Optional Mark Reader.
FCHECK	Requests details about the file input/output errors.
FCLOSE	Closes a file.
FCONTROL	Performs various control operations on a file or terminal device.
FDEVICECONTROL	Performs various control operations on a spoolfile associated with a 2680 Laser Printer.
FERRMSG	Returns message corresponding to FCHECK error number.
FGETINFO	Requests access and status information about a file.
FINDJCW	Searches the job control word table for a specified job control word (JCW).
FLOCK	Dynamically locks a file.
FMTCALENDAR	Converts the calendar date obtained with the CALENDAR intrinsic.
FMTCLOCK	Converts the time of day obtained with the CLOCK intrinsic.
FMTDATE	Converts calendar date and time of day obtained with the CALENDAR and CLOCK intrinsics.
FOPEN	Opens a file.
FPOINT	Resets the logical record pointer for a sequential disc file.
FREAD	Reads a logical record from a sequential file.
FREAD-BACKWARD	Reads a logical record on a tape drive from a point in front of the current record pointer.
FREADDIR	Reads a logical record from a direct-access disc file.
FREADLABEL	Reads a user file label.
FREADSEEK	Prepares, in advance, for reading from a direct-access file.

Intrinsic	Function
FREEDSEG	Releases an extra data segment. (Requires DS capability.)
FREELOCRIN	Frees all local RINs from allocation to a job.
FRELATE	Declares a file pair interactive or duplicative.
FRENAME	Renames a disc file.
FSETMODE	Activates or deactivates file access modes.
FSPACE	Spaces forward or backward on a file.
FUNLOCK	Dynamically unlocks a file.
FUPDATE	Updates a logical record residing in a disc file.
FWRITE	Writes a logical record to a sequential file.
FWRITEDIR	Writes a logical record to a direct-access disc file.
FWRITELABLE	Writes a user file label.
GENMESSAGE	Accesses the message system.
GET	Receives the next request from the remote master program.
GETDSEG	Creates an extra data segment. (Requires DS capability.)
GETJCW	Fetches contents of job control word.
GETLOCRIN	Acquires a local RIN.
GETORIGIN	Determines source of process activation call. (Requires PH capability.)
GETPRIORITY	Reschedules a process. (Requires PH capability.)
GETPRIVMODE	Dynamically enters privileged mode. (Requires PM capability.)
GETPROCID	Requests PIN of a son process. (Requires PH capability.)
GETPROCINFO	Requests status information about a father or son process. (Requires PH capability.)
GETUSERMODE	Dynamically returns to non-privileged mode. (Requires PM capability.)
INITUSLF	Initializes a buffer for a USL file to the empty state.
IODONTWAIT	Initiates completion operations for an I/O request. (Continue execution if I/O not completed.)

Intrinsic	Function
IOWAIT	Initiates completion operations for an I/O request. (Wait for I/O to complete.)
JOBINFO	Returns job or session related information.
KILL	Deletes a process. (Requires PH capability.)
LOADPROC	Dynamically loads a library procedure.
LOCKGLORIN	Locks a global RIN.
LOCKLOCRIN	Locks a local RIN.
MAIL	Tests mailbox status. (Requires PH capability.)
MYCOMMAND	Parses (delineates and defines parameters) for user-supplied command image.
PAUSE	Suspends the calling process for a specified number of seconds.
PCHECK	Returns an integer code specifying the completion status of the most recently executed slave program-to-program intrinsic.
PCLOSE	Terminates the remote slave program's process.
PCONTROL	Transmits a tag field to the remote slave program and receives a tag field back from the slave.
POPEN	Initiates and activates a slave process in a remote HP 3000 and initiates program-to-program communication with the slave program.
PREAD	Sends a read request to the remote slave program asking the slave to send a block of data back to the master.
PRINT	Prints character string on job/session list device.
PRINTFILEINFO	Prints a file information display on the job/session list device.
PRINTOP	Prints a character string on operator's console.
PRINTOREPLY	Prints a character string on the operator's console and solicits a reply.
PROCTIME	Returns a process's accumulated central-processor time.

Intrinsic	Function
PTAPE	Accepts input from tapes not containing X-OFF control characters.
PUTJCW	Puts the value of a particular job control word (JCW) in the JCW table.
PWRITE	Sends a block of data to the remote slave program.
OPENLOG	Provides access to a logging facility.
QUIT	Aborts a process.
QUITPROG	Aborts a program.
READ	Reads an ASCII string from an input device.
READX	Reads an ASCII string from an input device.
RECEIVEMAIL	Receives mail from another process. (Requires PH capability.)
REJECT	Rejects a request received by the preceding GET intrinsic call and returns an optional tag field back to a remote master program.
RESETCONTROL	Resets a terminal to accept a CONTROL-Y signal.
RESETDUMP	Disables the abort stack analysis facility.
SEARCH	Searches an array for a specified entry or name.
SENDMAIL	Sends mail to another process. (Requires PH capability.)

Intrinsic	Function
SETDUMP	Enables the stack analysis facility.
SETJCW	Sets bits in job control word.
STACKDUMP	Dumps selected parts of stack to a file.
SUSPEND	Suspends a process. (Requires PH capability.)
SWITCHDB	Switches DB register pointer. (Requires PM capability.)
TERMINATE	Terminates a process.
TIMER	Returns system-timer bit-count.
UNLOADPROC	Dynamically unloads a library procedure.
UNLOCKGLORIN	Unlocks a global RIN.
UNLOCKLOCRIN	Unlocks a local RIN.
WHO	Returns user attributes.
WRITELOG	Writes a record to a logging file.
XARITRAP	Arms the software arithmetic trap.
XCONTRAP	Arms or disarms the CONTROL-Y trap.
XLIBTRAP	Arms or disarms the software library trap.
XSYSTRAP	Arms or disarms the system trap.
ZSIZE	Changes size of Z-DB area.



Appendix D:
HP 3000 Series 68
Hardware Features

This appendix provides a summary of hardware features of the HP 3000 Series 68 computer system. For more details on the HP 3000 architecture, refer to Chapter 1; System Introduction, and Chapter 3; System Architecture.

The HP 3000 Series 68 is designed as a full function computer, providing the processing power and capabilities to handle a full range of EDP and Distributed Data Processing. Based on a modular concept, the HP 3000 Series 68 allows independent elements to be interconnected through a newly designed, powerful multiple I/O bus structure. These elements consist of a CPU module with dual arithmetic logic units (ALUs), cache memory, main memory, I/O Adaptors, General I/O Channels, and DSN/Advanced Terminal Processors. Communication between modules is accomplished using a high speed Central System Bus and up to two Intermodule Buses. The system also includes a system console, system display panel, and a Diagnostic Control Unit (DCU). Peripheral devices are connected to the system through the General I/O Channels. Interactive terminals are at-

tached to the system through the DSN/Advanced Terminal Processor or DSN/Intelligent Network Processors running the DSN/Multipoint Terminal Software. Data communication links are established via DSN/Intelligent Network Processors.

The CPU is centered around a Hewlett-Packard designed microcoded processor using high speed Emitter Coupled Logic (ECL) technology and a dual arithmetic logic unit (ALU). This implementation provides the highest performance ever offered in an HP 3000 while allowing flexibility in the machine instruction set. The Series 68 also employs high speed, semiconductor, random access main memory modules which use automatic fault detection and correction.

The hardware design of the HP 3000 Series 68 provides the ability to expand the system as the user's needs grow and applications change. The system software accommodates these changing needs by allowing additional hardware modules and peripheral devices to be easily configured on the system.

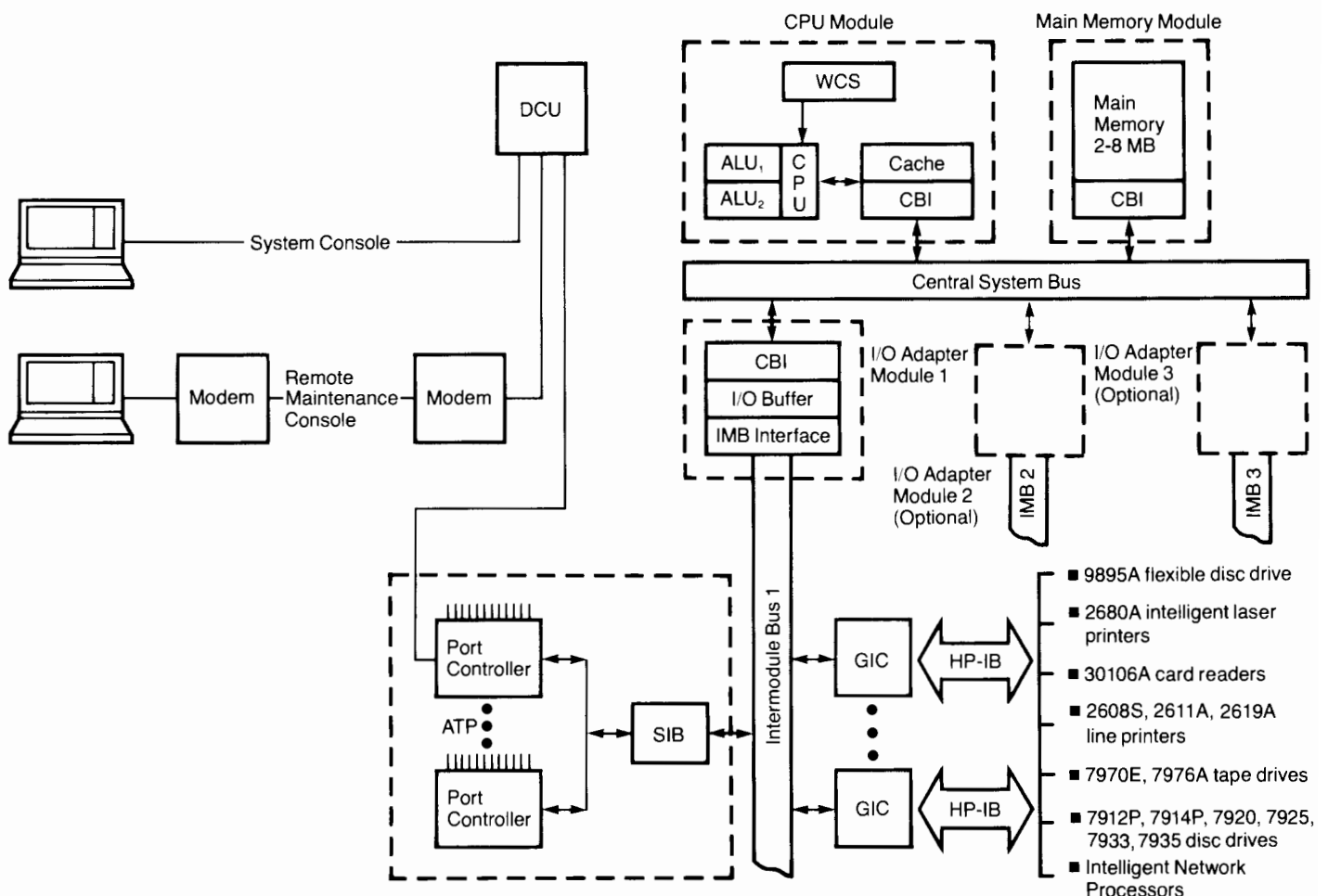


Figure D-1. HP 3000 Series 68 Hardware Organization

Central Processing Unit (CPU)

The significant features of the HP 3000 Series 68 central processing unit (CPU) are listed in Table D-1.

**Table D-1 HP 3000 Series 68
CPU Features**

Architecture
<ul style="list-style-type: none"> ■ Hardware-implemented stack ■ Separation of code and data ■ Variable-length code segmentation ■ Non-modifiable re-entrant code ■ Virtual memory for code ■ Dynamically relocatable programs
Implementation
<ul style="list-style-type: none"> ■ Microprogrammed Emitter Coupled Logic (ECL) CPU ■ 75 nanosecond microinstruction cycle time ■ 8 Kbyte cache memory ■ Dual 16-bit arithmetic logic units (ALUs) ■ 60 Kbyte Writeable Control Store (WCS) ■ Automatic restart after power failure ■ Central System Bus ■ Overlapping CPU and I/O operations
Instructions
<ul style="list-style-type: none"> ■ Over 214 powerful instructions ■ Instructions are 8, 16 and 32 bits in length ■ 16 and 32 bit integer arithmetic ■ 32 and 64 bit floating point arithmetic ■ 28 digit packed decimal arithmetic ■ Special instructions that optimize the efficiency of the operating system.

The CPU converts an instruction in the current instruction register (CIR) into a starting address for the microcode contained in the Writeable Control Store (WCS) and determines various initial conditions required for executing the instruction. As the current instruction is being executed, the next instruction is fetched and placed in the next instruction register (NIR). Upon completion of the current instruction, the contents of NIR are loaded into CIR and the cycle is repeated. This "pipelining" of the current instruction execution with the next instruction-fetch improves throughput by overlapping operations.

The HP 3000 Series 68 instruction set is presented in Appendix E. Instructions are 16 or 32 bits in length except stack operations, which are 8-bit instructions. These include a variety of memory reference, branch, arithmetic and data manipulation instructions that operate on integer, real, logical, packed decimal, character and string data. Floating point arithmetic can be performed in single precision (32 bits) or double precision (64 bits), integer arithmetic in 16-bit and 32-bit lengths, and packed decimal instructions extended to 28 digits in precision. In addition, there are a number of instructions designed to aid in creating the multi-programming environment of the system. These include procedure call and exit instructions and others which implement various operating system functions previously done in software.

Firmware storage and control consists of microcode stored in WCS, and associated logic control. All of the microcode except for a 256 word bootstrap is contained in the WCS. At initial power on, the bootstrap PROM in the Diagnostic Control Unit (DCU) is enabled, and the microcode contained in it is executed. Under operator control from the system console, this microcode loads the system microcode into the WCS and then branches to the starting point in the WCS. The WCS contains all of the microcode to implement the instruction set. Microdiagnostics can also be loaded into the WCS and run by the DCU. It is possible to load microdiagnostics before loading the instruction set to verify the integrity of the machine before the system begins execution. Microcode routines control the operation of the instruction decoder and the hardware processor in order to create the HP 3000 operating environment. The microinstruction cycle time is 75 nanoseconds.

The hardware processor consists of two arithmetic-logic units, a shift register, specific purpose registers, and related data manipulating and testing logic. A unique dual ALU design is used to increase arithmetic processing power. Using this design, the Series 68 CPU is able to perform two 16-bit, or a single 32-bit operation in a single CPU cycle. Performance is increased as fewer microinstruction cycles are required. Since the HP 3000 architecture (see chapter 3) is structured on code segments and data segments, most of the CPU registers are used for defining the segment limits and operating elements within the segments. Table D-2 lists all registers and their associated functions.

The eight top of stack registers are of special interest. In order to improve execution speed, up to seven elements from the top of a data stack may be contained in these registers. This allows many functions to be treated as register-to-register operations rather than the slower speed memory-to-register or register-to-memory type operations. These registers are manipulated by the CPU, and their use is fully transparent.

Table D-2 Series 68
Hardware Registers

Register	Function	Register	Function
PB P PL BNKP	Code Segment Pointers	PDB CAR	Cache Memory Address and Data Registers
BNKD BNKS DL DB Q SM Z		NAMER ENAMR	
RA RB RC RD RE RF RG RH	Stack Pointers	CSAR RAC RAR	Top of Stack Address Register
X STAT PERF CIR NIR		RRGA SRGA RRGB SRGB	
OPA OPB		BKX3-BKX7 SP0-SP4 CPX1-CPX2 F1-F5 SR ESR CTRS XTRA0-XTRA127 XTRB0-XTRB127	
			Firmware Registers
			ALU Registers
	Top of Stack Registers		Scratch Pad, Flag, Interrupt Registers, and Counters
	Index Register Status Register Performance Register Current Instruction Register Next Instruction Register		
	Cache Operand Register		

Cache memory

The Series 68 uses a hierarchical memory to reduce the time required to access memory. The hierarchical memory consists of a main memory module and an 8 kbyte cache memory.

Cache memory, acting as a high speed buffer between the CPU and main memory, increases system performance by decreasing memory access time. High speed ECL random access memories (RAMs) used in the cache allow it to supply a word to the CPU in 75 nsec, or one CPU cycle. However, only a fraction of main memory can be in the cache at one time. With the design of the Series 68 memory system, the CPU will on average find

the word in the cache 95% of the time. The other 5% of the time, the word required by the CPU resides in main memory and the CPU must wait while it is read. The net result is the fastest average memory access time of any HP 3000, 145 nsec.

The cache is responsible for supplying data to the CPU upon request. CPU requests are made by passing the address of the word desired to the cache. This address is compared with the addresses currently stored in the cache and if the word is found, it is passed to the CPU. However, if not found, the word is loaded into the cache from main memory and passed to the CPU.

Data is transferred from cache to main memory and from main memory to cache in eight word blocks. Since the CPU is likely to request adjacent words of main memory, this method reduces the number of times the CPU cannot find words in the cache. Whether or not the cache has the word is transparent to the CPU, except in how long it takes to service the request. At any time there can be up to two requests pending to the cache, one for each ALU.

Main memory

The significant features of the HP 3000 Series 68 main memory are listed in Table D-3.

**Table D-3: HP 3000 Series 68
Main Memory Features**

- High-speed dynamic NMOS random access memory (64 K RAMS)
- Automatic fault detection and correction
- Memory sizes ranging from 2 to 8 megabytes
- 8 word block memory access
- Average memory access time (with cache): 145 nsec
- Write cycle time: 375 nsec minimum
- Read cycle time: 145 nsec (average with cache)
- 15 minute (minimum) rechargeable battery pack to maintain memory data during power failure. Total amount of backup time depends on memory size and battery condition (age and level of charge)

The HP 3000 Series 68 uses high speed random access memory (64 Kb NMOS RAM chips). The memory subsystem has the capability of detecting and logging single bit and double bit errors and correcting single bit errors.

The modular design of the HP 3000 Series 68 allows the system to have its main memory expanded from 2 megabytes to 8 megabytes in 1 megabyte increments.

At any time, the latest updated copy of a block of memory may reside in either main memory, the I/O buffer of an I/O Adapter, or in cache memory. To assure that the latest updated copy is accessed, cache memory and I/O buffers will supply data for any memory request if they have the latest updated copy.

Central System Bus

The Central System Bus (CSB) is the communication link between the CPU Module, Main Memory Module,

and Input/Output Adapter Modules. A 53 Mbyte/second overall bandwidth allows the CSB to support multiple IMBs. Over the CSB, no module has implied control. Modules operate independently of each other except when it is necessary to transfer data or send commands. When this is necessary, the initiating module asks for and receives control of the CSB. All transfers to and from memory are in 8 word blocks.

The Common Bus Interface (CBI) serves as the interface to the CSB. Each of the three modules; cache, I/O Adapter, and main memory; require a CBI to communicate across the CSB to another module.

I/O Adapter

Providing an interface between the Central System Bus and Intermodule Busses, Input/Output Adapters (IOAs) allow communication between the I/O system and the main memory and CPU. IOAs synchronize the relatively slow speed IMB with the CSB. The IOA controls direct memory access (DMA) between main memory, which is on the CSB, and I/O channels on the IMB. A 64 byte buffer cache memory included with each IOA buffers communication between the 16-bit IMB and the 32-bit CSB. To the devices on the IMB, the IOA appears as an effective memory by responding to IMB requests for read, write, and read/write. These requests are generated by the I/O controllers on behalf of the devices.

Series 68 system serviceability

The Series 68 is designed to be extremely reliable and easy to service. New hardware which enhances Series 68 serviceability (both hardware and software) includes the Diagnostic Control Unit (DCU) and the Power Distribution and Monitor (PDM) Unit. They help reduce system downtime and allow the Series 68 to have low monthly maintenance costs. Table D-4 lists the features provided by the DCU.

**Table D-4: HP 3000 Series 68
Diagnostic Control Unit Features**

- Friendly user interface—MPE-like commands
- Remote diagnostic capability
- Operator access and modification of CPU registers
- Control for loading and running system diagnostics
- Intelligence that controls system power through the Power Distribution and Monitor (PDM)
- Over temperature warning preceding temperature shutdown

The microprocessor controlled DCU can interrogate, modify, and diagnose all of the registers and most of the data paths in the CPU. It can also run diagnostic tests on the Main Memory, Cache Memory, I/O Adapter (IOA), and 80% of the General I/O Channel.

The DCU selftest is resident on the DCU in read only memory. Other microdiagnostics can be loaded into the WCS. Commands are MPE-like and error indications are given in simple English statements. The self-test takes approximately 30 seconds to execute and was designed to be simple enough to be initiated by the customer prior to requesting service from HP.

A unique logic design in the Series 68 allows the contents of any register in the CPU and many other system boards to be serially streamed into the DCU. This design requires no operational hardware except for the DCU. Each board has a single serial line running directly to the DCU. All of the registers and many other elements defining the machine state are connected in a serial shift string with this line. Under DCU control, the contents of any register can be examined or modified by appropriately shifting the serial shift string. Using this feature, DCU fault locating microdiagnostics can isolate most failures to a three board set and list that set on the system console. It can also be used by the system technician to examine register contents during system troubleshooting.

The DCU also protects main memory, user files, spool files and executing processes upon occurrences of a power failure. Similarly, when the DCU detects an over-temperature condition, it first generates a warning to the system console, and if the condition persists, will simulate a power failure condition shutting the system down and protecting circuits from being temperature stressed beyond normal operating specifications.

Most of the functions performed by the DCU are available remotely. By connecting a remote terminal to the DCU through a modem, a remote console may be operated in parallel with the system console. With this facility, an HP Customer Engineer can utilize DCU diagnostic features to identify and solve hardware problems while at a remote location.

The Series 68 Power Distribution and Monitor (PDM) provides an interface to the power system. The PDM monitors the power supply system to ensure the computer operates with valid power supplies. Also, it aids in the diagnosis and troubleshooting of power supply system faults and failures through status reporting to the DCU and to the System Status Display Panel (SSDP-B). Control of the PDM is accomplished through a control program running on the DCU.

Input/output

All access to input/output devices is by way of the device-independent MPE file system. All location of data, buffering, data transfers, and deblocking are handled automatically by MPE. All devices can be operated concurrently (within system bandwidth). Peripherals that fail are taken off-line from the operating system by operator command.

Peripheral I/O hardware

HP 3000 Series 68 peripheral I/O hardware consists of the General I/O Channels (GIC), DSN/Advanced Terminal Processors (ATP), DSN/Intelligent Network Processors (INP), and the peripheral units.

When an I/O request is issued, the device driver in the CPU performs different actions depending on whether or not a terminal is involved. With devices other than terminals, the CPU assembles a channel program, then issues a Start I/O Program (SIOP) instruction to the General I/O Channel over which the device communicates. For terminals, the host software driver receives calls from any of multiple sources for terminal actions and instructs the DSN/Advanced Terminal Processor to initiate the action. The DSN/Advanced Terminal Processor will interrupt the CPU when the action is completed.

The General I/O Channel is the hardware I/O channel which provides the electrical interface between the computer system via the IMB and peripheral devices connected to the computer system Hewlett-Packard Interface Bus (HP-IB). The HP-IB is HP's computer system implementation of the IEEE standard 488-1975 interface, used on the Series 68 to connect peripheral devices to the channel. The HP-IB consists of eight data lines and eight control lines. The ATP provides an intelligent interface between computer system via the IMB and terminals.

General I/O Channel

The General I/O Channel (GIC) is the primary channel for communication between the CPU and the I/O devices other than terminals. Each GIC communicates via the computer system Hewlett-Packard Interface Bus (HP-IB) and translates I/O commands from the CPU into the proper HP-IB protocol. Nearly all transactions with I/O devices are accomplished without software interrupts, since I/O is achieved with channel programs. Software is responsible for setting up a channel program, but the execution of this program is performed by the CPU's channel microcode. The CPU's channel microcode is devoted to I/O tasks and implements the necessary algorithms for decoding the channel instructions and effecting the required I/O operations. Once the channel program is running, device control and data flow are normally carried to completion with no software intervention and without altering the system environment.

Several devices may simultaneously need service, and the CPU must decide which one will receive attention. First, all channels are polled, and the highest priority channel with a device request pending is chosen. The CPU then obtains from that channel the number of the highest priority device needing service. Once the device number is determined, execution of the channel program will begin. The CPU fetches each channel instruction and breaks it down into several IMB commands addressed to the proper GIC. The GIC interprets these commands and directs them onto the HP-IB device.

The GIC contains Direct Memory Access (DMA) hardware which allows large records of data to be transferred at the maximum speed of the HP-IB (about 1 Mb/second). The channel microcode enables the device and then initiates the DMA hardware on the GIC. After initial addressing of a device to talk or listen, the CPU relinquishes control of the IMB and allows the GIC to perform its function through DMA operation. During this time the GIC becomes the master of the IMB and the IOA and controls traffic flow. On a read operation the DMA hardware will read the bytes, pack them into words and place them directly into memory, all without assistance from the CPU. The CPU is free to service other devices while DMA is in progress. Upon completion of a DMA transfer, the GIC returns to a slave condition and awaits the next operation.

Device controller

The device controller is the hardware linkage between a peripheral device and the computer system. Its primary function is to translate I/O commands from a General I/O Channel (GIC) to the unique signals required to control a particular device. When an I/O program is in execution, the device controller responds to and requests service from the GIC. The device controller also generates interrupts when required by some device condition or by channel command.

Device reference table (DRT)

Device controllers are identified by a logical device number which is used to access the device reference table (DRT). The DRT is known to both hardware and software, containing among other things, a pointer to the start of the SIO program for each device controller. Each device controller connects to a general I/O channel (GIC). Certain device controllers may control several logical devices. In such cases, each logical device attached to the controller is addressed separately using a unit number assigned when the device is installed.

Data service and interrupt priorities

In addition to a logical device number, there are two other characteristic numbers associated with each device controller. These are the data service priority and

interrupt priority. In the Series 68 both of these are determined by the logical device number in the DRT: the lower the number, the higher the priority.

DSN/Advanced Terminal Processor

An intelligent interface between terminals and the CPU is provided by the DSN/Advanced Terminal Processor (ATP). The significant features of the DSN/Advanced Terminal Processor are listed in Table D-5.

Table D-5: DSN/Advanced Terminal Processor Features
<ul style="list-style-type: none">■ Data transfer rates up to 9600 bits/second■ Handles character processing, eliminates CPU interrupts■ Expandable from 12 to 96 terminals■ RS-232-C and RS-422 support (local terminals up to 15m or 1200m, respectively from the CPU)■ Full-duplex asynchronous modem support■ Direct memory access of user data

An ATP is composed of one System Interface Board (SIB) and from one to eight Port Controllers. The SIB provides a hardware interface to the Intermodule Bus (IMB) and, under microprocessor control, performs byte packing and unpacking and controls direct memory access (DMA) of user data. Port Controllers provide the hardware interface for terminal/workstation devices to the Series 68. With a microprocessor dedicated to each terminal port, the Port Controller handles all handshaking between the system and the connected devices, character echoing, speed sensing, and input character buffering. The Series 68 may use multiple ATPs.

The ATP allows terminals to transmit and receive data on either a character-by-character basis or a block-at-a-time basis. For both types of operations, the ATP transfers data directly to and from memory. Because this eliminates the need for character processing by the CPU, the ATP significantly reduces CPU utilization.

A flexible set of physical attachment interfaces are available to allow asynchronous terminals to be attached directly, for local (intrafacility) use, or through full-duplex modems (Bell type 103, 202T, 212A, and CCITT V.24 type modems) for remote installations.

DSN/Intelligent Network Processor

The DSN/Intelligent Network Processor (INP) allows HP 3000 computers to be linked to other computers in a distributed data processing environment and supports multipoint terminals. The significant features of the DSN/Intelligent Network Processor are listed in Table D-6.

Table D-6 DSN/Intelligent Network Processor Features

-
- 16-bit SOS microprocessor
 - Data communications protocol handling
 - Character handling and 32 kbytes buffer storage
 - Modem and hardwired interfaces up to 56,000 bits/second
 - Full- and half-duplex asynchronous modem support
 - Bisync and HDLC/SDLC protocol compatible
 - RS-232-C, RS-422, CCITT V.24 and V.35 interfacing
 - Direct memory access for data
 - Auto call capability
-

The INP microprocessor performs all of the communication data link protocol management, including: serialization, protocol management, frame/block management, and data buffering. This reduces CPU utilization and frees it to perform other tasks. Throughput is increased by overlapping the transfer of data already received from the communication channel with the processing and buffering of new data coming from the communication channel. The protocol driver may be dynamically changed. This allows the INP to be easily reconfigured from one data link protocol to another and permits several subsystems to use a single INP. An auto call capability is included with the INP. It allows a remote connection to occur in a dial-up environment without the intervention of a human operator.

Interrupt system

The interrupt system provides for up to 105 external interrupt levels. When interrupts occur, the microprogrammed interrupt handler identifies each interrupt and grants control to the highest priority interrupt. Current operational status is saved in the microprogram, which then sets up the interrupt processing environment and transfers control to the interrupt routine.

Interrupt routines operate on a common stack (interrupt control stack) which is known to both hardware and software. This feature permits nesting of interrupt routines in the case of multiple interrupts, thus allowing higher priority devices to interrupt lower priority devices.

The interrupt system also provides for 20 internal interrupts (for user errors, system violations, hardware faults, and power fail/restart) plus fourteen traps for arithmetic errors and illegal use of instructions.

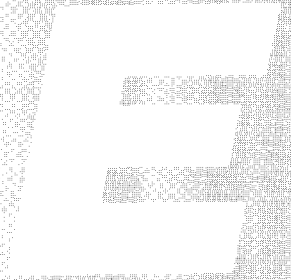
Peripherals

The peripheral devices used on the HP 3000 Series 68 are connected primarily to GICs, while the ATP is reserved solely for terminals. Peripherals attached to GICs through the HP-IB include disc drives, line printers, magnetic tape drives, card readers and DSN/Intelligent Network Processors. For a complete configuration of the supported peripherals on the HP 3000 Series 68, refer to the current HP 3000 Configuration Guide.

Automatic restart after power failure

An integral part of the HP 3000 Series 68 is a power fail/automatic restart capability. When the system high DC voltage level drops below 80% of rated voltage (300V DC nominal), the system initiates a powerfail warning (PFW). During PFW the system (hardware and MPE) writes all register contents to a reserved section of main memory, critical activities in the system are completed, and then the power down signal is generated to shut the system down. The battery back-up power supply refreshes main memory and ensures its validity for at least 15 minutes, depending on memory size and battery condition.

The system is automatically restarted when the high DC voltage level reaches 80% of rated voltage (300V DC nominal) and all register values are automatically restored and processing resumes.



Appendix E:
HP 3000 Series 68
Machine Instructions

Stack Op Instructions

ADAX	Add A to X	FIXT	Fix and truncate
ADBX	Add B to X	FLT	Float an integer
ADD	Add A to B	FMPY	Floating point multiply
ADXA	Add X to A	FNEG	Floating point negate
ADXB	Add X to B	FSUB	Floating point subtract D, C - B, A
AND	Logical AND of A and B	INCA	Increment A
BTST	Test byte on TOS and set CC	INCB	Increment B
CAB	Rotate A-B-C	INCX	Increment X
CMP	Integer compare B, A and set CC	LADD	Logical add A + B
DADD	Double integer add D, C, + B, A	LCMP	Logical compare B, A and set CC
DCMP	Double integer compare and set CC	LDIV	Logical divide C, B ÷ A
DDEL	Double delete TOS	LDXA	Load X into A
DDIV	Double integer divide	LDXB	Load X into B
DDUP	Double duplicate TOS	LMPY	Logical multiply B × A
DECA	Decrement A	LSUB	Logical subtract B - A
DECB	Decrement B	MPY	Multiply integers, integer product
DECX	Decrement X	MPYL	Multiply integers, long integer product
DEL	Delete TOS	NEG	Integer negate
DELB	Delete B	NOP	No operation
DFLT	Float a double integer	NOT	Logical complement TOS
DIV	Integer divide B by A	OR	Logical OR of A, B
DIVL	Divide long integer C, B ÷ A	STAX	Store A into X
DMUL	Double integer multiply	STBX	Store B into X
DNEG	Double integer negate	SUB	Integer subtract B - A
DSUB	Double integer subtract D, C - B, A	TEST	Test TOS and set CC
DTST	Test double word on TOS and set CC	XAX	Exchange A and X
DUP	Duplicate TOS	XBX	Exchange B and X
DXCH	Double exchange	XCH	Exchange A and B
DZRO	Push double zero onto stack	XOR	Logical exclusive OR of A, B
FADD	Floating point add, D, C + B, A	ZERO	Push integer zero onto stack
FCMP	Floating point compare and set CC	ZROB	Zero B
FDIV	Floating point divide D, C ÷ B, A	ZROX	Zero X
FIXR	Fix and round		

Shift Instructions

ASL	Arithmetic shift left	DLSR	Double logical shift right
ASR	Arithmetic shift right	LSL	Logical shift left
CSL	Circular shift left	LSR	Logical shift right
CSR	Circular shift right	QASL	Quadruple arithmetic shift left
DASL	Double arithmetic shift left	QASR	Quadruple arithmetic shift right
DASR	Double arithmetic shift right	TASL	Triple arithmetic shift left
DCSL	Double circular shift left	TASR	Triple arithmetic shift right
DCSR	Double circular shift right	TNSL	Triple normalizing shift left
DLSL	Double logical shift left		

Legend

TOS	Top of stack	A	Top of stack	D	Location below C
CC	Condition Code	B	Location below A	DB	Data Base
X	Index Register	C	Location below B	DL	Data Limit

Program Control and Special Instructions

DISP	Dispatch	PCN	Push CPU code (% 4)
EXIT	Exit from procedure	PSDB	Pseudo interrupt disable
HALT	Halt	PSEB	Pseudo interrupt enable
IXIT	Interrupt exit	RSW	Push cold load chan/dev
LLBL	Load label	SCAL	Subroutine call
LLSH	Linked list search	SXIT	Exit from subroutine
PAUS	Pause, interruptable	XEQ	Execute stack word
PCAL	Procedure call		

Machine and I/O Instructions

DUMP	Programmatic dump	SCLR	Set system clock limit
FLSH	Flush CPU and I/O Cache	SED	Set enable/disable external interrupts
HIOP	Halt I/O program	SINC	Set system clock interrupt
INIT	Initialize I/O channel	SIOP	Set I/O channel program
MCMD	Message command	SMSK	Set device mask
MLOG	CPU log	STRT	Programmatic warm start
RCCR	Read system clock	TOFF	Hardware timer off
RDCU	DCU log	TON	Hardware timer on
RIOA	Read I/O Adapter	WIOA	Write I/O Adapter
RMSK	Read device mask		

Loop Control Instructions

MTBA	Modify variable, test against limit, branch	TBA	Test variable against limit, branch
MTBX	Modify X, test against limit, branch	TBX	Test X against limit, branch

Memory Address Instructions

ADDM	Add memory to TOS	LDX	Load X
CMPM	Compare TOS with memory	LOAD	Load word onto stack
DECM	Decrement memory	LRA	Load relative address onto stack
INCM	Increment memory	MPYM	Multiply TOS by memory
LDB	Load byte onto stack	STB	Store byte on TOS into memory
LDD	Load double word onto stack	STD	Store double on TOS into memory
LDPN	Load double from program, negative	STOR	Store TOS into memory
LDPP	Load double from program, positive	SUBM	Subtract memory from TOS

Extended Instruction Set

Extended-Precision Floating Point		Decimal Arithmetic	
EADD	ADD	ADDD	Decimal add
ECMP	Compare	CMPD	Decimal compare
EDIV	Divide	CVAD	ASCII to decimal conversion
EMPY	Multiply	CVBD	Binary to decimal conversion
ENEG	Negate	CVDA	Decimal to ASCII conversion
ESUB	Subtract	CVDB	Decimal to binary conversion
		DMPY	Double logical multiply
		MPYD	Decimal multiply
		NSLD	Decimal normalizing left shift
		SLD	Decimal left shift
		SRD	Decimal right shift
		SUBD	Decimal subtract

Field and Bit Instructions

DPF	Deposit field, A bits to B	TCBC	Test and complement bit, set CC
EXF	Extract specified field, right-justify	TRBC	Test and reset bit, set CC
SCAN	Scan bits	TSBC	Test and set bit, set CC
TBC	Test specified bit and set CC		

Branch Instructions

BCC	Branch on specified CC	BRO	Branch on TOS odd (bit 15 = 1)
BCY	Branch on carry	CPRB	Compare range and branch
BNCY	Branch on no carry	DABZ	Decrement A, branch if zero
BNOV	Branch on no overflow	DXBZ	Decrement X, branch if zero
BOV	Branch on overflow	IABZ	Increment A, branch if zero
BR	Branch unconditionally	IXBZ	Increment X, branch if zero
BRE	Branch on TOS even (bit 15 = 0)		

Move Instructions

CMPB	Compare bytes in two memory blocks	MVB	Move bytes in memory, addresses + / -
MABS	Move using absolute addresses	MVBL	Move words from DB + to DL + area
MDS	Move using data segments	MVBW	Move bytes while of specified type
MFDS	Move from data segment	MVLB	Move words from DL + to DB + area
MOVE	Move words in memory, addresses + / -	SCU	Scan bytes until test or terminal byte
MTDS	Move to data segment	SCW	Scan bytes while equal to test byte

Privileged Memory Reference Instructions

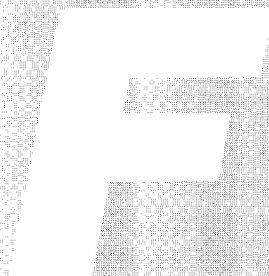
LDEA	Load double word from extended address	PSTA	Privileged store into absolute address
LSEA	Load single word from extended address	SDEA	Store double word into extended address
LST	Load from system table	SSEA	Store single word into extended address
PLDA	Privileged load from absolute address	SST	Store into system table

Immediate Instructions

ADDI	Add immediate to integer in A	LDXI	Load X immediate
ADXI	Add immediate to X	LDXN	Load X negative immediate
ANDI	Logical AND immediate with A	MPYI	Multiply immediate with A
CMPI	Compare A with immediate, set CC	ORI	Logical OR immediate with A
CMPN	Compare A with negative immediate	SBXI	Subtract immediate from X
DIVI	Divide immediate into A	SUBI	Subtract immediate from A
LDI	Load immediate to TOS	XORI	Logical exclusive OR immediate
LDNI	Load negative immediate to TOS		

Register Control Instructions

ADDS	Add operand to stack pointer	SETR	Set specified registers from stack
PSHR	Push specified registers onto stack	SUBS	Subtract operand from stack pointer
RCLK	Read clock	XCHD	Exchange DB and TOS
SCLK	Store clock		



Appendix F:
HP 3000 Series 48
Hardware Features

This appendix provides a summary of hardware features of the HP 3000 Series 48 computer system. For more details on the HP 3000 architecture, refer to Chapter 1—System Introduction and Chapter 3—System Architecture.

The HP 3000 Series 48 design is based on a modular concept which allows independent elements to be interconnected through a central system bus structure. These elements consist of a central processing unit which controls memory via a controller, General I/O Channels, DSN/Advanced Terminal Processor, DSN/Asynchronous Data Communications Controllers, and the bus system which allows communication between the I/O devices. Also, the system includes a system console and a Control and Maintenance Processor (CMP). Peripheral devices are connected to the system through the General I/O Channels. Interactive terminals are attached to the system through either the DSN/Advanced Terminal Processor, DSN/Asynchronous Data Communications Con-

trollers, or DSN/Intelligent Network Processors running the DSN/Multipoint Terminal Software. Data communication links are established via DSN/Intelligent Network Processors.

The CPU is centered around a Hewlett-Packard designed microcoded processor using Schottky TTL technology. Implementation using this technology provides high speed execution of instructions while allowing flexibility in the machine instruction set. The Series 48 also employs high speed, semiconductor, random access memory modules which use automatic fault detection and correction.

The hardware design of the HP 3000 Series 48 provides the ability to expand the system as the users needs grow and applications change. The system software accommodates these changing needs by allowing additional hardware modules and peripheral devices to be easily configured on the system.

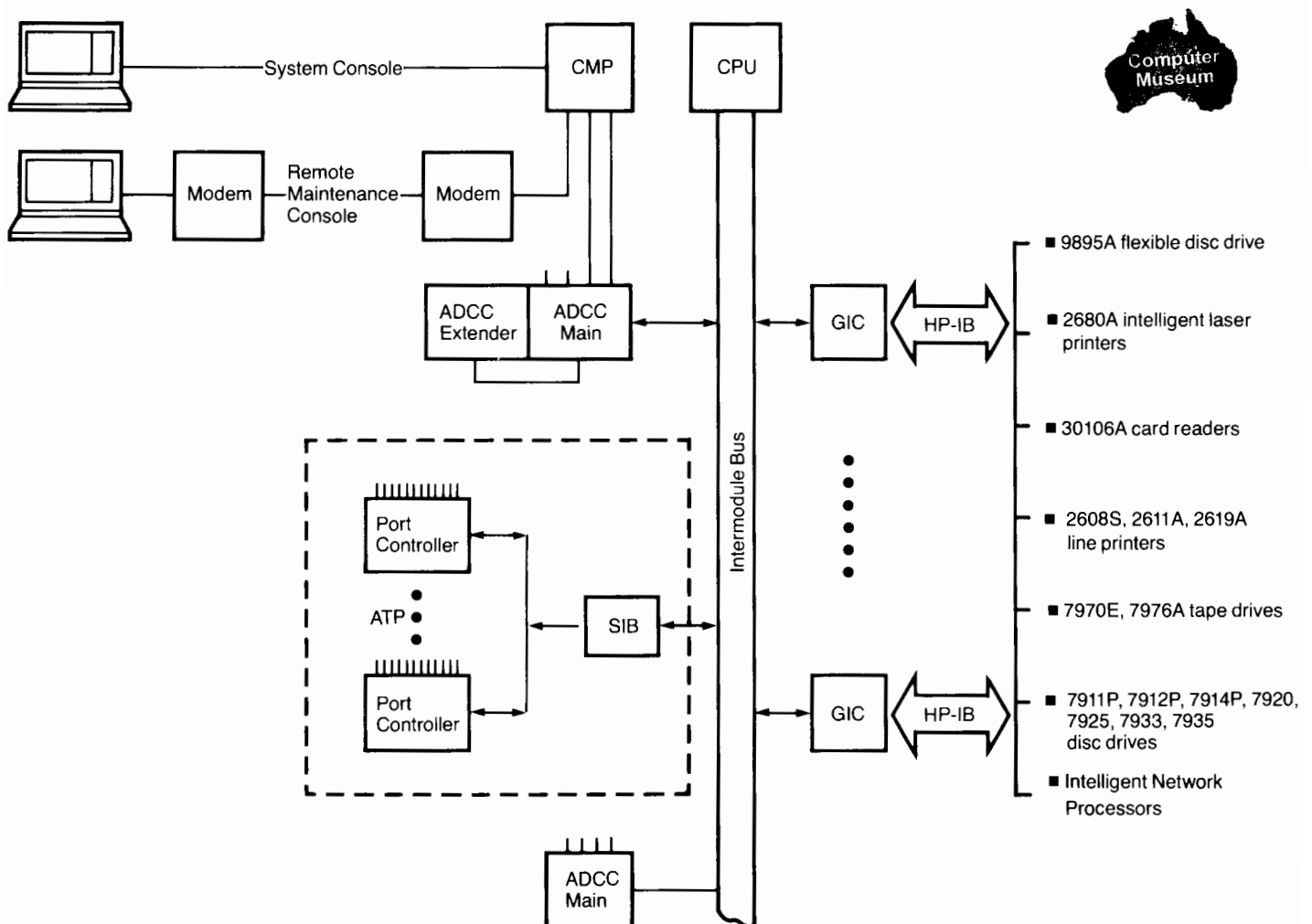


Figure F-1. HP 3000 Series 48 Hardware Organization

The HP 3000 Series 48 instruction set is presented in Appendix H. Instructions are 16 or 32 bits in length except stack operations, which are 8 bit instructions. These include a variety of memory reference, branch, arithmetic and data manipulation instructions that operate on integer, real, logical, packed decimal, character and string data. Floating point arithmetic can be performed in single precision (32 bits) or double precision (64 bits), integer arithmetic in 16-bit and 32-bit lengths, and packed decimal instructions extended to 28 digits in precision. In addition, there are a number of instructions designed to aid in creating the multi-programming environment of the system. These include procedure call and exit instructions and others which implement various operating system functions previously done in software.

Firmware storage and control consists of microcode stored in read-only memory (ROM), and associated logic control. Microcode routines control the operation of the instruction decoder and the hardware processor in order to create the HP operating environment. The microinstruction cycle time is 105 nanoseconds.

The hardware processor consists of an arithmetic-logic unit, shifting network, 72 specific purpose registers—18 of which are accessible to user programs—and related data manipulating and testing logic. Since the HP 3000 architecture (see chapter 3) is structured on code segments and data segments, most of the CPU registers are used for defining the segment limits and operating elements within the segments. As shown in Figure F-2, three of the CPU registers point to locations in a code segment defined as the current code segment. Five of the registers point to locations in a data segment defined as the current data segment. Table F-2 lists all 72 registers and their associated functions.

Central Processing Unit (CPU)

The significant features of the HP 3000 Series 48 central processing unit (CPU) are listed in Table F-1.

The CPU converts an instruction in the current instruction register (CIR) into a starting address for the microcode contained in the read only memory (ROM), and determines various initial conditions required for executing the instruction. As the current instruction is being executed, the next instruction is fetched and placed in the next instruction register (NIR). Upon completion of the current instruction, the contents of NIR are loaded into CIR and the cycle is repeated. This "pipelining" of the current instruction execution with the next instruction-fetch improves throughput by overlapping operations.

Table F-1. HP 3000 Series 48
CPU Features

Architecture

- Hardware-implemented stack
- Separation of code and data
- Non-modifiable re-entrant code
- Variable-length code segmentation
- Virtual memory for code
- Dynamically relocatable programs

Implementation

- Microprogrammed Schottky TTL CPU
- 105 nanosecond microinstruction cycle time
- Automatic restart after power failure
- Intermodule bus
- Overlapping CPU and I/O operations

Instructions

- 214 powerful instructions
- Instructions are 8, 16, and 32 bits in length
- 16 and 32 bit integer arithmetic
- 32 and 64 bit floating point arithmetic
- 28 digit packed decimal arithmetic
- Special instructions that optimize the efficiency of the operating system

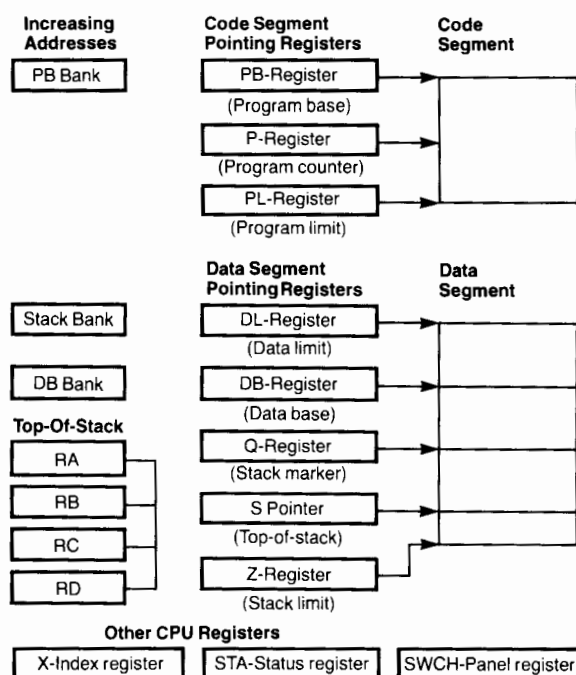


Figure F-2. HP 3000 CPU REGISTERS

The four top of stack registers are of special interest. In order to improve execution speed, up to four elements from the top of a data stack may be contained in these registers. This allows many functions to be treated as register-to-register operations rather than the slower speed memory-to-register or register-to-memory type operations. These registers are manipulated by the CPU, and their use is fully transparent.

**Table F-2 Series
48 Hardware Registers**

Registers accessible to user programmers

Register	Function
PB	Code Segment Pointers
P	
PL	
PBNK	
DBNK	Stack Pointers
SBNK	
DL	
DB	
Q	
SM	
Z	
RA	Top of Stack Registers
RB	
RC	
RD	
X	Index Register
STA	Status Register
SWCH	Front Panel Switch Register

Registers dedicated for system use

CIR	Current Instruction Register
NIR	Next Instruction Register
R4-R13	Scratch Pad, Flag, and Interrupt Registers
R15-R39	
SIR	
SP0	
CTR	
ABNK	
BBNK	
F1-F4	
NF1-NF4	
OPND	Memory Address and Data Registers
UBUS	
CSAR	Firmware Address Register

Main memory

The significant features of the HP 3000 Series 48 main memory are listed in Table F-3.

**Table F-3: HP 3000 Series 48
Main Memory Features**

- High-speed dynamic NMOS random access memory
- Automatic fault detection and correction
- Memory sizes ranging from 1 megabyte to 4 megabytes
- Write cycle time: 530 nsec minimum
- Read access time: 300 nsec
- Read cycle time: 430 nsec
- 15 minute (minimum) rechargeable battery pack to maintain memory data during power failure. Total amount of backup time depends on memory size and battery condition (age and level of charge)

The HP 3000 Series 48 uses high speed random access memory. The memory subsystem has the capability of detecting single bit and double bit errors and correcting single bit errors.

The modular design of the HP 3000 Series 48 allows the system to have its main memory expanded from 1 megabyte to 4 megabytes.

The word length transmitted over the intermodule bus is 16 bits. In the memory modules, the word length is expanded to 39 bits; 32 bits of data and 7 bits of automatic fault detection and correction.

Series 48 system serviceability

The Series 48 is designed to be extremely reliable and easy to service. Hardware which enhances Series 48 serviceability (both hardware and software) is the Control and Maintenance Processor (CMP). The microprocessor based CMP provides diagnostic and control functions for identifying hardware problems. It helps reduce system downtime and allows the Series 48 to have low monthly maintenance costs. Table F-4 lists the features provided by the CMP.

**Table F-4. HP 3000 Series 48
Control and Maintenance Processor**

-
- Friendly user interface—MPE-like commands
 - System Console Flexibility—any 262x, 264x, 2382A, or 2635B terminal may be used as a system console
 - System selftest
 - Remote diagnostic capability
 - Over temperature shutdown
-

The diagnostics are resident on the CMP in read only memory. Commands are MPE-like and error indications are given in simple English statements. The self-test takes approximately 30 seconds to execute and was designed to be simple enough to be initiated by the customer prior to requesting service from HP. The CMP also protects main memory, user files, spool files and executing processes upon occurrences of a power failure.

All the functions performed by the CMP are available remotely. By connecting a remote terminal to the CMP through a modem, a remote console may be operated in parallel with the system console. With this facility, an HP Customer Engineer can utilize CMP diagnostic features to identify and solve hardware problems while at a remote location.

I/O System

Communication between the CPU, memory and I/O modules is carried over the Intermodule Bus (IMB). Because the CPU generates greater than 90 percent of the bus activity, it is given continuous access to the bus and relinquishes control to the I/O channels only on request.

The IMB has separate address and data paths, each with handshake controls that operate in a master/slave mode to transfer data between modules. The CPU talks to memory and to the I/O system and always functions as a master. The I/O channels function as masters to memory but become slaves when talking with the CPU. To access memory, the I/O channels must request the bus through a priority structure. Any channel request will cause the CPU to relinquish control of the IMB so that the request can be serviced.

Input/output

All access to input/output devices is by way of the device-independent MPE file system. All location of data, buffering, data transfers, and deblocking are handled automatically by MPE. All devices can be operated concurrently (within system bandwidth). Peripherals that fail are taken off-line from the operating system by operator command.

Peripheral I/O hardware

HP 3000 Series 48 peripheral I/O hardware consists of the General I/O Channels (GIC), DSN/Advanced Terminal Processors, DSN/Asynchronous Data Communication Controllers, DSN/Intelligent Network Processors, and the peripheral units. DSN/Advanced Terminal Processors and DSN/Asynchronous Data Communications Controllers interface log-on and data-entry terminals.

When an I/O request is issued, the device driver in the CPU performs different actions depending on whether or not a terminal controlled by an DSN/Advanced Terminal Processor is involved. With devices other than terminals controlled by an ATP, the CPU assembles a channel program, then issues a Start I/O Program (SIOP) instruction to the General I/O Channel or DSN/Asynchronous Data Communications Controller over which the device communicates. For terminals controlled by an ATP, the host software driver receives calls from any of multiple sources for terminal actions and instructs the DSN/Advanced Terminal Processor to initiate the action. The DSN/Advanced Terminal Processor will interrupt the CPU when the action is completed.

The General I/O Channel is the hardware I/O channel which provides the electrical interface between the computer system via the IMB and peripheral devices connected to the computer system Hewlett-Packard Interface Bus (HP-IB). The HP-IB is HP's computer system implementation of the IEEE standard 488-1975 interface, used on the Series 48 to connect peripheral devices to the channel. The HP-IB consists of eight data lines and eight control lines. The ATP provides an intelligent interface between computer system and terminals.

General I/O Channel

The General I/O Channel (GIC) is the primary channel for communication between the CPU and the I/O devices other than terminals. Each GIC controls a computer system Hewlett-Packard Interface Bus (HP-IB) and translates I/O commands from the CPU into the proper HP-IB protocol. Nearly all transactions with I/O devices are accomplished without software interrupts, since I/O is achieved with channel programs. Software is responsible for setting up a channel program, but the execution of this program is performed by the CPU's channel microcode. The CPU's channel microcode is devoted to I/O tasks and implements the necessary algorithms for decoding the channel instructions and effecting the required I/O operations. Once the channel program is running, device control and data flow are normally carried to completion with no software intervention and without altering the system environment.

Several devices may simultaneously need service, and the CPU must decide which one will receive attention. First, all channels are polled, and the highest priority channel with a device request pending is chosen. The CPU then obtains from that channel the number of the highest priority device needing service. Once the device number is determined, execution of the channel program will begin. The CPU fetches each channel instruction and breaks it down into several IMB commands addressed to the proper GIC. The GIC interprets these commands and directs them onto the HP-IB device.

The GIC contains Direct Memory Access (DMA) hardware which allows large records of data to be transferred at the maximum speed of the HP-IB (about 1 Mb/second). The channel microcode enables the device and then initiates the DMA hardware on the GIC. After initial addressing of a device to talk or listen, the CPU relinquishes control of the IMB and allows the GIC to perform its function through DMA operation. During this time the GIC becomes the master of the bus and memory and controls traffic flow. On a read operation the DMA hardware will read the bytes, pack them into words and place them directly into memory, all without assistance from the CPU. The CPU is free to service other devices while DMA is in progress. Upon completion of a DMA transfer, the GIC returns to a slave condition and awaits the next operation.

Device controller

The device controller is the hardware linkage between a peripheral device and the computer system. Its primary function is to translate I/O commands from a general I/O channel (GIC) to the unique signals required to control a particular device. When an I/O program is in execution, the device controller responds to and requests service from the GIC. The device controller also generates interrupts when required by some device condition or by channel command.

Device reference table (DRT)

Device controllers are identified by a logical device number which is used to access the device reference table (DRT). The DRT is known to both hardware and software, containing among other things, a pointer to the start of the SIO program for each device controller. Each device controller connects to a General I/O Channel (GIC). Certain device controllers may control several logical devices. In such cases, each logical device attached to the controller is addressed separately using a unit number assigned when the device is installed.

Data service and interrupt priorities

In addition to a logical device number, there are two other characteristic numbers associated with each device controller. These are the data service priority and interrupt priority. In the Series 44 both of these are determined by the logical device number in the DRT: the lower the number, the higher the priority.

DSN/Advanced Terminal Processor

An intelligent interface between terminals and the CPU is provided by the DSN/Advanced Terminal Processor (ATP). The significant features of the DSN/Advanced Terminal Processor are listed in Table F-5.

**Table F-5 DSN/Advanced
Terminal Processor Features**

-
- Data transfer rates up to 9,600 bits/second
 - Handles character processing, eliminating CPU interrupts
 - Expandable from 12 to 60 terminals
 - RS-232-C and RS-422 support (local terminals up to 15m or 1220m, respectively from the CPU).
 - Direct memory access of user data
-

An ATP is composed of one System Interface Board (SIB) and from one to five Port Controllers. The SIB provides a hardware interface to the Intermodule Bus (IMB) and, under microprocessor control, performs byte packing and unpacking and controls direct memory access (DMA) of user data. Port Controllers provide the hardware interface for terminal/workstation devices to the Series 44. With a microprocessor dedicated to each terminal port, the Port Controller handles all handshaking between the system and the connected devices, character echoing, speed sensing, and input character buffering.

The ATP allows terminals to transmit and receive data on either a character-by-character basis or a block-at-a-time basis. For both types of operations, the ATP transfers data directly to and from memory. Because this eliminates the need for character processing by the CPU, the ATP significantly reduces CPU utilization.

A flexible set of physical attachment interfaces are available to allow asynchronous terminals to be attached directly, for local (intrafacility) use.

DSN/Asynchronous Data Communication Controller

The DSN/Asynchronous Data Communications Controller (ADCC), the second channel type used in the system, provides a way to connect modem connect terminals and an alternative way to connect direct connect terminals. The Series 48 requires at least one ADCC for the CMP.

This channel performs for terminals essentially the same functions as the GIC but not in the same manner. Data is transferred from memory to the ADCC in parallel form, then converted to a serial bit stream for transmission over the RS-232-C lines to the device. Information being read from a device is in serial form and is converted to eight-bit bytes for transfer to memory.

Two types of ADCC boards may be used, the Main ADCC and the Extender ADCC. Each board contains four ports for connections to devices through RS-232-C data communication lines. The Main ADCC supports full duplex (Bell type 103, 212, and 202T modems in the U.S.) only. The Main ADCC is used when four or fewer devices are connected to a channel. The Extender ADCC extends the device capability of the channel to eight. Most of the control circuitry is on the Main ADCC. For this reason, the Main ADCC is required for the Extender ADCC to function.

When more than eight devices are to be attached to an ADCC channel, additional Main ADCCs are required, since each ADCC can accommodate only one Extender.

Unlike the GIC, the ADCC does not have a DMA facility, and therefore, cannot be the master of an IMB or of memory. Also, terminals on the channel do not respond to a parallel poll. As a result, the ADCC is always a slave and must be directly controlled by the CPU through the use of channel programs. Circuitry on the ADCC decodes the address information relating to channels and devices, and selects the correct device for operation.

The ports on the ADCC (Main and Extender) may be either hardwired to devices or to full-duplex modems.

DSN/Intelligent Network Processor

The DSN/Intelligent Network Processor (INP) allows HP 3000 computers to be linked to other computers in a distributed data processing environment and supports multipoint terminals. The significant features of the DSN/Intelligent Network Processor are listed in Table F-6.

Table F-6 DSN/Intelligent Network Processor Features

-
- 16-bit SOS microprocessor
 - Data communications protocol handling
 - Character handling and 32 kbytes buffer storage
 - Modem and hardwired interfaces up to 56,000 bits/second
 - Full- and half-duplex asynchronous modem support
 - Bisync and HDLC/SDLC protocol compatible
 - RS-232-C, RS-422, CCITT V.24 and V.35 interfacing
 - Direct memory access for data
 - Auto call capability
-

The INP microprocessor performs all of the communication data link protocol management, including: serialization, protocol management, frame/block management, and data buffering. This reduces CPU utilization and frees it to perform other tasks. Throughput is increased by overlapping the transfer of data already received from the communication channel with the processing and buffering of new data coming from the communication channel. The protocol driver may be dynamically changed. This allows the INP to be easily reconfigured from one data link protocol to another and permits several subsystems to use a single INP. An auto call capability is included with the INP. It allows a remote connection to occur in a dial-up environment without the intervention of a human operator.

Interrupt system

The interrupt system provides for up to 105 external interrupt levels. When interrupts occur, the microprogrammed interrupt handler identifies each interrupt and grants control to the highest priority interrupt. Current operational status is saved in the microprogram, which then sets up the interrupt processing environment and transfers control to the interrupt routine.

Interrupt routines operate on a common stack (interrupt control stack) which is known to both hardware and software. This feature permits nesting of interrupt routines in the case of multiple interrupts, thus allowing higher priority devices to interrupt lower priority devices.

The interrupt system also provides for 20 internal interrupts (for user errors, system violations, hardware faults, and power fail/restart) plus fourteen traps for arithmetic errors and illegal use of instructions.

Peripherals

The peripheral devices used on the HP 3000 Series 48 are connected primarily to GICs, while the ATP and ADCC are reserved solely for terminals and remote printers. Peripherals attached to GICs through the HP-IB include disc drives, line printers, and magnetic tape drives. For a complete configuration of the supported peripherals on the HP 3000 Series 48, refer to the current HP 3000 Configuration Guide.

Automatic restart after power failure

An integral part of the HP 3000 Series 48 is a power fail/automatic restart capability. When the system AC line voltage falls below 90% of rated voltage, the system initiates a powerfail warning (PFW). During PFW the system (hardware and MPE) writes all register contents to a reserved section of main memory, activities in the system are successfully completed, and then the power down signal is generated and the system is shut down. The battery back-up power supply refreshes main memory and ensures its validity for up to 15 minutes, depending on memory size and battery condition.

The system is automatically restarted when power supply voltages reach 90% of their rated values and all register values are automatically restored and processing resumes.



Appendix G:
HP 3000 Series 39 and 42
Hardware Features

This appendix provides a summary of hardware features of the HP 3000 Series 39 and 42 computer systems. For more details on the HP 3000 architecture, refer to Chapter 1—System Introduction and Chapter 3—System Architecture.

The HP 3000 Series 39 and 42 design is based on a modular concept which allows independent elements to be interconnected through a central system bus structure. These elements consist of a central processing unit which controls memory via a memory controller, General I/O Channels, DSN/Asynchronous Data Communication Controllers and the bus system which allows communication between the I/O devices. Also, the system includes a system console and a Control and Maintenance Processor (CMP). Peripheral devices are connected to the system through the General I/O Channels. Interactive terminals are attached to the system through the DSN/Asynchronous Data Communications Controllers or DSN/Intelligent Network Processors running the

DSN/Multipoint Terminal Software. Data communication links are established via DSN/Intelligent Network Processors.

The CPU is centered around a Hewlett-Packard designed microcoded processor using Schottky TTL technology. Implementation using this technology provides high speed execution of instructions while allowing flexibility in the machine instruction set. The Series 39 and 42 also employs high speed, semiconductor, random access memory modules which use automatic fault detection and correction.

The hardware design of the HP 3000 Series 39 and 42 provide the ability to expand the system as the user's needs grow and applications change. The system software accommodates these changing needs by allowing additional hardware modules and peripheral devices to be easily configured on the system.

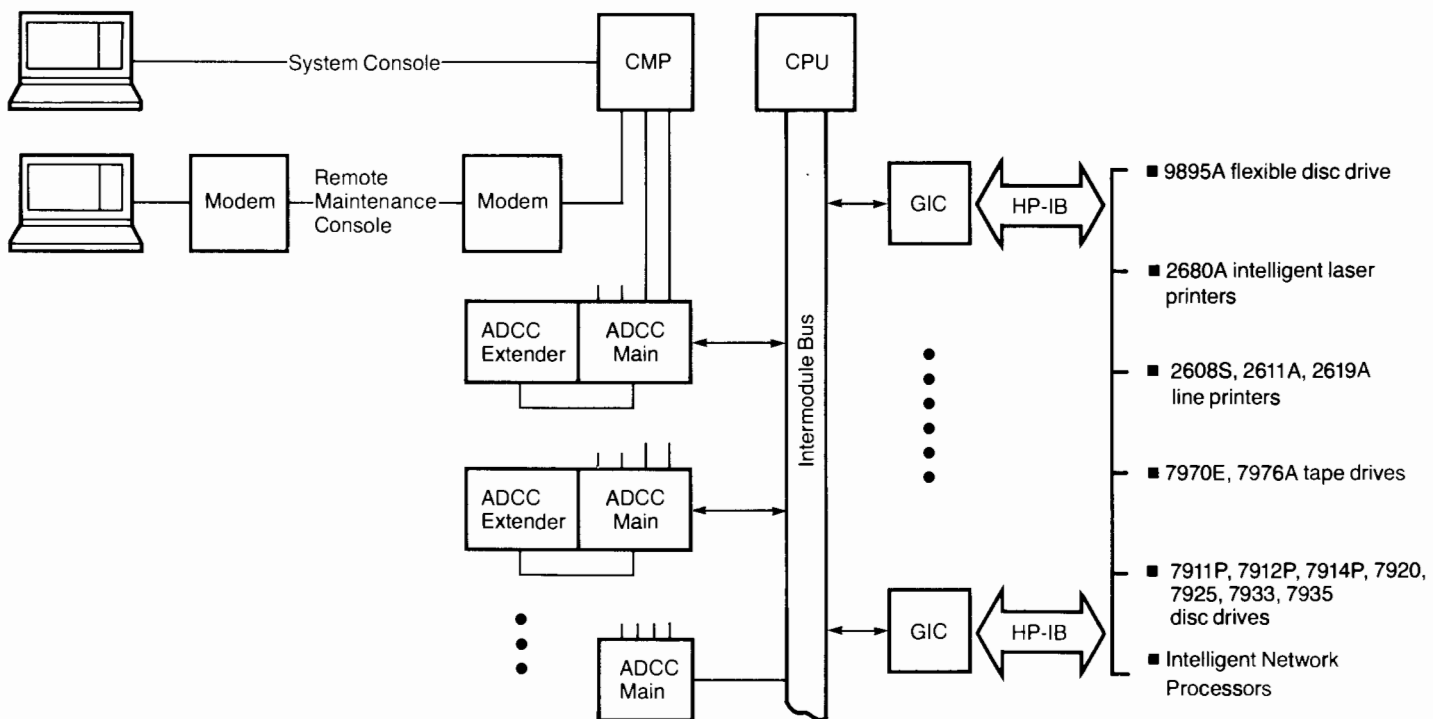


Figure G-1 HP 3000 Series 39 and 42 Hardware Organization

Central Processing Unit (CPU)

The significant features of the HP 3000 Series 39 and 42 central processing unit (CPU) are listed in Table G-1.

Table G-1 HP 3000 Series 39 and 42 CPU Features

Architecture	
■	Hardware-implemented stack
■	Separation of code and data
■	Non-modifiable re-entrant code
■	Variable-length code segmentation
■	Virtual memory for code
■	Dynamically relocatable programs
Implementation	
■	Microprogrammed Schottky TTL CPU
■	105 nanosecond microinstruction cycle time
■	Automatic restart after power failure
■	Intermodule bus
■	Overlapping CPU and I/O operations
Instructions	
■	214 powerful instructions
■	Instructions are 8, 16 and 32 bits in length
■	16 and 32 bit integer arithmetic
■	32 and 64 bit floating point arithmetic
■	28 digit packed decimal arithmetic
■	Special instructions that optimize the efficiency of the operating system.

The CPU converts an instruction in the current instruction register (CIR) into a starting address for the microcode contained in the read-only memory (ROM), and determines various initial conditions required for executing the instruction. As the current instruction is being executed, the next instruction is fetched and placed in the next instruction register (NIR). Upon completion of the current instruction, the contents of NIR are loaded into CIR and the cycle is repeated. This "pipelining" of the current instruction execution with the next instruction-fetch improves throughput by overlapping operations.

The HP 3000 Series 39 and 42 instruction set is presented in Appendix H. Instructions are 16 or 32 bits in length except stack operations, which are 8-bit instructions. These include a variety of memory reference, branch, arithmetic and data manipulation instructions that operate on integer, real, logical, packed decimal, character and string data. Floating point arithmetic can be performed in single precision (32 bits) or double pre-

cision (64 bits), integer arithmetic in 16-bit and 32-bit lengths, and packed decimal instructions extended to 28 digits in precision. In addition, there are a number of instructions designed to aid in creating the multi-programming environment of the system. These include procedure call and exit instructions and others which implement various operating system functions previously done in software.

Firmware storage and control consists of microcode stored in read-only memory (ROM), and associated logic control. Microcode routines control the operation of the instruction decoder and the hardware processor in order to create the HP operating environment. The microinstruction cycle time is 105 nanoseconds.

The hardware processor consists of an arithmetic-logic unit, shifting network, 72 specific purpose registers—18 of which are accessible to user programs—and related data manipulating and testing logic. Since the HP 3000 architecture (see Chapter 3) is structured on code segments and data segments, most of the CPU registers are used for defining the segment limits and operating elements within the segments. As shown in Figure G-3, three of the CPU registers point to locations in a code segment defined as the current code segment. Five of the registers point to locations in a data segment defined as the current data segment. Table G-2 lists all 72 registers and their associated functions.

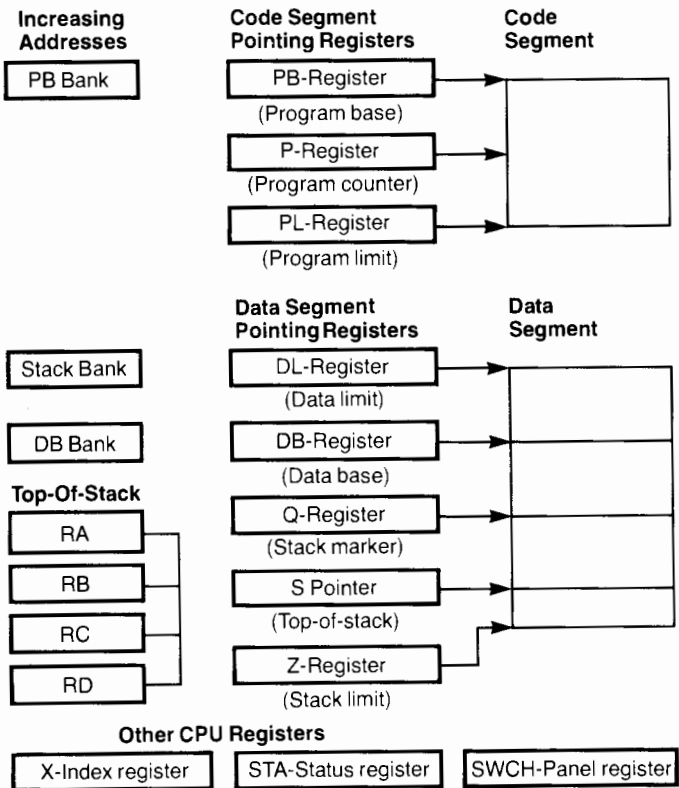


Figure G-2 HP 3000 CPU Registers

The four top-of-stack registers are of special interest. In order to improve execution speed, up to four elements from the top of a data stack may be contained in these registers. This allows many functions to be treated as register-to-register operations rather than the slower speed memory-to-register or register-to-memory type operations. These registers are manipulated by the CPU, and their use is fully transparent.

**Table G-2 Series
39 and 42 Hardware Registers**

Registers accessible to user programmers	
Register	Function
PB	Code Segment Pointers
P	
PL	
PBNK	
DBNK	Stack Pointers
SBNK	
DL	
DB	
Q	
SM	
Z	
RA	Top-of-Stack Registers
RB	
RC	
RD	
X	Index Register
STA	Status Register
SWCH	Front Panel Switch Register
Registers dedicated for system use	
CIR	Current Instruction Register
NIR	Next Instruction Register
R4-R13	Scratch Pad, Flag, and Interrupt Registers
R15-R39	
SIR	
SP0	
CTR	
ABNK	
BBNK	
F1-F4	
NF1-NF4	
OPND	Memory Address and Data Registers
UBUS	
CSAR	Firmware Address Register

Main memory

The significant features of the HP 3000 Series 39 and 42 main memory are listed in Table G-3.

**Table G-3 HP 3000 Series 39 and 42
Main Memory Features**

- High-speed dynamic NMOS random access memory
- Automatic fault detection and correction
- Memory sizes ranging from 512K bytes to 3 megabytes
- Write cycle time: 530 nsec minimum
- Read access time: 300 nsec
- Read cycle time: 430 nsec
- 15 minute (minimum) rechargeable battery pack to maintain memory data during power failure. Total amount of backup time depends on memory size and battery condition (age and level of charge)

The HP 3000 Series 39 and 42 uses high speed random access memory. The memory subsystem has the capability of detecting and logging single bit and double bit errors and correcting single bit errors.

The word length transmitted over the intermediate bus is 16 bits. In the memory modules, the word length is expanded to 39 bits; 32 bits of data and 7 bits of automatic fault detection and correction.

Series 39 and 42 system serviceability

The Series 39 and 42 are designed to be extremely reliable and easy to service. Hardware which enhances Series 39 and 42 serviceability (both hardware and software) is the Control and Maintenance Processor (CMP). The microprocessor based CMP provides diagnostic and control functions for identifying hardware problems. It helps reduce system downtime and allows the Series 39 and 42 to have low monthly maintenance costs. Table G-4 lists the features provided by the CMP.

**Table G-4 HP 3000 Series 39 and 42
Control and Maintenance Processor**

- Friendly user interface—MPE-like commands
- System Console Flexibility—any 262x, 264x, 2382A, or 2635B terminal may be used as a system console
- System selftest
- Remote diagnostic capability

All CMP programs are resident in read only memory. Commands are MPE-like and error indications are given in simple English statements. The selftest takes approximately 30 seconds to execute and was designed to be simple enough to be initiated by the customer prior to requesting service from HP.

All the functions performed by the CMP are available remotely. By connecting a remote terminal to the CMP through a modem, a remote console may be operated in parallel with the system console. With this facility, an HP Customer Engineer can utilize CMP diagnostic features to identify and solve hardware problems while at a remote location.

I/O System

Communication between the CPU, memory and I/O modules is carried over the Intermodule Bus (IMB). Because the CPU generates greater than 90 percent of the bus activity, it is given continuous access to the bus and relinquishes control to the I/O channels only on request.

The IMB has separate address and data paths, each with handshake controls that operate in a master/slave mode to transfer data between modules. The CPU talks to memory and to the I/O system and always functions as a master. The I/O channels function as masters to memory but become slaves when talking with the CPU. To access memory, the I/O channels must request the bus through a priority structure. Any channel request will cause the CPU to relinquish control of the IMB so that the request can be serviced.

Input/output

All access to input/output devices is by way of the device-independent MPE file system. All location of data, buffering, data transfers, and deblocking are handled automatically by MPE. All devices can be operated concurrently (within system bandwidth). Peripherals that fail are taken off-line from the operating system by operator command.

Peripheral I/O hardware

HP 3000 Series 39 and 42 peripheral I/O hardware consists of the General I/O Channels (GIC), DSN/Asynchronous Data Communication Controllers, DSN/Intelligent Network Processors, and the peripheral units. DSN/Asynchronous Data Communications Controllers interface log-on and data-entry terminals. Table G-5 lists the features which the peripheral I/O hardware offers.

**Table G-5 HP 3000 Series 39 and 42
Peripheral I/O Hardware Features**

-
- 4 ports per DSN/Asynchronous Data Communication Controller
 - Up to 8 terminal controllers per system
 - Bell type 103 and 212 modem support
-

When an I/O request is issued, the device driver in the CPU assembles the channel program, then issues a Start I/O Program (SIOP) instruction to one of two types of channels on the Intermodule Bus: the General I/O Channel (GIC) and the DSN/Asynchronous Data Communications Controller (ADCC). The GIC is the hardware I/O channel which provides the electrical interface between the computer system via the IMB and peripheral devices connected to the computer system Hewlett-Packard Interface Bus (HP-IB). The HP-IB is HP's computer system implementation of the IEEE standard 488-1975 interface, used on the Series 39 and 42 to connect peripheral devices to the channel. The HP-IB consists of eight data lines and eight control lines. The ADCC provides a bit-serial data interface between computer system and terminals. The two channels operate in a similar manner; however, the GIC has a DMA facility to permit high-speed transfer of large blocks of data, while the ADCC can transfer data only one character at a time.

General I/O Channel

The General I/O Channel (GIC) is the primary channel for communication between the CPU and the I/O devices other than terminals. Each GIC controls a computer system Hewlett-Packard Interface Bus (HP-IB) and translates I/O commands from the CPU into the proper HP-IB protocol. Nearly all transactions with I/O devices are accomplished without software interrupts, since I/O is achieved with channel programs. Software is responsible for setting up a channel program, but the execution of this program is performed by the CPU's channel microcode. The CPU's channel microcode is devoted to I/O tasks and implements the necessary algorithms for decoding the channel instructions and effecting the required I/O operations. Once the channel program is running, device control and data flow are normally carried to completion with no software intervention and without altering the system environment.

Several devices may simultaneously need service, and the CPU must decide which one will receive attention. First, all channels are polled, and the highest priority channel with a device request pending is chosen. The CPU then obtains from that channel the number of the

highest priority device needing service. Once the device number is determined, execution of the channel program will begin. The CPU fetches each channel instruction and breaks it down into several IMB commands addressed to the proper GIC. The GIC interprets these commands and directs them into the HP-IB device.

The GIC contains Direct Memory Access (DMA) hardware which allows large records of data to be transferred at the maximum speed of the HP-IB (about 1 Mb/second). The channel microcode enables the device and then initiates the DMA hardware on the GIC. After initial addressing of a device to talk or listen, the CPU relinquishes control of the IMB and allows the GIC to perform its function through DMA operation. During this time the GIC becomes the master of the bus and memory and controls traffic flow. On a read operation the DMA hardware will read the bytes, pack them into words and place them directly into memory, all without assistance from the CPU. The CPU is free to service other devices while DMA is in progress. Upon completion of a DMA transfer, the GIC returns to a slave condition and awaits the next operation.

Device controller

The device controller is the hardware linkage between a peripheral device and the computer system. Its primary function is to translate I/O commands from a General I/O Channel (GIC) to the unique signals required to control a particular device. When an I/O program is in execution, the device controller responds to and requests service from the GIC. The device controller also generates interrupts when required by some device condition or by channel command.

Device reference table (DRT)

Device controllers are identified by a logical device number which is used to access the device reference table (DRT). The DRT is known to both hardware and software, containing among other things, a pointer to the start of the SIO program for each device controller. Each device controller connects to a general I/O channel (GIC). Certain device controllers may control several logical devices. In such cases, each logical device attached to the controller is addressed separately using a unit number assigned when the device is installed.

Data service and interrupt priorities

In addition to a logical device number, there are two other characteristic numbers associated with each device controller. These are the data service priority and interrupt priority. In the Series 39 and 42; both of these are determined by the logical device number in the DRT: the lower the number, the higher the priority.

DSN/Asynchronous Data Communication Controller

The DSN/Asynchronous Data Communications Controller (ADCC) is the second channel type used in the system. This channel performs for terminals essentially the same functions as the GIC but not in the same manner. Data is transferred from memory to the ADCC in parallel form, then converted to a serial bit stream for transmission over the RS-232-C lines to the device. Information being read from a device is in serial form and is converted to eight-bit bytes for transfer to memory.

Two types of ADCC boards may be used, the Main ADCC and the Extender ADCC. Each board contains four ports for connections to devices through RS-232-C data communication lines. The Main ADCC supports full duplex (Bell type 103, 212, and 202T modems in the U.S.) only. The Main ADCC is used when four or fewer devices are connected to a channel. The Extender ADCC extends the device capability of the channel to eight. Most of the control circuitry is on the Main ADCC. For this reason, the Main ADCC is required for the Extender ADCC to function.

When more than eight devices are to be attached to an ADCC channel, additional Main ADCCs are required, since each ADCC can accommodate only one Extender.

Unlike the GIC, the ADCC does not have a DMA facility, and therefore, cannot be the master of an IMB or of memory. Also, terminals on the channel do not respond to a parallel poll. As a result, the ADCC is always a slave and must be directly controlled by the CPU through the use of channel programs. Circuitry on the ADCC decodes the address information relating to channels and devices, and selects the correct device for operation.

The ports on the ADCC (Main and Extender) may be either hardwired to devices or to full-duplex modems.



DSN/Intelligent Network Processor

The DSN/Intelligent Network Processor (INP) allows HP 3000 computers to be linked to other computers in a distributed data processing environment and supports multipoint terminals. The significant features of the DSN/Intelligent Network Processor are listed in Table G-6.

Table G-6 DSN/Intelligent Network Processor Features

-
- 16-bit SOS microprocessor
 - Data communications protocol handling
 - Character handling and 32 kbytes buffer storage
 - Modem and hardwired interfaces up to 56,000 bits/second
 - Full- and half-duplex asynchronous modem support
 - Bisync and HDLC/SDLC protocol compatible
 - RS-232-C, RS-422, CCITT V.24 and V.35 interfacing
 - Direct memory access for data
 - Auto call capability
-

The INP microprocessor performs all of the communication data link protocol management, including: serialization, protocol management, frame/block management, and data buffering. This reduces CPU utilization and frees it to perform other tasks. Throughput is increased by overlapping the transfer of data already received from the communication channel with the processing and buffering of new data coming from the communication channel. The protocol driver may be dynamically changed. This allows the INP to be easily reconfigured from one data link protocol to another and permits several subsystems to use a single INP. An auto call capability is included with the INP. It allows a remote connection to occur in a dial-up environment without the intervention of a human operator.

Interrupt system

The interrupt system provides for up to 105 external interrupt levels. When interrupts occur, the microprogrammed interrupt handler identifies each interrupt and grants control to the highest priority interrupt. Current operational status is saved in the microprogram, which then sets up the interrupt processing environment and transfers control to the interrupt routine.

Interrupt routines operate on a common stack (interrupt control stack) which is known to both hardware and software. This feature permits nesting of interrupt routines in the case of multiple interrupts, thus allowing higher priority devices to interrupt lower priority devices.

The interrupt system also provides for 20 internal interrupts (for user errors, system violations, hardware faults, and power fail/restart) plus fourteen traps for arithmetic errors and illegal use of instructions.

Peripherals

The peripheral devices used on the HP 3000 Series 39 and 42 are connected primarily to GICs, while the ADCC is reserved solely for terminals and remote printers. Peripherals attached to GICs through the HP-IB include disc drives, line printers, and magnetic tape drives. For a complete configuration of the supported peripherals on the HP 3000 Series 39 and 42, refer to the current HP 3000 Configuration Guide.

Automatic restart after power failure

An integral part of the HP 3000 Series 39 and 42 is a power fail/automatic restart capability. When the system AC line voltage falls below 90% of rated voltage, the system initiates a powerfail warning (PFW). During PFW the system (hardware and MPE) writes all register contents to a reserved section of main memory, activities in the system are successfully completed, and then the power down signal is generated and the system is shut down. The battery back-up power supply refreshes main memory and ensures its validity for up to 15 minutes, depending on memory size and battery condition.

The system is automatically restarted when power supply voltages reach 90% of their rated values and all register values are automatically restored and processing resumes.



Appendix H:
HP 3000 Series 39, 42 and 48
Machine Instructions

Stack Op Instructions

ADAX	Add A to X	FIXT	Fix and truncate
ADBX	Add B to X	FLT	Float an integer
ADD	Add A to B	FMPY	Floating point multiply
ADXA	Add X to A	FNEG	Floating point negate
ADXB	Add X to B	FSUB	Floating point subtract D, C - B, A
AND	Logical AND of A and B	INCA	Increment A
BTST	Test byte on TOS and set CC	INCB	Increment B
CAB	Rotate A-B-C	INCX	Increment X
CMP	Integer compare B, A and set CC	LADD	Logical add A + B
DADD	Double integer add D, C + B, A	LCMP	Logical compare B, A and set CC
DCMP	Double integer compare and set CC	LDIV	Logical divide C, B ÷ A
DDEL	Double delete TOS	LDXA	Load X into A
DDIV	Double integer divide	LDXB	Load X into B
DDUP	Double duplicate TOS	LMPY	Logical multiply B × A
DECA	Decrement A	LSUB	Logical subtract B - A
DECB	Decrement B	MPY	Multiply integers, integer product
DECX	Decrement X	MPYL	Multiply integers, long integer product
DEL	Delete TOS	NEG	Integer negate
DELB	Delete B	NOP	No operation
DFLT	Float a double integer	NOT	Logical complement TOS
DIV	Integer divide B by A	OR	Logical OR of A, B
DIVL	Divide long integer C, B ÷ A	STAX	Store A into X
DMUL	Double integer multiply	STBX	Store B into X
DNEG	Double integer negate	SUB	Integer subtract B - A
DSUB	Double integer subtract D, C - B, A	TEST	Test TOS and set CC
DTST	Test double word on TOS and set CC	XAX	Exchange A and X
DUP	Duplicate TOS	XBX	Exchange B and X
DXCH	Double exchange	XCH	Exchange A and B
DZRO	Push double zero onto stack	XOR	Logical exclusive OR of A, B
FADD	Floating point add, D, C + B, A	ZERO	Push integer zero onto stack
FCMP	Floating point compare and set CC	ZROB	Zero B
FDIV	Floating point divide D, C ÷ B, A	ZROX	Zero X
FIXR	Fix and round		

Shift Instructions

ASL	Arithmetic shift left	DLSR	Double logical shift right
ASR	Arithmetic shift right	LSL	Logical shift left
CSL	Circular shift left	LSR	Logical shift right
CSR	Circular shift right	QASL	Quadruple arithmetic shift left
DASL	Double arithmetic shift left	QASR	Quadruple arithmetic shift right
DASR	Double arithmetic shift right	TASL	Triple arithmetic shift left
DCSL	Double circular shift left	TASR	Triple arithmetic shift right
DCSR	Double circular shift right	TNSL	Triple normalizing shift left
DLSL	Double logical shift left		

Legend

TOS	Top of stack	A	Top of stack	D	Location below C
CC	Condition Code	B	Location below A	DB	Data Base
X	Index Register	C	Location below B	DL	Data Limit

Program Control and Special Instructions

DISP	Dispatch	PCN	Push CPU code (% 10)
EXIT	Exit from procedure	PSDB	Pseudo interrupt disable
HALT	Halt	PSEB	Pseudo interrupt enable
IXIT	Interrupt exit	RSW	Push cold load chan/dev
LLBL	Load label	SBM	Set Bank Mask
LLSH	Linked list search	SCAL	Subroutine call
PAUS	Pause, interruptable	SXIT	Exit from subroutine
PCAL	Procedure call	XEQ	Execute stack word

Machine and I/O Instructions

HIOP	Halt I/O program	SCLR	Set system clock limit
INIT	Initialize I/O channel	SED	Set enable/disable external interrupts
RCCR	Read system clock	SMSK	Set device mask
RIOC	Read I/O channel	TOFF	Hardware timer off
RMSK	Read device mask	TON	Hardware timer on
STRT	Programmatic warm start	WIOC	Write I/O channel
SIOP	Start I/O channel program	MCS	Memory controller read status
		DUMP	Programmatic dump

Loop Control Instructions

MTBA	Modify variable, test against limit, branch	TBA	Test variable against limit, branch
MTBX	Modify X, test against limit, branch	TBX	Test X against limit, branch

Memory Address Instructions

ADDM	Add memory to TOS	LDX	Load X
CMPM	Compare TOS with memory	LOAD	Load word onto stack
DECM	Decrement memory	LRA	Load relative address onto stack
INCM	Increment memory	MPYM	Multiply TOS by memory
LDB	Load byte onto stack	STB	Store byte on TOS into memory
LDD	Load double word onto stack	STD	Store double on TOS into memory
LDPN	Load double from program, negative	STOR	Store TOS into memory
LDPP	Load double from program, positive	SUBM	Subtract memory from TOS

Extended Instruction Set

Extended-Precision Floating Point		Decimal Arithmetic	
EADD	ADD	ADDD	Decimal add
ECMP	Compare	CMPD	Decimal compare
EDIV	Divide	CVAD	ASCII to decimal conversion
EMPY	Multiply	CVBD	Binary to decimal conversion
ENEG	Negate	CVDA	Decimal to ASCII conversion
ESUB	Subtract	CVDB	Decimal to binary conversion
		DMPY	Double logical multiply
		MPYD	Decimal multiply
		NSLD	Decimal normalizing left shift
		SLD	Decimal left shift
		SRD	Decimal right shift
		SUBD	Decimal subtract

Field and Bit Instructions

DPF	Deposit field, A bits to B	TCBC	Test and complement bit, set CC
EXF	Extract specified field, right-justify	TRBC	Test and reset bit, set CC
SCAN	Scan bits	TSBC	Test and set bit, set CC
TBC	Test specified bit and set CC		

Branch Instructions

BCC	Branch on specified CC	BRO	Branch on TOS add (bit 15 = 1)
BCY	Branch on carry	CPRB	Compare range and branch
BNCY	Branch on no carry	DABZ	Decrement A, branch if zero
BNOV	Branch on no overflow	DXBZ	Decrement X, branch if zero
BOV	Branch on overflow	IABZ	Increment A, branch if zero
BR	Branch unconditionally	IXBZ	Increment X, branch if zero
BRE	Branch on TOS even (bit 15 = 0)		

Move Instructions

CMPB	Compare bytes in two memory blocks	MVB	Move bytes in memory, addresses +/-
MABS	Move using absolute addresses	MVBL	Move words from DB+ to DL+ area
MDS	Move using data segments	MVBW	Move bytes while of specified type
MFDS	Move from data segment	MVLB	Move words from DL+ to DB+ area
MOVE	Move words in memory, addresses +/-	SCU	Scan bytes until test or terminal byte
MTDS	Move to data segment	SCW	Scan bytes while equal to test byte

Privileged Memory Reference Instructions

LDEA	Load double word from extended address	PSTA	Privileged store into absolute address
LSEA	Load single word from extended address	SDEA	Store double word into extended address
LST	Load from system table	SSEA	Store single word into extended address
PLDA	Privileged load from absolute address	SST	Store into system table

Immediate Instructions

ADDI	Add immediate to integer in A	LDXI	Load X immediate
ADXI	Add immediate to X	LDXN	Load X negative immediate
ANDI	Logical AND immediate with A	MPYI	Multiply immediate with A
CMPI	Compare A with immediate, set CC	ORI	Logical OR immediate with A
CMPN	Compare A with negative immediate	SBXI	Subtract immediate from X
DIVI	Divide immediate into A	SUBI	Subtract immediate from A
LDI	Load immediate to TOS	XORI	Logical exclusive OR immediate
LDNI	Load negative immediate to TOS		

Register Control Instructions

ADDS	Add operand to stack pointer	SETR	Set specified registers from stack
PSHR	Push specified registers onto stack	SUBS	Subtract operand from stack pointer
RCLK	Read clock	XCHD	Exchange DB and TOS
SCLK	Store clock		



30

H E W L E

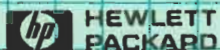
BATTERY
REMOTE

A B C D E F G H P R



H E

POV



HEWLETT
PACKARD

Corporate Headquarters:
Hewlett-Packard
3000 Hanover Street
Palo Alto, CA 94304 USA

European Headquarters:
Hewlett-Packard S.A.
150, Route du Nant-d'Auril
P.O. Box
CH-1217 MEYRIN 2
Geneva, Switzerland

Intercontinental Headquarters:
Hewlett-Packard
3495 Deer Creek Rd.
Palo Alto, CA 94304 USA

Printed in USA 8/83
5953-7553